Measuring the Complexity of Computer Security Visualization Designs

X. Suo, Y. Zhu, and G. Scott Owen

Abstract We present a novel method to measure the complexity of computer security visualization designs. The complexity is measured in terms of visual integration, number of separable dimensions for each visual unit, the complexity of interpreting the visual attributes, and the efficiency of visual search. Visualization developers can use this method to quickly evaluate multiple design choices in the early stage of their design before any user study can be conducted. To demonstrate this method, we have conducted complexity analysis on two open source security visualization tools – TNV and RUMINT.

1 Introduction

Evaluation is an integral part of the information visualization design process. For example, visualization developers often need to make a design choice among multiple design options. In a group setting, different developers may come up with different designs for the same set of data. In these situations, developers need methods to assess and compare different designs. The most common evaluation method is user study, which can be used to measure task completion time, the number of errors, and user satisfaction. However, user studies are often time consuming, difficult to control and can only be conducted after a prototype has been developed. In addition, the results of these studies often do not explain what causes usability problems.

In this paper, we propose an alternative evaluation method – complexity analysis. Following this method, developers systematically evaluate a set of factors that influence the efficiency of processing visual information. Here the complexity is measured in terms of visual integration, number of separable dimensions for each

X. Suo, Y. Zhu, and G.S. Owen
Department of Computer Science, Georgia State University, Atlanta, Georgia, USA, e-mail: yzhu@cs.gsu.edu

visual unit, the complexity of interpreting the visual attributes, and the efficiency of visual search. Based on well established psychological theories on human cognition, this method allows visualization developers to quickly compare the complexity of multiple visualization designs and is particularly useful during the early design stage before any user study can be conducted.

Comparing with other heuristic visualization evaluation methods that deal with a broader range of usability issues, the proposed method focuses on complexity. However, the complexity of a visualization design may greatly influence common usability issues, such as task completion time, error rate, learnability, and user satisfaction. Therefore it would be beneficial to correlate the proposed complexity analysis with other heuristic methods as well as user studies. For example, the results of the complexity analysis can help generate hypotheses for user studies to verify.

The paper is organized as follows. In Sect. 2, we discuss related work. In Sect. 3, we present our method, using open source security visualization tool TNV as an example to demonstrate the steps of complexity analysis. As an additional case study, we apply our method to another open source security visualization tool RUMINT. In Sect. 4, we discuss our future work. Section 5 is the conclusion.

2 Related Work

Assessing the effectiveness of the visualization design is a complicated issue. There is no doubt that formal user study is and should be the primary form of evaluation. However, as Tory and Moller (2005) pointed out, formal user studies have their limitations. Controlled user studies require lots of time and resources and are hard to manage. As a result, Tory and Moller have argued that expert review can be a valuable alternative to formal user studies.

Expert review is a heuristic evaluation method, and there have been a number of heuristics for evaluating information. For example, Tufte (2001) has proposed some heuristics for evaluating the visual display of quantitative information, such as the data-ink ratio. Shneiderman's "Visual Information Seeking Mantra" (Shneiderman, 1996) is sometimes used as a heuristic for evaluation. Amar and Stasko (2005) have proposed a knowledge task-based framework for the design and evaluation of information visualizations. This is a high level framework that covers a broad range of issues.

Overall, the current heuristics for evaluating information visualization are far from comprehensive and not well organized. To address this issue, Zuk et al. (2006) have performed a meta-analysis of existing heuristics with an aim to develop a set of appropriate heuristics for the evaluation of information visualization. As a first step, they attempt to build a hierarchical structure to organize the many heuristic rules of evaluation. Their proposed information visualization heuristic structure includes the following groups: perceptual, cognition, usability, etc. Drawing from three sets of heuristics (including Shneiderman's and Amar and Stasko's heuristics), their analysis suggests that different heuristics are needed to evaluate different aspect of a visualization design.

Our complexity analysis is a type of heuristic evaluation of information visualization. It can be seen as a specific type of expert review with a focus on evaluating the complexity. The specific guidelines and procedures presented in this paper is a new addition to the heuristics of information visualization evaluation. Our work is compatible with the analysis by Zuk et al. (2006). The analysis of visual integration, the number of separable dimensions for each visual unit, and the complexity of interpreting the visual attributes belong to the cognition group in their heuristics hierarchy, while the analysis of visual search efficiency belongs to the perception group.

