Density Functions for Visual Attributes and Effective Partitioning in Graph Visualization

Ivan Herman M. Scott Marshall Guy Melançon*

Centre for Mathematics and Computer Sciences (CWI)

Abstract

Two tasks in Graph Visualization require partitioning: the assignment of visual attributes and divisive clustering. Often, we would like to assign a color or other visual attributes to a node or edge that indicates an associated value. In an application involving divisive clustering, we would like to partition the graph into subsets of graph elements based on metric values in such a way that all subsets are evenly populated. Assuming a uniform distribution of metric values during either partitioning or coloring can have undesired effects such as empty clusters or only one level of emphasis for the entire graph. Probability density functions derived from statistics about a metric can help systems succeed at these tasks.

CR Categories and Subject Descriptors: I.3.6 [Computer Graphics]: Methodology and Techniques – Interaction Techniques; I.3.8 [Computer Graphics]: Applications

Additional Keywords: graph visualization, graph navigation, metrics, clustering

1. INTRODUCTION

A key issue when visualizing graphs in information visualization is the *size* of the data. Many applications of graph visualization require analysis of graphs with several thousand nodes and edges. Innovative techniques are needed to navigate, to filter, or to create abstractions from these graphs in order to make them usable in practice. Many interesting results have been published in the past few years and this area of research is still very active (see, for example, the survey on graph navigation[1]).

The use of metrics is one of the interesting techniques in this area. The concept of a *metric* appears in several places in the literature[2-6], although the terminology varies. In this paper, we will use the term to refer to *a measure that is associated with a node or an edge in the graph.* The measure can be application-specific, can be the result of some function (usually combinatorial) of the graph structure, or a combination of these. A few examples of metrics based on graph structure are the degree of a node (i.e., the number of edges adjacent to the node), the size of a subtree for a tree, or the measure of the flow of information in a directed graph. In general, the goal is to define the relative importance of a node or an edge with respect to some semantics, where elements with high metric values are considered more interesting than those with low values.

Metric values that are associated with nodes and edges can be used to determine visual attributes such as color and saturation in order to emphasize differences among elements. A technique that we find useful renders an edge with continuously shaded color that reflects the metric values of the nodes at its endpoints. In this approach, higher metric values are considered more interesting and are assigned higher saturation values for emphasis. The overall effect is the emphasis of edges in the graph with the “most interesting” metric values. This design of graphical attributes based on metrics has already been discussed in [5, 6]. For example, Figure 1 shows this effect when zooming into the details of a graph; the darker and thicker lines help to navigate towards more complex areas of the graph (this particular example uses the Strahler metric, described in Herman *et al.* [5]).

Another use of metrics is the generation of fisheye views, as presented in the seminal paper of Furnas[2][1], where he computes the “degree of interest” of elements in a tree. Elements with low values are hidden to improve the display of the structure (sometimes referred to as *semantic fisheye*) and help emphasize the more important elements in the tree.

Generating such visual cues is not the only way to use metrics. In a type of *divisive clustering*, data sets are partitioned according to metric values, with the metric value determining group membership. This subdivision helps the user to partition the graph into subgraphs of manageable sizes. This technique is not only vital to navigation in large graphs but also helps the user to identify important relations among elements, thereby making the information visualization application much more effective (see Section 5 for an example). Such subdivision procedures become particularly important if the underlying graph structure is not a tree, where no “natural” subdivision (i.e., subtrees) exists.

It is important to note that all these techniques are automatic, in the sense that no further user input is necessary to generate the visual attributes or the clusters. A straightforward approach is to apply a simple linear mapping from the metric values to, for example, color saturation. This approach can work well when there is a uniform distribution of metric values. However, experience shows that more control over this mapping is necessary for cases

*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Email: {I.Herman, M.S.Marshall, G.Melancon}@cwi.nl

[1] Furnas used the term “degree of interest” but, in our terminology, his DOI function could be considered a type of metric.

