# Performance Evaluation of Velocity Partitioned Moving Object Indexing in the Context of PCA and ICA

G. Jaya Suma, PhD
Associate Professor
Department of Information Technology
University College of Engineering, JNTUK

P. Lavanya
Department of Computer Science Engineering
University College of Engineering, JNTUK

## ABSTRACT

With the advent of new mobile services and access feasibility, moving object indexing has grabbed a great number of new research techniques and challenges. But the data exploitation in this area is based on the straitened or symmetric data. The real world data is multi-dimensional and the objects like GPS enabled devices, flights and vehicles on road networks would be skewed in nature. Velocity of an object is of utmost importance as these objects are dynamic. Velocity Partitioning (VP) is a method to cater the need and to support the real world multi-dimensional data. This technique not only improves the query processing but also indexes the moving objects efficiently. In this paper, Velocity Partitioning technique is adopted and Independent Component Analysis (ICA) technique is used instead of Principle Component Analysis to find velocity axis. In the experimental section, different real-time and synthetic datasets are considered; Indexed the moving objects in both VP with PCA and ICA cases. Indexing structures like R*-tree and $B^+$-trees are used for making the analysis. The performance of state-of-the-art indexing structures like $B^x$ and TPR* trees can be improved if VP is applied afore.

## General Terms

Velocity Partitioning, Indexing.

## Keywords

Global Positioning System (GPS), R*-tree, $B^+$-tree, $B^x$-tree ,TPR*-tree, Principal Component Analysis, Independent

## 1. INTRODUCTION

Rapid deployment of mobile devices and geo-positioning systems being capable of communicating wirelessly leads to efficient management of data and tracking of continuously moving objects (individuals with mobile phones, vehicles with on-board navigation systems) has become feasible in terms of implementation cost and technology. The location based services involve current, near future locations querying of moving objects. Indexing has become even more important for spatio-temporal data and for complex data, due to the high performance requirements and proper access methods should be used for efficient querying.

Objects in the real world travel mostly in few predominant directions and routes, say people walking on the roads, flights, Vehicles etc. The query search space expands based on the velocity distribution of the objects in the index tree. The search space expansion varies depending upon the indexing structure used i.e. for TPR*- tree [5] the expansion is due to the Minimum Bounding Rectangles (MBR) and for $B^x$-tree [6] it is query expansion. The search space enlargement in both the x and y-axes is due to the random velocity distribution.

With this observation it is clear that the search space expansion is mainly based on the spread of the data and as with the predominant routes and directions [1], the data spread is along an axis. Velocity Partitioning got triggered from this observation. VP speeds up the query processing by exploiting skew in velocity distribution using the moving object indexes.

The VP technique's initial step is to find out the Principal Velocity Axes (PVAs) for which a combination of Clustering and finding correlations among the attributes of data so as to find the dominant axes is used. For the clustering process k-means clustering is used, for finding the dominant axes (for finding the correlations and variances) Independent Component Analysis is used. The PVA is an axis to which velocities of the objects are parallel. With this PVA as axes for the underlying coordinate system, objects along this PVA are given a moving object index. Objects would change their position as they are mobile and the objects would move between one PVA to another according the PVA index will also be changed. The objects whose distance from the PVA is more and the objects which do not belong to any PVA are regarded as outlier objects. The objects identified as outlier are given a different index called outlier index. So it is understood that objects in the PVA index would move linearly along the PVA, with which it is evident that objects are moving in a near 1-dimensional space which results in reduction of search space from quadratic function of maximum speed to linear function of maximum speed. Extensive set of experiments conducted using real and synthetic datasets sows that VP improves the performance of the index structures consistently.

This paper goals are 1) Improved performance when VP is used. 2) Usage of Independent Component Analysis for PVA identification and maintaining separate indexes. 3) Performance comparison of PCA and ICA. 4) Identification of outlier objects. 4) Applying VP on index structures like B+ and R* trees.

