ISSN: 0976-9102(ONLINE)
DOI: 10.21917/ijivp.2012.0056

LOCALLY GLOBAL CODEBOOK FOR IMAGE RETRIEVAL AND CLUSTERING USING VECTOR QUANTIZATION

B. Janet[1], A.V. Reddy[2] and S. Domnic[3]
Department of Computer Applications, National Institute of Technology Tiruchirappalli, India
E-mail: [1]janet@nitt.edu, [2]reddy@nitt.edu and [3]domnic@nitt.edu

Abstract
In this paper, the incremental codebook generation process, which is a technique for representing a database of images as a single codebook, that captures the content of all the images is proposed. Vector quantization (VQ) is used for creating the codebook of the image. The main problem with VQ is the size of the training sequence that is used to generate the global codebook. This paper explains a method, where the local codebooks are generated for each image. Then, the locally global codebook for the images is computed from the local codebooks by the incremental process. In order to handle large number of images of similar classes, a new method of incrementally creating locally global codebook based on hierarchy of image classes is also explained. It gives an automated representation of the image database compared to the previous work of selecting a random set of representative images, to compute the global codebook. The locally global codebook is used to cluster the images using the encoding distortion. This codebook helps to match images, and retrieve images that are similar to it by comparing the encoding distortion. The precision and recall curve and mean average precision gives better results than the previous works. The classification error is reduced as the distortion is always the smallest for similar images.

Keywords:
Vector Quantization, Codebook Generation, Content Based Image Retrieval, Cube Index Model, Image Clustering

1. INTRODUCTION

With the advent of digital and mobile cameras, there has been an explosion in the number of images stored in a memory device. To sift through this large number of images and retrieve the required or relevant image has become a herculean task. The need is an image index structure [1] like the text index structures that indexes the features of the image. But an image index structure is difficult to create and maintain like the text index structure, as the image features are represented as a set and not as a single value.

Vector quantization (VQ) [2,3,4,5] is a technique used in image compression for compressing the large number of feature vectors generated. It is used in SIFT keypoint descriptor [6, 7] compression also. VQ is an indexing technique where the image is represented as a set of code vectors. It gives lossy encoding of the image, as a codebook that reflects the pixel intensity and spatial content of the image. This codebook has helped in the content based image retrieval tasks [8,9,10,11,12,13,14,15].

VQ is used to compress the feature vectors or reduce the number of features representing the image in applications [13,16]. Past work on information retrieval using vector quantization used the global codebook [12,17,18,19] in 1999, and later, the local codebook [14,15] in 2005 for each image. The disadvantage of a local codebook is that it cannot be used to compare with the entire database. As the size of the database increased, randomly user selected images [12,19] were taken to generate the global codebook. This is not auto generated as the set is randomly user picked images.

This paper explains a locally global codebook generation process that is incremental or hierarchical and automatic based on the class of images present. It can be applied to any type of image of the compressed or uncompressed domain. The experimental results are compared for the image database and the retrieval efficiency is studied for an image index model that uses the locally global codebook.

1.1 MOTIVATION

The human brain has the ability to store and retrieve millions of data over a period of time. The image data is stored as a rich set of features of the image interlinked with the text and other sensory input, as an experience. This experience when remembered or recalled based on either textual information or images of the scene or voice tracks that were heard in that place or clips of video of that experience, is retrieved accurately. This is because the human brain retrieves the information from the single storage structure.

If a text of the name of a person is input, the picture of the face of the person immediately pops out. Given an audio clip, the lyric (as text) comes out with the picture of the person, who wrote the lyric. The brain correctly tells the description of the park where the incident happened. In order to represent the image as a single value, VQ is used to create an image index structure [1] similar to a text index structure [20]. A cube index model [21] was proposed to include both text and image as a single structure. The problem encountered was to create a single codebook for the image database, which was not possible. So, the incremental codebook generation process has been proposed.

1.2 HIGHLIGHTS

A single codebook is generated for a large image database automatically using the incremental codebook generation process. It propels research in using VQ for image retrieval, which stopped in 2005, due to lack of technological support for the large amount of processing required by this method. In VQ, the encoding distortion value represents the image. It is a single feature used to represent the image. It is used to cluster, match and retrieve the images. It can be used for image compression to transmit one codebook for the entire image database.

2. REVIEW OF LITERATURE

Idris and Panchanathan [17] used VQ to index compressed images and video data. Images were initially compressed using a single universal (global) VQ codebook [12,17,18,19]. Idris [17] used a global codebook to represent a set of compressed images.