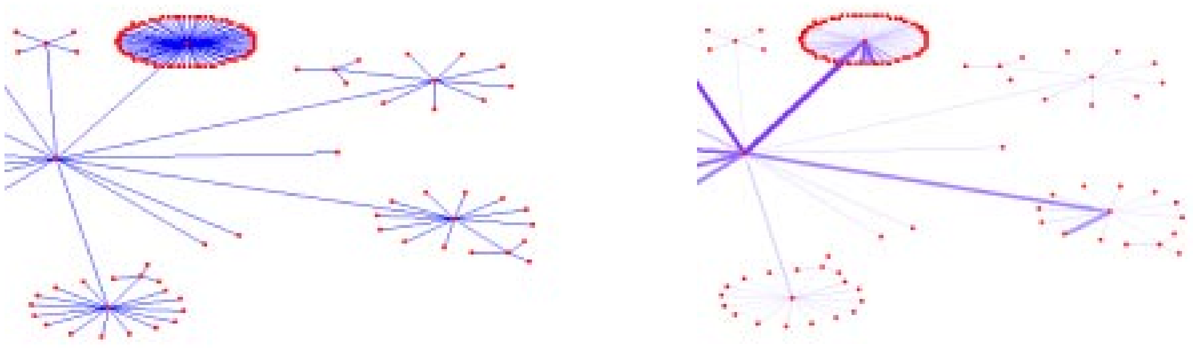


Figure 1. Effects of emphasis mapping on a tree. The right side shows the same portion of a tree but with edges emphasized through metric values.

where the distribution is not uniform. Essentially, the spread of the metric values over the full interval should be taken into account: a finer subdivision might be necessary in more densely populated areas when doing clustering or assigning colors. Mathematically, this means that the statistical behavior of the metric values should be taken into consideration in the mapping. Presenting this approach is the focus of this paper.

The examples in this paper use directed acyclic graphs (DAGs). DAGs form an intermediary class between trees and general graphs: efficient methods exist which first extract a DAG from a directed graph as a pre-processing step (see, for example, the book of Battista *et al.*[7]), which makes them generally useful in information visualization. The general methodology presented here is not restricted to DAGs, although some of the details may have to be investigated for the general case.

The rest of the paper is organized as follows. In Sections 2 and 3, we discuss visual attribute mapping, generalizing the method illustrated by Figure 1. The same methodology can be reused for hierarchical clustering; Section 4 will present the details. A detailed example is shown in Section 5, followed by conclusions and directions for future research in Section 6.

2. ASSIGNING COLORS USING METRIC DENSITY

Assigning a visual attribute (e.g., color, brightness, color saturation, or line width) consists of two steps:

1. Assign an abstract value, usually between zero and one, to each displayable element based on the element's metric value. We will refer to this abstract value as the *emphasis* of the element, and we will also refer to this mapping as the *emphasis mapping*.

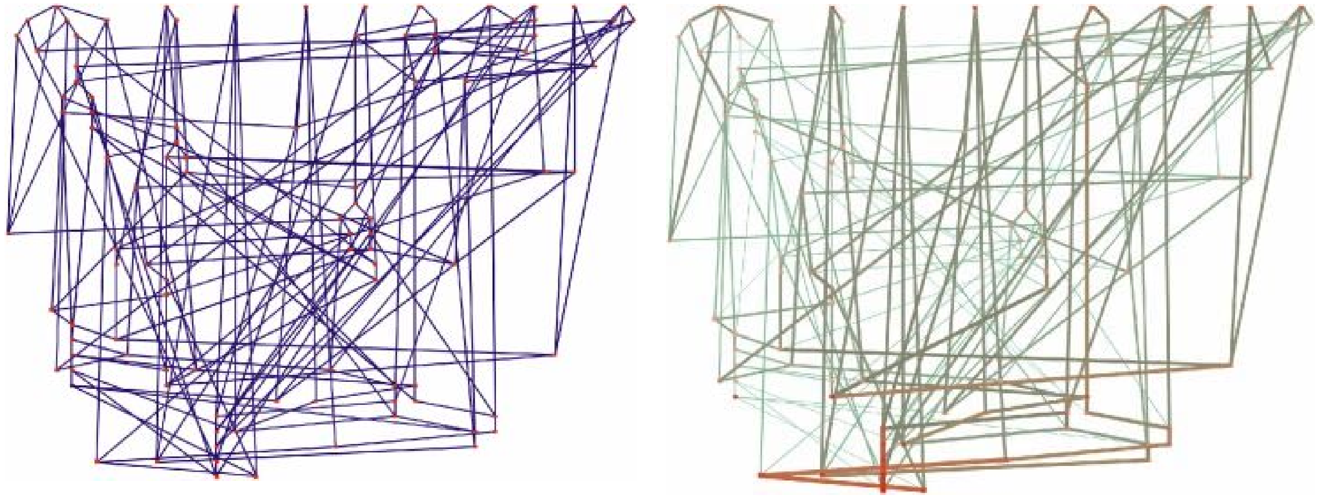
2. Map the emphasis to a visual attribute. We will refer to this mapping as the *attribute mapping*.

