Leaf Glyphs: Story Telling and Data Analysis using Environmental Data Glyph Metaphors

Johannes Fuchs, Dominik Jäckle, Niklas Weiler, and Tobias Schreck

University of Konstanz, Data Analysis and Visualization Group,
Universitätstr. 10, 78457 Konstanz, Germany
fuchs@dbvis.inf.uni-konstanz.de

Abstract. In exploratory data analysis, important analysis tasks include the assessment of similarity of data points, labeling of outliers, identifying and relating groups in data, and more generally, the detection of patterns. Specifically, for large data sets, such tasks may be effectively addressed by glyph-based visualizations. Appropriately defined glyph designs and layouts may represent collections of data to address these aforementioned tasks. Important problems in glyph visualization include the design of compact glyph representations, and a similarity- or structure-preserving 2D layout. Projection-based techniques are commonly used to generate layouts, but often suffer from over-plotting in 2D display space, which may hinder comparing and relating tasks.

Inspired by contour and venation shapes of natural leafs, and their aggregation by stems, we introduce a novel glyph design for visualizing multi-dimensional data. Motivated by the human ability to visually discriminate natural shapes like trees in a forest, single flowers in a flowerbed, or leaves at shrubs, we design a flexible leaf-shaped data glyph, where data controls main leaf properties including leaf morphology, leaf venation, and leaf boundary shape. Our basic leaf glyph can map to more than a dozen of numeric and categorical variables. We also define custom visual aggregation schemes to scale the glyph for large numbers of data records, including prototype-based, set-based, and hierarchic aggregation. We show by example that our design is effectively interpretable to solve multivariate data analysis tasks, and provides effective data mapping. The design provides an aesthetically pleasing appearance, and lends itself easily to storytelling in environmental data analysis problems, among others. The glyph and its aggregation schemes are proposed as a scalable multivariate data visualization design, with applications in data visualization for mass media and data journalism, among others.

Keywords: Glyph visualization and layout, nature-inspired visualization, leaf shape, multi-dimensional data analysis, data aggregation.

1 Introduction

Glyph-based data visualization has a long tradition in Information Visualization research and application. The basic idea in glyph visualization is to map data

properties to visual properties of some appropriately designed visual structure. By the interplay of the different visual properties, each glyph then represents a data record. Many data records can be compared by appropriately laid out glyph displays. Glyph visualization, like other areas in Information Visualization, can be considered both a science and an art. Specifically, the design of glyphs may be inspired intuitively by common, well-known shapes or icons. For example, Chernoff faces were inspired by face properties, and sticky figures by abstraction of human body shapes.

A subset of the designs studied in Information Visualization to date has been inspired by nature. For example, tree structures have inspired hierarchical node-link diagrams. As another example, the notion of information landscapes or terrains is also borrowed from nature. There is reason to believe that the human visual sense, due to long evolutionary processes, is highly trained in recognizing, distinguishing and comparing natural forms. These visual recognition processes typically work well even in low illumination conditions, or in presence of partial occlusion of natural objects. By background knowledge and experience, humans are able to efficiently recognize natural shapes, also often in cases where only parts of the shape or their boundary are visible.

Based on this motivation, we investigate the design space for leaf shapes as natural metaphors for data glyphs. From observing leaves in nature, it is clear that there is a large variability in the different types and forms of leaves that exist. Overall leaf shape, shape boundary, and shape interior all comprise several visual parameters that can in principle, be used to map data to generate glyphs. To the best of our knowledge, this is the first work to systematically study the design space of leaf-based glyph visualization, and identify an encompassing set of leaf variables to map data to. In conjunction with appropriate glyph layouts (based e.g., on projection), and visual aggregation techniques, effective and intuitive data displays can be realized. Our rationale for using leaf-based data visualization is two-fold. First, the design space is large, giving ample opportunities for the visualization expert to map data variables to visual variables. As will be discussed, our variable space amounts to more than 20 different visual variables that can be controlled. While we have not formally evaluated the effectiveness of these variables or their combinations, we presume this is a large design space from which appropriate effective selections can be found. Second, we propose that nature-inspired designs, by their potential aesthetic appearances and familiarity, can be suited to spark interest in visual data analysis for wider audiences, e.g., for use in mass media. Also, it resonates well with visualization of environmental data, as has been previously demonstrated, e.g., by a respective infographic used by OECD (see Section 2.2).

The remainder of this paper is structured as follows. In Section 2, we discuss glyph-based and nature-inspired data visualization approaches. Section 3 defines the design space for leaf glyphs, based on identification of main visual leaf properties which are candidates for data mapping. Then, in Section 4, we define several visual aggregation schemes to scale 2D glyph layouts for large numbers of data points. Section 5 then applies our design to several data sets. By exemplary

data analysis cases, we demonstrate the principal applicability of our approach. Finally, Section 6 summarizes our work and outlines future research in the area.

2 Related Work

Our work extends the design space of two existing branches of research by introducing a compact data representation making use of environmental cues. The related work is, therefore, split into two parts. The first part covers the area of space efficient visualization techniques, namely, data glyphs. The second part addresses research using environmental cues to convey data. We do not address research in the area of computer graphics, since this work mainly focuses on photo-realistic representation of the environment. We refer the interested reader to a summary work about this topic by Deussen and Lintermann [Deussen and Lintermann, 2005].

2.1 Glyphs

In the literature, there exists a large variety of glyph designs. Elaborate summaries can be found in [Borgo et al., 2012] [Ward, 2008]. To come up with a comprehensive categorization we make use of Ward's classification of data glyphs [Ward, 2008]. In his research he distinguishes between three different ways a data point can be mapped to a glyph representation.

First, many-to-one mapping: All data dimensions and their respective value are mapped to a common visual variable. Therefore, these designs can be systematically created by choosing the most effective visual variable for a certain task. Additional guidance is given by Cleveland et al. with a ranking of visual variables [Cleveland and McGill, 1984]. Well-known examples making use of a position/length encoding are star glyphs [Siegel et al., 1972], whisker and fan plots [Pickett and Grinstein, 1988][Ware, 2012], or profile glyphs [Du Toit et al., 1986]. The designs just differ in their layout of the dimensions (i.e., circular or linear) and some minor variations like the presence or absence of a surrounding contour line. Other glyph designs make use of color encodings to represent the data value. Clock glyphs [Kintzel et al., 2011] map the dimensions in a radial fashion, whereas pixel-based glyph designs [Levkowitz and Herman, 1992] layout the dimensions linearly. Of course, color cannot convey the data as accurate as a position/length encoding [Fuchs et al., 2013], however, for certain tasks like spotting outliers the color encoding is a reasonable choice. There is even a design mapping the data values to the angle of its rays. Sticky figures [Pickett and Grinstein, 1988] use the visual variable orientation, which is not so accurate in communicating exact data values. However, when used as an overview visualization the designs convey individual shapes, which are perceived as a whole nicely approximating the underlying data point.

