Single-step K-way Isoperimetric Bipartite Clustering of Images in Content-based Image Retrieval

[1]Temilola Adijat OKUNADE and [2]Olufade F. W. ONIFADE
1 Department of Computer Science, Lagos State Polytechnic, Lagos, Nigeria
2 Department of Computer Science, University of Ibadan, Ibadan, Nigeria
taokunade@yahoo.com

Abstract

Content-based image retrieval has been an active research area for quite some time. It evolved as a result of the disparity in actual image content and terms used to describe such images. While it is necessary to have the large image feature database in memory for efficient image retrieval, the huge memory requirement has continued to be a challenge. Attempts have been made to manage memory requirement by partitioning the large image feature database and loading each partition on separate distributed systems to speed up search. Although an appreciable reduction in time was achieved, the quality of search still needs to be improved. This research employed Single-step K-way Isoperimetric Bipartite Clustering (of images and terms in the surrounding text) to index the large image database. Clustering ensured that images that are semantically similar are together in a cluster. The new approach ensured that only the relevant cluster is searched at query response time thereby reducing the disk access overhead. Test images were retrieved at an average precision of 0.83 and average recall of 0.83. This is an indication of improvement in the quality of image retrieval.

Keywords: *Bipartite Clustering, Content-based Image Retrieval, Isoperimetric Bipartite Clustering,*

1. Introduction

The advancement in technology and the dynamic nature of the internet have made the volume of digital images available grow to the extent that efficient retrieval has become a challenge. The traditional approach to Image Retrieval involves annotating images with text and then applying Text Information Retrieval (TIR) methods to retrieve images whose annotations contain the keyword supplied by the user. The disadvantages of this include the high cost of manual annotation. This is due to the level of expertise required and the large volume of images. Inaccurate automated annotation is also a disadvantage because unlike humans that can easily differentiate between images, machine learning methods cannot. [29]

The development of image retrieval has been faced with some challenges. These include the semantic gap (the disparity between the contents of an image and its meaning as perceived by the user), feature ambiguity (caused by lack of global information from within a window or aperture) and the large volume of machine generated data (due to the alarming increase in the volume of digital data) [25].

Content-based Image Retrieval (CBIR) is the task of retrieving images based on their contents. It is an alternative to the traditional approach to image retrieval [20] Semantic information associated with images is ignored and image features are used for retrieval [24]. In CBIR, it is necessary to have the index structure to be searched in memory for fast access. This has huge memory requirements considering the volume of data that is to be searched. The efficiency of the index structure used determines (to a very great extent) the speed of response to user queries. However attempts to reduce memory requirement by partitioning the database has resulted in high disk access overhead and poor search quality [31].

The aim of the research is to develop a clustering model that reduces the disk access overhead and improve the quality of search through improved method of partitioning.

The remaining portion of this paper is structured as follows; section 2 contains a review of related literature; section 3 discusses the methodology while 4 presents results of experiment. Section 5 is a conclusion of the paper.

2. Literature Review

The major challenges facing the development of image retrieval include the semantic gap, feature ambiguity and large volume of machine generated data. Semantic gap refers to the disparity between the contents of a multimedia signal and its meaning as perceived by the user while feature ambiguity arises as a result of lack of global information from within a window or aperture. The large volume of machine generated data is due to the alarming increase in the volume of digital data that has become available. In CBIR a less than ideal indexing scheme often results in a search time that is not better than the linear search.

There are CBIR systems that make use of feedback from user on relevance of retrieved images to improve the quality of search. Research in this category includes the work of [29] in which by mining user navigation patterns, they were able to overcome the problem of redundant browsing and also exploration convergence. [17] were able to improve the quality of search by using semantics for the initial retrieval and through user feedback on relevance were able to retrieve more similar images using image features. [33] presented a review of relevance feedback techniques in image retrieval. Other efforts in relevance feedback include [9, 32].

There are also techniques that cluster the image feature database by keeping images with similar features in the same cluster and those with dissimilar features in different clusters. Images from a cluster that contains images similar to a query image are ranked based on a similarity measure and returned to the user. Several clustering methods have been implemented. [4] used unsupervised learning technique. [5] proposed KRA+ blocks. [12] employed co-clustering algorithm to achieve data co-reduction on both the data size and feature dimensionality. [28] applied hierarchical clustering to texture features extracted using Local Binary Patterns (LBP) in image retrieval. The research conducted a comparative evaluation of LBP to colour (RGB) and also different hierarchical clustering algorithms. Results show that colour is effective for images with huge textural differences although with the disadvantage of not being able to distinguish between different semantic elements that have the same colour. The method is more suited for images which may have major similarities with one another. Generally, results show high similarity of images returned to query image and an improvement of up to 9 times in query response time [28]. However, it is still necessary to develop an optimal CBIR system configuration that will provide an improvement in response time and relevance of returned images. [9] applied spectral co-clustering to image retrieval and also demonstrated that spectral clustering has computational advantage over k-means clustering. [22] applied bipartite clustering to images and image features in CBIR. [8] presented a research in which image features, images and terms in the surrounding text were represented as a tripartite graph. They applied spectral partitioning to cluster the graph. A similar research was carried out by [24]. They also represented image features, images and text as a tripartite graph but isoperimetric partitioning was applied. In

