

Difference Metrics for Interactive Orthogonal Graph Drawing Algorithms*

Stina Bridgeman and Roberto Tamassia

Department of Computer Science
Brown University
Providence, Rhode Island 02912-1910
{`ssb,rt`}@`cs.brown.edu`

Abstract. Preserving the “mental map” is major goal of interactive graph drawing algorithms. Several models have been proposed for formalizing the notion of mental map. Additional work needs to be done to formulate and validate “difference” metrics which can be used in practice. This paper introduces a framework for defining and validating metrics to measure the difference between two drawings of the same graph.

1 Introduction

Graph drawing algorithms have traditionally been developed using a batch model, where the graph is redrawn from scratch every time a drawing is desired. These algorithms, however, are not well suited for interactive applications, where the user repeatedly makes modifications to the graph and requests a new drawing. When the graph is redrawn it is important to preserve the look (the user’s “mental map”) of the original drawing as much as possible, so the user does not need to spend a lot of time relearning the graph.

The problems of incremental graph drawing, where vertices are added one at a time, and the more general case of interactive graph drawing, where any combination of vertex/edge deletion and insertion is allowed at each step, have been starting to receive more attention. See, for example, [2, 3, 4, 6, 9, 14, 16, 17, 18, 19, 22]. However, while the algorithms themselves have been motivated by the need to preserve the user’s mental map, much of the evaluation of the algorithms has so far focused on traditional optimization criteria such as the area and the number of bends and crossings (for example, [2, 9, 18, 19]). Mental map preservation is often achieved by attempting to minimize the change between drawings — typically by allowing only very limited modifications (if any) to the position of vertices and edge bends in the existing drawing — making it important to be able to measure precisely how much the look of the drawing changes. Animation can be used to provide a smooth transition between the drawings and can help compensate for greater changes in the drawing, though it is still important to limit, if not minimize, the difference between the drawings

* Research supported in part by the U.S. Army Research Office under grant DAAH04-96-1-0013, by the National Science Foundation under grants CCR-9732327 and CDA-9703080, and by a National Science Foundation Graduate Fellowship.

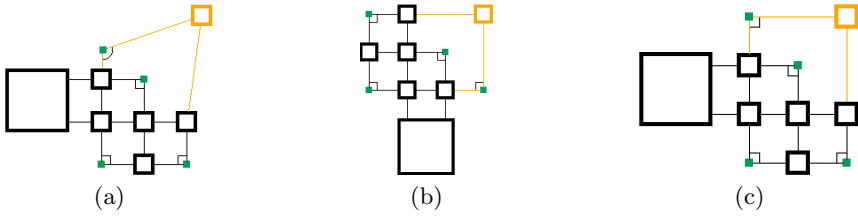


Fig. 1. The rotation problem. (a) is the user-modified graph (the user’s changes are shown in gray); (b) and (c) are the unrotated and rotated *InteractiveGiotto* outputs, respectively. While (b) and (c) are both clearly drawings of the graph shown in (a), the resemblance is more readily seen in the properly rotated drawing (c).

because if there is a very large change it can become difficult to generate a clear, useful animation. It is thus still important to have a measure of how the look of the drawing changes.

Studying “difference” metrics to measure how much a drawing algorithm changes the user’s mental map has a number of benefits, including

- providing a basis for studying the behavior of constraint-based interactive drawing algorithms like *InteractiveGiotto* [4], where meaningful bounds on the movement of any given part of the drawing are difficult to obtain,
- providing a technique to compare the results of different interactive drawing algorithms, and
- providing a goal for the design of new drawing algorithms by identifying which qualities of the drawing are the most important to preserve.

Finding a good difference metric also has an immediate practical benefit, namely solving the “rotation problem” of *InteractiveGiotto*. *Giotto* [23], the core of *InteractiveGiotto*, does not take into account the coordinates of vertices and bends in the original drawing when constructing a new drawing — and, as a result, the output of *InteractiveGiotto* may be rotated by a multiple of 90 degrees (Figure 1). The problem can be solved by computing the value of the metric for each of the possible rotations and choosing the rotation with the smallest value.

Several models have been proposed for formalizing the notion of mental map [8, 13, 15], though more work needs to be done to formally define potential difference metrics and then experimentally validate (or invalidate) them. Validation can be via user studies similar to those done in [20, 21] to evaluate the impact of various graph drawing aesthetics on human understanding.

Motivated by applications to *InteractiveGiotto*, this paper will focus primarily on difference metrics for orthogonal drawings, though many of the metrics can be used without modification for arbitrary drawings. Section 2 describes the metrics being proposed, Section 3 presents a framework for evaluating the suitability of the metrics along with a preliminary evaluation, and Section 4 outlines plans for future work.

2 Metrics

The difference metrics being proposed fall into five categories: *distance*, *proximity*, *orthogonal ordering*, *shape*, and *topology*. While the distance category could be considered a subset of proximity, it is kept separate to distinguish between metrics using the Euclidean distance between points and those using relative orderings based on the distances. Proximity, ordering, and topology are suggested in [8, 13, 15] as qualities which should be preserved; distance, suggested in [13], and shape reflect intuition about what causes drawings to look different. Within each category the metrics were chosen to capture intuition about what qualities of the drawing are important to preserve.

