The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Segmentation of Dense Range Information in Complex Urban Scenes

Jonathan R. Schoenberg, Aaron Nathan, and Mark Campbell
Sibley School of Mechanical and Aerospace Engineering
Cornell University, Ithaca, NY 14853
{jrs55, amn32, mc288}@cornell.edu

Abstract—**In this paper, an algorithm to segment 3D points in dense range maps generated from the fusion of a single optical camera and a multiple emitter/detector laser range finder is presented. The camera image and laser range data are fused using a Markov Random Field to estimate a 3D point corresponding to each image pixel. The textured 3D dense point cloud is segmented based on evidence of a boundary between regions of the textured point cloud. Clusters are discriminated based on Euclidean distance, pixel intensity and estimated surface normal using a fast, deterministic and near linear time segmentation algorithm. The algorithm is demonstrated on data collected with the Cornell University DARPA Urban Challenge vehicle. Performance of the proposed dense segmentation routine is evaluated in a complex urban environment and compared to segmentation of the sparse point cloud. Results demonstrate the effectiveness of the dense segmentation algorithm to avoid over-segmentation better than incorporating color and surface normal data in the sparse point cloud.**

I. INTRODUCTION

Large scale autonomous vehicles operating in human-populated areas are required to accurately perceive their environment to make safe and robust decisions. One challenge in perception is the accurate identification of objects in the environment. Identification of objects requires accuracy and robustness to maintain target track, avoid collisions, minimize phantom objects, and predict future intent of objects [1]. The autonomous vehicle must rely on its onboard sensors to perceive the environment and data processing techniques to identify objects of interest.

The available sensors on many autonomous vehicles include laser range finders and high precision optical cameras. Most autonomous vehicles segment sensor data because of the large amount of data, but have done so independently with each sensor [2]. Miller et al. [3] describe the existing algorithm for performing laser range finder point clustering on the Cornell University DARPA Urban Challenge Vehicle. In particular, the algorithm creates ground models from lidar data to identify drivable paths and reason about objects in the environment [3]. This algorithm fails to utilize the richness of information available from optical sensors. For example, a common mistake occurs when the vehicle determines the road is blocked, when in fact the car is moving along a road with a large change in pitch. One solution to these problems is data fusion, which has been shown to increase robustness for urban based perception [3]. Here, it is proposed to fuse lidar and vision to create a dense range map with more information than one sensor alone. This dense range map is segmented and can be used for ground detection or as a pre-processing step for object tracking. This paper is concerned with segmentation of point clouds that have not undergone any temporal filtering at a higher level to ensure uncorrelated measurements that are passed to higher level functions such as stable cluster tracking [2].

Diebel and Thrun [4] model the correlations between the camera image and laser range data using a Markov Random Field (MRF). Gould et al. [5] relax the fronto-parallel planes assumption in Diebel and also use a Huber cost instead of the l_2 cost; better results are claimed because of the first-order model. Harrison and Newman [6] add an extrapolation capability to Diebel's MRF approach and were the first to find the MRF maximum a posteriori (MAP) estimate with a direct method (using Matlab®) instead of an iterative method. Andreasson et al. [7] identify five methods for interpolation of 3D laser scans on pixel density that are outside the MRF framework and introduce a confidence measure. However, mixed performance results of the different methods are seen when comparing to Diebel's MRF. The algorithm presented here uses Diebel's original MRF formulation to produce a textured 3D dense point cloud, but uses an efficient direct solver to move towards real-time performance. In this work, real-time performance is the goal because the entire frame of laser data overlapping the image is captured in near simultaneity with a multiple emitter/detector laser range finder, enabling dense ranging in dynamic scenes.

After the lidar and camera data is fused, the next step is segmentation of the textured 3D dense point cloud. A segmentation algorithm that is deterministic, runs efficiently, and incorporates information about the 3D geometry and pixel intensity is desired. Besl and Jain [8] introduce segmentation of point clouds using variable order functions to model the surface and an iterative refinement of groups, but the algorithm presented here use a non-iterative and non-parametric segmentation technique. Gachter et al. [9] present an algorithm for extracting planes by first extracting lines segments in an indoor environment. However, the algorithm presented here does not make assumptions about the environment. Rusu et al. [10] demonstrate segmentation of lidar returned in an indoor environment based on surface normal and curvature estimates,

J. Schoenberg is a Graduate Research Fellow at the Sibley School of Mechanical and Aerospace Engineering, Cornell University.

