

Lane Boundary Extraction from Satellite Imagery

Andi Zang

Department of Electrical Engineering and Computer
Science, Northwestern University
2145 Sheridan Rd
Evanston, Illinois 60208
andi.zang@u.northwestern.edu

Zichen Li

Department of Computer Science, New York University
251 Mercer St
New York City, New York 10012
zichenli@nyu.edu

Runsheng Xu

Department of Electrical Engineering and Computer
Science, Northwestern University
2145 Sheridan Rd
Evanston, Illinois 60208
runshengxu2017@u.northwestern.edu

David Doria

HERE North America
425 W Randolph St
Chicago, Illinois 60606
david.doria@here.com

ABSTRACT

Automated driving is becoming a reality. In this new reality, High Definition (HD) Maps play an important role in path planning and vehicle localization. Lane boundary geometry is one of the key components of an HD Map. Such maps are typically created from ground level LiDAR and imagery data, while useful in many ways, have many limitations such as prohibitive cost, infrequent update, traffic occlusions, and incomplete coverage. In this paper, we propose a novel method to automatically extract lane boundary from satellite imagery using pixel-wise segmentation and machine learning, and convert unstructured lines into structured road model by using hypothesis linking algorithm, which addresses the aforementioned limitations. We also publish our experiment dataset consisting of satellite imagery and the corresponding lane boundaries as ground truth to train, test, and evaluate algorithms.

KEYWORDS

HD Maps, lane boundary geometry, satellite image processing

ACM Reference format:

Andi Zang, Runsheng Xu, Zichen Li, and David Doria. 2017. Lane Boundary Extraction from Satellite Imagery. In *Proceedings of ACM SIGSPATIAL WORKSHOP ON HIGH-PRECISION MAPS AND INTELLIGENT APPLICATIONS FOR AUTONOMOUS VEHICLES, Los Angeles, California, USA, November 2017 (AUTONOMOUSGIS'17)*, 8 pages.

DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION AND MOTIVATION

Highly accurate, precise, and detailed lane-level maps, also known as High Definition (HD) Maps, as described in Open Lane Model by the Navigation Data Standard (NDS) [27], are critical to enable safe automated driving. Lane-level maps augment vehicle sensor information for contextual analysis of the environment, assist the vehicle

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

AUTONOMOUSGIS'17, Los Angeles, California, USA

© 2017 ACM. 978-x-xxxx-xxxx-x/YY/MM...\$15.00

DOI: 10.1145/nnnnnnn.nnnnnnn

in executing controlled maneuvers beyond its sensing range, and provide precise vehicle positioning and orientation in map coordinate. These maps work in conjunction with localization objects such as signs, barriers, poles, and surface markings to provide the vehicle a more comprehensive and accurate knowledge of the environment. One of the most important attributes in lane-level maps is the geometry of the lane boundaries. As there are millions of kilometers of roads in the world, it is cost-prohibitive and time-consuming to manually create and maintain such lane information at a centimeter-level precision. Autonomous vehicles are becoming more of a reality, but it is still way too far to achieve fully autonomous driving of all vehicles, all around the world/road scenarios, and this evolution may take decades. In the foreseeable future, high level road networks (i.e. interstate, expressways, and freeways) - compared to low level networks (i.e. principal/minor arterial and local roads), they have higher quality constructions, clearer traffic condition, and consume higher logistics and transportation labor cost - are the first stage to apply autonomous driving techniques (e.g. autonomous truck [9]). The lengths of three largest highway networks in the world, U.S., China, and India, are 103, 446, and 79 thousand kilometers [1, 28] respectively. Considering current reported HD Maps manual modeling efficiency, it takes years to mapping the entire high level road networks even with thousands workers and high costs to maintain the data, which motivates us to concentrate on high level road networks in this paper.

Ground based data - mainly comes from imagery and LiDAR - is the primary data sources used to automatically extract lane information [13] in academia and industry. It has distinct advantages such as high precision, rich information (e.g. color and geometry), and hard to be affected by top-down occlusion (e.g. trees, buildings, overpasses and tunnels). Some researchers [15, 20, 32] have proposed to detect road surface and lane markings from LiDAR using the highly accurate and precise 3D measurements in a LiDAR point cloud. Moreover, a point cloud aligned with perspective imagery can be used to generate training data [10] to assist lane-marking detection in perspective imagery. On the other hand, ground based data have many limitations such as object occlusion, infrequent updates, prohibitive cost (i.e. data storage, computation and acquisition) and limited coverage.