It should be noted that all of the metrics are concerned only with the geometric aspects of the drawing; other features such as node color and line styles are also very important in preserving the look of the drawing and may be able to at least partially compensate for geometric changes.

2.1 Preliminaries

Terminology Every metric compares two drawings D and D' of the same graph G . Each object of the graph G can be associated with two sets of coordinates, one describing the position in D and the other the position in D' . A *matched set* of objects is a set of the pairs describing the object's position in the two drawings. The matched sets used are sets of points and sets of edges. A *point* is any aspect of the graph that has a single coordinate, such as the centers and corners of vertices. The matched point set P_m is the set of pairs (p_i, p'_i) where p_i is the location of point i in D and p'_i is the location of point i in D' . Let $d(p, q)$ be the (Euclidean) distance between two points p and q . An *edge* is simply an edge of a graph; the important features are the coordinates of the endpoints and bends. The matched edge set E_m is the set of matched edges (e_i, e'_i) where e_i is the edge i in D and e'_i is the edge i in D' .

Drawing Alignment Most of the metrics compare coordinates between drawings, which means that they are sensitive to the particular values of the coordinates. Figure 2 illustrates this sensitivity — if the difference metric relies on the distance $d(p_i, p'_i)$ between points, adjusting the scale and translation of one point set relative to the other makes a large difference.

To eliminate this effect, the drawings are *aligned* before the metric is computed. This is done by extracting a (matched) set of points from the drawings and applying a point set matching algorithm to obtain the best fit. In general the matching algorithm should take into account scaling, translation, and rotation, though it may be possible to eliminate one or more of the transformations for certain metrics or if something is known about the relationship between the two drawings. For example, interactive drawing algorithms often preserve the rotation of the drawing (see [2, 19] for examples), eliminating the need to consider rotation in the alignment stage. For orthogonal drawings, there are only eight possible rotations for the second drawing relative to the first — four multiples of 90 degrees, applied to the original drawing and its reflection about the x axis. These eight possibilities can be handled by computing the metric separately for

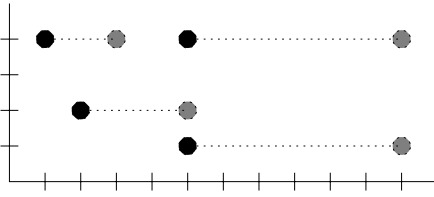


Fig. 2. Two point sets (black and gray) superimposed. (Corresponding points in the two sets are connected with dotted lines.) As shown, the Euclidean distance metric (Section 2.2) would report a distance of 4.25. However, translating the gray points one unit to the left and then scaling by $1/2$ in the x direction allows the point sets to be matched exactly, for a distance of 0. It should be noted that exact matches are not possible in general.

each rotation and taking the minimum value instead of incorporating rotation into the alignment process.

A great deal of work has been done on point set matchings; see [1, 5, 11] for several methods of obtaining both optimal and approximate matchings. Different methods can be applied when the correspondence between points is known as it is here; see [12] for an algorithm that minimizes the maximum distance between corresponding points under translation, rotation, and scaling. In the implementation used in Section 3, the alignment is performed by using gradient search to minimize the distance squared between points.

Point Set Selection Points can be selected in a number of ways. Two vertex-centered methods — *centers* and *corners* — are used here, to reflect the idea that vertex positions are a significant visual feature of the drawing [17]. “Centers” consists of the center points of each vertex; this captures how vertices move. “Corners” uses the four corners of each vertex, taking into account both vertex motion (the movement of the center) and changes in the vertex dimension. It seems important to take into account changes in vertex dimension because a vertex with a large or distinctive shape can act as a landmark to orient the user to the drawing; loss of that landmark makes orientation more difficult.

2.2 Distance

The distance metrics reflect the simple observation that drawings that look very different cannot be aligned very well, and vice versa. In order to make the value of the distance metrics comparable between pairs of drawings, they are scaled by the graph’s *unit length* u . For orthogonal drawings the unit length can be computed by taking the greatest common divisor of the Manhattan distances between vertex centers and bend points on edges. Non-orthogonal portions of the drawing, such modifications of an orthogonal drawing made by the user, can be ignored during the computation. While the determination of the unit length will be unreliable if a small portion of the drawing is orthogonal, scaling by the unit length is not necessary in some applications (e.g., solving the rotation problem of *InteractiveGiotto*) and can often be supplied manually if it is required.