Second, one-to-one mapping: Each dimension is mapped to a different visual variable. Probably, the most well-known representations here are Chernoff faces [Chernoff, 1973]. The single data values are mapped to face characteristics, like the size of the nose or the angle of the eyebrows. Other more exotic designs are bugs [Chuah and Eick, 1998] (changing the shape, length or color of wings, tails and spikes), or hedgehogs [Klassen and Harrington, 1991] (manipulating the spikes by changing the orientation, thickness and taper). The major drawback of these kinds of glyph representations is that they are often sensitive to the order by which the data dimensions are mapped to visual variables. Variation of the order could significantly change the final glyph representation and its visual perception by users. Additionally, measuring differences between single dimension values within a data point is typically a difficult task, as the analyst has to compare different kinds of visual variables with each other (e.g., compare length with saturation or angle, etc.)

Third, one-to-many mapping: The dimensions are represented by two or more visual variables. This redundant mapping can be useful to strengthen the perception of individual dimensions. For example, in star or profile glyphs the dimensions can be additionally encoded by coloring the single data rays. Clock glyphs can make use of an additional length encoding for the single colored slices to encode the underlying data values more accurately.

Metaphoric glyph designs: Another category of glyph representations are metaphors for communicating domain specific data. A well-known example are Chernoff faces [Chernoff, 1973], which were already introduced in the one-to-one mapping category. In two quantitative experiments conducted by Jacob and Flury et al. these faces were compared against other visual representations like polygons or simple digits. In both evaluations data from human beings like anthropometric variables [Flury and Riedwyl, 1981] or medical patient information [Jacob, 1978] had to be encoded. The results indicate that metaphors outperform the more abstract designs. In addition, also other metaphoric glyph designs like clock glyphs [Fuchs et al., 2013] or car glyphs [Surtola, 2005] have been subject to quantitative experiments yielding similar results.

As can be seen from these experiments metaphoric designs seem to be superior for specific domains compared to more abstract representations. This insight is an interesting starting point to think about designs for visualizing environmental data.

2.2 Environmental Cues

Visualizations making use of environmental cues need not necessarily be glyph representations. Stefaner uses an abstract tree layout to show the editing history of Wikipedia entries represented as single branches [Stefaner, 2014a]. The branches grow to the right whenever people decided to delete an article or to the left in the other case. The resulting tree nicely summarizes 100 articles with the

longest discussion whether to keep them or not. Another tree-based approach in combination with leaves visualizes poems in a more artistic way [Müller, 2014]. The branches of the tree are invisible just dealing as an anchor point to arrange the glyphs. Each word in the poem is represented with a leaf glyph and attached along the tree structure. The work is not eligible of representing the text data accurately but tries to illustrate a creative unique picture or fingerprint of the underlying poem.

A more data-driven glyph design is the botanical tree [Kleiberg et al., 2001], which again uses a 3D tree layout to represent hierarchical information. The single nodes are represented as fruits. The authors argue that people can more easily identify single nodes in this visualization compared to a more abstract representation because they are used to detect fruits or leaves on shrubs or trees. A 2D visualization using a botanical tree metaphor are so-called Contact-Trees [Sallaberry et al., 2012] which show relationships in data, e.g., contacts between persons. The branches consist of single lines representing an attribute in the data, e.g., a longer line refers to an older tie between people. Finally, fruits or leaves are added to the tree according to some data property, e.g., the kind of relation between people (friends, co-workers etc.). However, the fruits and leaves are highly abstract representations (mainly colored dots) and their shape does not change according to some data characteristics. The OECD's Better Life Index visualization [Stefaner, 2014b], on the other hand, systematically changes the appearance of the single flower glyphs used to represent data. Stefaner uses such environmental cues to visualize multi-dimensional data about country characteristics. Each country is represented by one flower. The petals encode the different economic branches with varying sizes and lengths for the corresponding values. The flowers are arranged according to their weighted rank across all dimensions. People can change the layout by changing the weights of the dimensions or simply focusing on just one dimension.

We contribute to this body of existing work with the definition of a highly detailed leaf glyph, which closely follows the main morphological and functional variations among leaves. It is able to effectively map data variables. We also provide a custom aggregation scheme to scale leaf layouts for large number of records.

3 Environmental Glyph

According to Biological literature, leaves may be categorized by their function or usage in the environment [Beck, 2010]. For our purposes, we divide leaves according to their shape (or morphology). The overall appearance of a leaf consists of the combination of (1) the overall shape type, (2) the boundary details, and (3) the leaf venation. We consider these three aspects as the main dimensions for controlling the leaf glyph by mapping data. As a result we come up with a design space structured along the overall leaf shape, which we discuss next.

3.1 Leaf Shape Design Space

Following Palmer who pointed out: "Shape allows a perceiver to predict more facts about an object than any other property" [Palmer, 1999], this visual variable should be used for the most important data dimension. In the environment, there exists a nearly endless amount of different leaf shapes since each leaf is unique. However, it is possible to distinguish leaves according to their overall shape [Deussen and Lintermann, 2005]. A first categorization can be done between conifer and deciduous leaves.

Conifer leaves can be found for example at fir or pine trees and have a thin long needle-like shape. Therefore, they do not offer much space for a venation pattern, which we want to use later for mapping additional attributes (e.g., Acicular leaves). Since the differences in shape are quite small for the different kinds of this group and the provided area is limited due to the distorted aspect ratio, we do not consider them in our design space.

Deciduous leaves cover a large group of different shapes and can again be further divided into four sub-categories [Deussen and Lintermann, 2005].

Pinnate and *palmate* compound leaves are shapes, which consist of several smaller leaflets attached to a shared branch (e.g., Alternate, or Odd and Even Pinnate leaves etc.). In order to avoid any misinterpretation between single leaflets at a branch and individual leaves, we discard this group from our final design space. However, these kinds of leaves seem an appropriate representation to visually summarize multiple data points where one leaflet corresponds to a single leaf.

Lance-like leaves have a parallel venation and are thin and long, similar to conifer leaves. Therefore, it is difficult to distinguish different kinds of these leaves since the differences in the overall shape are limited. Like the conifer leaves, we do not keep them in our design space because of the limited area to map a venation pattern, and because of possible confusion of different lance-like shapes.

Leaves with *net veins* or *reticulate* venation patterns encompass the largest group of deciduous leaves with a big diversity in shape. We restrict ourselves to the most common leaf shapes for this category to avoid misinterpretation of intermediate structures, which could not clearly be distinguished. Additionally, we focus on leaves with a big surface to show venation patterns and small stems to save space. Leaves similar to Flabellate, Unifoliate, etc. will, therefore, not be considered.

The most important requirement for shapes in visualizations is that they should be easily distinguishable. Therefore, our final design space covers elliptic (e.g., Ovate, Obtuse, Obcurdate etc.), circular (e.g., Orbicular), triangular (e.g., Deltoid), arrow-like (e.g., Hastate, Spear-shaped etc.), heart-like (e.g., Cordate, Deltoid etc.), two variations of tear-drop like (e.g., Acuminate, Cuneate etc.), wave-like (e.g., Pinnatisect), and star-like (e.g., Palmate, Pedate, etc.) shapes. Figure 1 illustrates the nine different leaf shape categories covered by our design space. In Section 5 we will introduce a heuristic to map data points to leaf shapes, based on the idea of representing outlying points by the more jagged leaf shapes;

conversely, non-outlying points will be represented by the more regular or smooth leaf shapes.

