International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 13, Issue 1 (January 2017), PP.47-55

Coding and Indexing Shape Feature using Golomb-Rice Coding for CBIR Applications

P. Sumathy[1]
[1]*Department of Computer Science, Bharathidasan University, Trichy, Tamilnadu, India.*

ABSTRACT:- The Content-Based Image Retrieval application uses large amount of low-level features of images for retrieving relevant images. Numerous low-level features are used for retrieval and among them the shape based feature is important and can be represented in the form of histogram. It is well known that the low-level feature requires more storage space and thus entire content has to be coded, indexed etc. to reduce the storage requirement, feature processing time and accessing time. In this work, indexing and encoding mechanism is proposed to effectively access the feature and represent the feature in lesser space. The bin values are used as reference for reducing the dimension of the histogram. The non-zero values are coded using Golomb-Rice coding and represented as compact histogram. The size of the compact histogram is variable in size and thus a similarity measure is proposed to handle the issue. Various well-known benchmark datasets are used for experiments to evaluate the performance. It is found that the truncated and encoded histogram performs well and achieves high precision of retrieval.

Keywords:- Encoding, Golomb-Rice code, Shape Histogram, Similarity measure, Image Retrieval.

I. INTRODUCTION

Recently, multimedia information retrieval is fetching attention from both academia and industry. For multimedia image retrieval applications, the low-level features are extracted for capturing the semantics. These extracted low-level features are higher in dimension and requires large storage space and the transmission and processing time is huge. It is understood that the feature has to be indexed, encoded and compressed to reduce disk space and to improve the processing speed for image retrieval. The feature can be reduced by using suitable coding scheme with less decoding time. Even if there is information loss during coding, the precision of retrieval should not be affected. During encoding, number of bits that are needed to be transferred and stored should be reduced considerably. Though the encoding/decoding overhead increases the communication latency and decreases the bandwidth, the encoding technique reduces the size of the low-level feature to a smaller size. Multimedia applications use adaptive predictors, entropy coders, de-correlating transforms with quantizes [1, 2] as coders and typical entropy coders used are Run-Length, Huffman, Golomb-Rice coders [3]. Due to huge variations in the statistics of blocks of integers to be encoded, designing the efficient entropy coders for such applications is found to be difficult. Among various entropy coders, Golomb-Rice (GR) coders [4] have been used widely in modern compression systems because of its flexibility in changing the encoding tables by modifying a single integer parameter [5, 15, 16, 17, 18]. The output code words are easily computed for the corresponding input symbols so that explicit tables are not required. This approach makes GR coders quite attractive and computations are much faster compared to memory access.

In this work, the size of the low-level feature vector is reduced by retaining only the necessary information for indexing purpose. The GR coding is used to encode the bin values adaptively for achieving lower average bit length. During retrieval, the similarity is calculated in encoded form and observed that precision of retrieval is encouraging. It is observed that performance of normal and encoded feature has achieved similar precision of retrieval. The quotient code of the encoded histogram alone is sufficient to retrieve relevant images. The rest of the paper is organized as follows. In Section II, we review the related work. Section III explains the proposed approach in detail. Section IV presents the experimental results and we conclude in the last section of the paper.