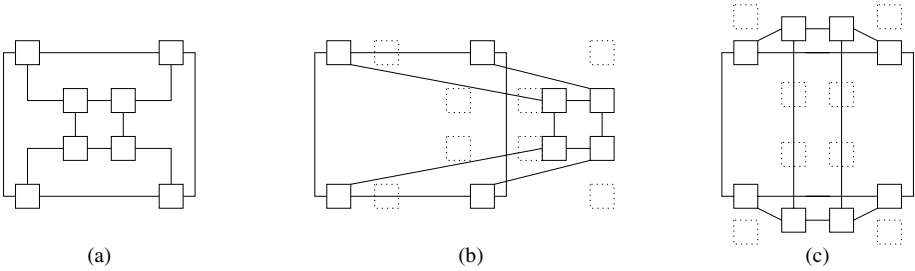


Fig. 3. Proximity: (b) looks more like (a) than (c) does because the relative shape of both the inner and outer squares are preserved even though the distance (using the Euclidean distance metric) between (c) and (a) is smaller. An aligned version of the vertices of (a), used in the computation of the distance metric, is shown with dotted lines in (b) and (c).

Hausdorff Distance The *Hausdorff distance* is a standard metric for determining the distance between two point sets, and measures the largest distance between a point in one set and its nearest neighbor in the other.

The (undirected) Hausdorff distance for a matched point set P_m is

$$\text{hausdorff}(P_m) = \frac{1}{u} \max\left\{\max_i \min_j d(p_i, p'_j), \max_j \min_i d(p'_i, p_j)\right\}$$

Euclidean Distance *Euclidean distance*, introduced in [13], is a simple metric measuring the average distance moved by each point from the first drawing to the second; it is motivated by the notion that if points move a long way from their locations in the first drawing, the second drawing will look very different.

$$\text{dist}(D, D') = \frac{1}{u |P_m|} \sum_{(p_i, p'_i) \in P_m} d(p_i, p'_i)$$

2.3 Proximity

The proximity metrics reflect the idea that points near each other in the first drawing should remain near each other in the second drawing. This is stronger than the distance metrics because it captures the idea that if a subgraph moves relative to another (but there are no changes within either subgraph), the distance should be less than if each point in one of the subgraphs moves in a different direction (Figure 3).

Three different metrics are used to try to capture this idea: nearest neighbor within (nn-within), nearest neighbor between (nn-between), and ϵ -clustering.

Nearest Neighbor Within *Nearest neighbor within* is based on the reasoning is that if p_j is the closest point to p_i in D , then p'_j should be closest point to p'_i in D' . Considering only distances within a single drawing means that nn-within is alignment-independent and thus not subject alignment errors, but means that it is not suitable for solving the rotation problem of InteractiveGiotto.

This metric has two versions, *weighted* and *unweighted*. In the weighted version the number of points closer to p'_i than p'_j is considered, whereas in the unweighted version it only matters if p'_j is or is not the closest point. The reasoning behind the weighted version is that if there are more points between p'_i and p'_j , the visual linkage between p'_i and p'_j has been disrupted to a greater degree and the drawing looks more different.

In both cases the distance is scaled by the number of points being considered and W , the maximum weight contributed by a single point.

$$\text{nnw}(D, D') = \frac{1}{W |P_m|} \sum_{(p_i, p'_i) \in P_m} \text{closer}(p'_i, p'_j)$$

where p_j is the closest point to p_i in D and

$$\text{closer}(p'_i, p'_j) = |\{k \mid d(p'_i, p'_k) < d(p'_i, p'_j)\}|, \quad W = |P_m| - 2 \quad (\text{weighted})$$

$$\text{closer}(p'_i, p'_j) = \begin{cases} 0 & \text{if } d(p'_i, p'_j) \leq d(p'_i, p'_k), k \neq i \\ 1 & \text{otherwise} \end{cases}, \quad W = 1 \quad (\text{unweighted})$$

Nearest Neighbor Between *Nearest neighbor between* is similar to nn-within but instead measures whether or not p' is the closest of the points in D' to p when the two drawings are aligned. The idea that a point should remain nearer to its original position than any other is also the force behind layout adjustment algorithms based on the Voronoi diagram [13].

$$\text{mb}(D, D') = \frac{1}{W |P_m|} \sum_{(p_i, p'_i) \in P_m} \text{closer}(p_i, p'_i)$$

where

$$\text{closer}(p_i, p'_i) = |\{j \mid d(p_i, p'_j) < d(p_i, p'_i)\}|, \quad W = |P_m| - 1 \quad (\text{weighted})$$

$$\text{closer}(p_i, p'_i) = \begin{cases} 0 & \text{if } d(p_i, p'_i) \leq d(p_i, p'_j) \\ 1 & \text{otherwise} \end{cases}, \quad W = 1 \quad (\text{unweighted})$$

Unlike nn-within, nn-between is not alignment- and rotation-independent and thus is suitable for solving the rotation problem.

