

A Study on the Effect of Codebook and CodeVector Size on Image Retrieval Using Vector Quantization

B. Janet*, A. V. Reddy

Dept. of CA, National Institute of Technology, Trichirappalli - 620015, India

Abstract In this paper, we study the effect of codebook size and codevector size using vector quantization (VQ) for retrieval of images, not restricted to the compressed domain. We use the image index model, to study the precision and recall values for different similarity measures. The study presents the following findings. The codebook size and codevector size are directly proportional to the precision value for a locally global codebook, but are dependent on the size of the source image for a local codebook. The Encoding Distortion similarity measure calculated from the local codebook produces the highest precision for the same recall over all other similarity measures. The Histogram Intersection using locally global codebook gives higher precision for higher codebook sizes. It is established that VQ can be used to create a single valued feature to represent the image in the image index model. This feature based on distortion measure can be effectively used for image retrieval based on the experimental results.

Keywords Image Processing, Image Retrieval, Information Retrieval, Indexing, Vector Quantization

1. Introduction

The image capturing devices have increased the number of images that are stored in the digital formats to such an extent that retrieval of the images has become a big problem. Content based retrieval[1] in large image databases has become the need of the hour. To facilitate the retrieval process, an index[2] of the image database is constructed that contains the similarity measure of the image to other images in the database.

Many image indexing methods, to enable fast retrieval of images, have been proposed. Keywords[1] which are manually assigned or automatically assigned from tag and title or as annotations are used to index an image. The drawback with keywords is that the manual process is user dependant as different users assign different keywords for the same image. Automatic tagging depends on proper naming convention. If name is not clearly assigned then, it cannot reflect on the content of image. Another method is to assign tags for an image based on the textual content[1] close to it in the document. The text around the image will be closely associated with the content of image. But advertisements and company logos may also be considered to represent the content of the image, in this case.

A second, widely researched area is based on feature vectors[1] derived from images. Histograms[3] give the

color content of the images. Segmentation and Edge detection[1] algorithms are used to trace edges and identify or classify the shape of the images. The above mentioned techniques do not use the spatial content of the image and the storage requirement is large.

Vector quantization (VQ)[4] is an efficient technique for low bit rate image compression. It is also an indexing technique where the image is represented as a set of code vectors. VQ represents the image as a codebook that reflects the intensity and spatial content of the image. The code book is a representation of the content of the image. We can use a local codebook[12, 13, 14], a global codebook[5, 6, 7], or a locally global codebook[20, 22] to represent the image.

In this paper, we give a brief description of how VQ is applied to create the index structure for images using various similarity measures. The precision and recall graph is analyzed and evaluated for the various codebook sizes and for various codevector sizes, computed for the same image database. The locally global and local codebooks are used for the experiments. The distortion with the codebook or the intersection distance of the histogram is taken as a single feature to represent the image in the image index model.

2. Previous Work

The past work on CBIR mostly represented the images by feature vectors. Color was the mostly used discriminating feature[8, 9]. Histogram is used to represent the color distribution in an image. To reduce complexity, the color space is first quantized into a fixed number of colors (bins) and images are compared by comparing their color histograms.

* Corresponding author:

janet@nitt.edu (B. Janet)

Published online at <http://journal.sapub.org/computer>

Copyright © 2011 Scientific & Academic Publishing. All Rights Reserved

Examples of schemes for image retrieval based on color histograms are [3, 8, 10]. An analysis of color histograms can be found in [11]. VQ based image classification and retrieval has been proposed in the past. As quantization is performed on image blocks instead of single pixels, some spatial information is implicitly included. Idris and Panchanathan [5] use VQ to index compressed images and video data. VQ is a lossy technique efficiently used for image compression to capture the content of image and maintain an acceptable quality based on the feature base. Images are initially compressed using a global VQ codebook [5, 6], then for each code vector in the codebook, a histogram of image blocks is generated (the number of blocks is taken to be the number of code vectors used in the image). Their results compare favorably to methods based on color histograms. Idris [6] used a global code book to represent a set of compressed images. The histogram of the code vectors of the image is used as an index to measure the similarity of the image to other images. Later in [5], a usage map is computed for the code vectors of a global code book. Similar images were found to have similar usage maps. It reduced the time to process a query as it is a simple XOR operation to find the similarity of an image.

