Dynamic Sampling for Visual Exploration of Large Dense-Dense Matrices

Philipp Roskosch[1], James Twellmeyer[2], and Arjan Kuijper[1,2](✉)

[1] Technische Universität Darmstadt, Darmstadt, Germany
[2] Fraunhofer IGD, Darmstadt, Germany
kuijper@igd.fraunhofer.de

Abstract. We present a technique which allows visual exploration of large dense-occupied similarity matrices. It allows the comparison of several dimensions of a multivariate data set. For the visualization, the data are reduced by sampling. The access time to individual elements is an ever increasing problem with increasing matrix size. We examine various database management systems and compare the access times for different problem sizes. The visualization responds to user interaction and allows the focus to specific areas within the data. For this, the data is filtered according to user interests and the visualization is refined with subsamples of the filtered data. The context is preserved in this process. The focus allows the discovery of relationships that would otherwise remain hidden.

1 Introduction and Motivation

VIS-SENSE is an international research project. Fraunhofer IGD is one of six partners involved in this project [4,6,18]. Visual Analytics techniques [12,13] are developed that the handling of large and complex data sets of IT-Security to improve international network security [20,21]. The data are compared with each other. Thus large similarity matrices occur with up to 100,000 $\times$ 100,000 entries. With a visualization of these matrices, homogeneous groups within the data set are examined on similarities or differences [15]. We develop a sampling algorithm that enables displaying all matrices in parallel on a standard computer and monitor. Our algorithm satisfies a real-time requirement, so load times for the user are not noticeable. In addition, we take the idea from [7] and adjust the sample visualized to the user interest. By interacting with the visualization, data with certain properties are increasingly displayed and thus replace less interesting data. For a better overview, we discuss an interaction process and emphasizing added elements. By increasingly choosing additive data with certain characteristics, the distribution of the sample deviates more and more from that of the data set. If the distribution is uniformly distributed at the beginning, it will be more and more adapted to the user's interest through interaction.

S. Yamamoto (Ed.): HIMI 2016, Part I, LNCS 9734, pp. 308–318, 2016.
DOI: 10.1007/978-3-319-40349-6_29

We have implemented the concept of Degree of Interests in a whole new way. In contrast to previous approaches, no DOI function is needed that calculates a value for each date. We work with multivariate data, which have no specific context. Unlike e.g. in geographic data there is no natural distance between objects, but only a calculated similarity. In [7] the DOI is calculated from a specified center. Our data have only an abstract distance or similarity between them. After the initial sample, the user receives context information from the data set. He does not choose a center initially, but rather determines by interaction with the visualization the data that are relevant to him and thereby gradually gets a larger focus on certain parts. We use the characteristics of the data for the determination of *interesting* subsets. Thus, computing power and memory is saved because an explicit calculation of the DOI does not take place of irrelevant data. In [3] it is made clear that each record is just a sample of real-world data. If a sub-sample is used for a visualization the accuracy is only slightly affected. The same is observed for algorithms with higher complexity. A waiver of the last thousandth accuracy enables the calculation of complex algorithms in reasonable time [9]. This behavior is confirmed in this work. It is not possible to visualize the entire data set with all its dimensions in parallel. Sampling allows the examination of a smaller subset of the complete data set with the same statistical properties. The findings and analysis, which are made possible in this way, are of exactly the same quality as when the complete data set is considered. This results from the fact that essential characteristics of the data set are also pronounced available in a sufficiently large sample [10,11].

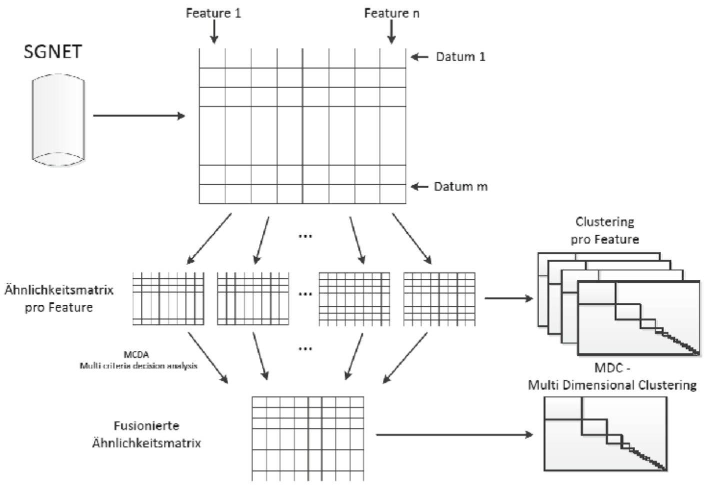
Feature 1
Feature n
SGNET
Datum 1
Datum m
Clustering pro Feature
Ähnlichkeitsmatrix pro Feature
MCDA
Multi criteria decision analysis
MDC - Multi Dimensional Clustering
Fusionierte Ähnlichkeitsmatrix

Fig. 1. VIS-SENSE pipeline.