A. Nathan is a staff member at the Sibley School of Mechanical and Aerospace Engineering, Cornell University.

M. Campbell is an Associate Professor at the Sibley School of Mechanical and Aerospace Engineering, Cornell University.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

but the algorithm also relies on the structure of the indoor environment and sequentially registered 3D point clouds. Klasing et al. [11] present a deterministic radially bounded segmentation scheme. For performance comparison, the algorithm in [11] is evaluated on the data collected in this study and the algorithm is modified to incorporate information about pixel intensity and estimated surface normal. Lim and Suter [12] use Conditional Random Fields (CRF) to segment point clouds of terrestrial features, but the computations require accurate registration of sequential scans and several hours of run-time. Felzenszwalb and Huttenlocher [13] introduce a fast greedy predicate based segmentation algorithm for computer vision. The algorithm is extended to arbitrary points clouds, instead of images. The algorithm in [13] is used here because of its fast, nearly linear run-time, deterministic segmentation, global guarantees on segment size, and ease of incorporating pixel intensity and 3D geometry into the segmenter predicate.

The novelty of this paper is the ability to efficiently and accurately segment dense range maps produced by the fusion of camera and range data in a complex urban environment in near real-time. The algorithm is demonstrated on experimental data recorded from the Cornell University DARPA Urban Challenge vehicle. Section II describes the fusion of the camera and laser range finder data, segmentation of the textured dense point cloud and an efficient implementation. Section III shows the algorithm applied to experimental data collected in an urban environment from a moving platform. Finally, Section IV summarizes with conclusions demonstrating the success of the algorithm to segment dense point clouds in a complex urban environment.

II. Segmentation of Dense Range Maps

The 3D textured point cloud segmentation algorithm begins by fusing camera images with a multiple emitter/detector laser range finder. Multiple emitter/detectors enable the near simultaneous capture of an entire sparse 3D point cloud that overlays the camera image (Figure 1). The camera and laser range finder data are fused to produce a textured dense point cloud. A graph $G(V,E)$ is formed from the dense point cloud where the vertices (V) are the 3D points and edges (E) connect vertices. An edge cost function is computed for all edges in the graph. The segmentation algorithm proceeds linearly over the edges and joins vertices into segments if the edge cost falls below an adaptive threshold. The final segments are available for object tracking or other high-level processing.

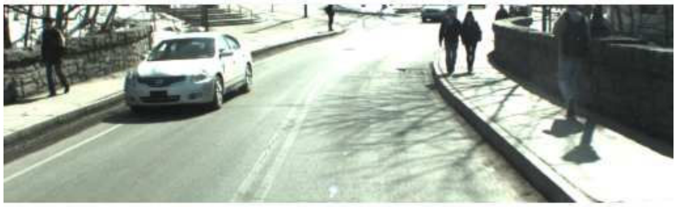

Fig. 1. Color camera image shows four pedestrians and another vehicle on a bridge near Cornell University's campus.

A. Fusion of Camera and Laser Range Finder Data

The fusion of camera and laser range finder data is accomplished by using an MRF to model the correlations between changes in the pixel intensity and depth changes in the corresponding laser range finder [4]. The MRF defined to interpolate the sparse laser range finder data at the image pixel resolution is:

$$\begin{aligned} p(y|x,z) &= \frac{1}{Z}\exp\left\{-\frac{1}{2}(\Phi+\Psi)\right\} \qquad (1)\\ \Psi &= \sum_{i\in\mathscr{L}} k[y_i - z_i]^2 \\ \Phi &= \sum_i \sum_{j\in\mathscr{N}_i} w_{ij}[y_i - y_j]^2 \\ w_{ij} &= \exp(-c\cdot u_{ij}) \\ u_{ij} &= \left\|x_i - x_j\right\|^2 \end{aligned}$$

where y is the desired interpolated 3D point cloud, x is the image intensity data, and z is the sparse laser range finder returns. Additionally, k and c are tuning constants, the set $\mathscr{L}$ corresponds to the pixels that have aligned laser range data, and $\mathscr{N}_i$ corresponds to the cardinal neighbors of pixel i in the image.

