



Metrics for Graph Drawing Aesthetics

HELEN C. PURCHASE

*School of Computer Science and Electrical Engineering, The University of Queensland,
St. Lucia 4072, Queensland, Australia*

Received 8 March 2001; revised 11 January 2002; accepted 16 January 2002

Graph layout algorithms typically conform to one or more aesthetic criteria (e.g. minimizing the number of bends, maximizing orthogonality). Determining the extent to which a graph drawing conforms to an aesthetic criterion tends to be done informally, and varies between different algorithms. This paper presents formal metrics for measuring the aesthetic presence in a graph drawing for seven common aesthetic criteria, applicable to any graph drawing of any size. The metrics are useful for determining the aesthetic quality of a given graph drawing, or for defining a cost function for genetic algorithms or simulated annealing programs. The metrics are continuous, so that aesthetic quality is not stated as a binary conformance decision (i.e. the drawing either conforms to the aesthetic or not), but can be stated as the extent of aesthetic conformance using a number between 0 and 1. The paper presents the seven metric formulae. The application of these metrics is demonstrated through the aesthetic analysis of example graph drawings produced by common layout algorithms.

© 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Graph drawing, Aesthetics, Metrics

1. Introduction

AUTOMATIC GRAPH layout algorithms take as input an abstract graph structure comprising relational information about objects and the associations between them, and produce a visual representation of the graph, where objects are typically represented as circles on a two-dimensional plane, and the associations represented as lines between the circles. Many such algorithms have been produced [1], all of which typically take into account one or more aesthetic criteria, with the assumption that by doing so, the readability of the drawing is increased. Such aesthetic criteria include, for example, minimizing the number of edge crossings, maximizing the depiction of symmetry, and maximizing the minimum angle between adjacent edges leaving a node.

This paper presents a set of formal, objective metrics (scaled to lie between 0 and 1), which measure the extent to which a graph drawing conforms to each of seven common aesthetic criteria. Currently, the measurement of these aesthetic criteria within a graph drawing is done informally, and may differ between different algorithms. There is no standard, objective way for analysing a graph drawing with respect to the presence of different aesthetics. Continuous measures are necessary, so that analysing a drawing with respect to an aesthetic is not merely a binary decision (that is, a drawing is considered

‘orthogonal’ or ‘not orthogonal’), but is rather an indication of the extent to which the drawing conforms to the aesthetic (that is, a drawing may be considered to have 65% presence of orthogonality).

Aside from providing a formal method for analysing the aesthetic quality of graph drawings, these computational aesthetic metrics can be used for the definition of cost functions for genetic algorithms and simulated annealing programs. That is, an ideal aesthetic quality can be defined in advance (for example, the drawing should have at least 70% symmetry, at most 5% ‘crossiness’, and 0% ‘bendiness’). An evaluation function, which determines whether these criteria are satisfied, may then be implemented using computational metrics for each of these aesthetics, and used to indicate whether further iterations are required.

Many algorithms attempt to conform to the extreme of the aesthetics (i.e. removing all bends, or ensuring that all nodes are placed on an invisible grid). While it is generally assumed within the graph drawing community that these aesthetics improve the readability of graph drawings (although these assumptions have recently been investigated [2]), it may be the case that the usefulness of an aesthetic is related to a critical mass rather than an extreme: perhaps drawings with at most 10% ‘crossiness’ are as useful as those with 0% ‘crossiness’ or perhaps a drawing needs to be at least 90% orthogonal before the usability effects of the orthogonality aesthetic are evident. Without continuous measures, this notion of critical mass, and the computational implication of relaxing the requirement that the aesthetic be satisfied at the extreme cannot be investigated.

The metrics are intended to be applicable in the analysis of drawings of any graph of any structure or size, enabling quantitative comparisons between drawings of different graphs, and to ensure that they can be used universally in generic cost functions in iterative algorithms. While Bridgeman and Tamassia [3] define some metrics associated with graph drawings, their concern is with the measurement of differences between two drawings of the same graph in a dynamic environment.

Metrics for seven aesthetics are proposed here. All the aesthetics are defined such that the measurement is a real number in $[0,1]$, where 1 indicates the positive aspect of the aesthetic (i.e. the amount of the aesthetic for which it is assumed the drawing is easiest to read: for example, a low number of bends, or a high amount of symmetry). Scaling the metrics in this way ensures that the metric value does not depend on the nature of the underlying graph. The seven aesthetics for which metrics have been defined are:

- minimizing edge crossings,
- minimizing edge bends,
- maximizing symmetry,
- maximizing the minimum angle between edges leaving a node,
- maximizing edge orthogonality,
- maximizing node orthogonality,
- maximizing consistent flow direction (directed graphs only).

With the exception of symmetry (for which an objective metric definition is trivial and not very useful, see Section 5, below), all the metrics have been defined objectively, and are not intended to take human value judgements based on perception of what appears ‘good’ into account. Such value judgements could only be considered valid if they were the results of an empirical study (that is, taking only the personal opinion of the author into account would be inappropriate: more extensive user studies would be required).

Experimental research to determine whether these objective measurements correspond to human perception is left for a later study.