2 Pipeline

The data for the research extracts VIS-SENSE[1] from WOMBAT[2]. In this project, safety raw data were collected and analyzed. See Fig. 1 for the full pipeline. The analytical results provide detailed information about individual records and are therefore provided along with the raw data. The data are collected with SGNET [14], a distributed honeypot system. It automatically learns the behavior of network protocols with techniques from bioinformatics and can handle attacks of any kind. Each attack is classified with the *epsilon-gamma-pi* model of Crandall et al. [2]. The information stored has as a result a high level of granularity and provides a good basis for further processing with the Triage Framework [19].

In this framework, each data item is assigned a category and depending on this only certain – not all – attributes used for further analysis. The term feature is used in the remainder instead of attribute. For each datum, a so-called feature vector is created, which summarizes the features of the datum. This process is called attack feature selection. A similarity matrix is then created from the collected data for each feature. A similarity matrix or dissimilarity matrix is a set of distances between all pairs of data of a data set [8]. The similarity is derived directly from the dissimilarity and vice versa.

The similarity metric depends on the characteristics of the feature [19]. It creates many large similarity matrices, the number is determined by the number of features, the size of the individual matrices by the size of the data set. In the phase graph-based clustering, these matrices are clustered and thus combined homogeneous groups. The clustering algorithm used is not specified by Triage. In VIS-SENSE, an algorithm is used, that recognizes the dominant sets [16] within an adjacency matrix. Reasons for this are, inter alia, a variable number of clusters and parallelization issues [19].

In the third and final step, an MCDA - Multi-Criteria Decision Analysis - is performed. The values of the similarity matrices of all features are combined with aggregation functions and there a similarity matrix is created, which is determined by all features. It is also clustered. This clustering is called MDC - Multi Dimensional Clustering. Because of the many dimensions (features) an aggregation of all values is not trivial. Dependencies of individual features with each other must be recognized and taken into account in the aggregation function. OWA - Order Weighted Averaging functions and Fuzzy Integrals like the Choquet integral, solve these problems (see. [Thod]). Due to the high complexity of this method, the results are not transparent. For the viewer of the MCDA matrix it is not clear how the values were calculated or composed. This is exactly of interest. The goal of MCDA is to assign different malware derivatives to a superclass. The superclasses are investigated and similarities and features with large deviations found. The key factors as sought that are responsible for the outcome in MCDA. With this knowledge, distinct malware groups can be graded better.

[1] http://www.vis-sense.eu/.
[2] http://www.wombat-project.eu/Projekt.

For example, suppose there is a cluster in the MCDA whose elements all have identical values of the individual features, apart from the MD5 hash, then it is assumed that the binary was only slightly changed. With this knowledge a heuristic can be created, with which other derivatives of this malware class can be found.

For this analysis, all matrices are visualized in parallel and the calculated clusters are highlighted. Marks in a matrix are transmitted to all other matrices. When a cluster of feature X is marked, it can be seen how the elements are distributed in the other matrices. The data set is very large, it covers up to 10,000 data. The matrices thus contain up to $10{,}000 \times 10{,}000$ entries. For the visualization of such large amounts of data the limits of traditional PCs are quickly reached. If the memory should still be sufficient, the resolution of the monitor is not nearly high enough. For this reason, the data must be reduced. For this we use sampling, since dealing with very large amounts of data is possible while not adversely affecting the validity of the visualization [3]. An increase in the amount of data to up to 100,000 elements is planned. Assuming that an element of a similarity matrix is one byte in size, and 20 of these matrices exist, the amount of data is about 186 GB. From this set submatrices are created using a sample of the node set. The elements required are distributed randomly in principle, with the result that each element is read one by one. Important here is a scaling data management. If the access time to the individual elements increases with the number of stored items, then the use of the program with a real-time request is not feasible. We examine in this context the performance of two different database concepts [1] (Fig. 2).