[23], the superiority of bipartite clustering over spectral clustering was established. Other clustering techniques studied include [35]. The research presented a clustering technique based on artificial neural networks. The research demonstrated the superiority of Growing Hierarchical Self-Organising Map (GHSOM) over Self-Organising Map (SOM) which requires knowledge of the data to define the network size. The algorithm was applied to document and word clustering. It was demonstrated that GHSOM has a shorter training time and also produces disjoint clusters of input data. [36] proposed hierarchical clustering based on k-means algorithm (HCKM). It consists of a first phase which clusters the dataset into a large number of non-empty sub-clusters using enhanced k-means algorithm. The algorithm is able to reduce the number of computations by ensuring that a point is assigned to a new cluster only if the distance to the old centre is greater than the distance to the new centre. This makes further computation to determine appropriate cluster unnecessary. The second phase employs the improved single-link algorithm in merging the sub-clusters. HCKM is able to overcome noise and outliers that were problems of single-link.

There are works that represent image feature database as a tree structure. The research of [30] led to the development of 'Non-Overlapping Hierarchical Index Structure (NOHIS-Tree)'. This research solved the problem of overlapping bounding forms when the index structure is data partitioning based. It was also able to solve the problem of sparsely populated or empty clusters being searched when the index structure is space partitioning based.

Instead of avoiding collision like the regular hashing, Locality Sensitive Hashing (LSH) works towards ensuring that similar images were hashed to the same bucket and dissimilar images are hashed to different buckets. Various forms of enhancements have been added to the original scheme. Research in this category includes MultiProbe LSH [18] in which a derived probing sequence was used to check multiple buckets that are likely to contain the nearest neighbours of a query object. This is on the assumption that if an object is close to a query but not hashed to the same bucket, it is likely to be in a bucket close by.

[31] proposed a multi-partition indexing approach to reduce the memory requirement. Attempts were made to improve the speed of similarity search. These include the research of [6, 26]. [2, 14] presented approaches to improve the hashing function while [11] considered distributed similarity search.

There are systems that are made up of a combination of any of the above techniques. The research of [3] is of interest in this regard. They presented LSH Forest in which each hash table was implemented as a tree structure. It was an attempt to improve the speed of searching each bucket. [13] combined more than one clustering technique (hierarchical and divide and conquer k-means) to improve performance of CBIR.

Most CBIR systems calculate similarities between a query image and images in the database instead of finding an exact match. The similarity measure is a measure of how close (or similar) the images are. The result of a search therefore, is not a single image but a list of images [16]. Each image returned is with a value showing how similar the image is to a query image. The images in the list are ranked by their levels of similarity to the query image. The distance measure is one minus the similarity measure. The suitability of the similarity measure used contributes to the performance of a CBIR system. Examples of similarity measures include Euclidean, Hamming, Minkowski-Form, Quadratic form, Mahalanobis, etc. Other measures include Jaccard, Cosine and Edit [15].

Image feature is either general or domain specific. Colour, texture and shape are examples of general features. [19] in his survey on feature extraction techniques identified properties of good features as Distinctiveness (or informativeness), Repeatability, Locality, Accuracy, Efficiency, Quantity and Repeatability. [21] described fusion approaches as combining image descriptors. They classified fusion approaches into early (when features are combined into a single representation before computation of similarity between images) and late (the combination of the outputs produced by different retrieval systems or the combination of similarity rankings). They were able to show that early fusion (colour, texture, shape, etc.) is able to handle the reduced inter-class variation experienced when only one feature is used. However, increase in computation cost and query response time for web applications is a concern. An application of early fusion approach was proposed and implemented by [1]. The researchers employed auto-correlogram, Gabor wavelet and wavelet transform to extract colour, texture and shape features respectively. Although the precision and recall for the system were high, it still necessary to assess the query response time of the system. In the research of [34], an image features fusion approach was also applied. They employed a linear combination of distances to calculate similarity of query image to the database images. The method combined Joint Composite Descriptor (JCD), Color Correlogram (CC) and Discrete Cosine Transform (DCF) as image features. This combined features approach showed a percentage improvement of up to 31.2% for an image category (Building) and an average improvement of 10.1%. Future study is targeted at automatic classification of images into categories. This is expected to make it possible to focus search on image collection rather than the entire database.

With relevance feedback techniques, after each round of user interaction the top results with respect to the query have to be recomputed using a modified similarity measure [7]. Another problem is the issue of user's patience in supporting multi-round feedbacks. An approach that has been used in reducing the problem is to incorporate logged feedback history into the current query.