The two mappings have different characteristics. It is therefore important to conceptually separate them. The mapping that generates the final visual attributes, is closely related to issues of perception and cognition, lighting conditions, display gamma values, and underlying graphics systems (see Ware's book[8], for example). In some cases, a simple linear mapping from an emphasis value to, for example, color saturation is acceptable. In other cases, a non-linear mapping is necessary. In our view, a visualization system should give the end-user some means of controlling the mapping used in order to adjust for his/her viewing conditions. The technique used to produce the images in this paper involves mapping emphasis values to three visual attributes: color, saturation, and line width. We interpolate between two colors as well as between low and high saturation based on the emphasis value. Similarly, high emphasis values map to thicker line widths. This paper, however, focuses on the emphasis mapping and *not* on the attribute mapping, which would require a separate investigation of its own.

An obvious approach to emphasis mapping is to apply a simple linear function: the metric values for a specific graph are normalized and linearly mapped onto the unit interval. More precisely, the emphasis value associated with a metric value x is computed by $(x-m)/(M-m)$, where m and M are the minimum and maximum metric values in the graph. This is the mapping used to generate the images in Figure 1 which shows that, at least in some specific cases, this mapping works well. But this is not always the case. Figure 2/(a) shows the image of a DAG. We can use what we call the *flow metric* to emphasize important parts of the graph. This metric, introduced in [6], simulates the flow of information in a DAG using concepts similar to Kirchoff's equations for electrical current.[2] However, if we use a linear emphasis mapping, the result is a practically blank image, because most of the edges have very low color saturation. It should be noted that this is *not* the result of artifacts in the attribute mapping: modifying the distortions of the attribute mapping will not improve the picture, just make *all* edges uniformly darker, for example.

The reason is that the linear mapping does not take into consideration how the metric values are spread over the available interval. If the distribution is uniform, the linear mapping works fine. The values of the Strahler metric on trees, for example, are "almost" uniformly distributed and this is why the example on Figure 1 works well. However, the flow metric produces a relatively large number of low values. If a linear mapping is used with the flow metric, most of the metric values yield a low emphasis value, which leads to low visual attribute values.

[2] Another useful analogy for this metric is the flow of water if poured into the entry nodes at the top of the graph.



(a) Graph without emphasis mapping

(b) Graph with emphasis mapping: highlighted edges are the "important" edges with regard to flow metrics

Figure 2. Two views of the same graph. Right side uses emphasis mapping.

A more precise formulation is that the linear mapping does not take the *distribution* of the metric values into account. Using a linear mapping is equivalent to assuming that the distribution of values is uniform. In probabilistic terms, the distribution function *f* can be used to calculate the probability that a given metric value *x* is associated with an element in the graph. This is true if the set of possible metric values is discrete. When the set of possible values is infinite and fully covers an interval of real numbers, it only makes sense to ask for the probability that a metric value lies in a given sub-interval *[a, b]*. In this case, the probability can be obtained by summing up probabilities using the integral:

$$\int_a^b f(x)dx$$

If all values have equal probability of appearing, then the distribution is uniform and appears as a simple horizontal line above the set of possible values. (*f(x)* is zero outside this domain). If the distribution is not uniform, some intervals will have a higher probability than others and the curve will be more complex. For technical reasons, the *density* function is often used rather than the distribution:

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

The density function also has an intuitive meaning: it gives the probability that a value is smaller than a specific value *x*. Since the density function is monotonically non-decreasing, and it is, in most cases, one-to-one, it can be inverted over the domain of metric values[3]. This property is essential, as we shall see in Section 4. It is easy to show that the density function obtained from a uniform distribution is piecewise linear and is given by:

$$F(x) = \begin{cases} 0 & x < m \\ \dfrac{x-m}{M-m} & m \le x \le M \\ 1 & M < x \end{cases}$$

Note that this coincides with the expression used for linear emphasis.

By definition, the curvature of the density function reflects the "uneven" spread of values that we want to characterize. Figure 3 shows an approximation of the density function for the flow metric (see Section 3 for details on how this density function can be derived): it shows that for relatively low values the function increases sharply, which indicates that it is highly probable that metric values will be concentrated in this area. This is exactly the information that we need to perform the emphasis mapping. This leads to the main message of this paper: *in order to effectively use metric values, statistical knowledge about a metric should be applied when mapping to an emphasis value.* In particular, a metric value should be evaluated by the density function for that metric, and the result of this evaluation should be used as an emphasis value.