II. LITERATURE REVIEW

It is observed that most of the research work that involves encoding and compression techniques uses floating-point number that focuses on audio, image, scientific measurement and simulation domains. Encoding and compression technique have been proposed for the double-precision output of a numerical solver for ordinary differential equations [6]. Integer delta and extrapolation algorithms are used to encode and compress the data. However, this method is particularly beneficial for data, which gradually changes and may not be suitable for real-time applications. An efficient encoding and compression scheme for image data has been designed with an emphasis on 2D and 3D data [7]. The data is predicted using the Lorenzo predictor and encoded the residual with a range coder based on Schindler's quasi-static probability model [8]. A technique that combines differentiation and zero suppression is presented to encode and compress floating-point data arising from experiments conducted at the Laser Interferometer Gravitation Wave Observatory [9]. It is observed that the compression ratio of this approach is same as GZIP and is significantly faster. However, its success is realized only with the nature of LIGO data and thus value changes gradually. An algorithm has been proposed to encode the audio data [10]. It transforms the floating-point values into integers and generates an additional binary stream for the lossless reconstruction of the original floating-point values [19]. An extension to the JPEG2000 [12, 15] standard has been proposed that allow data to be encoded efficiently with bit-plane coding algorithms where the floating-point values are represented as big integers [11]. An arithmetic encoder is used for single-precision floating-point fields that represent residual vectors between the actual and the predicted vertex positions in triangular meshes [13]. In an extended precision algorithm, the Haar wavelet transform and Huffman coding is used to achieve lossless compression in 3D curvilinear grids [14]. This approach performs differential coding and clustering to generate separate data residuals for the mantissa and exponent. The Huffman coder and GZIP have been used to encode the mantissa and exponent residuals.

An approach has been proposed to encode and compress the histogram by run length coding of nonempty leaf nodes [20]. The statistical analysis inaccurately models the underlying random process generating the histogram. A feature descriptor codec is presented in [21], where each descriptor is transformed by a decorrelating transform, quantized and entropy-coded using Huffman coding or arithmetic coding. A lossy buffer compression/decompression technique has been used to find the error [23]. While the error grows relatively more, the method uses a lossless compression approach. However, this algorithm has operated only on Low Dynamic Range (LDR) data. A new color buffer compression algorithm is presented in [24] for coding floating-point buffers. It is considered that the approximate mode is error bounded and the amount of error introduced is accumulated and is controlled via few parameters. Hierarchical quad tree decomposition is performed and subsequently, hierarchical prediction as well as Golomb- Rice encoding are applied. However, this method is applicable for low-dynamic range color buffer. A method for encoding and compressing 16-bit floating-point color is presented and depth buffers in a unified manner, with several limitations [25]. This scheme does not allow negative values and assumes that the alpha channel is 1.0f. Compressing the alpha channel [26] has addressed complexity using two separate compressors for color data.

Based on the above discussion, it is observed that none of the above method is found to be suitable for CBIR applications and also the encoding approach is found to be more complex. As a result, we propose a simple indexing and adaptive coding scheme for encoding the bin values of low-level features. The encoding scheme has reduced the storage requirement space considerably. The shape histogram is truncated and retained only necessary feature values for indexing. The floating point bin values are converted to integer for GR coding. We have retrieved the images using only quotient part of GR encoding and found that the precision of retrieval is encouraging compared to the retrieval using original histogram. In addition, fast retrieval is achieved with lower bit length and low disk space

The proposed approach has three stages such as indexing, encoding and distance calculation. The schematic diagram of the proposed work is shown in Fig. 1. In the primary stage, an indexing scheme is proposed for reducing the dimension of the shape feature and has used indexing parameter. The indexed histograms are encoded using Golomb-Rice coding approach in the second stage. A new distance measure is proposed in the third stage, where the common bin indices between the query and database features are used to calculate the distance.

III. PROPOSED APPROACH

A. Indexing through Reducing Shape Histogram Dimension

In this paper, we have used shape features of objects in an image for capturing the fuzzy boundary information to estimate the degree of closeness of objects with well-known primitive shapes. The radius properties of the object are used for obtaining information about the shape irregularity. In this paper, *Mr* and *mR* are the maximum and minimum length that bisects the diagonal in each of the primitive shapes. For the sake of convenience and better understanding, in the rest of the paper, *Mr* and *mR* are considered as the maximum radius (*Mr*) and minimum radius (*mR*) of the object. Also, these values are second maximum (omitting the diagonal) and minimum distance between the centre to edge of the object respectively. The minimum and maximum radius is measured from the centre of the object. The centre of the object is the centroid and from which the distance to the boundary is measured with certain angular displacement. Among the measured sample, minimum and maximum value is considered as minimum and maximum radius. It is well known that while *Mr*=*mR*, the object is a perfect circle. However, for an object with irregular boundaries, the fuzzy membership function is defined in Table I for primitive shapes circle, ellipse, square, rectangle, rhombus, cone and cylinder.