## 2. RELATED WORK

A preface for moving object indexing and brief review of the techniques used in this paper is provided below.

### 2.1) Moving object illustration &querying:

The real time scenario of spatio-temporal moving objects can be considered as Client-server architecture where the client sends its information to the server at discrete time intervals. The server takes this information from these mobile objects to provide services. The other way is to take samples of locations of moving objects periodically, but this process requires enormous update rate. To reduce this high reporting rate a linear function describing the trajectory can be used. For this, initial location and velocity of the object are required and an update is made only when the velocity of the object changes. An update involves one deletion and insertion of velocity information. This model is used in many other indexing and querying scenarios. The different types of range queries are: Time slice range query, Time interval range query, moving range query. VP works for all the three types of queries.

## 2.2) Independent Component Analysis:

Independent Component Analysis (ICA) [8] is a computational method in signal processing for separating a multivariate signal into additive subcomponents and that they are all statistically independent of each other. ICA is an advanced version of Principal Component Analysis (PCA). ICA maximizes the statistical independence of the estimated components to find the independent components. Algorithms of ICA typically use centering i.e. subtract the mean to create zero mean signal, whitening and dimensionality reduction as preprocessing steps [9].

The data is represented by the random vector $x = (x1,x2\ ,\ ..\ xm)T$ and the components as a Random vector $s = (s_1, .. s_m)^T$.The task is to transform the observed data $x$, using a linear static transformation W as $s = Wx$ into maximally independent components s measured by some function $F(s_1, \ldots s_n)$ of independence.

The most commonly used dimensionality reduction techniques like Factor Analysis (FA) and PCA identify the underlying variables that are uncorrelated with each other. It is desirable to use these methods because the underlying variables that account for a set of measured variables should correspond to physically different processes, which in turn will have outputs uncorrelated with each other.

## 2.3) K-means clustering:

This method is used commonly for grouping of similar objects by partitioning a data set into k- clusters where each cluster contains the points with the value near to its centroid. The process starts by heuristically assigning points to each cluster, as and when new objects arrive, clusters will be refined and centroids will be recomputed [10] .

# 3. LITERATURE REVIEW

Moving object indexing has a great number of recent trends & techniques and an increasing number of database applications deals with spatial objects whose position changes over time. A review of the existing work is given below and some techniques are deliberately discussed here.

**R-trees and R*-trees:** These are tree data structures; used for spatial access methods for indexing multi-dimensional information such. R*-trees [11] is a more optimized version of R-trees and these are used to index the extents of objects including their current velocities. R*-trees supports point and spatial data at the same time. Minimum bounding Rectangle (MBR) is an entry which denotes the extent of an object if it is leaf entry and if it is non-leaf, it is used to bound all the MBRs tightly.

**TPR and TPR*- trees:** R*-tree is still enhanced to index the linear functions of time and velocities which gives us the Time Parameterized R-tree (TPR-tree). So an object is represented with location at that reference time and the velocity of that object. The bounding Rectangles (BRs) used in TPR are in accordance with time, i.e. they are functions of time. Query and update performance would depreciate if no updates occur.

TPR*-tree [5] is a variant of TPR and it applies different algorithms for managing the indices. The data structure is same for both the trees but TPR* -trees tries to optimize the time range queries where as TPR tries to optimize the time slice queries. TPR* takes comprehensive decisions for operations like insertions but in TPR they are level-by level basis. TPR* is strict at tightening of bounding rectangles.

**MOVIES:** Moving object Indexing using frequent snapshots [3] is a novel approach to index moving objects, this paper uses the technique used by a cinematographer where in snapshots of the moving objects are collected and stored for a very short period of time they are called short lived throw away indices, which are memory resident. MOVIES support time parameterized predictive queries and for this it uses Non-predictive and predictive MOVIES algorithms.

