Eurographics Conference on Visualization (EuroVis) (2012)
M. Meyer and T. Weinkauf (Editors)

Short Papers

ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data

Fabian Fischer[1], Johannes Fuchs[1] and Florian Mansmann[1]

[1]Data Analysis and Visualization Group, University of Konstanz, Germany

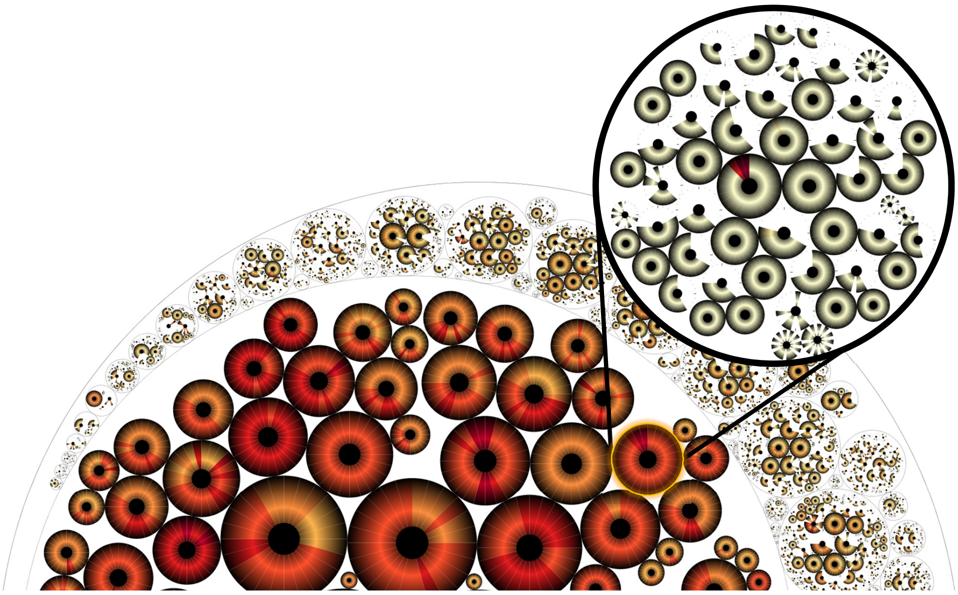

Figure 1: *ClockMap visualization showing the network traffic of a large number of subnets. Each glyph circle represents a 24-hour time-series of either a subnet or an IP address with respect to the semantic zoom level. They are laid out according to a circular treemap algorithm. Color is mapped to the amount of network traffic in bytes.*

Abstract

Treemaps are a powerful method to visualize especially time-invariant hierarchical data. Most attention is drawn to rectangular treemaps, because their space-filling layouts provide good scalability with respect to the amount of data that can be displayed. Since circular treemaps sacrifice the space-filling property and since higher level circles only approximately match the aggregated size of their descendants, they are rarely used in practice. However, for drawing circular glyphs their shape preserving property can outweigh these disadvantages and facilitate comparative tasks within and across hierarchy levels. The interactive ClockMap *visualization effectively supports the user in exploring and finding patterns in hierarchical time-series data through drill-down, semantic zoom and details-on-demand. In this study, the technique's applicability is demonstrated on a real-world dataset about network traffic of a large computer network and its advantages and disadvantages are discussed in the context of alternative layouts.*

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI)

1. Introduction

Many real-world datasets contain an intrinsic hierarchy, which can provide important information to the analyst. In network security, for example, such a hierarchy is often given through the network definitions encoded in prefixes of IP addresses. Especially for the analysis of network traffic of large computer networks, it is important to monitor the network usage to detect anomalies or to understand the behavior at different levels of detail. On the one hand, there is the need to gain an overview about the current situation. On the other hand, obtaining details and more information is crucial to understand such overall trends to eventually identify the underlying cause. To provide an integrated overview and detailed time-series information within a single visualization, we propose a visualization technique, called *ClockMap*, which uses the approach of circular treemaps as layout algorithm for a large number of temporal glyphs representing data values of a time-series. In particular, we apply this idea to a clock-based glyph inspired by the work of [KFM11], which we call *clockeye*. The advantage of this circular design is, that we can smoothly switch between different levels of the hierarchy and either show aggregated overview data for a subnet or show all individual time-series as glyphs.

