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

Building Lane-Graphs for Autonomous Parking

Young-Woo Seo, Chris Urmson, David Wettergreen, and Jin-Woo Lee

Abstract— **An autonomous robotic vehicle can drive through and park in a lot more reliably if it is guided by a parking lot map. This specialized map creates structure, including the centerlines of drivable regions and intersection locations in an often unstructured and unmarked environment and enables the vehicle to focus its attention on regions that require detailed analysis. Existing methods of building such maps require manually driving vehicles for collecting sensor measurements. Instead of pursuing a labor-intensive approach, we analyze an aerial image of a parking lot to build a topological map. In particular, our algorithm produces a lane-graph of a parking lot's drivable regions by executing several image processing steps. First it estimates drivable regions by superimposing detection of a parking spots onto the parking lot boundary segmentation. Second, a distance transform is applied to drivable regions to reveal its skeleton. Lastly our algorithm searches the distance map to identify a set of the peak points and connects them to generate a lane-graph that concisely represents drivable regions. Experiments show promising results of real-world parking lot aerial-imagery analysis.**

I. INTRODUCTION

Parking is an important and uniquely complicated requirement for driving an automobile. It involves several activities that are performed in parallel: recognizing the parking lot structure (e.g., the geometric structure of the drivable region); understanding traffic rules; continuously observing moving obstacles (e.g., pedestrians and other vehicles), and steering. Despite this complexity, parking is also a mundane task familiar to many, even when driving new parking lots. Drivers have long dreamt of self-driving cars that would ease their daily parking tasks.

Autonomous parking can be implemented with or without a model of a parking lot. For example, a reactive robotic vehicle, operating by responding to its environment, might manage to perform parking itself without such model of a parking lot. However it would be hard to simultaneously perceive road-markings and moving obstacles, and infer the structure of a parking lot for planning its motions, resulting in inconsistent and inefficient maneuvers.

A model of a parking lot can describe the detailed geometry of a parking lot such as centerlines and intersection locations of drivable regions and informs the vehicle where it can drive. The model would help a robotic vehicle perform reliable and safe autonomous parking maneuvers in that it enables a robotic vehicle to focus its attention on regions that require detailed analysis.

To exploit the benefits of a parking lot's model, one might consider existing cartographic databases for retrieving such models. However, this would be very difficult because any roadmap database primarily built for human driving does not contain such models at all – at best a parking lot in any roadmap database is depicted as a point in a 2-dimensional map space.

Alternatively one can build the model of a parking lot by fitting a geometric model to sensor measurements [6], [8], [10]. This approach requires a labor-intensive step because of the need to manually drive a robot to collect sensor measurements.

In this paper we present ortho-image[1] analysis algorithms that produce a lane-graph of drivable regions by analyzing a single parking lot image. From our previous work [14], [15], drivable regions in a parking lot are determined by a combination of results of parking spot detection and results of parking lot boundary segmentation. A distance transform is used to reveal the skeleton of drivable regions' geometry. Our lane-graph generation algorithm iteratively searches for a lane-graph in the distance transform map that concisely represents drivable regions.

II. RELATED WORK

Robotic vehicles participating in the 2007 DARPA Urban Challenge[2] demonstrate autonomous parking as they entered parking lots, interacted with other vehicles, and parked themselves into designated locations. Although their demonstrations were very complex and well performed, they are not close to those of daily auto-trips: no traffic rules are enforced, interactions between vehicles are sparse, and no pedestrians are around. The robotic vehicle models a parking lot as an open space with several check points [17] and iteratively searches for optimal motion trajectories to reach a check point while ignoring the geometric structure of drivable regions. This approach should be discouraged when autonomous parking is applied to the real world.

The majority of research for realizing autonomous parking is about developing control and planning algorithms for road-side parallel parking [1], [2], [12], where a model of a parking lot might not necessarily be required; assumed to be given; or robotic vehicles are operating reactively, simplifying complex perception issues.

Top-view aerial imagery are very useful for mapping the operational environment because it provide structural overview of environments. Owing to this unique benefit, overhead imagery has been analyzed to provide alternative, but important views of environments that enables robots to

Young-Woo Seo, Chris Urmson, and David Wettergreen with Robotics Instititue, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, {ywseo,curmson,dsw}@ri.cmu.edu

Jin-Woo Lee with Electrical and Controls Integration Lab, R&D General Motors, 30500 Mound Rd, Warren, MI 48090, jin-woo.lee@gm.com

[1] An ortho-image is an aerial image in which terrain relief and camera tilt are removed through a rectification process.