Figure 2/(b) shows the result of using the same metric and attribute mapping, but with the density function as the emphasis mapping. Although the graph is relatively small, the important nodes and edges are picked up by the image, showing where most of the information flows through the network.

This improved emphasis mapping leads to a powerful set of navigation tools. For instance, it is easy to choose the direction for panning in a zoomed graph (see Figure 1). It is also possible to make an abstraction of the image by hiding elements with lower metric values. Some of these techniques have been investigated before[4, 6], but the use of a density function is essential to exploit these techniques to their fullest potential.

[3] If the function is not one-to-one, i.e., it has constant values on some intervals, the inverse can be deduced using some simple heuristics on those intervals.

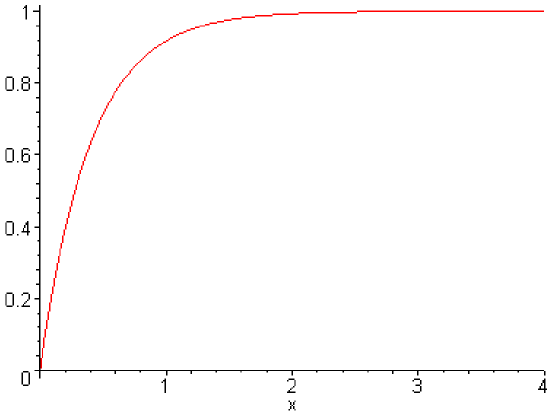


Figure 3. Density function for the flow metric.

3. GATHERING STATISTICS FOR A METRIC

The methodology described in Section 2 is based on the knowledge of the density function for a specific metric. However, finding an analytical description of the density function for a given family of graphs is usually a difficult mathematical problem, although some results are already available. As an example, the distribution of the width of a (sub)tree (i.e., the number of leaves in a tree) can be approximated by a normal distribution (see Drmota[9]; see also Herman *et al.*[5] where this result has already been exploited for a simple version of visual clustering). Metrics that are related to combinatorics may have already been investigated by the mathematics community (this is the case for the Strahler metric[10], for example). Unfortunately, this is not often the case, and the density function must be constructed from empirical measurements.

When no theoretical density function is available, one can calculate an approximation based on the metric values in the graph under investigation. It is possible to build an approximation of the probability distribution by computing the frequency histogram of metric values. Once normalized, this histogram gives the discrete form of the distribution. By accumulating the frequencies along the range of values, we produce the discrete version of the density function, yielding what we call the "local density function". The cumulative histogram is then stored as a look-up table for the graph, and this table is used to calculate the density function for a specific metric value.

If both theoretical and local densities are available, which one should we use? The choice between theoretical or local densities depends on the task we wish to carry out. If the goal is only to look at a specific graph, then the local density is probably the most appropriate. It provides more accurate information about the distribution of values for *that* particular graph; the local distribution of the values may indeed differ from the theoretical distribution. However, if the graph is used as part of the exploration of a dataset, or if two graphs representing similar datasets are to be compared, then the theoretical density should be used. When used in this way, the theoretical density serves as a common reference, which is valuable for tasks involving comparison.

It is important to note that metrics can be associated with either nodes or edges, and sometimes with both. The distribution for nodes or edges, and thus their density functions, should be treated separately, because they may not be identical. This is the case for the flow metric, for example: although we can assign a flow value for both nodes and edges, their density functions greatly differ.

There is yet another approach that can help to get an approximation of the theoretical density, without relying solely on the local density. Suppose you have a large set of graphs, all belonging to the same family. Their local density functions can be merged, and used to infer what could be the density function for metric values on that family. This approach is usable if one can randomly generate graphs for that family: merging the local densities on a larger sample of graphs can lead to a better approximation of the theoretical density. This is the method we used, based on a tool which one of the authors has developed to generate random DAGs [11]. Random DAGs can be generated in large quantities by this tool, by controlling some structural features of the graphs (e.g., maximal degree of nodes). With a large number of random graphs at our disposal, we collected and merged the local densities for several metrics. As a last step, either look-up tables were generated for the density functions or, if possible, a suitable analytical approximation was found. It is worth noting that it was possible to define such an analytical approximation for most of the metrics that we examined. As an example, the density function of the flow metric for edges could be approximated by $F(x) = 1 - e^{-Cx}$, where *C* is a suitable constant depending on the ratio of edges and nodes in the graph (see Figure 3).