The first step in interpolating the laser range finder data (range r, yaw θ, and pitch ϕ) is to align the sparse laser range finder data to the image:

$$\begin{aligned} \vec{v}_i &= C(x,y,f)\cdot T(\vec{p})\cdot f(r,\theta,\phi) \qquad (2)\\ u &= \lfloor \vec{v}_{i_x}/\vec{v}_{i_z} \rfloor \\ v &= \lfloor \vec{v}_{i_y}/\vec{v}_{i_z} \rfloor \end{aligned}$$

where $f(\cdot)$ is the transform from spherical to Cartesian space, $T(\vec{p})$ is the transformation matrix from the laser range finder to camera coordinates, parameterized by a translation and rotation in $\vec{p}$, $C(x,y,f)$ is the 3D to 2D camera projection matrix and u and v are the corresponding width and height image pixels, as shown in Figure 2. The calibration of the camera to laser range finder transformation was performed by placing colored IR reflective markers at various points in a stationary image and recording laser range finder points and determining the transformation that aligned the high-intensity IR returns with the color markers in the image.

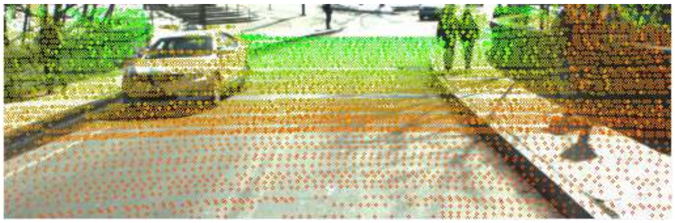

Fig. 2. Aligned sparse laser range finder data (falsely colored points) on an image of an urban scene.

The desired 3D point cloud y in (1) is found by determining the maximum a posteriori estimate of $p(y|x,z)$, which is equivalent to minimizing the negative exponential cost function

$J(y) = 0.5(\Phi + \Psi)$. The minimization of $J(y)$ is cast as an unconstrained quadratic program $f(y) = \frac{1}{2}y^T Ay - b^T y$, where $A = (A_\phi + A_\psi) \in \Re^{n_{\text{pix}} \times n_{\text{pix}}}$ is a large sparse symmetric positive definite matrix (n_{pix} is the number of pixels in the image) with $b = A_\psi^T z$. The solution y is found by solving the linear system $Ay = b$. The large sparse linear system is solved via a direct method using the Intel®Math Kernel Library DSS sparse matrix solver [14] that takes advantage of the banded matrix structure to produce dense 3D point clouds efficiently. The final dense textured 3D point cloud is shown in Figure 3 and is passed on to the segmenter.

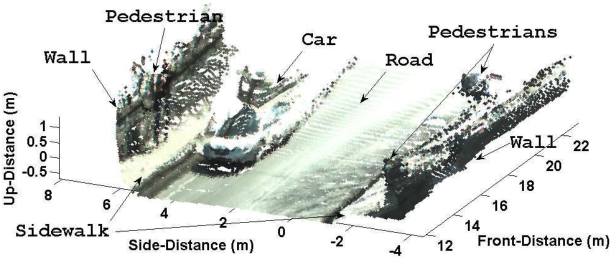

Fig. 3. Interpolated dense 3D point cloud textured according to the original image.