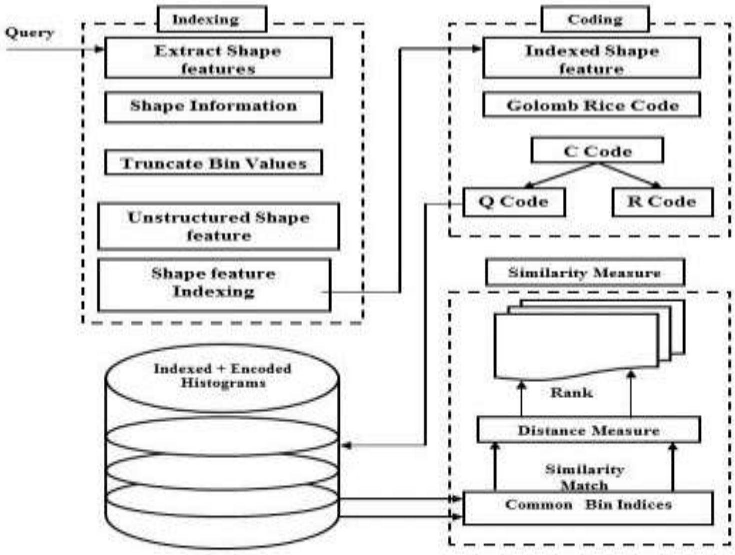


Fig.1. Schematic Diagram of the Proposed Work

Table I Fuzzy Membership Function for Primitive Shapes

Primitive Shapes	**Fuzzy Membership Function**
Circle	$\mu_C(x) = e^{\{-(x_c - 1)^2\}}$
Ellipse	$\mu_{EL}(x) = e^{\{-x_{el} - 1\}}$
Square	$\mu_{SQ}(x) = e^{\{-[x_{sq} - 1]^2\}}$
Rectangle	$\mu_{REC}(x) = e^{\{-(x_{rec} - 1)^2\}}$
Rhombus	$\mu_{RH}(x) = e^{\{-(x_{rh} - 1)^2\}}$
Cone	$\mu_{CN}(x) = e^{\{-(x_{cn} - 1)^2\}}$
Cylinder	$\mu_{CY}(x) = e^{\{-(x_{cy} - 1)^2\}}$

The extracted shape feature is represented as histogram and it is a normalized histogram in the form of single-precision floating-point format of the IEEE 754 standard as shown in Fig. 2.

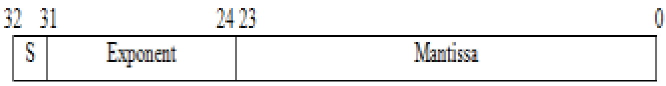


Fig. 2: 32-bit IEEE Floating-Point Representation

The trivial bins of the shape histogram are removed to reduce their dimension and the bins having four digits after the decimal points are also truncated. Apart from this, bins having 0 values and the bins having four 0's after the decimal point are removed. As a result, the shape histogram dimension is reduced to a great extent having only non-zero bins. As each shape histogram represents the shape information of the individual object, the presence of unequal non-zero bins results in the structural variation of the histogram as shown in Fig. 3.

1	2	3	4	5	6	....	61	62	63	64
√			√	√				√	√	
		√			√		√			√
√	√			√				√	√	
	√				√		√		√	√

Fig. 3: Schematic Structural variation of the Reduced Histogram

Since, the shape histogram bins are normalized, the floating point values are converted as integer without losing information. This is done since, GR coding uses integer values. Here, suitable multiplication factor (*Multi-Fact*) is found by counting number of zero's after the decimal point to convert floating point values to integer values. Here, we preferably selected Multi-Fact as 1000 since, more shape histograms bin values are having two 0's after the decimal point. Table.2 represents the Multi-Fact used in this approach and the entire bin values of the shape histograms are converted into integer values. Thus, the obtained integer histogram values can now use for indexing.