The uniqueness of our proposed method is its focus on the complexity of visualization design. The complexity of visualization design is rarely mentioned in previous heuristic evaluation methods. Only a few visualization researchers have touched on the issues of complexity in visual display, but none of them have dealt with it in a systematic way. For example, Bertin (1983) and Trafton et al. (2000) have used the number of dimensions as a measure for the complexity of visual displays, and considered visualizations with more than three variables to be complex. Brath (1997) has proposed a heuristic method to measure the effectiveness of the mapping from the data dimension to the visual dimension by classifying the visual mappings into one of the four categories. In terms of analyzing complexity, our evaluation method is more systematic than these previous methods and considers many more factors.

The proposed complexity analysis is based on a number of psychological theories, including Guided Visual Search theory (Wolfe and Horowitz, 2004), Gestalt theory (Wertheimer and King, 2004), and cognitive load theory (Clark et al., 2006). According to the cognitive load theory, there are three types of cognitive load: intrinsic cognitive load, extraneous cognitive load, and germane cognitive load. The mental effort to comprehend a data visualization is part of the extraneous cognitive load, which is a major factor that influences the task performance. The proposed visualization complexity analysis is an attempt to measure the extraneous cognitive load of visualization comprehension.

It should be noted that the proposed complexity analysis is not a direct measure of a visualization design's usability or task performance. Although we hypothesize that the complexity of a visualization design is likely to have a significant impact on task performance and usability, the exact relationship between complexity and usability or task performance is not well understood and will be the focus of our future work. It also means that our method should be combined with user studies and other heuristic methods that directly measure usability and task performance.

3 Technical Approach

The processing of a data visualization depends on a host of psychological processes, including information read-off, integration, and inference (Trafton et al., 2000). The main goal of the proposed complexity analysis is to systematically evaluate the

Table 1 Overview of the complexity analysis

Complexity analysis steps	Step 1	Step 2	Step 3	Step 4
Purpose	Divide a data visualization into hierarchical layers	Analyzes the efficiency of visual integration	Analyze the efficiency of reading visual units	Analyze the efficiency of visual search
Theoretical basis	(Trafton et al., 2000; Zhou and Feiner, 1997)	(Wertheimer and King, 2004; Trafton et al., 2000)	(Clark et al., 2006; Garner, 1974)	(Wolfe and Horowitz, 2004)
Outcome Data structure	N/A	Visual integration complexity tree	Visual mapping complexity tree	N/A
Quantitative metrics	N/A	Maximum number of visual integrations	• Number of different types of visual units • Number of separable dimensions per visual unit • Complexity scores for visual units	Target-distracter difference score and visual search complexity scores for color, motion, size, and orientation

major factors that influence the efficiency of information read-off and integration. Table 1 gives an overview of the complexity analysis process.

The complexity analysis is carried out in the following steps:

(1) We divide the visualization into five layers: workspace, visual frame, visual pattern, visual units, and visual attributes.
(2) Next, we analyze the efficiency of integrating visual elements. We build a tree structure that depicts how visual frames are organized in each workspace, and how visual patterns are organized in each visual frame. We call this a visual integration complexity tree (Fig. 2) because it shows how a reader might mentally integrate visual frames and visual patterns. The number of nodes on this tree is the maximum number of visual integrations that a reader might perform.
(3) We then analyze the efficiency of interpreting visual units. For each type of visual unit, we identify the visual attributes that are used to encode data parameters. Each of the encoded visual attributes is called a dimension. For each encoded visual attribute, we estimate the complexity of mapping the visual attribute to its corresponding data parameter. The outcome is a visual mapping complexity tree (Fig. 6).
(4) Finally we analyze the efficiency of visual search. We evaluate the target-distracter difference for four attributes (color, motion, size, and orientation) that are important for efficient visual search.

In the following sections, we will discuss the complexity analysis in more detail, using TNV as an example to demonstrate our ideas. TNV (version 0.3.7) is a computer network traffic visualization tool developed by Goodall et al. (2005). Later we will apply our method to RUMINT, another open source network security visualization tool developed by Conti (2005, 2007). The dataset we used to evaluate both tools is called cigi.pcap, available at http://wiki.ethereal.com/CIGI.

