A Review on High Dimensional Data Visualization

Vinod S. Bawane [#1], Shireesh P. Bhoyar [#2], Manish P. Tembhurkar [#3]

#1 Student M.E.(II sem), Dept. of CSE, G. H. Raisoni College of Engineering, Nagpur, +91 9096424164.
#2 Student M.E.(II sem), Dept. of CSE, G. H. Raisoni College of Engineering, Nagpur, +91 8698412240.
#3 Asst. Prof., Dept. of CSE, G. H. Raisoni College of Engineering, Nagpur, +91 9890596309.

ABSTRACT

Large high dimensional dataset encloses billions of entries and contains different attributes and relational databases. Data cube aggregation operation is a well-known technique used to implement data-mining for larger size databases. Spatial data are sometimes assumed to have absorbed prohibitively large amount of space, which consequently requires disk storage. Thus it is required to pre-compute every possible aggregate query over the database. Modern scientific applications are generating larger and larger volume of data at ever increasing rate. As datasets become bulkier, exploratory data visualization turns out to be more difficult and complex, and data-fetching turns into a time-consuming process in small devices. Nanocubes are in-memory data structures, specifically designed to speed up queries for multidimensional data cubes, and could eventually be used as a backend for these types of applications. Nanocubes offer efficient storage and querying of large, multidimensional, spatiotemporal datasets and high dimensional datasets.

Keywords: Data Mining, Data visualization, Data warehousing, Datasets, Nanocubes, spatiotemporal dataset,

INTRODUCTION

High dimensional datasets (Spatial data)[1]having different tuples and relational databases, which acquire large amount of space and disk storage. For exploring large, high dimensional data sets, nothing matches the power of interactive visualizations that let users directly control data. With mouse clicks, user can drill down for more detail or zoom out for a broader perspective while quickly seizing on the anomalies and outliers that reveal much about the data [1]. Traditional data visualization tools[2] are often inadequate to handle big data. While it is debatable what is meant by “big”, visualization researchers have regularly used one million or more data cases as a threshold. Research on big data visualization must address two majorchallenges: perceptual and interactive scalability[3]. The resolution of conventional displays (~1-3 million pixels),visualizing every data point can lead to over-plotting andmay overwhelm users’ perceptual and cognitive capacities.On the other hand, reducing the data through sampling orfiltering can elide interesting structures or outliers. Big dataalso impose challenges for interactive explorationsuch as querying large data acquire high latency and disrupting fluent interaction [2].

As datasets become bulkier, data fetching is time-consuming process[1], especially for an android device. New visualization techniques, such as dense pixel displays[4], have

been proposed for dealing with large datasets, but most of these approaches still attempt to draw each item in the dataset. The technique is not practical for enormous datasets. Another solution to this data excess problem that it is technical in nature, to initiate abstraction that reduces the amount of data to exhibit, moreover in data space or in visual space; include hierarchical parallel coordinates, color histograms, and clustered time-series data [1], [2], [4], [5]. Modern data analytics applications involve computing aggregates over a large number of records to roll-up web clicks, online transactions, content downloads, and other features along a variety of different dimensions, including demographics, content type, region, and so on. Traditionally, such queries have been executed using sequential scans over a large fraction of a database. Increasingly, new applications demand near real-time response rates. Examples may include applications that (i) update ads on a website based on trends in social networks like Facebook and Twitter, or (ii) determine the subset of users experiencing poor performance based on their service provider and/or geographic location.

