Semantic Alignment of LiDAR Data at City Scale

Supplementary Material

Fisher Yu Jianxiong Xiao Thomas Funkhouser

Princeton University

This supplementary material has two sections. The first provides small implementation details and parameter settings for the feature detection algorithms described in the paper, along with visualizations for several examples. The second shows quantitative and qualitative alignment results that would not fit in the paper, including visualizations of example success and failure cases. Please view the submitted video to see larger-scale visualizations of results.

1 Implementation Details

This section provides small implementation details and parameter settings for each of the feature detection algorithms.

Facade and Road Feature Detection: We detect building facade and road features by extracting planes with a hierarchical clustering algorithm. That algorithm begins with every point in a separate cluster, forms links between points that are within 4 meters and have normals within 45 degrees of each other, and then merges across links iteratively as long as the centroids of the merging clusters are within 100 meters, the normals to their best-fitting planes are within 45 degrees, and the distances from the centroids of one cluster to the best-fitting plane of the other is not more than 1 meter. This method clusters points that are on large planes whose off-plane variation is less than 1 meter. Example results are shown in Figure1a.

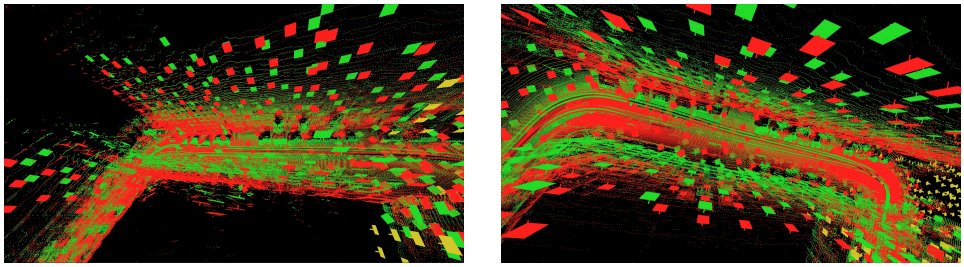
Pole Feature Detection We detect features for signs, lights and trees on the roadside using a Hough transform that accumulates evidence for tall, thin, cylinders at the same XY coordinate. Specifically, for every point within 10 meters from the scanner, we check the point depth on its left and right sides with distance of expected pole radius (0.25 meters). If both sides are more than 0.5 meters further than the current point, the current point is votes for a pole at its XY position. Fea-

tures are then extracted at local maxima positions where at least 50% of the points at that XY position have voted for a pole. Example results are shown in Figure1b.

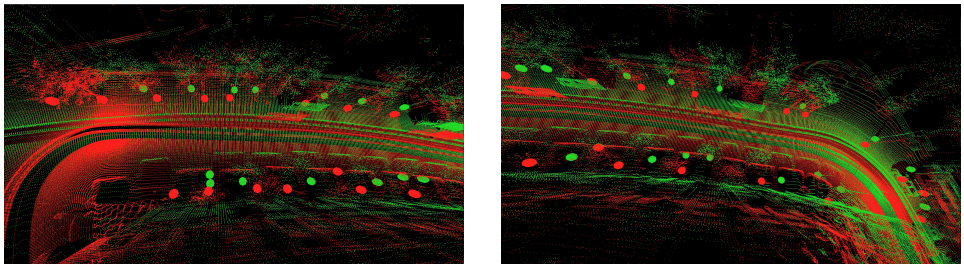
Object Detection: We detect features for cars by training an exemplar SVM on 230 examples. Each example is represented by HOG features in DH images. Two SVMs are trained: one for the original orientation and the other flipped horizontally. In training the SVM, negative examples are sampled randomly and two rounds of hard negative mining are used to improve the result. Then, the detectors are used to detecting vehicles with sliding windows. For each detected bounding box, mean shift clustering is used to segment the object points. Example results are shown in Figure1c.

Segment Feature Detection: We detect features for clusters of points with high objectness using a hierarchical clustering algorithm like the one described in reference [16] of the paper. Specifically, we first remove all points on building facades and roads. Then, the remaining points are separated into segments with a hierarchical clustering algorithm. That algorithm begins with every point in a separate cluster, forms links between points that are within 0.5 meters of each other, and then merges across links iteratively as long as the centroids of the merging clusters are within 8 meters, as long as the distance between clusters is not more than 10 times larger than the maximum pairwise distance within either cluster. This process creates clusters of colocated points separated distinctly. Example results are shown in Figure1d.