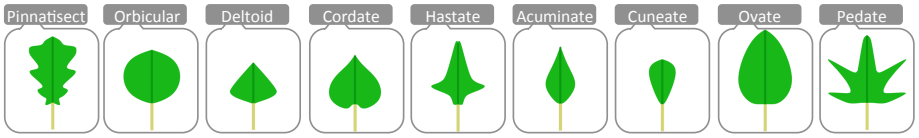

Fig. 1. Leaf shapes: Selected from our overall design space, these are the shapes used in our final glyph design. From left to right: Wave-like, circular, triangular, heart-like, arrow-like, tear drop up, tear drop down, elliptic, and star-like shapes.

We take these categories as a starting point and further extend them by mapping additional attribute dimensions to the width and the height of the glyph, scaling the overall shape. Therefore, similar shapes according to a certain data characteristic can look different because of the varying aspect ratio. However, the individual shape categories can still be distinguished (Figure 2). Because of this decision, we will deviate from the precise environmental reference, where leaves typically show a homogeneous aspect ratio. However, we thereby are able to encode additional data dimensions. Note that we do not want to represent leaves as accurate as possible (or even photo realistic), but use their expressiveness to visualize data.

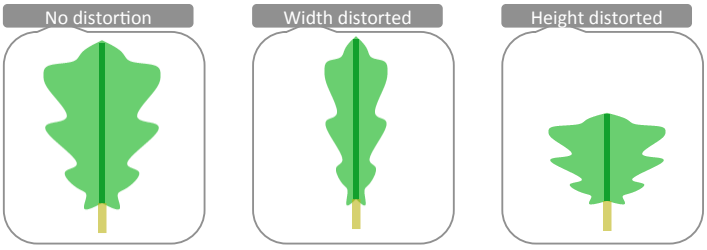

Fig. 2. Leaf scaling: The Lobate leaf shape is scaled using either the width (middle), or the height (right) of the glyph. Even after scaling, the glyph can still be recognized as a wave-like leaf, although the precise environmental reference to the Lobate leaf is reduced.

3.2 Leaf Boundary Design Space

Basically, the boundary (or margin) of a leaf can be described as either serrated or unserrated. *Unserrated* boundaries have a smooth contour adapting to the

overall leaf shape. *Serrated* boundaries are toothed with slight variations depending on the size of teeth, their arrangement along the boundary, and their frequency. Of course, there are more detailed differences and variations in nature. However, especially in overview visualizations (the major domain of data glyphs), distinguishing between small variations of the contour line of a leaf shape is nearly impossible. We therefore focus on just the two main boundary categories of teethed or smooth (serrated or unserrated). For mapping data values to the leaf boundary, we distinguish between a smooth and a toothed contour line and vary the width, height, and frequency of the teeth according to the underlying data value (Figure 3).

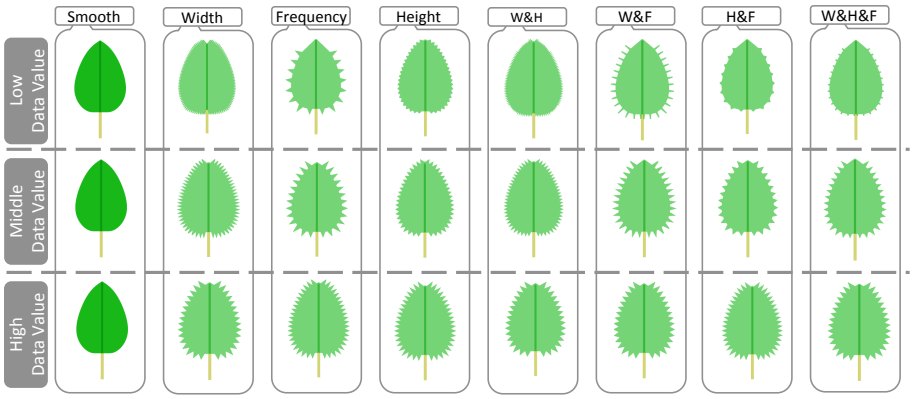

Fig. 3. Leaf boundary: Modifying the boundary in our design is realized by changing the frequency, the height, or the width of the boundary serration (teeths). Combinations of these three variables are possible and increase the expressiveness of the glyph. The figure illustrates all possible combinations for low, middle, and high data values for an elliptically shaped leaf glyph.

3.3 Leaf Venation Design Space

We also control the leaf venation pattern as to map additional data variables to the glyph. Several main leaf venation patterns exist, which differ in their overall structure within the leaf. A rough distinction can be made between single, not intersecting (e.g., Parallel), paired (e.g., Pinnate), or net-like (e.g., Reticulate) veins. The venation is perceived as an additional texture for the glyph and further increases the glyph expressiveness. Since it is hard to find a natural order within this texture, we propose to use the venation type for visualizing qualitative (or categorical) data, similar than the overal leaf shapes discussed in Section 3.1. Within a given venation type, we may also encode numeric data. This works as follows. Generally, the leaf is split in the middle by a main vein, with small veins

growing from there in a given direction (angle). For mapping numerical data, we may either control this *angle of the veins* branching out from the main vein. An alternative is to control the *number of veins* shown on the surface Figure 4. As a result, we come up with a venation texture able of encoding categorical and numerical data.

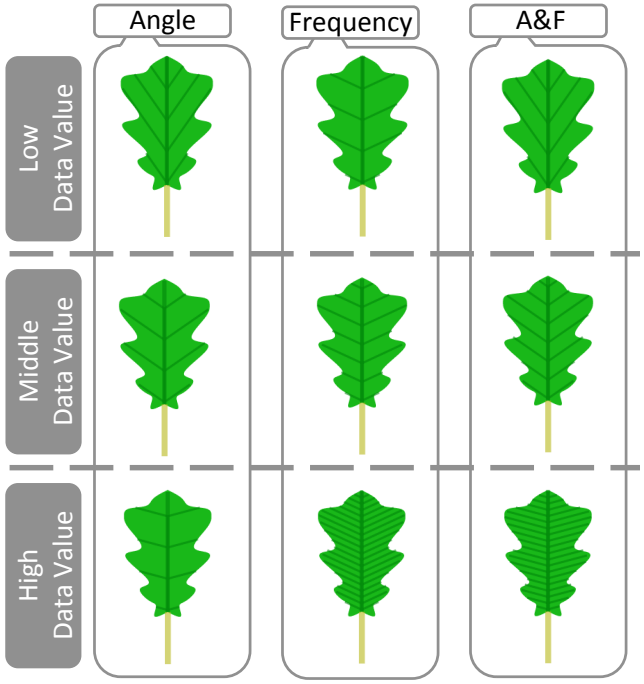

Fig. 4. Leaf venation: The texture for the venation system can either be created by mapping data values to the angle or frequency of the veins separately, or by combining the two. The figure illustrates all possible combinations for low, middle, and high data values for a wave-like leaf shape.