The histogram of the code vectors of the image was used as an index to measure the similarity of the image to other images. Later in [18], a usage map $U_i = \{u_{ij}: 0 \leq j \geq N-1\}$ where u_{ij} takes a Boolean value 1 or 0, is computed for the code vectors of a global codebook. The experiment was carried out in a database of 55 images.

But, as the number of images stored became large, computation of a single codebook became impossible. Then Schaefer [14] proposed an algorithm that used local (or individual) codebook for each image. Each local codebook represented the content of the individual image respectively. The codebook similarity between similar images was used as a measure of the Hadusdorff distance and a median Hadusdorff distance and a modified Hadusdorff distance was proposed [15]. The experiment was carried out in a test database of 80 image pairs and 160 images only.

In 2005, Daptardar and Storer [22] proposed to find the Encoding distortion distance between the images for local codebooks and tested it on a test bed of 1000 images. Jeong and Gray [11,12] used minimum distortion image retrieval using Gaussian mixtures. The image with the lowest distortion was the best match. All the experiments with this database used only local codebooks.

A representative codebook from a set of images was used as a global codebook for a database of 1000 images [12,19]. The images were selected based on user preference and not automatically.

Later, VQ was used to reduce the number of descriptors used in SIFT image retrieval where a codebook of the descriptors was generated [22].

After 2005, to the best of our knowledge, VQ has not been used for image retrieval, as it required more processing and the codebook could not be generated, due to the large size of the training vector.

In the search for a single value representation of the image, VQ provided the answer with the encoding distortion value. To find out the histogram of the images, a single global codebook was needed. To reduce the number of source vectors used to create the global codebook, the incremental codebook generation method was proposed.

3. INCREMENTAL CODEBOOK GENERATION

Example 1: The image *i* is divided into m block vectors of size *b*x*b*. The image block vectors are stored as source vector or training sequence T. For example, consider a 4x4 image as shown below, with block of size 2x2.

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

where, P_i is the pixel value in an image and X_i is the block vector, and *m* is the number of block vectors in the image.

Consider *n* images stored in the database. Each image *i* is divided into *m* block vectors of size *b*x*b*. The block vectors of the image *i* is $B_i = \{X_1, X_2, X_3 \ldots. X_m\}$.

Then, source vector, $T = \{B_1, B_2, B_3 \ldots. B_n\}$, where *n* is the total number of images in the database.

The LBG VQ algorithm is applied to T to find the global codebook C_G for all the images in the database. $C_G = \{C_1, C_2, \ldots, C_{256}\}$ (for a codebook size of 256). The size of the codebook can be varied.

3.1 GLOBAL CODEBOOK

Universal or global codebook [12,17,18,19] can be generated for the entire set of images in the database using the LBG VQ algorithm as shown in example 1. In this case, the training sequence consists of all the block vectors of all the images in the database. If the size of the database is large, then the training sequence is large. Then, it becomes difficult to handle the training sequence.

For example, consider a database with 1000 images of size 4x4. If the block size is 2x2, then each image has 4 block vectors. Thus, the training sequence has 4x1000 block vectors. If the image database size is 1 million, with the size of the image as 512x512, then the size of the training sequence makes it difficult to find a codebook for the database.

Example 2: The image *i* is divided into *m* block vectors of size *b*x*b*. The image block vectors are stored as source vector or training sequence T. For example, consider a 4x4 image as shown below, with block of size 2x2.

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

where, P_i is the pixel value in an image and X_i is the block vector, then $T = \{X_1, X_2, X_3 \ldots. X_m\}$, and *m* is the number of block vectors in the image.

The LBG VQ algorithm is applied to T of the image *i*, to find the local codebook C_{Li} for each of the image in the database.

The local codebook for the database is $\{C_{L1}, C_{L2}, \ldots, C_{Ln}\}$, where *n* is the number of images in the database.

Each $C_{Li} = \{C_{i1}, C_{i2}, \ldots, C_{i256}\}$ (for a codebook size of 256). The size of the codebook can be varied.

3.2 LOCAL CODEBOOK

The local codebook [1,14,15,22] for an image is computed as shown in example 2. The training sequence is the block vectors of the image. The codebook represents the content of the image only. Similar images have similar encoding distortion values.

For example, consider a 4x4 image. If the block size is 2x2, then it has 4 blocks in the image. Thus, the training sequence T has 4 block vectors. With the training sequence T, the codebook for the image is computed using the LBG VQ algorithm. This codebook is representative of the content of the image for which it is generated.

