wjert, 2018, Vol. 4, Issue 4, 203-217. *Original Article* ISSN 2454-695X

World Journal of Engineering Research and Technology
WJERT
www.wjert.org SJIF Impact Factor: 5.218

GEO-CYCLICAL STRUCTURE-BASED HASHING

Abubakar Usman Othman*, Boukari Souley, Abdulsalam Ya'u Gital and Iliya Musa Adamu

Abubakar Tafawa Balewa University, Bauchi.

Article Received on 15/05/2018 Article Revised on 05/06/2018 Article Accepted on 26/06/2018

***Corresponding Author**
Abubakar Usman Othman
Abubakar Tafawa Balewa University, Bauchi.

ABSTRACT

in the past few years, approximate nearest neighbor (ANN) search has become the core in search field such as machine learning, pattern recognition, and data mining were the applications involves huge data bases. Hashing-based ANN have been known for their low memory requirements and computational cost. For this, many hashing-based approaches have been proposed for efficient similarity search from large-scale image collection for a given query. However, most of these methods generates their projections randomly which increases the cost of storage, they do not also consider the search time, and the data points are not spatially coherently mapped into compacts binary codes. To address these draw back, this paper proposes a scalable algorithm named Geo-cyclical structure-based hashing that can explore and preserve the underlying geometric structure information of the data. Extensive experiment would show that the new scheme will be more robust and more effective in terms of performance compared with state-of-the-art hashing approaches.

KEYWORDS: Hashing, indexing, image retrieval, similarity search, binary code embedding.

1. INTRODUCTION

The availability of internet such as with the use of the wide spread broadband Internet access,[2] coupled with these hand held devices (mobile devices), resulted to the easy collection of digital information in form of structured and unstructured[3] data, had contributed to the availability of large volumes of data known as big data. As unstructured data contributed to the availability of big data, they need to be structuralised for its effective understanding and processing through some optimised techniques used for extracting information. These information extracting techniques have been vastly used to extract meaningful information from raw or unstructured data.[4]

However, different indexing techniques were proposed by researchers with particular direction to big data in cloud computing. Approximate Nearest Neighbour (ANN) search has gained much attention recently for similarity searches and tree-based approaches.[5] R-tree-based indexing was proposed by[6] to allow multi-dimensional data to be indexed in the cloud. Distributed B-tree is an indexing method designed to perform consistent and concurrent updates and at the same time permits high concurrency reading operations.[1] In biometric system, databases are organised so that the search space for query image can be reduced to ensure higher throughput through an efficient indexing scheme. Nearest neighbours are classifiers mostly employed for image recognition and shape matching.[7] Nearest neighbours searches are classified as hash-based techniques,[8,9] and tree-based techniques.[10,11] For a reduced search time, a multi-dimensional indexing method, KD-tree[12] was proposed for finding best matches. Hash based indexing techniques are known for their effectiveness in application area like large-scale vision problems including image retrieval[13,14] image search,[15] object recognition,[16] local descriptor compressing,[17] fast multimedia search,[18] image matching,[19] and are efficient in search and similarity computation. Approximate similarity search,[20] while the c^2[22] is used for maintaining index items in d-dimensional data.

Less time and cost in indexing moving objects are needed to analyse big data with indexing techniques thus efficiently indexing of big data results to reduce time while still tolerating high cost when designing such indexing methods.[23] Effective methods for indexing, updating and querying this dataset were developed. Such effective methods are evident in the field of car tracking,[24] gaming engines[25] and tracking of mobile phones[26] and airplane surveillance.[27] Content based image indexing and retrieval, video indexing audio indexing aims at obtaining a structured indexing of the original video content and get familiar with its embedded semantics just as with human beings.[28] The drawback of these schemes is that the performance of the system degrade as the database increases which results to low speed, retrieval accuracy, and high search time.

To address the inefficiency of the above mentioned approaches, we propose a geometric cyclical-based hashing (GCBH) framework that optimise search accuracy and search time simultaneously. The GCBH also utilises the underlying structure information among the data for similarity preservation and efficient compact hash codes.