3.4 Summary

Besides modifying the leaf shape given by morphology, boundary and venation, further dimensions can be assigned to the color hue or saturation of the glyph. Of course, the designer has to pay attention to the contrast between the venation texture and the background color. Additionally, orientation of the glyph in the display can be used to encode further numeric information. We draw a short stem to each leaf shape, showing its orientation. Finally, it is also possible to modify the stem's width or height as well.

This represents a comprehensive design space for mapping data to leaf glyphs, controlled by 12 categorical and 14 numeric parameters, summing up to 26 variables altogether (see Table 1 for an overview of all variables.) We propose this design space as a toolbox from which the designer may select visual variables as appropriate. The number of 26 parameters is considered more a theoretical upper limit of data variables that we can show. We expect not all visual parameters in this design space to be of the same expressiveness; but some variables may be more effective than others, and may not all be orthogonal to each other. Careful choice should be done in selected and prioritizing the variables. An option is of course always, to redundantly code data variables to different glyph variables, to emphasize perception of important data variables. In Section 5, we will illustrate by practical examples, how glyph variables can be combined to form data displays.

Leaf Design	**Numeric Variables**	**Categorical Variables**
Shape	2 (x/y scale)	9 (selected morphologies)
Boundary	3 (frequency, width, height of teeth)	–
Venation	2 (number, angle of child veins)	3 (parallel, paired, net)
Other	7 (hue, saturation, orientation, x/y position, stem width/height)	–
Sum	14	12

Table 1. Summary of the parameters of our glyph design. It comprises 14 numeric and 12 categorical variables, which form the theoretic upper limit for the expressiveness of our glyph. Note that in practice, these variables are expected to not all be orthogonal, and comprise different perceptional performance, depending also on the data.

4 Leaf Glyph Aggregation

When visualizing large data sets, leaf glyphs, like many other glyphs, are prone to overlap in the display, reducing the effectiveness of perceiving data from individual glyphs. Generally, an increasing amount of multivariate points in a visualization produces significant clutter resulting in perceptional problems – the user is not able to distinguish between data points properly anymore. This is mainly due to our design intention to use larger shapes for adding e. g., venation patterns. Next, we discuss three different aggregation techniques, to help cope with large numbers of data points in our glyph display: Alpha Compositing, Prototype Generation, and Abstraction.

First, we apply transparency in Figure 5 to provide a visually pleasing representation that also reveals differences between data points. In some cases, the application of transparency is not enough. For example, if multiple data points share the same position, the opacity might sum up until no difference is perceivable. Therefore, we propose two different aggregation techniques that build on

top of transparency and the application of a grid-based aggregation. Specifically, we place a user-defined grid on top of the visualization. All data points sharing the same cell are aggregated (see Figure 6).

These effects can at the same time be perceived in nature: leaves can overlap or coincide with others. We adapt the proposed aggregation techniques and extend them in order to find a representative aggregate glyph which summarizes multiple leaf glyphs.

In Figure 5 and Figure 6 we point out the application of the aggregation techniques – Alpha Compositing, Prototype Generation, and Abstraction – with respect to nature. We next explain them in terms of their counterpart in nature, and apply them to our visualization of leaf glyphs.

4.1 Alpha Compositing

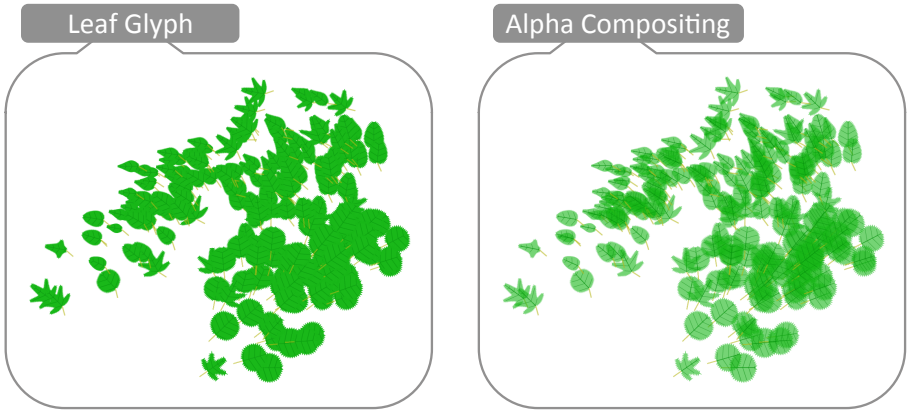

Fig. 5. Aggregation by Alpha Compositing. When multiple leaves overlap or coincide, we are not able to distinguish properly between their shapes and related characteristics. To overcome this issue, we propose to apply alpha compositing. It reveals details by applying transparency to the leaves.

We use Alpha Compositing [Porter and Duff, 1984] to reveal details on overlapping glyphs by applying transparency. This technique describes the process of combining multiple, separately rendered images in order to provide a transparent appearance. The result of the application of transparency to the glyphs is shown in Figure 5.

As mentioned in Section 3, different leaf shapes and characteristics need to be taken into account. In nature, leaves own the characteristic that even when multiple leaves overlap, we perceive differences due to their diverse shape and color. To support this, we apply transparency to the leaves. Figure 5 presents

the first results. The application of transparency works well, in our experience, for a limited amount of leaf glyphs. When too many leaves overlap, perceptional problems can arise: Since the transparency also aggregates, from a certain extent on, the glyphs can become occluded and not be distinguishable anymore. For this reason, we propose two additional aggregation techniques we observed in nature: Prototype Generation and Abstraction.

