

An Efficient Indexing Technique for 2D Face Data

Ankesh Gupta

P.G. Student, Department of Computer Science & Engineering, Baba Saheb Bhimrao Ambedkar University
(A Central University), Vidya Vihar, Lucknow, India

ABSTRACT: Biometric identification has emerged as a reliable means of controlling access to both physical and virtual spaces. The area of biometrics related to 2D face images has been of great importance over years. Current state of art faces identification algorithms focus on accuracy, so work well in terms of response time on small databases only. In real life scenarios, face databases are usually huge and without any intelligent scheme the response time could be really high as these algorithms require an exhaustive search on the database. A good indexing technique is required to narrow down the search space. In our project an indexing technique suitable for 2D Face database is proposed. This thesis classifies the indexing problem into two parts namely feature extraction and indexing. This approach aims at reducing the search space and retention of the final matches in the reduced search space with very high probability. The feature extraction consists of locating landmark feature points in the 2D face images and then producing a feature set using Delaunay Triangulation. The technique of Delaunay triangulation, so far implemented on fingerprint images only, helps in giving a robust feature set invariant of small transformations and luminance variations. This feature set is then indexed through R- trees to reduce the search space.

KEYWORDS: Biometric, Indexing, SVM, classification.

I. INTRODUCTION

Biometric identification has emerged as a reliable means of controlling access to both physical and virtual spaces. The area of biometrics related to 2D face images has been of great importance over years.

Current state of art faces identification algorithms focus on accuracy, so work well in terms of response time on small databases only. In real life scenarios, face databases are usually huge and without any intelligent scheme the response time could be really high as these algorithms require an exhaustive search on the database. A good indexing technique is required to narrow down the search space.

There are various ways to solve the identification problem. Simplest approach does an exhaustive search on database. This approach is very costly for large database. The second approach is based on classification, where database is divided into small number of classes. A query face could belong to one or more classes. Further matching is done only with members of the class to which the query belongs. Third approach, a candidate set is produced for matching. So, instead of searching through the complete database for a match, only the candidate set is searched. This makes the identification process cheap. No similar indexing work as proposed in this thesis has been done yet.

The problem is to reduce the number of templates/IDs to be considered for a match by an identification system. The set produced for identification system and containing possible match ID is called 'Candidate set'. It is also called reduced search space. The probability that possible match is there in the search space is called 'Hit Rate'. It is desired that the result candidate set should retain the match, if any with a very high probability i.e. the hit rate must be high. High hit rate also ensured that the accuracy of underlying identification algorithm would not be hampered. The fraction of the database being touched to produce candidate set is called 'penetration rate'. Candidate set must be generated quickly and cheaply in terms of computation. This means penetration rate must be low. Formally, the problem is defined as follows,

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

“If there is face database D containing a large number of images and let a query face image be Q , produce a candidate set C such that if Q has a match M in database D i.e. M is in D and M matches Q , then C must contain M with a high probability .”

II. RELATED WORK

1. Face recognition algorithms use subspace methods such as PCA, ICA, and SVM etc. for classification, but to the best of our knowledge, there are no direct use of classification or indexing in literature just to reduce the search space for face recognition. An indexing technique by classifying data based on Gabor features, k-means clustering has been proposed by Jai Kumar Singh. [12].
2. In this work, an approach to reduce the search space for 2D frontal face recognition has been proposed. It is based on extraction of landmark facial feature points (such as corner points of eyes, mouth, nose etc.).
3. These points are then considered to form Delaunay Triangulation which helps in finding robust features. The values obtained from each triangle form the key for R trees.

III. PROPOSED APPROACH

Traditional databases index the records in an alphabetical or numeric order for efficient retrieval. In biometric templates, there is no natural order by which one can sort the biometric records, making indexing a challenging problem. Indexing the database implies a logical partitioning of the data space. In this approach of search-space reduction we make use of a Tree structure for organizing the data, such that each of the leaf nodes store one or more biometric templates. Thus given a test template, only the templates having similar index values would be considered as the search space, or in the case of range search, only the fraction of the database lying within the range of the indexes of the test template would be the new search space. The reduced search space could then be passed on for actual identification using a slower but a more reliable biometric such as fingerprint. With Biometric data being inherently multi-dimensional, it is indispensable that the indexing technique support multi-dimensional data. Besides supporting high-dimensional data, there are several additional requirements that the indexing technique must satisfy for lending itself to the biometric domain, such as the following:

1. Tree should be approximately balanced

A balanced Tree will ensure that the time needed to reach to every leaf node is nearly the same among all the possible paths from the root to the leaf.