4. PARTITIONING THE DATASET

Mapping metric values to colors brings with it several issues of cognition and perception. When carried out properly, it allows the user to get an idea of the spectrum of metric values but even at its best the number of perceptible values is small (there are numerous studies on this subject, see again Ware's recent book[8], for example). This is especially true for a dense graph. The cognitive limits are often aggravated by the limitation of the physical display and lighting conditions.

An abstract view of the graph could be used as an alternative to color-based navigation. One way of creating such an abstract view is to use clusters. A fundamental technique in graph visualization represents the groups, or clusters, of a graph using a special type of node called a *meta-node*. This technique makes it possible to represent a large graph by displaying fewer elements, allowing the user to control the level of detail by "opening" and "closing" meta-nodes (see, for example, Eades and Feng[12] or Schaffer *et al.*[13] for two possible approaches to clustered graphs). To create a clustered view of a graph, we first divide the data into clusters, with each cluster representing a particular sub-interval or range of metric values. This process is identical to that for mapping to colors. A node is added to a cluster if its metric value is within the range defined for that cluster. This is a general approach: as long as you can map the attributes of interest into a metric value, this technique will succeed in producing groups based on the semantics implicit in the metric. Furthermore, this technique is automatic, i.e., it does not require additional user input for the creation of the clusters.

When the partitioning process is repeated on each of the clusters from the previous step, it is called *hierarchical clustering*, and a separate tree, or *overview diagram*, can represent the result. In this tree, the top node represents the graph before partitioning and children represent clusters resulting from partitioning their

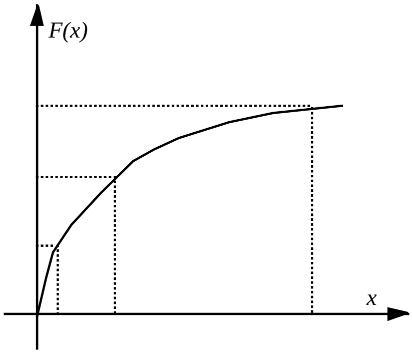


Figure 4. Using the inverse density function for partitioning.

parents. The overview diagram can be used to navigate the original graph (see Figure 7). Rich interaction facilities can be devised using this overview graph (see, for example, the papers [14, 15] [16] on possible interaction techniques with overview diagrams). Of course, partitioning can also be performed based on edge metrics.

It is important to note that when applying hierarchical clustering to create an overview diagram, the goal is different from that of a typical clustering process. Most clustering processes are meant to find clusters or classes in the data. Although classes may be discovered as a result of using a particular metric, this application of hierarchical clustering is primarily meant to divide the data into manageable chunks and provide a map into the data based on ranges of metric values. Empty clusters and clusters with large populations are therefore undesirable because they may either provide the user with too little or too much visual information. Although it isn't always possible with a given dataset, the goal is an even distribution of population among the clusters.

If we don't apply knowledge about the distribution of values, we might sort values and dynamically adjust the boundaries to achieve the desired populations. However, this technique is too expensive for large graphs. Alternatively, we can divide the range of values into sub-intervals of equal lengths, but this leads to the same problems as for emphasis mapping: some clusters may end up with too many elements and others may end up too small. A partitioning with even cluster populations can be accomplished by using the density function of the metric on which the partitioning process is based.

The density function can be used as follows (see Figure 4). The *[0,1]* density interval is divided into equal subintervals (three in our example). By applying the inverse of the density function, these intervals give us the required upper and lower boundaries on the *x*-axis. (This inverse mapping is usually described by following the dotted lines going from the *y*-axis back to the *x*-axis.) These intervals will then be used to classify elements, based on their metric values. What happens here is a quantization of the emphasis mapping used for visual attributes. The advantage of this hierarchical subdivision is that it can be done recursively for any of the subintervals, either automatically or as a result of user interaction.

In the automatic case, the stop condition for the subdivision can be expressed in terms of the density function. The stop condition we use is:

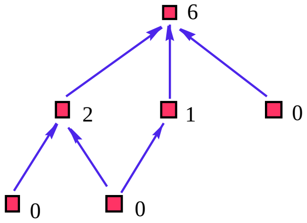


Figure 5. Metric definition for a citation graph. Nodes "higher" in the hierarchy represent papers with an earlier publication date.

$$\max_{a,b\in C}\left|F(a)-F(b)\right|<\varepsilon$$

where *C* is a cluster and ε is a suitably small number. The formula can be viewed as expressing a density-dependent size of a cluster, i.e., the stop condition halts the process when the density-dependent size of the cluster becomes small.