This paper begins with terminology and definitions, before describing the definition of each of the seven metrics in detail. Example drawings and their calculated aesthetic values are then presented to demonstrate the application of these metrics.

2. Definitions

A graph G has n nodes and m edges, with the i th node denoted u_i , and its degree as $degree(u_i)$. G can be rendered as a graph drawing $D(G)$ on a two-dimensional plane by associating a co-ordinate pair (x_i, y_i) with each node u_i .

For the purposes of this paper, it is assumed that G is connected, that it contains at least one edge, and that there is at most one edge between any two nodes. It is also assumed that no two vertices in $D(G)$ share the same co-ordinates.

Some of the metrics presented here (e.g. crosses, orthogonality), require that an auxiliary graph drawing be defined ($D'(G)$). The graph drawing $D(G)$, with polyline edges, is used to derive $D'(G)$ with straight-line edges. This derivation (called *bends promotion*) consists of promoting the bends in the edges of $D(G)$ into nodes in $D'(G)$, and replacing edge segments of the bent edges in $D(G)$ with new straight-line edges in $D'(G)$ (see Figure 1). The properties of $D'(G)$ are written primed, as n' , m' , etc. When $D(G)$ is a straight-line drawing, then $D(G)$ and $D'(G)$ are identical.

3. Crossings \aleph_c

The edge crossings aesthetic metric (\aleph_c) for $D(G)$ is based on c , the number of edge crossings in $D'(G)$, where an edge crossing is defined as a point on the plane where two edges intersect.

When calculating c , only pairwise edge intersections are considered. In the case where $k \geq 2$ edges cross at a single point, it is treated as though $\frac{1}{2}k(k - 1)$ individual pairwise crossings have occurred (see Figure 2).

To produce a metric value between 0 and 1, the number of edge crosses needs to be scaled against an upper bound of the number of possible crosses. Much work has been

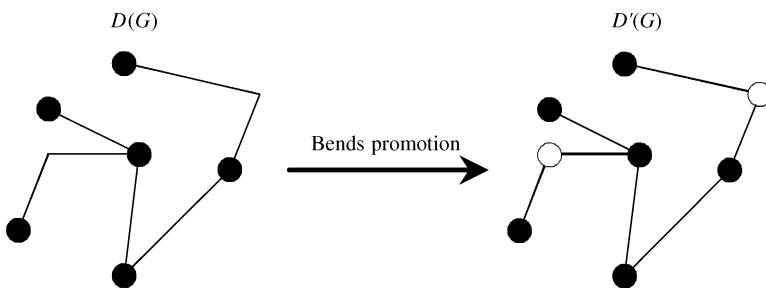


Figure 1. Example of *bends promotion*: the white nodes of $D'(G)$ are of the ‘bends promoted’ type

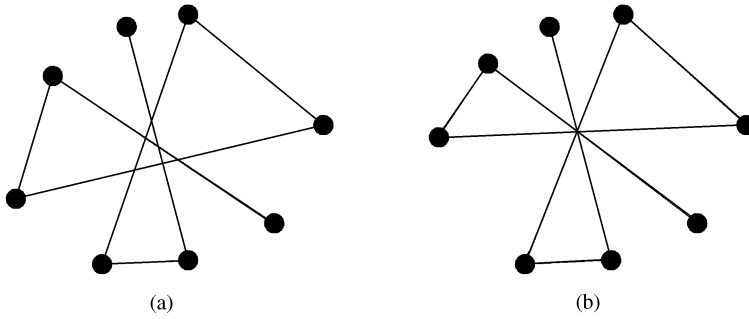


Figure 2. In both drawings, $k = 6$ edges are involved in intersections. Drawing (a) has $c = 6$ crossings, and drawing (b) has $c = \frac{1}{2}(4)(4 - 1) = 6$ crossings

done on defining the lower bound on the number of possible crosses in a graph drawing (e.g. [4]). While Jensen [5] defines a formula for the upper bound, he considers only fully connected graphs, and Shahrokhi [6] defines upper bounds in terms of an unspecified constant (assumed to be dependent on the structure of the graph).

For the purposes of the definition of a metric that can be universally applied to graphs of any structure, we define a reasonable approximation for the upper bound of the number of edge crossings.

First bends promotion is performed on $D(G)$ to form a straight-line graph drawing $D'(G)$. We then calculate the number of edge crossings if every edge were to cross every other edge (i.e. the total number of edge pairs). This is an overestimate from which we then subtract the total number of edge crossings that are known to be impossible in a straight-line graph drawing.

The first component of the formula is therefore

$$c_{all} = \sum_{i=1}^{m'} (i - 1) = \frac{m'(m' - 1)}{2}$$

arrived at by considering that every edge is crossed by every other edge.

In straight-line drawings of connected graphs with at most one edge between nodes, adjacent edges cannot cross. The total number of such impossible crossings is therefore

$$c_{impossible} = \frac{1}{2} \sum_{j=1}^{n'} degree(u_j)(degree(u_j) - 1)$$

where v_i and w_i are the end nodes of the i th edge.

