

Towards Texture-Based Visualization of Multivariate Dataset

Raja Majid Mehmood, Hyo Jong Lee

Division of Computer Science and Engineering
Center for Advanced Image and Information Technology
Chonbuk National University, Jeonju, South Korea

E-mail: rmeex07@jbnu.ac.kr, hlee@chonbuk.ac.kr

Abstract

Visualization is a science which makes the invisible to visible through the techniques of experimental visualization and computer-aided visualization. This paper presents the practical aspects of visualization of multivariate dataset. In this paper, we will briefly discuss a previous research work and introduce a new visualization technique which will help us to design and develop a visualization tool for experimental visualization of multivariate dataset. Our newly developed visualization tool can be used in various domains. In this paper, we have chosen a software industry as an application domain and we used the multivariate dataset of software components computed by VizzMaintenance. VizzMaintenance is software analysis tool which give us multiple software metrics of open source Java based programs. Main objective of this research is to develop a new visualization tool for large multivariate dataset which will be more efficient and easy to perceive by viewer. Perception is very important for our research work and we have decided to test the perception level of our proposed visualization approach by researchers of our research lab.

Keywords: Visualization, Software Metrics, Glyphs, Texture-based Visualization, Multivariate Dataset.

1. Introduction

Visualization is a technique that presents data in a visual form to facilitate rapid, effective, and meaningful analysis and interpretation. Example application domains include scientific simulations, land and satellite weather information, geographic information systems, and molecular biology [1]. In the following sections, we will describe some useful material which is related to our research work.

1.1 Software visualization

Software visualization tools are used by software designers to raise the level of abstraction and reduce the amount of information to the one needed. Mostly these are stand-alone programs, which force the user to switch between different windows and contexts. The development of software visualization frameworks is a significant step to bring visualization tools in the forward engineering process [2-4].

1.2 Software Metrics

Software metric is a measure of some property of a piece of software or its specifications. Since quantitative methods have been proved so powerful in other sciences, computer science practitioners and theoreticians have worked hard to bring similar approaches to software development. Tom DeMarco stated, “You can’t control what you can’t measure” [5]. There are many examples of software metrics: lines of code, cyclomatic complexity, function point analysis, bugs per line of code, code coverage, number of lines of customer requirements, number of classes and interfaces, Robert Cecil Martin’s software metrics for cohesion or coupling [5].

1.3 VizzMaintenance

VizzMaintenance is an Eclipse plug-in supporting Eclipse 3.5 or higher. It brings detailed information about the maintainability of our classes in our software system at a fingertip. It uses texture-based analysis to calculate 17 in literature well-discussed software quality metrics. It then combines these values in a software quality model. It supports our decisions which classes should be refactored first to

improve their maintainability. Information can be exported to the clipboard for further analysis in other tools [5].

1.4 Multivariate Dataset

In this paper, a multivariate data object, d , is defined as $d = \{d_1, d_2, \dots, d_N\}$, where d_i is a scalar and N is the number of attributes ($N \geq 2$). A multivariate dataset is then one comprising M data objects (instances of d), where $M \geq 2$. The values of M and N vary widely depending on the application area. An illustration of a multivariate dataset containing shows few data objects with series of attributes in Table 1.

Table 1. Multivariate Dataset of software components

Name	Maintainability	CMM_1_0_CBO	CMM_1_0_CYC_Classes	CMM_1_0_DAC	CMM_1_0_DIT	CMM_1_0_ILCOM	CMM_1_0_WMC
UnitFactory	0.068319	3	14	3	0	2	15
LErrorBarPixel	0.136638	2	136	2	0	1	5
AAxisAction	0.204957	4	136	4	1	1	2
StaticCollectorChart	0.136638	7	1	7	0	1	2
Trace2DDebugger	0	8	1	8	2	1	67
.
.
Chart2DActionSetName	0.068319	2	1	2	1	0	2

The above dataset is computed through VizzMaintenance [5] and it belongs to open-source project JChart2D [7]. Multivariate dataset is represented in tabular form, where a data object (or simply an item) corresponds to a row and each attribute corresponds to a column. The objects are software components where each object is described by a number of software metrics like (CMM_1_0_CBO, CMM_1_0_CYC_Classes, CMM_1_0_DAC etc.). These software metrics are used to describe the corresponding

property of the each software component.

The structure of this paper will describe as follows. Section-2 is a literature review of visualization approaches. Section-3 discusses proposed visualization technique. Section-4 describes the results and discussion. Section-5 concludes our research work.

2. Related Work

We will discuss existing work in short that is directly related to our proposed visualization technique, and we will highlight the areas where we may possibly able to offer improvements over existing methods.