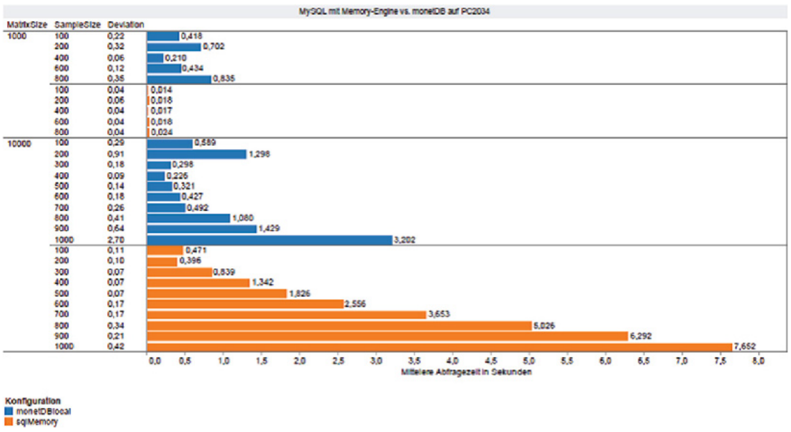

Fig. 2. Comparison of two configurations: (1) MonetDB standard configuration, Data on standard hard disk, 4 GB RAM, Intel Core i3-2120 CPU at 3,3 GHz mit 2 Kernels/4 Threads vs. (2) MySQL with Memoryengine, data in RAM, 24 GB RAM, Intel Xeon E5645 at 2,4 GHz with 6 Kernels/12 Threads.

3 Results

We develop a concept that allows the visualization of large similarity matrices. A similarity matrix is the representation of a graph as an adjacency matrix. There are various visualization types for the display of graphs. The node-link diagram is often used and provides a good overview [5]. For larger graphs, node-link diagrams quickly reach their limits. Especially in densely populated graphs, it is difficult to create a suitable layout. An alternative to node-link diagrams is the matrix representation. Here the edge weight is mapped usually on color or transparency. This representation has its strengths especially in large, densely populated matrices. An ordering of the rows/columns plays an essential role. Is the arrangement unfortunately selected, no information can be obtained from the visualization [5]. We investigated a variety of techniques to visualize large graphs. Of interest is on the one hand filtering by interestingness, on the other hand, a reduction by sampling. Due to the large amount of data and number of dimensions, the data for visualization must be greatly reduced. The goal is the two techniques combine together. For the selection of displayed nodes in the matrices, a sample of the complete set of nodes is needed. After starting the VIS-SENSE environment the sample should be evenly distributed. By interacting with the user the distribution changes and adapts to the User Interests.