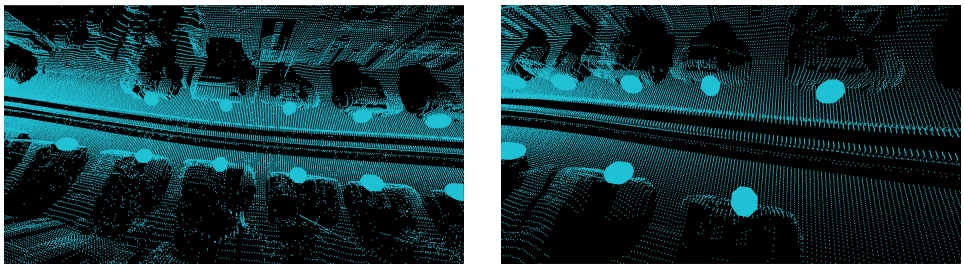
Line Feature Detection: We detect features for lines (straight silhouette edges) using a contour detector that first selects points that are closer to the scanner viewpoint than at least one of their neighbors (these are on silhouette boundaries). We merge clusters to form straight contour lines using the same hierarchical clustering algorithm used for roads and facades, but with links between points that are within 4 meters and have normals within 22.5 degrees of each other, and then merges across links iteratively as long as the the directions of their best-fitting lines are within 22.5 degrees, and the distances from the centroids of one cluster to the best-fitting line of the other is not more than 0.25 meter. Example results are shown in Figure1e.



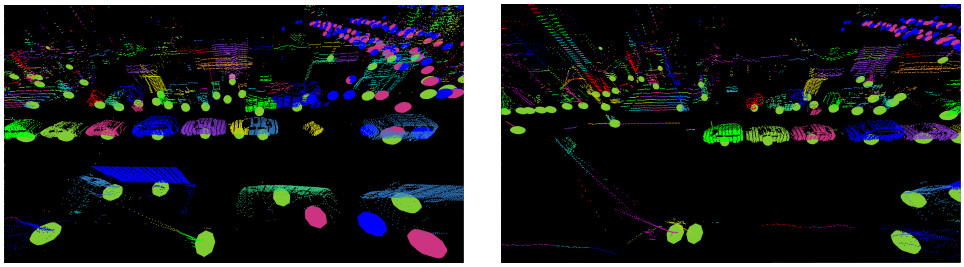
a) Road and Facade Features



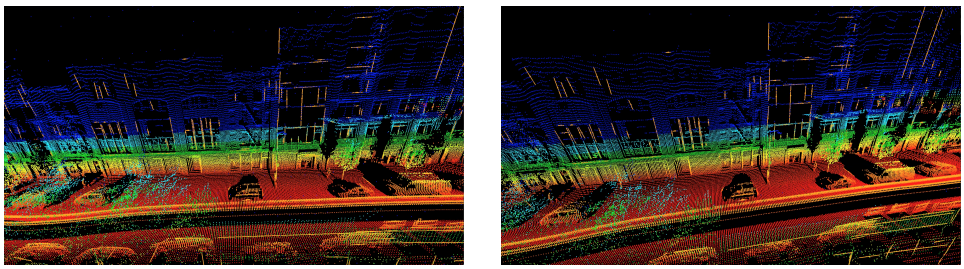
b) Pole Features



c) Object Features



d) Segment Features



e) Line Features

Figure 1: Visualizations of features extracted by our algorithms.

2 Results

This section provides images of our results that would not fit into the paper.

2.1 Quantitative Results

Figure 2 shows the same quantitative evaluation results that appear in Figures 7 and 8 of the paper. However, they include plots for all four cities in one figure for ease of comparison. Please refer to Section 6 of the paper for details.