Table1. Analysis of High Dimensional data with respect to time

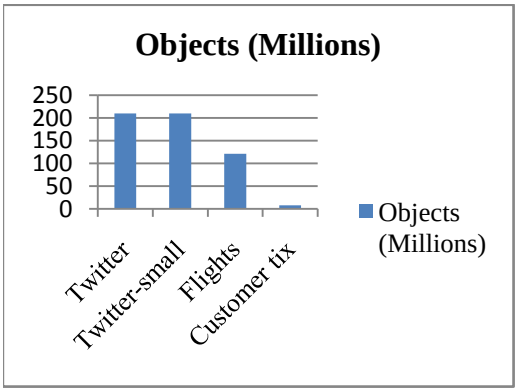
Datasets	Objects (in Millions)	Memory (in GB)	Time (Minute)	Size (no. of nodes in each data structure) (in Millions)	Sharing (Nanocube without sharing mechanism) (Multiplying Factor)
Twitter	210	46.4	352.2	5200	4
Twitter-Small	210	10.2	73.8	1200	3.72
Flights	121	2.3	31.13	274	16.50
Customer tix	7.8	2.5	8.47	213	2.93

When exploring large datasets, analysts often work through a process of "first, zoom and filter, then details-on demand". Multiscale visualizations arewell-organized technique for facilitating this process because they change the visual representation to present the data at different levels of abstraction as the user pans and zooms [6]. At a high level, because a large amount of data needs to be displayed, it is highly abstracted. As the user zooms, the data density decreases and thus more detailed representations of individual data points can be shown [6]. The two types of abstraction performed in these Multiscale visualizations are *data abstraction* and *visual abstraction*.

Data abstractions (e.g., aggregation or selection) change the underlying data before mapping them to visual representations [6]. The major purpose of a database system is to provide users with an **abstract view** of the system. The system hides certain details of how data is stored and created and maintained Complexity should be hidden from database users. Visual abstractions change the visual representation of data points (but not the underlying data itself) to provide more information as the user zooms; e.g., an image may morph from a simplified thumbnail to a full-scale editable version. Existing systems, such as Data Splash and Pad++ [6], focus primarily on visual abstractions with support for data abstractions limited to simple filtering and the ability to add or switch data sources. In addition, these systems primarily only allow for a single zooming path. Our goal is to develop a system for describing and developing Multiscale visualizations that support multiple zoom paths and both data and visual abstraction. We want to support multiple zoom paths because many large data sets today are organized using multiple hierarchies that define meaningful levels of

aggregation (i.e., detail). Data cubes are a commonly accepted method for abstracting and summarizing relational databases. By representing the database with a data cube, we can switch between different levels of detail using a general mechanism applicable to many different data sets. Combining this general mechanism for performing meaningful data abstraction with traditional visual abstraction techniques enhances our ability to generate abstract views of large data sets, a difficult and challenging problem.

Graphical analysis of High Dimensional Datasets with respect to time:



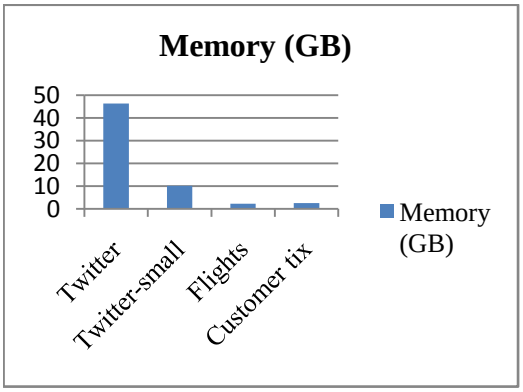
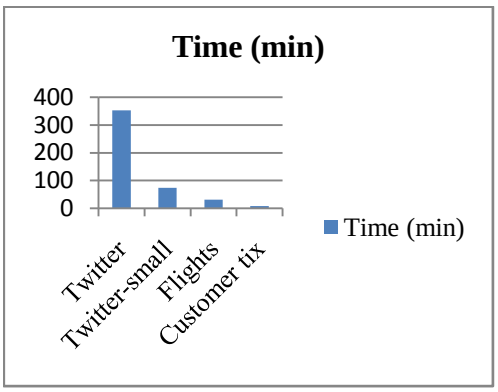