Clustering has the major challenge that as new images are added to the database clusters will have to be rebuilt. The use of inappropriate keywords to describe images may also result in images not being in the right clusters. Relevance feedback techniques have been used to overcome this challenge [13].

For tree structures, multidimensional algorithms break down when the dimensionality of the search space is greater than a few dimensions [27]. This happens because nearly all the nodes in the dataset will be searched. Thus, the search degenerates to no better than exhaustive linear search.

Although LSH has been shown to produce good results [18, 31], it has the challenge of huge memory requirements when dealing with very large databases. This happens because the entire structure has to be memory resident. [31] made effort to overcome this challenge by partitioning the large database. In their first attempt, hash tables were constructed for each partition separately. LSH query was applied to each of the partitions one after the other. The results were collated and ranked to return the top k images. The disk access overhead observed was a challenge of this approach. In subsequent effort a parallel multi-partition system approach was used. The partitions were loaded on the distributed systems (slaves) and the LSH query was executed in parallel. The results were later collated on the master. This approach was able to remove the disk access overhead and speedup search due to the parallel execution of LSH query on the slaves.

In earlier research [31], there was no basis for the partitioning of the database. All partitions, including the ones that might not contain relevant images were searched. Some not so similar images were returned in the process. This reduced quality of search.

In this research, Isoperimetric bipartite clustering of the database ensured that only the relevant cluster is searched. This reduced the disk access overhead. Also, focusing on only the relevant cluster ensured that images that are not so similar to the query image did not get into the final set from which the returned images are selected, thereby improving the quality of search. The following section discusses the Single-step Isoperimetric Bipartite Clustering technique.

3. Methodology

Mainly CBIR systems consist of a pre-processing phase where features of images in the database are extracted and stored. Appropriate index structures are also constructed. There is also a querying phase where user queries are accepted, features are extracted and compared to those of the images in the database and similar images are retrieved for the user. Essentially, a CBIR system will consist of these two phases because features of the images in the database are to be extracted and indexed prior to query time.

This research employed a methodology that also consists of two phases – the pre-processing phase and the querying phase which handled the user query. The pre-processing phase also consists of two stages. The framework is shown in Figure 1. The Single-step K-way Isoperimetric Bipartite Clustering makes this approach different from earlier methods.

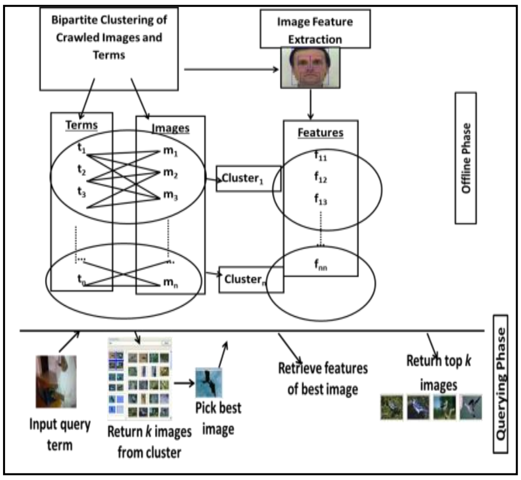

Figure 1. Single-step K-way Isoperimetric Bipartite Clustering in CBIR

3.1 The Pre-processing Phase

There are two stages in the pre-processing phase. The first stage – bipartite clustering of images and terms is geared towards minimizing the disk access overhead and speeding up CBIR. The bipartite clustering will ensure that images and terms that are similar belong to the same cluster. Therefore, only one of the clusters will be searched at query time. The second stage – image feature extraction is to ensure that a final selection of images from the identified cluster is done based on image features. This is to ensure that any clustering error that might have been introduced as a result of semantic gap is eliminated.

3.1.1 Single-step K-way Isoperimetric Bipartite Clustering of Crawled Images and Terms

Data clustering is the classification of data objects into different group (or clusters) in a way that data objects in one group are similar and dissimilar to data objects in other groups. The grouping is easily handled when the data objects are of the same data type. However, in real word problems, data objects are of two or more different types. Examples of such real world problems can be found in customer relationship management applications where customers and items purchased can be co-clustered in other to find out items of interest to a particular group. Another example is in biomedical applications where patient symptoms and diagnosis can be co-clustered to guide future diagnosis. Co-clustering has also been applied to documents and words to improve document retrieval. Other terms that have been used for data co-clustering include bi-clustering, bi-dimensional clustering and block clustering.

```
// Algorithm: Single-step k-way isoperimetric partitioning
// Input image-term matrix, number of clusters
input B
input no_of_clusters

construct A          // adjacency matric
construct D          // degree matrix
construct d          // degree vector
construct L          // Laplacian

// construct L* and d*
Find vertex with maximum degree
Remove row and column of maximum degree from L
// L*
Remove row with maximum degree from b
 // d*

Solve L* z*=d* using Least square method

Zmin = minimum(z)
Zmax = maxium(z)

Cluster_interval= (zmax – zmin) / no_of_clusters

Partition z into no_of_clusters using cluster_interval

Assign cluster_no to images and terms
```