B. Segmentation of Textured Dense Point Cloud

A predicate based segmentation routine [13] is utilized to segment the 3D points in the textured dense point cloud. Unlike the algorithm in [13], the segmentation routine uses 3D geometry, pixel intensity and estimated surface normal at each point in the dense cloud to discern segments. The segmentation algorithm operates on an undirected graph $G(V,E)$, where the vertices are the 3D points and the edges correspond to connections between vertices. The algorithm is initialized by computing the edge cost ω_{ij} between neighbors i and j in the graph, which is defined as a weighted combination of Euclidean distance, pixel intensity difference and the estimated surface normal divergence:

$$\omega_{ij} = k_e \cdot \left\|\vec{v}_i - \vec{v}_j\right\|^2 + k_i \cdot \left\|x_i - x_j\right\|^2 + k_N \cdot (1.0 - \vec{N}_i^T \vec{N}_j) \quad (3)$$

where $\vec{N}$ is the estimated surface normal at the 3D point, and k_e, k_i and k_N are the relative weights of the Euclidean distance between points, the pixel intensity difference and surface normal distance. The relative weights are hand-tuned for this implementation; the results presented below were insensitive to small changes in weights, but could be determined via cross-validation. Note that the lower edge weight implies that it is easier to merge two clusters.

The algorithm is initialized with each vertex (3D point) as a disjoint set and proceeds linearly along edges merging vertices into a cluster if the edge weight is below the cluster threshold η_i. The key to the algorithm is the update of the cluster threshold as vertices are joined. This has the effect that smaller clusters require greater evidence of a boundary to avoid being merged. The benefits of this segmentation routine are that it runs in near linear time to the number of pixels, is deterministic, and guarantees clusters of a certain size based on the tuning parameter σ_k listed in Algorithm 1.

Algorithm 1 Minimal Spanning Tree Segmentation [13]

```
Compute edge weights according to (3)
Initialize cluster thresholds: $\eta_i = 1/\sigma_k, \forall i \in V$
for all Edges in Graph $G(V,E)$ do
    Extract set $A$ and $B$ along the edge
    if ROOT{A} != ROOT{B} then
        if $\omega_{ij} < \eta_i$ then
            $A \leftarrow A \bigcup B$
            $\eta_i = \eta_i + \text{SIZE}(A)/\sigma_k$
        end if
    end if
end for
```

The primary difference between the approach in [11] and Algorithm 1 is the traversal over the edges, instead of vertices, and the adaptive threshold as a function of cluster size. There are many possible choices of edges (E) between the vertices (V) on the graph $G(V,E)$ when the vertices are 3D points in space. One approach uses $k-$nearest neighbor, where edges are created between the closest k vertices using any desired cost function. Another approach is to choose vertices that lie within a radial bound of each other, similar to [11]. The algorithm presented here begins by creating a dense range map, where a 3D point is estimated for each pixel in the underlying image. As a result, it is possible to create edges between 3D points corresponding to neighboring pixels in the image. It is important to note that two vertices not connected with an edge may end up in the same cluster through intermediate vertices and two vertices connected with an edge will not necessarily be in the same cluster.

C. Surface Normal Estimation

The surface normal is computed using Principle Component Analysis (PCA)[15]. Similar to [15], the analysis presented here found the PCA method performed better than the area averaging method [15] to estimate the surface normal. The surrounding points used to determine the surface normal can be determined from the N-nearest neighbors, where nearest is a Euclidean norm. The estimated surface normal is ambiguous in terms of sign; to account for this ambiguity the dot-product between estimated surface normals is repeated using the negative estimated surface normal of one of the vectors and the minimum result of the term multiplied by k_N in (3) is selected. For the sparse data set, if all nearest neighbors lie on a single laser scan line, the surface normal estimate is invalid. Therefore, these points have no valid surface normal estimate and k_N in (3) is set to zero. For the dense data set, the neighboring points used to estimate the surface normal are not from the same laser scan line, therefore all points have a valid surface normal.

III. Experimental Results

A. Autonomous Vehicle Testbed

Experimental data was collected for evaluation of the textured dense 3D point cloud segmentation routine using Cornell University's DARPA Urban Challenge vehicle [3], shown in Figure 4. The optical camera mounted along the center roofline is a color Basler A622f with a 4.8mm lens and a 30-deg horizontal and 30-deg vertical field of view. The Basler camera produces 1280 x 1024 pixel images at 16Hz. The multiple emitter/detector laser range finder located on top of the vehicle is the Velodyne HDL-64E. The Velodyne scanner receives 64 vertical lines over 26.8 degrees of laser range finder data over 360 degrees at 15 Hz with a maximum detection range of 50 m. The scan rate of the laser range finder means all the laser range data overlapping the image is collected in 5.56 msec, at an interval of 61 msec. The portion of the image with overlapping laser range data is 1280 x 330 pixels, corresponding to an estimation of 337,920 3D points in the dense range map, which is more than 60 times greater than the number of 3D points in the original laser range scan overlapping the image.