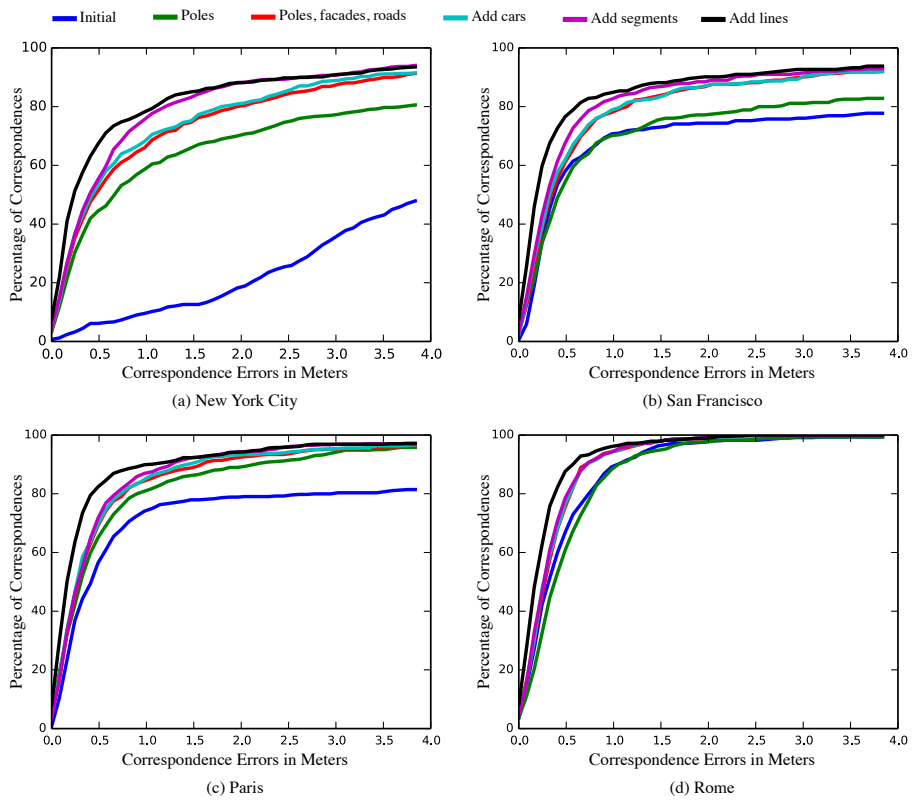


Figure 2: Alignment results using semantic features for Google Street View scans.

2.2 Qualitative Results

Figures 3-5 show images of alignments for areas sampled from New York City. The submitted video shows these results and much more via a real-time fly-over.

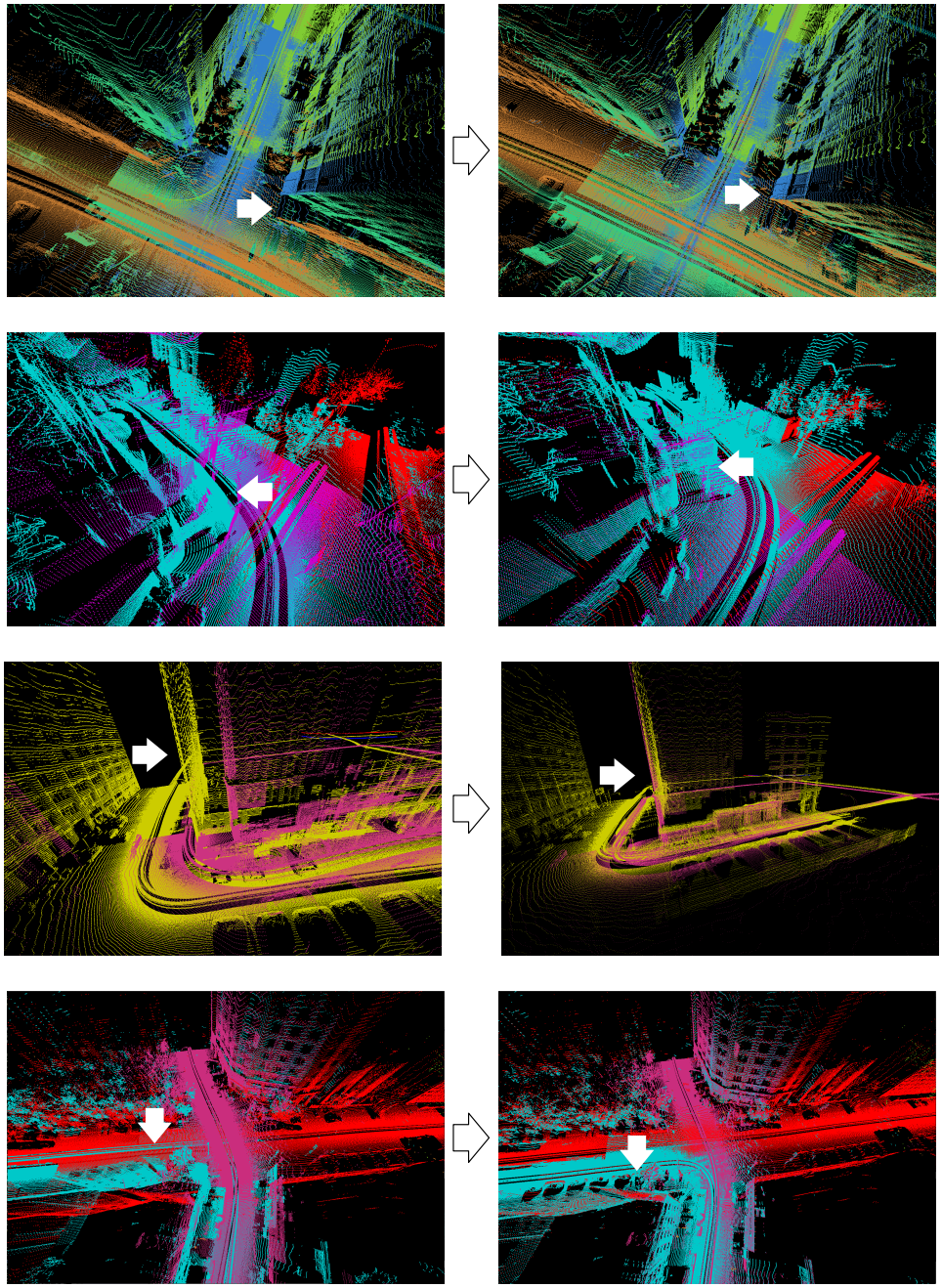


Figure 3: This figure shows examples of how much our algorithm improves the alignment of the LiDAR scans for several examples. The images on the left depict the initial alignment provided by Google (with different scan runs shown in different colors), and the images on the right show the alignment results of our algorithm. Note that significant movements and non-rigid warps were required in all these cases (white arrows highlight visible features aligned by the algorithm).