The rest of this paper is organised as follows: section two provides the review of related work concerning several indexing approaches. The proposed GCBH is presented in section three. Section four will provides the valuation parameters while the result of comparison will be provided in section five.

2. Related Work

In this section we review previous work related to approximate nearest neighbour search approaches.

2.1 Tree-Based Methods

B-tree indexing based techniques are related to an audit of multi-dimensional big data with regards to guiding rules and notable operators capable of mapping correlated data with search keys. They can handle data of different sizes in large volumes which is one of the features of big data,[29] proposes an efficient indexing scheme used to rank queries on temporal data. Top-k queries on temporal data takes near-linear time to answer any top-k (t) query with optimal I/O cost expected. The authors used the B-tree to design the SEB-Tree. They created a series of $l+1$ independent samples of S, where S is the set of segments. Although, the technique is efficient, it cannot handle online data stream that their behaviours are unknown. The authors in (Ling-Yin et al, 2013), proposed a novel key design based on $R^+ - tree$ for efficient retrieval of skewed spatial data. $R^+ - tree$ based technique is used to design the new indexing technique ($KR^+ - index$) that support efficient multi-attribute accesses for skewed data on cloud data management services (CDMS). $R^+ - tree$ is a balance search tree by dynamically splitting and merging nodes, and can restrict the number of elements in each node by controlling the M and m, where M and m are set parameters. Key names are designed for leaves of an $R^+ - tree$. The $R^+ - tree$ is used to divide the data, and the rectangle in the leave nodes of the tree index and are treated as grids.

The data used in the scheme are constructed by the $R^+ - tree$ with given M, m and the objects record for each rectangle say R1, R2 and R3 are maintained. R1, R2 and R3 are records for each rectangle representing restaurants.

To insert a new data point, the key of the point corresponding to the model to which the point belongs is first loop up and the data points are then inserted into the node. A check is performed on the current size of the node to ascertain if a split would be needed. When the number of points exceeded the required limit, the node is split to allow two new sub-nodes to be inserted and the old node will be deleted. The points in the old node will be allocated into one of the new sub-node. To efficiently retrieve skewed and spatial data, the techniques takes cloud data management into account.

The technique is efficient for data accessing and also provides support not only with range query but also with nearest-neighbour (NN) queries but it lacks a mechanism for fast responds to queries regardless of the query and data sizes. The method has the lower and upper bounds of rectangle (M, m), and the order o as parameters. The experimental result shows that the new indexing technique outperforms the state-of-the-art index method, MD-HBase, especially for skewed data. The system lacks confidentiality and it consumes large amount of memory space. There is high cost of computation because it takes long processing time.

2.2 Hashing Based Methods

In,[30] an indexing technique was propose for a biometric database consisting of variable number of features with high dimension. The scheme makes use of geometric properties of principal components of features, so that it can insert fewer features in the hash table after rotation of the first two highest principal components to the primary axis of the co-ordinate system. The triplet-based indexing technique which consist of two stages, is based on the triangle formed by the triplets of features. The features are the angles formed by the triangle. These features are extracted from models in the database through Speeded-Up Robust Features (SURF) for efficient computation to yield lower dimensional features. Salient points are first identified by using Hessian Matrix to detect key points and then take a rectangular window of these detected key point.

The stages of the triplet-based indexing technique are indexing and searching. In the indexing phase, principal component analysis transformation, triplet formation and hash table generation is done after the extraction of features by SURF. Principal component is used on each of the model in the database to make them invariant to translation, rotation and scaling. When the model images are translated, the extracted features are also translated from their original position. Mean centring is then used to translate each feature of all model images in the database to another image such that the mean of the translated image becomes zero.

Hashing with compact binary codes can achieve high performance for similarity search in huge database. A joint optimisation of search accuracy and search time algorithm was developed by[31] to generate compact hash codes for data of general formats with any similarity function. To preserve similarity, equation[35] is minimised. The authors analyse and model the search time which show that the search time can be minimised by balancing the hash bucket. Equation[36] is the maximum entropy criterion and a minimum mutual information criterion used in the balancing of the hashing buckets,[32] proposed a hashing technique that uses two hash codes of different length for stored images in the database and the queries. The compact hash code is used for the stored images in the database to reduce storage cost while the long hash code is used for the queries for searching accuracy. To retrieve images from the database, the Hamming distance of the long hash code is computed for the query and the cyclical concatenation of the compact hash code of the stored images for better precision-recall rate.