Example 3: The image *i* is divided into *b* x *b* block vectors. The image block vectors are stored as source vector or training sequence $T = \{X_1, X_2, X_3, \ldots, X_m\}$, and *m* is the number of block vectors in the image.

The LBG VQ algorithm is applied to T of each image *i*, to find the local codebook C_{Li} for each of the image in the database. The local codebook for the database is found as $\{C_{L1}, C_{L2}, \ldots, C_{Ln}\}$ where *n* is the number of images in the database and each $C_{Li} = \{C_{i1}, C_{i2}, \ldots, C_{i256}\}$ (for a codebook size of 256). The size of the codebook can be varied.

Consider *n* codebooks for the images, then, source vector, $T = \{C_{L1}, C_{L2}, \ldots, C_{Ln}\}$.The LBG VQ algorithm is applied to T to find the locally global codebook C_{LG} for the database. $C_{LG} = \{C_1, C_2, \ldots, C_{256}\}$.

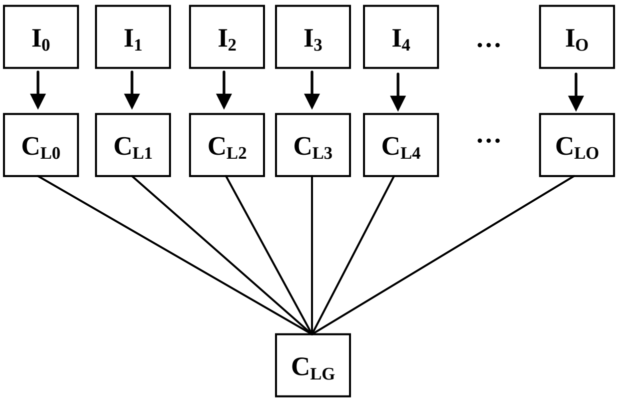

Fig.1. Locally global codebook generation

3.3 LOCALLY GLOBAL CODEBOOK

The technique for finding the codebook, which combines both the local and global codebook generation process and generates a locally global codebook [23,24] as shown in Fig.1 can be used to solve the problem for large image databases. The training sequence consists of the codevectors in the codebook of each individual image as explained in example 3. The number of vectors in the training sequence is reduced. This increases the distortion slightly, but preserves the image content needed for retrieval and clustering applications.

For example, consider a database of 1000 images of size 4x4. If the block size is 2x2, then each image has 4 blocks. A codebook of size 2 for each of the image is constructed. Now, there are 2x1000 codevectors, instead of 4x1000 block vectors in the training sequence. This is used to generate the locally global codebook.

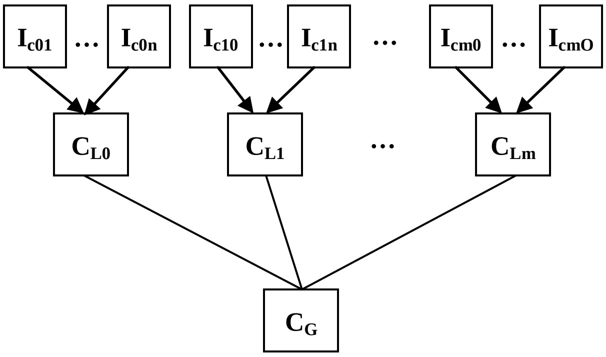

Fig.2. Incremental codebook generation

3.4 INCREMENTAL CODEBOOK GENERATION

When there is a classified image database, the locally global codebook can be generated incrementally. If there are *m* classes, then a codebook is generated for each of the *m* class of images. Now all the *m* codebooks are combined to generate one global codebook as shown in Fig.2 and explained in example 3. The levels in the hierarchy can be varied according to the size of the class to give a hierarchy of codebooks.

4. RESULTS AND DISCUSSION

4.1 BENCHMARK DATABASE

To test the index, three different types of database are used. The database due to Wang, Li, Wiederhold [25] available on the web is used. The database consists of 1000 jpeg images which are either 256x384 or 384x256 with 100 images per class. It belongs to the category of stock photo retrieval task. The classes are symmetrically classified as: Africans, Beach, Architecture, Buses, Dinosaurs (graphic), Elephants, Flowers, Horses, Snow Mountains and Foods. A sample of each class is given in Fig.3.

Fig.3. Sample images of wang database

Fig.4. Sample images of building database

The Zurich Buildings database for image based recognition is a benchmark created by Swiss federal institute of technology, Zurich [26]. It consists of 1005 training images and 115 query images of 201 buildings of 5 images each from different viewpoints and weather conditions. An example is given in Fig.4.