We have found a couple of visualization tools and techniques during our research work [7-18]. But we have selected only one visualization technique which is more relevant with our proposed visualization of multivariate dataset. The existing Kiviat diagram approach presented by Pinzger et al. [7] at the ACM SoftVis Conference 2005. The following Fig.1 shows the Kiviat diagrams and it is also known as star plots, it can be used to visualize multivariate dataset, like software metrics of source code.

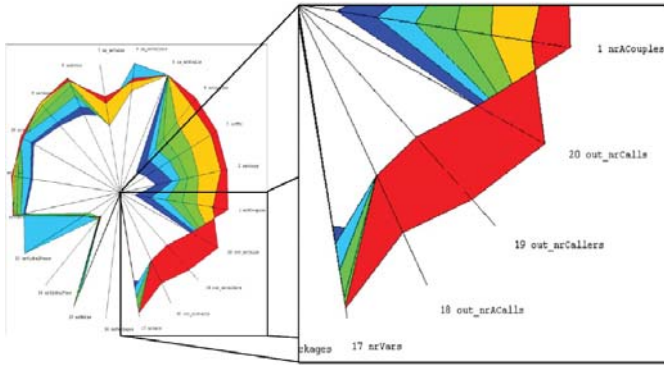


Figure 1. Screenshot improved Kiviat diagram [7]

Pinzger et al. Improved the standard Kiviat Diagram approach [8] in order to facilitate the visualization of software metrics. Each value of a metric is measured across N releases of source code and plotted in the diagram.

After intensive study of above visualization, we found two major issues in understanding of this approach; **1**: there is colors overlapping problem which hides the certain metric values for example it is hard to perceive the visualization in Fig.1 due to color overlapping (like, for the metric numbers 18, 19, and 20), **2**: it is difficult to perceive the metric values especially when viewer is looking for metric outliers.

In the following Section-3, we will introduce our newly developed approach which is completely different than previous visualization approaches [7-11].

3. Proposed Visualization Technique

In this research paper, we have studied existing software visualization techniques to display a multivariate dataset. These visualization tools have used a different type of visualization features like, texture, animation, and flow visualization in 2D or 3D environment. Mainly, we have chosen the latest software visualization tools like [7-11] and [1] that have visualized the multivariate dataset. After study of

existing visualization tools we have found some major deficiencies. Here, we will describe our proposed visualization approach for the visualization of multivariate dataset i.e. Static or Texture Base Visualization. Fig. 2, shows the texture-based glyph with a texture pattern, which represents the static display of attribute. We can see the graphical properties of glyph object in Fig. 2; a) Background color, b) Texture pattern, c) Placement, d) Border.

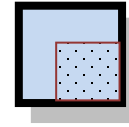


Figure 2. Proposed Texture-based glyph

Background Color—This property identifies an attribute itself, and is used to differentiate the attributes in the data object. In the Fig. 3, we can see that the background color of all glyph objects represents the attributes of the data object. We also see few glyphs in Fig. 3 with black background (without texture). These glyph objects show that there are no more attributes in the data object, and they are presented as a blank glyph objects.



Figure 3. Data Object

Texture—This property is used to represent the value of an attribute. Perhaps, high density texture represents a high attribute value and a low density texture represents a low attribute value. Texture is calculated through scaling of actual attribute value. Fig. 3 shows that many glyphs have the property of texture, where each glyph texture shows the scaled value of each attribute in that data object. The following Fig. 4 present different textures and placement for the same attribute in different data objects. Fig. 4 (a) shows the actual attribute value is close to highest attribute value. In Fig. 4 (b), the actual attribute value is close to the lowest attribute value.

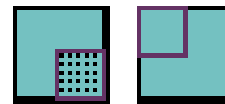


Figure 4. (a): Maximum Texture (b): Minimum Texture

Texture Value? We will compute the scaled value of a texture for glyph objects with the help of the following formulas:

$$I = (M\alpha - M_1) / N, A = (A\alpha - M_1) / I \quad (1)$$

Equation (1) will be used for positive values.

$$I = (|M_1| - |M\alpha|) / N, A = (|M_1| - |A\alpha|) / I \quad (2)$$

Equation (2) will be used for negative value. In both cases, for equation (1) and (2) “A” must be $0 \leq A \leq 100$. Equations (1, 2) have some variables like;

- $M\alpha$, Max Attribute Value in dataset
- M_1 , Min Attribute Value in dataset

outlier value of this attribute. In this case, all of *Red Highlighted Attribute* is LOC (software quality metrics), here analyst can easily decide the maintainability level of these classes (software components).