With the advent of digital camera, the number of images stored in a database exploded. Then, it was not possible to create a single global codebook for the entire database. Schaefer [12] proposed an algorithm that used local (or individual) codebook for each image. The code book similarity was used as a measure of the Hadusdorff distance [13] and a median Hadusdorff distance and a modified Hadusdorff distance was proposed [13].

In 2005, Daptardar and Storer [14] proposed to find the Encoding distortion distance between the images for local codebooks. Jeong and Gray [15] used minimum distortion image retrieval using Gaussian mixtures. The image with the lowest distortion was the best match.

As the size of the database increased, global codebook generation was not possible. So, the local codebook was used. Later, individual codebook for each image was generated to calculate the distortion distance. Now, a locally global codebook [22] is generated using the incremental codebook generation process. This is representative of the global codebook generated already.

In this paper, we analyze the precision and recall curve for various codebook sizes generated, and use the similarity measures Hadusdorff distortion, Encoding distortion and Histogram Intersection for finding precision and recall from the image index model [17].

3. Vector Quantization

Vector Quantization [4] is used to quantize and compress the pixels of an image. An image is blocked into M , K -dimensional source vectors of pixel values. Using the LBG clustering algorithm [4], a codebook is computed as described below.

Consider M to be a source vector of the images, D to be a distortion measure, N to be the number of code vectors or the codebook size, and then we can compute the codebook C and a partition space P which result in the smallest average distortion.

An image is divided into blocks consisting of M source vectors called the training sequence $T = \{x_1, x_2, x_3, \dots, x_m\}$.

This training sequence can be obtained from the large image database used for content based retrieval. M is assumed to be sufficiently large, so that all the content of the images are captured by the training sequence. Let the source vector be K -dimensional given as $x_m = (x_{m,1}, x_{m,2}, x_{m,3}, \dots, x_{m,k})$, where $m=1, 2, \dots, M$.

Let N be the number of codevectors and let $C = \{c_1, c_2, c_3, \dots, c_n\}$ represent the codebook. Each code vector is K -dimensional, $c_n = (c_{n,1}, c_{n,2}, c_{n,3}, \dots, c_{n,k})$, $n=1, 2, \dots, N$.

Let S_n be the encoding region associated with code vector C_n and P be the partition of the space, then $P = \{S_1, S_2, S_3, \dots, S_n\}$.

If the source vector x_m is in the encoding region S_n , then its approximation (denoted by $Q(X_m)$) is C_n

$$Q(X_m) = C_n, \text{ if } X_m \in S_n \quad (1)$$

Let the average distortion be:

$$D_{ave} = \frac{1}{Mk} \sum_{m=1}^M \|X_m - Q(X_m)\|^2 \quad (2)$$

We calculate C and P such that D_{ave} is minimized for any given T and N . If C and P are to be found, then it must satisfy the following criteria.

The Nearest Neighbor Condition which states that the encoding region S_n should consist of all source vectors that is closer to C_n than any of the other code vectors. For those vectors lying on the boundary, the one with the smallest distortion value is considered.

$$S_n = \{x: \|x - c_n\|^2 \leq \|x - c_{n'}\|^2 \forall n' = 1, 2, \dots, N\} \quad (3)$$

The LBG VQ algorithm is an iterative algorithm which solves the above two optimality criteria. The algorithm requires an initial codebook C_0 . This initial codebook is obtained by the splitting method. In this method, an initial code vector is set as the average of the entire training sequence. This code vector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two code vectors are split into four and the process is repeated until the desired number of code vectors is obtained. When a codebook is defined, then the source vectors of the image can be mapped to code vectors of the codebook according to the nearest neighbor rule.