Fig.5. Sample images of flower database

The flowers dataset [27] consists of 17 species of flowers with 80 images each. There are species that have a very unique visual appearance, and very similar appearance. There are viewpoint, scale, and illumination variations also. An example is given in Fig.5. Simulations are carried out using these image databases.

4.2 IMAGE RECONSTRUCTION

The local codebook and locally global codebook are generated for the image database. The image is reconstructed from the local codebook. The locally global codebook is also used to reconstruct the same image. When the two images are compared, they are both similar to the original image. They retain the content of the original image and can be used to identify the image and hence used for image retrieval. A sample of the reconstructed image is compared with the original image in Fig.6 and Fig.7.

The encoding distortion of the image when it is encoded with the local codebook is compared with the locally global codebook. In Table.1, average distortion difference between the local codebook and locally global codebook with respect to the image block vectors is measured for the wang, building and flower databases for a codebook size of 128.

In previous work by Jeong [11], a training set of 16 images are taken from the wang database to create the global codebook. It was selected after visual comparison, to accommodate diverse colors. This same set is used in our experiments to construct a global codebook. It is compared with the locally global codebook. The locally global codebook produces lower distortion as shown in Table.1 when compared with the previous work. The codebook is automatically generated.

4.3 IMAGE RETRIEVAL

The locally global codebook is used to generate the histogram of the images from which the histogram intersection distance is calculated.

Indexing Algorithm

1. Let I = i_1, i_2, … i_n be the images in the database where n is the total number of images.
2. The SIFT keypoint descriptors are generated for the image using the image*j* surf program [26]. The keypoint descriptors are stored for each of the images. The keypoint descriptor is taken as a training vector for the image.
3. The LBG VQ algorithm is applied to T of each image i, to find the locally global codebook C_{Li} for each of the image in the database.
4. Using the locally global codebook, a Histogram of codevectors for an image i is calculated as $H_i = (h_{i1}, h_{i2}, \ldots, h_{i128})$ for the entire image database. It is a 128 dimensional vector $\{H_i: i = 1, 2, \ldots, 128\}$ where H_i is the number of source vectors that fall into the encoding region of a code vector.
5. The Euclidean distance [12] is calculated between two image histograms h and g as,
$$d^2(h,g) = \sum (h(x) - g(x))^2. \quad (1)$$
Similarity measure in the index is given as,
$$SHI_{hg} = \sqrt{\sum (h(x) - g(x))^2} \quad (2)$$
where, $h(x)$ is the histogram of one image and $g(x)$ is the histogram of the other image.
6. We find the histogram vector for all i images. The Euclidean distance between two image histograms H_i and H_j in the database is calculated for all the images, according to Eq.(2). It is stored in an image index model as shown in Fig.8.

Fig.6. Original and local codebook

Fig.7. Original and locally global codebook

Table.1. Average Distortion Difference between Local and Locally Global Codebook

Codebook Size	Local and locally global codebook			Local and global codebook (Jeong et al)
	Wang	Flower	Building	Wang
128	296.329	86.044	171.055	320.051

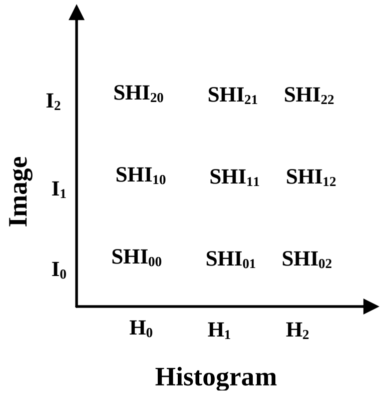

Fig.8. Image Index Model

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 = $a/(a+c)$

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 as in Fig.9 to Fig.11. The ranking of an image is based on the Euclidean distance of the histograms compared with the encoding distortion [1] value. The encoding distortion performs better as it is local comparison and the Histogram intersection is global comparison.

The mean average precision is compared with the results of Deselaers [10] as the same database has been used to compare various features including the SIFT LS and color histogram as shown in Fig.12. The SIFT Histogram Index has higher mean average precision than the various features calculated by Deselaers [10]. The feature vector is also reduced to a single measure value. This has significant reduction in the processing time of a query. It also is an image index structure.