ϵ -Clustering An ϵ -cluster for a point p is the set of points q such that $d(p, q) \leq \epsilon$, where a reasonable value to use for ϵ is (see [8])

$$\epsilon = \max_p \min_{q \neq p} d(p, q)$$

The ϵ -cluster metric measures how the ϵ -cluster for p_i compares to that for p'_i . Let C be the graph where the vertex set is the set of points and $(i, j) \in E_C$ if $d(p_i, p_j) \leq \epsilon_D$ or $d(p'_i, p'_j) \leq \epsilon_{D'}$. Let E_{C_D} be the set of edges for which the first condition holds and $E_{C_{D'}}$ be the set of edges for which the second condition holds. The distance is thus

$$\text{cluster}(D, D') = 1 - \frac{|E_{C_D} \cap E_{C_{D'}}|}{|E_C|}$$

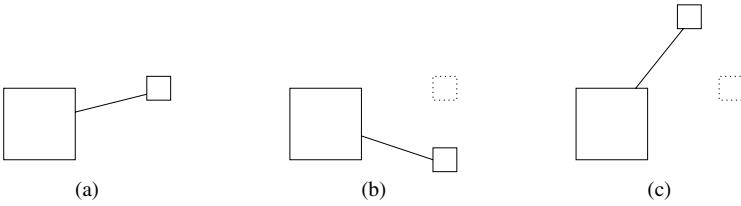


Fig. 4. Orthogonal ordering: Even though the angle the vertex moves relative to the center of the large vertex is the same from (a) to (b) and from (a) to (c), the perceptual difference between (a) and (c) is much greater. The original location of the vertex is shown with a dotted box in (b) and (c) for comparison purposes.

2.4 Orthogonal Ordering

The *orthogonal ordering* metric reflects the desire to preserve the relative ordering of every pair of points — if p is northeast of q in D , p' should remain to the northeast of q' in D' [8, 15]. The simplest measurement of difference in the orthogonal ordering is to take the angle between the vectors $q - p$ and $q' - p'$ (*unweighted* orthogonal ordering). This has the desirable feature that if q is far from p , $q' - q$ must be larger to result in the same angular move, which reflects the intuition that the relative position of points near each other is more important than the relative position of points that are far apart.

However, simply using the angular change fails to take into account situations such as that shown in Figure 4. This problem can be addressed by introducing a weight that depends on the particular angles involved in the move in addition to size of the move (*weighted* orthogonal ordering).

$$\text{order}(D, D') = \frac{1}{W |P_m|} \sum_{i,j} \min(\text{order}(\theta_{ij}, \theta'_{ij}), \text{order}(\theta'_{ij}, \theta_{ij}))$$

where θ_{ij} is the angle from the positive x axis to the vector $p_j - p_i$, θ'_{ij} is the angle from the positive x axis to the vector $p'_j - p'_i$, and

$$\begin{aligned} \text{order}(\theta_{ij}, \theta'_{ij}) &= \int_{\theta_{ij}}^{\theta'_{ij}} \text{weight}(\theta) \, d\theta, \\ W &= \min \left\{ \int_0^\pi \text{weight}(\theta) \, d\theta, \int_\pi^{2\pi} \text{weight}(\theta) \, d\theta \right\} \end{aligned}$$

The weight functions are

$$\text{weight}(\theta) = \begin{cases} \frac{\frac{\pi}{2} - (\theta \bmod \frac{\pi}{2})}{\frac{\pi}{4}} & \text{if } (\theta \bmod \frac{\pi}{2}) > \frac{\pi}{4} \\ \frac{\theta \bmod \frac{\pi}{2}}{\frac{\pi}{4}} & \text{if } (\theta \bmod \frac{\pi}{2}) \leq \frac{\pi}{4} \end{cases} \quad (\text{weighted})$$

$$\text{weight}(\theta) = 1 \quad (\text{unweighted})$$

The λ -matrix model for measuring the difference of two point sets, introduced in [13], is based on the concept of order type of a point set [10]. This model tries

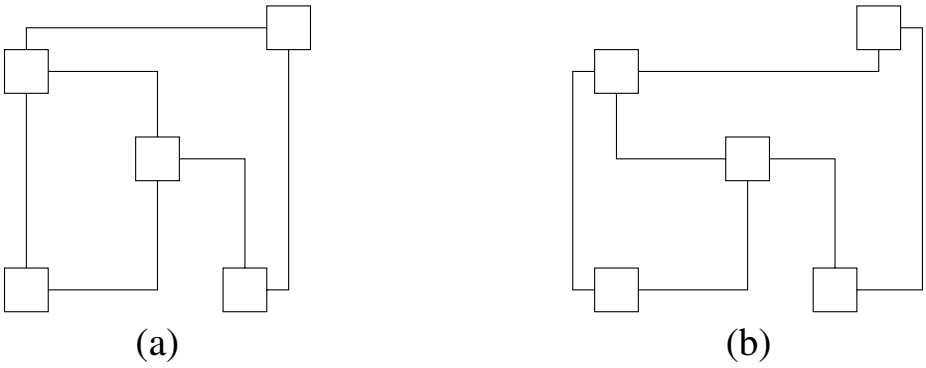


Fig. 5. Shape: (a) and (b) look different even though the graphs are the same and the vertices have the same coordinates.

to capture the notion of the relative position of vertices in a straight-line drawing and is thus related to the orthogonal ordering metric.