The main contribution of this paper is the novel combination of clock-based glyphs with circular treemaps. Although, there are major drawbacks of such treemaps, we show in a case study, that the integration as *layout algorithm* for the placement of circular glyphs is quiet effective and can successfully be applied to network security data.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss related work. In Section 3 we describe our proposed visualization technique, and provide a case study in Section 4, discuss the technique in Section 5, and conclude with a summary and future work in Section 6.

2. Related Work

In the last decade, treemaps [Shn91] became one of the most popular techniques for visualizing hierarchical data. While there are many different treemap types, rectangular treemaps are used most often. Variants of such rectangular treemaps usually represent several data dimensions using area and color of the different rectangles within their actual hierarchy. Much research was conducted in the area of layout algorithms, but also in visual improvements of the different treemap variants. Cushion treemaps [VWvdW99], for example, use intuitive shading to provide better insights in the hierarchical structure. Since it is often important to compare different treemaps from different points in time, stability is an important criterion of the algorithms. The layout algorithms can be modified to consider such constraints. For example, [MKN*07] use treemaps to visualize data traffic and use geographic location to optimize the layout. This helps to compare different datasets of different points in time. Other adjustments of treemaps focused on the integration of temporal information within a single treemap to handle hierarchical *time-variant* data. [CSN*09] use animation in treemaps to be useful for dynamic data. Other improvements integrate glyphs or small charts to represent additional time-series information for a particular leaf node. However, this requires further optimizations of the layout algorithms [SKM06], because the different aspect ratio of the rectangles makes it hard to compare the different embedded time-series. Besides of the aforementioned rectangular treemaps other types have been developed like voronoi treemaps [BDL05] and circular treemaps [Wet]. However, for good reasons the circular treemap has not been frequently used. Circular treemaps waste space, because they “do not fill the available space completely” [Wet], which also means that “they fill the available space to a varying degree” and thus introduce imprecision in the aerial representation of the upper levels. In contrast to space-filling techniques, glyph visualizations are suitable representations [War08] for many different purposes. Especially to visualize a large amount of multi-dimensional data points or time-series, glyphs are thus widely used. In the essence, our approach is a combination of circular nested treemaps (e.g., Pebble Maps [Wet]) and a clock-like glyph for time-series data (cf. ClockView [KFM11]).

3. ClockMap

In the following, we will describe our novel visualization, called *ClockMap*, which is based on the combination of temporal glyphs, called *clockeyes*, and a circular treemap layout.

3.1. Clockeye Design for Time-Series Data

The basic idea of *clockeyes* is to make use of the metaphor of a classic clock. A circle is subdivided into sectors, each sector representing a time span of one hour. When 24 slices are used, we have a 24-hour clock as seen in Figure 2. In this example, there was no data from 00:00 to 06:00 o’clock and from 23:00 to 24:00, which results in a noticeable empty area in the representation. This can be very helpful to find specific patterns without data or zero data values. At one point between 06:00 and 07:00, the time-series seems to start, having high peaks between 08:00 to 09:00 and 10:00 to 11:00. Afterwards there is a downward trend until 24:00.