2. Tree should be dynamic

The Tree structure used must be dynamic with respect to frequent insertions and not so frequent deletions. Since we refer to an online real-time application, such as an airport deployment, the number of new enrollments into the database can be expected to be ever increasing. The Tree should not lose its original properties of being approximately height balanced or being non-skewed, with these online operations of insertions and deletions.

3. Tree should support range queries

Since the feature values measured for a biometric template is dependent on the data acquisition device and the feature extraction algorithm, we would want to provide a tolerance to each of the features measured, based on the intra-user variance observed for the feature. Thus in effect we would place the template into a bounding box, such that the extent along each dimension signifies the tolerance we provide for that dimension/feature. In such cases, the effective search space would be the candidates that lie within this hyper-rectangle. Indexing techniques for biometrics must hence be able to support range query searches.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

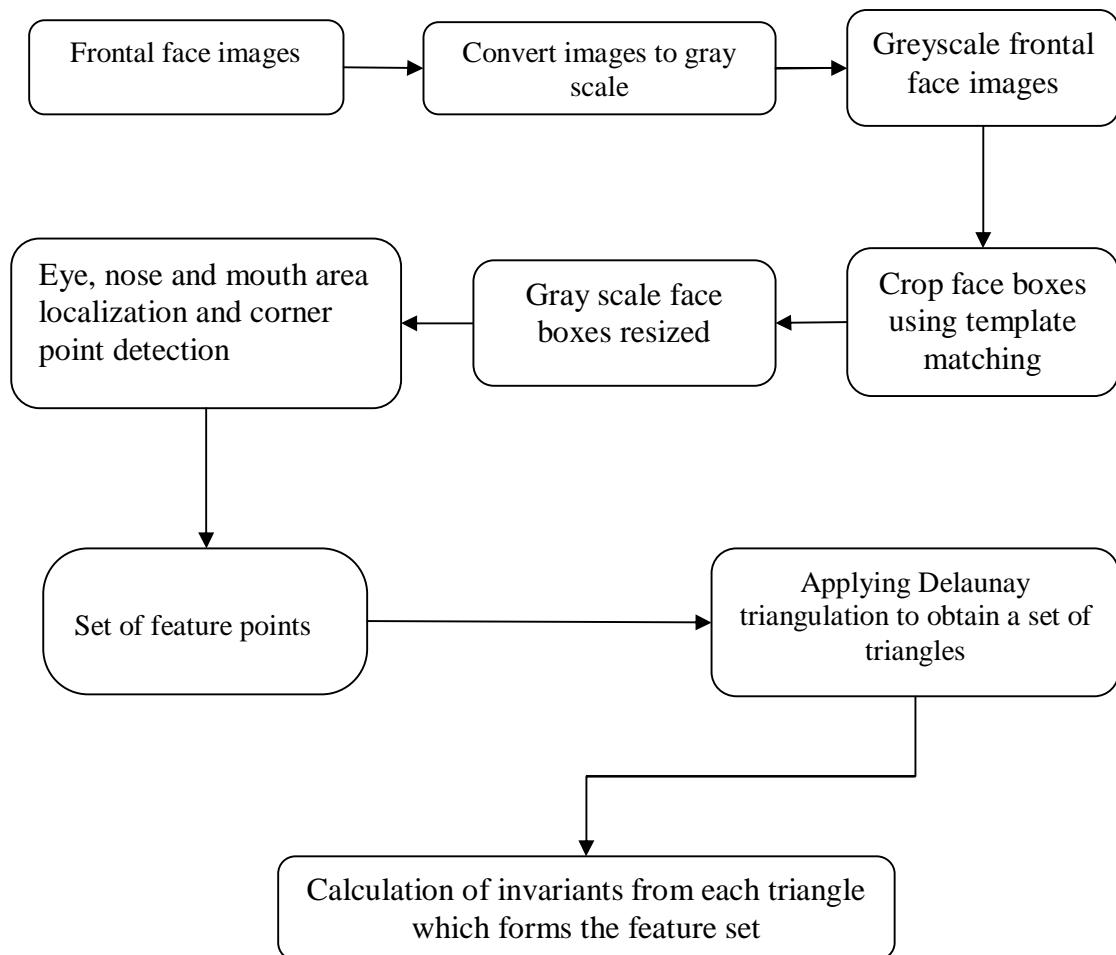
4. Tree should not have overlapping intermediate nodes

Overlapping intermediate nodes imply that while parsing through the tree to reach a certain leaf node, we might have to check different paths and thus search in several different leaf nodes.

