Spectral Sequencing Based on Graph Distance

Rong Liu, Hao Zhang, and Oliver van Kaick

GrUVi Lab, School of Computing Sciences,
Simon Fraser University, Burnaby, British Columbia, Canada
{lrong, haoz, ovankaic}@cs.sfu.ca

Abstract. The construction of linear mesh layouts has found various applications, such as implicit mesh filtering and mesh streaming, where a variety of layout quality criteria, e.g., span and width, can be considered. While spectral sequencing, derived from the Fiedler vector, is one of the best-known heuristics for minimizing width, it does not perform as well as the Cuthill-Mckee (CM) scheme in terms of span. In this paper, we treat optimal mesh layout generation as a problem of preserving graph distances and propose to use the subdominant eigenvector of a kernel (affinity) matrix for sequencing. Despite the non-sparsity of the affinity operators we use, the layouts can be computed efficiently for large meshes through subsampling and eigenvector extrapolation. Our experiments show that the new sequences obtained outperform those derived from the Fiedler vector, in terms of spans, and those obtained from CM, in terms of widths and other important quality criteria. Therefore, in applications where several such quality criteria can influence algorithm performance simultaneously, e.g., mesh streaming and implicit mesh filtering, the new mesh layouts could potentially provide a better trade-off.

1 Introduction

Computing linear mesh layouts is an instance of the graph layout problem [1], where an optimal labeling of the vertices of a given graph is sought. Many optimization problems, including sparse matrix reordering [2–4], circuit layout [5], DNA sequencing [6], and ranking [7] , are formulated as graph layout problems.

Consider a weighted graph $G = (V, E, w)$ with $V = \{v_1, \ldots, v_n\}$ the set of vertices, E the set of edges, and $w : E \rightarrow \mathbb{R}$ the edge weights. A (linear) *layout* of G is a labeling π of its vertices, $\pi : V \rightarrow \{1, 2, \ldots, n\}$. For a real number $0 < p < \infty$, the *p-discrepancy* [7] of G with respect to a layout π is defined as $\sigma_p(G, \pi) = \left(\sum_{uv \in E} w_{uv} |\pi(u) - \pi(v)|^p\right)^{1/p}$. If $p = \infty$, then $\sigma_\infty(G, \pi) = \max_{uv \in E} |\pi(v) - \pi(v)|$, and is also called the *bandwidth* of the layout. The minimum value $\sigma_p(G) = \min_\pi \sigma_p(G, \pi), 0 < p \leq \infty$, is called the *min-p-sum* of the graph G. Another important layout cost measure is *vertex separation* [1], defined as $\max_{1 \leq i \leq n} |\{\pi(u) \leq i : \exists \pi(v) > i, uv \in E\}|$. Intuitively, it measures, at a certain point of the linear layout, the number of edges for which only one

end vertex has been encountered. In the field of numerical analysis, various measures, such as bandwidth [2], *profile* or *envelope size* [3] and *workbound* [4], are considered for sparse matrix reordering. It turns out that these measures are related to different p-discrepancies of a graph layout, where the matrix of interest can be considered as the adjacency matrix of the graph G.

Several problems in geometry processing [8, 9] benefit from having an optimized mesh layout. A good example is mesh streaming [10], where the *span* and *width*, corresponding to bandwidth and vertex separation, respectively, heavily influence the mesh streamability. Profile, workbound, and bandwidth all affect the cost of Cholesky factorization [11], which becomes necessary when large sparse linear systems for the implicit mesh filtering problem need to be solved [9].

The optimization problems associated with most of the criteria mentioned so far are NP-hard [1]. In practice, one resorts to efficient heuristics. Two best known heuristics for minimizing span (bandwidth) and having comparable performances are the Cuthill-Mckee (CM) [2] and minimum degree schemes [11], both essentially conduct a degree-oriented breadth-first search. For the other costs mentioned so far, spectral sequencing using the Fiedler vector, which is the eigenvector corresponding to the smallest non-zero eigenvalues of the graph Laplacian matrix, has been quite successful empirically [3, 7]. Note that the graph Laplacian is derived from the graph's adjacency matrix.