3.1 Hierarchical Analysis of Data Visualization

In order to systematically analyze a visualization design, we divide data visualization into five hierarchical layers: workspace, visual frame, visual patterns, visual units, and visual attributes. Each layer is a component of the previous layer. That is, a workspace is one or more visual frames that are designed for a specific purpose. A visual frame is a window within a workspace and contains multiple visual patterns. A visual pattern is a set of visual units that are readily perceived as a group; and they are identified based on four Gestalt laws: proximity, good continuation, similarity, and common fate. Some examples of visual units include point, line, 2D shape (glyph), 3D object, text, and image. Each visual unit is defined by seven visual attributes (Bertin, 1983): position, size, shape, value, color, orientation, and texture.

The hierarchical analysis of data visualization is inspired in part by Zhou and Feiner (1997), who used a hierarchical framework to construct visualizations. Our hierarchical analysis is also influenced by Trafton et al. (2000), who propose a three layer hierarchical framework – information read-off, integration, and inference – to describe the cognitive process of graph comprehension. In our hierarchical framework, visual units and visual attributes are related to the information read-off layer, while visual patterns and visual frames are related to the visual integration layer.

In the case of TNV, the workspace is the entire visual interface, while the four visual frames are marked by red boxes (see Fig. 1).

3.2 Visual Integration

Larkin and Simon (1987) point out that a main advantage of visualization is that it helps group together information that is used together, thus avoiding large amounts of search. In complex problem solving, the visual units need to be integrated (Trafton et al., 2000), which adds to the extraneous cognitive load.

In this study, the cognitive load of visual integration is estimated by building a visual integration complexity tree (Fig. 2). For each visual frame, we identify the visual patterns in that frame based on four Gestalt laws: proximity, good continuation, similarity, and common fate. The number of nodes on the visual integration complexity represents the upper bound of visual integration a reader might perform. The visual integration complexity for TNV is shown in Fig. 3.