We use the LBG VQ algorithm to generate the local codebook for each image. It is also used to generate the locally global codebook for the entire set of images based on the incremental codebook generation process.

3.1. Similarity Measures

3.1.1. Histogram Intersection (HI)

We can compute the histogram [15] of an image i for a

locally global codebook of the entire image database. It is a N dimensional vector $\{H_i : i=1,2,\dots,N\}$ where H_i is the number of source vectors that fall into the encoding region of a code vector. N is the number of codevectors in the code book. The Euclidean distance[13] is calculated between two image histograms h and g as

$$d^2(h, g) = \sum (h(x) - g(x))^2 \quad (4)$$

Where $h(x)$ is the histogram of one image and $g(x)$ is the histogram of the other image.

The intersection distance[15] is calculated as

$$d(h, g) = \frac{\sum \min(h(x), g(x))}{\min(|h|, |g|)} \quad (5)$$

$|h|, |g|$ are the magnitude of each histogram which is equal to number of samples.

Using the locally global codebook[22], a histogram of codevectors for an image i is calculated as $H_i = (h_1, h_2, \dots, h_{256})$. We find the feature vector for all the S codevectors as histogram Intersection distance HI_{ij} between two images i and j in the database for all the images using equation 5.

3.1.2. Encoding Distortion (ED)

Schaefer[13] uses a local codebook for each image as its feature vector. To determine the image similarity based on encoding distortion distance (EDD)[13], codebooks of images are directly compared with the images as

$$d_{EDD}(A, B) = \max\{d_{ED}(A, C_B), d_{ED}(B, C_A)\} \quad (6)$$

where the ED is determined by

$$d_{ED}(X, C) = \frac{1}{MK} \sum_{i=1}^M \min_j \|x_i - y_j\|^2 \quad (7)$$

is the average per component squared Euclidean distance between source vectors $X = \{x_i\}_{i=1}^M$ and codevectors $C = \{y_j\}_{j=1}^N$. The ED[21] is used as a measure to determine the similarity of the images.

3.1.3. Hausdorff Distortion (HD)

Modified Hausdorff Distance[12] compares two local code books to measure their similarity. Let C_A and C_B be code books for the images A and B. Then Hausdorff Distance is calculated as

$$d_{HD}(C_A, C_B) = \max\{d_{HD}(C_A, C_B), d_{HD}(C_B, C_A)\} \quad (8)$$

Where the Hausdorff distortion (HD) is determined as

$$d_{HD}(C_A, C_B) = \frac{1}{N} \sum_{i=1}^N \min \|C_A(i) - C_B(i)\|^2$$

The HD[21] is used as a similarity measure to determine the similarity of the images.

3.1.4. Incremental Codebook Generation

The codebook for an image can be computed either for each image (local)[12, 13, 14] or one universal or common codebook for the entire set (global)[5, 6, 7] using the LBG VQ algorithm. The technique for finding the codebook

which combines both the above mentioned techniques and generates a locally global codebook[22] is shown in figure 1.

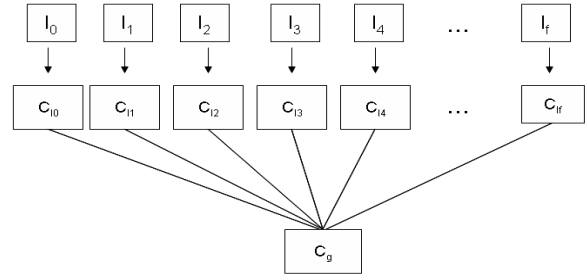


Figure 1. Incremental Codebook Generation.