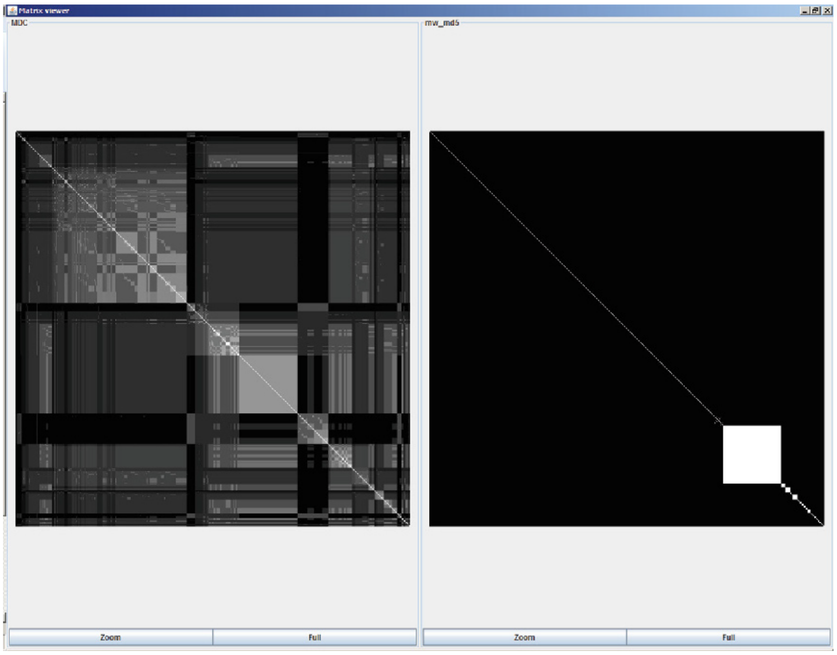

Fig. 3. Initial sampling 1

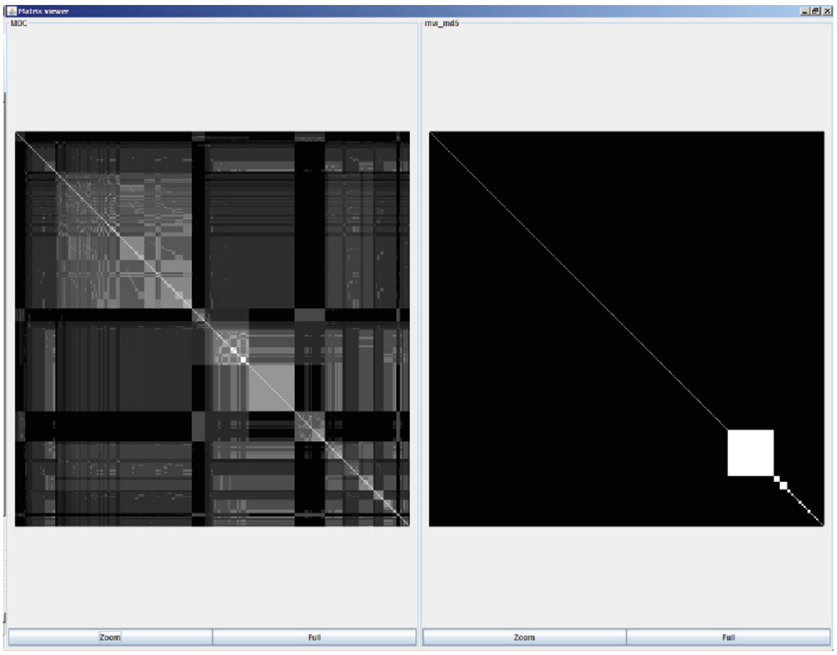

Fig. 4. Initial sampling 2

For visualization properties we use a SGNET record including approximately 14,000 nodes, each with 22 features. For each feature, there is a similarity matrix and cluster information. The information is in text files that have a total size of about 12 GB. We implemented our system in MySQL. For our evaluation, we examined technical and visual aspects. The technical parts deal with the access times to large similarity matrices in different database configurations. The visual aspect describes the effects of our dynamic subsampling on the visualization in relation to the conservation of the context at the simultaneous focus on specific areas.

4 Visual Impressions

With a small matrix (1,000 $\times$ 1,000) configuration 2 is a clear advantage. The processing time increases only very slightly with the sample size, and is even with a relatively large sample less than 0.1 s. MonetDB is in absolute terms with configuration 1 is also very fast, but in a direct comparison can not keep up with the memory engine of MySQL. Much more interesting are matrices with sizes of 10,000 $\times$ 10,000 entries. Here MonetDB shows its strengths. The access times are only slightly different for small sample sizes compared to the 1000 $\times$ 1000 matrix. Creating a 1000 $\times$ 1000 entries large sample is with 3 s in a noticeable