Object - especially moving object - occlusion is one of the biggest challenges of road/lane extraction in either LIDAR point clouds and ground-level imagery. As an example, in Figure 1, a truck that drives alongside the data acquisition vehicle at a similar speed produces a "wall" in the middle of the road. This scenario happens frequently and unavoidable everywhere on road networks, which leads to an empty region behind the "wall" (black region behind highlighted yellow "wall" shown in Figure 1(b)), and makes road model creation much difficult/impossible.

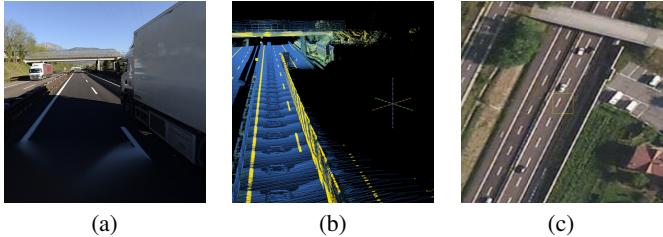


Figure 1: A common moving object occlusion scenario on highway. a) A truck drives alongside the acquisition vehicle, b) a "wall" caused by truck occlusion in the LiDAR point cloud, and c) the corresponding satellite image.

Compared to ground based data, aerial data - for example, aerial and satellite, or even drone in future - has its advantage over terrestrial data due to its frequent updates, global coverage, and occlusion non-sensitive, but has lower resolution. Hence, aerial data is an ideal data source to complement the ground based road model automation solutions.

In this paper, we propose a novel, automated lane boundary extraction technique from satellite imagery. Our approach consists of two stages: pixel-wise line segmentation and hypothesis linking. The pixel-wise line segmentation approach first performs patch-based lane-marking classification. Then for each positive patch, we segment line pixels to generate line candidates. Hypothesis linking step groups, classifies, connects line candidates by minimizing a cost function to generate a structured lane model. A formalized road model accuracy metric is designed to evaluate the results rigorously. For further evaluation, we created a ground truth data set by manually labeling lane boundaries for some sections of the road.

2 RELATED WORK

Techniques for road extraction from orthogonal imagery can be traced back to more than forty years ago; however, limited by image resolution (typically over 2 meters per pixel), traditional approaches rely on edge detection, color segmentation, linear feature detection, and topological linking [3, 31] to extract road networks. In recent years, patch/pixel-wise machine-learning based approaches have been proposed to detect the road region [6, 11, 14, 18, 19] with the increasing of satellite image resolution. These road centric approaches still cannot model the lane-level features to meet HD Maps requirements even though the satellite image resolution has improved to sub-meter level. Recently¹, satellite imagery can have a

resolution of 0.5 meter per pixel or even higher (e.g. WorldView-4), which allows us to utilize the state-of-the-art approaches to model lane-level features [16, 17, 22, 26]. With the developing of Deep Learning, more image segmentation approaches [7] also broaden the horizon to tackle this problem.

There are many challenges we need to solve:

- (1) Ground truthing of accurate line location in high resolution satellite imagery is difficult. Even though the satellite imagery data is claimed as "high" resolution, its ground resolution is compatible with line marking paint width, which means the true location of each polypoint is only one pixel width - compared to line marking paint width in perspective imagery is normally from fifty pixels to ten pixels, from near end to far end. Direct Deep Learning based segmentation approaches are sensitive to such high ambiguous ground truth ratio.
- (2) The evaluation metric of road lane boundary model accuracy and performance need to be redefined. Road lane boundary model consists of bundle of lines, and these lines need to be considered as one structured object. In State-of-the-art (e.g. [22]), the road model is always represented as a collection of unstructured lines without attributes. This may cause road segments with same "accuracy" have totally different visual effects. Also, pixel-wise Intersection Over Union (IOU) percentage [25, 26] is always considered as the metric to evaluate lane boundary accuracy. This evaluation system is less persuasive since road lane boundary is geometry polyline.

3 METHODOLOGY

Our lane-boundary-geometry extraction approach consists of two stages. We use the ground truth lane boundary geometry and the corresponding satellite imagery from the Bing Maps Tile Server [4] as input, project the lane boundary lines into image, crop the resulting image into small patches, and train our patch level classifier. For extraction, our approach uses a pre-trained classifier, a trajectory of target road segment, and corresponding satellite imagery as input. It then detects patch-level lane marking candidates [8], segments the pixel-wise lane marking candidates, and links [26] the pixel-wise candidates to generate the lane boundary geometry.

