Graph Laplacian Data Fusion applied to Optical/Lidar dataset

Geoffrey Iyer

November 19, 2016

Preliminary Results

I got the Nystrom Extension code to work with our example, and used it to calculate eigenvectors of the Graph Laplacian for a fusion of the Optical/Lidar data from the 2015 Data Fusion Contest.

Let $n =$ (number of pixels), and label them $\{x_1, \dots, x_n\}$. Recall, for two pixels x_i, x_j, we define the weight

$$w_{ij} = \exp\left(-\max\left(\|x_i - x_j\|_{\text{Optical}}, \|x_i - x_j\|_{\text{Lidar}}\right)\right).$$

This gives us a weight matrix W, from which we construct the normalized Graph-Laplacian

$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

Here D is the degree matrix, a diagonal matrix with

$$d_i i = \sum_{j=1}^{n} w_{ij}.$$

The eigenvectors of the graph laplacian correspond to solutions of the relaxed graph-min-cut problem.

For computational efficiency, we use the Nystrom Extension to avoid calculating the entire W. Instead we choose $m << n$, and calculate only m columns of the full matrix W. This is enough to give us a reasonable approximation of the first m eigenvectors of L (where by 'first' I mean the eigenvectors corresponding to the smallest eigenvalues).

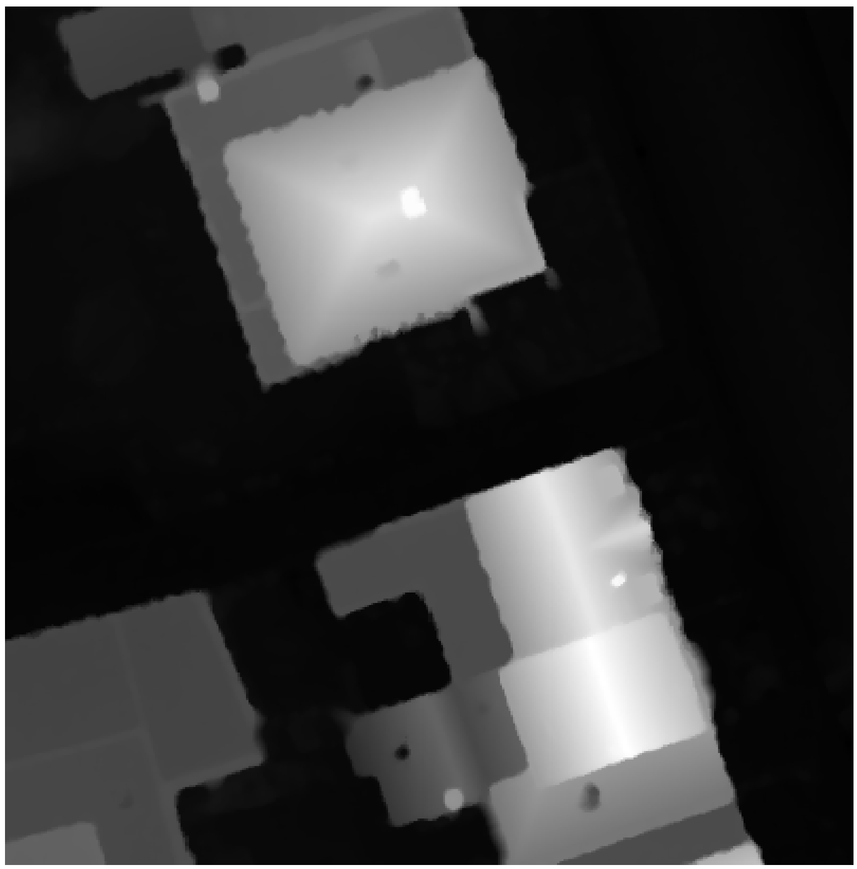
See the example pictures below. I've reprinted the original data, followed by the first 6 eigenvectors. The most important thing to notice here is that the eigenvectors are truly fusing the data. Compare the picture of the eigenvector#1 to the lidar data, then to the optical data. You will see elements of both.