Algorithm 1. Single-step k-way isoperimetric bipartite clustering

As a first stage in the pre-processing phase of this research, images and terms in the surrounding text of the images were clustered. The problem (Fig 1) was modelled as a weighted bipartite graph. The two sets of vertices were images and terms. An edge between image m and term t indicates that term t occurred in the surrounding text of image m. The frequency of occurrence of the term is the weight of the edge. The graph is represented as a matrix (B). An entry Bij refers to the frequency of occurrence of term j in the surrounding text of image i. The matrix was then partitioned into clusters. Rather than the recursive k-way partitioning applied in the original Isoperimetric Clustering Algorithm (ICA), this research applied Single-step K-way partitioning to improve the clustering speed. The algorithm for the procedure can be found in Algorithm 1.

3.1.2 Feature Extraction

In this section we describe the feature extraction methods applied to the images. For each of the clusters identified in 3.1.1, image features were extracted. Three features (colour, texture and shape) were considered.

Colour

Colour moments are measurements that can be used to differentiate between images based on their colours. When computed, these moments provide a measurement for colour similarity between images. The use of colour moments is based on the assumption that the distribution of colour in an image can be interpreted as a probability distribution. For the purpose of this research, grid moments calculated are mean, variance and skewness. Images were converted from RGB to HSV colour space and uniformly divided it into 3x3 blocks.

```
// Input array of images
input image_array
initialize image_features_array
for each image in image_array do
// for colour feature
 if image not HSV then
   convert image to HSV
 endif
  divide hsv_image into 3 x 3 blocks
  // for texture feature
 if image not gray scale then
   convert image to gray scale
  endif
  divide gray_scale image into 3 x 3 blocks
 initialize block_features
 for each block do    // compute colour features
   compute mean_colour
   compute variance_colour
   compute skewness_colour
   add mean_colour, variance_colour, skewness_colour to block_features
   // compute texture features
   compute mean_texture
   compute stdv_texture
   compute skewness_texture
   compute entropy
   compute correlation
   compute contrast
   compute homogeneity
   compute energy
   add mean_texture, stdv_texture, skewness_texture, entropy, correlation, contrast,
        homogeneity, energy to block_feature
 enddo
 add block_feature to image_feature_array
enddo
store image_feature_array
```

Algorithm 2: Feature Extraction

Texture

The texture of an image region is determined by the way the gray levels are distributed over the pixels in the region. For first order statistics, mean, variance and skewness were calculated. For second order statistics, information obtained from the co-occurrence matrix include Energy (or Angular Second Moment), Contrast, Correlation, Homogeneity and Entropy. In this research, the gray scale images were divided into 3 x 3 blocks and the statistics were computed for each of the blocks.

Shape

Images are segmented into regions to identify the shapes of objects contained in the images. Features of these identified regions are then extracted and stored for later comparison with those of the query image. Features that may be extracted from the region properties include area, centroid, perimeter, Euler number and diameter. For the purpose of this research, shape features extracted from the identified objects are area and perimeter

Algorithm for the feature extraction is shown in Algorithm 2.

3.2 Online Phase

The online phase is where the user searches for the required image. It consists of two stages. In the first stage, the user inputs a query term and the system identifies the cluster that the image belongs to. Images that correspond to the term in this cluster are retrieved. From this list, the user selects the image that best represents the required image. Features of the selected image are retrieved and another search is conducted to retrieve images that have similar features.

3.3 Test Data

In order to test the proposed scheme, images were crawled from Google Images. These images represent different categories. For example, mammal, reptile, amphibian, etc. Terms in the surrounding text of the images were also extracted. Words extracted were limited to nouns and pluralisation was removed. To prune the data, words with frequencies of occurrence less than 5 or greater than 10 and words less than 3 characters long were removed. A total of 11000 images and 1000 words were used for the research.

3.4 Performance Evaluation

The performance of the CBIR system was evaluated using precision and recall. Precision is the proportion of relevant images in the set of returned images. Recall is the ratio of the number of relevant images returned to the total number of relevant images in the database. The efficiency of the system was also evaluated by measuring the response time to a query. This was done in two stages. The first stage measures the time taken to retrieve sample images based on terms. Since the user's response in selecting an appropriate representation of his/her interest cannot be determined by the system, this phase is not measured. The second stage begins after the user has made a choice till the returned images are presented. The top 50 images were returned from each of the tested categories.

4. Results

Average precision and recall were calculated for all image categories. 50 images were returned for each query.