The approximation for the upper bound on the number of edge crosses in a drawing of any graph is therefore

$$c_{max} = c_{all} - c_{impossible} = \frac{m'(m' - 1)}{2} - \frac{1}{2} \sum_{j=1}^{n'} degree(u_j)(degree(u_j) - 1)$$

The actual number of crosses in the drawing is therefore scaled against the maximum possible number of crossings, to give a number between 0 and 1. So that 1 represents

maximum ‘crosslessness’ (i.e. assumed to be easier to read), the scaled measurement of crosses is subtracted from 1. The edge crossing aesthetic metric is therefore defined by

$$\aleph_c = 1 - \begin{cases} \frac{c}{c_{mx}} & \text{if } c_{mx} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The metric is limited to the range $0 \leq \aleph_c \leq 1$. Note that the internal structure of many straight-line graphs does not allow for every edge to cross all other edges, and therefore the approximation of the upper bound is likely to be an overestimate. In particular, this upper bound of edge crossings is impossible for drawings of any graph that contains a cycle, and in this case, \aleph_c can never equal 0. Using an approximation of the upper bound is necessary, however, if the metric is to be universally applicable to graphs of any structure.

4. Bends \aleph_b

The aesthetic metric for bends (\aleph_b) for $D(G)$ is based on b , the number of bent edges in the drawing; that is, internal points of an edge whose co-ordinates do not lie on the straight line between the two end nodes of the edge. The number of bends can be calculated directly as a result of bends promotion:

$$\begin{aligned} b &= n' - n \\ &= m' - m \end{aligned}$$

We cannot scale against an upper bound of the number of bends (as this is infinite), so we scale by the total number of edge segments:

$$b_{avg} = \frac{m' - m}{m'}$$

where $0 \leq b_{avg} < 1$. When $D(G)$ has many bends, $m' \gg m$ and $b_{avg} \approx 1$. Similarly, when $D(G)$ contains no bends, $m' = m$ and $b_{avg} = 0$.

So that 1 represents maximum ‘bendlessness’ (i.e. assumed to be easier to read), this scaled measurement of bends is subtracted from 1. The bends aesthetic metric is therefore defined by

$$\aleph_b = 1 - b_{avg}$$

Note that $\aleph_b = 0$ is impossible, as it implies the existence of an infinite number of bends.

5. Symmetry \aleph_s

From a strictly geometric point of view, axial symmetry in a graph drawing may be determined simply by considering a single axis and the mirrored congruence of the nodes and edges on either side. This would produce a binary indication of whether the drawing is symmetric or not.

However, an approach that insists on exact geometric symmetry, and returns merely a binary value, is limited in its use. The computational aesthetic metric (\mathbb{N}_s) is more inclusive, as its definition takes into account some assumptions about the human perception of symmetry, and returns a numeric value (scaled between 0 and 1) to indicate the extent to which the drawing can be considered symmetric. This metric considers only reflective symmetry, ignoring rotational symmetry.

The steps of the algorithm are as follows:

- for each pair of nodes, generate a potential axis;
- for each axis, determine whether there are sufficient reflected nodes around the axis for a symmetric subgraph to be identified;
- for each symmetric subgraph, calculate a symmetric value depending on whether the reflected nodes are of the same type or not; weight this symmetric value by multiplying it by the area of the symmetric subgraph;
- add the weighted symmetry values of all the symmetric subgraphs, and scale this value by dividing by the total area of the graph drawing (or the total area of all the subgraphs, whichever is the maximum), to give a value for symmetry that lies between 0 and 1.

This metric is very computationally expensive, with a worst case of $O(n^7)$ and a best case of $O(n^5)$. The pseudo-code algorithm to determine the symmetry value of a graph drawing is presented in Figures 3 and 4.

Symmetry_Aesthetic ($D(G)$, $THRESHOLD$, $TOLERANCE$, $FRACTION$)

1. Generate $D'(G)$ from $D(G)$ using bends promotion (see Figure 1)
2. Generate $D''(G)$ from $D'(G)$ using crosses promotion (see Figure 7)
3. Set $TOTAL_SYM$ to 0
4. Set $TOTAL_AREA$ to 0
5. Generate the set of all possible axes in $D''(G)$ (the set of the perpendicular bisectors of all possible node pairs)
6. For each axis, A
 - (a) Determine whether there is a symmetric subgraph around A . For a symmetric subgraph to exist around A , there should be at least $THRESHOLD$ edges which are mirrored around A (where the end nodes of the edges are mirrored within $TOLERANCE$ pixels)
 - (b) If such a subgraph exists for A
 - (i) calculate the symmetry value of the subgraph, called SUB_SYM [see function **Subgraph_Symmetry**(Figure 4)].
 - (ii) calculate the convex hull area of the subgraph, called SUB_AREA .
 - (iii) add SUB_AREA to $TOTAL_AREA$
 - (iv) add $SUB_SYM \cdot SUB_AREA$ to $TOTAL_SYM$
7. Calculate the convex hull area of $D''(G)$, called $WHOLE_AREA$
8. Divide $TOTAL_SYM$ by the maximum of $WHOLE_AREA$ and $TOTAL_AREA$, and return this value as the scaled symmetry aesthetic value for $D(G)$.