The propose method is compared with Iterative Quantisation (ITQ), Iterative Quantisation with Random Fourier Feature (ITQRFF), Orthogonal K-mans (OKM), Asymmetric Hashing (AH), LSH and Shift-invariant Kernel based Locality Sensitive Hashing (SKLSH). This method gives a better performance than ITQFF because the asymmetric hashing approach of the ACH provides better precise location data of the query. The experimental results show that ACH yields the best precision with the code length of 64-bit than the existing methods. The drawback of ACH is that the technique cannot distinguish favourably between images of birds and planes dew to similar feature values used. The scheme depends on the semantic similarity preservation of the features used and the weighted Hamming distance increase the computational complexity of the scheme,[33] proposed a method to preserve the underlying geometric information among data. The authors explore the sparse reconstructive relationship of data to learn compact hash codes. Usually, it gets over fitting in measuring the empirical accuracy on labelled data as such information provided by each bit is utilised to obtain desired properties of hash codes. The information theoretic constraint is incorporated into the relaxed empirical fitness as a regularising term to obtain the objective function,[34], proposed a novel hashing algorithm for effective high dimensional nearest neighbour search. DSH uses k-means to roughly partition the data set into k-groups. Then for each pair of adjacent group, DSH generate one projection vector which can well split the two corresponding group. From the generated projections, DSH select the final ones according to the maximum entropy principle in order to maximise the information provided by each bit. Given n_i data points $X = [x_i, \dots, x_n] \in R^{il*n}$, is to find L hash functions to map a data point x to a L-bits hash code. Table 2.3 provides a literature of existing techniques with their associated problems and solutions.

3. Geo-Cyclical Structure-Based hashing

In this section, we present a detailed Geo-CSBH Here we present our proposed system and its operational principle. The proposed system is composed by four components that performs a specific function to achieve the set objectives. Geo-CSBH can preserve the underlying discriminative geometric information among the data points. It explores the magnitude structure of geometric features of data. The image feature are indexed from the quantised hashing results. Figure 1 gives the conceptual framework of the proposed system. The working principle of the propose system is given in details with a detailed explanation of the responsibilities of each of the component that made up the model. This architecture incorporates the solutions to the identified problems in the various components that made up the proposed system.

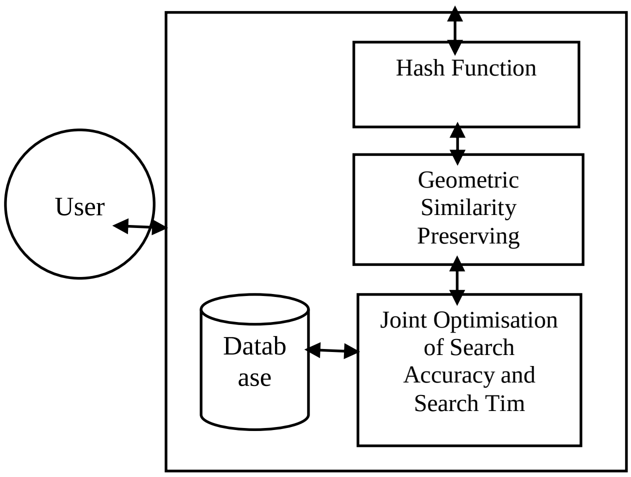

Figure 3.1: Conceptual framework of the proposed System.

3.1 Hash Function

This component of the propose system is responsible for the compression of the high-dimensional data into compact hash codes minimise storage cost. The hash function of the propose system compresses the original descriptor of the stored images in the database to a low-dimensional k-bit compact binary hash code with high compression ratio for small storage cost. The low-dimensional binary embedding can preserve the original geometric coordinate structure information of the data in the database. The samples in the original image database of images which correspond to the non-negative entries are used to approximate the given data vector. A geometrical hashing function that utilises the hypersphere-based hashing function is used to define a pivot in a D-dimensional vector space with a distance threshold. The key advantage of hypersphere-based function is that a higher number of region that are closed can be created using multiple hyperspheres, with distances between the points that are located in each of the region are bounded. To locate a nearest neighbours from a query point ANN search, closed region are formed with tight bounded distances. With this tighter regions, effective candidates for the nearest neighbours can be found within the range or region that is been indexed by the binary code of the query point.