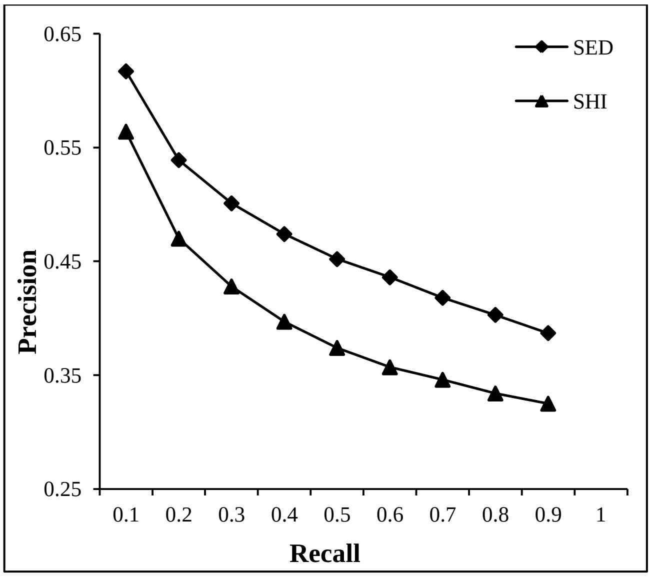

Fig.9. P R Graph for HI for wang database

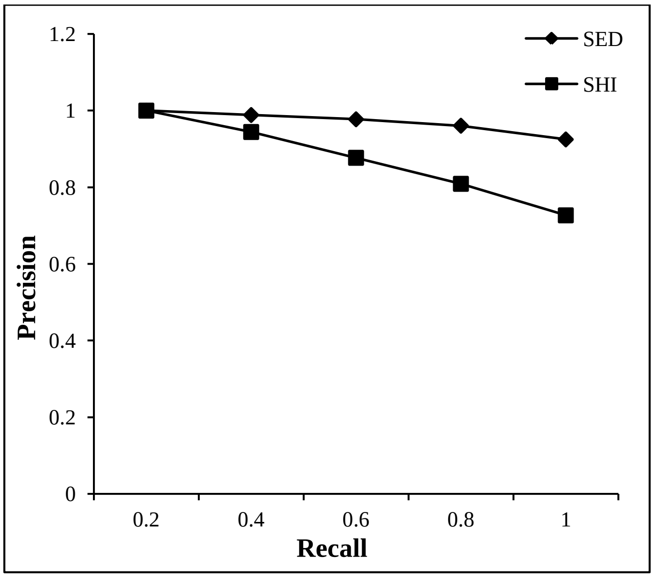

Fig.10. P R Graph for HI for building

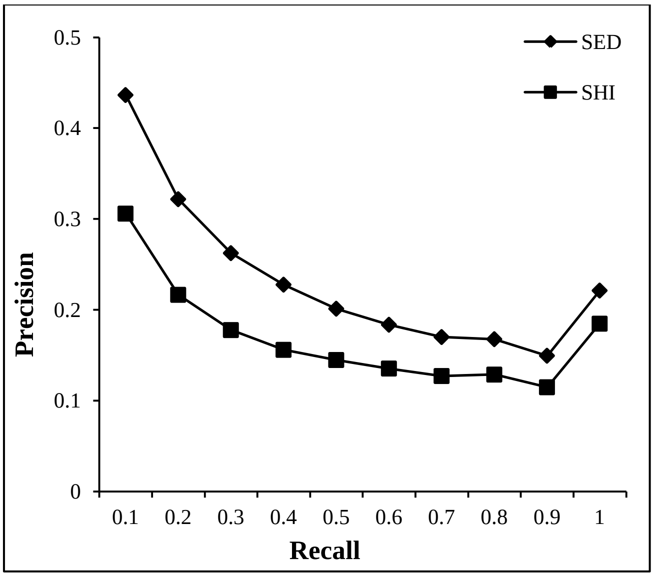

Fig.11. P R Graph for HI for flower

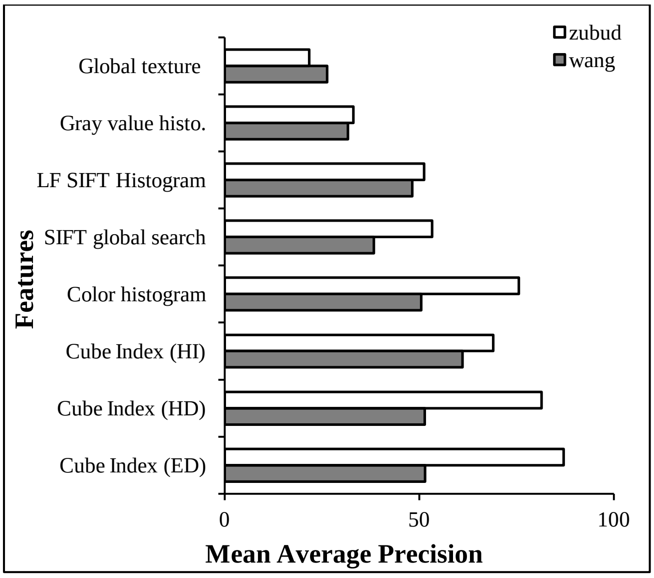

Fig.12. Mean Average Precision

4.4 IMAGE CLUSTERING

The encoding distortion d_{ED} is calculated using the formula,

$$d_{ED}(X,C) = \frac{1}{MK}\sum_{i=1}^{M}\min_{j}\left\|x_i - y_j\right\|^2. \qquad (3)$$

It is the average per component squared Euclidean distance between source vectors, $X = \{x\}_{i=1}^{M}$ and codevector $C = \{y\}_{i=1}^{N}$. Each image in the database is compared with the locally global codebook vectors and the encoding distortion value is calculated. The similar images have similar distortion value. A sample of the wang and building database is shown in Fig.13 and Fig.14. Their distortion values are displayed below the image.