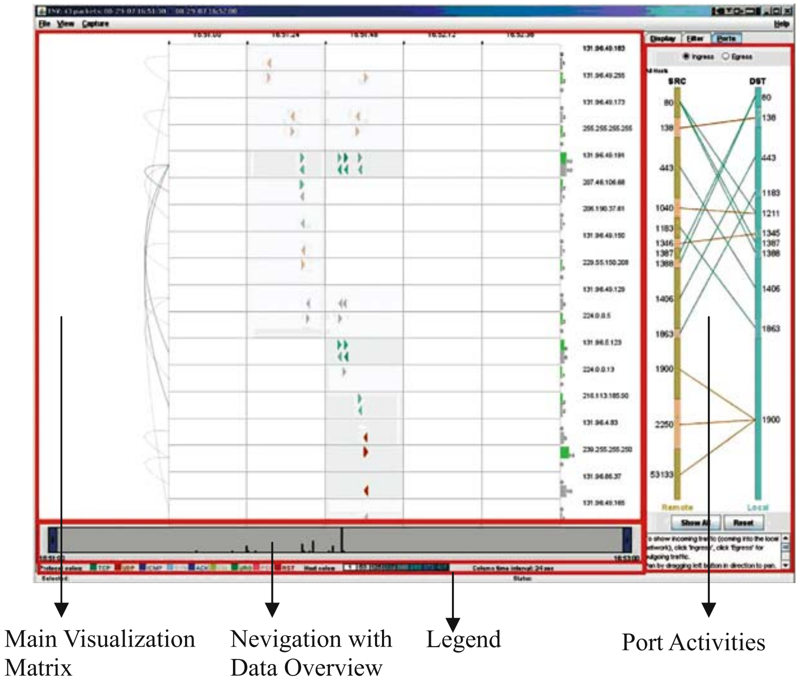


Fig. 1 Four different frames of TNV

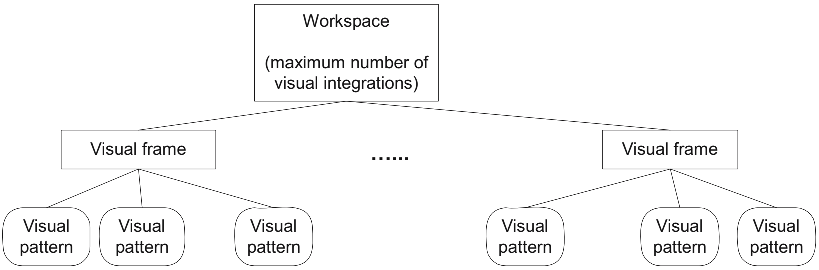


Fig. 2 Visual integration complexity tree

3.3 Separable Dimensions for Visual Units

In a visualization design, different data parameters are mapped to different visual attributes of different visual units. Each encoded visual attribute is called a dimension. Readers need to identify and remember these visual mappings, thus adding to the extraneous cognitive load. Psychological studies have showed that human eyes can see only three variables at the same time, and the difficulty of using a graph is determined by how many fixations are required (Kosslyn, 1985). Some researchers have suggested using the number of dimensions as an indicator for the complexity

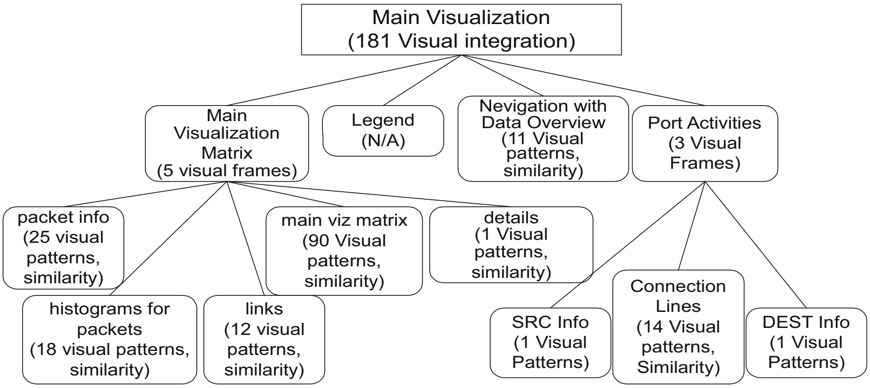


Fig. 3 Visual integration tree for TNV. Each of the child nodes contains the number of visual patterns and the Gestalt laws they are identified with. Each parent node contains the number of visual frames. The number of visual integrations is calculated by multiplying the number of visual frames and the sum of the visual patterns

of visualization (Kosslyn, 1989; Brath, 1997; Trafton et al., 2000). However, we believe that it is important to consider the difference in visual processing of integral and separable dimensions.

Garner (1974) has discussed the difference between integral and separable dimensions and how they are processed differently by human. In short, integral dimensions are processed together, while separable dimensions are processed individually. As a result, integral dimensions are likely to be processed faster than the separable dimensions. Garner has also provided a number of guidelines for identifying integral and separable dimensions. For example, in perceptual classification, stimulus sets defined by integral dimensions are classified primarily in relation to similarities; sets defined by separable dimensions are classified in relation to dimensional structure. Also in perceptual classification, dimensional preferences exist only for separable dimensions.

Based on the guidelines in (Garner, 1974), we have identified the following integral and separable dimensions:

- X and Y coordinates are integral dimensions. (Here we only consider two dimensional displays.)
- Color and value are integral dimensions.
- Shape, size, and orientation are each considered as a separable dimension.

It should be noted that the guidelines in (Garner, 1974) are rather abstract. Different people may have different interpretations and come up with different groupings of integral dimensions. Much work needs to be done to develop more specific guidelines for identifying integral and separable dimensions in visualization design.

In the case of TNV, there are five different dimensions in the main visual frame (Fig. 4), and three different dimensions in the port visualization frame (Fig. 5).