4.1 SKIBC and ES

Six (6) categories of images were used in testing the system. Each contains 50 images. When the term 'cup' is supplied by the user, a set of 50 images from the cluster containing the term 'cup' is returned (Figure 2). The user then selects the image that best represent his interest. For 'cup', 46 images were returned out of the 50 in the database (Figure 3). This is at a response time of 7.18s. A search for 'butterfly' returned 42 images out of the 50 in the database (Figure 4) at a response time of 6.20s. The average precision for SKIBC is 0.83. Recall is also 0.83. Average query response time is 6.60s. The performance of Single-step K-way Isoperimetric Bipartite Clustering (SKIBC) was compared to Exhaustive Search (ES). For SKIBC, the precision and recall of retrieval ranged from 0.74 to 0.92 with an average of 0.83. Query response time ranged from 6.20s and 7.18 with an average of 6.60s. For ES, precision and recall varied from 0.14 to 0.86 with an average of 0.37. The query response time ranged from 7.63s to 13.26s with an average of 9.88s. Clearly, SKIBC outperformed ES in terms of precision, recall (Table 1, Figure 4 and Figure 5) and query response time (Table 2, Figure 7)

Table 1 A Comparison of SKIBC and ES (Precision and Recall)

Images	NO. OF IMAGES IN THE DATABASE	NO. OF IMAGES RETURNED		PRECISION		RECALL	
		SKIBC	ES	SKIBC	ES	SKIBC	ES
Cup	50	46	7	0.92	0.14	0.92	0.14
Butterfly	50	42	9	0.84	0.18	0.84	0.18
Car	50	37	24	0.74	0.48	0.74	0.48
Zebra	50	39	15	0.78	0.30	0.78	0.30
Dish	50	46	43	0.92	0.86	0.92	0.86
Tortoise	50	39	13	0.78	0.26	0.78	0.26
Average				0.83	0.37	0.83	0.37