In this paper, we propose a new spectral sequencing operator for mesh layout generation based on *graph distance*. The sequences obtained using our algorithm outperform those derived from the Fiedler vector, in terms of spans, and sequences obtained from CM, in terms of widths, profiles, workbounds, and a few other important layout quality criteria. Thus in applications where several such criteria can influence algorithm performance simultaneously, e.g., mesh streaming and implicit mesh filtering, the new mesh layouts could potentially provide a better trade-off. Although the matrix we eigendecompose is in general non-sparse, approximate eigenvector computation via subsampling and Nyström method [12] allows us to compute layouts for large meshes efficiently.

The rest of the paper is organized as follows. In Section 2, we describe our spectral sequencing algorithm. Various practical issues, such as subsampling for efficient layout computation, are addressed in Section 3. Experimental results are given in Section 4. Finally, we conclude and suggest possible future work.

2 Spectral sequencing for graph layout

Although we focus on mesh layouts, our discussions are cast in the general context of graph layout. To produce a sequence of mesh vertices or faces, the graph of interest can be chosen as the primal or dual graph of the mesh, respectively. For the mesh layout problem we currently consider, edge weights will be assumed to be unit. Our approach however can trivially adapt to weighted graphs.

Independent of any specific layout costs, observe that if two graph vertices are close to each other based on connectivity, they should also be close to each other in the sequence. Take the *min-1-sum* problem for example. If all the adjacent

vertices in the graph were consecutive in the sequence, the closest possible in 1-D, an optimal layout would be obtained. This is of course not possible in 1-D in general, but we can look for an embedding ϕ of the graph vertices in a high dimensional space, in which adjacent vertices are always close to each other. To obtain a sequence, the vertices are projected onto a properly chosen vector, so that their mutual distances are preserved as mush as possible. To summarize, given a graph $G = (V, E)$ with n vertices, our algorithm works as follow.

1. Calculate the graph distance, $g(v_i, v_j)$, between each pair of vertices.
2. Compute an embedding $\phi(V) = \{\phi(v_i)\}$ of V, such that given any vertex v, $\forall(i, j), ||\phi(v) - \phi(v_i)||^2 < ||\phi(v) - \phi(v_j)||^2$, if and only if $g(v, v_i) < g(v, v_j)$.
3. Project points in $\phi(V)$ onto a vector $\mathbf{p}^*$, along which their relative positions are preserved as much as possible.
4. Sort the projections of $\phi(V)$ on $\mathbf{p}^*$ to obtain a sequencing of the vertices.

2.1 Graph distance and the kernel matrix

To find the embedding $\phi(V)$, we resort to relevant concepts from Kernel PCA [13]. Imagine a mapping $\phi : V \to \mathcal{F}, v \mapsto \phi(v)$, where $\mathcal{F}$ is typically called the *feature space* which may have a very high, and possibly infinite, dimensionality. For this reason, ϕ is never explicitly used and instead, it is implicitly defined by a kernel (affinity) matrix K, with $K_{ij} = k(v_i, v_j) = \langle \phi(v_i), \phi(v_j) \rangle$. Thus if K is known, the embedding $\phi(V)$ is implicitly induced. To define K, we first form a distance matrix W, where $W_{ij} = g(v_i, v_j)$ is the graph distance between the vertices.

The next step is to convert W into the kernel matrix K, using a certain kernel function. One of the most popular kernels is the Gaussian radial basis function,

$$k(v_i, v_j) = \exp\left(W_{ij}^2/2\delta^2\right), \text{where } \delta \text{ is the kernel width.} \tag{1}$$

In this case, it can be shown [14] that the square distance between two vertices i and j in the feature space $\mathcal{F}$ is $\mathcal{W}_{ij}^2 = ||\phi(v_i) - \phi(v_j)||^2 = 2 - 2k(v_i, v_j)$. Thus distances in $\mathcal{F}$ are seen to be proportional to those in the graph, in that the order among distances between vertices in the graph is preserved in the feature space. Specifically, since the graph edges have unit length, neighboring vertices in the graph are always closest to each other in the feature space $\mathcal{F}$.