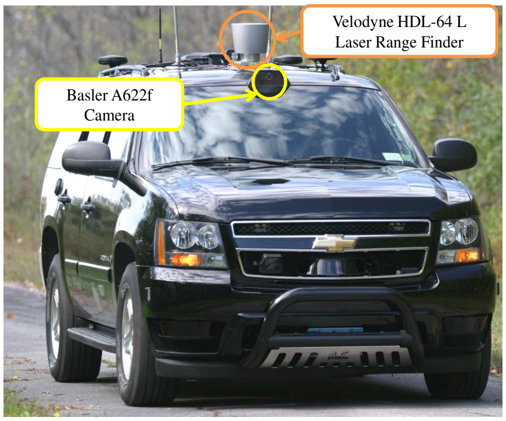

Fig. 4. Cornell University's autonomous Chevrolet Tahoe, equipped with a high precision optical camera and a multiple emitter/detector laser range finder.

B. Urban Environment Data Collection

The algorithm was run on data collected near Cornell University in Ithaca, NY with the vehicle under human control on a busy afternoon. The scenes observed include moving cars, pedestrians, parked cars and standard urban features such as buildings, newspaper stands, and street signs. The initial sequence of analyzed scenes correspond to the vehicle passing by pedestrians on the side of the road and another vehicle while on a bridge (Figure 1). As a performance metric, the object-level consistency error (OCE) introduced by Polak et al. [16] is used to compare the segmented images with hand-labeled truth. The OCE metric weights segmentation performance as a function of the segment size. The OCE heavily penalizes over-segmentation, so the global consistency error (GCE) and local consistency error (LCE) introduced by Martin et al. [17] are also shown. These metrics favor segmentation that is intuitively reasonable. There are 13 frames analyzed for segmentation performance, 9 proceeding and 3 following the frame in Figure 1. The textured dense 3D point cloud segmentation routine described in Section II is run and performance summarized. To demonstrate the importance of densifying the sparse 3D point cloud, the radially-bounded nearest neighbor (RBNN) algorithm from [11] and a modification of the RBNN algorithm to incorporate pixel intensity and estimated surface normals are used to segment the sparse data and performance presented.

C. Segmentation with Sparse Range Data Only

The sparse laser range finder data from the Velodyne HDL-64E sensor is shown in Figure 5 with false coloring and hand-labeled segments to indicate the ideal segmentation of the scene. The scene contains four pedestrians, two walking next to each other and two near the bridge walls and another car on the road surface. The difficulty in segmenting this scene using the sparse range data alone is illustrated in Figures 6 and 7. The sparse range data is segmented using the RBNN approach from [11]. The RBNN performs well with a lower threshold (Figure 6) in identifying the pedestrians, but has difficulty creating a coherent cluster for the road surface because successive range scans are intersecting the road surface at increasing distance. To correct this problem, a potential solution is to increase the radial threshold, but the smallest threshold that clusters the road leads to nearly the entire range scan being segmented into a single cluster (Figure 7). The other problem is that the vehicle is in contact with the road and this leads to difficulty in separating the vehicle from the road surface.

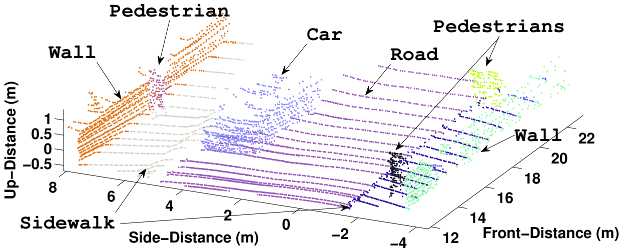

Fig. 5. Labeled truth data of the sparse laser range finder data.