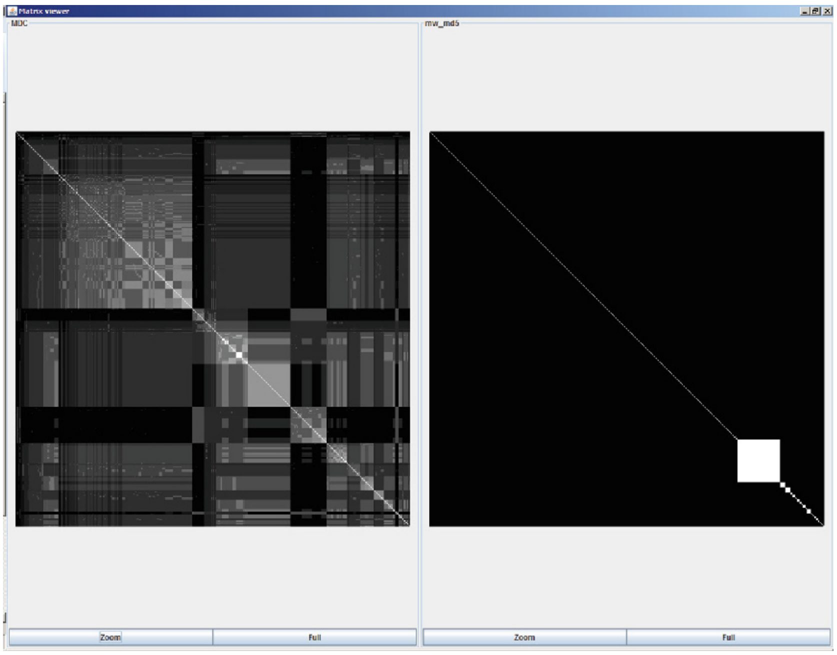

Fig. 5. Program start

area, but is compared to configuration 2 very good. Full results can be found in [17].

We have built our implementation on an existing project. This system has defined the way of visualizing matrices. To deal with the size of the data, it has used a static sample of the data. Thus, always the same subset of the data was visualized and the visualization also remained the same. After successful implementation of our dynamic sampling was necessary to clarify the extent to which the visual impression or the user's knowledge are affected.

4.1 Initial Sampling

In a static sample as a data source, a segment of the data is displayed and this does not change. The most striking characteristics can be seen in it. In our case, these are the largest clusters of a matrix. If only these are of interest, such a sample fulfilled his request. The visualization serves mainly the analysis of the MCDA matrix whose clusters represent relationships of all features. Here the size of a cluster represents the frequency of occurrence in the data set and is thus an indicator of the importance of this relationship. It is not said that smaller clusters represent an unimportant or useless connection, though. A statement about the importance is to be taken by an analyst. For a sample, it is possible that smaller clusters are not represented by any node in the sample and are therefore not