2.2 Sequencing via spectral embedding

The positioning of the embeddings $\phi(V)$ provides a good start to extract a vertex sequence since the distances between pairs of vertices in the graph are relatively preserved in $\mathcal{F}$. We simply project each $\phi(v)$ onto a vector $\mathbf{p}$. Afterwards, the projections are sorted to obtain the sequence. Since after the projection the dimensionality of the embedding space is reduced to 1, to obtain a smaller layout cost for the resulting sequence, it is desirable to preserve their mutual distances, hence the relative positions, as much as possible. To this end, we choose the optimal direction $\mathbf{p}^*$ subject to the following objective function

$$\mathbf{p}^* = \underset{\mathbf{p} \in \mathbb{R}^{dim(\mathcal{F})}, ||\mathbf{p}||=1}{\operatorname{argmax}} \sum_{i<j} ||\mathbf{p}^T(\phi(v_i) - \phi(v_j))||^2,$$

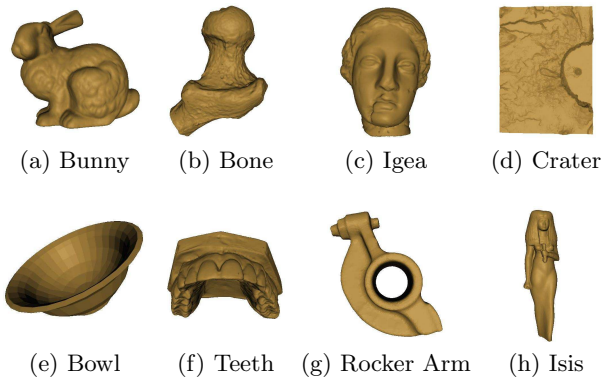
To motivate this approach, let us note that as the projection of distances onto $\mathbf{p}$ is always smaller than the corresponding original distances, maximizing the sum of projected distances tends to preserve the original distances. Assume that $\phi(V)$ is centered around the origin, it can be proven that $\mathbf{p}^*$ is the first principal component of the point set $\phi(V)$. Denote by $\phi(V)_{\mathbf{p}^*}$ the projection of $\phi(V)$ onto $\mathbf{p}^*$, it can be demonstrated that the sequence obtained from $\phi(V)_{\mathbf{p}^*}$ is identical to that obtained from $\mathbf{U}_1$, the largest eigenvector of the kernel matrix K. Therefore the algorithm simply sorts the elements of $\mathbf{U}_1$ to produce the sequence for V for the layout problem. More details on these and our subsequent discussions can be found in an extended version of this paper in [14].

3 Some practical issues

Subsampling: The algorithm described so far is unable to handle large meshes due to overhead caused by pair-wise distance computation required by W and the eigenvalue decomposition of K, giving rise to a time complexity of at least $O(n^2 \log n)$. To overcome this complexity, we adopt subsampling and the Nyström method [15], which allows for computing only the distances between m sampled vertices to the remaining vertices and approximating the leading eigenvectors of K by knowing only m rows of it. Throughout this paper, we fix the number of samples to 10, reducing the overall complexity to $O(n \log n)$. With farthest point sampling [15], our experiments demonstrate that Nyström approximation works remarkably well, even at an extremely low sampling rate.
Centering of K: Till now we have assumed that the set $\phi(V)$ is centered at the origin, which is not true in general. Thus, the kernel K has to be first centered [13] to obtain $\bar{K}$, where $\bar{K} := (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)K(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$ and $\mathbf{1}$ is the column vector of 1's. However, this is inefficient for large K. At the same time, due to subsampling, the full K is unavailable for centering. We thus propose to use the *subdominant eigenvector* of the un-centered K, which can be both formally and experimentally shown [14] to be a good approximation to the dominant eigenvector of $\bar{K}$, as long as the kernel width δ used in (1) is sufficiently large. In our experiments, we choose δ as the average of all the sampled graph distances.
Choice of kernel function: To convert a distance matrix to a kernel matrix, we use the Gaussian kernel, given in equation (1). Note that other kernels, e.g., step function, exponential, polynomial and rational polynomial kernels, are also possible. In fact, the use of the Fiedler vector can be seen as a special case of our general paradigm, when applied to regular graphs, since the eigenvectors of the graph Laplacian coincide with the eigenvectors of K derived from distance matrix W with a step function kernel of width 1. We have experimented with other kernels and have not found particular reasons to prefer one over the other, but this issue requires further investigation.