Figure 3. Pseudo-code for the algorithm to measure the symmetry metric of a graph drawing. Four parameters are required: the graph drawing ($D(G)$), the minimum number of mirrored edges required for identification of a symmetric subgraph ($THRESHOLD$), the pixel range within which two nodes are deemed to be symmetric around an axis ($TOLERANCE$), and the proportional weight given to symmetric nodes which are not of the same type ($FRACTION$)

Subgraph_Symmetry (*SUBGRAPH* , *FRACTION*)

(The subgraph is defined by a set of edges that are mirrored around an axis. The end nodes of these edges may be of three types: real, bends-promoted and crosses-promoted. *FRACTION* is a weighting parameter within the range 0 – 1.)

1. Set *TOTAL* to 0
2. For each pair of mirrored edges in *SUBGRAPH* (see Figure 8)
 - (a) for the first pair of mirrored end nodes, if they are of the same type, set $P=1$; if they are of different types, set $P = FRACTION$
 - (b) for the second pair of mirrored end nodes, if they are of the same type, set $Q=1$; if they are of different types, set $Q = FRACTION$
 - (c) Add $P \cdot Q$ to *TOTAL*
3. Calculate the average symmetric value per edge pair by dividing *TOTAL* by the number of edge pairs in *SUBGRAPH* , and return this value as the symmetric value for *SUBGRAPH* .

Figure 4. Pseudo-code for the **Subgraph_Symmetry** function

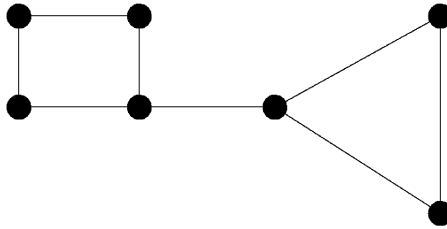


Figure 5. Example of a drawing with obvious local symmetries, but no global symmetry

5.1. Assumptions Relating to the Perception of Symmetry

The metric definition takes into account the following assumptions about the perception of symmetry in a graph drawing, which are based on informal tests and observations:

1. Both local and global symmetries can be observed in a graph drawing. For example, a drawing may be considered to be highly symmetric if it has two non-overlapping symmetric components but no global symmetry (see Figure 5). This assumption is taken into account in the algorithm by identifying all the component symmetric subgraphs of the graph drawing.
2. The amount that a symmetric component contributes to the overall symmetry of the drawing is related to its area. This assumption is taken into account in the algorithm by weighting the symmetry value of each identified symmetric subgraph by its convex hull area.
3. It is hard to identify symmetric components as separate objects when their areas overlap. The edge crossings interfere with the perception of symmetry (see Figure 6). This assumption is taken into account in the algorithm by considering crosses as playing as important a role in the identification of symmetric subgraphs as nodes do. The algorithm performs *crosses promotion* on the bends-promoted graph drawing $D'(G)$ to generate $D''(G)$. This introduces ‘promoted nodes’ at edge crossings (similar to those produced by bends promotion) and the two crossing edges in $D'(G)$ are converted into four new replacement edges (see Figure 7). This crosses-promotion results in

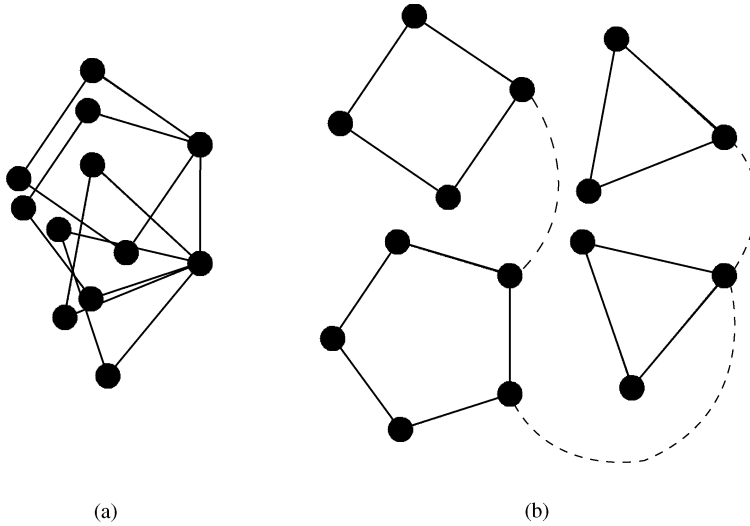


Figure 6. Example of non-obvious geometric symmetry in sub graph drawings due to overlap. The graph drawing in (a) is composed of the highly symmetric sub graph drawings shown in (b): the dashed lines indicate identical nodes, separated for clarity. Despite the overlapping symmetry, (a) does not have perceptually obvious symmetry. Hence, it is considered inappropriate to identify local symmetries without considering the effect of edge crossings