Figure 9. Changing the Background Color of Attribute

The testers group also agreed that this texture-based visualization is much better for displaying more glyphs in a *2D-Grid*. The texture-based visualization is very fast in rendering the display of glyphs in a *2D-Grid* with low utilization of resources, and this is suitable for slow machines.

5. Conclusion

In this paper, we have presented our newly developed visualization tool to represent the multivariate dataset in *2D-Grid*. This visualization technique is useful for different perspectives and it can be useful in other application domain as well. As a main goal of this work, our focus is on practical evaluation of our visualization approach. Practically, loading process time of dataset in visualization approach is normal; all glyphs objects are loaded at once in start with static texture. We concluded that the perception of texture-based visualization in *2D Grid* is much better than *Kiviat Diagram* approach [8] and our approach is more convenient to display or accommodate large dataset in *2D-Grid* by using of sliding technique.

References

- [1] A. P. Sawant and C. G. Healey, "Visualizing multidimensional query results using animation," in *Visualization and Data Analysis 2008*, vol. 6809, K. Borner, M. T. Grohn, J. Park, and J. C. Roberts, Eds., ed, 2008.
- [2] T. DeMarco, "Controlling Software Projects: Management, Measurement and Estimation., " vol. ISBN 0-13-171711-1, 1982.
- [3] S. Diehl, "Future perspectives," in *Software Visualization*, vol. 2269, S. Diehl, Ed., ed, 2002, pp. 347-353.
- [4] D. Hubbard, "The IT Measurement Inversion CIO Magazine," 1999.
- [5] Wikipedia. (2008). Available: en.wikipedia.org/wiki/Software_metric
- [6] S. Kim, R. Maciejewski, A. Malik, Y. Jang, D. Ebert, and T. Isenberg, "Bristle Maps: A Multivariate Abstraction Technique for Geovisualization," 2013.
- [7] M. Pinzger, H. Gall, M. Fischer, and M. Lanza, "Visualizing Multiple Evolution Metrics," in *Proceedings of the 2005 ACM Symposium on Software Visualization (SoftVis '05)*, St. Louis, Missouri, 2005, pp. 354-362.
- [8] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey, "Graphical Methods for Data Analysis (vol 17, pg 180, 1983)," *Journal of Sleep Research*, vol. 21, pp. 484-484, Aug 2012.
- [9] C. G. Healey and J. T. Enns, "Building perceptual textures to visualize multidimensional datasets," *Proceedings Visualization '98 (Cat. No.98CB36276)*, pp. 111-18, 525, 01 01 1998.
- [10] C. G. Healey and J. T. Enns, "Large datasets at a glance: Combining textures and colors in scientific visualization," *Ieee Transactions on Visualization and Computer Graphics*, vol. 5, pp. 145-167, Apr-Jun 1999.
- [11] D. E. Huber and C. G. Healey, *Visualizing data with motion*, 2005.
- [12] G. Dzemyda, O. Kurasova, and J. Žilinskas, "Strategies for Multidimensional Data Visualization," in *Multidimensional Data Visualization*, ed: Springer, 2013, pp. 5-40.
- [13] J. M. Rzeszutarski and A. Kittur, "TouchViz:(multi) touching multivariate data," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 1779-1784.
- [14] K. Reda, A. Febretti, A. Knoll, J. Aurisano, J. Leigh, A. Johnson, M. Papka, and M. Hereld, "Visualizing Large, Heterogeneous Data in Hybrid Reality Display Environments," 2013.
- [15] G. Dzemyda, O. Kurasova, and J. Žilinskas, "Multidimensional Data and the Concept of Visualization," in *Multidimensional Data Visualization*, ed: Springer, 2013, pp. 1-4.
- [16] J. A. Guerra-Gómez, A. Buck-Coleman, M. L. Pack, C. Plaisant, and B. Shneiderman, "TreeVersity: Interactive Visualizations for Comparing Hierarchical Datasets," *Center for Advanced Transportation Technology Laboratory*, vol. 301, p. 0722, 2013.
- [17] C. Lee, W. Zhuo, J. Choo, D. H. P. Chau, and H. Park, "Fast interactive visualization for multivariate data exploration," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 1773-1778.
- [18] H. Janicke, M. Bottinger, and G. Scheuermann, "Brushing of attribute clouds for the visualization of multivariate data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, pp. 1459-1466, 2008.
- [19] R. M. Mehmood, "Dynamic and Static Approaches for Glyph-Based Visualization of Software Metrics," Master, School of Mathematics and Systems Engineering, Växjö University, Sweden, 2008.