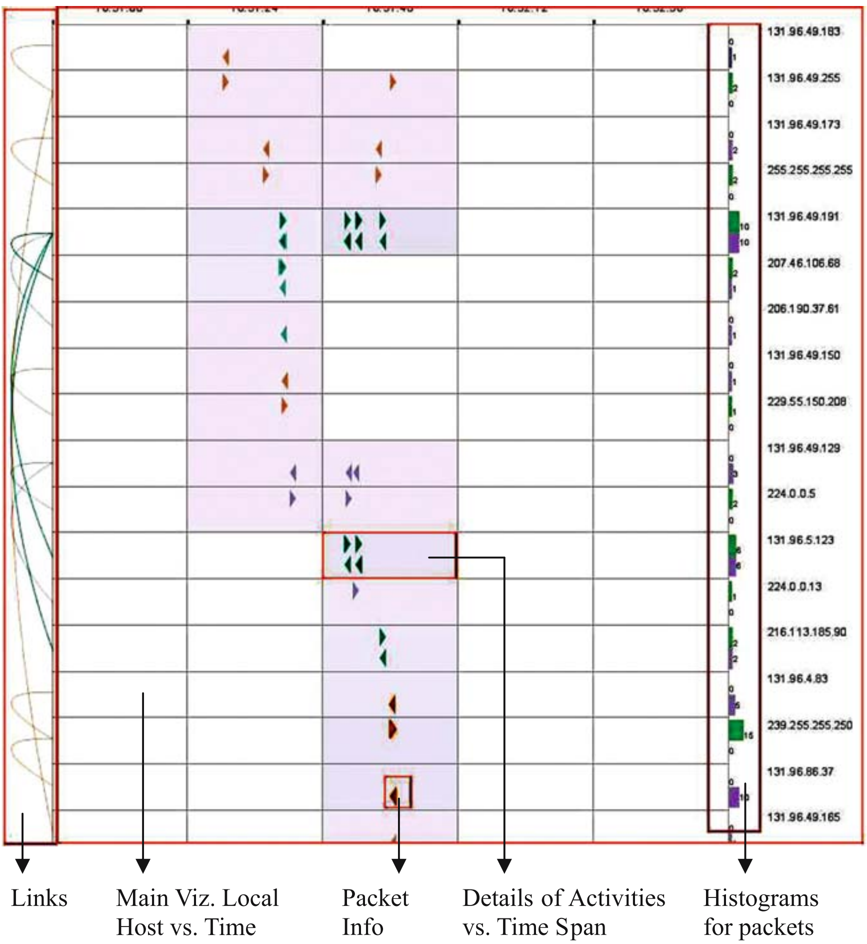


Fig. 4 Five different dimensions in TNV main visualization matrix. (**a**) Histograms for packets: categorized based on their shape/size/color. (**b**) Details of activities; categorized based on color/shape. (**c**) Main visualization matrix: categorized based on X–Y coordinate; in this case, they are local host vs. time. (**d**) Package information triangles: categorized based on shape/size/color. (**e**) Links: categorized based on shape/size/color

3.4 Interpreting the Values of Visual Attributes

Readers need to interpret the values of each visual attribute, which is another source of extraneous cognitive load. In our analysis, for each separable dimension, we assign a score for the complexity of interpreting the values of the visual attribute based on the following criteria:

For integral dimensions, the visual attributes are considered together. For example, X and Y coordinates are considered together and a single complexity score is assigned to both coordinates. In the end, each separable dimension has a complexity

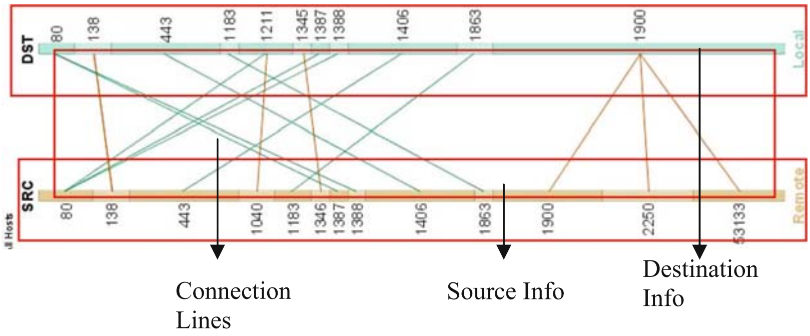


Fig. 5 Three different dimensions in port visualization. (**a**) Source and destination information: categorized based on the coordinate (*vertical axis*). (**b**) Connection lines between the two axes: categorized based on shape/color

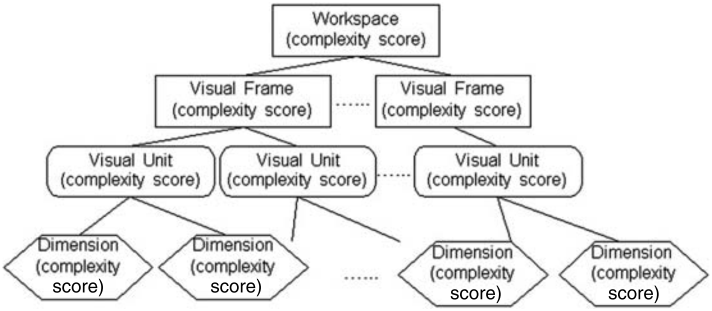


Fig. 6 Visual mapping complexity tree

score. The complexity score for a type of visual unit is the sum of complexity scores of its separable dimensions. The complexity score for a visual frame is the sum of scores of different types of visual units it contains, and so on. The structure of a typical visual mapping complexity tree is shown in Fig. 6.