[2] Visit the following for more information, http://www.darpa.mil/grandchallenge/index.asp

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

plan globally to achieve their goals. Aerial imagery may become outdated so in-vehicle sensors would need real-time assurance that regions are drivable. In combination with other onboard sensors such as vision sensors and range finders, aerial images have been used for generating cost maps for long-range traversal [16], global localization [7], building and maintenance of robots' world model [13], [17], and mapping [11].

In the GIS (Geographic Information System) community, there has been an extensive amount of research work on aerial and satellite image analysis for maintaining road map databases [3], [4], [5]. To the best of our knowledge, our work is unique in that we recover the geometry of drivable regions in a parking lot ortho-image by using self-obtained image cues [15].

There are three similar works in the realm of parking lot structure analysis. Wang and Hanson present an algorithm that uses multiple aerial images to extract the structure of a parking lot for simulation and visualization of parking lot activities [18]. Multiple images from different angles are used to build a 2.5 dimensional elevation map of a parking lot. This usage of multiple images makes it difficult to generalize their method because it is not easy to obtain such images on the same geographic location from publicly available imagery. Dolgov and Thrun present algorithms that build a lane-network of a parking lot from sensor measurements [6]. They first build a grid map of static obstacles from range measurements about a parking lot and use an optimization technique to infer a topological graph that most likely fits the grid map. Similarly, Kummerle and his colleagues build a multilevel (or multilevel surface) map of a parking building from range sensor measurements [10]. A multilevel-map is a 2D grid map that each of cells maintains a stack of patches. As individual patches in a cell correspond to different height estimates, this multilevel structure is used to represent drivable regions and vertical objects. To fill in individual cells, they first formulate a mapping as a graph construction problem that a node represents a vehicle pose and an edge represents a relative motion between poses; and then find optimal nodes based on constraints imposed on edges. A new node is continuously added to the graph until a loop closure is found. These two works are very close to ours in that they are building a road network for autonomous parking, but different in that they need to drive the robot to collect sensor measurements, whereas our approach does not require manual data collection. Another potential problem of this approach is that it might fail to collect data along the structure of a parking lot when the parking lot is empty.

III. Topological Map of Drivable Regions in Parking lot

A lane-graph of a parking lot is a topological representation of drivable regions. Our lane-graph generation algorithm requires a map of non-drivable (or an obstacle map) as an input. We built this map by using two aerial image analysis results from our previous work [14], [15].

Several different of ortho-image analysis algorithms are developed to detect all of the visible parking spots in a

Algorithm 1 Lane-graph generation algorithm.

```
Input: - $I$, a parking lot ortho-image,
       - $I_{non-drivable}$, a binary image of non-drivable regions
Output: - $\mathbb{G}$, a lane-graph that maximally covers drivable
        regions in the parking lot
$\mathbb{G} = \{\mathbf{V}, \mathbf{E}\}, \mathbf{V} = \{\phi\}, \mathbf{E} = \{\phi\}$
$I_{dt}$ = generate_distance_map($I_{non-drivable}$)
$\mathbb{M}$ = find_local_maxima($I_{dt}$), $\mathbb{M} = \{m_1, ..., m_{|\mathbb{M}|}\}$
$\mathbb{M}_1$ = prune_local_maxima($\mathbb{M}$), $|\mathbb{M}_1| \ll |\mathbb{M}|$
$\mathbb{M}_1$ = sort($\mathbb{M}_1$), $m_1 > m_2 >, ..., > m_{|\mathbb{M}_1|}$
$\mathbb{M}_2$ = define_rendezvous_point($\mathbb{M}_1$), $|\mathbb{M}_2| \ll |\mathbb{M}_1|$
repeat
    $mc_i$ = find_maximal_circle($m_i$), $m_i \in \mathbb{M}_1$
    Remove all of the local maxima within the circle, $mc_i$
    Create a vertex, $v \leftarrow m_i$,
    $\mathbf{V} = \{\mathbf{V} \cup v\}$
until all of the rendezvous points, $\mathbb{M}_2$, are visited
for all $v_i \in \mathbf{V}$ do
    Identify neighboring vertices, $N(i)$, of $v_i$
    Create an edge, $e_{ij}$, if the $j$th neighboring vertex,$v_j \in$
        $N(i), (i \neq j)$, is visible from the $i$the vertex, $v_i$.
    $\mathbf{E} = \{\mathbf{E} \cup e\}$
end for
Return $\mathbb{G} = \{\mathbf{V}, \mathbf{E}\}$
```