4.2 Prototype Generation

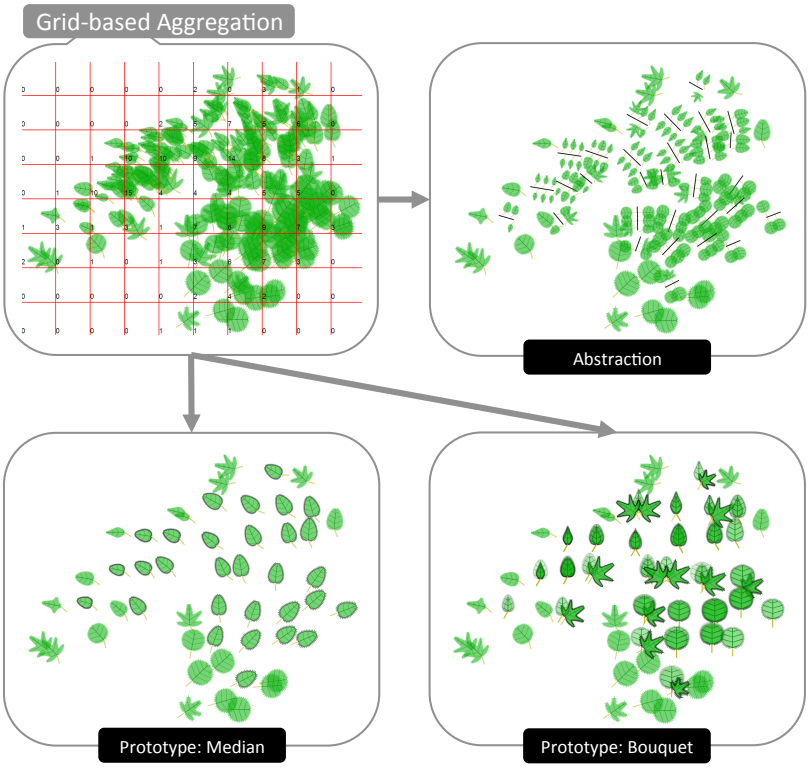

Fig. 6. Grid-based aggregation. We apply a grid to the visualization and calculate the center point of each leaf glyph, and aggregate all glyphs whose center points coincide within the same cell. Two different aggregations can be used: *Prototype Generation* and *Abstraction*. The first determines a representative glyph for the corresponding cell in the form of a median glyph or a bouquet glyph. The second creates (similar to what we observe in nature), a branch with multiple leaves based on the attributes of the considered leaves.

As mentioned above, transparency may not be enough when aggregating multiple glyphs. Therefore, we propose to additionally generate a prototype glyph that aggregates the characteristics of all considered glyphs. We apply a grid and aggregate all leaves the calculated center point of which fall into the same grid cell; the cell dimensions are user defined. The glyph representing each cell can be given either by 1) a single glyph, determined by statistical aggregates of the member element dimensions, e.g., the mean or median values, or 2) a visual aggregate combining small multiples of the member elements, by a connecting structure (so-called bouquet glyph, inspired by combinations of different flower types). Figure 6 shows the result of both techniques, visualization of the median as well as the visualization in form of a bouquet. For both techniques, the transparency is preserved to be able to distinguish between different attribute values that determine the shape of a leaf glyph.

Our first proposed prototype is the representation of the median. We therefore create a new leaf glyph that has a simple appearance by means of its shape. We use the median venation, margin, and shape in order to describe a set of leaves that coincide in one cell.

Similar to a bouquet, we derive our second proposed prototype by combining and aligning all contained leaf glyphs. First, all leaf glyphs sharing the same shape are stacked using transparency as described in Section 4.1. Second, stacked leaf glyphs are aligned in a radial manner according to their shape. This means, while in the first step glyphs are stacked according to their shape, in the second step they are radially moved and aligned according to the shape classes as pointed out in Section 3. As a result, we get a representation similar to a bouquet.

4.3 Abstraction by Visual Aggregation

Based on the grid aggregation, we need to address issues that emerge when too many glyphs fall into one cell. Prototype generation may fail, if too many glyphs along too many different shapes are aggregated, and the visualized prototype may then suffer from clutter. Therefore, we propose abstraction by visual aggregation. We describe the new visual representation for an aggregated set of glyphs. Similar to growth characteristics of leaves we observe in nature, this aggregation technique represents an aggregated set of leaf glyphs as a new branch with multiple leaves on it. All leaf glyphs are aligned side-by-side along a branch according to Figure 6.

4.4 Hierarchical Aggregation

The previously introduced aggregation techniques are not only suitable to visualize dense areas in 2D projections. Another design alternative is to use hierarchical arrangements, which can convey aggregate information and therefore, help with scalability. The relevant concept is that of a dendrogram (see Figure 7). Each parent node in a dendrogram may be represented by an aggregate prototype showing properties of the represented data partitions. Basic hierarchical

visualizations can, therefore, be enriched with additional information like the composition of data points for individual clusters.

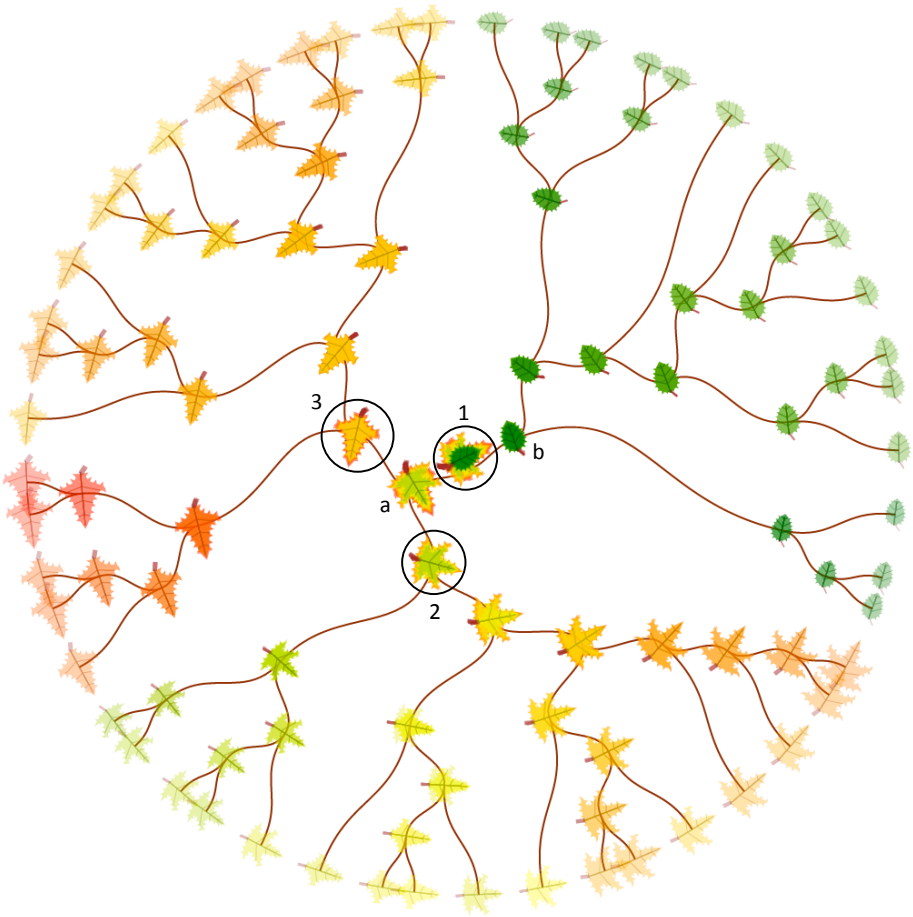

Fig. 7. Enhanced dendrogram: A selection of data points from the iris dataset have been hierarchically clustered and their structure represented in a radial dendrogram. Leaf glyphs are used to visualize the groups and individual data points along the hierarchy. As can be seen, the visual structure of the leaf glyph is getting more and more precise when approaching the leaf nodes illustrating the homogeneity of the lower levels in the dendrogram.