Let I_i where $i=1$ to f images, be the blocked images in the database. Using the LBG VQ algorithm, the local codebooks are generated for all the images as C_i . The codevectors in all the codebooks are given as a training sequence for LBG algorithm and one codebook C_g that is locally global is generated. This C_g is a universal representation of the entire database of images. The need for a representative set of images is removed.

The incremental codebook generation will help in the generation of a single codebook that represents the entire set of images, which will help with the retrieval of large scale images.

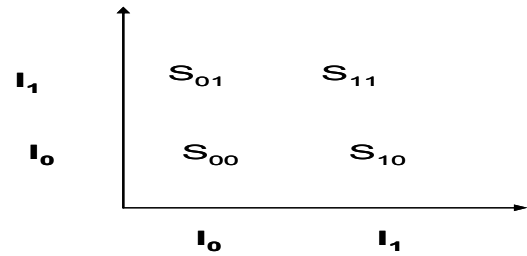


Figure 2. Image Index Model.

3.2. Indexing Algorithm

Step 1: Preprocess

We resize the image to 256 x 256 pixels. Then we convert it to 8 bit grayscale image.

Step 2: Blocking the image

We divide the image into $K \times K$ block vectors and store the image block vectors as source vector or Training sequence. For example, if $K=4$, then for an image, P_i is the pixel value in an image and X_i is the block vector, then $T = \{X_1, X_2, X_3, \dots, X_M\}$, where $X_i = \{P_{i1}, P_{i2}, \dots, P_{iK}\}$.

P_1	P_2	P_3	P_4	---->	P_1	P_2	P_5	P_6	----->	X_1
P_5	P_6	P_7	P_8	---->	P_3	P_4	P_7	P_8	----->	X_2
P_9	P_{10}	P_{11}	P_{12}	---->	P_9	P_{10}	P_{13}	P_{14}	----->	X_3
P_{13}	P_{14}	P_{15}	P_{16}	---->	P_{11}	P_{12}	P_{15}	P_{16}	----->	X_4

Step 3: Codebook generation

We apply LBG VQ algorithm to T of each image i, to find the codebook C for each of the image in the database.

$C = \{C_1, C_2, \dots, C_B\}$ where B is the total number of images and each $C_i = \{C_{i1}, C_{i2}, \dots, C_{iS}\}$ where S is the codebook size used in the experiments. C_{ij} is the codevector of the codebook

The codebook is generated to be Local, Global and Locally global. The size of the codebook is varied as 8, 16, 32,

64, 256.

Step 4: Similarity measure

Similarity between two images i and j is calculated using any one of the similarity measures described in section 3. It is stored as an Image Index model as shown in figure 2.

3.3. Retrieval Algorithm

If the query by example image is in the database, then

- Retrieve the list of similarity values from the database
- Sort the index list in ascending order
- Top K images are relevant to the query image
- Return the list.

If query by example is not present in the database, then

- Repeat steps 1-4 for the image
- Add the image index list to the index
- Sort the list
- Top K images are relevant to the query
- Return the list

3.4. Index Structure

The code books are stored as a direct index using Terrier 2.1[18] Direct Index structure. Terrier is the Terabyte Retriever version 2.1 developed by University of Glasgow - Department of Computing Science, Information Retrieval Group. The open source version of Terrier is written in Java. Terrier was used to provide the direct index for a text database. As image was also considered for processing along with the text, terrier was also used for images to store the features as a direct index structure.

Direct index for an image consist of the image identifier and the code book vectors values for that image. The Document Index stores the Image information such as the Image identifier, Image title, path and offset of the Image codebook information in the direct file.

The Image index is created by calculating the similarity measure of the image i and j and is stored in the image index as S_{ij} .

4. Experiment and Evaluation

Terrier 2.1 is modified to implement an image direct and document index in Pentium 4 processor with 3.2 GHz and 1 GB RAM.