5. Scalability

Since we consider real-time applications, we can expect the database size to scale to millions of records. The indexing technique we use should be able to handle such a large amount of high-dimensional data.

III. (I) (FEATURE EXTRACTION BLUEPRINT)



III. (II) R TREE INDEX STRUCTURE

An R-tree is a height-balanced tree similar to a B-tree with index records in its leaf nodes containing pointers to data objects. Nodes correspond to disk pages. If the index is disk-resident, and the structure is designed so that a

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

spatial search requires visiting only a small number of nodes. The index is completely dynamic; inserts and deletes can be inter-mixed with searches and no periodic reorganization is required.

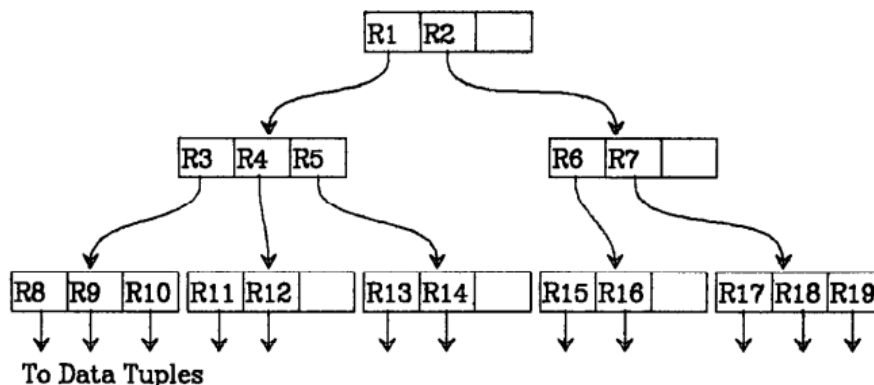
A spatial database consists of a collection of tuples representing spatial objects, and each tuple has a unique identifier which can be used to retrieve it. Leaf nodes in an R-tree contain index record entries of the form (I, tuples-Identifier) where tuples-Identifier refers to a tuple in the database and I is an n-dimensional rectangle which is the bounding box of the spatial object indexed. The non-leaf nodes contain entries of the form (I, child-pointer). Child pointer is the address of a lower node in the tree I covers all rectangles in the lower node's entries.

Let M be the maximum number of entries that will fit in one node and let $m \leq M/2$ be a parameter specifying the minimum number of entries in a node.

An R-tree satisfies the following properties:

1. Every leaf node contains between m and M index records unless it is the root.
2. For each index record (I, tuple-identifier) in a leaf node, I is the smallest rectangle that spatially contains the n-dimensional data object represented by the indicated tuples.
3. Every non-leaf node has between m and M children unless it is the root.
4. For each entry (I, child-pointer) for a non-leaf node, I is the smallest rectangle that spatially contains the rectangles in the child node.
5. The root node has at least two children unless it is a leaf.
6. All leaves appear on the same level.

Figure shows the structure of an R-tree and illustrates the containment and overlapping relationships that can exist between its rectangles. The height of an R-tree containing index records is at most $\log_m N - 1$ because the branching factor of each node is at least m. The maximum number of nodes is $\lceil \frac{N}{m} \rceil + \lceil \frac{N}{m^2} \rceil + \dots + 1$ worst-case space utilization for all nodes except the root is $\lceil \frac{m}{M} \rceil$. Nodes will tend to have more than m entries, and thus will decrease tree height and improve space utilization. If nodes have more than 3 or 4 entries the tree is very wide, and almost all the space is used for leaf nodes containing index records.



IV. SEARCHING

The search algorithm descends the tree from the root in a manner similar to a B-tree. However, more than one subtree under a node visited may need to be searched; hence it is not possible to guarantee good worst-case performance. Nevertheless with most kinds of data the update algorithms will maintain the tree in a form that allows the search algorithm to eliminate irrelevant regions of the indexed space, and examine only data near the search area.

In the following we denote the rectangle part of an index entry **E** by **EI**, and the tuples-identifier or child-identifier part by **E_p**.

Algorithm Search: Given an R-tree whose root node is T, find all index records whose rectangles overlap a search rectangle S.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

1. [Search sub trees] If T is not a leaf, check each entry E to determine whether EI overlaps S. For all overlapping entries, invoke **Search** on the tree whose root node is pointed to by **Ep**.
2. [Search leaf node] If T is a leaf, check all entries E to determine whether EI overlaps S If so, E is a qualifying record.