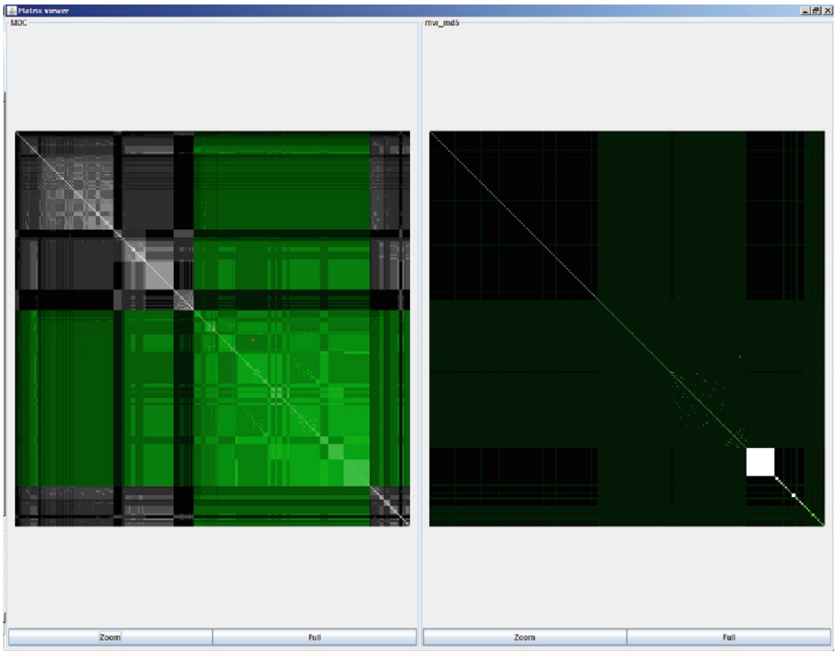

Fig. 6. First subsampling

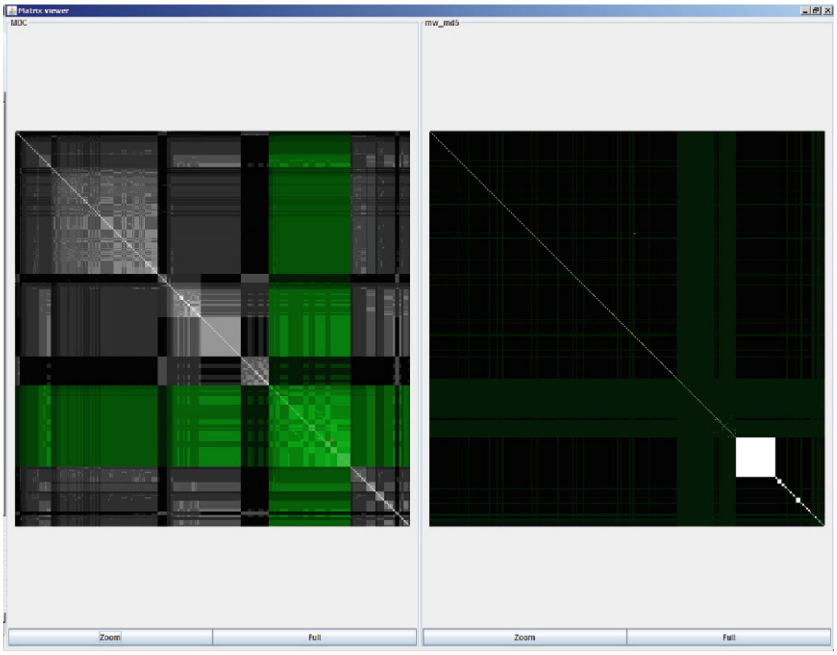

Fig. 7. Second subsampling

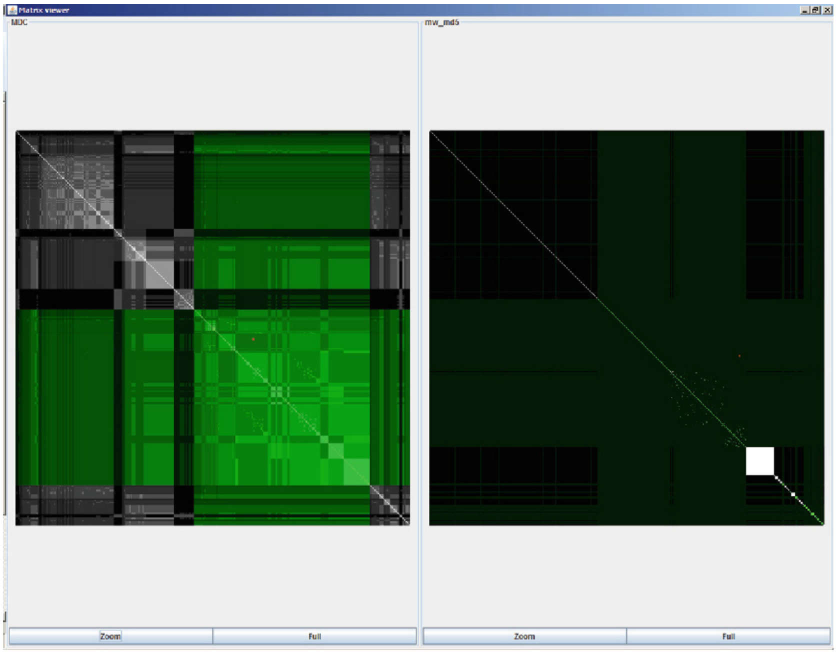

Fig. 8. n^{th} subsampling

visualized. With our dynamic sampling at each startup another sample is created for the analyst (see Figs. 3, 4 and 5). This does not guarantee that each cluster appears, but with each analytical process the probability increases that each cluster was examined at least once.

4.2 Subsampling

As with the initial sampling our implementation shows its strengths in the study of small clusters. If a cluster represented by relatively few nodes the statement about the distribution of these nodes in the clusters of the other feature is not meaningful. With our program nodes are subsampled from the selected cluster and there is a greater focus on this area realized (see Figs. 5, 6, 7 and 8).

Thus, a representative quantity is provided to detect the distribution, or to understand the emergence of this cluster in the MCDA. In the figures it is clear that for the feature mw_md5 after multiple zooming a cluster, noise can be seen, that after starting the program was impossible or very difficult to see. This shows that by zooming not only existing characteristics are enlarged, but also completely new properties can be recognized. The row and column array of matrices that can be seen in Figs. 5, 6, 7 and 8 are not optimal. For a final assessment of the new findings by subsampling, the rows and columns would have to be arranged accordingly. There is no upper limit on the number of nodes that must be present for each cluster in the sample. During the analysis, it can happen that only nodes of one cluster are displayed. The context is in this case completely lost and the program must be restarted. That would be an exemplary approach for improvements and enhancements. Regardless of this, the benefits of dynamic subsampling are clear.