To test the index, a sample of the database due to Wang, Li, Wiederhold[19] available on the web is used. The database consists of 1000 jpeg images which are either 256*384 or 384x256 with 100 images per class. The classes are symmetrically classified as: Africans, Beach, Architecture, Buses, Dinosaurs (graphic), Elephants, Flowers, Horses, Snow Mountains and Foods. Retrieval effectiveness is evaluated using two standard quantities: precision and recall.

For a given query, let

- a be the number of relevant images that are retrieved,
- b , the number of irrelevant items,
- c , the number of relevant items that were not retrieved

Then:

Precision = fraction of the images retrieved that are relevant. = $a/(a+b)$

Recall = fraction of the relevant images that are retrieved = $a/(a+c)$

IR is said to be more effective if precision values are higher at the same recall values.

The average precision is used to evaluate a ranked list. For ranked retrieval, the precision and recall is calculated for each rank. Then the average precision at ranks where relevant images occurred is calculated for a given recall and a precision vs recall graph is plotted.

In our experiment, we use various image block size for grayscale images. The codebook size is also varied. Each image is given as a query by example, and the average precision and recall curves calculated for various similarity measures.

4.1. Code Book Size

We now consider the codevector size of 2x2 which gives a vector length of 4 for each block. Figure 3 shows the ED measure of similarity, Precision Recall (PR) curves which show that there is increase in precision for the same recall, when the code book size is increased. Thus local codebook gives a higher precision when the codebook size is increased when using the ED Measure.

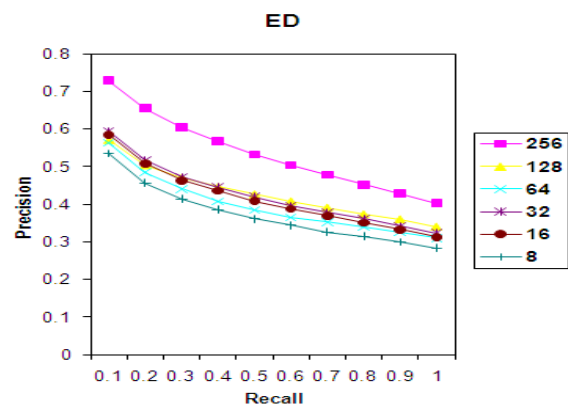


Figure 3. PR Graph for ED Measure.

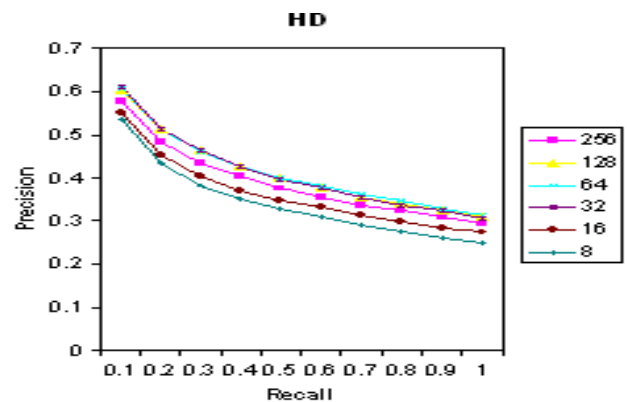


Figure 4. PR Graph for HD Measure.

Figure 4 shows the PR curves for the HD similarity measure. This is also found to be increasing till codebook of

size 64 and then it decreases. It shows that the size of the image considered as 256x256 impacts the precision based on the codebook size. It has a lower PR value than that of ED curve.

In figure 5, The MHD similarity measure is used. It is found to be slightly of lower precision than the HD measure. It also shows a decrease in precision, after a codebook size of 64.