Fig1. Objects in millions Fig2: Datasets memory in GB



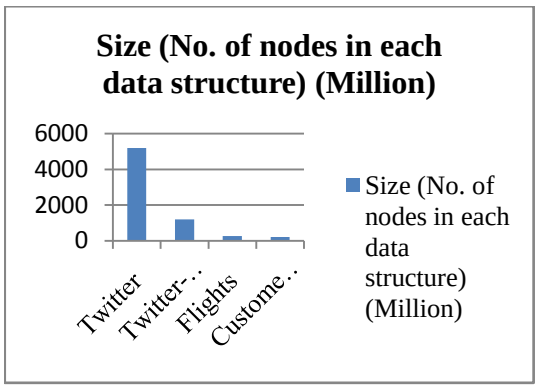


Fig3: Time required visualizing datasetsFig4: Size of number of nodes required

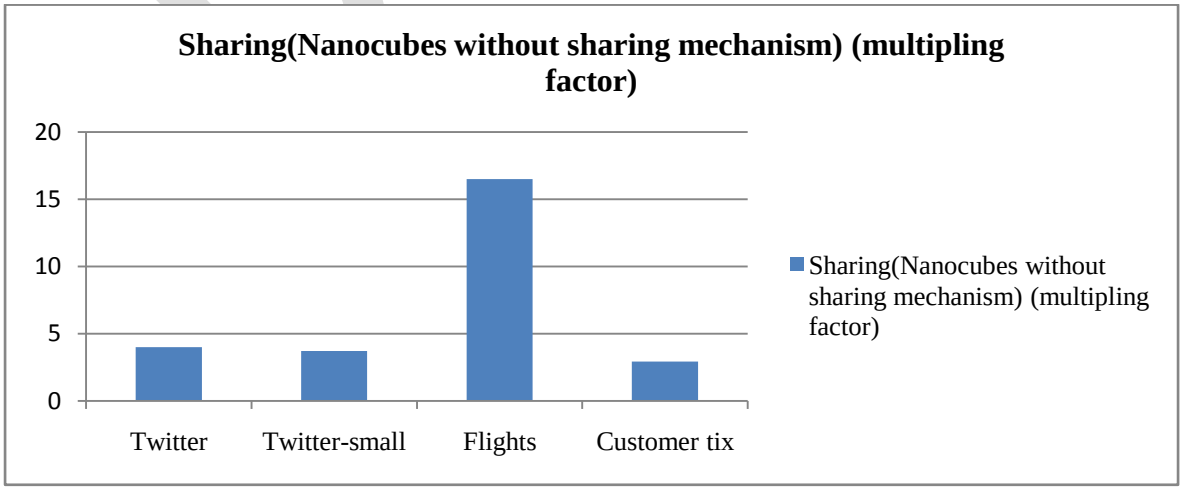


Fig5: Multiplying factor that uses Nanocubes without sharing mechanism

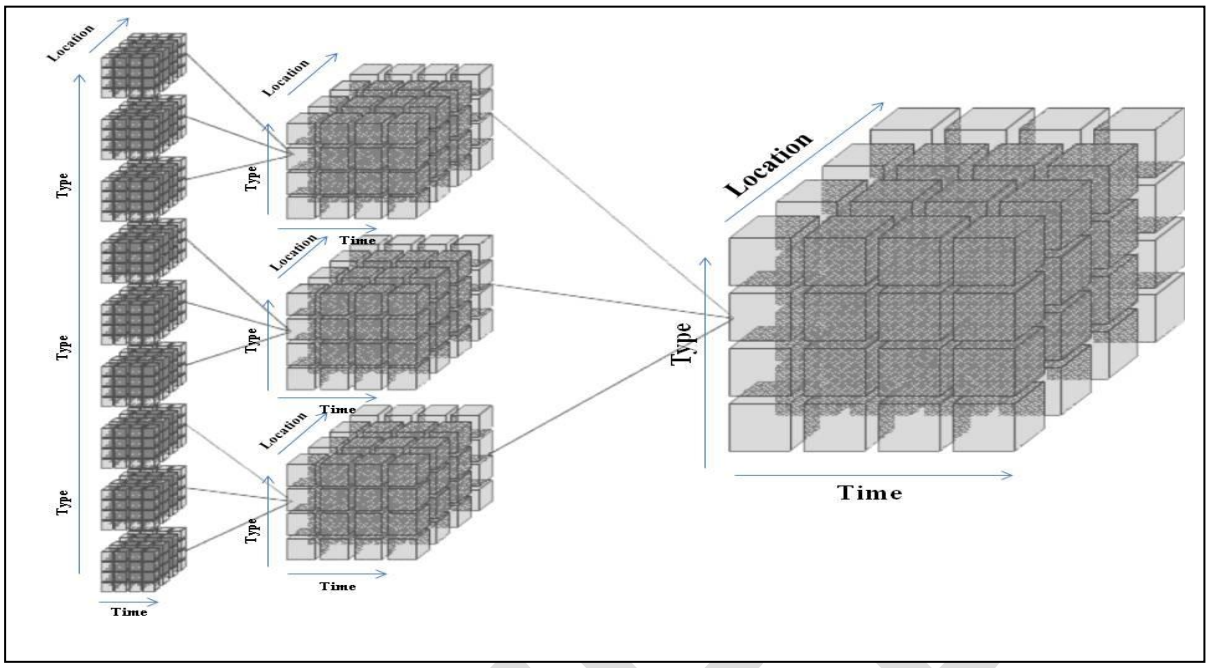


Fig6: Data Visualization

NANOCUBES

Nanocubes provide a possible solution for an efficient storage and to access high dimensional datasets shown in figure 1. As datasets get larger, exploratory data visualization becomes more difficult. Consider a dataset with a billion entries. We can compute a small summary of the dataset and visualize the summary instead of the dataset, summaries might help, but in order to understand if that is the case, we will inevitably find ourselves having to visualize one billion residuals. As far as scale goes, we are back to square one. In other words, data summarization alone will never solve the problem of scale in exploratory visualization. As visualization practitioners, what then can we do? Even drawing the simplest scatter plot is not straightforward. If we decide to produce the visualization by scanning the rows of a table, we will either need non-trivial parallel rendering algorithms or significant time to produce a drawing. Neither of these solutions is attractive or scales well with dataset size. Data cubes are structures that perform aggregations across every possible set of dimensions of a table in a database, to support quick exploration. Many visualization systems are built on top of data cubes, concretely or conceptually. By real-time, we mean query times on average under a millisecond for a single thread running on computers ranging from laptops, to workstations, to server-class computing nodes. By large, we mean that the datasets