**B$^x$-tree**: To map the linear functions, B$^x$ uses a specific transformation [6]. When given a nearby time as argument, a linear function is mapped on to the point location. A space filling curve (e.g. Z-curve/Hilbert curve) applied to obtain 1-dimensional point from four-dimensional. Transformations like window query enlargement are applied to each partition of B+ tree and they depend on the velocities of the objects indexed.

**B$^{dual}$-tree:** This obtains better query performance as it utilizes the velocity information [7]. To map the location and velocity vectors to one-dimensional space from four-dimensional space, B$^{dual}$ uses a four-dimensional Hilbert curve. A B$^{dual}$ comprises of two B+ trees and these two trees are used to swap states for every maximum interval of time. Set of

Moving Object rectangles (MOR) are contained in every internal entry. An MOR is square sized and it has continuous Hilbert values corresponding to the sub tree entries.

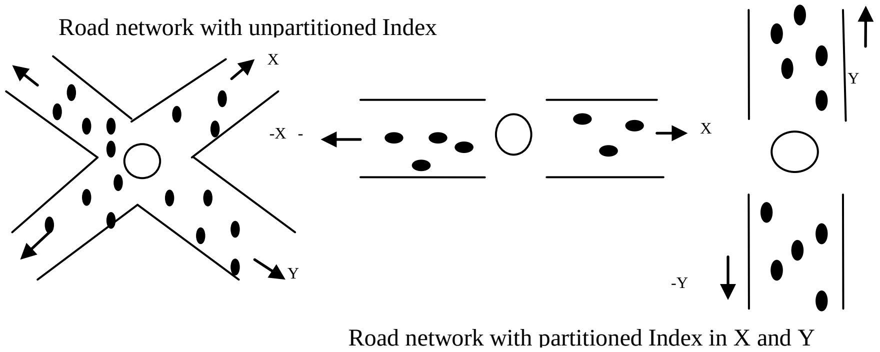
Road network with unpartitioned Index
X
-X -
X
Y
Y
-Y
Road network with partitioned Index in X and Y

**Figure 1: Pictorial representation of Road Network with Objects**

## 4. METHODOLOGY

The paper [1] worked upon the Principal Component Analysis (PCA) for finding the principal velocity axis. But as it was known that PCA makes the data uncorrelated and making the data uncorrelated is not sufficient for further processing. ICA makes the data statistically independent. Even if the ICA is superficially related to these classical techniques like PCA, Factor Analysis; it is capable of finding the essential factors even if these methods fail completely.

Usage of ICA would make the results accurate and promising. The motto of the Velocity Partitioning technique is to identify the Principal Velocity Axes (PVA) out of the independent components obtained using Independent component Analysis (ICA).This axes will have objects along it and these objects are indexed with reference to this line as axis of the underlying coordinate system. The usage of Partitioned index would certainly reduce the time required in searching an object. Because, if not partitioned; all the objects will be given same index and searching of an object would require twice the amount of time required for searching a partitioned index. If partitioned, index of objects along one axis will be different from the other.