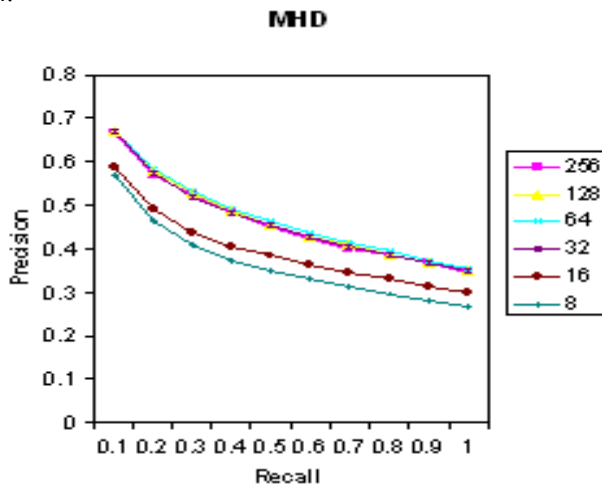


Figure 5. PR Graph for MHD Measure.

In figure 6, the HI measure shows that for lower codebook sizes the HI performs better for higher codebook size. Thus HI can be used for lower code book sizes of 8 or 16 to get the similar precision.

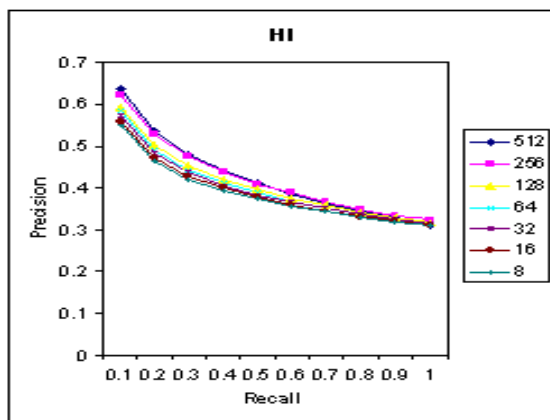


Figure 6. PR Graph for MHD Measure.

Figure 7 compares the HD, MHD and HI similarity measure PR average for a recall of 10 for various code book sizes. The precision decreases with the decrease in code book size. Larger codebooks give higher precision values for the same recall for a locally global codebook. For a local codebook, there is a decrease in precision after codebook size of 64 for the HD and MHD measure.

4.2. Code vector Size and Edge Detection

We now also consider the code vector size of 2x2 which gives a vector length of 4 for each block. Figure 8 and 9 shows the Precision curve for the HD and HI similarity measure for code vector sizes of 4 and 16. We can conclude

that the HD measure and HI measure give similar precision but in HD, a precision of 6.5 is obtained at the codebook size of 64 and in HI, at a codebook size of 256.

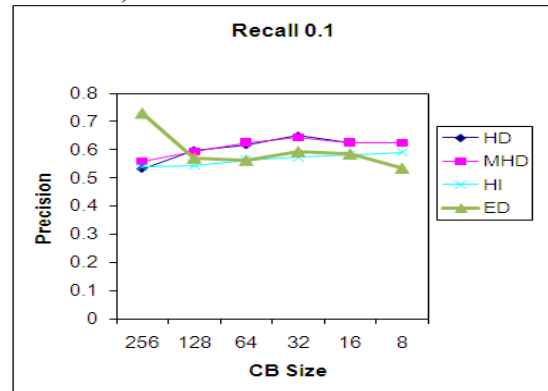


Figure 7. Precision for various similarity measures.

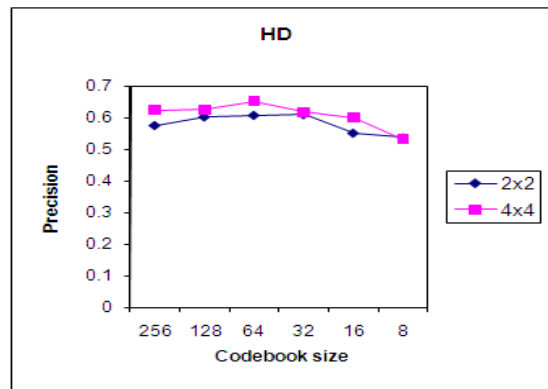


Figure 8. Precision using HD for various code vector sizes.