5. AN EXAMPLE APPLICATION

Two of our colleagues have collected the bibliography entries, as well as the cross-references, of all the articles published in the IEEE Visualization proceedings, starting from 1990 until 1999. A directed graph can be constructed from this dataset: a link from node A to node B represents a citation, i.e., the paper represented by A has a reference to the paper represented by B. This graph can be conveniently displayed by placing all nodes for a specific year at the same horizontal position, with the nodes for 1990 on the top row and those of 1999 on the bottom row. The resulting graph has edges directed upwards. Note also that this graph is acyclic. (With the rare exception of papers published the same year and referencing each other. This only occurred in two cases and we decided to simply ignore one of the two references).

The dataset is fairly large: around 600 nodes and 900 edges. If the full graph were displayed, it would lead to a uniform "cloud" of edges, with no way to discern any detail. This is why we refrained from including the complete image of the graph here.

To use the methods described earlier in this paper, a metric has to be defined. This metric has to reflect the kind of investigation one intends to pursue with the graph. What we set out to look for was the "influence" of papers on the series of the IEEE Visualization conferences measured through the number of direct or indirect citations. The metric we use is therefore as follows: for each node, we calculate the number of edges (citations) that, directly or indirectly, refer to that paper (see Figure 5 for a small example). This value is equal to the number of edges one can reach from a node by going backwards in the graph (i.e., "back in time"). The higher this value, the more "influential" the paper was (at least, in a citation index). Since there is no specific edge metric in this case, the emphasis value assigned to an edge is the minimum of the emphasis on the two end nodes.

Using a linear emphasis mapping for this metric is unsatisfactory, due to the problems described earlier. Therefore, we used a density-based emphasis mapping. We approximated the theoretical density function using the random DAG generation tool and we built a look-up table. (No satisfactory analytical approximation

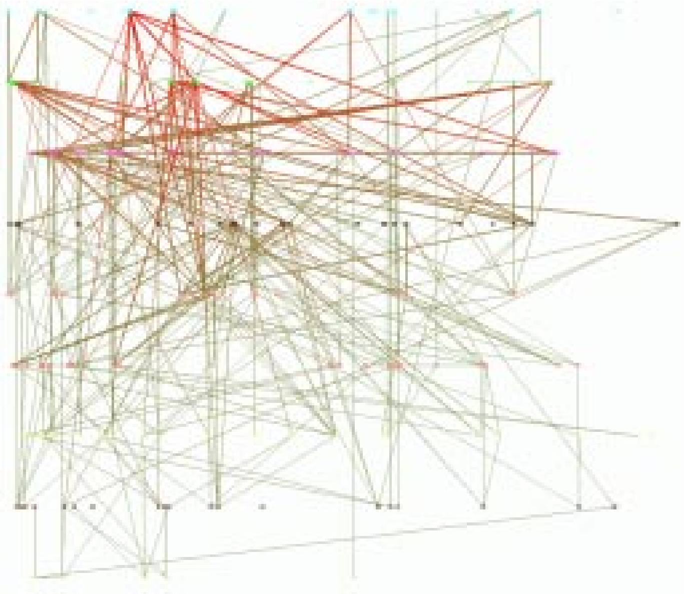


Figure 6. Density-based metric coloring of the citation data set produces a *skeleton*. Only elements with an emphasis greater than 0.33 are displayed.

could be found in this case). The use of the density-based emphasis mapping leads to Figure 6. For a better overview, only elements with an emphasis greater than 0.33 are displayed: a simple slider-based interaction has been implemented to discard low emphasis elements.

Figure 6 is certainly an improvement compared to a full image of the graph, because only the important nodes and edges (the "skeleton") are visible. However, it is still difficult to differentiate among the remaining elements. This is apparently due to the difficulty of comprehending relations between large numbers of visual attributes such as different shades of color. Zooming into a smaller area does not help.

As the next step in the investigation, a hierarchical partitioning was done on the full graph, in the manner described in Section 4. A zoomed-in view of the generated overview tree is shown on Figure 7/(b). Each node in the overview tree corresponds to a subinterval of density values. We can now use various interactive techniques on this overview graph (see [15] for more details on these techniques). For example, Figure 7/(a) shows a filtered view of the original graph with a number of clusters selected for emphasis. (The framed nodes in the overview tree are the selected clusters. Cluster members are emphasized in Figure 7/(a).) Using the overview tree instead of a slider gives the user more control over the selection process. In the figure, we selected nodes in the tree in order to highlight all nodes in the graph with emphasis value greater than 0.55. Edges and nodes not belonging to selected clusters provide a background context (in light gray).