When many clockeyes are plotted to a dense area, it is important that they can be separated from each other intuitively, without the need to have an additional border in between. Circular shapes are very suitable for this purpose, because they are perceived as separate items pre-attentively. However, if many have the same color values, this task can becomes difficult in dense areas. To visually improve the perception of the compactness and further emphasize the borders, we applied circular shading, which seems to be an improvement according to our experiments. This generally led to darker colors, therefore, we decided to use an intense yel-

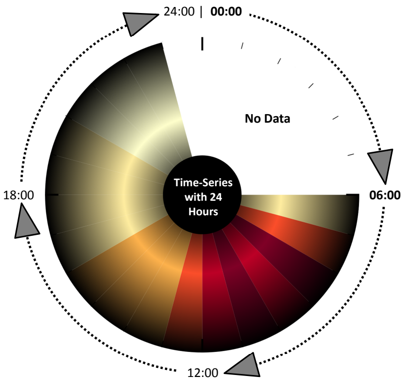

Figure 2: *Visual representation of a single clockeye showing a time-series of 24 hours. Each one hour sector is colored by its data value. Circular shading is applied to emphasize the borders of the glyph.*

low to red color mapping from ColorBrewer [Bre] to counterbalance this effect. The inner black circle can be used for additional meta labels or to indicate highlighting with color.

3.2. Combining Circular Treemaps with Clockeyes

As discussed in the Section 2 there are visualization techniques dealing with hierarchical data and others, e.g., glyphs, displaying temporal or multi-dimensional information. Especially in computer networks the combination helps to understand temporal dependencies in different substructures of the network. With *ClockMap* we use circular treemaps in combination with clockeyes. The circular treemap itself is often less powerful than rectangular layouts, however, in the combination with clockeyes it seems to be a promising use case. To make further use of the implicit characteristics of the layout algorithm, we implemented *ClockMap* on top of a zoomable user interface, which enables infinite zooming and panning possibilities. Each hierarchy can show the aggregated values for all underlying children to provide the user with a high-level overview as seen in Figure 3. While zooming into the aggregated areas more details and eventually each host represented as small clockeyes become visible. Through this semantic zooming, the scalability of the overall approach is improved, because less visual objects need to be drawn to the canvas when zooming out. Even with thousands of leaf nodes the visualization can be explored interactively. During exploration of real datasets it became obvious that in some cases very prominent nodes need to be removed or moved to another group. To facilitate this, we integrated edit operations to add hierarchies, remove nodes or place them freely into other circles or outside the main circle. After each modification the weights are changed accordingly to automatically recalculate the layout. To search for specific attributes of the nodes, a search field is integrated to *ClockMap*. The black inner circles of matching nodes are highlighted to guide the user to the relevant nodes.

4. Case Study: Visual Exploration of Network Traffic

Network operators of large networks use NetFlow data to analyze attacks and network usage. This datasets do not contain payload information, but do contain communication flows between hosts. We used an anonymized dataset of 24-hours with about 200 million NetFlow records collected at the core routers. The data is stored to a database and visually explored with *ClockMap*. The visual analysis does only focus on the records describing the outgoing traffic of all 6048 hosts belonging to our /16 IPv4 address block, which were active on that particular day. Figure 1 shows the upper part of the visualization. The analyst is interested in the highlighted subnet, because it has three hours (can be seen as deep red colored sectors), where much more traffic is transferred than usual. The total traffic originating form this subnet was 94.4 GiB. The tooltips show that most times of the day the transferred volume ranges only from ten to a few hundreds megabytes. The analyst selects this /24 subnet node and zooms in. The visual representation of this particular clockeye subnet will change to show all belonging hosts (shown as highlighted circle in Figure 1). This immediately shows that there is indeed a single host responsible for most of the traffic. It is up to the analyst if such nightly data transfers of an individual host in that particular subnet is legitimate or not. However, the visualization clearly shows, that compared to the other hosts in this group, this is indeed uncommon behavior. Figure 3 shows another very prominent pattern, which can be spotted in the *ClockMap* visualization. The subnet (which is shaped like a pac-man) reveals a strange time-series pattern. There was no traffic at all during night hours. This looks suspicious to the analyst. Zooming into this subnet reveals more details in Figure 4. This form