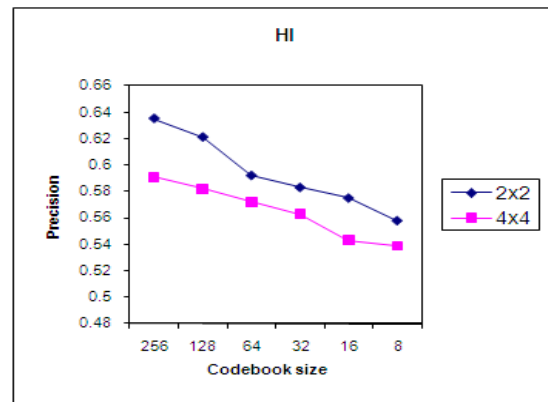


Figure 9. Precision using HI for various code vector sizes.

In figure 10 and 11, the code vector size of 8 and 16 are used to calculate the HI measure and Sobel's edge detection algorithm was applied for all the categories. Only for the buses and flowers, there is a marked increase in the precision average for a code book size of 8 and 16. It is compared with HI measure with no edge detection applied.

From the analysis, we can conclude that the precision and code book size are directly proportional and the code vector size helps to increase the precision. There is no change in precision of the MHD and HD measure when edge detection is applied to all the images in a preprocessing step. But similar images with dark or light background and images with

lesser detail can use edge detection to increase the precision using HI measure.

When the images are transformed as flip or rotate, there is no change in the similarity measure as the blocks remain the same in the image semantics.

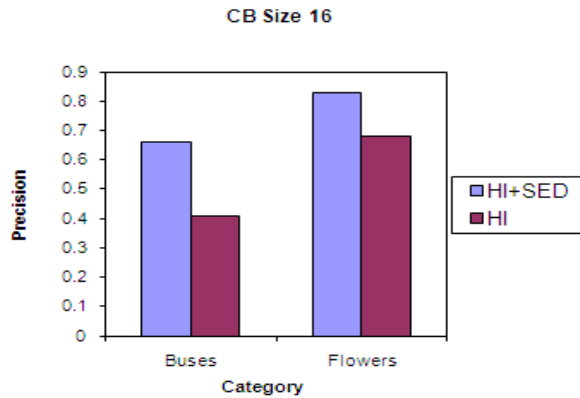


Figure 10. Precision using HI for two categories.

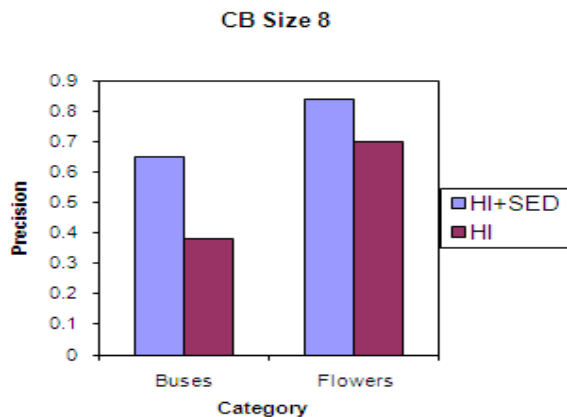


Figure 11. Precision using HI for two categories.

5. Conclusions

The study on the effect of various codebook sizes and the codevector sizes concludes that the codebook size and code vector size are directly proportional to the PR value but are dependent on the size of the sample image for a local codebook. ED can be used if the precision is to be higher and HI if the recall is to be higher.

Edge detection algorithm applied in the preprocessing step helps to increase the precision of images with uniform background. The ED gives the highest precision for the same recall values over all similarity measures. HI gives higher precision for larger codebook sizes.

Thus, Vector quantization can be used for image retrieval where each image is represented as a single value of distortion measure.