Table II Units for Magnetic Properties

No. of zeros after decimal point	**Multi-Fact**
0.0	100
0.00	1000
0.000	10000

The original shape histogram and reduced shape histogram representation is shown in the form of sparse matrix. Fig. 4(a) represents the original normalised shape histogram. Fig 4(b) represents the reduced shape histogram which gives the impact of data compressed and thereby encouraging increased computation speed. The Eq. (1) represents the indexing scheme (C_m) with 8 index levels (m) having different dimensions at each level.

$$C_m = [1 + (IF)*(m-1) \leq R_m \leq (IF)*m] \qquad (1)$$

In the above Eq, IF represents Index Factor=8, R_m is shape histogram dimension range and m is the index level [1-8]. A sample indexing structure with the shape histogram dimension range is depicted in Table. 2. In this work, the value of m and IF is fixed based on exhaustive retrieval experiments only. For instance, the shape histogram having 15 bins fit in index 2, since it falls in the dimension range of [9-16].

$$\begin{pmatrix} 0.5418 & 0.1043 & 0.2437 & 0.092 & 0.0043 & 0.0026 & 0.0011 & 0.0006 & 0.0003 \\ 0.0629 & 0.4843 & 0.2494 & 0.1101 & 0.0466 & 0.0214 & 0.0128 & 0.0076 & 0 \\ 0 & 0 & 0 & 0.0002 & 0.0008 & 0.6201 & 0.3403 & 0.0013 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0.0014 & 0.0714 & 0.6814 & 0.2222 \\ 0 & 0 & 0.0001 & 0.0004 & 0.0072 & 0.1134 & 0.6918 & 0.0925 & 0.0009 \end{pmatrix}$$

(a)

$$\begin{pmatrix} 541 & 104 & 243 & 92 & 4 & 2 & 1 & & \\ 62 & 484 & 249 & 110 & 46 & 21 & 12 & 7 & \\ & & & & & 620 & 340 & 1 & \\ & & & & & 1 & 71 & 681 & 222 \\ & & & & 7 & 113 & 691 & 92 & \end{pmatrix}$$

(b)

Fig 4: Sparse Matrix Representation (a) Original Shape Histogram (b) Reduced Shape Histogram

Table III Sample Indexing Structure

Index	Histogram Dimension	Range between	M-Parameter
1	8	[1-8]	108
2	16	[9-16]	82
3	24	[17-24]	74
4	32	[25-32]	63
5	40	[33-40]	44
6	48	[41-48]	32
7	56	[49-56]	25
8	64	[57-64]	12

The entire image database is grouped into several small blocks. For a given query image, the features are extracted, pre-processed, dimension is reduced and compared with the images in the matched block in lesser time.

B. Encoding the Indexed shape Histograms

Each block index has their header value in the form of M-parameter for which the encoding and decoding offset is calculated. The value of M for each block is calculated adaptively based on the indexed block size. The shape histogram blocks indexing works well even with uncontrolled data set of WWW. The indexing technique can be used to find out the ground truth of images easily for each block. Thus, considering the ground truth of each block index, the M-parameter is calculated using Eq. (2).

$$M = \sum_{i=1}^{n}\left[\left(\frac{h_i}{n}\right) * W\right] \qquad (2)$$

In above equation W is constant parameter h_i is fixed to 0.69, n is the total number of reduced shape histogram bins of respective block index, h_i is the reduced shape histogram bin value of respective block, M is the tunable parameter for quotient and remainder encoding. The shape histogram bin values are in the form of integer N and are encoded using tunable parameter M. The bin values are divided into quotient (Q) and remainder (R) part using Eq. (3) and Eq. (4).

$$Quotient\,(Q) = \mathrm{int}\,[N/M] \qquad (3)$$

$$\mathrm{Re}\,mainder\,(R) = [N \bmod M] \qquad (4)$$

While coding, if power of M is 2, then rice code is applied to the remainder and $\lceil \log_2(M) \rceil$ bits are needed for encoding. Alternatively, if M is not a power of 2, Eq. (5) is applied for encoding the remainder.