2.5 Shape

The *shape* metric is motivated by the reasoning that edge routing may have an effect on the overall look of the graph (Figure 5). The shape of an edge is the sequence of directions (north, south, east, and west) traveled when traversing the edge; writing the shape as a string of N, S, E, and W characters yields the *shape string* of the edge. For non-orthogonal edges the direction is taken to be the most prominent direction; for example, if the edge goes from (1,1) to (4,2) the most prominent direction is east. For each edge the minimum number of edits to transform the shape string in one drawing to the string in the other is computed, where an edit consists of inserting, deleting, or replacing a character in the shape string. The metric is the average number of edits per edge.

$$\text{shape}(D, D') = \frac{1}{|E_m|} \sum_{(e_i, e'_i) \in E_m} \text{edits}(e_i, e'_i)$$

Shape is scale- and translation-independent.

2.6 Topology

The topology metric reflects the idea that preserving the order of edges around a vertex is important in preserving the mental map [8, 15] — comparing the drawing produced by *Giotto* in Figures 6 and 7 to the user’s input illustrates this. However, since most interactive orthogonal drawing algorithms (see, for example, [4, 2, 9, 18, 19]) always preserve topology, it is not useful as a means of comparing these algorithms. It is also alignment-independent and so can not be used to solve the rotation problem of *InteractiveGiotto*. As a result, it is not discussed in any more detail here.

3 Analyzing the Metrics

Once defined, the suitability of the metrics must be evaluated. A good metric for measuring the difference between drawings should satisfy the following three requirements:

- it should *qualitatively* reflect the visual difference between two drawings, i.e. the value increases as the drawings diverge;
- it should *quantitatively* reflect the visual difference so that the magnitude of the difference in the metric is proportional to the perceived difference; and
- in the rotation problem of `InteractiveGiotto`, the metric should have the smallest value for the correct rotation, though this requirement can be relaxed when the difference between drawings is high since in that case there is no clear “correct” rotation.

The third point is the easiest to satisfy — in fact, most of the metrics defined in the previous section can be used to solve the rotation problem — but is still important worth considering since the problem was one of the factors that first inspired this work.

Evaluating the qualitative and quantitative behavior of potential metrics requires ranking pairs of drawings based on the visual difference between the existing drawing and the new drawing in each pair. `InteractiveGiotto` provides a convenient way of obtaining an ordered set of drawings because it allows the constraints preserving the layout to be relaxed. By default `InteractiveGiotto` preserves edge crossings, the direction (left or right) and number of bends on an edge, and the angles between consecutive edges leaving a vertex. Recent modifications allow the user to turn off the last two constraints on an edge-by-edge or vertex-by-vertex basis. This means that it is possible to produce a series of drawings which are progressively farther from the original by iteratively relaxing more of the constraints. A smooth way of relaxing the constraints is to use a breadth-first ordering, expanding outward from the user’s modifications — in the first step all of the constraints are applied, in the second step the bend and angle constraints are relaxed for all of the modified objects, in the third step the angle constraints are relaxed for all vertices adjacent to edges whose bends constraints have been relaxed, in the fourth step the bend constraints are relaxed for all edges adjacent to vertices whose angle constraints have been relaxed, and so on, alternating between angle and bend constraints until all of the constraints have been relaxed. This relaxation method is based on the idea that the user is most willing to allow restructuring of the graph near where her changes were made, so progressively expanding the sphere of influence of the changes results in a series of drawings in which the mental map is increasingly disrupted.

Figures 6 and 7 show two such relaxation sequences; the base graphs and user modifications are those used in the first two steps of Figure 2 in [4]. `Giotto`’s redraw-from-scratch drawing of the graph is also included for comparison. Figures 8 and 9 show the results of the difference metrics for each sequence of drawings.