This view shows a selection of the most "influential" papers[4]. Of course, this example should not be taken too seriously because the influence of a paper should not be measured solely based on IEEE Visualization publications (for example, the data set does not include the InfoVis symposium!). Furthermore, the number of citations is not necessarily the best measure of influence. This example is used merely as an illustration of the techniques developed in this paper. Nevertheless, even this simple example shows the power of using density functions and derived visualization, navigation, and clustering techniques in exploring large graphs[5].

6. CONCLUSIONS AND FUTURE WORK

Density-based emphasis mapping, whether based on a local or a theoretical density, is clearly necessary for a better separation of elements in a graph, if a metric is used to control navigation. Densities reflect the range and spread of values within a graph; disregarding them could not only lead to unsatisfactory interaction tools but, possibly, to erroneous conclusions drawn on the basis of the data represented by the graph.

Different classes of graphs can have different behaviors with respect to a particular metric. If such classes have been identified, care should be taken to gather separate statistics for each class. For a given application, careful analysis of the data might reveal that the graphs under investigation share some properties. For example, their node degrees might be bound by some value or certain types of connections among nodes might be missing or improbable. Those properties can then be used to infer the class to which a graph belongs. Because each class has its own density function, class membership determines the density function to apply to the graph. As an example, we found that it was not appropriate to only have one approximation for the density of the flow metric. Instead, a parameterized family of functions is used (see Section 3). A parameter value (denoted by C in the equation) based on the average number of edges per node is used in the function. In other words, this property is used to first classify the graph and yield a good approximation for the density on that graph.

Obviously, it would be advantageous for information visualization if the exact statistical behavior of the applied metrics were

[4] The two "top" papers selected by this process were [17] and [18]

[5] Our graph visualization framework (in Java) and an application built using it are available at http://www.cwi.nl/InfoVisu/GVF.

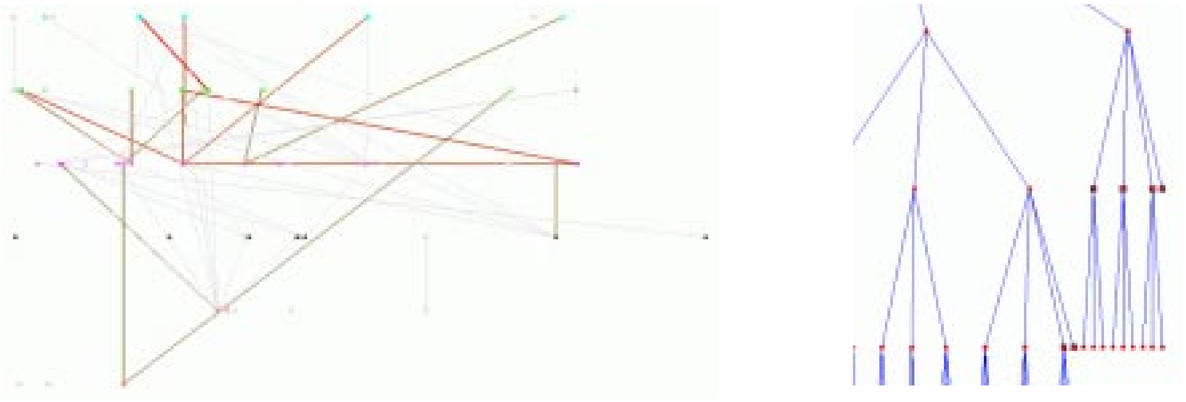


Figure 7. Clusters selected in the overview tree (right) are highlighted in the citation graph (left). The overview graph is generated by the hierarchical clustering described in Section 4, using the metric defined in Section 5 for citation graphs.

known. This would save us the process of finding an approximating function or building a look-up table. Unfortunately, some of these behavioral descriptions are very involved, and require a strong foundation in mathematics. Pursuit of this sort of knowledge is a potential source of fruitful cooperation between the visualization and the mathematics communities.

The use of the density function for emphasis mapping has been chosen with the assumption that the important nodes are those with higher metric values. The citation example in Section 5 focused on finding the most influential papers, but one could have used the same tools to find, for example, the least influential papers or the average papers. In general, depending on the application, other combinations are possible to slightly modify the emphasis mapping itself. For example, one could take the value of $F(X)-F(A)$, where *A* is the metric value of a fixed node, and normalize this to *[0,1]* to yield the emphasis value. This mapping would highlight the "distance" of a node in terms of the metric, compared to a fixed one. Alternatively, one could take the value of $F(X)-F(\mu)$, where μ denotes the mean of the distribution. Both of these variants can be considered to be generalizations of the DOI function of Furnas[2], except that the density values are used instead of the direct differences in the metric values.