D. Segmentation with Sparse Range Data Augmented with Color and Surface Normal Estimates

To attempt to solve the problem of a large threshold causing nearly a single cluster and a low threshold on the range data causing over segmentation, a modification to the RBNN algorithm is presented. The color of each sparse point and the estimated surface normal at each point are used to construct the cost metric in (3) for use as the radial bound in the RBNN algorithm. The surface normals are estimated at each point using the PCA technique [15] by joining the 8 nearest neighbors. Figure 8 shows the result from including pixel

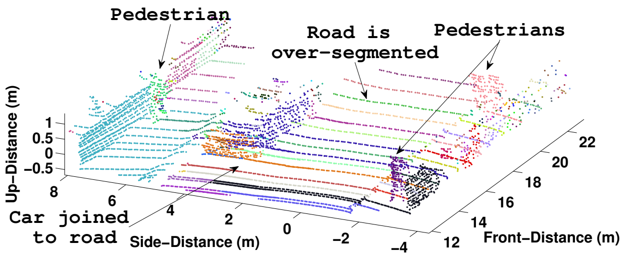

Fig. 6. RBNN algorithm from [11] shows over segmentation of the road surface due to increasing distance between scans intersecting the road.

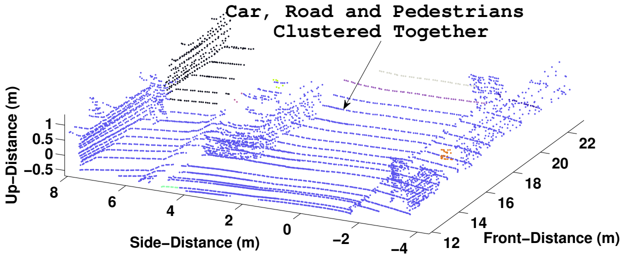

Fig. 7. In order to cluster the road surface, the RBNN algorithm from [11] creates nearly a single large cluster.

intensity and estimated surface normal into the segmentation of the sparse range data. The segmentation result shows the car is maintained as a set of separate clusters from the road, but that the road is not clustered into a coherent segment. Augmenting the sparse range data with the pixel intensity and surface normal estimates are not enough to achieve adequate segmentation performance.

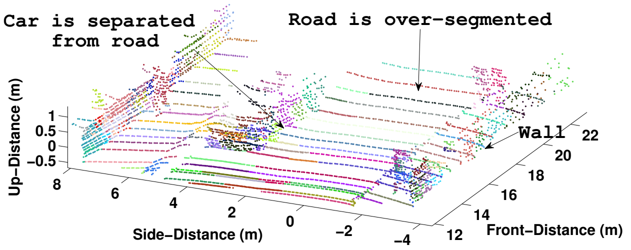

Fig. 8. Including color and estimated surface normal with only the sparse range data still creates over-segmented range scans, motivating the use of the dense range map achieved through interpolation with the MRF.

E. Segmentation of the Textured Dense Range Map

Finally, the full textured dense point cloud performance is shown using the algorithm described in Section II. The hand-labeled segments for the dense point cloud are shown in Figure 9. The final results of the segmentation of the dense range map are shown in Figure 10. The ability to estimate a 3D point at every pixel in the image provides contextual information for the algorithm to make decisions on what to segment. The road surface is a prime candidate to benefit from the algorithm due to the near continuous estimate points along the surface, thereby avoiding the need to handle increasingly larger spaces between successive line scans. The pedestrian along with left side of the image is joined with the wall because the interpolated range samples reduce the distance metric between the pedestrian samples and the wall.

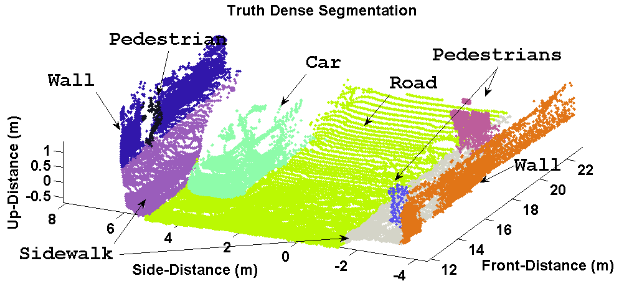

Fig. 9. Labeled truth data of the dense laser range finder data shows continuity between the vehicle and the road, which provides a challenge for accurate segmentation.