3.1 Patch Creation and Patch Level Classification

The objective of this step is to build a classifier that can determine whether an image patch contains any lane marking pixels. Even though the ground truth road model is organized in a chunk-wise structure (examples of "chunk" are illustrated in blue boxes, Figure 12), due to the mechanics of the tile system, generating training patches in chunks unnecessarily queries the image server twice (each tile always contains two to three chunks at high resolution levels). Hence, our solution operates tile-wise - for each tile along the trajectory, project all control points of each functional line and connect them. To reduce noise (e.g. lane marking pixel from adjacent road surfaces), the surface region is bounded by road boundaries (given

¹DigitalGlobe WorldView-1, which was launched in 2007, has 0.5 meter/pixel resolution, but the U.S. government banned companies from offering commercial satellite views

with a pixel resolution better than 50 centimeters. DigitalGlobes Satellite Pics Are So Good Theyre Almost Illegal, <https://www.nbcnews.com/science/space/digital-globes-satellite-pics-push-legal-limit-n313026>

in the ground truth dataset). Samples containing lane marking pixels and road surface regions are illustrated in Figure 2.



Figure 2: Satellite tile images with ground truth lane boundary geometry overlaid at tile location [557869, 363521], [557909, 363513], [558036, 363507] and [557879, 363518] at level 20 from left to right. The road region is highlighted in green, and is bounded by the road boundaries. Lane markings are highlighted in red.

A sliding window is designed to crop training patches from corresponding satellite image within the road surface. The label for each patch is determined by whether there are any lane marking pixels in the current patch. To reduce misleading ground truth patches (e.g. the patch contains two independent lines), an appropriate window size should be thinner than a single lane width, which we take to be 3.5 meters [34]. In this work we use imagery from tile level 20, so the ground resolution is approximately 0.15 meters per pixel. This means the patch size should be less than 24 pixels. Examples of positive (containing lane marking pixel) and negative patches are shown in Figure 3.

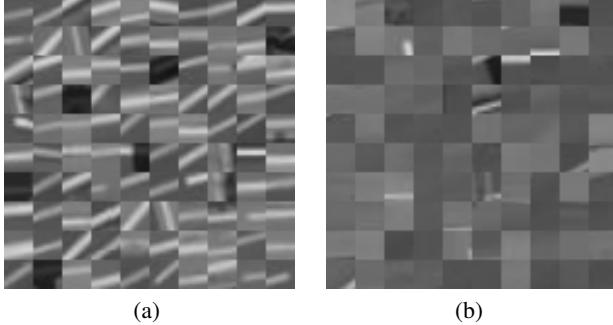


Figure 3: a) 100 sample patches of original orientation lane-marking patches and b) non-lane-marking patches. Note that some negative examples (e.g. row 4, column 6) look almost identical to positive examples cause of similar line-like objects, such as the part of truck cargo.

With cropped positive and negative samples and labels, the patch level classifier can be trained by using varying machine learning methods such as Random Forest, Support Vector Machine and Neural Network. A lane marking probability map is the sum up of all its patches at their locations. Given a satellite tile image (shown in Figure 4 (a)), its corresponding probability map of patch level lane marking with certain configuration (patch size is 12 pixels, use pixel representation feature and Random Forest classifier) is illustrated in Figure 4 (b).

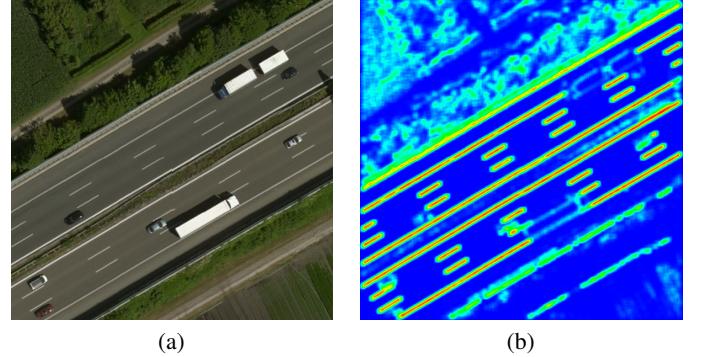


Figure 4: a) An original satellite tile image and b) its patch level lane marking probability map at location 48.2203°, 11.5126°. Red pixels represent the most lane-marking-like points.

3.2 Pixel-wise Lane Marking Segmentation

Our patch level classification produces a probability map that contains high probability lane marking regions (red pixels shown in Figure 4(b)). However, since each region is several pixels wide (the exact size depends on patch size, step length, and the actual paint stripe width), this region cannot directly be used to represent the lane marking to desired HD Map quality levels [5].