ACKNOWLEDGEMENTS

We are grateful to Robert van Liere and Wim de Leeuw, both at CWI, who provided us with the citation dataset that they collected. We are also grateful to Maylis Delest (Université Bordeaux I, France) for fruitful discussions at the early stage of this work.

REFERENCES

1. I. Herman, M. S. Marshall, and G. Melançon, "Graph Visualisation and Navigation in Information Visualisation: A Survey", *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 24-43, 2000.

2. G. W. Furnas, "Generalized Fisheye Views", in: *Proceedings of Human Factors in Computing Systems CHI '86*, ACM Press, pp. 16-23, 1986.

3. R. A. Botafogo, E. Rivlin, and B. Schneiderman, "Structural Analysis of Hypertexts: Identifying Hierarchies and useful Metrics", *ACM Transactions on Information Systems*, vol. 10, pp. 142-180, 1992.

4. D. Kimelman, B. Leban, T. Roth, and D. Zernik, "Reduction of Visual Complexity in Dynamic Graphs", in: *Proceedings of 2nd International Symposium on Graph Drawing*, Princeton, New Jersey, USA, Springer, pp. 218-225, 1994.

5. I. Herman, G. Melançon, and M. Delest, "Tree Visualisation and Navigation Clues for Information Visualisation", *Computer Graphics Forum*, vol. 17, pp. 153–165, 1998.

6. I. Herman, M. S. Marshall, G. Melançon, D. J. Duke, M. Delest, and J.-P. Domenger, "Skeletal Images as Visual Cues in Graph Visualization", in: *Data Visualization '99, Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, E. Gröller, H. Löffelmann, and W. Ribarsky, Eds., Springer–Verlag, pp. 13–22, 1999.

7. G. d. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualisation of Graphs*. Prentice Hall, 1999.

8. C. Ware, *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.

9. M. Drmota, "Systems of functional equations", *Journal of Random Structures and Algorithms*, vol. 10, pp. 103-124, 1997.

10. P. Flajolet, J. C. Raoult, and J. Vuillemin, "The number of registers required for evaluating arithmetic expressions", *Theoretical Computer Science*, vol. 9, pp. 99-125, 1979.

11. G. Melançon, I. Dutour, and M. Bousquet-Melou, "Random generation of DAGs for graph drawing", Centre for Mathematics and Computer Sciences, Amsterdam INS-R0005, 2000, ftp://ftp.cwi.nl/pub/CWIreports/INS/INS-R0005.pdf.

12. P. Eades and Q.-W. Feng, "Multilevel Visualization of Clustered Graphs", in: *Proceedings of Symposium on Graph Drawing GD '96*, Berlin, Springer-Verlag, pp. 101-112, 1997.

13. D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman, “Navigating Hierarchically Clustered Networks through Fisheye and Full–Zoom Methods”, *ACM Transactions on Computer–Human Interaction*, vol. 3, pp. 162–188, 1996.

14. Y. H. Fua, M. O. Ward, and E. A. Rundensteiner, “Navigating Hierarchies with Structure–Based Brushes”, in: *Proceedings of IEEE Symposium on Information Visualization (InfoVis '99)*, IEEE CS Press, pp. 58–64, 1999.

15. I. Herman, M. S. Marshall, and G. Melançon, “Automatic Generation of Interactive Overview Diagrams for the Navigation of Large Graphs”, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Technical Report INS–R0014, http://www.cwi.nl/InfoVisu/papers/Hierarchical.pdf, 2000.

16. R. M. Wilson and R. D. Bergeron, “Dynamic Hierarchy Specification and Visualization”, in: *Proceedings of IEEE Symposium on Information Visualization (InfoVis '99)*, IEEE CS Press, pp. 65–72, 1999.

17. G. V. Bancroft, F. J. Merrit, T. C. Plessel, P. G. Kelaita, R. K. McCabe, and A. Globus, “FAST: A Multi-Processed environment for Visualization of Computational Fluid Dynamics”, in: *Proceedings of IEEE Visualization '90*, Palo Alto, CA., IEEE CS Press, pp. 14-27, 1990.

18. A. Globus, C. Levit, and T. Lasinski, “A Tool for Visualizing the Topology of Three-Dimensional Vector Fields”, in: *Proceedings of IEEE Visualization '91*, Palo Alto, CA, IEEE CS Press, pp. 33-41, 1991.