The visual mapping complexity tree for TNV is shown in Fig. 7.

3.5 Efficiency of Visual Search

One of the main advantages of visualization is that they can support efficient visual search. According to Wolfe and Horowitz (2004), target-distracter difference is the key to efficient visual search, and there are four major factors that affect the target-distracter differences – color, motion, size, and orientation. Target-distracter difference indicates how a target stands out from the background, which can be any other surrounding objects or the neighboring background colors.

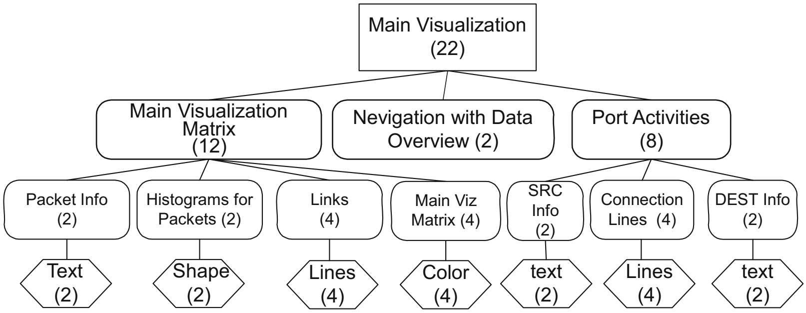


Fig. 7 Visual mapping complexity tree for TNV, each number below the visual units are the complexity scores based on Table 2. Every parent node's score is calculated as the sum of its children's scores

The target-background difference is determined differently for color, motion, size, and orientation through the following equations.

$$c = \left(\left| \frac{\sum_{i=1}^{n} T_r}{N} - D_r \right| + \left| \frac{\sum_{i=1}^{n} T_g}{N} - D_g \right| + \left| \frac{\sum_{i=1}^{n} T_b}{N} - D_b \right| \right) /765, \tag{1}$$

$$M = \frac{\left| \frac{\sum_{i=1}^{n} Tf}{N} - Df \right|}{F} \tag{2}$$

$$s = \frac{\sum_{i=1}^{n} \frac{|Ts - Ds|}{s}}{N} \tag{3}$$

$$o = \frac{\sum_{i=1}^{n} \frac{\sum_{i=1}^{n} |To - Do|}{N}}{180}. \tag{4}$$

Equation (1) is for calculating the target-background difference in color. Here T_r, T_g, and T_b are the R/G/B values of target color, D_r, D_g, D_b are the R/G/B values of the distracter whose color is the closest to the target color. N and n are the number of color components represented in the visualization. Number 765 is the distance between the two most distant colors.

Equation (2) is for calculating the target-background difference in motion. Here Tf and Df are the speed of motion for target and distracter, respectively. F is the speed of the fastest moving item in the visualization. N and n are the number of moving items.

Equation (3) is for calculating the target-background difference in size. Here Ts and Ds are the sizes of target and distracter items, respectively. S is the total area of visual frame that contains the target and distracters. N and n are the number of visual entities.

Table 2 Complexity scores for interpreting the meaning of visual units

Complexity score	Criteria
5	Very difficult to interpret. There is no legend. A typical reader has to memorize the mapping between the value of the visual attribute and the value of the corresponding data parameter
4	More difficult to interpret. A typical reader needs to frequently refer to a legend to interpret the value of the visual attributes
3	Somewhat difficult to interpret. A typical reader needs to refer to a legend from time to time
2	Relatively easy to interpret. A typical reader only needs to refer to a legend occasionally
1	Easy to interpret. This is based on common knowledge. There is no need to memorize or refer to a legend

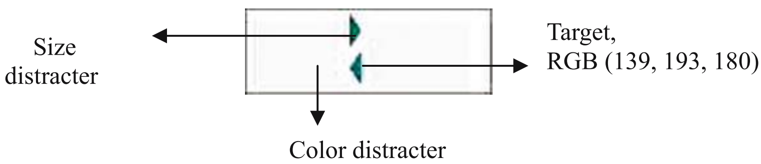


Fig. 8 Part of the main TNV visualization frame

Table 3 Target-distracter difference scores for TNV

	Color	Motion	Size	Orientation
Target-distracter difference scores	0.2850	N/A	0	1

Equation (4) is for calculating the target-background difference in orientation. Here *To* and *Do* represents the orientation of the target and distracters respectively. N and n are the numbers of distracters.