To segment and locate precise lane marking pixels, we consider pixels with the highest intensity in each slice (pixel by pixel) of lane marking region perpendicular to road trajectory as lane-marking candidate. Then, we fit a line segment through the lane marking pixel candidates. For example, in Figure 6, assuming the trajectory is facing up, for each row of this region, the highest intensity pixels have value 151, 154, and 150, respectively. The lane marking should be the center line (sixth column) of the figure. An example lane marking segmentation result is illustrated in Figure 5.

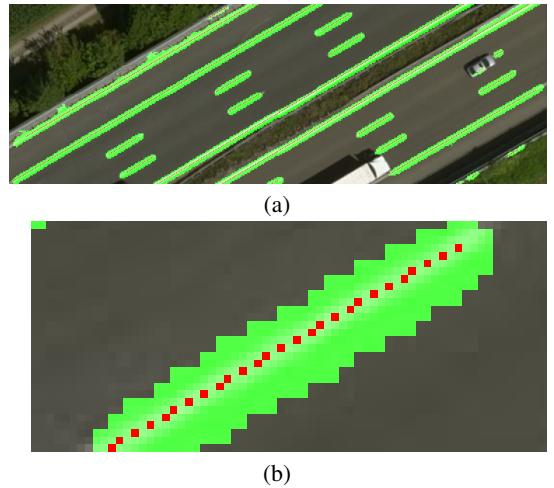


Figure 5: Lane marking region candidates (green regions) and pixel-wise lane marking pixel candidates (red dots) overview (top) and zoom view (bottom).

Although the satellite image resolution is close to the lane marking width, the lane markings will always appear blurred due to image compression, hardware imperfections (imperfect lenses, etc.), and optical limitations (i.e. angular resolution). To produce more precise lane marking pixel locations, we introduce sub pixel-wise segmentation. For each slice of the lane marking region, we fit a Gaussian model (green lines in Figure 6) and find the peak of each model (yellow circles in Figure 6). Then, the lane marking pixel location becomes sub-pixel-wise instead of the direct pixel-wise of each slice (row) of the lane marking region.

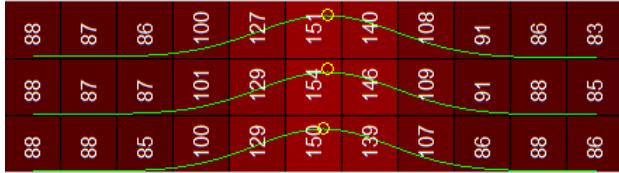


Figure 6: Pixel-wise and sub pixel-wise lane marking segmentation visualization. The numbers inside each pixel represents its intensity value converted from the raw RGB image. Green curves are the fitted Gaussian models of each row and yellow circles point out their peaks.

3.3 Line Candidates Grouping, Classification and Linking

The previous step generates unstructured line segments without function label (solid/dashed line). Because of occlusion (i.e. trees, vehicles, buildings, and their shadows) and poorly painted lane markings(examples shown in Figure 7 (a) and (b)), fewer true lane marking lines will be detected. Conversely, more misleading lines (false positive) will be detected if lane-marking-like objects appear (i.e. guardrail, curb, wall shown in Figure 7 (c) and (d)).



Figure 7: Four phenomena that can cause problems. Mis-detection: a) bad painting quality and b) shadow. Misleading objects that cause false positive: c) high reflective metal guardrail and d) cement curb.

Our method of transforming the unstructured lines to structured lines with function labels contains three steps: grouping line candidates from each chunk, classifying the function of each line group, and linking to interpolate the missing lines.

In the grouping step, the decision to add a line into a group depends on the relative distance² from the current line to all other

²The relative distance here is the average distance between each point from one line segment to the other.

line candidates in the current chunk, in the neighboring chunk(s), and their relative distances to road centerline/vehicle trajectory. For example, line candidates (gray) from four continuous chunks and vehicle trajectory (blue dashed) are illustrated in Figure 8 (a). After the grouping step, five groups are generated and colored in Figure 8 (b). On a certain portion of the road, for each group, the function label is determined by the ratio of the total length of detected line segments belonging to this group to the total length of road that contains this line group. Typically, considering reasonable mis-detections and incorrect detections, the length ratio of dashed line is significant lower than the ratio of solid line. In the task of modeling highway roads, there is an assumption that each road portion can have at most two solid lines bounding the (drivable) road surface. Figure 8 (c) illustrates the groups after the classification step, solid lines and dashed lines are colored in dark red and dark green, respectively. Lines outside of the solid lines (outside of the drivable road surface) are colored in gray and will be ignored. In the final step, if one chunk does not contain a line that belongs to the group which passes this chunk, a synthetic line will be interpolated (light green lines shown in Figure 8 (d)).