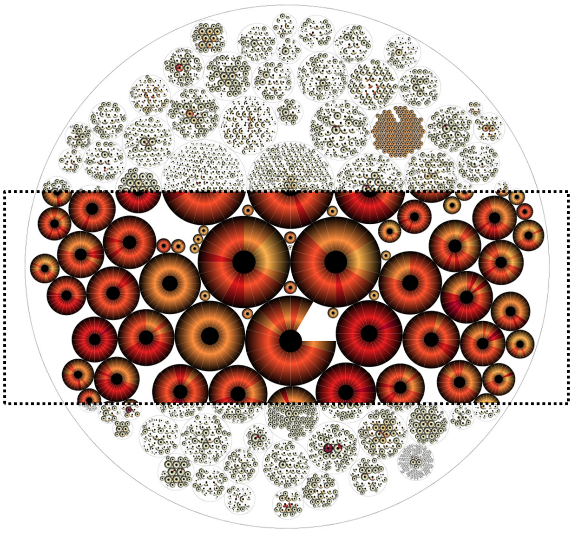

Figure 3: *A circular treemap is used to lay out hundreds of clockeyes into groups based on their hierarchy. The rectangle illustrates the visualization, when the user zooms out.*

of details on demand is implemented using semantic zooming. After a user-defined zooming threshold, the time-series for all underlying hosts become visible instead of the previously shown aggregated subnets. Such a pattern could be a network outage or indicate a broken switch in the building were the physical machines are located. However, in this case the pattern is legitimate, because it is known as wireless network subnet, which is not in use during night time.

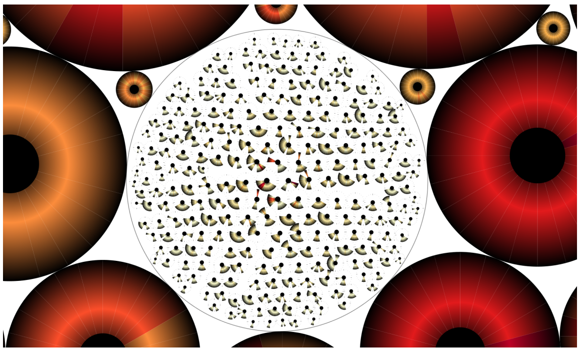

Figure 4: *Underlying hosts of a very prominent subnet outlier having no night time traffic.*

5. Discussion

The layout of glyphs is often determined by coordinate systems or matrix layouts. [KFM11] use a matrix representation to position IP addresses in a meaningful way. Compared to such matrix layouts, *ClockMap* has several advantages. Matrix representations cannot convey the hierarchy in an intuitive way. The circular treemap layout instead makes the hierarchy obvious, because it is visualized through nested circles. Another advantage is, that the aspect ratio does not change in *ClockMap*. We use circles, which can be further explored through interactive exploration with techniques like zooming and panning. The integration of semantic zooming helps to smoothly switch between general overviews and detailed time-series analysis. Both approaches are overlap-free, while the free arrangement in *ClockMap* results in a tighter packing of the glyphs and thus makes the approach slightly more scalable. In addition, the tight packing better supports the user to visually compare the shapes and color distributions of neighboring hosts in one branch of the displayed tree. Consequently, outliers with a different behavior in the group can be spotted pre-attentively. The used clockeye glyph has the advantage to use a common real-world metaphor. Everyone knows how to read a clock, which helps the user to identify particular hour values within the time-series. Visualizing time-orientated data effectively is important, non-trivial, and lead to a large variety of different visualization techniques. A systematic overview can be found in [AMST11]. However, it is even harder to visualize hundreds of different time-series simultaneously. Clockeye glyphs are very compact and general trends or patterns can be distinguished even on a very small scale. This helps to provide a scalable way to represent hundreds of time-series, and even more, when grouped within an hierarchy.