In Figure 7 we clustered the Iris dataset from the UCI Machine learning repository and represented the hierarchical structure in a radial dendrogram. The class attribute is used to assign different leaf shapes to the data. Other visual features like color, venation, and margin represent different attribute dimensions of the dataset. In each level, the nodes have been replaced with aggregated leaf

glyphs using alpha composition together with a position bundling. The leaf glyph positioned in the middle of the visualization (*#1*) aggregates the dimension values of all nodes in the diagram. It, therefore, contains many different sub-clusters as can be seen in Figure 7. When traversing the single branches to the lower levels (from inside out) the prototype representations of lower aggregate levels are getting more homogeneous. For example, after the first hierarchical split two main clusters are separated (*a* and *b*). The node labeled with *b* shows only green ovate leaves thus representing a homogeneous group of data points. The other aggregated prototype labeled with *a* seems to be more heterogeneous showing two different kinds of leaf shapes (hastate leaves and maple leaves). However, after descending to the next hierarchy level these two sub-clusters are separated. The inner node labeled with *#2* represents only maple leaves, whereas the other node labeled with *#3* contains hastate leaves. By traversing along the different branches the inner node is getting more and more homogeneous (e.g., similar colored leaves). Step by step different sub-clusters are divided till the lowest level of the hierarchy is reached.

5 Story Telling and Data Analysis

We defined an encompassing scheme to generate leaf glyph-based data visualizations for large data sets. We implemented the above described designs in an interactive system. We here exemplify results we obtained for analyzing the forest fire data set, showcasing the applicability of our approach. Note that a formal comparison against alternative glyph designs and user testing remain future work.

To facilitate memorizing the visual mappings we explain our design choices step by step (see Figure 9 - 12). Such a story telling approach guides the audience through a use case scenario, which analyzes complex data structures combining multi-dimensional characteristics with time-series data. Whenever possible metaphoric features are used to represent data dimensions. As studies suggest such an approach will help to better understand the underlying data.

Forest fire: The *forest fire* data set is available in the UCI machine learning repository [Cortez and Morais, 2007]. It contains data about burned areas of forests in Portugal on a daily basis for one year.

Additionally, weather information is included, e.g., temperature, humidity, rain and wind conditions at respective points in time. This data set does not contain any categorical data which could be directly mapped to the leaf shape. Therefore, we initially clustered the data points with the DBSCAN algorithm [Han et al., 2011] and assign local or global outliers to different glyph shapes (Figure 8). Our idea is to map outliers to the more jagged leaf shapes, while non-outlier points get mapped to more regular or smooth shapes, thereby providing a first visual assessment of the degree of outlyingness for the data. Our analysis task is to find similarities between burned areas to be able to predict fires due to certain weather conditions.

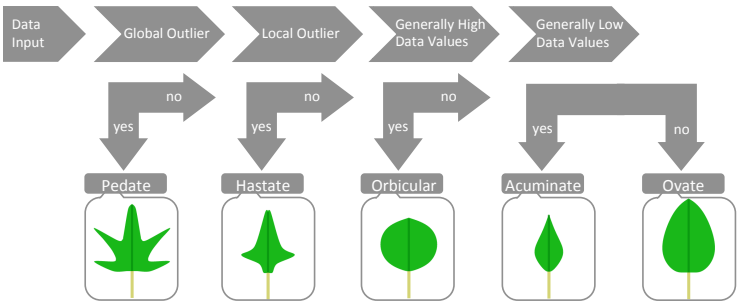

Fig. 8. Shape categories: Based on the results of the clustering we assign different leaf shape templates according to the data characteristics.

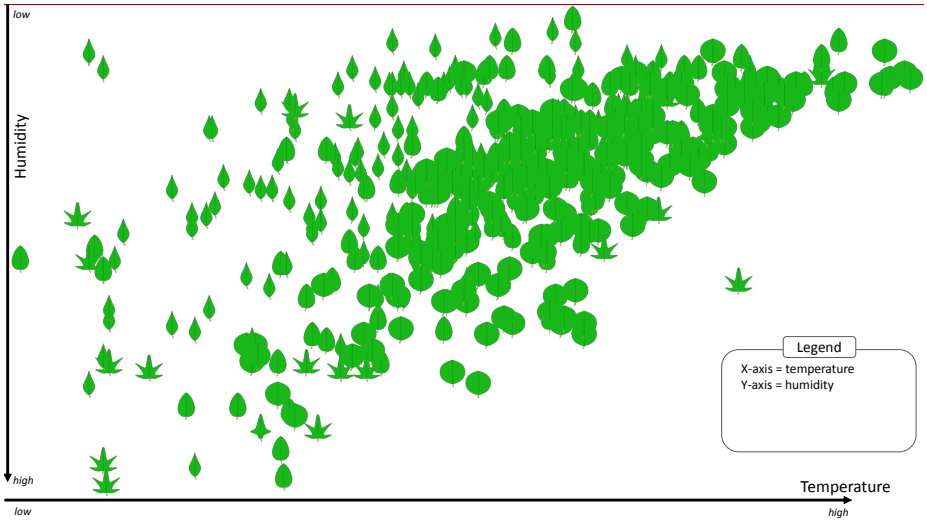

Fig. 9. Scatterplot layout: Leaf glyphs are positioned in a scatterplot according to their temperature and humidity. Since no aggregation technique is applied on the data a lot of overplotting occurs.

First, we wanted to get an idea about the data distribution. We used one data glyph for each data point and positioned the leaf glyphs in a common scatterplot layout. The x-axis is reflecting the temperature and the y-axis the humidity. By intention, we swapped the y-axis showing low data values at the top and high data values at the bottom. This reflects our background knowledge that possible indicators for forest fires are a high temperature and a low humidity. Potentially vulnerable areas are, therefore, positioned at the top right corner of the scatterplot. Figure 9 allows a first view of the data. There seems to be a positive correlation between temperature and humidity. However, because of the high number of data points, substantial information is lost due to overplotting.

As a next step, we applied transparency to the data points and also use color to show temporal information and orientation to encode the wind speed. The alpha composition technique helps to detect some more leaf shapes, however, especially in the dense area on the diagonal still a lot of overplotting exists. For the color encoding, we decided to use a metaphoric approach to help understand the encoding without a color legend. We try to associate the seasons (i.e., winter, spring, summer, autumn) with the leaves. During winter and autumn, the leaves in nature have a brownish or reddish color, whereas the color hue changes during spring and summer getting more green. Therefore, we colored our leaf glyphs accordingly. As can be seen in Figure 10 the data points are divided into 2 main clusters. Brown and red leaf glyphs are located above the diagonal and the more greener leaves are positioned on the diagonal. It seems as if humidity and temperature are both lower during autumn and winter times compared to spring or summer.