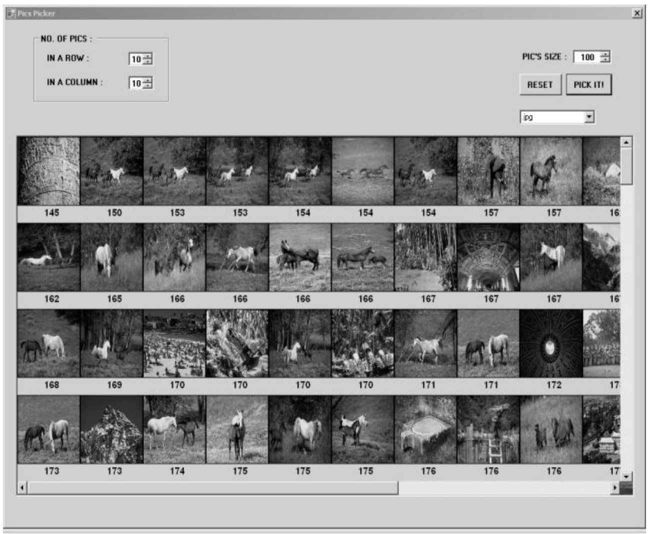

Fig.13. Image clustering for wang database

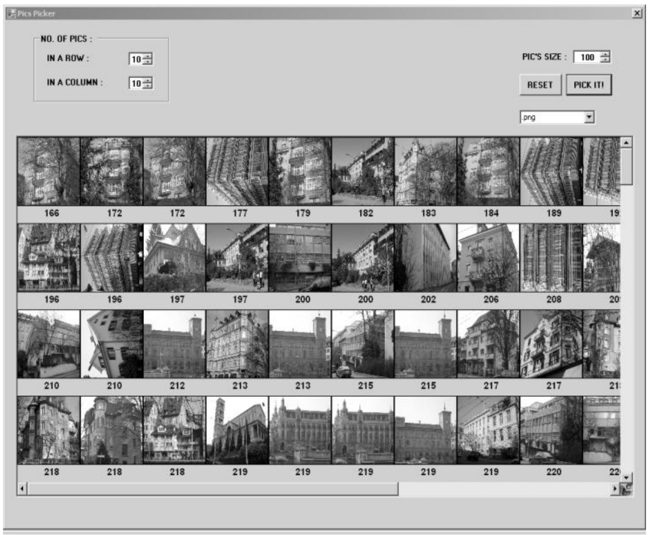

Fig.14. Image clustering for building

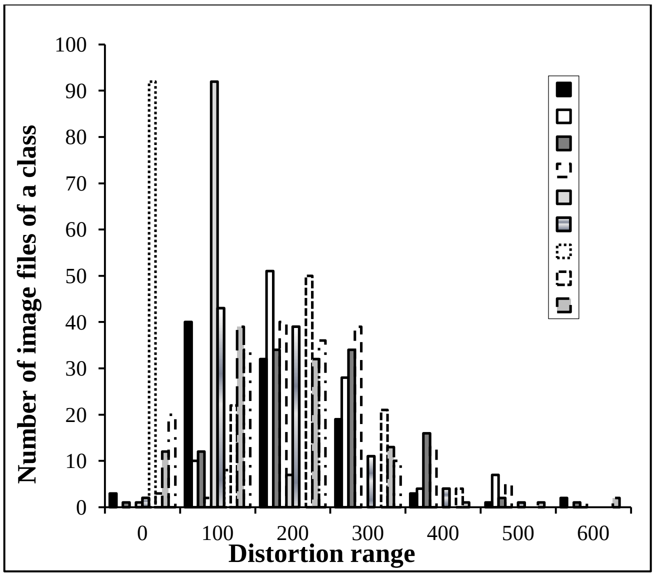

Fig.15. Clustering for wang database

Fig.15 gives the bar chart of the number of images that have a similar distortion value within a class. It uses a codebook size of 512 for the wang database. These figures show that the distortion can be used to cluster certain types of images together like horses and flowers in the wang database. These classes have high intra cluster similarity and hence similar distortion range. It will find application in video shot detection.

4.5 IMAGE MATCHING

Classification error rate, in retrieval is calculated. It is the first similar image checked to be relevant or not relevant. If relevant, then it is classified. Else, it is misclassified.

$$ER = \frac{1}{|Q|}\sum_{q \in Q}\left\{\begin{matrix} 0 & \text{if most similar image is relevant} \\ 1 & \text{if most similar image is irrelevant}\end{matrix}\right\} \quad (4)$$

The error rate for this retrieval is always zero as the distortion for the image for its codebook is always the lowest. Fig.16 and Fig.17 gives the sample images retrieved based on the HI measure from the index model. The first and the similar images are retrieved for some of the classes that have similar content. The image queried upon is given on the top of the picture retriever.

Fig.16. Image Matching in wang database

Fig.17. Image Matching in wang database

The best, average and worst case image retrieval example for the histogram intersection similarity measure in the image index model is shown in Fig.18, Fig.19 and Fig.20 for the Dinosaurs, horse and food classes respectively.