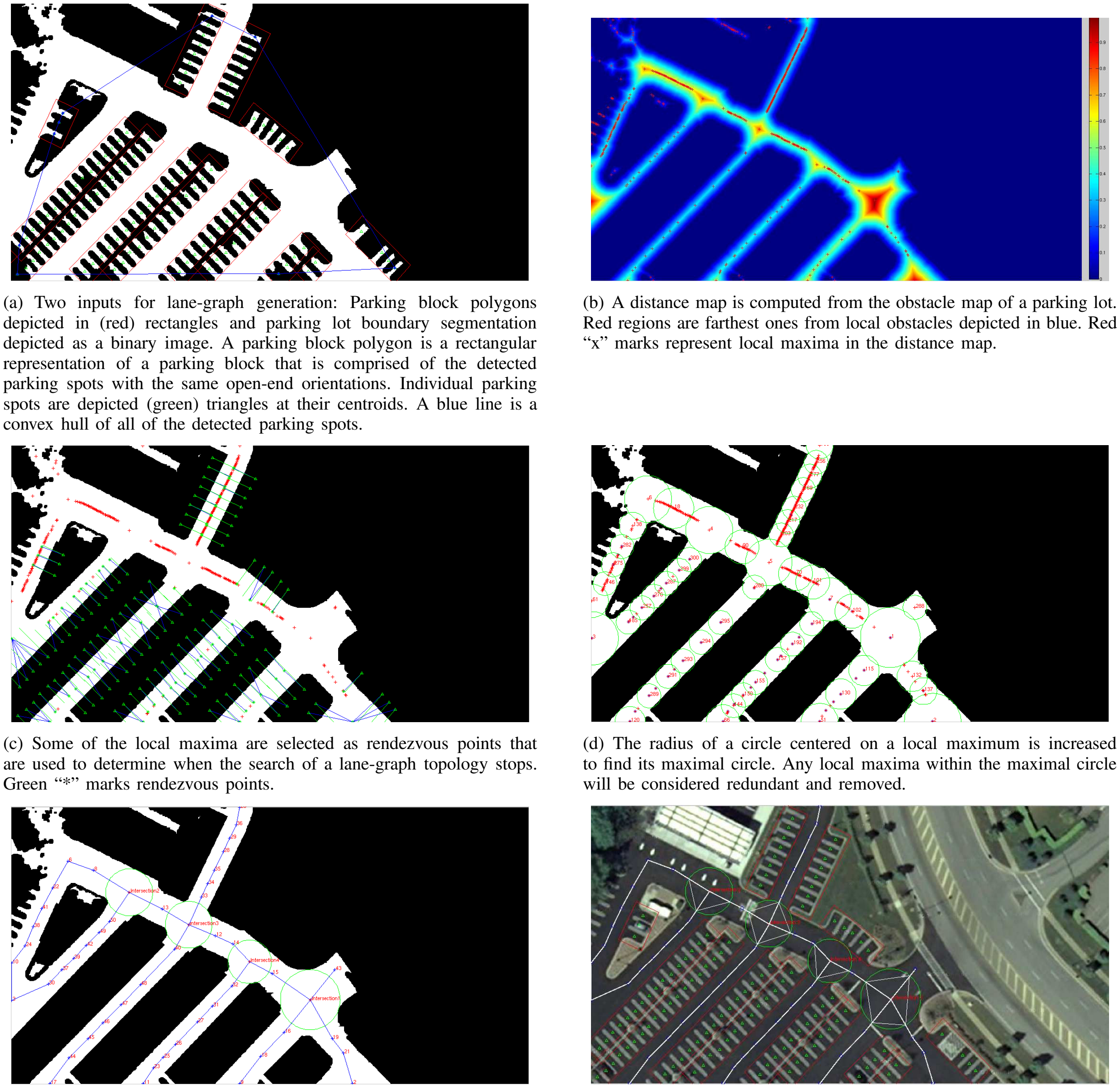
parking lot ortho-image. Our self-labeling method analyzes the spatial layout of extracted lines and automatically obtains some of the easy-to-detect true parking spots. These self-labeled parking spot image patches are used for several purposes. First, the geometric properties of self-labeled examples, such as average length, width and distance between them, are used to generate hypotheses that are predictions of the true parking spot locations. Second, these self-labeled parking spot images are used to train a binary classifier to filter out incorrect hypotheses. Lastly, the image characteristics of self-labeled examples are used to learn a road-marking classifier and a parking lot boundary segmentor [15]. The image regions of the estimated parking lot boundary are overlapped with the detected parking spots to produce the map of non-drivable regions in a parking lot ortho-image.