4 Experimental results

This section presents an experimental comparison which evaluates the quality of the sequence generated by the Fiedler vector, the Cuthill-Mckee scheme and



(a) Bunny (b) Bone (c) Igea (d) Crater

(e) Bowl (f) Teeth (g) Rocker Arm (h) Isis

Fig. 1. Models used in the experiments.

our approach, referred to as Laplacian, CM, and Affinity, respectively. Given a mesh, its vertex or face sequence is generated by considering the primal or the dual graph, respectively. In our experiments, we consider six quality measures of the sequence: span, width, profile, workbound, 1-discrepancy and 2-discrepancy, where the first two are of particular interest to mesh streaming [10].

Figure 1 shows the triangle meshes used in our experiments, which are performed on a Pentium 1.7GHz processor with 1GB RAM. Note that models with boundary (Crater), elongated aspect ratio (Isis) and non-zero genus (Rocker Arm) are all tested. Table 1 presents the characteristics of these models in conjunction with the timing of our algorithm. Note that CM works the fastest among the three, while Affinity and Laplacian perform similarly in speed.

Table 1. Characteristics of models and timing in seconds, I/O excluded.

Model	Bunny	Bone	Igea	Crater	Bowl	Teeth	Rocker Arm	Isis
Vertex #	34,834	50,002	60,002	100,000	102,402	100,002	160,704	187,644
Time	2.00	3.14	4.20	7.31	7.45	7.58	15.50	23.80

Figure 2 shows the comparison between the six layout measures for the primal graphs; more experimental results can be found in [14]. It can be seen that CM obtains the best results in terms of span, while the Laplacian operator generally provides the best results in terms of the other measures. However, since the Affinity operator outperforms the Laplacian in span and CM in width, it achieves a trade-off between these two sets of measures, thus providing potential benefits to applications where all these measures influence performance. Since the models we use have different structural properties, we believe our result is model independent. We have also experimented on dual graphs (for face sequencing) and found that Affinity tends to outperform both CM and Laplacian on regular graphs [14]. This issue is currently under investigation.

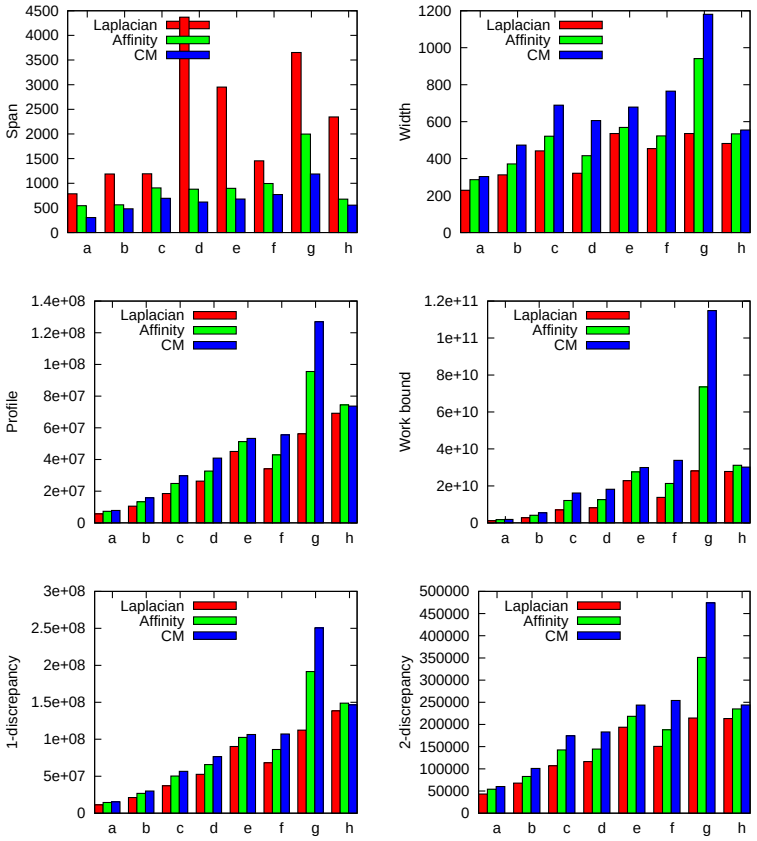


Fig. 2. Comparison of layout quality measures for primal graph of different models: (a) Bunny (b) Bone, (c) Igea, (d) Crater, (e) Bowl, (f) Teeth, (g) Rocker Arm, (h) Isis.