we support have millions to billions of entries. By spatiotemporal, we mean that Nanocubes support queries typical of spatial databases [1], such as counting events in a spatial region that can be either a rectangle covering most of the world, or a heat map of activity in downtown San Francisco [1]. By the same token, Nanocubes support temporal queries at multiple scales, such as event counts by hour, day, week, or month over a period of years. Data cubes in general enable the Visual Information-Seeking Mantra of "first, zoom and filter, then details on-demand" by providing summaries and letting users drill down by expanding along the wanted dimensions [4]. Nanocubes also provide overviews, filters, zooming, and details-on-demand inside the spatiotemporal dimensions themselves. By

multidimensional, we mean that besides latitude, longitude, and time, each entry can have additional attributes that can be used in query selections and rollups.

BACKGROUND

A. Datacubes: -Data cubes are structures that perform aggregations across every possible set of dimensions of a table in a database, to support quick exploration. Many visualization systems are built on top of data cubes, concretely or conceptually. Still, only recently have researchers started to examine data cube creation algorithms in the context of information visualization.Data cubes are often problematic in that they can take prohibitivelylarge amounts of memory as the number of dimensions increases. Data cubes in general enable the Visual Information-Seeking Mantra of "first, zoom and filter, then details on-demand" by providing summaries and letting users drill down by expanding along the wanted dimensions [4]. The CUBE operation is the result of collecting all possible GROUP BY aggregations into a single relation for a given list of attributes.