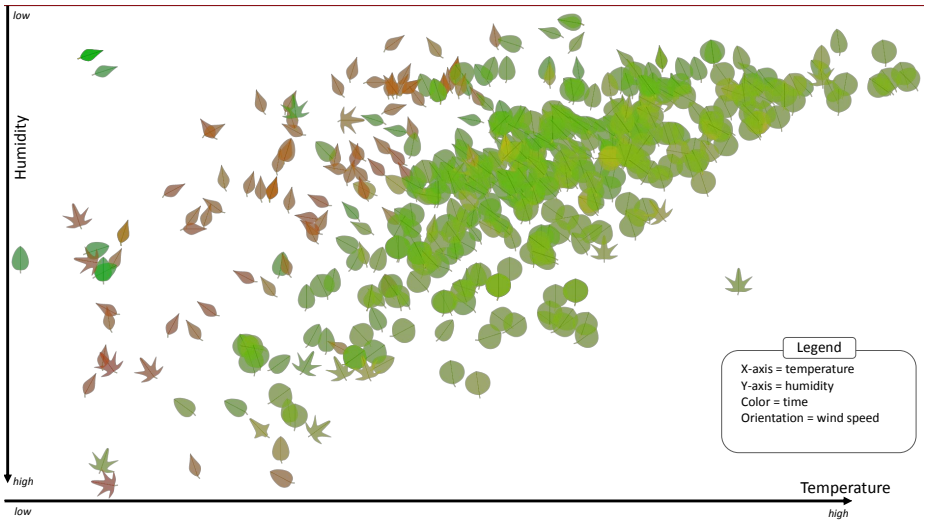

Fig. 10. Alpha Composition: Transparency is used to better perceive the data in cluttered areas. Since too many data points are located in the dense regions this aggregation technique does not provide the best view on the data.

Another metaphoric approach was used to represent the magnitude of wind. The orientation of the leaf glyphs is changing according to the wind speed. Data points with low speed are oriented to the left. With an increasing wind speed the angle changes pointing right. The idea was to simulate a blast blowing from left to right catching all leaves and changing their direction accordingly. However, no additional visual pattern can be perceived. The leaf glyphs are pointing in various directions showing no correlation between wind magnitude and temperature, humidity, or time.

To find similarities between burned forest areas, we map the size of the burned regions to the size of the glyphs. While this encoding is not strictly a metaphoric representation, it does help to associate the information with the respective visual dimension. When inspecting Figure 11, it appears all leaf glyphs are reduced in size, and differences according to size cannot be perceived. This is surprising, since we would expect the size of burned forest areas to be different. One possible explanation is that some data points with different size are located in the cluttered area on the diagonal.

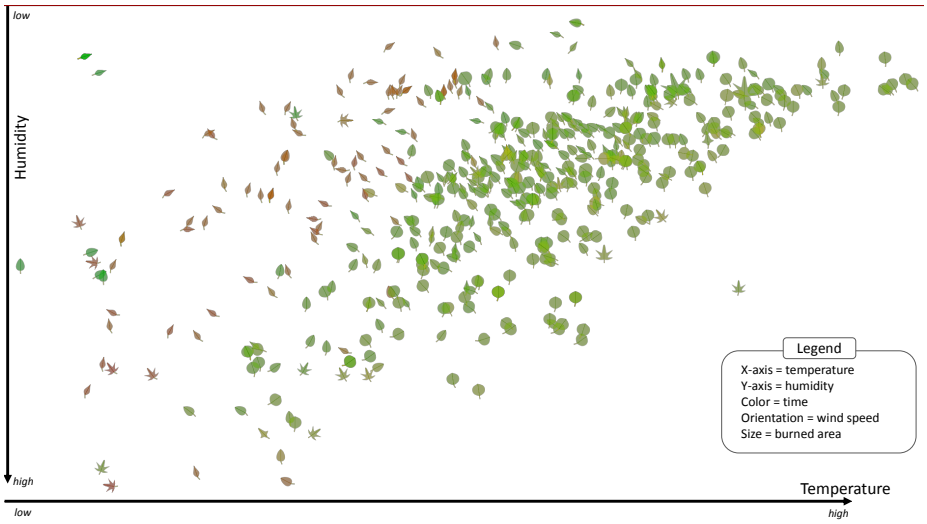

Fig. 11. Forest fire data set: We applied alpha compositing as an aggregation technique to get a first overview of the data set. We used the following mapping to represent the multi-dimensional data: Shape ≙ local/global outlier, x-position ≙ temperature, and y-position ≙ humidity, color hue/saturation ≙ time (i.e., month), size ≙ area of burned forests, orientation ≙ magnitude of wind.

To get a different perspective on the data, and to further reduce overplotting, we switch to an alternative aggregation technique to better understand the highly cluttered area (Figure 12). Due to the design of the bouquet prototype generation, the visual attribute of orientation is lost, and therefore, we cannot map the wind magnitude to this variable anymore. In the highly cluttered area in the middle of the plot, several different maple leaf shapes become apparent. These refer to outliers detected by our previous clustering algorithm. However, more interesting are the two big maple leaf shapes located at the top right corner. They represent huge areas of burned forests during the summer time with high temperature and low humidity. When switching to Figure 11, and keeping in mind the concrete location of these data points, we can further extract the wind magnitude, which seems to be medium. With this understanding of the

data, it is plausible why the burned forest areas are large. High temperature, medium winds, and low humidity all support the spread of forest fires. However, since there are more smaller data points with similar data characteristics, these features are not necessarily an indication for large forest fires. Perhaps other factors, e.g., the area or the coverage of fire stations, which are not covered in the data visualization discussed, may constitute additional factors.

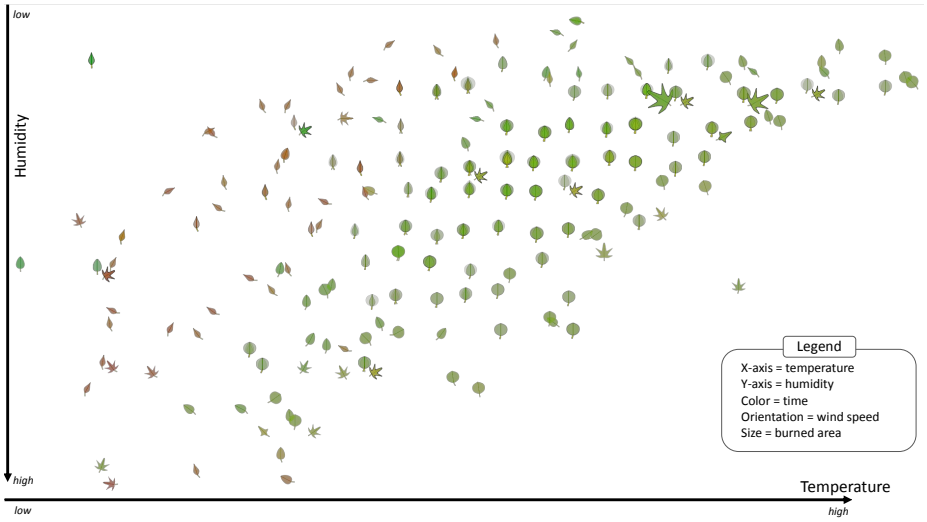

Fig. 12. Forest fire data set: We applied a prototype aggregation technique to reveal insights to the highly cluttered areas in the plot. Interesting to note are the relatively big outlier leaf shapes, which were not visible beforehand.