A. Connecting Maximal Circles for Discovering Topology of Lane-Graph

Algorithm 1 describes the procedure of our lane-graph generation in detail.

The algorithm requires an obstacle map of a parking lot image. This map of non-drivable regions, $I_{non-drivable}$, is obtained by combining parking spot detection results and parking lot boundary segmentation results. Figure 1(a) shows examples of parking spot detection results and parking lot boundary segmentation results.

A distance transform is often applied to a robot's operational environment for identifying obstacle-free regions. In our case, the function, *generate_distance_map*($I_{non-drivable}$), implements the brush-fire algorithm to propagate distance values from



(a) Two inputs for lane-graph generation: Parking block polygons depicted in (red) rectangles and parking lot boundary segmentation depicted as a binary image. A parking block polygon is a rectangular representation of a parking block that is comprised of the detected parking spots with the same open-end orientations. Individual parking spots are depicted (green) triangles at their centroids. A blue line is a convex hull of all of the detected parking spots.

(b) A distance map is computed from the obstacle map of a parking lot. Red regions are farthest ones from local obstacles depicted in blue. Red "x" marks represent local maxima in the distance map.

(c) Some of the local maxima are selected as rendezvous points that are used to determine when the search of a lane-graph topology stops. Green "*" marks rendezvous points.

(d) The radius of a circle centered on a local maximum is increased to find its maximal circle. Any local maxima within the maximal circle will be considered redundant and removed.

(e) Connection of visible neighboring vertices reveals the topology of a lane-graph.

(f) The output of our lane-graph generation algorithm. A vertex is represented as an intersection if its edges are more than 2.

Fig. 1: These figures show the sequence of our lane-graph generation algorithm. Viewed best in color.

non-drivable regions. Figure 1(b) depicts the resulting distance map where farthest points from local obstacles have highest values. Ridge points on the distance map, which are local maxima of the map, are good candidates for building a lane-graph because they are skeleton points of drivable regions.

To locate these local maxima, we use the discrete analog of derivative because the second derivative of a distance map function is zero when the magnitude of the derivative is extremal. To implement this idea, our search is carried out by investigating individual columns and rows in the distance map, I_{dt}. The distance map, I_{dt}, is a m-by-n real-valued matrix where $I_{dt}(i,j)$ is the distance transform value of the ith row and the jth column. The function, *find_local_maxima*(I_{dt}), computes numerical derivatives of individual columns (and rows) in the distance map using the forward difference, $dx_i(k) = I_{dt}(i,k+1) - I_{dt}(i,k)$, $(dy_j(k) = I_{dt}(k+1,j) - I_{dt}(k,j))$, where $k = 1,...,n-1$ [3]. This is the first derivative of the distance map that locates the changes of distance values in a column (or a row). We then compute the second derivative: Compute

[3] For example, computing the changes of distance values at the ith row, $dx_i(2) = I_{dt}(i,2) - I_{dt}(i,1), dx_i(3) = I_{dt}(i,3) - I_{dt}(i,2), ..., dx_i(n-1) = I_{dt}(i,n) - I_{dt}(i,n-1)$

the forward difference again only for elements which all of the $dx_i(k)$ are greater than zero. The value of the second derivative is zero when the distance value is a local extremum. A local extremum is a maximum if the slope of the second derivatives at its neighbor points is changed from negative to positive.[4] These points correspond to peaks in a column (or row) of the distance map. A cross-check of these points with other rows and diagonal elements in the matrix results in local maxima. Because of the discrete nature of an image, this method does not always guarantee to find all of the true extrema, but provide a sufficient number of local maxima for our lane-graph generation.