There are also drawbacks of our visualization technique, which are implicit by design. Circular treemaps are indeed not space-filling. This means that, at least compared to rectangular treemaps, space is wasted. However, compared to a matrix representation, this is not necessarily the case, because nodes are packed tightly together while still conveying the hierarchy information. The ordering within a group of the circular layout is also challenging and non-intuitive. This drawback can be overcome to a certain degree by interaction and tooltips. While comparison of shape and color distribution in circular layouts is effective, the comparison of the area of the circles is not. Additionally, the higher level circles only approximately match the aggregated size of their descendants. Consequently, the visualization is probably less precise with respect to these attributes. Clockeyes are using color to represent the data values, which makes it hard to precisely compare the values, which would be better in length-encoded glyphs. The basic design idea of clockeyes uses a clock metaphor. Obviously, this metaphor cannot be applied any more, if an arbitrary time-series length is used. This means, that a clockeye glyph is best suited for 12 or 24-hour time-series. Other lengths of time-series will be less intuitive, but are still possible from a technical point of view.

6. Conclusion

This paper described a novel visualization technique called *ClockMap* for hierarchical time-series data. The technique combines a circular nested treemap layout with a circular glyph representation for time-series data and appears to be effective for comparative tasks on large amounts of hierarchically structured time-series data. When being used in combination with circular glyphs, the shape preserving property of circular nested treemaps seems to outweigh the known disadvantages of such treemap variants and facilitates comparative tasks within and across hierarchy levels.

Since preliminary results of our experiments with the tool on network traffic data were promising, our next steps will be to generalize the basic idea of *ClockMap* in such a way that it can be applied to a wider range of datasets originating from different application fields. Furthermore, we plan to formally evaluate the effectiveness of the visualization in a user study and seek feedback of expert users. From such a study we expect to be able to judge which specific tasks of analysts can be improved with respect to both precision and performance when using the novel *ClockMap* representation.

7. Acknowledgements

The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257495, "Visual Analytic Representation of Large Datasets for Enhancing Network Security" (VIS-SENSE).

References

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of Time-Oriented Data*, 1st ed. Human-Computer Interaction. Springer Verlag, 2011. 4

[BDL05] BALZER M., DEUSSEN O., LEWERENTZ C.: Voronoi Treemaps for the Visualization of Software Metrics. In *Proceedings of the 2005 ACM symposium on Software visualization* (New York, NY, USA, 2005), SoftVis '05, ACM, pp. 165–172. 2

[Bre] BREWER C. A.: Colorbrewer - Color Advice for Maps. URL: http://www.colorbrewer.org/. 3

[CSN*09] CHIN G., SINGHAL M., NAKAMURA G., GURUMOORTHI V., FREEMAN-CADORET N.: Visual Analysis of Dynamic Data Streams. *Information Visualization 8*, 3 (2009), 212–229. 2

[KFM11] KINTZEL C., FUCHS J., MANSMANN F.: Monitoring Large IP Spaces with ClockView. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security* (New York, NY, USA, 2011), VizSec '11, ACM, pp. 2:1–2:10. 2, 4

[MKN*07] MANSMANN F., KEIM D. A., NORTH S. C., REXROAD B., SHELEHEDA D.: Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007). 2

[Shn91] SHNEIDERMAN B.: Tree Visualization with Tree-Maps: A 2-D Space-Filling Approach. *ACM Transactions on Graphics 11* (1991), 92–99. 2

[SKM06] SCHRECK T., KEIM D. A., MANSMANN F.: Regular TreeMap Layouts for Visual Analysis of Hierarchical Data. In *Proceedings of the Spring Conference on Computer Graphics (SCCG'2006)* (Casta Papiernicka, Slovak Republic, 2006), ACM Siggraph. 2

[VWvdW99] VAN WIJK J. J., VAN DE WETERING H.: Cushion Treemaps: Visualization of Hierarchical Information. In *Proceedings of the 1999 IEEE Symposium on Information Visualization* (Washington, DC, USA, 1999), IEEE Computer Society, pp. 73–. 2

[War08] WARD M.: Multivariate Data Glyphs: Principles and Practice. *Handbook of Data Visualization* (2008), 179–198. 2

[Wet] WETZEL K.: Pebbles - Using Circular Treemaps to Visualize Disk Usage. URL: http://lip.sourceforge.net/ctreemap.html. 2