HISTOGRAM-BASED IMAGE RETRIEVAL USING GAUSS MIXTURE VECTOR QUANTIZATION

Sangoh Jeong, Chee Sun Won†, and Robert M. Gray

Stanford University, Dept. of Electrical Engineering, Stanford, CA 94305, USA
†Dongguk University, Dept. of Electronic Engineering. Seoul, 100-715, Korea

ABSTRACT

Histogram-based image retrieval requires some form of quantization since the raw color images result in large dimensionality in the histogram representation. Simple uniform quantization disregards the spatial information among pixels in making histograms. Since traditional vector quantization (VQ) with squared-error distortion employs only the first moment, it neglects the relationship among vectors. We propose Gauss mixture vector quantization (GMVQ) as the quantization method for a histogram-based image retrieval to capture the spatial information in the image via the Gaussian covariance structure. Two common histogram distance measures are used to evaluate the similarity of histograms resulting from GMVQ. Our result shows that GMVQ with a quadratic discriminant analysis (QDA) distortion outperforms the two typical quantization methods in the histogram-based image retrieval.

1. INTRODUCTION

Despite the importance of quantization, only a few quantization methods have been considered in most papers dealing with histogram-based image retrieval[1][2]. One simple but popular method is uniform quantization of each color channel for every pixel, which has the defect of ignoring the interdependency among pixels. As a result, there have been various attempts to unify the color and the spatial information. In most cases, this has caused an increase in the complexity of comparison at querying time, an important factor in image retrieval.

Some authors[2] have suggested the use of the VQ approach represented by the generalized Lloyd algorithm[3]. Since VQ is an asymmetric compression scheme, its structure is suitable for image retrieval application requiring much faster comparison time than the calculation time of the features. Nevertheless, the Lloyd algorithm with mean squared error(MSE) as a distortion measure has a limitation in using the histogram of labels as a feature representation. Because the simple Lloyd algorithm uses only the first moment in its calculation, it cannot make full use of the spatial relationship among the blocks of pixels.

Recently, a collection of works dealing with a VQ using a Gauss mixture model have been developed. A GMVQ using a minimum discrimination information (MDI) distortion measure was introduced in [4] to motivate the use of a Gauss mixture model in robust compression systems. It was then further investigated in [5], with an emphasis on image compression and classification. Gray *et al.*[6] also demonstrated the potential for GMVQ using the MDI distortion for applications to image retrieval, classifying images with a simple decision tree.

In our work, we use the log-likelihood considered in [5] [6] as the distortion measure for GMVQ design in HSV color space. We use two well-known histogram distance measures to compare GMVQ with two other quantization methods. Since GMVQ exploits not only the mean vectors but also the covariance matrices, it incorporates the spatial characteristics of the images into the histogram better than the other quantization methods.

2. GAUSS MIXTURE VECTOR QUANTIZATION

2.1. Gauss Mixture Model

Although Gaussian densities have been popular for density estimation of texture images, they are not general enough to capture the multi-modal characteristics of generic color images. Thus, Gauss mixtures have been emerging as an effective density model of generic color images, for they can represent the multi-modalities.

A finite Gauss mixture model is a probability density of the following form:

$$f(\boldsymbol{x}) = \sum_{i=1}^{L} p_i g_i(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ represents a k-dimensional random vector, L is the number of the Gaussian components, and p_i represents the probability of the i^{th} Gaussian component. The density

We thank the NSF for funding.

$g_i(\boldsymbol{x})$ is the pdf of the i^{th} Gaussian component.

$$g_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{k}{2}}|\mathbf{K}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\mathbf{m}_i)^t\mathbf{K}_i^{-1}(\boldsymbol{x}-\mathbf{m}_i)} \quad (2)$$

where the $\mathbf{m}_i$ is the mean vector and $\mathbf{K}_i$ is a nonsingular covariance matrix.

2.2. Lloyd algorithm for a Gauss mixture model

In order to vector quantize an image, we first need to find a codebook using the training set of images. For a GMVQ, we have to obtain the covariance matrices, the probabilities, and the mean vectors of each cell. The steps of the Lloyd algorithm for the Gauss mixture source are as follows.

• *Step 1:* Set the iteration number $m = 0$, the distortion $D_{(0)} = 0$ and the threshold ϵ. Start with an initial set of Gaussian components $\{g_i\}_{(0)}$ with $i = 1, 2, \cdots, L$ and a set of training vectors $\{\boldsymbol{x}_n\}$, with $n = 1, 2, \cdots, N$, where L is the number of the Gaussian components making up of the Gauss mixture. N is the number of the training vectors.