A simple connection of all of the detected local maxima might produce a lane-graph that has an unnecessarily detailed structure due to an imperfect obstacle map. For example, there is a small creek in the upper left corner of the figure 1(b) that depicts cracks of the obstacle map and causes the distance transform to produce a lot of small-valued local maxima. Therefore we should handpick some of the local maxima for constructing the topology of a lane-graph.

To facilitate the lane-graph building process, there are three initialization steps: elimination of irrelevant local maxima; sorting of the selected maxima; and generation of rendezvous points. The function, *prune_local_maxima*($\mathbb{M}$), remove any local maxima that the radius of its maximal circle is smaller than the average width of the detected parking spots because their surrounding regions are not wide enough for the navigation of a common-size vehicle. To define the maximal circle of a local maximum, the initial radius of an inscribed circle is set to the average width of the detected parking spots. The radius is increased until the circle touches any of neighboring obstacles. An inscribed circle is maximal if no other inscribed circle, without touching neighboring obstacles, contains it properly. The idea of maximal circle has been studied for shape recognition and abstraction [9]. Figure 1(c) depicts a set of the selected local maxima. A sorting of the selected local maxima in descending order of their distance map values is necessary because the surrounding region of a local maximum with higher value contains more important geometric structure in a parking lot and it should be considered before any other local maxima with smaller values. Lastly we need a criterion to determine when to stop our topology building step. One might think this iteration can be stopped when it connects all of the selected local maxima. However because of incomplete boundary segmentation result, a connection of all of the selected local maxima will result in a lane-graph that is inconsistent to the actual shape of drivable regions. To properly stop the iteration while ensuring the consistency of a resulting graph, we utilize the locations of some local maxima. We call them rendezvous points because their locations must be visited for building a consistent lane-graph. A rendezvous point is a local maximum point that represents more than one detected parking spot. Given the fact that our parking spot detection algorithm recovers the open-end orientation of a parking spot [14], in the function, *define_rendezvous_point*($\mathbb{M}_1$), for each of the detected parking spots, we find the local maximum point that is orthogonally closest to that parking spot. These local maxima points are orthogonal projections of the detected parking spots onto the center-line of drivable regions.

Although the collection of rendezvous points does not always cover all of the area of the drivable regions, the ordering of the selected local maxima based on their values ensures that the topology visiting all of the rendezvous points completely aligns with the area of drivable regions. Thus the topology of a lane-graph is consistent to the shape of drivable regions if it includes all of the rendezvous points. Figure 1(c) depicts the identified rendezvous points.

Once these initialization steps are completed, our algorithm examines each of the selected local maxima by investigating its surrounding region. In particular, for each of local maxima, the function, *find_maximal_circle*(m_i), defines a maximal circle centered at the local maxima under investigation. Any local maxima within the maximal circle can be removed from further consideration. These steps are repeated until all of the rendezvous points are visited. Figure 1(d) depicts the selected local maxima. In this example, there are 402 local maxima initially identified and 55 of them are selected as vertices for possible lane-graphs.

The last step of our algorithm is to connect each of the identified vertices to neighboring vertices if the vertex is visible from its neighbors. The visibility test checks whether a line segment between two vertices passes through any obstacles and any neighboring vertices. In particular, we use the Bresenham algorithm to examine image coordinates along the line linking two vertices whether they are overlapped with any non-drivable regions.