$$b = \lceil \log_2(M) \rceil \qquad (5)$$

Further, two important conditions are noted and they are:

1. If [(r<2)]^b - M, code r in plain binary using b-1 bits

2. If [(r>2)]^b - M, code the number [(r+2)]^b-M

in plain binary, using b bits. Based on the quotient and remainder values, code words are generated as per code format given in Eq.(6)

$$CodeFormat = <Quotient\,Code><\mathrm{Re}\,mainder\,Code> \qquad (6)$$

The sample encoded shape histogram is shown in Table. 3. The average bit length of the encoded shape histogram is calculated and it is found that the average bit length of dimension reduced shape histogram is always low.

The C Code, R Code and Q Code of sample block index is obtained and used for the retrieval. The retrieval performance is tested for all the code formats. It is noticed that the retrieval performance of Q Code C Code is similar. The performance of R Code is poor due to the fact of minimum information content. As a result, Q Code information alone is sufficient to achieve good result, which shown that Q Code information is equal to C Code information. This is represented in terms of entropy in Eq. (7)

$$E(C\ Code) = E(\langle Q\ Code\rangle\langle R\,Code\rangle)$$
$$= E(\langle Q\ Code\rangle\langle 0\rangle)$$
$$E(C\,Code) \cong E(\langle Q\,Code\rangle) \qquad (7)$$

since <R Code> is negligible in (7)

The entropy for C Code and Q Code for various images are calculated and it is noticed that the information content of C Code is slightly greater [w=0.1-0.4] compared to the Q Code and is represented in Eq. (7). This is due to the fact that C Code includes both Q Code and R Code. Thus, the difference between the entropy of C Code and Q Code is very low and Q Code alone is sufficient for representing the histogram, thereby reducing the bit length.

$$E(C\,Code) \cong w * E(Q\,Code) \qquad (8)$$

C. Common Bins Similarity Measure (CBSM)

The conventional distance measures face the curse of dimensionality issue due to nearest-neighbor search. Here, CBSM is proposed, which has good response time without affecting the precision. The entire database image is split various blocks and indexed as DB= (B_1, B_2,…, B_m). The block contains unstructured shape histograms with non-zero bins and represented as B_m = (h_1, h_2, h_3,…, h_k). The shape histogram bin is represented as $h_k(N_j)$, such that Nj represents the bin value for the bin index j where j=(1, 2, 3, ...). Given a query $h_q(N_i)$, the matching Block Index (B_1,B_2,…,B_m) is identified using its shape histogram dimension range (R_1,R_2,…,R_m). The CBSM computes the distance between query histogram (h_q) and histogram (h_k) in matched block database. Since, there exist a structural variation in the shape histograms; the similarity approach considers common bins of the shape histogram to compute the distance between them. The non-zero bin counts of query (*Count_q*) and block database histogram (*Count_k*) are considered. The common bins (Common (q)) between h_q and h_k are identified and degree of common bins *($Common_{(Degree)}$)* is computed using Eq. (9).

$$Common_{(\mathrm{Degree})} = \left(\frac{Common_{(q)}}{(Count\_q)}\right) \qquad (9)$$

The common bin values of both the shape histograms (h_q and h_k) are compared. As the bits are binary, XOR logic is used for comparison which gives the difference in the common bin values *($CB_{[diff]}$)* for complete histogram. This is represented in Eq. (10). Here, $N_i(h_q)$represents the value of i^{th} bin index of query histogram *(h_q)* and $N_j(h_k)$ is the value of j^{th} bin index of clustered database histogram *(h_k)*, which are common *(i==j)* where, *i=(1,2, 3,....,n)* and *j=(1,2,3,...,n)*.

$$CB_{(diff)} = CB_{(diff)} + \sum_{i=j=1}^{n}\left(N_i(h_q) - N_j(h_k)\right) \qquad (10)$$