REFERENCES

[1] Ritendra Datta, Dhiraj Joshi, Jia Li, And James Z. Wang,[2008]

- Image Retrieval: Ideas, Influences, and Trends of the New Age, ACM Computing Surveys, Vol. 40, No. 2, Article 5,
- [2] G. Salton and C. S. Yang,[1973]On the Specification of Term Values in Automatic Indexing, Journal of Documentation, Vol. 29, No. 4, December, pp. 351-372.
- [3] M. J. Swain and D. H. Ballard[1991]. "Color Indexing", International Journal of Computer Vision 7:1, 11-13.
- [4] R. M. Gray, "Vector quantization," IEEE Acoustics, speech and Signal Processing Magazine, pp. 4-29 (1984)
- [5] F. Idris and S. Panchanathan[1995]. "Storage and Retrieval of Compressed Images", IEEE Transactions on Consumer Electronics, 43:5, 937-941
- [6] F. Idris and S. Panchanathan[1997]. "Image and Video Indexing Using Vector Quantization", Machine Vision and Applications 10:2, 43-50.
- [7] G. Lu and S. Teng[1999]. "A Novel Image Retrieval Technique Based on Vector Quantization", Proceedings of International Conference on Computational Intelligence for Modeling, Control and Automation, 36-41.
- [8] B. Furht, S. W. Smoliar and H. Zhang[1995]. Video and Image Processing in Multimedia Systems, Kluwer.
- [9] T. Gevers[2001]. "Color in Image Search Engines", Principles of Visual Information Retrieval, M. S. Liew Editor, Springer
- [10] J. R. Smith and S. Chang[1996]. "VisualSEEK: A Fully Automated Content-Based Image Query System", Proceedings of ACM International Conference on Multimedia 96, 87-98.
- [11] M. Stricker and M. Swain[1994], "The Capacity of Color Histogram Indexing", IEEE Conference on Computer Vision and Pattern Recognition, 704-708
- [12] Schaefer, G. (2002) Compressed Domain Image Retrieval by Comparing Vector Quantization Codebooks. In: Proceedings of the SPIE Visual Communications and Image Processing, Volume 4671, 959-966,
- [13] Schaefer, G., Naumienko, W. (2003) Midstream Content Access by VQ Codebook Matching. In: Imaging Science, Systems and Technology. Volume 1. 170-176
- [14] Daptardar Ajay H., Storer James A., (2005) Content Based Image Retrieval via Vector Quantization, ISVC 2005 LNCS 3804, pp 502-509
- [15] S. Jeong and R. M. Gray[2005]. "Minimum Distortion Color Image Retrieval Based On Lloyd-Clustered Gauss Mixtures", Proc. IEEE Data Compression Conference, 279-288.
- [16] Y. Linde, A. Buzo, and R. M. Gray,[1989] An algorithm for Vector Quantizer design. IEEE Trans. Communications, 28:84-95,
- [17] B. Janet, A.V. Reddy, and S. Domnic, *An Image Index Model for Retrieval*, Communications in Computer and Information Science BAIP, CCIS 70, pp. 465-467.
- [18] Terrier 2.1, <http://ir.dcs.gla.ac.uk/terrier/>
- [19] J. Z. Wang, J. Li and G. Wiederhold[2001]. "SIMPLiCity: Semantics-Sensitive Integrated Matching for Picture Libraries", IEEE Transactions on Pattern Analysis and Machine Intelligence 23:9, 947-963. Web: <http://wang.ist.psu.edu/docs/related/>

- [20] B. Janet, A.V. Reddy, and S. Domnic, *Vector Quantization based Index Cube Model for Image Retrieval*, Proc. PSIVT 2010, Singapore (in press)
- [21] B. Janet, A.V. Reddy, *Image Index model for Retrieval using Hausdorff Distortion*, Proc. ICCAIE 2010, Malaysia (in press)
- [22] B. Janet, A.V. Reddy, and S. Domnic, *Incremental Codebook Generation for Vector Quantization in Large Scale Content Based Image Retrieval*, Proc. IEEE Int. Conf. CISR 2010, India