SDE: Graph Drawing Using Spectral Distance Embedding[⋆]

Ali Civril, Malik Magdon-Ismail, and Eli Bocek-Rivele

Computer Science Department, RPI, 110 8th Street, Troy, NY 12180
{civria, magdon, boceke}@cs.rpi.edu

Abstract. We present a novel graph drawing algorithm which uses a spectral decomposition of the distance matrix to approximate the graph theoretical distances. The algorithm preserves symmetry and node densities, i.e., the drawings are aesthetically pleasing. The runtime for typical $20,000$ node graphs ranges from 100 to 150 seconds.

1 Introduction

The graph drawing problem is to compute an aesthetically pleasing layout of vertices and edges so that it is easy to grasp visually the inherent structure of the graph. A general survey can be found in [4]. We consider only straight-line edge drawings, which reduces to embedding the vertices in two dimensions.

Spectral graph drawing, which has become popular recently [2, 3], produces a layout using the spectral decomposition of some matrix related to the vertex and edge sets. We present a spectral graph drawing algorithm SDE (Spectral Distance Embedding), in which we use the spectral decomposition of the graph theoretical distance matrix. We present the results of our algorithm through several examples, including run-times. Compared to similar techniques, we observe that our results achieve superior drawings, while at the same time not significantly sacrificing on computation time. The details of the algorithms, implementations and performance analysis can be found in an accompanying technical report [1].

The two stages in the algorithm are: (i) computing all-pairs shortest path (APSP) lengths; (ii) finding a rank-2 approximation to the matrix of squared distances $\mathbf{L}$. Step (i) involves a BFS for each node. For (ii), we use a standard procedure referred to as the power iteration to compute the eigenvalues and eigenvectors of $\mathbf{M} = -\frac{1}{2}\boldsymbol{\gamma}\mathbf{L}\boldsymbol{\gamma}$, where $\boldsymbol{\gamma} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$. The complexity of the algorithm is $O(|V||E|)$, using $O(|V|^2)$ space (to store all the pair-wise distances). The algorithm is summarized below (PowerIteration returns the final coordinates specified by the top 2 eigenvectors of the input matrix, to a precision specified by ϵ). Let $\mathbf{D}$ be the matrix of distances.

```
SDE(G) 1: Use an APSP algorithm to compute $\mathbf{L}$, where $L_{ij} = D_{ij}^2$.
       2: return $\mathbf{Y}$ = PowerIteration($-\frac{1}{2}\boldsymbol{\gamma}\mathbf{L}\boldsymbol{\gamma}, \epsilon$) % epsilon is a tolerance.
```

[⋆] This material is based upon work partially supported by the National Science Foundation under Grant Nos. 0323324, 0324947.

P. Healy and N.S. Nikolov (Eds.): GD 2005, LNCS 3843, pp. 512–513, 2005.

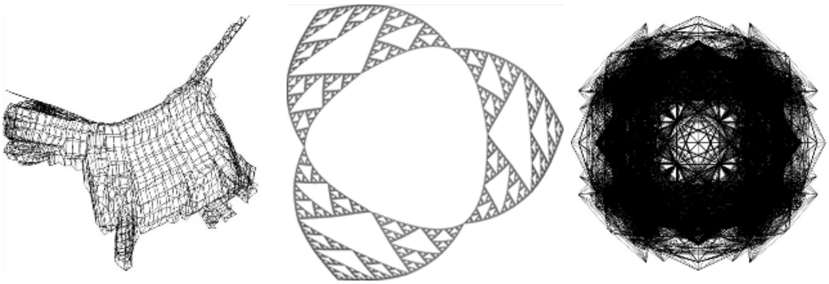
2 Results and Conclusion

The table below shows that SDE is reasonably fast for graphs up to 20,000 nodes. As can be seen from the following figures, it also produces aesthetically pleasing drawings of graphs varying in size, node density and degree of symmetry. Our algorithm has the advantages of exact (as opposed to iterative) computation of spectral graph drawing techniques and the quality of slower force-directed methods.

Graph	$\lvert V \rvert$	$\lvert E \rvert$	**APSP time**	**Power Iteration time**	**Total time**
jagmesh1	936	2664	0.10	0.11	0.21
Grid 50x50	2500	4900	0.65	0.54	1.19
3elt	4720	13722	4.67	3.80	8.47
whitaker3	9800	28989	25.24	8.18	33.42
sphere	16386	49152	106.96	29.73	136.69

The justification for SDE stems from the following (heuristically stated) theorem,

Theorem 1 (Theorem 3, [1]). *When the distance matrix is nearly embedable and satisfies some regularity conditions, SDE recovers (up to rotation) a close approximation to the optimal embedding.*



(a) Cow (b) Sierpinski 8 (c) Vibrobox

References

1. A. Civril, M. Magdon-Ismail, and E. B. Rivele. SDE: Graph drawing using spectral distance embedding. Technical report, Rensselaer Polytechnic Institute, 2005.
2. D. Harel and Y. Koren. Graph drawing by high-dimensional embedding. In *GD02*, LNCS. Springer-Verlag, 2002.
3. Y. Koren, D. Harel, and L. Carmel. Drawing graphs by algebraic multigrid optimization. *Multiscale Modeling and Simulation*, 1(4):645–673, 2003. SIAM.
4. I. G. Tollis, G. D. Battista, P. Eades, and R. Tamassia. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, 1999.