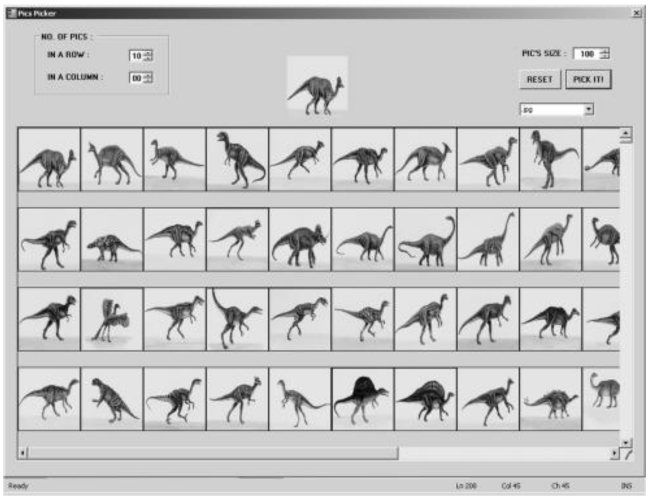

Fig.18. Retrieval Based on HI Index (Best Case)

Fig.19. Retrieval Based on HI Index (Average Case)

Fig.20. Retrieval Based on HI Index (Worst Case)

5. CONCLUSIONS AND FUTURE WORK

An incremental codebook generation process for VQ based large scale image retrieval has been proposed. The locally global codebook generated from the local codebook is found to represent the image with good image quality and acceptable encoding distortion value. This helps to represent an image database as a single codebook automatically. The incremental codebook generation process is used for large database by creating a hierarchy of classes of images. It helps with clustering and retrieval of images based on encoding distortion measure. The image index model with histogram measure can be used for image retrieval.

In future, the image index model has to be constructed for very large database. The applications of the global codebook have to be explored further.

REFERENCES

[1] Janet B., A.V. Reddy and S. Dominic, "An Image Index Model for Retrieval", *Proceedings of the International Conference on Recent Trends in Business Administration and Information Processing, Communications in Computer and Information Science*, Vol. 70, pp. 465–467, 2010.

[2] Abut H., "Vector Quantization", *IEEE press*, 1990.

[3] Gersho A. and R. M. Gray, "Vector Quantization and Signal Compression", Springer, 1991.

[4] Gray R. M.,"Vector quantization", *IEEE Transactions on Acoustics, speech and Signal Processing Magazine*, Vol. 1, pp. 4-29, 1984.