Figure 2. Sample images of cups in the database

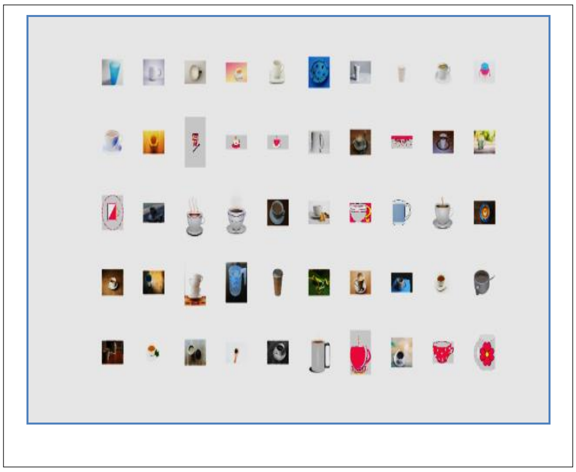

Figure 3. Images of cups returned

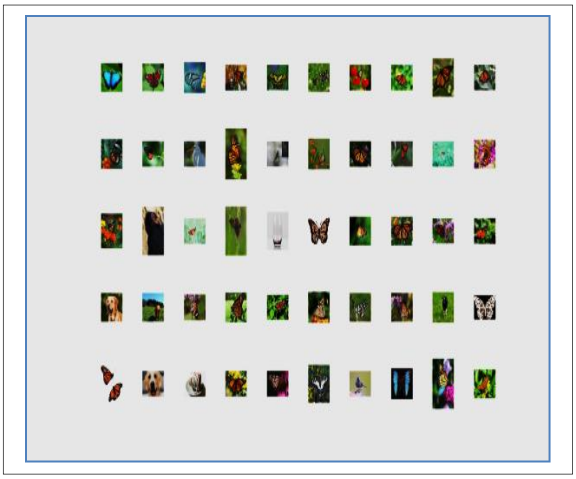

Figure 4. Images of butterfly

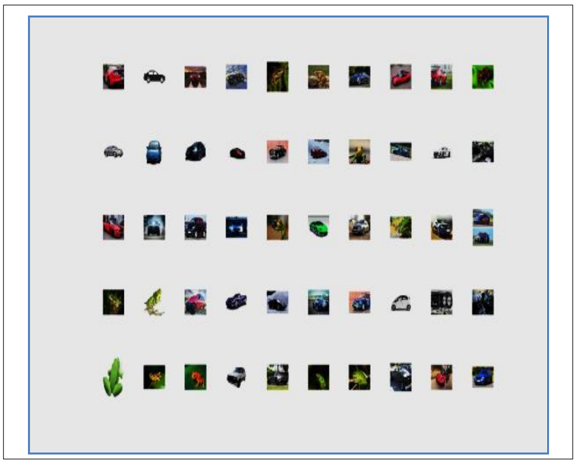

Figure 5. Images of car

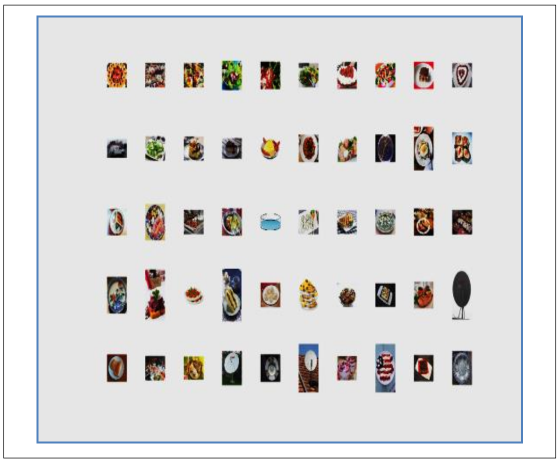

Figure 6. Images of dish

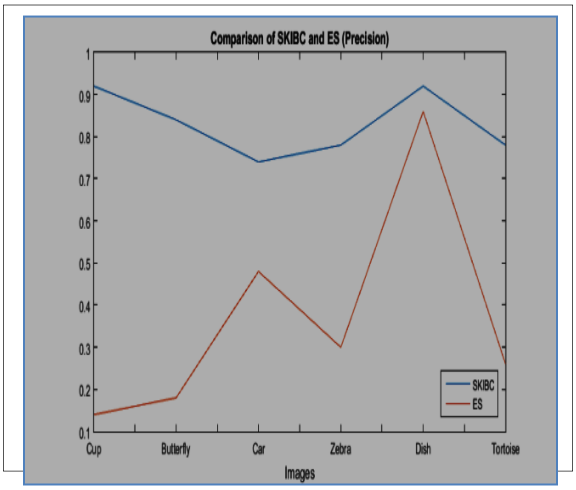

Figure 7. Comparion of SKIBC and ES (Precision)

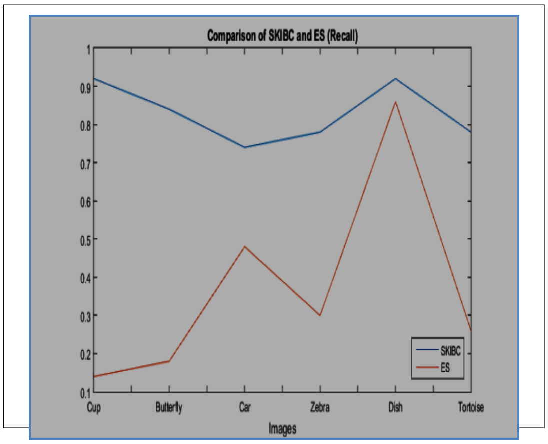

Figure 8. Comparison of SKIBC and ES (Recall)

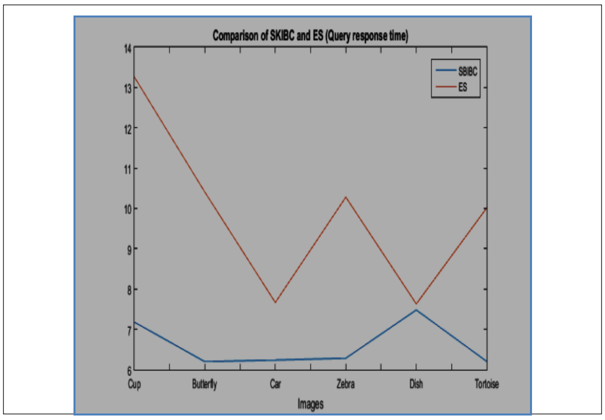

Figure 9. Comparison of SKIBC and ES (Query Response Time)

4.2 Comparison of SKIBC, with other methods

When compared with Disk-based Multi-partition Locality Sensitive Hashing (DLSH) and Parallel Multi-partition Locality Sensitive Hashing (PLSH), SKIBC performed best in terms of precision and recall. As compared to [1], the performance of SKIBC is slightly better in terms of precision while for recall, the performance is comparable.

Table 2: Comparison of SKIBC and ES (Query Response Time)

	TOTAL TIME (s)	
Images	**SKIBC**	**ES**
Cup	7.18	13.26
Butterfly	6.20	10.41
Car	6.24	7.67
Zebra	6.28	10.28
Dish	7.48	7.63
Tortoise	6.20	10.01
Average	**6.60**	**9.88**

Table 3: Comparison of SKIBC with DLSH and PLSH

	Precision	**Recall**	**Query Response Time (s)**
SKIBC	0.83	0.83	6.60
ES	0.37	0.37	9.88
DLSH	0.15	0.10	0.03
PLSH	0.13	0.08	0.02
Ananth *et. al.* 2016	0.80	0.83	-

However, the query processing time was not as good as the former techniques. Average precision for SKIBC, ES, DLSH, PLSH and [1] were 0.83, 0.37, 0.15, 0.13 and 0.80 while average recall values were 0.83, 0.37, 0.10, 0.08 and 0.83 respectively (Table 3 and Figure 10). Figures for query response times were 6.6s, 9.88s, 0.03s and 0.02s respectively (Figure 11). Average query response time for [1] was not available.