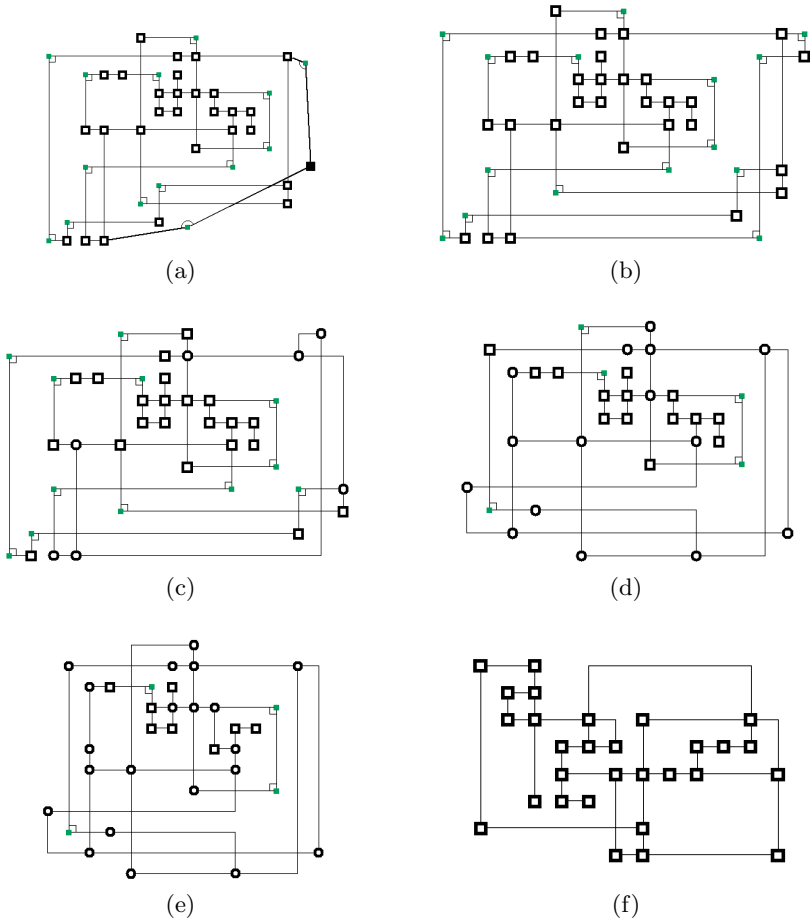


Fig. 6. Relaxations for stage 1. (a) shows the user’s modifications, (b)–(e) show the output from *InteractiveGiotto* for successive relaxations, and (f) is the output from *Giotto*. Rounded vertices and bends without markers indicate vertices and bends for which the constraints have been relaxed.

Some preliminary work has been done on evaluating the proposed metrics based on their qualitative behavior. Six sets of graphs consisting of a base graph modified by several successive sets of modifications were generated from two applications where interactive graph drawing algorithms are useful — mapping how a user might explore topics when querying a search engine and mapping the exploration of a web site by a user or web crawler. Two other sets were generated from two of the same sets by breaking the modifications generated by each of the user’s steps into single changes which were applied individually. A total of 62 graphs were generated in this way; for each a series of drawings was produced using *InteractiveGiotto* with progressively relaxed constraints. *Giotto* was also run on each graph.

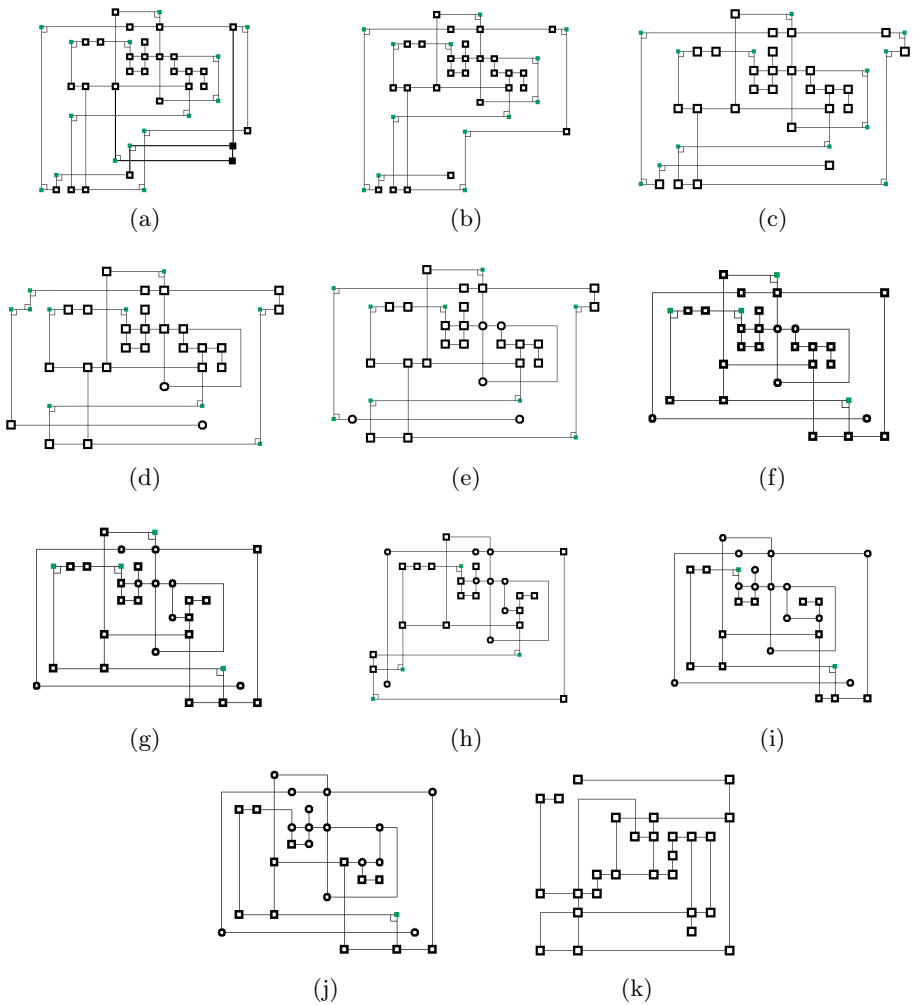
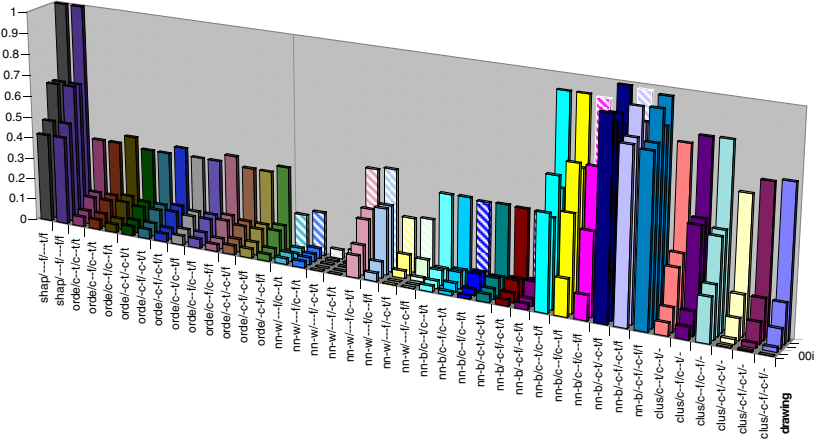
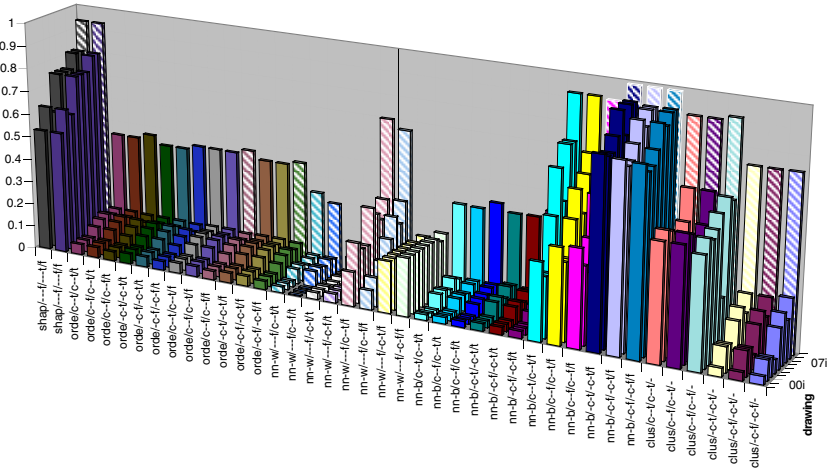