[5] Linde Y., Buzo A., and Gray R. M., "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, pp. 702-710, 1980.

[6] Lowe D.G., "Object Recognition from Local Scale-Invariant Features", *Proceedings of the Seventh International Conference on Computer Vision*, Vol. 1, pp. 1150-1157, 1999.

[7] Lowe D.G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[8] Anil K. Jain, Jung-Eun Lee, Rong Jin and Nicholas Gregg, "Content-Based Image Retrieval: An Application to Tattoo Images", *Proceedings of International Conference on Image Processing*, pp. 2745-2748, 2009.

[9] Ritendra D, Dhiraj Joshi, Jia Li and James Ze Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age", *ACM Computing Surveys*, Vol. 40, No. 2, 2008.

[10] Deselaers Thomas, Keysers Daniel and Ney Hermann, "Features for image retrieval: an experimental comparison", *Information Retrieval, Springer*, Vol. 11, pp. 77-107, 2008.

[11] Jeong S., C. S. Won, and R. M. Gray, "Histogram-Based Image Retrieval using Gauss Mixture Vector Quantization", *Computer Vision and Image Understanding: Special issue on Color for Image Indexing and Retrieval*, Vol. 94, No. 1-3, pp. 44-66, 2004.

[12] Jeong S. and. Gray R. M, "Minimum Distortion Color Image Retrieval Based on Lloyd-Clustered Gauss Mixtures", *Proceedings of the IEEE Transactions on Data Compression Conference*, pp. 279-288, 2005.

[13] Mirmehdi and Perissamy, "CBIR with perceptual region features", *In Proceedings of the 12th British Machine Vision Conference, BMVA Press*, pp. 511-520, 2001.

[14] Schaefer G, "Compressed Domain Image Retrieval by Comparing Vector Quantization Codebooks", In: *Proceedings of the SPIE Visual Communications and Image Processing*, Vol. 4671, pp. 959–966, 2002.

[15] Schaefer, G, Naumienko and W. Midstream, "Content Access by VQ Codebook Matching", *In: Imaging Science, Systems and Technology*, Vol. 1. pp. 170–176, 2003.

[16] Ryutarou Ohbuchi and Takahiko Furuya, "Accelerating Bag-of-Features SIFT Algorithm for 3D Model Retrieval", *Proceedings of the SAMT workshop on Semantic 3D Media*, 2008.

[17] Idris and S. Panchanathan, "Storage and Retrieval of Compressed Images", *IEEE Transactions on Consumer Electronics*, Vol. 43, No. 5, pp. 937-941, 1995.

[18] Idris and S. Panchanathan, "Image and Video Indexing Using Vector Quantization", *Machine Vision and Applications*, Vol. 10, No.2, pp. 43-50, 1997.

[19] Lu G. and S. Teng, "A Novel Image Retrieval Technique Based on Vector Quantization", *Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automation*, pp. 36-41, 1999.

[20] Janet B and Reddy A.V, "Cube Index for Unstructured Text Analysis and Mining", *Proceedings of International Conference on Computational Science*, pp. 397-402, 2011.

[21] Janet B and Reddy A.V, "Cube Index: A Text Index Model for Retrieval and Mining", *International Journal on Computer Applications*, Vol. 1, No. 9, pp. 88-92, 2010.

[22] Daptardar Ajay H and Storer James A, "Content Based Image Retrieval via Vector Quantization", *International Symposium on Visual Computing – Lecture Notes in Computer Science*, Vol. 3804, pp. 502-509, 2005.

[23] Janet B., A.V. Reddy, and S. Domnic, "Vector Quantization based Index Cube Model for Image Retrieval", *In Proceedings of the Pacific Rim Symposium on Image and Video Technology*, Vol. 12, pp. 27-32, 2010.

[24] Janet B., A.V. Reddy and S. Domnic, "Incremental Codebook Generation for Vector Quantization in Large Scale Content Based Image Retrieval", *Proceedings of IEEE International Conference on Center for Information System Research*, pp. 223-226, 2010.

[25] J. Z. Wang, J. Li and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 9, pp. 947-963, 2001.

[26] Shao H, Svoboda T, Tuytelaars T and Gool L V. HPAT, "Indexing for Fast Object/Scene Recognition Based on Local Appearance", *In: Conference on Image and Video Retrieval of Lecture Notes in Computer Science, Springer Verlag*, Vol. 2728, pp. 71–80, 2003.

[27] Nilsback M.E and Zisserman A., "A Visual Vocabulary for Flower Classification", *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 1447-1454, 2006.