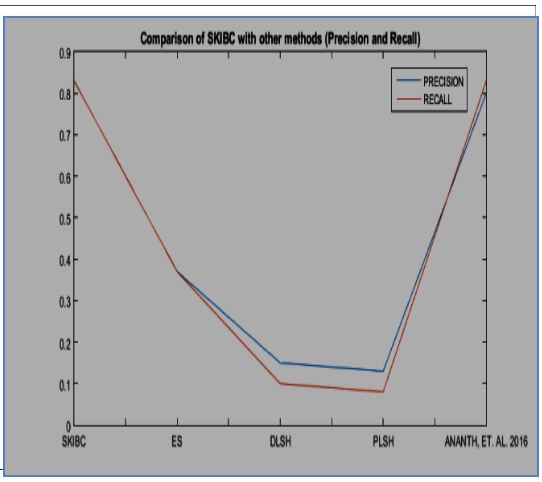

Figure 10. Comparison of SKIBC with other methods (Precision and Recall)

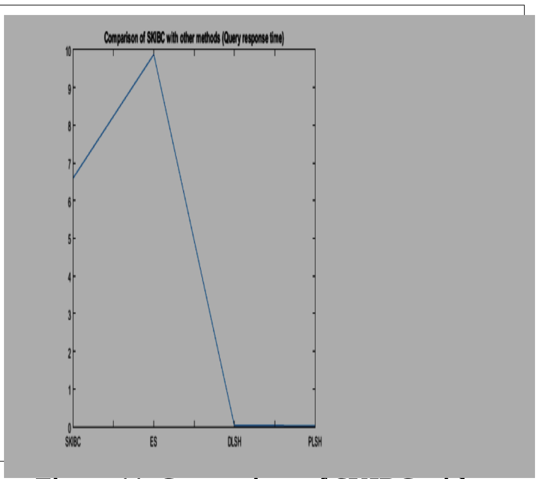

Figure 11. Comparison of SKIBC with other methods (Query Response Time)

This research presented Single-step K-way Isoperimetric Bipartite Clustering for content-based image retrieval. The presented system was implemented and tested using data crawled from Google images. SKIBC showed superior performance as compared to other methods in terms of precision and recall. The disk access overhead was also eliminated as it was no more necessary to load all the clusters in turn. However, the query response time needs to be improved upon. The elimination of recursion in the method of clustering the database is novel.

Despite the improvement in precision and recall of retrieval, avenues to improve the query response time will be explored in future.

References

[1]. A. Ananth, A., K. Mala and S. Sauganya, "Content-Based Image Retrieval System based on Semantic Information using Color, Texture and Shape Features." IEEE 978-1-4673-8437-7/16, 2016.

[2]. A. Andoni and P. Indyk, "Near optimal Hashing Algorithms for Approximate nearest Neighbour in High Dimension." Communications of the ACM. Vol. 51 No. 1, 2008, pp 117 - 122.

[3]. M. Bawa, T. Condie, and P. Ganesan, "LSH Forest: Self-Tuning Indexes for Similarity Search." WWW. Chiba, Japan: ACM, 2005.

[4]. Y. Chen, J. Z. Wang and R. Krovetz, "An Unsupervised Learning Approach to Content-Based Image Retrieval. IEEE Proceedings of the International Symposium on Signal Processing and its Applications." IEEE Proceedings of the International Symposium on Signal Processing and its Applications. IEEE, 2003, pp 197–200.

[5]. I. Daoudi, K. Idrissi, S. E. Quatik, A. Baskurt, and D. Aboutajdine, “An Efficient High-dimensional Indexing Method for Content-based Retrieval in Image Databases.” Signal Procesing: Image Communication Vol. 24, 2009, pp 775-790.

[6]. A. Dasgupta, R. Kumar and T. Sarlos, “Fast Locality sensitive Hashing.” Knowledge Discovery and Data Mining Conference, 2011.

[7]. R. Datta,, J. Li and J. Z. Wang' “Content-Based Image Retrieval – Approaches and Trends of the New Age.” MIR, 2005.

[8]. B. Gao, T. Liu, X. Qin, Q. Zheng, S. Cheng, and W. Ma, “Web Image Clustering by Consistent Utilization of Visual Features and Surrounding Texts.” Proc. of ACM Multimedia, 2005.

[9]. P. H. Gosselin and M. Cord, “Active Learning Methods for Interactive Image Retrieval.” IEEE Transactions on Image Processing. Vol. 17, 2008, pp 1200 – 1211.

[10]. J. Guan, G. Qiu and X. Y. Xue, “Spectral Image and Features Co-Clustering With Application to Content-based Image Retrieval.” Proceedings of the 17th International Conference on Pattern Recognition (Volume 4), 2004.

[11]. P. Haghani, S. Michel and K. Abarer, “Distributed Similarity Search in High Dimensions Using Locality Sensitive Hashing.” Extending Database Technology, 2009.