B. *Results of Lane-Graph Generation*

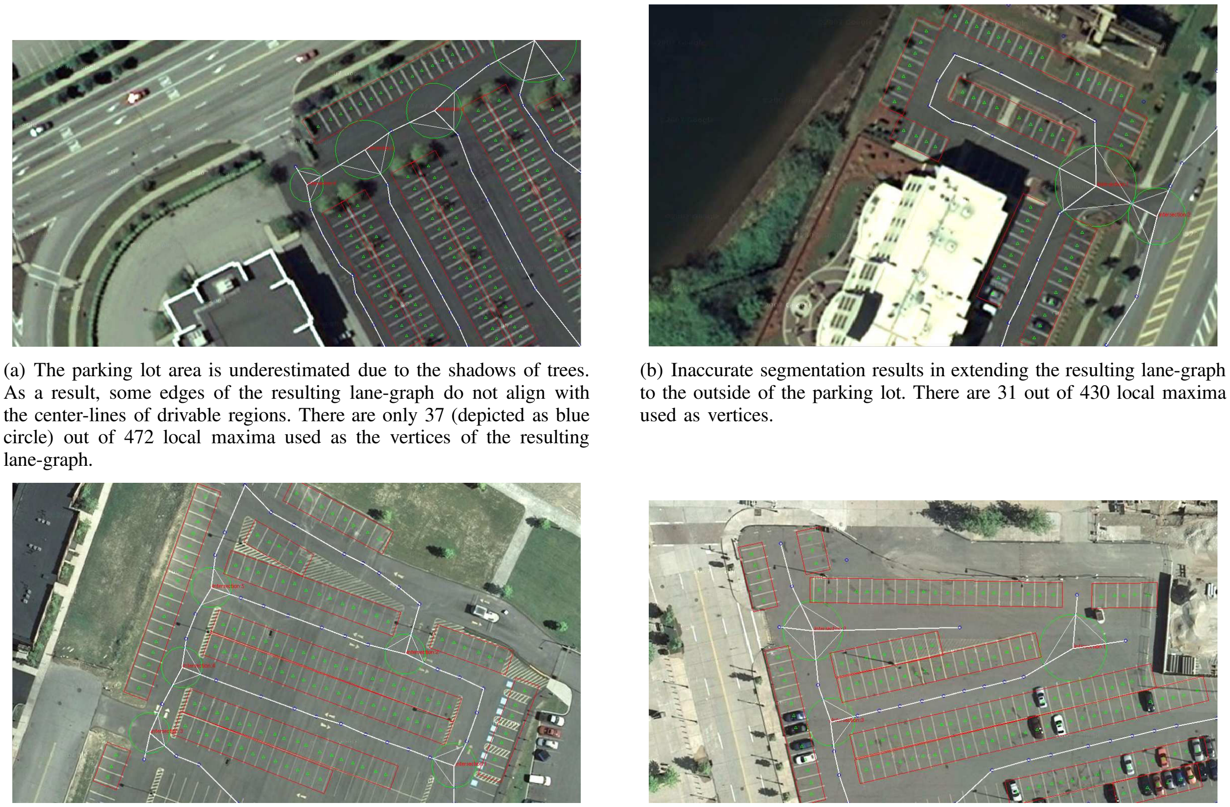
Figure 2 shows some results of our lane-graph generation algorithm. Testing ortho-images are downloaded from the *Google* map service.[5]

For each of the testing images, we first execute our parking spot detection and parking lot boundary segmentation algorithms to produce the map of non-drivable regions in the image, and then run our lane-graph generation algorithm.

For most test images, our algorithm works well in that the topologies of resulting lane-graphs concisely represent drivable regions in parking lot images: edges align with the center lines of road segments and vertices at intersection points connect merging road segments. Results shown in figure 1(f), 2(a), 2(b), 2(c), and 2(d) are example images of successful cases. These successful results rely on the map of non-drivable regions: highly accurate results of parking spot detection and parking lot boundary segmentation. Any false positive result by either of these tasks overestimates the true area of drivable regions in a parking lot image. For example, in the figure 2(a), the shadows of trees are segmented as non-drivable regions and the resulting edge

[4] One can also find a local minimum by looking at the point where its second derivative is zero and the slope is changed from positive to negative. In practice, this extrema search can easily be done by convolving individual columns (or rows) with a Laplacian operator, $[1, -4, 1]^T$ and then looking for the changes of slopes.

[5] http://map.google.com



(a) The parking lot area is underestimated due to the shadows of trees. As a result, some edges of the resulting lane-graph do not align with the center-lines of drivable regions. There are only 37 (depicted as blue circle) out of 472 local maxima used as the vertices of the resulting lane-graph.

(b) Inaccurate segmentation results in extending the resulting lane-graph to the outside of the parking lot. There are 31 out of 430 local maxima used as vertices.

(c) Some of the parking spots at the bottom left are missed by the parking spot detection. This cause our algorithm to overestimate the actual parking lot area. There are 43 out of 428 local maxima used as vertices.

