A Symmetry Metric for Graphs and Line Diagrams

Roman Klapaukh[1,2], Stuart Marshall[2], and David Pearce[2]

[1] Research IT Services, University College London, London NW1 2DN, United Kingdom
r.klapaukh@ucl.ac.uk
[2] School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand

Abstract. Symmetry is often considered a desirable feature of diagrams. However, quantifying the exact amount of symmetry present is often difficult. We propose a novel symmetry metric that can score the amount of rotational, translational, and reflective symmetry present in a graph or line diagram.

Keywords: Symmetry Detection, Line Diagrams, Diagram Evaluation

1 Introduction

Symmetry is a property of visual layouts that is frequently considered to be desirable. Many believe greater symmetry improves understandability and that, for example, force-directed layout promotes symmetry [1, 2]. In the study of Kieffer *et al.*, human subjects showed a preference for graphs with reflectional symmetry [3]. Likewise, Marriot *et al.* confirmed that various layout features — including symmetry — make graphs more memorable [4].

For graphs, Purchase defined an algorithm for computing a symmetry score and used this to test various claims [5]. Purchase's symmetry algorithm has two important limitations which this paper attempts to address. First it focused only on the symmetries of vertices, but ignored edges. Second it measured only reflective symmetries, ignoring rotational and translational symmetries.

In this paper we develop a new symmetry metric for straight line diagrams. Our symmetry metric is an extension of that described by Loy and Eklundh [6] which extracts reflective, rotational and translational symmetries of feature points from photographs. We adapt their algorithm to work on known vector lines, rather than feature points detected in raster images. Unlike points, lines can be symmetrical with respect to themselves, and for the line drawing we consider you have perfect knowledge about the lines. This holds true for images stored in vector formats (e.g. SVG).

2 Background

Purchase [5] defined a symmetry metric for graphs that measured how much reflective symmetry was present between vertices. However, an important limi-

tation of Purchase's algorithm is that it does not consider edges when calculating a symmetry score. In all connected graphs which are not trees, edges will outnumber vertices. Additionally, edges are drawn as lines connecting the vertices. This results in the edges having a lot of influence on visual symmetry within the graph. Therefore, in this work we focus on quantifying the symmetry of edges.

We extend the symmetry detection method of Loy and Eklundh as it is works in 2D and can identify rotational, translational, and reflective symmetries [6]. However, Loy and Eklundh focus on identification of symmetry for points that have been extract from photographs, in order to identify regions of symmetry. However, as we consider lines, which are symmetrical with respect to themselves, we develop a new method following their process.

An important feature of Loy and Eklundh's algorithm is that it can detect when multiple different detected symmetry axes / centres are actually very similar (i.e. rather than only when they are identical), and groups them together into a single axis. This is very useful in generated diagrams as they may have minor imperfections which should not be penalised.

3 Symmetry metric

Our extended symmetry metric is described in Algorithm 1. The rest of this section describes each part of the algorithm. The implementation can be found as part of our open source graph analysis library [7].

The general idea of the algorithm is that it finds every possible axis of mirror, translational, or rotational symmetry that is present in the diagram. It then votes to identify which axes affect the largest number of lines. The number of axes detected is a user defined parameter N. Note that the score for each type of symmetry is computed separately.

We start with an empty list of pairs of an axis (two floating point numbers) and its quality score (a number between 0 and 1). Each kind of symmetry has a different sort of axis: rotational symmetry has a centre of rotation, translational symmetry has a direction vector, and reflective symmetry has the Hough transform [8] of the mirror line.

We then convert the lines into a standard format for easy manipulation. Following Loy and Eklundh [6] we convert them into Scale Invariant Feature Transform (SIFT) [9] features. Each SIFT feature is a four-tuple consisting of a location (centre of the line), orientation (angle between the line and x-axis in degrees), scale (length of the line in pixels), and identifying characteristics (always 1).