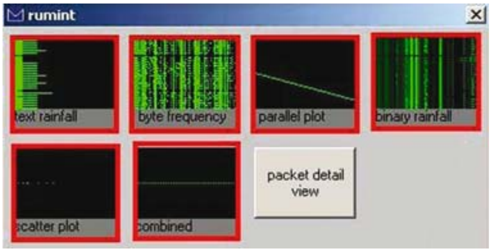
In the case of TNV, we calculated the target-distracter difference for the color, size and orientation of the visual units in the main visualization frame (Fig. 8). The result is shown in Table 3.

3.6 Case Study with RUMINT

RUMINT is an open source network security visualization tool developed by Conti (2005, 2007). Figure 9 shows a RUMINT thumbnail view after it finishes capturing 949 packets from the test dataset.

Figures 10 and 11 are the visual integration complexity tree and visual mapping complexity for RUMINT, respectively. Since RUMINT is a rather complicated visualization tool with many frames and dimensions, we have to simplify the trees

Fig. 9 RUMINT thumbnail overview



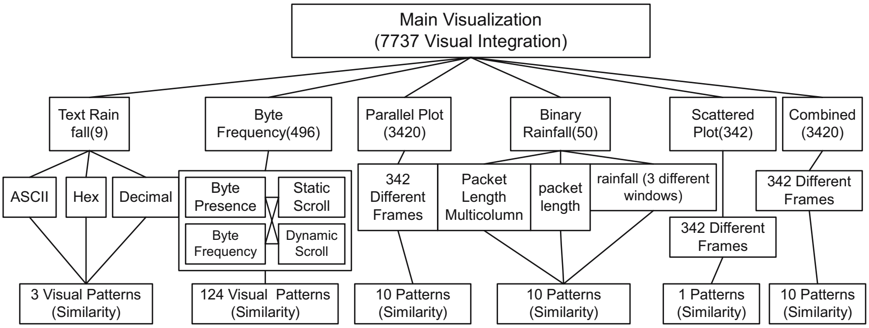


Fig. 10 Visual integration complexity tree for RUMINT

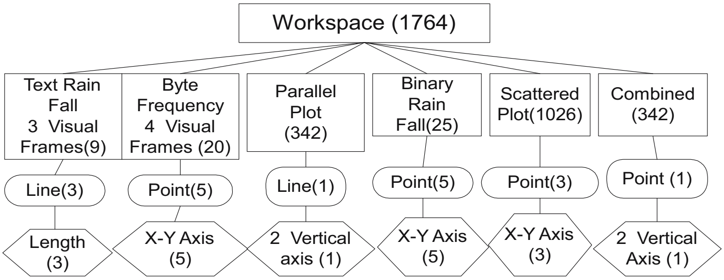


Fig. 11 Visual mapping complexity tree for RUMINT

in order to fit them into this paper. Table 4 shows the target-distracter difference scores for two visual frames of RUMINT.

It is necessary to point out that the case studies discussed above is not an attempt to directly compare TNV and RUMINT. Because the visual interfaces of these two software tools are quite different, it is hard to make meaningful comparisons between their complexity scores. For example, the two software systems may have the same complexity scores for different reasons. In other words, our proposed complexity analysis is primarily for comparing different design choices in the same visualization tool. It is not meant to compare visualization tools with drastically different visual interfaces.

Table 4 Target-distracter difference scores for two visual frames in RUMINT

Visual frames	Visual search guidance			
	Color	Motion	Size (pix^2)	Orientation
Overview	0.33333	n/a	0.43064	0
Byte frequency	0.33333	n/a	0.32169	0

4 Future Work

In the future, we plan to continue refining our complexity analysis methods. For example, the process of identifying integral and separable dimensions for visualization designs need to be further clarified and improved. The current equations for calculating the target-distracter differences are based on our intuition and need to be verified and revised through user studies.