In Table. 4, sample h_q and h_k is presented for better understanding of the working principle of CBSM. The non-zero bins of both the histograms are counted and analysed. It is observed that only five bins are common between them and considered for calculating the similarity. The Common $_{(Degree)}$ of query shape histogram is computed using Eq. (9). The encoded values *(Q Code)* of common bins of both histograms are compared and represented as *CB (diff)* using Eq. (10). Similarly, Common $_{(Degree)}$ and CB $_{[diff]}$ between query h_q and all the other histograms *(h_1, h_2, h_3,…,h_k)* in the Block B_m are calculated. The *$CB_{(diff)}$ (dist)* is calculated by counting number of 1’s in CB $_{(diff)}$ of the histograms as shown in Eq. (11).

$$CB_{(diff)}(dist) = CB_{(diff)}(Number of\ 1's) \qquad (11)$$

The *Common $_{(Degree)}$* and *$CB_{(diff)}$ (dist)* are normalized and their combined values give distance between hq and *(h_1, h_2, h_3,…, h_k)* of the Block B_m. This is represented in Eq. (12).

$$CBSM = \left(CB_{(diff)}(dist) + Common_{(Degree)}\right) \qquad (12)$$

Table IV Comparison of Encoded Values Between Query and Clustered Database Histogram

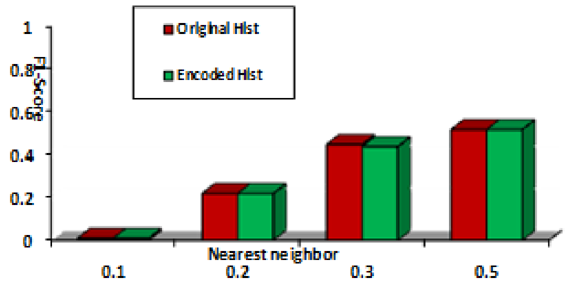
Common(q)	count_q	Common(Degree)	h_q <Q Code>	h_k <Q Code>	XOR	CB(diff)= XOR
3	23	5/(23)	1110	110	1000	
4			1110	111110	110000	11111100
5			11110	11111110	11100000	
54			111110	110	111000	
55			1110	10	1100	

IV. EXPERIMENTAL RESULT

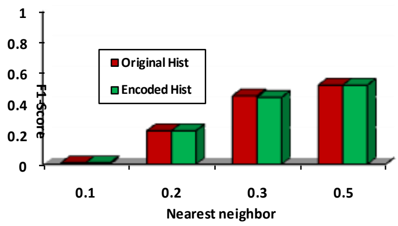
The effectiveness of the proposed encoded histogram is evaluated in an image retrieval system. For our experiment, we have used label me benchmark dataset (http://www.cucl.mit.edu/database.html) with 9356 images in which 9144 are object images categorized into 101 classes and 212 are texture images categorized into 19 classes. We have selected query images from various classes for computing recall and precision and the top 50 retrieval images are considered. The CBSM is used as the distance metric to measure the distance between query and database are images the Recall Vs. Precision is depicted in Fig.5 (a). It is observed that for lower values of recall, the precision is getting higher, which is reaching around 94%. The difference in precision for various recalls for both encoded and original shape histogram is almost zero. Further to consolidate the performance of the proposed approach, the F-measure is calculated using Eq. (13) which represents harmonic mean of recall and precision.

$$F = \left(\frac{2PR}{P+R}\right) \tag{13}$$

It can be noticed from the Fig. 5(b) that the difference in F-Measure between encoded and original histogram for various nearest neighbours is almost zero. This performance enhancement is due to the fact that the encoding procedure just assigns the code for minimizing the size of the histogram and the information is not lost. It is noticed from the result that the proposed approach performs well and retrieve the same result set as original histogram and also consume less space with good computation time.



