

University of Wales Trinity St David
School of Computing

Visualisation Methods for Big Data

Richard Green
1704571

Foundation BSc Computing

Academic Writing (M4X01498)

Contents



1	Introduction	2
1.1	The Big Data Pipeline	2
2	Visualising Big Data	3
2.1	Challenges	3
2.2	Current Approaches	4
2.3	Dynamic Data	4
2.3.1	Time-to-time (Animation)	5
2.3.2	Time-to-space (Timeline)	5
3	Conclusion	6

1 Introduction

Big data describes a large volume of data that may be structured, unstructured, or somewhere in between. This data is incredibly difficult to analyse using conventional methods due to the size of the datasets and is generally characterized by three defining properties: volume, variety, and velocity [1].

Datasets that fall under the ‘big data’ umbrella are becoming more common due to the relatively low cost of digital memory and the ease of data gathering from different sources [2]. Big data sources tend to require the use of parallel computing tools to handle the volume of data.

Visualising big data occurs after the data has been reduced in some way via an analysis procedure. This is referred to as the ‘pipeline’. Visualisation is the final step in the pipeline. Processing and analysis is split in this manner as it is not feasible to visualise terabytes of data and expect a user to derive meaning from the resulting image [3].

Visualisation is a powerful tool for conveying information more quickly and more effectively than words. This is particularly powerful for Big Data where visualisation tools allow large amounts of data to be reviewed for trends and correlations (among others) very quickly [2]. It is also a useful way of presenting data findings to others. 

1.1 The Big Data Pipeline

Wu et al. [4] propose a Big Data processing framework that has a three tier structure centering around a big data mining platform (Tier I). The second tier focuses on high-level semantics of information sharing, privacy, and knowledge. The third and final tier focuses on challenges for data mining algorithms applied to the data.

With this in mind it is possible to generalise a Big Data pipeline to six steps:

- Data Mining
- Data Collection
- Data Storing
- Data Cleaning
- Data Analysis
- Data Consumption

Big Data visualisation sits primarily in the final step, though some preliminary visualisations may be carried out at the analysis stage to aid in explorations of the data.

2 Visualising Big Data

Visualisation has proven effective for both presenting information and driving complex analyses [5] and most fields will eventually encounter the need to visualise a large amount of data. However, there is no consensus in the visualisation field for specific methods of handling big data; though many have been proposed.

2.1 Challenges

The ‘3V’ properties of Big Data often cause multiple problems that mean that visualisation is not an easy task. Scalability and dynamics are two of the major challenges [6].

Visual Noise Due to the massive amount of information in these datasets it is not sufficient to just display all the data on the screen simultaneously - even with the use of a large display it is impossible to accurately convey data meaning simply due to the loss of data visibility [6]. There are also implications for the mental load of the user as it is not realistic to expect a user to hold terabytes of information in their mind and carry out a full, accurate, analysis procedure [7].

Image Perception There is a threshold for human perception of elements being displayed [8]. This threshold is reached much more quickly if an approach displays too much information resulting in an data overloaded view. This means that visualisation methods for big data are not just limited with computing or display power but also by physical human perception limits [8].

Information Loss In attempt to address the previous challenges it is generally accepted that data must be filtered or sampled in some way before rendering for visualisation is carried out. However, it is possible to oversample or reduce the data so that the analyst cannot see or know about some hidden features in the data [6].

Performance Requirements Live analysis, such as with a visualisation dashboard, requires data filtering and rendering to be carried out in real-time. However, poor rendering or speed performance can lead to user dissatisfaction and poor analysis results [8]. There is a tradeoff between data filtering and increasing the resources available for the analysis but at some point the ceiling threshold for resources will be reached [6]. Dynamic data (data that changes over time) is a particular issue when it comes to visualisation [9].

2.2 Current Approaches

Visualisation techniques commonly provide an overview first, with zooming and filtering used to provide details on demand [10].

These visualisation approaches are generally more ‘dashboard’ based in that they provide multiple, usually linked, representations that are leveraged to provide an interaction layer between the user and the data for analysis [11]. These dashboards tend to offer multiple visualisation representations. These will be discussed below.

Treemaps are a space-filling visualisation intended to be used for hierarchical data. Circle packing and sunbursts are subset-types of treemaps. DAGMap [12] is a visualisation technique which represents directed acyclic graphs in a similar manner to TreeMaps. Research by Koenig et al. showed that DAGMap outperformed node link representations of the same information [12].

Parallel coordinates allow analyses to be extended with multiple data factors for different objects.

Streamgraphs are a type of stacked area graph which displaces data values from a central axis.

Circular network diagrams are also useful for dealing with a large volume and variety of data. 

2.3 Dynamic Data

The visualisation of large dynamic data is a subset of the larger problem of visualising big data. Due to the greater availability of large time-series data from social networks there is a research focus on designing methods for dynamic graph visualisation. It is more challenging than the visualisation of static data due to the requirement to provide some sort of encoding for time. This could be a time-to-time mapping [9] - such as animation - though this has serious implications for user memory and general usability when dealing

with large datasets [7]. An alternative is a time-to-space mapping where time is encoded along some visual axis in the representation [9]. This type of encoding generally performs better for large datasets but the loss of one axis for encoding can be a serious issue for many visualisations [8].