B. Aggregation: - Hierarchical aggregation in visualization is based on aggregation in data space and corresponding simplified visual representations of the aggregates in chart space [1, 4]. Basically, the aggregation process turns visualization into a Multiscale structure that can be rendered at any desired level. This provides the user with a convenient overview that hides any clutter arising from details in the dataset while still giving a reasonable indication of data size through the visual aggregates. A set of basic interaction techniques allow for navigating the structure. The visual aggregates can express different information about the underlying data items, such as their average,or even their allocation. Also, the interface allows the user to drill down and retrieve details on-demand. Like this, hierarchically aggregated visualization techniques directly support the visual information seeking mantra "first, zoom and filter, then details on-demand."

C. Visualization: -In this, here review several existing Multiscale visualizationsystems, focusing on how the systems perform both data and visualabstraction. Data abstraction refers to transformations applied tothe data before being visually mapped, including aggregation, filtering,sampling, or statistical summarization. Visual abstractionrefers to abstractions that change the visual representation,change how data is encoded in the retinal attributes of the glyphs, or apply transformations to the set of visual representations. Visualization is of two types such as:

***1.* Multiscale Visualization in Cartography: -** Cartography is the source of many early examples of Multiscale visualizations. Cartographic generalization refers to the process of generating small scale maps by simplifying and abstracting large scale source material and consists of two steps: (1) employing selection to decide which features should be shown and (2) simplifying the visual representations of the selected features. A map series developed using this process and depicting a single geographic region at varying scales is a Multiscale visualization. While the initial selection process is a specialized form of data abstraction, the subsequent manipulations are all visual abstractions.

***2.* Multiscale Information Visualization: -** Several information visualization systems provide some form of zooming or Multiscale interface. Given our goal in expressing general

Multiscale visualizations, we only discuss general systems; domain-specific tools may apply both data and visual abstraction but their abstractions are not applicable.