5 Conclusions

Three issues appear to be important: storage, loading, and visualization: (1) The data must be stored for further processing as appropriate. The volume presents in today's resources are not a problem. The time required to insert in a database can be realized with appropriate methods of data base systems in a reasonable time. (2) Access to the data is the biggest problem. By sampling data to non-contiguous positions are read from memory. This is a process that challenges the weaknesses of a hard disk. We were able to show that these weaknesses can be compensated with MonetDB. (3) We have extended the visualization with our technique and allow focusing parts of the data. We discussed many functions that bring visibility and usability.

References

1. Abadi, D.J., Madden, S.R., Hachem, N.: Column-stores vs. row-stores: how different are they really? In: ACM SIGMOD International Conference on Management of Data, pp. 967–980 (2008)
2. Crandall, J.R., Wu, S.F., Chong, F.T.: Experiences using minos as a tool for capturing and analyzing novel worms for unknown vulnerabilities. In: Julisch, K., Kruegel, C. (eds.) DIMVA 2005. LNCS, vol. 3548, pp. 32–50. Springer, Heidelberg (2005)
3. Dix, A., Ellis, G.: By chance enhancing interaction with large data sets through statistical sampling. In: Advanced Visual Interfaces, pp. 167–176 (2002)
4. Eicke, T.N., Jung, Y., Kuijper, A.: Stable dynamic webshadows in the X3DOM framework. Expert Syst. Appl. **42**(7), 3585–3609 (2015)
5. Elmqvist, N., Do, T.N., Goodell, H., Henry, N., Fekete, J.: ZAME: interactive large-scale graph visualization. IEEE PacificVIS 2008, pp. 215–222, March 2008
6. Engelke, T., Becker, M., Wuest, H., Keil, J., Kuijper, A.: MobileAR browser - a generic architecture for rapid AR-multi-level development. Expert Syst. Appl. **40**(7), 2704–2714 (2013)
7. van Ham, F., Perer, A.: Search, show context, expand on demand: supporting large graph exploration with degree-of-interest. IEEE TVCG **15**(6), 953–960 (2009)
8. Han, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco (2005)
9. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. Eur. J. Oper. Res. **126**(1), 106–130 (2000)
10. Kuijper, A., Florack, L.: The relevance of non-generic events in scale space models. Int. J. Comput. Vis. **57**(1), 67–84 (2004)
11. Kuijper, A., Olsen, O.F.: Transitions of the pre-symmetry set. In: 17th International Conference on Pattern Recognition, ICPR 2004, pp. 190–193 (2004)

12. von Landesberger, T., Bremm, S., Kirschner, M., Wesarg, S., Kuijper, A.: Visual analytics for model-based medical image segmentation: opportunities and challenges. Expert Syst. Appl. **40**(12), 4934–4943 (2013)
13. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J.J., Fekete, J., Fellner, D.W.: Visual analysis of large graphs: state-of-the-art and future research challenges. Comput. Graph. Forum **30**(6), 1719–1749 (2011)
14. Leita, C., Dacier, M., Wicherski, G.: SGNET: a distributed infrastructure to handle zero-day exploits. Technical report, EURECOM+2164, Eurecom, February 2007
15. Nazemi, K., Stab, C., Kuijper, A.: A reference model for adaptive visualization systems. In: Jacko, J.A. (ed.) Human-Computer Interaction, Part I, HCII 2011. LNCS, vol. 6761, pp. 480–489. Springer, Heidelberg (2011)
16. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. IEEE Trans. Pattern Anal. Mach. Intell. **29**(1), 167–172 (2007)
17. Roskosch, P.: Dynamisches sampling zur visuellen exploration von groen dichtbesetzten matrizen. Technical report, TU Darmstadt (2013)
18. Stein, C., Limper, M., Kuijper, A.: Spatial data structures to accelerate the visibility determination for large model visualization on the web. In: Web3D 2014, pp. 53–61 (2014)
19. Thonnard, O.: D22/d5.2 root causes analysis: experimental report. Technical report, FP7-ICT-216026-WOMBAT, WP5 - Threats Intelligence, FP7-ICT (2011)
20. Zhou, X., Kuijper, A., Veldhuis, R.N.J., Busch, C.: Quantifying privacy and security of biometric fuzzy commitment. In: IEEE International Joint Conference on Biometrics, pp. 1–8 (2011)
21. Zhou, X., Wolthusen, S.D., Busch, C., Kuijper, A.: A security analysis of biometric template protection schemes. In: Kamel, M., Campilho, A. (eds.) ICIAR 2009. LNCS, vol. 5627, pp. 429–438. Springer, Heidelberg (2009)