Fig. 7. Relaxations for stage 2. (a) shows the starting graph, (b) shows the user's modifications (two vertices and their adjacent edges deleted), (c)–(j) show the output from InteractiveGiotto for successive relaxations, and (k) is the output from Giotto. Rounded vertices and bends without markers indicate vertices and bends for which the constraints have been relaxed.

The initial analysis shows that when the user's modifications are small (affecting only a single vertex or edge) most of the metrics are well-behaved — that is, they increase in value as the constraints are relaxed, with the highest value being for the Giotto drawing. When the size of the changes is increased, shape, Euclidean distance, and to some degree clustering tend to remain the



(a) Values of metrics for each drawing in stage 1. Drawings (b)–(f) are shown from front to back.



(b) Values of metrics for each drawing in stage 2.

Fig. 8. Striped bars indicate entries for which the metric reported the wrong rotation; white outlines indicate that multiple rotations had the same smallest value for the metric. The label for each metric is of the form $\langle \text{metric name} \rangle / \langle \text{weighted} \rangle / \langle \text{alignment} \rangle / \langle \text{points} \rangle$, where $\langle \text{weighted} \rangle$ is t or f indicating if the weighted version of the metric was used (if applicable), and $\langle \text{alignment} \rangle$ and $\langle \text{points} \rangle$ are strings indicating the choice of points used for alignment and points, respectively. c– indicates using vertex centers, -c- indicates vertex corners, and the last position (t or f) indicates whether the edges and vertices modified by the user were considered.

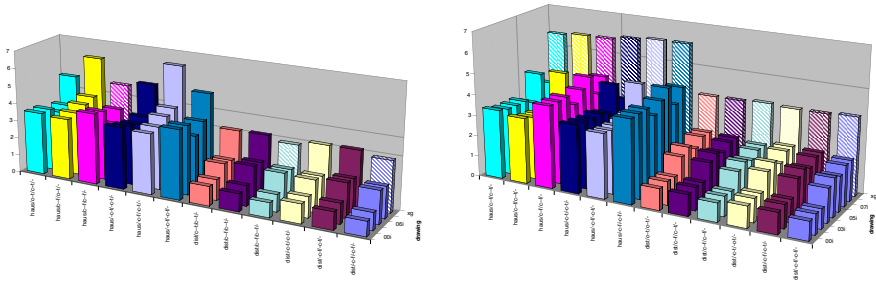


Fig. 9. Hausdorff and Euclidean distance metrics for stages 1 and 2, respectively.

most well-behaved. Further study is needed to determine why these effects are being observed and if they are indeed typical of the metrics.