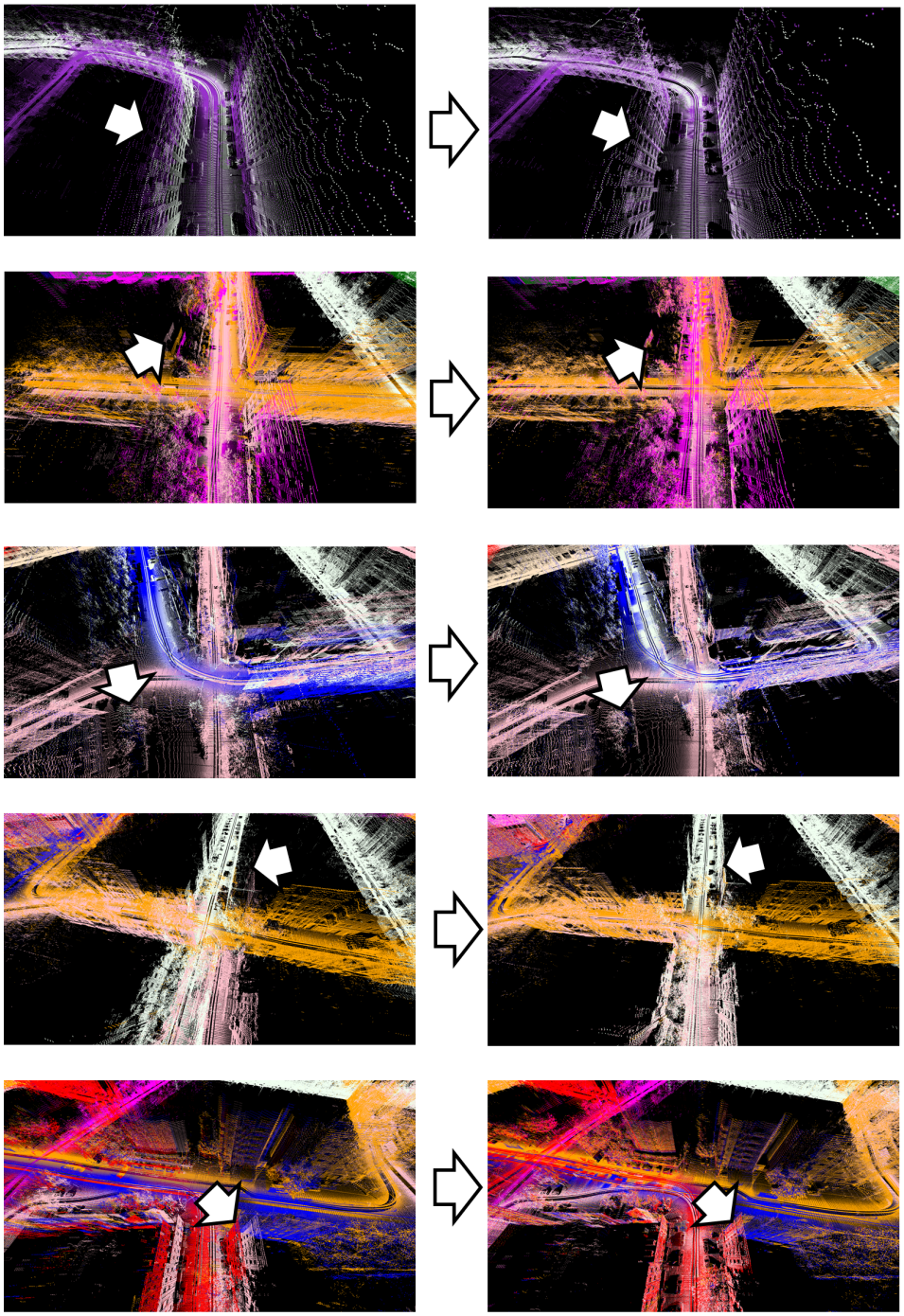


Figure 4: More alignment examples.

Figure 5: This image shows a failure case for our alignment method. The white arrows are pointing to the same building facade. The initial misalignment is tens of meters, which is beyond the search range of our algorithm.