- overlapping symmetric components not always being considered as symmetric, as the crosses that they include introduce additional constraints on the detection of symmetry.
4. Both crosses and bends determine the observed shape of symmetric components in the same way that nodes do. This assumption is taken into account in the algorithm by using all three types of nodes (real, bends and crosses) in the same way when identifying symmetric subgraphs. The nature of the end-nodes of symmetric edges in the subgraph of the bends- and crosses-promoted drawing is taken into account when computing the symmetric values: if the mirrored nodes are of the same type, their weighting is 1; if not, their weighting is a value between 0 and 1 (called *FRACTION*)^a. Figure 8 illustrates how the algorithm treats the symmetry of edges with nodes of different types.
 5. Perception of symmetry does not require exact geometrical correspondence between mirrored nodes. This assumption is taken into account in the algorithm by using a degree of tolerance (called *TOLERANCE*, measured in pixels) when comparing node co-ordinates around an axis. If the magnitude of the vector between a node and its mirrored counterpart is less than the tolerance value, the two nodes are deemed to be symmetric around the axis^b.
 6. Trivial symmetries are ignored; for example, when shown a simple line, the property of symmetry will tend to be overlooked. This assumption is taken into account in the algorithm by setting a minimum threshold on the number of symmetric edges in a subgraph drawing for it to be considered symmetric, called *THRESHOLD*^c.

^a The examples in Section 9 use a value of $\frac{1}{2}$ for *FRACTION*.

^b The examples in Section 9 use a value of 3 pixels for *TOLERANCE*.

^c The examples in Section 9 use a value of 2 for *THRESHOLD*.

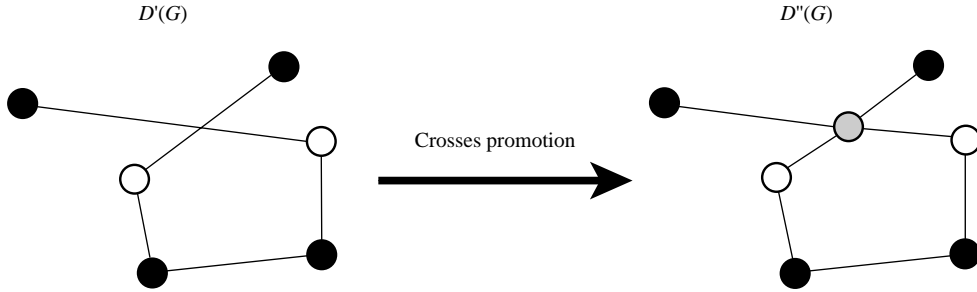


Figure 7. Example of *crosses promotion*: the shaded node of $D''(G)$ is of the ‘cross promoted’ type

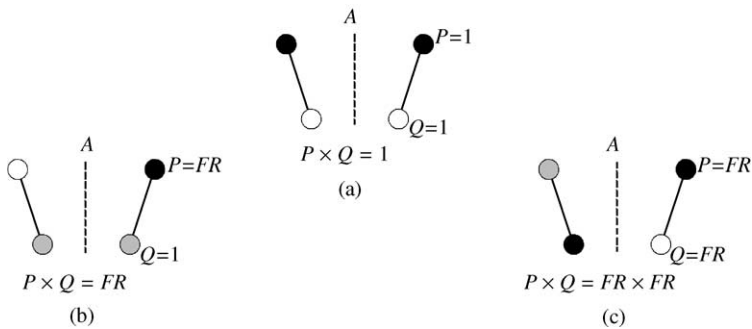


Figure 8. Example of calculating the edge-pair weighting ($P \times Q$) for three kinds of node-type combinations in the **Subgraph Symmetry** function in the algorithm described in Figure 4. The three types of nodes are ‘real’ nodes, ‘bends-promoted’ nodes, and ‘crosses-promoted’ nodes. In this diagram, the shading of the nodes indicates whether they are of the same type. *FRACTION* is a parameter value between 1 and 0 (abbreviated to *FR* for clarity). Edges e and f are symmetric about axis A . In (a) all nodes agree with the type of their reflected counterpart, leading to $P = Q = 1$ and $P \times Q = 1$. In (b) one of the node pairs is type-asymmetric leading to $P \times Q = FR$. When both pairs are type-asymmetric, as in (c), the contribution of the edge pair (e, f) is $P \times Q = FR^2$

The metric is scaled to lie within the range $0 \leq \aleph_s \leq 1$, through division by the maximum of the sum of the area of all the component symmetric subgraphs, and the total area of the drawing.

6. Minimum Angle \aleph_m

The minimum angle aesthetic metric (\aleph_m) for $D(G)$ is based on d , the average deviation of adjacent incident edge angles from the ideal minimum angle^d:

$$d = \frac{1}{n} \sum_{i=1}^n \left| \frac{\vartheta_i - \theta_{i \min}}{\vartheta_i} \right|$$

^d Note that this metric could be adapted to use the promoted drawing $D'(G)$, thus also taking into account the angles at the edge bends.

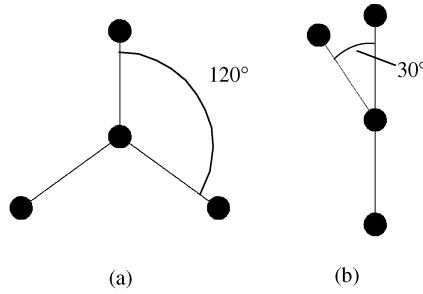


Figure 9. Example graph drawings where (a) has maximized angles at its centre node, of $\vartheta = \theta_{min} = 120^\circ$, resulting in $\aleph_m = 1$. Drawing (b) has a small minimum angle of $\theta_{min} = 15^\circ$ at the same node and thus has $\aleph_m = 1 - \frac{1}{4} \times ((120 - 15)/120 + 0 + 0 + 0) = 0.78$

where ϑ_i is the ideal (maximal) minimum angle at the i th node

$$\vartheta_i = \frac{360^\circ}{degree(v_i)}$$

and $\theta_{i min}$ is the actual minimum angle between the incident edges at the i th node.