4 Future Work

As mentioned, the natural next step is to perform more extensive experimentation using the relaxation sequences produced by *InteractiveGiotto* from both real-world and randomly generated “typical” graphs. The randomly generated graphs can be created using a method similar to that used in [7] to produce realistic graphs for the experimental analysis of several orthogonal drawing algorithms.

The sequences produced by *InteractiveGiotto* allow a qualitative assessment of the metrics — it is easy to see visually that the difference between the base drawing and the relaxed drawing increases as the constraints are relaxed — but do not allow for a very good quantitative assessment because there is only a relative ordering of the drawings within a sequence and there is no ordering of drawings from different sequences relative to each other. User studies can be used to identify the importance of factors such as distance between points, proximity, orthogonal ordering, and shape in the user’s mental map and assign difference values to each drawing, which can then be compared to the values of the metrics.

Once suitable metrics have been identified and validated through user studies, they can be used to compare the behavior of interactive graph drawing algorithms as well as potentially providing inspiration for new drawing algorithms.

References

- [1] H. Alt, O. Aichholzer, and G. Rote. Matching shapes with a reference point. *Internat. J. Comput. Geom. Appl.*, 1997. to appear.
- [2] T. Biedl and M. Kaufmann. Area-efficient static and incremental graph drawings. In R. Burkard and G. Woeginger, editors, *Algorithms — ESA ’97*, volume 1284 of *Lecture Notes Comput. Sci.*, pages 37–52. Springer-Verlag, 1997.
- [3] U. Brandes and D. Wagner. A bayesian paradigma for dynamic graph layout. In G. Di Battista, editor, *Graph Drawing (Proc. GD ’97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 236–247. Springer-Verlag, 1997.

- [4] S. S. Bridgeman, J. Fanto, A. Garg, R. Tamassia, and L. Vismara. Interactive-Giotto: An algorithm for interactive orthogonal graph drawing. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 303–308. Springer-Verlag, 1997.
- [5] L. P. Chew, M. T. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under Euclidean motion. *Comput. Geom. Theory Appl.*, 7:113–124, 1997.
- [6] R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis. Dynamic graph drawings: Trees, series-parallel digraphs, and planar *ST*-digraphs. *SIAM J. Comput.*, 24(5):970–1001, 1995.
- [7] G. Di Battista, A. Garg, G. Liotta, R. Tamassia, E. Tassinari, and F. Vargiu. An experimental comparison of four graph drawing algorithms. *Comput. Geom. Theory Appl.*, 7:303–326, 1997.
- [8] P. Eades, W. Lai, K. Misue, and K. Sugiyama. Preserving the mental map of a diagram. In *Proceedings of Compugraphics 91*, pages 24–33, 1991.
- [9] U. Fößmeier. Interactive orthogonal graph drawing: Algorithms and bounds. In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 111–123. Springer-Verlag, 1997.
- [10] J. E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12:484–507, 1983.
- [11] M. T. Goodrich, J. S. B. Mitchell, and M. W. Orletsky. Practical methods for approximate geometric pattern matching under rigid motion. *IEEE Trans. Pattern Anal. Mach. Intell.* to appear.
- [12] K. Imai, S. Sumino, and H. Imai. Minimax geometric fitting of two corresponding sets of points. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 266–275, 1989.
- [13] K. A. Lyons, H. Meijer, and D. Rappaport. Algorithms for cluster busting in anchored graph drawing. *Journal of Graph Algorithms and Applications*, 2(1):1–24, 1998.
- [14] K. Miriyala, S. W. Hornick, and R. Tamassia. An incremental approach to aesthetic graph layout. In *Proc. Internat. Workshop on Computer-Aided Software Engineering*, 1993.
- [15] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Lang. Comput.*, 6(2):183–210, 1995.
- [16] S. Moen. Drawing dynamic trees. *IEEE Software*, 7:21–8, 1990.
- [17] S. North. Incremental layout in DynaDAG. In *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes Comput. Sci.*, pages 409–418. Springer-Verlag, 1996.
- [18] A. Papakostas, J. M. Six, and I. G. Tollis. Experimental and theoretical results in interactive graph drawing. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 371–386. Springer-Verlag, 1997.
- [19] A. Papakostas and I. G. Tollis. Interactive orthogonal graph drawing. In *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes Comput. Sci.* Springer-Verlag, 1996.

- [20] H. Purchase. Which aesthetic has the greatest effect on human understanding? In G. Di Battista, editor, *Graph Drawing (Proc. GD '97)*, Lecture Notes Comput. Sci., pages 248–261. Springer-Verlag, 1997.
- [21] H. C. Purchase, R. F. Cohen, and M. James. Validating graph drawing aesthetics. In F. J. Brandenburg, editor, *Graph Drawing (Proc. GD '95)*, volume 1027 of *Lecture Notes Comput. Sci.*, pages 435–446. Springer-Verlag, 1996.
- [22] K. Ryall, J. Marks, and S. Shieber. An interactive system for drawing graphs. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 387–393. Springer-Verlag, 1997.
- [23] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18(1):61–79, 1988.