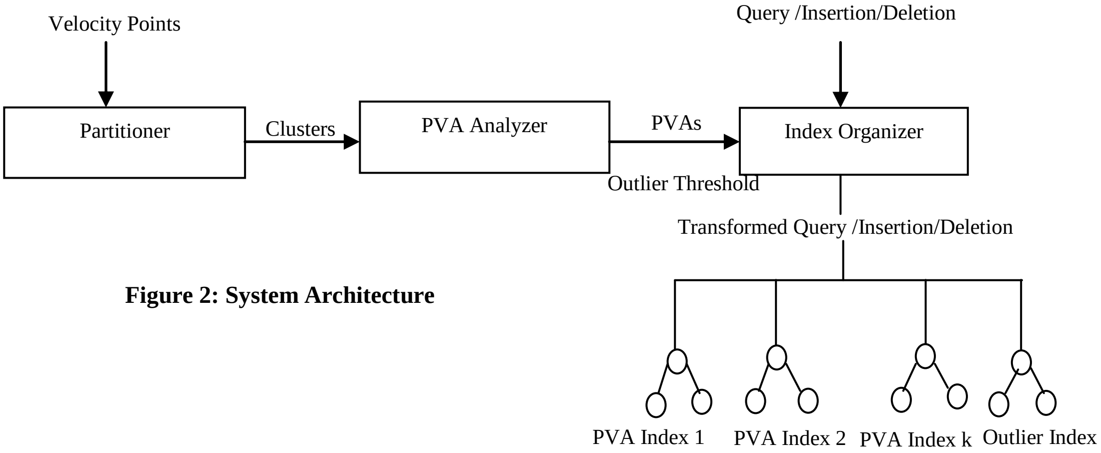

**Figure 2: System Architecture**

When given a query the data space accessed is called search space. A cost model proposed by Tao et al. [5] is used to make an estimate of the number of tree nodes accessed to which the search space is proportional. If S is a search space then S`$_x$ is the search space in x-axis and in y-axis it is S`$_y$ i.e. the search space of partitioned index as shown in Figure 1. The expansion of search space and the search area function are adopted from Thi nguyen [1]. An equation expressing the combined search space before partitioning is given below. Search area of node N` as a function of time t:

$$A_{N`}(t) = d^2 + 4vtd + 4v^2t^2$$

Here d is the maximum length of all nodes. The search space of Partitioned index will be as follows and the combined search space is $A_{CN`}(t)$ is:

$$A_{CN`} = 2d^2 + 4dvt$$

As sown in the Figure 2, architecture of the VP contains three core components, a partitioner, PVA analyzer, Index organizer. The partitioner takes the velocity points as input and partitions or makes groups of the given data. The clusters formed are given to the PVA analyzer which applies Independent Component analysis and makes the data independent and gives as output set of independent components, so the PVA analyzer analyzes the more variant component out of the obtained ones. And it also estimates the outlier threshold. Index Organizer is the other main component which builds the indexes for the obtained PVAs and the outlier data. This component gets the input from PVA analyzer for transforming the query insertion/deletion operations so as to operate on the PVAs and outliers. For the construction of Index trees for PVA data, Index Organizer uses the transformed coordinate space and original data for the outlier partition. B+ and R* trees are used as Index structures. Objects with minimal outlier threshold $\tau = n_d\,(v_{yd}(n_d)\text{-}v_{ymax})$ are inserted into the PVA index by Index organizer and the data far away from PVAs is kept back in outlier index. Query processing would involve transformation of the query into the coordinate space of each index, all the indices are queried and combined results are shown. As it was given in [1] same algorithms can be adopted for the components, with a change where Velocity analyzer component was split into two components namely Partitioner and PVA Analyzer. The partitioner applies k-means algorithm and for PVA Analyzer independent components are found and algorithm required for ICA is FastICA algorithm which includes two preprocessing steps: 1.Centering of the data where in mean is subtracted from each component. 2. Whitening involves linearly transforming the data so as to make the new components uncorrelated and with variance to be an identity matrix.

## 5. EXPERIMENTAL ANALYSIS

Experiments are conducted based on different sizes real time and synthetic datasets defined in Chen et al. [2] for moving object indexes. For the real data set, Oldenburg road network data set is used and for the synthetic data sets Chicago road network is used. A comparison has been made to evaluate the performance of indexing when using PCA and ICA. A sample number of records are considered for indexing through R* and B+ -trees. Below fig 1 and tables depicts the time taken when indexing the moving objects data (Oldenburg ol Road Network).

**Table 1: Time taken for Indexing when Principal Component Analysis is applied**

| Number of Records | Cumulative time for R* & B+-trees(in secs) |
|---|---|
| 5000 | 55 |
| 10000 | 89 |
| 20000 | 219 |
| 30000 | 636 |
| 40000 | 1114 |

**Table 2: Time taken for Indexing when Independent Component Analysis is applied**

| Number of Records | Cumulative time for R* & B+-trees(in secs) |
|---|---|
| 5000 | 58 |
| 10000 | 95 |
| 20000 | 220 |
| 30000 | 640 |
| 40000 | 1120 |

Table1 depicts the performance when Principal Component Analysis is used, number of records used for making the analysis and the cumulative time required for indexing using both the structures R* and B+ trees. The time taken is shown in seconds and time taken for indexing of 5k, 10k, etc. records are shown respectively.