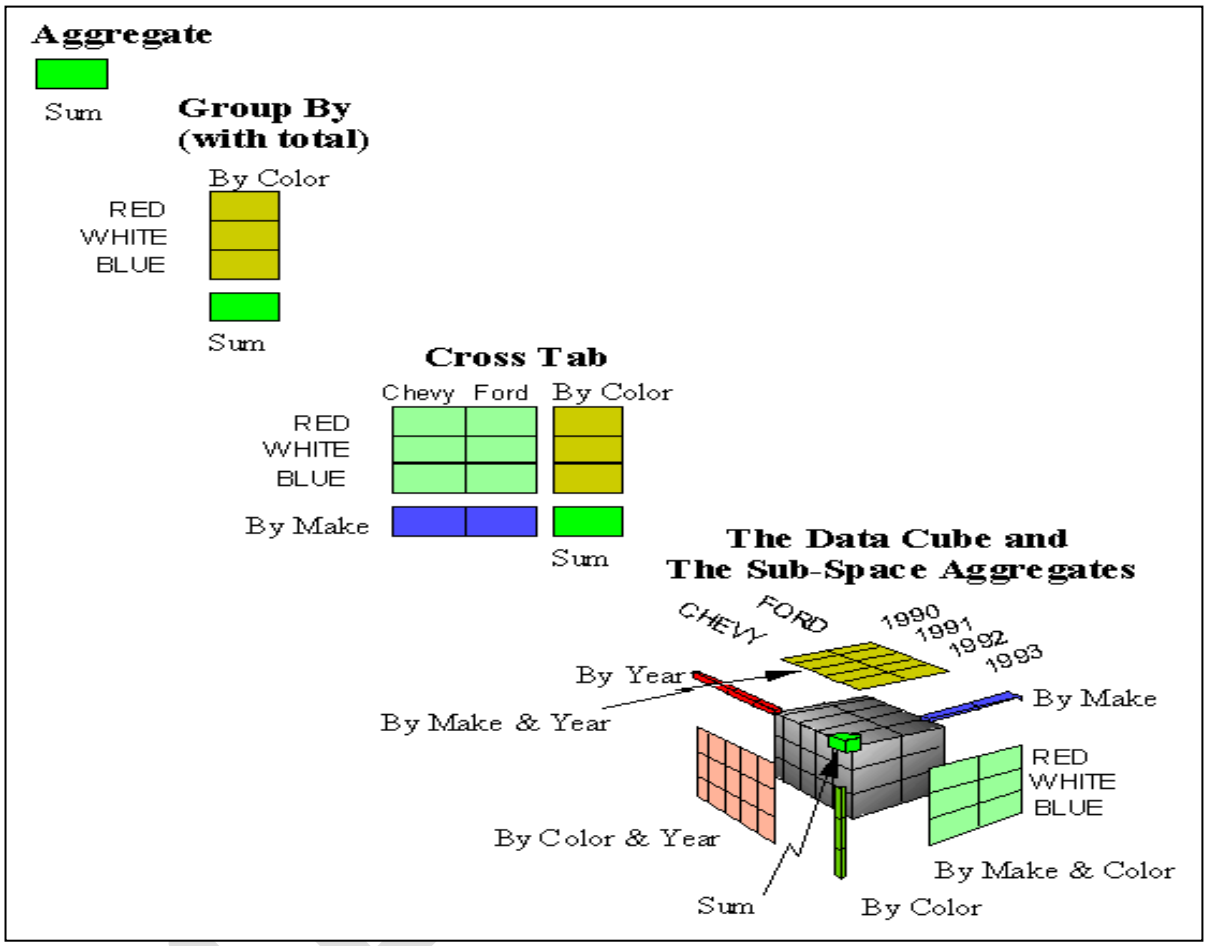


Fig7: Data cube aggregation

CONCLUSION

Using Nanocubes techniques we are able to

- accessing high dimensional data sets
- Performing data mining with the help of Nanocubes.

It can also offer efficient storage and querying large multidimensional datasets. This technique is helpful in all relational datasets which contain billions of entries. Hence data visualization of spatiotemporal dataset is more efficient. It will also help to improve speed of data visualization in small devices.

REFERENCES

[1] LauroLins, James T. Klosowski, and Carlos Scheidegger, "*Nanocubes for Real-Time Exploration of Spatiotemporal Datasets*", IEEE Transactions on Visualization And Computer Graphics, Vol. 19, No. 12, December 2013.

[2] Zhicheng Liu, BiyeJiangz and Jeffrey Heer, "*imMens: Real-time Visual Querying of Big Data*", Eurographics Conference on Visualization (EuroVis), 2013.

[3] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb, "*Queries with bounded errors and bounded response times on very large data*", In Proceedings of EuroSys, to appear. ACM, 2013.

[4] N. Elmqvist and J. D. Fekete, "*Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines*", IEEE Transactions on Visualization and Computer Graphics, 16(3):439–454, May 2010.

[5] Q. Cui, M. Ward, E. Rundensteiner, and J. Yang, "*Measuring data abstraction quality in multiresolution visualizations*", IEEE Transactions on Visualization and Computer Graphics, 12(5):709–716, Sept. 2006.

[6] C. Stolte, D. Tang, and P. Hanrahan, "*Multiscale visualization using data cubes*" , IEEE Transactions on Visualization and Computer Graphics, 9(2):176–187, 2003.

[7] *"Real – Time exploration of spatiotemporal datasets"* Official web site http://www.nanocubes.net

[8] American Statistical Association Data Expo. Flights dataset, 2009. http://stat-computing.org/dataexpo/2009

[9] F. J. Anscombe. Graphs in statistic analysis. The American Statistician, 27(1):17–21,1973.

[10] M. Bostock, V. Ogievetskey, and J. Heer. D3: Data-driven documents. IEEE Transactoins on Visualization and Computer Graphics, 17(12), 2011.

[11] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield. Scat-terplot matrix techniques for large n. Journal of the American Statistical Association, 82(398), 1987.

[12] S.-M. Chan, L. Xiao, J. Gerth, and P. Hanrahan. Maintaining interactivity while exploring massive time series. In IEEE Symposium on Visual Analytics Science and Technology, pages 59–66. IEEE, 2008.

[13] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In Proceedings of SIGKDD. ACM, 2011.