So that 1 represents drawings with optimal angles (i.e. assumed easier to read), this measurement of edge deviation is subtracted from 1. The minimum angle aesthetic metric is therefore defined by

$$\aleph_m = 1 - d$$

The metric is constrained, $0 \leq \aleph_m \leq 1$, and is at a maximum when all the nodes have equal angles between all incident edges (see Figure 9). Restricting the aesthetic to connected graphs with at least one edge ensures that $\forall i degree(v_i) \geq 1$ (Figure 9).

7. Orthogonality \aleph_{eo}, \aleph_{no}

The concept of orthogonality in a graph drawing is here separated into two independent measurements:

- the extent to which edges and edge segments follow the lines of an imaginary Cartesian grid (*edge orthogonality*, \aleph_{eo}),
- the extent to which nodes and bend points make maximal use of the grid points in an imaginary Cartesian grid (*node orthogonality*, \aleph_{no}).

7.1. Edge Orthogonality \aleph_{eo}

The edge deviation factor of the i th edge segment (δ_i) represents how far away from an orthogonal angle the edge segment has deviated. It is computed as a proportion of the angular deviation of the i th edge segment from the horizontal or vertical gridlines:

$$\delta_i = \frac{\min(\theta_i, |90^\circ - \theta_i|, 180^\circ - \theta_i)}{45^\circ}$$

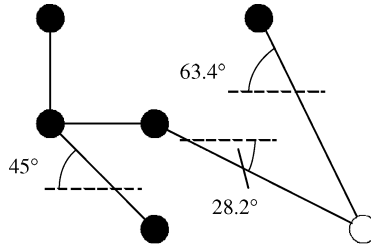


Figure 10. Example of calculating the edge orthogonality aesthetic metric when $m' = 5$. There are three edges with $\delta > 0$, giving

$$\aleph_{eo} = 1 - \frac{1}{m'} \sum_{i=1}^{m'} (\delta_i) = 1 - \frac{1}{5} \left(0 + 0 + \frac{45}{45} + \frac{26.6}{45} + \frac{28.2}{45} \right) = 1 - \frac{1}{5} (1 + 0.59 + 0.63) = 0.44$$

where θ_i is the positive angle between the i th edge and the x -axis, restricted to the range $0^\circ \leq \theta_i < 180^\circ$ and $0 \leq \delta_i \leq 1$. Note that using edge segments in the definition implies that $D(G)$ needs to be promoted to $D'(G)$.

So that 1 represents drawings with optimal edge deviation with respect to the orthogonal grid (i.e. assumed easier to read), the average edge deviation factor over all edge segments is subtracted from 1. The edge orthogonality aesthetic metric is therefore defined by (Figure 10)

$$\aleph_{eo} = 1 - \frac{1}{m'} \sum_{i=1}^{m'} \delta_i$$

7.2. Node Orthogonality \aleph_{no}

The definition of the node orthogonality metric is motivated by a desire to fix nodes and bend points to intersections on an imaginary unit grid, while making maximal use of the grid area.

The size (in pixels) of the cells in an imaginary grid on which all the nodes of $D'(G)$ lie can be determined by calculating the greatest common divisor (GCD) of the set of vertical and horizontal pixel differences between all geometrically adjacent nodes. After shifting the drawing such that the vertex with the least value co-ordinates is at the origin, a transformation function that divides the co-ordinates of the vertices in $D'(G)$ by GCD can then be used to determine their position on the imaginary grid's gridpoints^c.

The bounding rectangle of the vertices of the transformed $D'(G)$ has integer height and width values b and w , and the number of available grid-point intersections in the imaginary unit grid is therefore

$$A = (w + 1)(b + 1)$$

We define the node orthogonality metric as the extent to which the drawing makes maximal use of the grid area, i.e. the proportion of available gridpoints occupied by

^c Note that this means that the resolution of the grid on which the drawing is based is effectively being reduced.

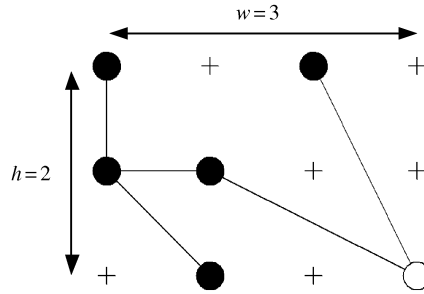


Figure 11. Example of calculating the node orthogonality aesthetic metric when $n' = 6$. There are $A = (w + 1)(h + 1) = (3 + 1)(2 + 1) = 12$ available gridpoints so that $\aleph_{no} = n'/A = 6/12 = 0.5$

the nodes in $D'(G)$ ^f:

$$\aleph_{no} = \frac{n'}{A}$$

Since no two nodes share the same co-ordinates, $n' < A$ and $0 \leq \aleph_{no} \leq 1$ (Figure 11).

