

An Analytical Approach for the Creative Design of New Visualizations

Garth Griffin

Shaomeng Li

Connor Gramazio

Remco Chang

Department of Computer Science
Tufts University

1 INTRODUCTION

Educators have long understood the power of learning by example. Lee et al. made a compelling case for the efficacy of designing by example [3]. However, designing by example is often approached in an ad-hoc fashion. Such opportunistic tactics may overlook promising avenues that would readily be discovered using a systematic method.

Phylogenetic analysis has been successfully utilized in a number of domains for identifying previously unknown or undiscovered data points based on a corpus of example input data [1]. In light of this success, we propose a method for using phylogenetic analysis of existing visualizations to discover novel visualizations in a systematic example-based creative design process. We algorithmically map existing visualizations into a phylogenetic tree based on their similarity to one another in terms of visual attributes. We use the phylogeny to find potential novel visualizations that are related to the existing examples, and we use the visual attributes of the existing examples to compute descriptions of these potential new designs. This represents a quantitative and systematic approach for example-based visualization design. In this paper, we briefly describe our approach, present a prototype system for visualization design using this method, and demonstrate the success of our technique through a sample usage scenario of the system.

2 APPROACH

Our technique is designed to run on a corpus of example visualizations that have been tagged with verbal descriptors of the visual attributes exhibited by the examples. We call these descriptors *design features*. After consultation with a graphic design expert, we concluded that binary-valued features (e.g. "two-dimensional versus three-dimensional") would be the most comprehensible for an artist or designer. We worked with the expert to select a preliminary set of binary features.

We treat each example visualization as a point in a design-feature space. We use Hamming distance between feature vectors in this space to quantify the similarity between examples, and based on this we construct a phylogenetic tree. Each example visualization is represented by a node in the tree. During construction of the tree, additional nodes are created that do not correspond directly to examples in the dataset. We interpret these nodes as novel visualizations that are related but not identical to the existing examples. We use the structure of the tree to compute the design features of these nodes, and these computed features comprise a description of the potential visualization implied by the node. This supports a systematic example-based creative design process.

2.1 Design-Feature Space

We define a space where each dimension corresponds to one of the design features selected to describe the example visualizations. This is called the *design-feature space*. We treat each example as a point in that space with coordinates corresponding to the attributes

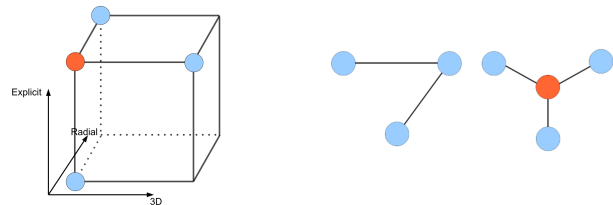


Figure 1: Left: A representation of three-dimensional binary design-feature space. The three blue nodes, representing existing examples, are equidistant by Hamming distance in the space (distance two). The orange node is a location in design-feature space that connects the three blue nodes. Middle and Right: Three equidistant nodes are joined by an internal node to improve the fitness of the phylogenetic tree. Without adding a node, it is impossible to construct a tree where the blue nodes are equidistant by edge length (one pair of nodes must always be two edges apart). By adding the orange internal node, the blue nodes can be joined in such a way that they are all equidistant by edge length (they are all two edges away from one another), which corresponds to their equidistance in the design-feature space.

it exhibits. Points in the space represent descriptions of visualizations in terms of the design features, and every unique combination of feature values corresponds to a unique point in the design-feature space.

As an example, the left of Figure 1 shows a design space for the following three dimensions: (1) whether the visualization is shown in 3D or 2D, (2) whether the orientation of the hierarchy within the visualization is axis-parallel or radial, and (3) whether the edges are described explicitly or implicitly. These binary features define a space with eight possible nodes. As more features are added, the number of possible nodes increases exponentially, and brute-force search quickly becomes infeasible. To address this, we use a phylogenetic tree to constrain the search space to nodes that are related to the existing examples.

2.2 Phylogenetic Tree

Though all points in the design-feature space are mathematically valid, some points may not correspond to meaningful visualization designs. Given the high number of combinations of feature values, it is likely that our corpus of examples will only sparsely populate the space. We use a phylogenetic tree to model the relationships between the examples. We interpret the internal nodes of the tree, which are added algorithmically based on relationships between the examples, as representing potential new visualizations. This constrains exploration to points near the existing examples.

We generate the phylogenetic tree using Neighbor Joining (NJ) [4], a well-established algorithm in bioinformatics. NJ constructs a binary tree where every leaf node corresponds to one sample point. Edge lengths and internal nodes are determined by NJ in an iterative optimization process (see Figure 1, right). The result is that nearby examples in the design-feature space will tend to be near one another in the tree. The right-hand window in Figure 2 shows the phylogenetic tree for our dataset.

2.3 Computing Attributes of Internal Nodes

Intuitively, an added internal node can be thought of as a missing link between its neighbors. We interpret these missing links as undiscovered visualizations that are related to the existing visualization examples. For any internal node, there is some location in the design-feature space that most closely fits with the position of the node in the tree. By finding this location, we obtain quantitative information about potential new visualizations in the form of design-feature space coordinates. To find the location in design-feature space for a particular internal node, we examine the distances between that node and the other nodes in the tree, and attempt to minimize discrepancy between feature distance and tree distance. We implemented a linear-time heuristic algorithm that gives a close approximation of the optimal brute-force solution.