Table2 depicts the performance when Independent Component analysis is used, number of records used for making the analysis and the cumulative time required for indexing using both the structures R* and B+ trees. The time taken is shown in seconds and time taken for indexing of 5k, 10k, etc. records are shown respectively.

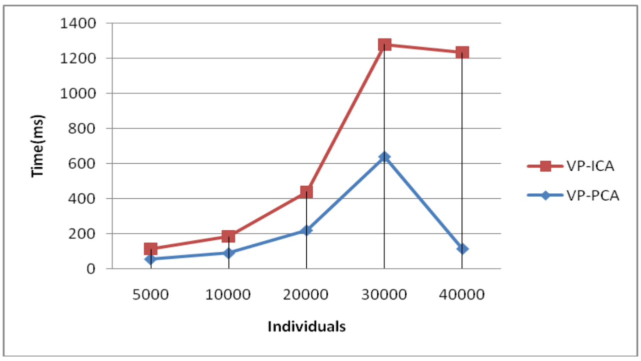

**Figure 3: Comparison of VP-ICA and VP-PCA**

From the above tables and fig-3, it is obvious that with respect to time PCA performs well. And ICA is delayed by some seconds compared to PCA. It is just because ICA finds the independent components with the help of PCA, i.e. as it was previously mentioned; ICA includes PCA in its preprocessing steps. The algorithm for finding Independent components includes a preprocessing step called whitening in which PCA is applied on the given data and then further processing takes place.

## 6. CONCLUSION

Velocity Partitioning is a method which can be applied in any Spatio-Temporal applications that require continuous monitoring and management of moving objects. This technique can be applied on the top of any of the indexing structures like $B^x$ and TPR* tree to improve the functioning of indices creation and management. An empirical performance comparison has made for indexing moving objects using the index structures B+ and R* -trees and different techniques can be adopted based on the kind of application and in consideration with the given specifications. Imbibing ICA into VP would make the results accurate and promising and PCA works faster.

## 7. REFERENCES

[1] Thi Nguyen, Zhen He, Rui Zhang and Phillip Ward. Boosting Moving Object Indexing through Velocity Partitioning. In *VLDB*, 2012.

[2] S. Chen, C. S. Jensen, and D. Lin. A benchmark for evaluating moving object indexes. *PVLDB*, 1920:1574-1585,2008

[3] J. Dittrich, L. Blunschi, M. Antonio, and V. Salles. Indexing moving objects using short-lived throwaway indexs. In *SSTD*, 2009.

[4] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[5] Y. Tao, D. Papadias, and J.Sun. The TPR*-tree:an optimized spatio-temporal access methods for predictive queries. In *VLDB*,2003.

[6] C.S. Jensen, D.Lin, and N. Tradisauskas. Robust B+ -tree-based indexing of moving objecs. In *MDM*, 2006.

[7] M. Yiu, Y.Tao, and N. Mamoulis. The $B^{dual}$-tree: Indexing moving Objects by space filling curves in the dual space.*VLDB Journal*, 17(3):379-40, 2008.

[8] A. Hyvarien. Survey on Independent Component Analysis. *NCS*, 1999.

[9] A. Hyvarien and Erkki Oja. Independent Component Analysis: Algorithms and Applications. *Neural Networks, 13(4-5):411-430,2000.*

[10] J. MacQUEEN. Some Methods for Classification and Analysis of Multivariate Observations. University of California, Los Angeles.

[11] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger. The R*- tree: An Efficient and Robust Access Method for Points and Rectangles.