V. EXPERIMENTAL RESULTS

The test images were randomly chosen from the face database. These images were then used as query template and the results were recorded. The database consists of several images of the each individual. The results of the search query consist of a fraction of IDs. The results represented vote count for each ID. The results were sorted. The top three vote counts were obtained. The IDs having these vote counts were taken as the Candidate set. If this resulted in more than 10 % of the original database, then only top 10 % values from the result of vote count were included.

If the image imprint was already present in the database, the hit rate was trivially, 100 %. The hit rate was found to be approximately 90% in the case there were other imprints of the individual, but not the one being used as a query.

VI. FUTURE SCOPE

There are few scopes of future work to improve the indexing performance:

1. A more efficient point localization scheme would lead in calculation of feature vectors that would be robust to different imprints of the same individual. Thus, resulting in larger hit rate and low penetration rate.
2. There is a scope to parallelize the approach to further enhance the response time of the scheme. For example, query part can be parallelized by taking each query (Ith triangle) separately on parallel processors.
3. The approach must be tested on other databases as well to check the efficiency.
4. The size of database must be further increased to check the response time, penetration rate and hit rate. On very large databases such as ranging from 10,000 to 20,000 the approach for building R trees can be modified.

VII. CONCLUSION

A scheme for indexing of face database is proposed. This takes forward the implementation of Delaunay triangulation to obtain robust feature sets, which were formerly used in case of fingerprint data. The results showed that the features obtained were robust and also helped in classification to reduce the size of R trees.

A significant amount of reduction in search space was obtained. The results depend greatly on the efficiency of feature point detection scheme. The approach used shows good performance and can be further modified to check its efficiency

REFERENCES

- [1] R. Brunelli and T. Poggio, "FaceRecognition:Features versusTemplates", IEEE Trans on PAMI,1993, 15(10), pp 1042-1052.
- [2] S. Y. Lee, Y. K. Ham and R. H. Park, "Recognition of HumanFront Faces Using Knowledge-Based Feature Extraction and Neuro-Fuzzy Algorithm", Pattern Recognition, 1996,29(11), pp 1863-1876.
- [3] R. S. Feris, T. E. de Campos and R.M. Cesar Junior, "Detection and tracking of facial features in video sequences", Lecture Notes in Artificial Intelligence, 2000, 1793(4), pp 127-135.
- [4] G. Chow and X. Li, "Towards A System for Automatic Facial Feature Detection", Pattern Recognition, 1993, 26(12), pp 1739-1755.
- [5] M. Gargesha and S. Panchanathan, "A Hybrid Technique for Facial Feature Point Detection", IEEE Proceedings of SSIAT'2002, 2002' pp 134-138.
- [6] R. S. Feris, J. Gemmell, K. Toyama and V. Kruger, "Hierarchical Wavelet Networks for Facial Feature Localization", Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR.02) , 2002, pp 118-123.
- [7] Searching in high dimensional spaces. Bohm, Berchtold, Keim. ACM Computing Survey, Volume 33, September 2001
- [8] R-Trees: A dynamic index structure for spatial searching, ACM SIGMOD inter-national conference on Management of data, 1984
- [9] A. Samal and P.A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: a survey // Pattern Recognition.-1992.- Vol. 25.-P.65-77.
- [10] R. Chellapa, C.L. Wilson, S. Sirohey and C.S. Barnes, Human and machine recognition of faces: a survey // Proc. of IEEE.-1995.- Vol. 83.- P.705-739



ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

- [11] N. Otsu, A threshold selection method from the gray-level histograms // IEEE Trans. on Syst.,Man, Cybern.-1979.- Vol. SMC-9.- P. 62-67
- [12] Jai kumar singh: "A clustering and indexing technique suitable for Biometric databases"
- [13] J.C. Gee and D.R. Haynor, Rapid coarse-to-fine matching using scale-specific priors // MedicalImaging, SPIE Conf.-1996.- Vol.2710.- P. 416-427Jai kumar singh: "A clustering and indexing technique suitable for Biometric databases"
- [14] de Berg, Mark; Otfried Cheong, Marc van Kreveld, Mark Overmars (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag. [ISBN 978-3-540-77973-5](https://doi.org/10.1007/978-3-540-77973-5). <http://www.cs.uu.nl/geobook/interpolation.pdf>.