2.3 Table of Literature

Table-1.

S/no	Author	Title of the work	Method used	Description of the method used	Problem associated with the method	Proposed solution
1	Mehrotra et al. (2010)	Robust iris indexing scheme using geometric hashing of SIFT key points.	Geometric Hashing (point based)	Local descriptors and relative spatial configuration are used for identity matching. SIFT is used to extract local features from noise independent annular iris image to detect key points. Geometry hashing is then applied to the detected key points for indexing in the database. In the retrieval phase, geometric hash location of query image is used to access the exact bin of the table and a vote is cast and images with certain number of votes are considered. Key point descriptors of possible candidates is matched with the query iris to get the potential match.	The method is redundant in that the features are mapped into the hash table multiple times. The feature points are not normalised. There is high memory consumption and computational cost.	Use SURF extraction technique to extract feature points from images. The feature points should be pre-processed and normalised. DOG should be used to detect interest points. Data number should be reduced in the hash bins to improve the performance of recognition. We employ a technique to evenly distribute features into hash table (DSH).
2	Li et al. (2010)	Top-k queries on temporal data	B –tree	B-tree is used as a building block to design a SEB-tree to support temporal ranking queries. SEB-tree answer a top-k query for any time instance *t* in the optimal number of I/Os in expectation. Piecewise linear functions are break into segments and uses the upper envelop U(S) which is made of the portion of the segment visible from $+\infty$ along y-axis. A series of $l+1$ independent samples are created and the query algorithm is designed.	The scheme is impractical when dealing with unknown behaviour of online data stream and it is also faced with high computational cost. It suffer from the curse of dimensionality and cannot deal with large-scale data base because of the memory constraint.	Graph partitioning and B+-tree (hybrid B-tree).

3	Vandana et al. (2013)	An efficient indexing scheme for face database using modified geometric hashing.	Modified Geometric Hashing	It uses the modified geometric hashing. SURF operators are used to extract control points from the face database. A pre-processing method mean centering, principal component, normalisation and rotation are used to make the control points invariant to translation, rotation and scaling.	Redundancy. The scheme does not support indexing of a database that is dynamic (increase and decrease in size) to enable modification. Variant are not uniformly distributed into the hash space. High memory cost.	Employ dynamic geometric hashing technique to support insertion, deletion of feature points, data points, data and updating the bin table. Use prime hash function to maximise the distance of keys with collision and also to ensure uniform distribution of variant into the hash space (bin).
4	Umarani et al. (2013)	Use of geometric features of principal components for indexing a biometric database.	Triplet-based hashing	Geometric features of principal components are used to insert fewer features into the hash table. SURF is used to extract features from the database by using Hessian matrix to detect key points. .	The scheme is not suitable for variety and does not support modification of the database. There is high search time and memory cost.	We use compact hash codes to reduce memory cost. Use dynamic geometric hashing technique to support modification of data. There should be minimal number of data points in the hash table to improve the speed of recognition. The number of feature points for each image in hash bins should be equal.
5	Ling-Yin et al. (2013)	Indexing spatial data in cloud data management	R+-tree	A key names for leaves on an R-tree are designed. R-tree is used to divide the data and the rectangle in the leave nodes are treated as dynamic grids. Data points are inserted into the node. Range query is used to get the geographic coordinates of the overlapped grids.	There is high consumption of space and high response time for large dataset. The search performance drops with data of high dimensionality.	There should be an efficient scheme suitable for velocity and volume
6	Wang et al. (2013)	High volumes of event stream indexing and	Composite tree (B-tree)	The solution is built based on the composite index data structure which shares a single list of event indices for	There is high consumption of computing resources	Hybrid indexing classifier that considers dynamic graph partitioning. Graph