(d) There is an isolated vertex at the top left because of inaccurate segmentation. There are 35 out of 298 local maxima used.

Fig. 2: Four additional examples of lane-graph generation. Viewed best in color.

passing through that region is bended to avoid false non-drivable regions. A drivable region with different visual appearances such as shadows and occlusions is one of the common causes for erroneous segmentation and parking spot detection results. From our previous study [15], we developed a very accurate parking spot detector that produces a very small false positive errors (less than 0.06%). In this study, most of the overestimated drivable region is caused by parking lot boundary segmentation.

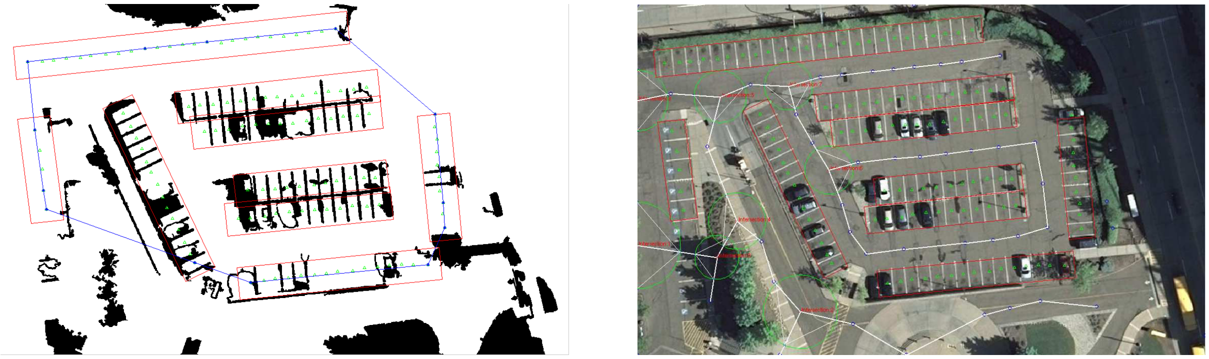
Figure 3 shows an example that our algorithm did not work well. The segmentation result in the figure 3(a) produces the overestimated boundary, resulting in a lane-graph inconsistent to the actual drivable regions. Owing to the relatively accurate parking spot detector, the right side of drivable regions in the figure 3(b) is partially covered by the resulting lane-graph. The lane-graph is inconsistently generated primarily due to the simplicity of our segmentation algorithm in that it connects two neighboring pixels if their image characteristics (i.e., magnitudes of image gradients and color) are similar. Thus it fails to correctly segment regions when the appearance of pixels greatly vary.

By contrast, a false negative one underestimates the true area of drivable regions in a parking lot image. For example, in the figure 2(c), the edge of the intersection at the bottom left is passing through a part of the non-drivable regions because the parking block is completely missed by our parking spot detection algorithm.

IV. Conclusions and Future Work

We have presented ortho-image analysis algorithms that produce a lane-graph of drivable regions in a parking lot image. Our previous work in parking spot detection and parking lot boundary segmentation are used to produce a map of non-drivable regions of a parking lot [14], [15]. Our lane-graph generation algorithm first produces a distance transform map from the obstacle map to reveal the skeleton of drivable regions; second it identifies the locations of peaks in the distance transform map; and lastly connects some of these peaks to produce a lane-graph consistent to drivable regions.

Our contribution includes novel computer vision algorithms that parses a static parking lot ortho-image and produces the lane-graph of a parking lot's drivable regions. As the geometric model of a parking lot, the resulting lane-graph is a step toward creating consistent and reliable autonomous parking. Additionally, our lane-graph generation



(a) A fail case. An overestimated parking lot area.

(b) Due to the inaccurately estimated parking lot boundary, the resulting lane-graph fails to capture the topology consistent to drivable regions.

Fig. 3: The topology of the resulting graph is sensitive to the accuracy of non-drivable map. Viewed best in color.

algorithm can be utilized to extract the skeleton of non-convex polygonal shapes.