8. Upward Flow \aleph_f

This metric determines the proportion of edge segments of $D'(G)$ which have a consistent direction. Edge segments are used rather than edges, as edges with edge segments of alternating direction are generally considered undesirable.

The desired direction is usually upwards or downwards with respect to a vertical axis, but the metric makes no assumptions about its orientation^g. It is assumed that G is a directed graph.

The notation \mathbf{e}_i indicates the vector corresponding to the i th directed edge in $D'(G)$. The inner product of two vectors is denoted $\langle v_1 \cdot v_2 \rangle$. The unit vector parallel with the desired direction is denoted $\mathbf{1}$ and is considered to point in the direction of desired flow.

We define the upward flow metric as

$$\aleph_f = \frac{1}{m'} \sum_{i=1}^{m'} \begin{cases} 1 & \text{if } \langle \mathbf{e}_i \cdot \mathbf{1} \rangle > 0 \\ 0 & \text{otherwise} \end{cases}$$

The metric is 0 for undirected graphs, and is limited to the range $0 \leq \aleph_f \leq 1$.

^f Note that using n' instead of n in the definition of \aleph_{no} means that an increase in the number of bends (such as in a space-filling curve) may increase the node orthogonality value artificially.

^gThe examples in Section 9 below use 'upwards with respect to the vertical axis' as the desired flow direction.

9. Example Application of the Metrics

To demonstrate the application of these metrics, this section presents example graph drawings and associated aesthetic values (see Figures 12 and 13). All the metrics have been implemented in Java, and use the output files of the GraphLet system^h as their input.

The graphs in Figure 12 are from the set of undirected graphs collected by Stephen North at AT&Tⁱ. Duplicate edges between the same two nodes have been removed, and the order of nodes specified in the edge definitions was used to determine edge directionality for the flow aesthetic. These following graphs were selected to cover a variety of graph sizes (Table 1):

Table 1. Node and edges specifications for the North Graphs in Figure 12

	Nodes	Edges
N2	19	21
N5	5	5
N14	24	36
N15	58	87
N19	30	45

The following layout algorithms from the GraphLet system have been used in these examples [7]:

- GEM (spring): based on the algorithm by Frick [8],
- DAG: based on the algorithm by Sugiyama and Misue [9],
- EXT-DAG: an extension of the DAG algorithm,
- ITS (iterative constraint spring, grid based): based on the algorithm by Fruchterman and Reingold [10],
- ITSC (iterative constraint spring, with constraints): an adaptation of the ITS algorithm,
- KAM (spring): based on the algorithm by Kamada and Kawai [11].

The graphs in Figure 13 are created by the GraphLet system itself:

- CIRCLE-4-ITS: circular graph with four nodes, with the ITS algorithm applied,
- COMPLETE-4: complete graph with four nodes,
- CIRCLE-8: circular graph with eight nodes,
- FIBONACCI-4: fibonacci tree with four levels,
- BINARY-4: complete binary tree with four levels.

As expected, the value of the crosses aesthetic is high for all drawings: all the algorithms attempt to minimize the number of edge crosses, and the definition of the metric itself is biased towards high values.

The spring-based drawings are all straight-line drawings, and therefore all have a value of 1 for the bends aesthetic. The bends aesthetic values for the other drawings are also high as the algorithms embody the bends-minimization aesthetic.

^h <http://infosun.fmi.uni-passau.de/Graphlet>.

ⁱ <http://www.research.att.com/sw/tools/graphviz/refs.html>.

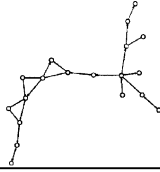





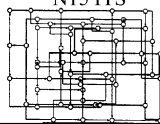
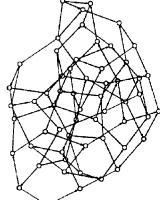
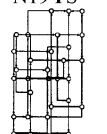

	\aleph_c	\aleph_b	\aleph_s	\aleph_m	\aleph_{eo}	\aleph_{no}	\aleph_f
N2 GEM 	1	1	0.68	0.75	0.57	0	0.62
N2 EX-TDAG 	0.98	0.78	0.52	0.65	0.65	0	0
N5 ITS 	1	0.83	0.64	0.75	1	0	0
N5 DAG 	1	0.83	0.75	0.63	0.33	0.67	0
N14 ITS 	0.96	0.82	0.51	0.64	0.98	0	0.18
N14 ITSC 	0.94	1	0.02	0.43	0.45	0	0.11
N15 ITS 	0.97	0.81	0.57	0.64	1	0	0.20
N15 GEM 	0.95	1	0.01	0.53	0.48	0	0.66
N19 TS 	0.96	0.82	0.55	0.64	1	0	0.20
N19 KAM 	0.93	1	0.01	0.49	0.52	0	0.42

Figure 12. Examples of the application of the aesthetic metrics: North graphs with GraphLet algorithms applied


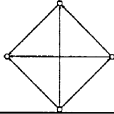
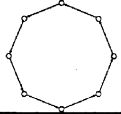


	\aleph_c	\aleph_b	\aleph_s	\aleph_m	\aleph_{eo}	\aleph_{no}	\aleph_f
CIRCLE-4-ITS 	1	1	1	0.5	1	1	0.25
COMPLETE-4 	0.67	1	1	0.37	0.33	0.57	0.67
CIRCLE-8 	1	1	1	0.75	0.5	0.04	0.5
FIBONACCI-4 	1	1	0	0.78	0.57	0.01	0
BINARY-4 	1	1	1	0.77	0.34	0.01	0

Figure 13. Examples of the application of the aesthetic metrics: graphs produced by GraphLet

For the symmetry aesthetic, the grid-based drawings generally perform better than the spring ones: this is surprising, as the depiction of symmetric sub-structures is one of the aesthetics implicitly underlying the spring layout model. It may be that a higher degree of pixel tolerance is required.