		efficient multi-keyword searching for cloud monitoring.		all the leave nodes on the B-tree. Search index data structure is used to efficiently process timestamp-base queries.	because of many operation involved. The scheme suffers from the curse of data dimensionality.	partitioning and B+-tree (hybrid B-tree).
7	Meshram et al. (2013)	Different Indexing Techniques	Content-Based Indexing	Semantically meaningful movie events are extracted from movies. An online audio indexing system is used to create a searchable index speech content contained in digital audio files. Boundaries of the acoustically segment of data are searched and the data is then classified as speech, music or a mixture.	The focus was on online audio indexing system. The system is not suitable for veracity, variety and complexity.	Manifold learning
8	Liu et al (2012)	Supervise hashing with kernels	Hashing	Minimal amount of supervise information is used for high quality hashing. Supervised information are similar and dissimilar data pairs. The authors utilised the equivalence between optimising the code inner products and the hamming distances.	It cannot handle large spectrum of information such as duplicate document detection.	Design an algorithm that an minimise the minimum information criterion and the hamming distance.
9	Wang et al (2015)	Learning compact binary codes for hash-based fingerprint indexing	Hashing	The method on the existing Minutiae Cylinder Code (MCC). The characteristic of MCC are outlined and pointed out that it binary representation are bit correlated which makes it possible to represent minutiae feature in a more compact binary form. The theory of Markov Random Field (MRF) was used to model adjacent bit correlations in the MCC binary representation which makes it easier in learning the hash bit from a generalised linear model (GLM)	The orthogonal coordinate system used in this scheme increases the computational complexity and there is a very high memory consumption. Also, the sample size used for the training is small. An increase in the size of database results to an increase in search time.	The hash table should contain fewer entries equally distributed to reduce memory requirement and improve the search accuracy.

10	Liu et al. (2014)	Cross-indexing of binary SIFT codes for large-scale image search	Scale invariant feature transformation (SIFT) binarisation	PCA is perform to the SIFT descriptors with much focus on larger variant in dimension. The SIFT descriptor is then transformed into binary codes with the targeted length. Visual information is represented in compact binary code storage cost reduction and computation.	There is high retrieval time cost due to multiple cells checks besides the query feature cell. There is high memory cost due to indexing of many features in the hash table.	Optimisation of the hamming distance of the binary codes between features. To reduce memory cost, fewer features needs to be indexed.
11	Lv et al (2015)	Asymmetric cyclical hashing for large scale image retrieval.	Asymmetric cyclical hashing.	Two hash codes of different length are used for stored images in the database and the queries. The compact hash code is used for the stored images in the database to reduce storage cost while the long hash code is used for the queries for searching accuracy. To retrieve images from the database, the Hamming distance of the long hash code is computed for the query and the cyclical concatenation of the compact hash code of the stored images for better precision-recall rate.	There is long response time and additional computational cost for calculating the Hamming distance of the compact hash code of the stored image and long hash code of the query. There is large storage cost due to the use of long codes.	The hash function designed should be based on the distribution of data for effective short compact hash codes.

3.2.2 Geometric Similarity Preserving

This component of the propose system is responsible for preserving the similarities of two sample data points in the training data set. Given a database X, two data samples X_i and X_j contained in the training set of data. We then have similarity between the two data samples as Wij from the similar geometric feature points of image data. Hashing methods require geometric coordinate properties for similarity preserving. The data points that are similar usually have similar hash codes with small hamming distance.

3.2.3 Joint Optimisation of search accuracy and search time

In this section, we integrate the similarity preserving term $D(Y)$ for search accuracy and the minimum information criterion for the search time to form a single entity. To enable a high search accuracy with fast search time, the joint optimisation component of the propose system formulated and is responsible for the simultaneous optimisation of the search accuracy and search time. A parametrisation of a linear function is perform for easy optimisation, and a relaxation is perform. To relax the similarity preserving term and the minimum information criterion, the binary constraint is being removed from the similarity preserving term and the minimisation of the minimum information criterion.

The joint optimisation is responsible for the computing the hash bit that will be used for query and the identification of the bucket with the same hash bits with the query, and to also oversee the loading of data samples from the selected buckets into the memory.