• *Step 2:* Find the cells $\{Z_i\}_{(m)}$ that satisfy

$$\{Z_i\}_{(m)} = \{\boldsymbol{x}_n : \rho(\boldsymbol{x}_n, g_i, p_i)_{(m)} \leq \rho(\boldsymbol{x}_n, g_j, p_j)_{(m)}, \ \ \forall j \neq i\} \quad (3)$$

where $j = 1, 2, \cdots, L$. The $\rho(\boldsymbol{x}_n, g_i, p_i)_{(m)}$ is the Lagrangian distortion function given by

$$\rho(\boldsymbol{x}_n, g_i, p_i)_{(m)} \ = \ \{d_{LL}(\boldsymbol{x}_n, g_i) + \lambda \ln \frac{1}{p_i}\}_{(m)} \quad (4)$$

The left term is the LL distortion between a training vector and a Gaussian component.

$$\begin{aligned} d_{LL}(\boldsymbol{x}_n, g_i) &= -\ln \frac{1}{(2\pi)^{\frac{k}{2}}|\mathbf{K}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\mathbf{m}_i)^t\mathbf{K}_i^{-1}(\boldsymbol{x}-\mathbf{m}_i)} \\ &= \frac{1}{2}\big(k\ln(2\pi) + \ln|\mathbf{K}_i| + (\boldsymbol{x}-\mathbf{m}_i)^t\mathbf{K}_i^{-1}(\boldsymbol{x}-\mathbf{m}_i)\big) \quad (5) \end{aligned}$$

The right term of (4)represents the number of bits required by a noiseless code to specify *i* to the decoder, where λ is the Lagrangian multiplier and k is the dimension of a training vector. p_i is the probability that the training vector $\boldsymbol{x}_n$ is encoded to the Gaussian component g_i.

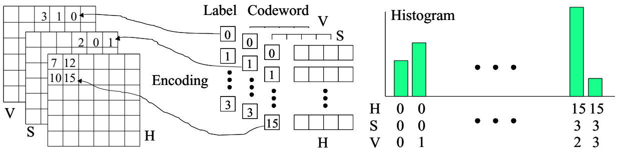
• *Step 3:* Compute the total distortion $D_{(m)}$ between the training vectors and the corresponding Gaussian components:

$$D_{(m)} = \sum_{i=1}^{L} \sum_{n:\boldsymbol{x}_n \in \{Z_i\}_{(m)}} \rho(\boldsymbol{x}_n, g_i, p_i)_{(m)} \quad (6)$$

• *Step 4:* If $\left|\frac{D_{(m)} - D_{(m-1)}}{D_{(m)}}\right| < \epsilon$, stop the process. Otherwise, continue.

• *Step 5:* Find the new values of the mean vectors, the covariance matrices and the probabilities used to define the Lagrangian distortion function.

$$p_{i(m+1)} = \frac{N_{i(m)}}{N} \quad (7)$$



(a) Labelled maps (b) Histogram of the labels

Fig. 1. Generation of histogram for MSE VQ and GMVQ

$$\{\mathbf{m}_i\}_{(m+1)} = \frac{1}{N_{i(m)}} \sum_{n:\boldsymbol{x}_n \in \{Z_i\}_{(m)}} \boldsymbol{x}_n \quad (8)$$

$$\{\mathbf{K}_i\}_{(m+1)} = \frac{1}{N_{i(m)}} \sum_{n:\boldsymbol{x}_n \in \{Z_i\}_{(m)}} (\boldsymbol{x}_n - \mathbf{m}_{i(m)})(\boldsymbol{x}_n - \mathbf{m}_{i(m)})^t \quad (9)$$

where $N_{i(m)}$ is the number of the training vectors in the cell $Z_{i(m)}$. Then, go to the *Step 2* with $m = m + 1$.

2.3. Quantizer for the Gauss mixture model

Once the Gaussian components and the related probabilities are obtained through the steps in the previous section, the vector quantizer maps an input vector into an index(label) of the closest Gaussian component. The distortion measure used in encoding can be the MSE for the image compression purpose as in [5]. However, for image retrieval, the same LL distortion is required in the encoding step for better discrimination between images.

3. GENERATION OF THE HISTOGRAMS

Generation of the histograms of color images in the SQ case is simple. All we have to do is to count number of pixels that correspond to a specific color in uniformly quantized color space, whether it is RGB color space or HSV color space. One histogram bin corresponds to one color in the quantized color space.

In the cases of the MSE VQ (VQ using MSE as a distortion measure) and the GMVQ, the histogram of the labels representing the Gaussian components is generated instead of the quantized pixel colors. In Fig. 1 (a), we can regard each label as a piece of information having the spatial variation of a color channel within a group of pixels. A combination of three labels representing a vector in the HSV color space constitutes one histogram bin as in Fig. 1 (b).