TODO list:

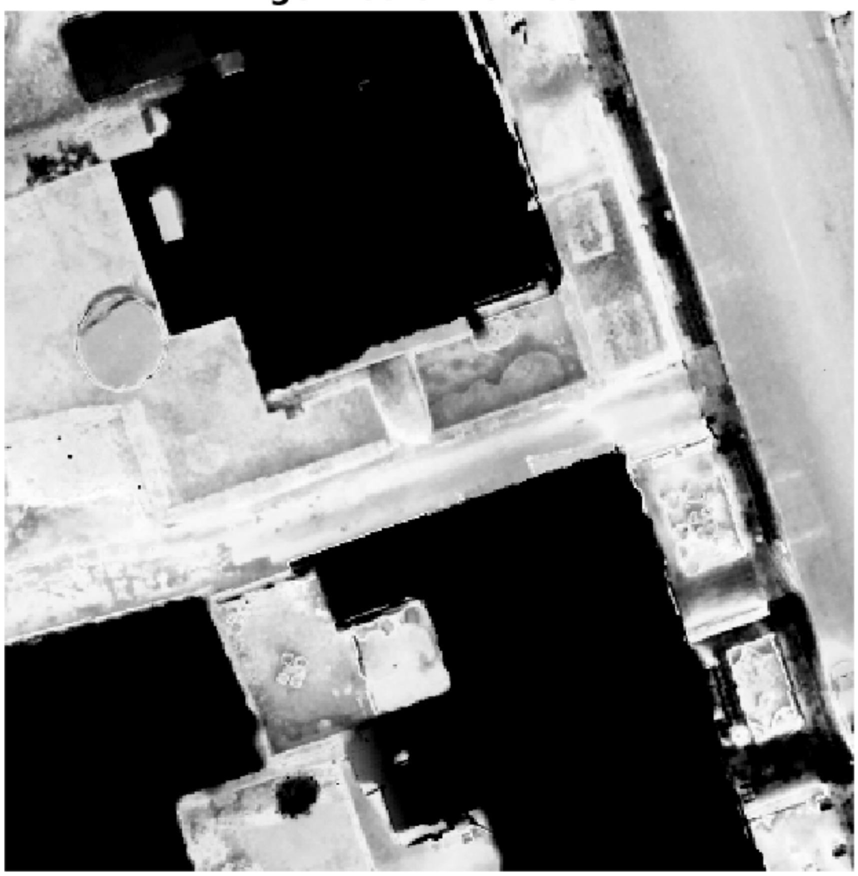
- Lots of tuning of parameters
- Use these eigenvectors to perform some sort of classification
- Code can still be further optimized (currently it runs for these images in about 10 seconds, but there are operations that I know can be improved).

Original Optical Image

Original Lidar Image



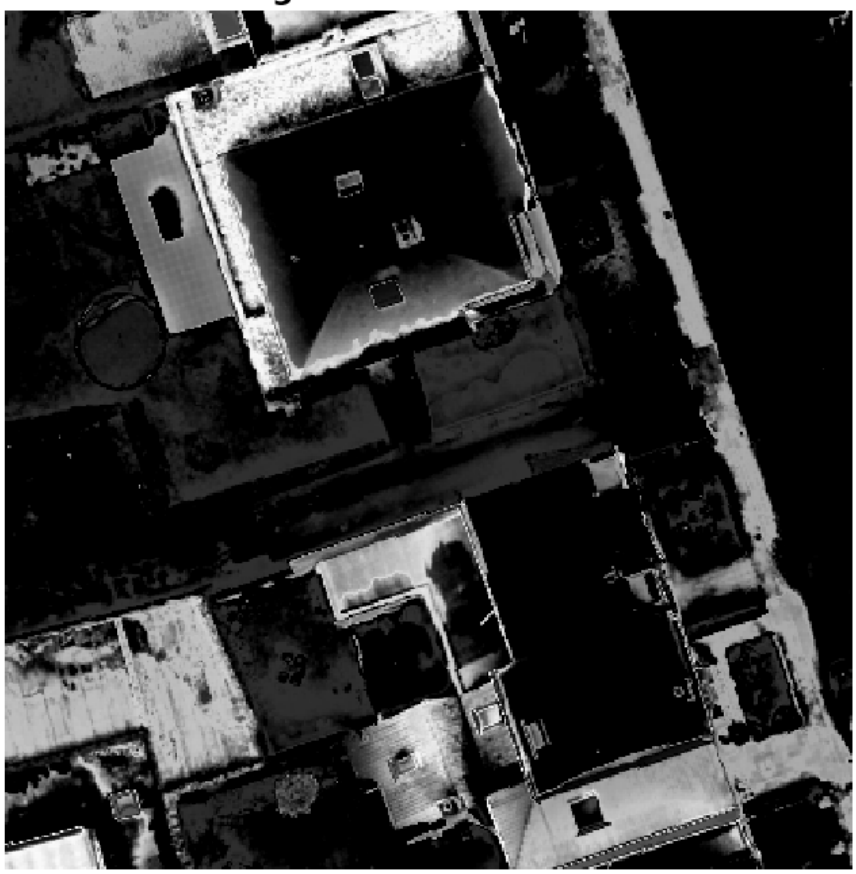
Eigenvector number1



Eigenvector number2

Eigenvector number3

Eigenvector number4



Eigenvector number5

Eigenvector number6