(a)



(b)

Fig. 5. Performance between Original and Encoded Shape Histograms
(a) Precision Vs. Recall (b) F-Measure

To consolidate the performance of the proposed approach, we compared with other similar approaches such as RLE and arithmetic encoding. In RLE approach, a rate-efficient codec is designed for tree-based retrieval and run length encoding is used in [22]. But the approach had a limitation that runs length encoding is not able to work well at continuous-tone images. In arithmetic coding, the images are divided into histograms blocks. The maximum and minimum pixel value for the entire image is found, with range (R). Further, block size (R× R) is set using contrast of the image and arithmetic coding is performed. The performance measures for all the approaches are estimated in terms of Precision, Recall and F-measure and shown in Fig. 6 and it represents Precision vs. Recall for different query images. The ground truth and Precision vs. Recall is taken individually for various blocks and then average Precision vs. Recall is calculated. Proposed approach gives 70% and RLE approach gives 68% for recall at 0.1.

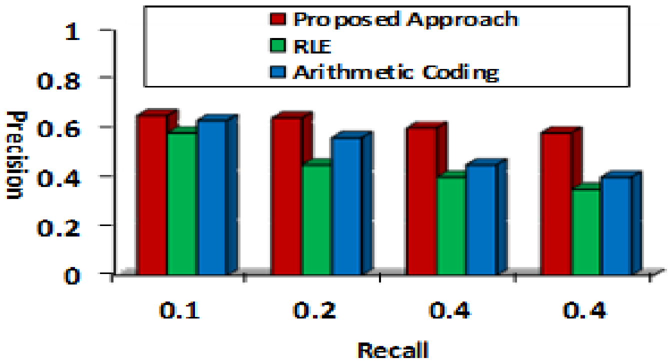


Fig. 6. Precision Vs. Recall of Proposed and Comparative Approach

Thus, the proposed approach encourages good precision of retrieval apart from reduced bit length, retrieval time and computation speed along with better search process.

V. CONCLUSION

The paper has considered the issues related to coding scheme for feature, storage and indexing for achieving higher retrieval time. An encoding scheme is proposed along with suitable distance measure to handle variance in structure of the histogram of query and database. The feature is coded using GR coding and the dimension of the histogram is used for indexing and clustering the features. The appropriate cluster is identified based on the dimension of the query histogram. Only quotient part of the histogram for image retrieval is used and retrieves the images without decoding. The performance of the proposed approach is evaluated using precision, recall and F-Measure and found that the difference in precision of retrieval between original and encoded histogram is almost negligible.

REFERENCES

[1]. K. Sayood, "Introduction to Data Compression", Morgan Kaufmann, 1996.

[2]. E. G. Richardson, "H.264 and MPEG-4 Video Compression, Video Coding for Next-Generation Multimedia" Wiley, 2003.

[3]. S. W. Golomb, "Run-length encoding", IEEE Transactions on Information Theory, Vol. IT-12, pp. 399-401, 1966.

[4]. H. S. Malvar, "Fast progressive wavelet coding", in Proceedings of the Data Compression Conference, Snowbird, Utah, March 29-31, pp. 336-343, 1999.

[5]. V. Engelson, D. Fritzson, and P. Fritzson, "Lossless Compression of High-Volume Numerical Data from Simulations", in Proceedings of Data Compression Conference, pp. 574-586, 2000.

[6]. P. Lindstrom, and M. Isenburg, "Fast and Efficient Compression of Floating-Point Data", IEEE Transaction on Visualization and Computer Graphics, Vol. 12, No. 5, 2006.

[7]. M. Schindler, "A Fast Renormalisation for Arithmetic Coding", in Proceedings of the Data Compression Conference, Viena, Austria, January, pp. 572, 1998.

[8]. S. Klimenko, B. Mour, Shawhan, and A. Sazonov, "Data Compression Study with the E2 Data", LIGO-T010033-00-E Technical Report, March, pp. 1-14, 2001.

[9]. F. Ghido, "An Efficient Algorithm for Lossless Compression of IEEE Float Audio", in Proceedings of the Data Compression Conference, IEEE Xplore, Romania, March 23-25, pp. 429-438, 2004.

[10]. B. E. Usevitch, "JPEG2000 Extensions for Bit Plane Coding of Floating Point Data", in Proceedings of the Data Compression Conference, IEEE Xplore, Texas, March 25-27, pp. 451-461, 2003.