Note that in all our experiments, the sample size is fixed at 10, which is as small as 0.0053% of the mesh size. We also conduct experiments to compare between using and not using subsampling and report the results in [14].

5 Conclusion and future work

In this paper, we present a spectral sequencing algorithm for linear mesh layout. We study this problem in the more general context of linear graph layout. Independent of any specific cost measure, we abstract that a desirable vertex sequence should be one in which close-by vertices in the graph are also close-by in the sequence. To this end, a high dimensional distance preserving embedding and 1D projection are applied to extract the sequence, with the help of Kernel PCA. To make the algorithm work efficiently for large meshes, several practical issues are addressed, including subsampling, eigenvector extrapolation, and the usage of the subdominant eigenvector of the un-centered kernel matrix.

Our experiments demonstrate that for span and width, the two principal quality measures for mesh streaming, as well as other graph layout cost criteria, e.g., profile and workbound, our algorithm potentially provides a better trade-off compared to ordering schemes based on localized graph traversal, e.g., Cuthill-Mckee, and spectral sequencing using the Fiedler vector.

The geometric appeal of our framework and analysis based on Kernel PCA has shed some light on possible further improvement of layout qualities. One particularly intriguing problem is to investigate the distribution properties of the vertex embedding in the feature space, so as to achieve better results in both span and width. It is also interesting to investigate the role of kernel functions in the layout result, as well as more sophisticated distance measures between graph vertices that take into consideration more global mesh connectivity information.

References

1. Díaz, J., Petit, J., Serna, M.: A survey of graph layout problems. ACM Computing Survey **34**(3) (2002) 313–356
2. Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proc. 24th Nat. Conf. ACM. (1969) 157–172
3. Barnard, S.T., Pothen, A., Simon, H.D.: A spectral algorithm for envelope reduction of sparse matrices. In: Proc. ACM/IEEE Conference on Supercomputing. (1993) 493–502
4. Corso, G.M.D., Romani, F.: Heuristic spectral techniques for the reduction of bandwidth and work-bound of sparse matrices. Technical report, Universitá di Pisa, Dipartimento di Informatica (2001)
5. Hur, S.W., Willis, J.: Relaxation and clustering in a local search framework: application to linear placement. In: Proc. ACM/IEEE Conference on Design Automation. (1999) 360–366
6. Karp, R.M.: Mapping the genome: some combinatorial problems arising in molecular biology. In: Proc. ACM Symposium on Theory of Computing. (1993) 278–285
7. Mohar, B., Poljak, S.: Eigenvalues in combinatorial optimization. In Brualdi, R.A., Friedland, S., Klee, V., eds.: IMA Volumes in Mathematics and Its Applications. Volume 50. Springer-Verlag (1993) 107–151
8. Isenburg, M., Lindstrom, P., Gumhold, S., Snoeyink, J.: Large mesh simplification using processing sequences. In: Proc. of the 14th IEEE Visualization. (2003) 61–68
9. Zhang, H., Fiume, E.: Butterworth filtering and implicit fairing of irregular meshes. In: Proceedings of Pacific Graphics. (2003) 502–506
10. Isenburg, M., Lindstrom, P.: Streaming meshes. In: IEEE Visualization. (2005)
11. George, A., Liu, W.H.: Computer Solution of Large Sparse Positive Definite Systems. Prentice-Hall (1981)
12. Press, W., Tekolsky, S., Vetterling, W., Flannery, B.: Numerical Recipies in C. Cambridge Univ. Press (1992)
13. Schölkopf, B., Smola, A., Muller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10** (1998) 1299–1319
14. Liu, R., Zhang, H., van Kaick, O.: An investigation on spectral sequencing using graph distance. Technical Report TR 2006-08, School of Computing Science, Simon Fraser University (2006)
15. Liu, R., Jain, V., Zhang, H.: Subsampling for efficient spectral mesh processing. In: Proceedings of Computer Graphics International 2006 (to appear). (2006)