Future work will improve the performance of our parking lot boundary segmentation algorithm because the main drawback of our algorithm is inaccurate boundary segmentation. However it is known to be very hard to achieve a highly precise segmentation result, particularly in a self-supervised framework, we will also work on making our lane-graph generation algorithm robust to noise in the map of non-drivable regions. Our ultimate goal is to produce the lane-graph of a route from a series of aerial images and to use the resulting lane-graph for autonomous driving. In this regard, we will test resulting lane-graphs with actual autonomous parking and investigate the applicability of our algorithm for analyzing the geometry of street aerial images.

V. Acknowledgments

This work is funded by the GM-Carnegie Mellon Autonomous Driving Collaborative Research Laboratory (AD-CRL).

References

[1] Oliver Buhler and Joachim Wegener, Automatic testing of an autonomous parking system using evolutionary computation, *Journal of Society of Automotive Engineers*, 1844: 115-122, 2004.

[2] David C. Conner, Hadas Kress-Gazit, Howie Choset, Alfred A. Rizzi, and George J. Pappas, Valet parking without valet, In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS-07), pp. 572-577, 2007.

[3] Yao-Yi Chiang and Craig A. Knoblock, Automatic extraction of road intersection position, connectivity and orientations from raster maps, In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (GIS-08), 2008.

[4] Ching-Chien Chen, Craig A. Knoblock, and Cyrus Shahabi, Automatically conflating road vector data with orthoimagery, *GeoInformation*, Vol. 10, pp. 495-530, 2006.

[5] Yang Chen, Runsheng Wang, and Jing Qian, Extracting contour lines from common-conditioned topographic maps, *IEEE Transactions on Geoscience and Remote Sensing*, 44(4): 1048-1057, 2006.

[6] Dmitri Dolgov and Sebastian Thrun, Autonomous driving in semi-structured environments: Mapping and planning, In *Proceedings of IEEE International Conference on Robotics and Automation* (ICRA-09), pp. 3407-3414, 2009.

[7] C.U. Dogruer, B. Koku, and M. Dolen, Global urban localization of outdoor mobile robots using satellite images, In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS-08), pp. 3927-3932, 2008.

[8] Martial Hebert, Building and navigating maps of road scenes using an active sensor, In *Proceedings of IEEE International Conference on Robotics and Automation* (ICRA-89), pp. 1136-1142, 1989.

[9] Ron Kimmel, M. Brostein, and A. Brostein, *Numerical Geometry of Images: Theory, Algorithms, and Applications*, Springer, 2003.

[10] Rainer Kummerle, Dirk Hahnel, Dmitri Dolgov, Sebastian Thrun, and Wolfram Burgard, Autonomous driving in a multi-level parking structure, In *Proceedings of International Conference on Robotics and Automation* (ICRA-09), pp. 3395-3400, 2009.

[11] Martin Persson, Tom Duckett, and Achim Lilienthal, Improved mapping and image segmentation by using semantic information to link aerial images and ground-level information, *Recent Progress in Robotics*, pp. 157-169, 2008.

[12] Yizhi Qu, Lingxi Li, Yaobin Chen, and Yaping Dai, Design of fault tolerant controllers in parallel parking systems, In *Proceedings of International IEEE Conference on Intelligent Transportation Systems*, 2009.

[13] Chris Scrapper, Ayako Takeuchi, Tommy Chang, Tsai Hong, and Michael Shneier, Using a priori data for prediction and object recognition in an autonomous mobile vehicle, In *Proceedings of the SPIE Aerosense Conference*, 2003.

[14] Young-Woo Seo and Chris Urmson, Utilizing prior information to enhance self-supervised aerial image analysis for extracting parking lot structures, In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS-2009), pp. 339-344, 2009.

[15] Young-Woo Seo, Chris Urmson, David Wettergreen, and Jin-Woo Lee, Augmenting cartographic resources for autonomous driving, In *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (GIS-09), pp. 13-22, 2009.

[16] Boris Sofman, Ellie Lin, J. Andrew Bagnell, Nicolas Vandapel, and Anthony Stentz, Improving robot navigation through self-supervised online learing, In *Proceedings of Robotics Science and Systems* (RSS-06), 2006.

[17] Chris Urmson et al., Autonomous driving in urban environments: Boss and the Urban Challenge, *Journal of Field Robotics: Special Issues on the 2007 DARPA Urban Challenge*, pp. 425-466, 2008.

[18] Xiaoguang Wang and Allen R. Hanson, Parking lot analysis and visualization from aerial images, In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 36-41, 1998.