Table 3.1: Parameters used.

S/no	Symbol	Uses
1	$D(Y)$	A similarity preserving term
2	r	A representation for r –adjacent groups
3	α	A parameter that controls the numbers of groups
4	$h(x)$	A hash function
5	c_i	The centre of hypersphere
6	w_i	The radius of the hypersphere
7	X_i	represents data sample
8	X_j	represents data sample

4.0 Performance parameters (Evaluation).

This section provides and avenue for evaluating the proposed technique and make a comparison to the state-of-the-art-techniques.

4.1 Performance metrics

The geometric cyclical hashing will be compared with state-of-the-art-techniques to obtain the mean average precision based on parameter analysis, the precision-recall rate, and the search accuracy and search time trade-off.

4.2 Evaluation Techniques.

The algorithms used in the evaluation of the proposed system are the DSH, SH, KLSH, LSH

5.0 EXPECTED RESULTS

To generates discriminative binary hash codes that yield high search accuracy and an improved search time with less memory consumption. Develop a framework for a geometric structure-based hashing for large scale image retrieval.

REFERENCES

1. Aguilera M, K., Golab, W., & Shah, M. A. A practical scalable distributed b-tree. *Proceedings of the VLDB Endowment*, 2008; 1(1): 598–609.
2. Huang, Z., Heng, T. S., & Shao, J. Bounded Coordinate System Indexing for Real-time. *ACM Transactions on Information Systems*, 2010; 10(10): 1-32.
3. Gartner M, Rauber, A., & Berger, H. Briging structured and unstructured data via hybrid semantic search and interactive ontology-enhanced query formulation. *Knowledge information system*, 2013; 1-32.
4. Doan, A, N. J., Baid, A., Chai, X., Chen, F., Chen, T., . . .Vuong, B. Q. The case for a structured approach to managing unstructured data. *In Proceeding of the 4th Biennial Conference on Innovative Data Systems Rsearch*, 2009.
5. Muja, M., & Lowe, D. G. Fast Approximate Nearest Neighbours with automatic algorithm configuration. *VISAPP*, 2009; 331-340.
6. Wang, J., Wu, S., Gao, H., Li, J., & Ooi, B. C. Indexing Multidimensional Data in a Cloud System. *ACM SIGMOD International conference on management of data*, 2010; 591-602.
7. Frome, A., Singer, Y., Sha, F., & Malik, J. Learning globally-consistent local. *ICCV*, 2008, June.
8. Wu, C., Zhu, J., Cai, D., Chen, C., & Bu, J. Semi-supervised nonlinear hashing using bootstrap sequential projection learning. *IEEE transaction on Knowledge data Engineering*, 2013, 06; *25*(6): 1380-1393.

9. Raginsky, M., & Lazebnik, S. Locality-sensitive Binary Codes from Shift Invariant Kernels. *Proceedings of Advance Neural Information Processing System*, 2009; 1509-1517.
10. Silpa-Anan, C., & Hartley, R. Optimised KD-trees for fast image descriptor matching. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2008; 1-8.
11. Beygelzimer, A., Kakade, S., & Langford, J. Cover Trees for Nearest Neighbour. *In Proceedings of 23rd International Conference on Machine Learning*, 2006; 97-104.
12. Friedman, J., Bentley, J. L., & Finkel, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software,* 1977; *3*(3): 209-226.
13. Prateek Jain, B. K. Fast image search for learned metrics. *In proceeding of the IEEE conference on computer vision and pattern recognition*, 2008; 6.
14. Xu, H., Wang, J., Li, Z., Zeng, G., Li, S., & Yu, N. Complementary hashing for approximate nearest neighbor search. *In Proc. ICCV,* 2011.
15. Kulis, B., Jain, P., & Grauman, K. Fast similarity search for learned metrics. *TPAMI,* 2009; *31*(12): 2143–2157.
16. Torralba, A., Fergus, R., & Freeman, W. T. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *TPAMI,* 2008; *30*(11): 1958–1970.
17. Strecha, c, A. M., M, M. B., & P, F. Ldahash: Improved matching with smaller descriptors. *TPAMI*, 2012; *34*(1): 66-76.
18. Zhu, X., Huang, Z., Cheng, H., Cui, J., & Shen, H. T. Sparse hashing for fast multimedia search. *ACM Transaction on information system*, 2013; *3*(2): 1-24.
19. Avidan, S., & Korman, S. Coherency sensitive hashing. *In Proceedings of ICCV*, 2011.
20. Datar, M., Immorlica, N., Indyk, P., & Mirrokni, P. Locality sensitive hashing scheme based on p-stable distributions. *In Proceedings of the Symposium on Computational Geometry*, 2004; 253–262.
21. Zhou, A. c^2: a new overlay network based on can and chord. *international journal of high performance computing network*, 2005; *3*(4): 248-261.
22. Kadiyala S, S. N. A compact multi-resolution indedx for variable length queries in time series database. *Knowledge information system*, 2008; *15*(2): 131-147.
23. Jensen, C., & Pakalnis, S. TRAX—real-world tracking of moving objects. *n:VLDB*, 2007.
24. White, M., Demers, W, K., A, J. G., C, R., & G, R. Scaling games to epic proportion. *In SIGMOD*, 2007.