Another focus of our future work is to study the relationship between the visualization complexity and usability. It will be interesting to see how the various complexity parameters affect the task completion time, error rate, learnability, and user satisfaction. We will need to correlate our complexity analysis results with formal user studies. We plan to use our complexity analysis to generate hypotheses of usability, and then design specific user studies to test these hypotheses.

Although the proposed complexity analysis method is developed in the context of computer security visualization, it can be easily extended to other information visualization areas. In particular, we plan to apply our complexity analysis method to bioinformatics visualization.

5 Conclusion

In this paper, we have presented a systematic methodology to measure the complexity of information visualization. Here the complexity is measured in terms of visual integration, number of separable dimensions for each visual unit, the complexity of interpreting the visual attributes, and the efficiency of visual search. These measures are based on well established psychological theories. Together they indicate the amount of cognitive load involved in comprehending a particular visualization design. The underlying hypothesis is that by reducing visualization complexity, we can improve the usability of the visualization.

The proposed complexity analysis is particularly useful during early design phase before any user studies can be conducted. It is intended to help developers quickly compare different design choices for a visualization tool. We have demonstrated this method by evaluating the complexity of two computer security visualization tools – TNV and RUMINT.

The proposed complexity analysis is not intended to be a comprehensive assessment of the usability of a visualization design. Rather it is focused on measuring the complexity of visualization designs in terms of visual cognition. Therefore it should be combined with user studies as well as other heuristic evaluation methods.

Acknowledgements The authors would like to thank Hsiu-Chung Wang for her contribution to the early works of this research.

References

Amar RA and Stasko JT (2005) Knowledge Precepts for Design and Evaluation of Information Visualizations. IEEE Transactions on Visualization and Computer Graphics 11(4): 432–442.

Bertin J (1983) Semiology of Graphics, University of Wisconsin Press.

Brath R (1997) Metrics for Effective Information Visualization. In Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis), IEEE.

Clark R, Nguyen F, Sweller J (2006) Efficiency in Learning: Evidence-Based Guidelines to Manage Cognitive Load: Pfeiffer.

Conti G (2005) Visual Exploration of Malicious Network Objects Using Semantic Zoom, Interactive Encoding and Dynamic Queries. In Proceedings of Workshop on Visualization for Computer Security (VizSEC).

Conti G (2007) Security Data Visualization: Graphical Techniques for Network Analysis: No Starch Press.

Garner WR (1974) The Processing of Information and Structure: Lawrence Erlbaum.

Goodall JR, Ozok, AA, Lutters WG, Rheingans P, Komlodi, A (2005) A User-Centered Approach to Visualizing Network Traffic for Intrusion Detection. In Proceedings of Conference on Human Factors in Computing Systems (CHI) extended abstracts, ACM.

Kosslyn SM (1985) Graphics and Human Information Processing: A Review of Five Books. Journal of the American Statistical Association 80(391): 499–512.

Kosslyn SM (1989) Understanding Charts and Graphs. Applied Cognitive Psychology 3: 185–226.

Larkin JH, Simon HA (1987) Why A Diagram is (Sometimes) Worth Ten Thousand Words. Cognitive Science 11: 65–99.

Shneiderman B (1996) The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In Proceedings of the IEEE Conference on Visual Languages, IEEE.

Tory M, Moller T (2005) Evaluating Visualizations: Do Expert Reviews Work? IEEE Computer Graphics and Applications 25(5): 8–11.

Trafton JG, Kirschenbaum SS, et al. (2000) Turning Pictures into Numbers: Extracting and Generating Information from Complex Visualizations. International Journal of Human–Computer Studies 53(8): 827–850.

Tufte ER (2001) The Visual Display of Quantitative Information, Graphics Press.

Wertheimer M, King D (2004) Max Wertheimer and Gestalt Theory: Transaction Publishers.

Wolfe JM, Horowitz TS (2004) What Attributes Guide the Deployment of Visual Attention and How do They Do It? Nature Reviews/Neuroscience 5: 1–7.

Zhou MX, Feiner SK (1997) Top-Down Hierarchical Planning of Coherent Visual Discourse. In Proceeding of International Conference on Intelligent User Interface (IUI), ACM.

Zuk T, Schlesier L, et al. (2006) Heuristics for Information Visualization Evaluation. In Proceedings of the Working Conference on Advanced User Interface (AVI), ACM.