Each type of time encoding can be further reduced into three approaches - node link, matrix, and hybrid.

2.3.1 Time-to-time (Animation)

Combining easy to understand node link diagrams with animation gives a visualisation that seems intuitive and is easy to understand when the data is small and the time period is short. Should the size or length of the data increase, however, the representation quickly becomes unreadable and meaningless for analysis [7]. These methods usually try to provide support for the mental map via dynamic stability (keeping nodes in the same places across time) [7] but there will always be a point at which the memory overhead exceeds user capabilities [7].

There are currently no time-to-time mappings that use any matrix-based approaches [].

2.3.2 Time-to-space (Timeline)

Compared to animation timeline-based approaches provide a better overview of dataset time as they show the complete sequence of graphs as a static image [9]. However, the increased number of representations mean that there is less space available for each visualisation which could potentially reduce readability. This is the primary challenge for visualisations that are in this category.

There are many methods that use node-link diagrams [13]. Node-link diagrams are reasonably easy for the non-expert user to understand, however they have readability problems when the graph becomes large or very dense [13].

For this type of mapping there are also multiple matrix-based methods. Matrix-based methods avoid the ‘hairball’ problem that can occur with node link diagrams but they are often complex to understand and they do not show the underlying graph structure at each time point [14].

3 Conclusion

With the decreasing costs of digital storage and the increasing availability and collection of large amounts of data it is vital that the current data processing pipelines are optimized. Visualisation is the end stage of the pipeline and it is increasingly becoming clear that traditional visualisation techniques are not suitable for large amounts of data [6] due to readability problems and the effect of human perception and memory when dealing with huge amounts of data [8].

The visualisation of big data is a particular issue in the area of social network analytics - it is very hard to understand graph structure without some form of visual aid but current approaches can cause problems with graph evolution events being lost through time aggregation [15].

Future directions seem to be aimed at leveraging machine learning techniques as a method of data reduction so that users are only shown visualisations that have some value for their task. However, machine learning comes with its own set of challenges (GIGO) which will need to be addressed first if these techniques are to be successful.

References

- [1] D. Laney, “3D data management: Controlling data volume, velocity, and variety,” META Group, Tech. Rep., February 2001. [Online]. Available: <http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>
- [2] M. Chen, S. Mao, and Y. Liu, “Big Data: A Survey,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, Apr. 2014. [Online]. Available: <https://doi.org/10.1007/s11036-013-0489-0>
- [3] D. A. Keim, “Information visualization and visual data mining,” *IEEE transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 1–8, 2002.
- [4] X. Wu, X. Zhu, G. Wu, and W. Ding, “Data mining with big data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, Jan 2014.
- [5] D. Keim, H. Qu, and K. Ma, “Big-data visualization,” *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 20–21, July 2013.

- [6] N. Bikakis and T. K. Sellis, “Exploration and visualization in the web of big linked data: A survey of the state of the art,” *CoRR*, vol. abs/1601.08059, 2016.
- [7] D. Archambault, H. Purchase, and B. Pinaud, “Animation, small multiples, and the effect of mental map preservation in dynamic graphs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 4, pp. 539–552, 2011.
- [8] M. Tory and T. Moller, “Human factors in visualization research,” *IEEE transactions on visualization and computer graphics*, vol. 10, no. 1, pp. 72–84, 2004.
- [9] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, “The state of the art in visualizing dynamic graphs,” *EuroVis STAR*, vol. 2, no. 7, p. 11, 2014.
- [10] L. Wang, G. Wang, and C. A. Alexander, “Big data and visualization: methods, challenges and technology progress,” *Digital Technologies*, vol. 1, no. 1, pp. 33–38, 2015.
- [11] A. Lee, D. Archambault, and M. Nacenta, “Dynamic network plaid: A tool for the analysis of dynamic networks,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’19. New York, NY, USA: ACM, 2019, pp. 130:1–130:14. [Online]. Available: <http://doi.acm.org/10.1145/3290605.3300360>
- [12] K.-C. Chen, J. Cong, Y. Ding, A. B. Kahng, and P. Trajmar, “Dag-map: Graph-based fpga technology mapping for delay optimization,” *IEEE Design & Test of Computers*, vol. 9, no. 3, pp. 7–20, 1992.
- [13] N. Kerracher, J. Kennedy, and K. Chalmers, “The design space of temporal graph visualisation,” in *EuroVis*, vol. 14, 2014, pp. 7–11.
- [14] N. Henry and J.-D. Fekete, “Matlink: Enhanced matrix visualization for analyzing social networks,” in *IFIP Conference on Human-Computer Interaction*. Springer, 2007, pp. 288–302.
- [15] D. Greene, D. Doyle, and P. Cunningham, “Tracking the evolution of communities in dynamic social networks,” in *2010 international conference on advances in social networks analysis and mining*. IEEE, 2010, pp. 176–183.