Of course, these findings would need to be substantiated by additional data considerations. Further information, e.g., the amount of firemen fighting the fire, the exact kind and amount of trees, or the time until the fire was recognized are important side factors not covered within the used data. However, with our new glyph approach, we were able to easily identify timely patterns, outliers, and similar behavior of data points. Other glyph designs (i.e., star glyphs etc.) might also be suitable to represent the data, however, our leaf glyph technique helps to easily associate the appearance of the data point with its attribute dimensions.

6 Conclusion and Future Work

We introduced Leaf Glyph, a novel glyph design inspired by an environmental metaphor. Due to its natural and familiar appearance, we expect users are likely to be able to discriminate data by its visual properties. The glyph is based on a naturally prominent shape, which should connect well to human perception,

supposedly also under conditions of partial overlap. We systematically structured the leaf glyph design space. Specifically, we mapped data to the main properties of the leaf glyph: Leaf morphology, leaf venation, and leaf boundary. Furthermore, we defined visual aggregations including set-oriented and hierarchical aggregation, to scale the glyph display for large numbers of data records, based on inspirations from nature. Finally, we exemplified the applicability and effectiveness of our approach in a multivariate data analysis task, showing its strengths in illustrative storytelling using a consistent metaphor.

This work is a first step in studying the effectiveness of nature-oriented data visualization. While we believe leaf glyphs can form intuitive and effective data glyphs, more thorough evaluation is needed. Specifically, we want to compare the leaf glyph against alternative glyphs from the literature, such as Chernoff faces, and pixel-oriented glyphs. This should also include user-studying of effectiveness and efficiency of the technique. We also believe our approach is aesthetically pleasing and may spark interest by a wider audience, for use, e.g., in mass media communication. The leaf glyph by design may fit well e.g., to visualization of environment survey data. Also, this should be evaluated by qualitative consideration.

As a next step, we will combine our multi-dimensional leaf glyph representation with related botanical tree metaphors to extend the design space with a hierarchical layout. A natural combination would be to pair it with the botanical tree layouts proposed in [Kleiberg et al., 2001]. We assume the combination of the two will support people with no computer science background more easily in understanding complex data structures due to the environmental reference. We further want to test this in a controlled environment against more abstract hierarchical representations such as TreeMaps.

Acknowledgments. This work has been supported by the Consensus project and has been partly funded by the European Commission's 7th Framework Programme through theme ICT-2013.5.4 ICT for Governance and Policy Modelling under contract no.611688.

References

[Beck, 2010] Beck, C. B. (2010). *An introduction to plant structure and development: plant anatomy for the twenty-first century.* Cambridge University Press.

[Borgo et al., 2012] Borgo, R., Kehrer, J., Chung, D. H., Maguire, E., Laramee, R. S., Hauser, H., Ward, M., and Chen, M. (2012). Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Proceedings of Eurographics*, pages 39–63. Eurographics.

[Chernoff, 1973] Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, pages 361–368.

[Chuah and Eick, 1998] Chuah, M. C. and Eick, S. G. (1998). Information rich glyphs for software management data. *Computer Graphics and Applications, IEEE*, 18(4):24–29.

[Cleveland and McGill, 1984] Cleveland, W. and McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, pages 531–554.

[Cortez and Morais, 2007] Cortez, P. and Morais, A. d. J. R. (2007). A data mining approach to predict forest fires using meteorological data.

[Deussen and Lintermann, 2005] Deussen, O. and Lintermann, B. (2005). *Digital design of nature*. Springer.

[Du Toit et al., 1986] Du Toit, S. H., Steyn, A. G. W., and Stumpf, R. H. (1986). *Graphical Exploratory Data Analysis*. Springer-Verlag, New York.

[Flury and Riedwyl, 1981] Flury, B. and Riedwyl, H. (1981). Graphical Representation of Multivariate Data by Means of Asymmetrical Faces. *Journal of the American Statistical Association*, 76(376):757–765.

[Fuchs et al., 2013] Fuchs, J., Fischer, F., Mansmann, F., Bertini, E., and Isenberg, P. (2013). Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting. In *Proceedings Human Factors in Computing Systems (CHI)*, pages 3237–3246. ACM.

[Han et al., 2011] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Elsevier Ltd, Oxford, 3rd edition.

[Jacob, 1978] Jacob, R. (1978). Facial Representation of Multivariate Data. In *Graphical Representation of Multivariate Data*, pages 143–168. Academic Press.

[Kintzel et al., 2011] Kintzel, C., Fuchs, J., and Mansmann, F. (2011). Monitoring Large IP Spaces with Clockview. In *Proceedings Symposium on Visualization for Cyber Security*, page 2. ACM.

[Klassen and Harrington, 1991] Klassen, R. V. and Harrington, S. J. (1991). Shadowed hedgehogs: A technique for visualizing 2d slices of 3d vector fields. In *Proceedings of the 2nd conference on Visualization'91*, pages 148–153. IEEE Computer Society Press.

[Kleiberg et al., 2001] Kleiberg, E., van de Wetering, H., and van Wijk, J. (2001). Botanical visualization of huge hierarchies. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 87–94. IEEE.

[Levkowitz and Herman, 1992] Levkowitz, H. and Herman, G. (1992). Color scales for image data. *Computer Graphics and Applications, IEEE*, 12(1):72–80.

[Müller, 2014] Müller, B. (2014). Poetry on the road. http://www.esono.com/boris/projects/poetry05/. Retrieved July 2014.

[Palmer, 1999] Palmer, S. E. (1999). *Vision science: Photons to phenomenology*, volume 1. MIT press Cambridge, MA.

[Pickett and Grinstein, 1988] Pickett, R. M. and Grinstein, G. G. (1988). Iconographic Displays for Visualizing Multidimensional Data. In *Proceedings of the Conference on Systems, Man, and Cybernetics*, volume 514, page 519. IEEE.

[Porter and Duff, 1984] Porter, T. and Duff, T. (1984). Compositing digital images. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 253–259, New York, NY, USA. ACM.

[Sallaberry et al., 2012] Sallaberry, A., Fu, Y.-C., Ho, H.-C., and Ma, K.-L. (2012). Contacttrees: Ego-centered visualization of social relations. Technical report.

[Siegel et al., 1972] Siegel, J., Farrell, E., Goldwyn, R., and Friedman, H. (1972). The Surgical Implications of Physiologic Patterns in Myocardial Infarction Shock. *Surgery*, 72(1):126.

[Stefaner, 2014a] Stefaner, M. (2014a). The deleted. http://notabilia.net/. Retrieved July 2014.

[Stefaner, 2014b] Stefaner, M. (2014b). Oecd better life index. http://moritz.stefaner.eu/projects/oecd-better-life-index/. Retrieved July 2014.

[Surtola, 2005] Surtola, H. (2005). The Effect of Data-Relatedness in Interactive Glyphs. In *Proc. IV*, pages 869–876.

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

[Ware, 2012] Ware, C. (2012). *Information Visualization: Perception for Design.* Morgan Kaufmann, Waltham.