[12]. Z. Huang, H. T. Shen, I. Liu, and X. Zhou, “Effective Data Co-Reduction for Multimedia Similarity Search.” ACM SIGMOD, 2011.

[13]. M. Jain and S. K. Singh, “A Survey On: Content Based Image Retrieval Systems Using Clustering Techniques For Large Data sets.” International Journal of Managing Information Technology (IJMIT) Vol.3, No.4, 2011.

[14]. H. Jegou, L. Amsaleg, C. Schmid, and P. Gros, “Query Adaptive Locality sensitive Hashing.” IEEE International Conference on Acoustic, Speech and Signal Processing, 2008.

[15]. J. Leskovec, A. Rajaraman and J. D. Ullman, Mining of Massive Datasets, 2014.

[16]. F. Long, H Zhang and D. D. Feng, Multimedia Information Retrieval and Management, Berlin Heidelberg: Springer-Verlag, 2003.

[17]. Y. Lu, H. Zhang and L. and Hu, C. Wenyiu, “Joint Semantics and Feature Based Image Retrieval Using Relevance Feedback.” IEEE Transactions on Multimedia, Vol 5, No. 3, 2003, 339 -347.

[18]. Q. Lv, W. Josephson, Z. Wang, and M. A. Charikar, “Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search.” ACM VLDB, Vienna Austria, 2007.

[19]. G. Mandloi, “ A Survey on Feature Extraction Techniques for Color Images. .” International Journal for Computer Science and Information Technologies, Vol. 5 (3), 2014.

[20]. O. F. W. Onifade and O. Osagbemi, “User Interfaces: a Bridge for the Semantic Gap in Image Retrieval System.”, 2014.

[21]. L. Piras and G. Giorgio, "Information fusion in content based image retrieval: A comprehensive overview." Information Fusion, Volume 37.Issue C, 2017, pp 50-60.

[22]. G.Qiu, "Bipartite Graph Partitioning and Content-based Image Clustering." 1st European Conference on Visual Media Production (CVMP), 2004.

[23]. M. Rege, M. Dong and J. Hua, "Clustering Web Images with Multi-modal Features." Pproc. of ACM Multimedia, 2007.

[24]. M. Rege, M. Dong and F. Fotouhi, "Bipartite Isoperimetric Graph Partitioning for Data Co-Clustering." Data Mining and Knowledge Discovery Journal, vol. 16, No. 3, 2008.

[25]. B. Y. Ricardo and R. N. Berthier, Modern Information Retrieval: the concepts and technology behind search, Addision Wesley, 2011.

[26]. V. Satuluri. and S. Parthasarathy, "Bayesian Locality Sensitive Hashing for Fast Similarity Search." Proceedings of the VLDB Endowment, Vol X, No. Y, 2012.

[27]. M. Slaney and M. Casey, "Locality-Sensitive Hashing for Finding Nearest Neighbors." IEEE Signal Processing Magazine, 2008, pp 128 - 131.

[28]. R. A. Stefan, I. A, Szöke, and S. Holban, "Hierarchical clustering techniques and classification applied in Content Based Image Retrieval (CBIR)." 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics. 2015.

[29]. J. Su, W. Huang, P.S. Yu, and V. S. Tseng, "Efficient Relevance Feedback for Content-Based Image Retrieval by Mining User Navigation Patterns." IEEE Transactions on Knowledge and Data Engineering Vol. 23, No. 3, 2011, pp 360-372.

[30]. M. Taileb and S. Touati, "NOHIS-Tree: High-Dimensional Index Structure for Similarity Search." Proceeding of World Academy of Science, Engineering and Technology, 2011.

[31]. Y. M. Wong, S. C. H. Hoi and M. R. Lyu. "An Empirical Study on Large-Scale Content-Based Image Retrieval." International Conference on Multimedia and Expo, IEEE, 2007.

[32]. J. Yu,Y. Lu, Y. Xu, N. Sebe, and Q. Tian, "Integrating Relevance Feedback in Boosting for Content-Based Image Retrieval." International Conference on Acoustics, Speech and Signal Processing, IEEE, 2007.

[33]. X. S. Zhou and T. S. Huang. "Relevance Feedback in Image Retrieval: A Comprehensive Review." Multimedia Systems, 8, 2003, pp 536-544.

[34]. A. A. Popova and N. N. Neshov. "Image Similarity Search Approach Based on the Best Ranking". Egyptian Computer Science Journal, Vol. 37, No. 1, 2013.

[35]. A. B. M. Salem, M. M. Syiam, and A. F. Ayad. "Unsupervised artificial neural networks for clustering of document collections". Egyptian Computer Science Journal, Vol. 26, No. 1, 2004.

[36]. A. M. Fahim, A. B. M. Salem, F. A. Torkey and M. A. Ramadan. "Hierarchical Clustering Based on K-Means as Local Sample (HCKM)". Egyptian Computer Science Journal, Vol. 29, No. 1, 2007.