

# Visualizing Large Hierarchically Clustered Graphs with a Landscape Metaphor

Jan Christoph Athenstädt, Robert Görke, Marcus Krug, and Martin Nöllenburg

Institute for Theoretical Informatics, Karlsruhe Institute of Technology, Germany

## 1 Introduction

Large graphs appear in many application domains. Their analysis can be done automatically by machines, for which the graph size is less of a problem, or, especially for exploration tasks, visually by humans. The graph drawing literature contains many efficient methods for visualizing large graphs, see e.g. [4, Chapter 12], but for large graphs it is often useful to first compute a sequence of coarser and more abstract representations by grouping vertices recursively using a hierarchical clustering algorithm. Then the task is to compute an overview picture of the graph based on a given cluster hierarchy, such that details of the graph, e.g., within clusters, remain visible on demand.

This problem has been studied before. For example, Bourqui et al. [1] presented a method to draw hierarchically clustered graphs in a top-down fashion using Voronoi cells as cluster regions and a force-directed algorithm to draw each cluster. They did not explicitly consider inter-cluster forces between different levels of the hierarchy. Didimo and Montecchiani [2] recently presented a fast space-filling force-directed algorithm to compute 2D layouts of large hierarchically clustered graphs, where cluster regions are rectangles obtained from a treemap layout.

The contribution of our work is the design and implementation of a 2D layout algorithm for hierarchically clustered graphs that is specifically tailored for lifting the vertices to the surface of a 3D landscape in order to visualize various features of the graph such as average vertex degree and cluster density; such visualizations are known as landscape metaphors [3]. Our algorithm runs in two phases. The first step computes a force-directed 2D layout with circular cluster regions that is tailored to the subsequent lifting step using a Gaussian filter that matches the circular cluster shapes.

## 2 Algorithm

*2D layout.* The force-directed algorithm for computing the underlying 2D layout is based on the following four desiderata: **(1) Intra-cluster locality**, i.e., all vertices within the same cluster must be spatially close to each other. This is achieved by confining all vertices to circular cluster regions that reflect the hierarchy by containment and are otherwise disjoint from each other. The circular shape directly supports the later landscape construction with a circular Gaussian filter. **(2) Inter-cluster attraction**, i.e., the local layout of each cluster also takes into account attracting forces to adjacent vertices in other clusters. **(3) Equal cluster density**, i.e., the layout area for each cluster is proportional to the number of vertices in that cluster. **(4) Good use of layout area**,

i.e., within each cluster region the vertices should use as much of the reserved circular drawing area as possible.

The actual layout algorithm traverses the cluster hierarchy in a top-down fashion and recursively places child clusters within the available area of their parent cluster. So assume we want to draw a set of cluster nodes  $C_1, \dots, C_k$  with a joint parent cluster  $C$ . Given the radius of the parent disk, the radii  $r_1, \dots, r_k$  for the child disks are set so that areas are proportional to cluster sizes. Then a modified spring embedder is used to determine an initial layout of the cluster nodes. During the first half of the iterations only local intra-level forces (weighted by cluster sizes) are considered, including a small gravitational force to the center of  $C$ . If in any of the first half of iterations more than 10% of the node displacement vectors must be shortened in order for nodes to stay within  $C$ , the spring lengths get shortened by 10%; this adaptive shortening of initially long springs aims to achieve (4). During the second half of the iterations, each cluster  $C_i$  is additionally pulled towards clusters  $C'$  whose lowest common ancestor with  $C_i$  is the parent of  $C'$ . To stabilize the positioning, a cooling scheme is applied. Finally, before we can recurse, cluster overlaps must be removed. This is done by applying appropriate repelling forces to pairs of overlapping clusters until all cluster disks are disjoint.

*3D lifting.* Once the 2D layout is fixed, we create a surface mesh and compute the elevation of each mesh vertex based on a weighted sum of the attribute values of the graph vertices in its vicinity. A Gaussian filter determines the weight of each graph vertex so that only close vertices have a strong impact on the elevation of a particular point of the landscape. If the unweighted contribution of each vertex is 1, this produces a density map. Alternatively, we can also compute the elevation based on average attribute values, which often reveals Voronoi-like structures in the layout. This is done by dividing the weighted attribute value in each mesh vertex by the density measured at that point.

We implemented both parts of the algorithm as a plugin for Autodesk Maya<sup>®</sup>, which allows for interactive exploration of the landscape and fine-tuning of parameters such as the elevation color scheme, textures, or width of the Gaussian filters. We can observe typical 3D occlusion problems, e.g., edges passing through hills, but in fact this can also be seen as a feature in an interactive environment as it tends to separate the intra-cluster edges (visible from below) from the inter-cluster edges (visible from above).

## References

1. R. Bourqui, D. Auber, and P. Mary. How to draw clustered weighted graphs using a multilevel force-directed graph drawing algorithm. In *Proc. 11th Int'l Conf. Inform. Vis. (IV'07)*, pages 757–764. IEEE, 2007. doi:10.1109/IV.2007.65.
2. W. Didimo and F. Montecchiani. Fast layout computation of hierarchically clustered networks: Algorithmic advances and experimental analysis. In *Proc. 16th Int'l Conf. Inform. Vis. (IV'12)*, 2012. To appear.
3. S. I. Fabrikant, D. R. Montello, and D. M. Mark. The natural landscape metaphor in information visualization: The role of commonsense geomorphology. *J. Am. Soc. Inform. Sci. and Technol.*, 61(2):253–270, 2010. doi:10.1002/asi.21227.
4. R. Tamassia, editor. *Handbook of Graph Drawing and Visualization*. CRC Press, 2012. To appear.