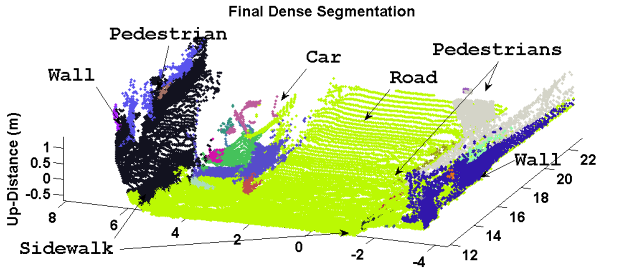

Fig. 10. Final result of the newly proposed algorithm shows segmentation of the dense range data achieves separation of the car and a coherent road segment.

Table I shows the computed performance calculated from hand-labeled truth data over 13 frames around the one presented in detail. The ideal performance for each of the performance metrics is 0 for perfect segmentation and 1 for the worst segmentation. As expected, the RBNN algorithm with the low threshold is penalized in the OCE metric for over-segmentation and has a higher score for the GCE and LCE as well. The RBNN algorithm with the high threshold segments nearly the entire scene into a single cluster. This is also penalized in the OCE metric, but less penalized in the GCE and LCE metrics, leading to a lower score. The modified RBNN algorithm, which includes the color and estimated surface normal, has worse performance than the simpler RBNN algorithm because the scene is still heavily over-segmented. The benefit of segmenting the full dense 3D point cloud developed using Algorithm 1 is clear in the OCE, GCE, and LCE metrics, because the point cloud is not over-segmented nor clustered into a single segment. The largest cluster in all the frames analyzed is the road, therefore, when using the full dense 3D point cloud the accurate segmentation of the road leads to better performance in the OCE metric. The performance benefit when utilizing the dense map instead of the sparse map is apparent, because the sparse map is not adequately segmented even when the color and surface normal information is included.

TABLE I
SEGMENT PERFORMANCE FOR 13 TOTAL FRAMES AS COMPARED WITH VARIOUS METRICS.

Algorithm	OCE	GCE	LCE
RBNN-Low	0.79	0.30	0.27
RBNN-High	0.80	0.14	0.09
Modified RBNN	0.89	0.16	0.14
Full Dense	0.53	0.06	0.07

F. Algorithm Run-Time

The algorithm run-time is dependent on the size of the camera image, because finding the MAP estimate to the MRF (1) requires the solution of n_{pix} simultaneous equations and the segmentation routine is linear in the number of pixels. The algorithm is run on a desktop computer with a 2.7GHz Intel®Core$^{\text{TM}}$i7 processor which enables different portions of the algorithm to be run as different threads each on separate cores. The run-time is summarized in Table II and shows that near real-time performance is achieved at full resolution with an update rate of 0.5 Hz; however, the half resolution image achieves a 2 Hz update rate. The 2 Hz update rate for this rich information is equivalent to the lane-finding algorithm used on the Cornell University DARPA Urban Challenge vehicle [3], while providing much more operational usefulness.

TABLE II
DENSE POINT CLOUD SEGMENTATION ALGORITHM RUN-TIME AT FULL RESOLUTION AND HALF RESOLUTION, ALONG WITH SPARSE DATA SEGMENTATION ROUTINE RUN-TIME (MSEC).

Process	RBNN	Modified RBNN	Full Res. 337,920 pixels	Half Res. 84,480 pixels
Align Range Data	-	7	7	7
Solve MAP Estimate	-	-	385	100
Compute Normals	-	120	1170	320
Segment Point Cloud	116	51	291	86
Total (msec)	116	178	1852	513

IV. CONCLUSION

An algorithm to segment textured dense 3D point clouds is presented and demonstrated in a complex urban environment. Textured dense point clouds are generated from interpolating sparse laser range finder data constrained by an aligned optical image. The dense point cloud is found using efficient matrix solver routines that enable near real-time performance. The segmentation is performed using an efficient and deterministic algorithm that utilizes the pixel intensity and 3D point geometry data. The algorithm is demonstrated in a complex urban environment which shows the successful ability to segment the road surface as a coherent cluster. The benefit of using the dense range data for interpolation is demonstrated by examining the segmentation performance with the sparse range data alone and augmenting the sparse range data with pixel intensity and surface normal estimates prior to segmentation. Finally, fast run-time of the algorithm provides operational usefulness for large-scale real-time autonomous vehicles.