The first set of axes we generate are those that are symmetries of an axis with itself. For reflective symmetry there are two axes that can be generated from a single line: its perpendicular bisector, and the line itself. For rotational symmetry there is one: the position of the line, as a line can be spun around its centre 180 degrees to get the same line. There are no such axes for translational symmetry. In each the symmetry is perfect, therefore the quality score for all

Algorithm 1: Symmetry metric.

```
Data: symmetryType ∈ {reflective,rotational,translational} & N the number of
      axes to find
Result: score ∈ [0..1]
axes = empty list
features = Convert all edges to SIFT features
for f_i ∈ features do
    if symmetryType == reflective then
        axes.add(perpendicularBisector(f_i), 1)
        axes.add(parallelAxis(f_i), 1)
    else if symmetryType == rotational then
        axes.add(getLocation(f_i), 1)
foreach f_i, f_j ∈ features do
    axis = find symmetry axis(symmetryType, f_i, f_j)
    quality = find symmetry quality(symmetryType, f_i, f_j)
    axes.add(axis, quality)
axes = quantiseAxes(axes)
bestAxes = pickBest(axes, N)
score = score(bestAxes)
return score
```

these axes is 1. Note that these single feature axes of symmetry are not present in Loy and Eklundh's algorithm.

All other possible axes of symmetry can be generated by calculating the axes of symmetry between every pair of lines. The quality score for each axes is the product of its scale quality (S_{ij}) and orientation quality (Φ_{ij}) scores. Each score is bound by $[0 \dots 1]$.

The scale quality (S_{ij}) is the same for all symmetry types and is the same as Loy and Eklundh's original paper [6]. In the equation S_{ij} is the scale similarity, s_k is the length of line k, and σ_s is a scaling factor (sensitivity).

$$S_{ij} = \left(e^{\frac{-|s_i - s_j|}{\sigma_s (s_i + s_j)}} \right)^2 \tag{1}$$

The axis of reflective symmetry is the perpendicular bisector of the line between the line centres. The orientation quality (Φ_{ij}), is adapted from Reisfeld et al. [10], with consideration for lines being symmetrical after 180 degree rotations.

$$\Phi_{ij} = |\cos(\theta_i + \theta_j - 2 * \theta_{ij})| \tag{2}$$

For translational symmetry the required translation is the vector difference in the position of the features. This needs to be normalised (multiplied by -1 if dy < 0) to compensate for ordering. For translational symmetry the orientation quality has to be adjusted as the lines are not mirrored.

$$\Phi_{ij} = |\cos(\theta_i - \theta_j)| \tag{3}$$

For each pair of features there can be up to two centres of rotational symmetry. This is because there are two ways to line up the feature orientations, head to head and head to tail, each of which may require its own centre. When $\theta_1 = \theta_2 = 180$ both centres will be the same. This is different from the original paper where there was only one possible centre, and is a result of a line being indistinguishable after a 180 degree rotation. The orientation metric score is always 1.

Having enumerated all of the axes of symmetry, the quality scores are now used to vote to find the N best axes. We quantise the space, to deal with minor deviations in location, and sum the symmetry scores for each distinct axis. The N axes with the highest total scores are the chosen axes.

The final stage of the algorithm is to turn the set of N best axes found into a number that can be used as a metric. We use the following equation:

$$\left(\frac{\sum_{axes} \text{number of lines that voted for this axis}}{N \times \text{number of lines}} \right)$$

4 Conclusion

We developed a novel metric to evaluate how symmetrical a given line diagram is with respect to reflective, rotational and translational symmetries.

References

1. Eades, P.: A Heuristic for Graph Drawing. Congressus Numeratum **42** (1984) 149–160
2. Purchase, H., Cohen, R., James, M.: Validating graph drawing aesthetics. In: Graph Drawing. Springer (1996) 435–446
3. Kieffer, S., Dwyer, T., Marriott, K., Wybrow, M.: HOLA: Human-like orthogonal network layout. IEEE TVCG **22**(1) (2016) 349–358
4. Marriott, K., Purchase, H., Wybrow, M., Goncu, C.: Memorability of visual features in network diagrams. IEEE TVCG **18**(12) (2012) 2477–2485
5. Purchase, H.C.: Metrics for Graph Drawing Aesthetics. JVLC **13**(5) (2002) 501–516
6. Loy, G., Eklundh, J.O.: Detecting symmetry and symmetric constellations of features. In: Proc. Computer Vision. Springer (2006) 508–521
7. Klapaukh, R.: GraphAnalyser https://github.com/klapaukh/GraphAnalyser accessed: 20 Dec 2013.
8. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1) (January 1972) 11–15
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. Computer Vision. (1999) 1150–1157
10. Reisfeld, D., Wolfson, H., Yeshurun, Y.: Context-free attentional operators: The generalized symmetry transform. Int. J. Computer Vision **14**(2) (1995) 119–130