3 PROTOTYPE SYSTEM AND USE CASE

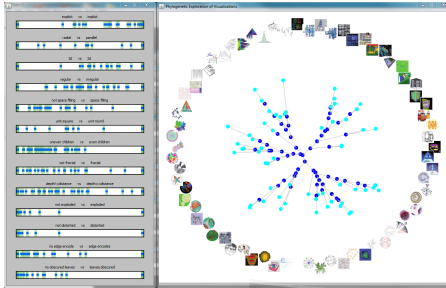


Figure 2: Screenshot of the prototype system showing both windows. The left-hand window shows the feature values of the nodes. The right-hand window shows the phylogeny of the dataset, surrounded by thumbnail images for the leaf nodes. This provides a high-level overview of the relationships between the examples, and gives a sense of the full design-feature space.

We developed a prototype system for exploration of the design-feature space. For this, we used a dataset from *A Visual Survey of Tree Visualizations* [2], a 2010 InfoVis poster containing 105 visualizations that have been separated according to various criteria. In addition to the three predominant descriptors from the poster (Figure 1, left), we further differentiated the examples with nine more design features, resulting in a twelve-dimensional design-feature space. The right-hand side of Figure 2 shows the generated phylogenetic tree as well as thumbnails of the example visualizations. Our system also shows a heat map of the design-feature values for the nodes of the tree (Figure 2, left).

From the phylogenetic tree, it is clear that the number of internal nodes in the tree is far less than the total number of unique combinations possible in our design feature space (i.e. 2^{12}). This shows how the phylogeny constrains the space to those points that are related to the existing examples. In addition, example visualizations that are near one another in the design-feature space are clustered in branches, enabling the designer to either work with an overview of the space or focus on deeper exploration of an area.

Using the tool, the designer builds a mental model of the design-feature space by examining nodes in the context of the full tree. The thumbnails provide a conceptual bridge between the verbal design-feature descriptions and the visual attributes they represent. By examining the feature-based descriptions of internal nodes in a branch, the designer can draw concrete conclusions about what potential novel visualizations might look like, and use this information to arrive at a new visualization design.

An artist used our tool to design a new visualization for a hypothetical hierarchical dataset exhibiting high branching factor near

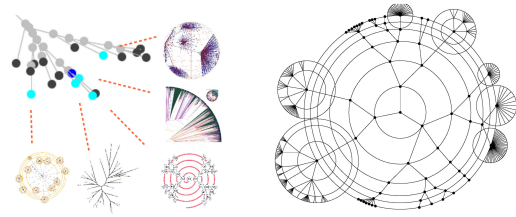


Figure 3: Left: A branch of the phylogenetic tree with five leaf nodes highlighted in cyan, with thumbnails of the corresponding visualization examples. Having selected these examples, the artist investigates the design features of a nearby internal node (blue). Right: Artist's sketch of a new visualization based on the selected examples and internal node shown at left.

its leaves. Figure 3 shows the artist's sketch of the new visualization (right) and the existing examples that the artist reported using in the design process (left). The artist described his design process as follows. First, he scanned over the leaf nodes in the phylogeny given by our system and selected a few to focus on. Next, he examined the chosen examples in more detail, and located a subtree of the phylogeny with examples exhibiting desirable attributes. Third, he chose a nearby internal node, which the artist described as the "parent" of the relevant leaves. Fourth, the artist consulted the design features the system computed for that internal node, and made a sketch based on the description implied by the attributes.

This process evidences substantial use of the internal nodes generated by the system, and indicates that the artist conceptualized internal nodes as joining multiple existing examples. This shows that the system can supply a useful conceptual framework. The artist's use of the visual attributes reported by the tool for internal nodes is evidence that having quantitative information about potential novel visualizations is advantageous in example-based design.

4 CONCLUSION

In this paper, we propose a technique for using existing examples to guide the creative process of designing a new visualization. Our technique facilitates the creative design process by providing a quantitative description of potential novel visualizations based on the examples. Though more work is needed to evaluate and expand on this, the use of our prototype system represents a proof of concept for our phylogenetic approach to systematic example-based design. This research has interesting implications for broader problems in the field of information visualizations. For example, our technique suggests possibilities for algorithmically generating visualizations using the computed values for internal nodes. The method presented here is a first step towards effectively supporting creativity with analytical example-based visualization design.

REFERENCES

- [1] M. Diaz-Banez, G. Farigu, F. Gómez, D. Rappaport, and G. Tous-saint. El compás flamenco: a phylogenetic analysis. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pages 61–70. Citeseer, 2004.
- [2] S. Jürgensmann and H. Schulz. Poster: A Visual Survey of Tree Visualization. In *Proceedings of IEEE Information Visualization*, volume 5, 2010.
- [3] B. Lee, S. Srivastava, R. Kumar, R. Brafman, and S. R. Klemmer. Designing with interactive example galleries. In *Proceedings of the 28th international conference on Human factors in computing systems, CHI '10*, pages 2257–2266, New York, NY, USA, 2010. ACM.
- [4] N. Saitou, M. Nei, et al. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425, 1987.