25. Anderson, I., Julie, M.,, Scott, S., Louise, B., Matthew, C., Malcolm, H., Barry, B., & Henk, M. Shakra: tracking and sharing daily activity levels with unaugmented mobile phones. *mobile network application,* 2007; *12*(2-3): 185-199.
26. Thirde D, Mark, B., Aguilera, Ferryman, J., Keith, B., Kampel, M., Evaluation of object tracking for aircraft activity surveillance. *In: 2nd joint. IEEE Transactrion,* 2005.
27. Meshram, B. B., & Gaikwad, G. P. Different indexing techniques. *International Journal of Engineering Research and Application,* 2013, 4; *3*(2): 1230-1235.
28. Li, F., Yi, K., & Le, W. Top-k queries on temporal data. *VLDB,* 2010; *19*(5): 715-733.
29. Umarani Jayaraman, S. P. Use of geometric features of principal components for indexing a biometric database. *Mathematical and computer modelling,* 2013; *58:* 147-164.
30. Junfeng, H., Regunathan, R., Shih-Fu, h., & Claus, B. Compact Hashing with Joint Optimisation of Search Accuracy an Time. *CVPR,* 2011.
31. Yueming, L., Wing, W. Y., Ziqian, Z., Daneil, S.
32. Ye, R., & Xuelong, L. Compact Structure Hashing Vis Sparse and Similarity Embedding. *IEEE Transactions on Cybernetics,* 2016, March; *46*(3): 718-728.
33. Jin Z, L. C., Lin, Y., & Cai, D. Density Sensitive Hashing. *IEEE transactions on Cybernetics,* 2014, august; *44*(8): 1362-1371.
34. Mehrotra, H., Majhi, B., & Gupta, P. Robust iris indexing scheme using geometric hashing of SIFT keypoints. *Journal of Network and Computer Applications*, 2010; *33:* 300–313.
35. Vandana, D. K., Jayaraman, U., Amit, K., Aman, K. G., & Gupta, P. An efficient indexing scheme for face database using modified geometric hashing. *Neurocomputing,* 2013; *116:* 208-221.
36. Ling-Yin, Y.-T. H.-C.-C. Indexing spatial data in cloud data management. *Pervasive mobile computing*, 2013; 1-14.
37. Wang, M., Holuba, V., Murphya, J., & Sulliva, P. O. High volumes of event stream indexing and efficient multi-keyword. *Future Generation Computer System*, 2013; *29:* 1943-1962.
38. Wang, Y., Lipeng, W., Yiu-Ming, C., & pong, C. Y. Learning Compact Binary Codes for Hash-Based Fingrprint Inexing. *IEEE Transation on Information Forensic and security,* 2015, August; 10(8): 1603-1614.
39. Lv, Y., Wing, W. Y., Ziqing, Z., & Daniel, S. Y. Asymmetric cyclical hashing for large scale image retieval. *IEEE transaction on multimedia,* 2015, August; *17*(8): 1225-1235.