AA220/CS238 Handout: Inertial Partitioning for 3D Nodal Graphs

Inertial graph partitioning for three-dimensional graphs with associated nodal coordinates is very similar to the two-dimensional example that was described in the class notes. The following notes may help in the implementation of the algorithm for your third assignment. All quantities in the following derivation refer to the symbols in Figure 1 below. The derivation here is a little different from the one in the notes, but this way you see a slightly different point of view.

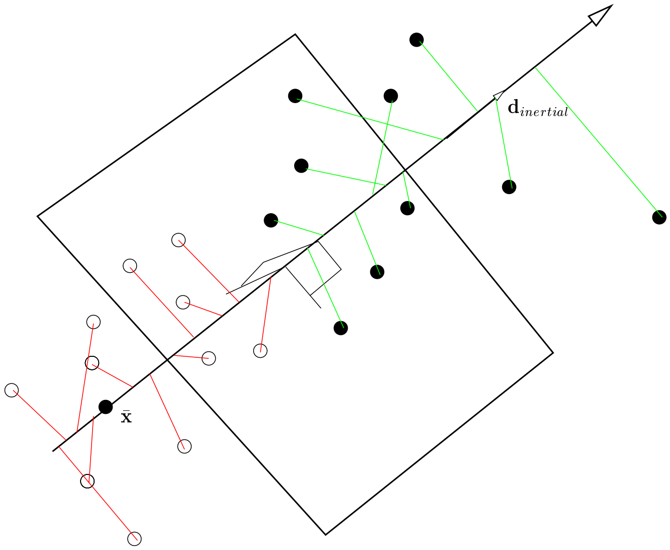

Figure 1: Inertial Graph Partitioning for Three-Dimensional Graphs

Based on the observation that a rotating object has minimal moment of inertia when rotating about its *long* axis (if it is reasonably compact), inertial partitioning treats each vertex as a point mass and calculates this direction, $\mathbf{d}_{inertial}$. This direction is used to create a *separator* plane with $\mathbf{d}_{inertial}$ as its normal vector that leaves half the nodes of the original partition on one side, and the other half on the other.

The calculation of $\mathbf{d}_{inertial}$ is not a computationally expensive one; the moment of inertia tensor $\mathbf{I}$ is formed by accumulating the contributions from each vertex. We then solve for its smallest eigenvalue, and the corresponding eigenvector is taken as $\mathbf{d}_{inertial}$ (which effectively points in the direction of the principal axis of the $\mathbf{I}$ tensor with minimum inertia). The inertia tensor is a 3×3 matrix, and its eigenvalues and eigenvectors can be found through any method (in particular one can use the LAPACK DSYEV and SSYEV routines).

The contribution of vertex v_i to the moment of inertia tensor is, in three dimensions

$$\mathbf{I}_i = \begin{pmatrix} \hat{y}_i^2 + \hat{z}_i^2 & -\hat{x}_i\hat{y}_i & -\hat{x}_i\hat{z}_i \\ -\hat{y}_i\hat{x}_i & \hat{x}_i^2 + \hat{z}_i^2 & -\hat{y}_i\hat{z}_i \\ -\hat{z}_i\hat{x}_i & -\hat{z}_i\hat{y}_i & \hat{x}_i^2 + \hat{y}_i^2 \end{pmatrix}$$

where $\hat{\mathbf{x}}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i) = \mathbf{x}_i - \bar{\mathbf{x}}$ and $\bar{\mathbf{x}}$ is the is the center of mass of the system given by $\bar{\mathbf{x}} = \sum_{i=1}^{n_v} \mathbf{x}_i / n_v$, where n_v is the total number of vertices in the graph.

$\mathbf{I}$ is then the sum of the contributions of all the vertices

$$\mathbf{I} = \sum_{i=1}^{n_v} \mathbf{I}_i = \begin{pmatrix} \sum_i \hat{y}_i^2 + \hat{z}_i^2 & -\sum_i \hat{x}_i\hat{y}_i & -\sum_i \hat{x}_i\hat{z}_i \\ -\sum_i \hat{y}_i\hat{x}_i & \sum_i \hat{x}_i^2 + \hat{z}_i^2 & -\sum_i \hat{y}_i\hat{z}_i \\ -\sum_i \hat{z}_i\hat{x}_i & -\sum_i \hat{z}_i\hat{y}_i & \sum_i \hat{x}_i^2 + \hat{y}_i^2 \end{pmatrix}$$

We then solve $\lambda\mathbf{i} = \mathbf{I}\mathbf{i}$ for the eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$ and corresponding eigenvectors $\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3$. The required direction is then $\mathbf{d}_{inertial} = \mathbf{i}_1$.

Having arrived at the definition of $\mathbf{d}_{inertial}$, we can define a line in 3D space with this orientation that passes through the center of mass of the partition. The partitioning plane can be thought of as a plane with $\mathbf{d}_{inertial}$ as its normal vector, which can be translated parallel to itself along this direction.

The final step of inertial partitioning is to fix the location of this, already oriented plane. We can follow a procedure similar to the one in the lecture notes, whereby every vertex in the graph is minimum-distance projected onto the line we have just defined, and the distance between the center of mass and this projection is recorded. The list of all these distances can be sorted to find the median distance, and all vertices whose associated distance is less than this median will be associated with one partition, while the remaining nodes will be associated with the second partition.