4. HISTOGRAM DISTANCES

There are several distance measures commonly used for finding the similarity between two color histograms [1]. We consider two famous histogram distance measures. Let H_1 and H_2 represent two color histograms of the query image and of an image in a database (DB), respectively. The Euclidean distance between H_1 and H_2 can be computed as

$$d_{HE}(H_1, H_2) = \sqrt{\sum_{X,Y,Z} (H_1(x,y,z) - H_2(x,y,z))^2} \quad (10)$$

where X, Y and Z are the gamuts of the discretized color channels. This distance is the L^2-norm.

The histogram intersection was proposed for color image retrieval in [7]. It was originally given by

$$I_{\mathrm{H}}(H_1, H_2) = \frac{\sum_{X,Y,Z} \min(H_1(x,y,z), H_2(x,y,z))}{\sum_{X,Y,Z} H_2(x,y,z)} \quad (11)$$

where H_1 is the histogram of a query image and H_2 is the histogram of an image in the DB. If the sizes of the histograms of the query image and the image in the DB are the same, the histogram intersection becomes equivalent to the L^1-norm[7].

Smith *et al.*[1] extended the definition to the case when the sizes of the two histograms are different, modifying the denominator of the original definition slightly as follows:

$$I_{\mathrm{H}}(H_1, H_2) = \frac{\sum_{X,Y,Z} \min[H_1(x,y,z),\ H_2(x,y,z)]}{\min[\sum_{X,Y,Z} H_1(x,y,z),\ \sum_{X,Y,Z} H_2(x,y,z)]} \quad (12)$$

Finally, the histogram intersection distance is defined as,

$$d_{HI}(H_1, H_2) = 1 - I_{\mathrm{H}}(H_1, H_2) \quad (13)$$

where $I_{\mathrm{H}}(H_1, H_2)$ = 1 when $H_1 = H_2$.

5. IMPLEMENTATION

The image DB used in this work contains generic images having both color and texture characteristics. It is the same as the DB exploited in [8]. The DB consists of 1000 color JPEG images whose size is either 384×256 or 256×384. It is a subset of the well-known Corel DB and has 10 classes with 100 images each. To reduce the computation time of the repeated experiment, the central region (256×256) of every image in the original DBs is selected and scaled to a 128×128 JPEG image.

In our image retrieval system, if a user selects a query image and a quantization method, the system returns several candidate images in order of similarity measured by a histogram distance between the histogram of the query image and the histogram of every image in the DB. We used uniform quantization for scalar quantization (SQ) as in [1]. The Lloyd algorithm using MSE was employed for VQ as in [2]. All procedures are realized using MATLAB.

In the case of SQ, the hue is quantized to 16 levels, the saturation to 4 levels, and the value to 4 levels for every pixel, for the hue is known to be most important in the HSV color model. In the MSE VQ case, a training set is used in generating a codebook. Here the training set consists of 16 images chosen from 1000 DB images. One or two images were taken from each of the 10 classes. The number 16 was selected from the simple performance comparison with other numbers of training images(10, 64 and 100) in the MSE sense. The codebooks for the three color channels in the HSV color space are designed separately in the way described in [3]. The dimension of a vector was determined to be 4 (2×2 square pixels) since the size of the images is only 128×128. In order to be equivalent to the SQ case, the sizes of the codebooks for the H,S, and V color channels are chosen as 16, 4, and 4, respectively. All images in the DB are then encoded with the codebooks. As a result, we get the labelled maps for the HSV color channels as in Fig. 1 (a), where each label represents a codeword with the dimension of 4.

Gaussian components for each color channel are calculated from the training set for the density estimation of the GMVQ. We set the number of the Gaussian components for each H, S, and V color channel to 16, 4 and 4 so that all three quantization methods can generate the same 256 histogram bins. The same vector dimension of 4 as for the MSE VQ was used. In order to get labelled maps, the encoding was done based on the Lagrangian distortion function between the input vectors and the Gaussian components. The Lagrangian multiplier λ was chosen as 1 since it gave the best visual quality of the encoded images among the values of $\lambda = 0, 1, 10, 100, 1000$.

After generating the histograms according to the section 3, we used two metrics to measure the retrieval effectiveness of an image retrieval system: one is *recall* and the other is *precision*[1]. In general, the *precision* and *recall* are used together in a graph so that they can show the change of the *precision* values according to the *recall* values. Since the *precision* drops typically as the *recall* increases, an image retrieval system is said to be more effective if the *precision* values are higher at the same *recall* values.