[11]. M.N. Gamito, and M.S. Dias, "Lossless Coding of Floating Point Data with JPEG 2000", Applications of Digital Image Processing, Portugal, SPIE, Vol. 5558, pp. 276-287, 2004.

[12]. M. Isenburg, P. Lindstrom, and J. Snoeyink, "Lossless Compression of Floating-Point Geometry", in Proceedings of CAD Conference, Thailand, pp. 495-502, 2004.

[13]. D. Chen, J. Y. Chiang, and N. Memon, "Lossless compression of point-based 3D models", Pacific Graphics, October, pp. 124-126, 2005.

[14]. M. Burtscher, and P. Ratanaworabhan, "High Throughput Compression of Double-Precision Floating-Point Data", In Proceedings of the Data Compression Conference, IEEE Computer Society, 2007.

[15]. M. Burtscher, "VPC3: A Fast and Effective Trace-Compression Algorithm", In proceedings of the Joint International Conference on Measurement and Modelling of Computer Systems, New York, Vol. 32, No. 1, pp. 167-176. 2004.

[16]. M. Burtscher, and I. Ganusov, S. J. Jackson, P. Ratanaworabhan, and N. B. Sam, "The VPC Trace-Compression Algorithms", IEEE Transactions on Computers, Vol. 54, pp. 1329-1344, 2005.

[17]. M. Burtscher, and M. Jeeradit, "Compressing Extended Program Traces Using Value Predictors", In Proceedings of 12th International Conference on Parallel Architectures and Compilation Techniques, IEEE Xplore, New York, September 27 – October 1, pp. 159-169, 2003.

[18]. P. Ratanaworabhan, J. Ke, and M. Burtscher, "Fast lossless compression of scientific floating-point data", In Proceedings of the Data Compression Conference, IEEE Computer Society, Washington, pp. 133-142, 2006.

[19]. D. Chen, S. Tsai, V. Chandrasekhar, G. Takac, J. P. Singh, and B. Girod, "Tree histogram coding for mobile image matching", In Proceedings of Data Compression Conference, IEEE Computer Society, Washington, pp. 143-152, 2009.

[20]. V. Chandrasekhar, G. Takacs, D. Chen, S. S. Tsai, J. Singh, and B. Girod, "Transform coding of image feature descriptors", In Proceedings of the SPIE Visual Communications and Image Processing, San Jose CA, January 18, Vol. 7257, 2009.

[21]. C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient visual correspondences using random projections", In Proceedings of the 15th IEEE International Conference on Image Processing, San Diego, October 12-15, pp. 217-220, 2008.

[22]. J. Rasmusson, J. Hasselgren, T. Akenine, and Moller, "Exact and Error-bounded Approximate Colour Buffer Compression and Decompression", Graphics Hardware, Switzerland, pp. 41-48, 2007.

[23]. J. Rasmusson, J. Strom, T. Akenine, and Moller, "Error-bounded Lossy Compression of Floating-Point Colour Buffers using Quadtree Decomposition", The Visual Computer: International Journal of Computer Graphics, Springer-Verlag, Vol. 26, Issue 1, 2009.

[24]. J. Strom, P. Wennersten, J. Rasmusson, J. Hasselgren, J. Munkberg, P. Clarberg, Akenine, and Moller, "Floating-point buffer compression in a unified codec architecture", In Proceedings of the 23rd ACM SIGGRAPH/ EUROGRAPHICS symposium on Graphics hardware, pp. 75-84, 2008.

[25]. P. Wennersten, and J. Strom, "Table-based alpha compression", Computer Graphics Forum 28, Vol. 2, pp. 687-695, 2009.

[26]. A.Vadivel, S. Sural, and A. K. Majumdar, "Robust Histogram Generation from the HSV Space based on Visual Colour Perception", International Journal of Signal and Imaging Systems Engineering, InderScience, Vol. 1(3/4), pp. 245-254, 2008.