The force-directed nature of the spring layout model also implicitly embodies the aesthetic of maximizing the minimum angle between edges leaving a node: while the \aleph_m aesthetic values for these drawings are not low, they are less than those of the grid-based drawings. While the inner nodes of the spring drawings may have appropriate minimum angles, those at the edges are clearly off target; the grid-based algorithms appear to have minimum angles closer to the desired value for all the nodes.

As expected, the grid-based drawings have an edge orthogonality value of 1 (or nearly 1), but it is surprising that most of them have a node orthogonality of 0. The actual \aleph_{no} values for these grid-based diagrams were in the order of 10^{-4} , with the underlying grid mirroring individual pixels. The simpler drawings in Figure 13 demonstrate more effective use of node orthogonality. It may be that a degree of tolerance, measured in pixels, should be used in determining the underlying grid such that exact pixel correspondence between vertices and the grid-points is not required (similar to that used in the symmetry aesthetic).

The values of the flow aesthetic were calculated with respect to a desired direction of ‘upwards’: values of 0 indicate that all the edges point downwards. As expected, the spring diagrams do not perform as well on this aesthetic as do the grid algorithms.

10. Conclusion

In an attempt to create a quantifiable method for assessing the aesthetic quality of any graph drawing, metrics for seven common aesthetic criteria have been defined. These metrics will be useful both for formally analysing the aesthetic quality of graph drawings

produced by different algorithms, and for measuring the status of intermediate drawings produced in iterative layout methods like genetic algorithms or simulated annealing. The metrics are continuous, and can therefore be used to investigate the extent to which a drawing needs to conform to an aesthetic, rather than always insisting on an extreme. Application of the metrics to graphs drawn with common layout algorithms reveal that assumptions about the aesthetics underlying some algorithms are not valid when they are computationally measured, and that a computational degree of tolerance is required in geometric matching of co-ordinates if any relationship to the effects of perception are to be taken into account.

Acknowledgements

This work has benefited from the contributions of David Leonard (who assisted with the original metric definitions), Daniel Choi and Steven Yong Tan (who implemented them), and Matthew McGill (who assisted with their presentation and revision). I am also grateful to Peter Eades for his extensive comments on earlier drafts of this paper. The Australian Research Council funded this work, and the Glasgow University Computing Science Department provided the facilities and environment, which enabled it to come to fruition.

References

1. G. Di Battista, P. Eades, R. Tamassia & I. Tollis (1994) Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications* **4**, 235–284.
2. H. C. Purchase (1997) Which aesthetic has the greatest effect on human understanding? In: *Proceedings of Graph Drawing Symposium 1997* (G. Di Battista, ed.), Lecture Notes in Computer Science, Vol. 1353, Springer-Verlag, Berlin, pp. 248–259.
3. S. Bridgeman & R. Tamassia (2000) Difference metrics for interactive orthogonal graph drawing algorithms. *Journal of Graph Algorithms and Applications* **4**, 47–74.
4. J. Pach & G. Toth (1998) What crossing number is it anyway? In: *Proceedings of Symposium on Foundations of Computer Science*. IEEE, New York, pp. 617–627.
5. H. F. Jensen (1971) An upper bound for the rectilinear crossing number of the complete graph. *Journal of Computational Theory, Series B* **10**, 212–216.
6. F. Shahrokhi, O. Sykora, L. Szekely & I. Vrto (1997) Crossing numbers: bounds and applications. In: *Intuitive Geometry*, Bolyai Society Mathematical Studies, Akademiai kiado, Budapest, Vol. 6, pp. 179–206.
7. F. Brandenburg (2000) Personal Communication, November 2000.
8. A. Frick, A. Ludwig & H. Mehldau (1994) A fast adaptive layout algorithm for undirected graphs. In: *Proceedings of Graph Drawing Symposium 1994* (R. Tamassia & I. Tollis, eds), Lecture Notes in Computer Science, Vol. 894, Springer-Verlag, Berlin, pp. 388–403.
9. K. Sugiyama & K. Misue (1991) Visualization of structural information: automatic drawing of compound diagraphs. *IEEE Transactions on Software Engineering* **21**, 876–892.
10. T. Fruchterman & E. Reingold (1991) Graph drawing by force-directed placement. *Software—Practice and Experience* **21**, 1129–1164.
11. T. Kamada & S. Kawai (1989) An algorithm for drawing general undirected graphs. *Information Processing Letters* **31**, 7–15.