6. RESULTS

Fig. 2 shows the results of our work. Together with the retrieval performance using GMVQ, results for the other two quantization methods are displayed. It is the average of 1000 *precision vs.recall* graphs when all images in the DB are used as the query images alternately. The retrieval performance using GMVQ is the best among the performances using the three color quantization methods, based on both histogram distance measures. Considering the same number of histogram bins are used, we got a significant amount of improvement in retrieval performance using GMVQ. We can also observe that the performance of image retrieval using GMVQ is less dependent on the particular histogram distance measure used since the performance gap between

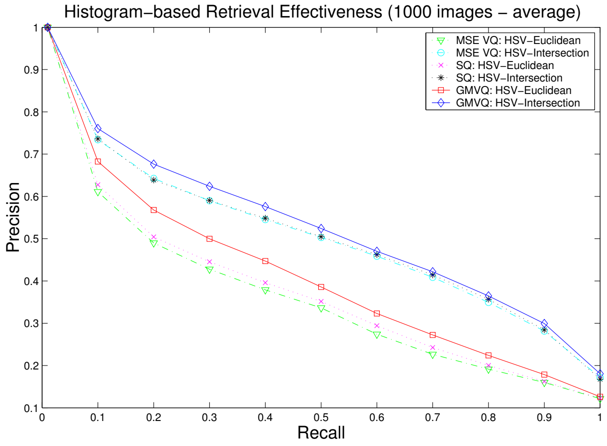


Fig. 2. Overall retrieval performances

the two histogram distances is much less than the other two quantization methods. There is little difference in retrieval performance between the MSE VQ and SQ in Fig. 2. Because MSE VQ does not use the covariance information between two vectors in generating the codebook or in encoding the images, it does not estimate the unforseen data accurately. Thus, it cannot overcome the decrease of the cardinality of histogram caused by using the vectors instead of the pixels, resulting in the similar performance to the SQ.

Regarding the retrieval complexity, SQ is the simplest in quantizing the query image, but it takes the longest time to calculate the histogram of all quantized pixel values. The MSE VQ and the GMVQ are complex in quantizing the query image[1], but they need much less time to calculate the histogram of the labels. In consequence, there is only a small difference in the complexity of generating the histogram of the query image among the three quantization methods at querying time. Moreover, the comparison complexity is smaller as long as the same histogram distance is employed since the three quantization methods use the same number of histogram bins. Therefore, in our experiment, the three quantization methods resulted in almost the same response time for a query image. In our retrieval system, the average response times for 5 arbitrary query images were 1.6846, 1.6862, 1.7008, 1.7042, 1.6828, and 1.6862 seconds from top to bottom in the legend of Fig. 2.

7. CONCLUDING REMARKS

We studied a new quantization method for histogram-based image retrieval, reflecting the spatial relationship on the labels of the histogram. GMVQ proved to be superior to other quantization methods in histogram-based image retrieval. We also observed that the GMVQ is less dependent on the different histogram-comparison measures than other quantization methods. Using a GMVQ, we yielded improvement in the retrieval performance for the same number of the histogram bins. We also reduced the possible range of each histogram value by $\log_2 k$ bits, where k is the vector dimension assumed to be a power of 2. The approach may be useful for implementing multimedia standards such as the MPEG-7[9], which defines a "Histogram Descriptor."

8. REFERENCES

[1] J. R. Smith and S. F. Chang, "Tools and techniques for color image retrieval," in *IST/SPIE - Storage Retrieval for Image and Video Databases IV*, San Jose, CA, February 1996, vol. 2670, pp. 426–437.

[2] F. M. Idris and S. Panchanathan, "Image and video indexing using vector quantization," *Machine Vision and Applications*, vol. 10, no. 2, pp. 43–50, 1997.

[3] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantization design," *IEEE Trans. on Communications*, vol. 28, no. 1, pp. 84–95, January 1980.

[4] R. M. Gray, "Gauss mixture vector quantization," in *Proceedings of IEEE ICASSP*, Salt Lake City, Utah, May 2001, vol. 3, pp. 1769–1772.

[5] A. K. Aiyer, *Robust Image Compression Using Gauss Mixture Models*, Phd thesis, Stanford University, August 2001.

[6] R. M. Gray, J. C. Young, and A. K. Aiyer, "Minimum discrimination information clustering: modeling and quantization with Gauss mixtures," in *Proceedings 2001 IEEE ICIP*, Thessaloniki, Greece, October 2001, vol. 2, pp. 14–17.

[7] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[8] J. Li, J. Z. Wang, and G. Wiederhold, "IRM:integrated region matching for image retrieval," in *Proceedings of the 2000 ACM Multimedia Conference*, Los Angeles, CA, October 2000, pp. 147–156.

[9] ISO/IEC, Sydney, Australia, *MPEG-7 Requirements Document V.15*, iso/iec jtc1/sc29/wg11/n4317 edition, July 2001.

[1] The GMVQ is more complex than the MSE VQ, but the computational time difference between them is negligible for a query image in comparison with the entire retrieval time