V. ACKNOWLEDGMENTS

The authors would like to thank Peter Moran and the Cornell University DARPA Urban Challenge Team for providing sensors, sensor interfaces, and data collection support. This work is supported under the Northrop Grumman Electronic Systems Scholars Program, ARO Grant #W911NF-09-1-0466 and NSF Cyber-Physical Systems Grant #CNS-0931686.

REFERENCES

[1] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F.-R. Kline, “The mit-cornell collision and why it happened,” *Journal of Field Robotics Special Issue on the DARPA Urban Challenge*, vol. 25, no. 10, pp. 775–807, 2008.

[2] I. Miller, M. Campbell, and D. Huttenlocher, “Efficient unbiased tracking of multiple dynamic obstacles under large viewpoint changes,” *Robotics, IEEE Transactions on*, In Submission.

[3] I. Miller, M. Campbell, D. Huttenlocher, F.-R. Kline, A. Nathan, S. Lupashin, J. Catlin, B. Schimpf, P. Moran, N. Zych, E. Garcia, M. Kurdziel, and H. Fujishima, “Team cornell’s skynet: Robust perception and planning in an urban environment,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, 2008.

[4] J. Diebel and S. Thrun, “An application of markov random fields to range sensing,” *Advances in Neural Information Processing Systems*, vol. 18, p. 291, 2006.

[5] S. Gould, P. Baumstarck, M. Quigley, A. Y. Ng, and D. Koller, “Integrating visual and range data for robotic object detection,” in *European Conference on Computer Vision*, vol. M2SFA2, Marseille, France, 2008.

[6] A. Harrison and P. Newman, “Image and sparse laser fusion for dense scene reconstruction,” in *7th International Conference on Field and Service Robotics*, Cambridge, Massachusetts, 2009.

[7] H. Andreasson, R. Triebel, and A. Lilienthal, “Non-iterative vision-based interpolation of 3d laser scans,” *Computational Intelligence (SCI)*, vol. 76, pp. 83–90, 2007.

[8] P. J. Besl and R. C. Jain, “Segmentation through variable-order surface fitting,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 2, pp. 167–192, 1988, 0162-8828.

[9] S. Gachter, V. Nguyen, and R. Siegwart, “Results on range image segmentation for service robots,” in *Computer Vision Systems, 2006 ICVS ’06. IEEE International Conference on*, 2006, pp. 53–53.

[10] R. B. Rusu, N. Blodow, Z. Marton, A. Soos, and M. Beetz, “Towards 3d object maps for autonomous household robots,” in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 3191–3198.

[11] K. Klasing, D. Wollherr, and M. Buss, “A clustering method for efficient segmentation of 3d laser data,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 4043–4048.

[12] E. H. Lim and D. Suter, “Multi-scale conditional random fields for over-segmented irregular 3d point clouds classification,” in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW ’08. IEEE Computer Society Conference on*, 2008, pp. 1–7.

[13] P. Felzenszwalb and D. Huttenlocher, “Efficient graph-based image segmentation,” *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004, 10.1023/B:VISI.0000022288.19776.77.

[14] Intel®, “Math kernel library 10.2 (http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation/),” Tech. Rep.

[15] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, “Comparison of surface normal estimation methods for range sensing applications,” in *Robotics and Automation, 2009. ICRA ’09. IEEE International Conference on*, 2009, pp. 3206–3211.

[16] M. Polak, H. Zhang, and M. Pi, “An evaluation metric for image segmentation of multiple objects,” *Image and Vision Computing*, vol. 27, no. 8, pp. 1223–1227, 2009, 0262-8856 doi: DOI: 10.1016/j.imavis.2008.09.008.

[17] D. Martin, C. Fowlkes, T. Doron, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, 2001, pp. 416–416.