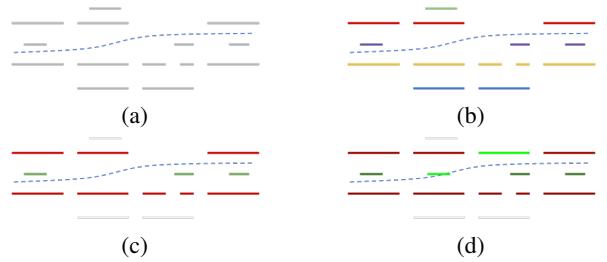


Figure 8: Illustrations of Line Candidates grouping, classification and linking steps.

Notice that in this grouping, classification, and linking procedure, numerous thresholds and constraints (i.e. distance threshold, search range, etc.) are needed to control the process. Generally speaking, if we abstract all these variables into one variable: loose (longer search range, wider distance threshold) and tight (shorter search range, narrower distance threshold) to reflect the abstract performance (i.e. geometry performance and function accuracy) of the model, the geometry accuracy increases with the tightness, and function accuracy is opposite. Hence, it is a trade off between function accuracy and geometry accuracy, and the balance point depends on varying road conditions and specific road modeling targets.

4 LANE BOUNDARY GROUND TRUTH COLLECTION

In [26], the author uses manually-drawn, pixel-based ground truth, represented in "mask" format [25], to evaluate accuracy. The author does not provide a detailed description and statistic of the dataset. The number of ground truth masks is very small (50 pictures), which also limits its persuasiveness.

To produce our lane boundary geometry dataset, we built an interactive tool which allows us to manually draw lane boundaries by displaying images various sources (i.e. point cloud projection,

satellite imagery, etc.). With this fully manually approach, we are able to label boundaries at 12.8 meters per minute³. However, to improve this rate, similar to previous work [10], we first perform automatic lane boundary geometry extraction. Then we use our tool to edit (delete, move, insert, etc.) the control points on lane boundary lines to make align them with the displayed imagery perfectly to achieve 29.2 meters per minute efficiency.

4.1 Lane Boundary Data Description

We collected lane boundary geometry on the Autobahn A99 (Bundesautobahn 99) in Germany, from 48.2206° , 11.5153° to 48.2057° , 11.4586° , divided into seven portions (five for training and two for testing) to exclude overpass structures and other unexpected scenarios.

This dataset contains training data (approximately 10.14 kilometers) and testing data (6.07 kilometers), which follows similar conventions but has two key differences. In training set (coverage shown in Figure 9 in red), dashed lines are represented as isolated line segments (two end points), and aligned with satellite imagery coordinates. In the testing dataset (coverage shown in Figure 9 in yellow), dashed lane markings belong to one set of continuous/sequential control points as they are semantically treated as one line and aligned with point cloud. All control points are placed in the center of their corresponding lane marking.

In this dataset we also include road boundaries (e.g. guardrail and curb) and use them to separate road surface from non-road surface (to exclude lane-marking like objects outside the road surface).

To represent lane boundary geometry and lane boundary lines for user convenience, lane boundary lines are evenly divided into 12-meter chunks. Each chunk is provided in a single JSON file and the structure document can be found in dataset.

4.2 Errors in latitude, longitude, and altitude

Aerial imagery and point clouds are stored/represented in different coordinate systems - Mercator projection [4] and Cartesian [2, 24] - because of their acquisition techniques. The aerial image tile system is designed to fulfill historical requirements, but it has heavy distortion.

To avoid distortion, point cloud processing procedures and our labeling tool are designed to process data in a Cartesian coordinate system. The different coordinate systems lead to a slight distance error when lines are labeled from these two layers. Let d_x and d_y denote vertical and horizontal distance in a local projection plan, and ϕ and λ denote latitude and longitude values. Assume location $[\phi, \lambda, 0]$ at zoom level l , the Euclidean distance $d_p(\phi, \lambda, d_x, d_y)$ between the points back-projected through Mercator projection and Cartesian coordinate transformations of d_x, d_y is complicated. After simplification, d_p can be represented as a function of latitude ϕ , pixel shift $[d_x, d_y]$, and zoom level l , and the illustration of function $d_p(\phi, d_x, d_y, l)$ at certain zoom level $l = 20$ and $d_y = 0$ is shown in Figure 10. According to precision requirements from most "HD" definitions [12, 29, 30], the error caused by fusion of two coordinate systems (less than 5 cm all around the world at tile level 20) does not have an impact on the final accuracy.

³Time efficiency is dependent on the number of lanes and the road structure. In our dataset, the majority of chunks contain 4 lane boundaries.

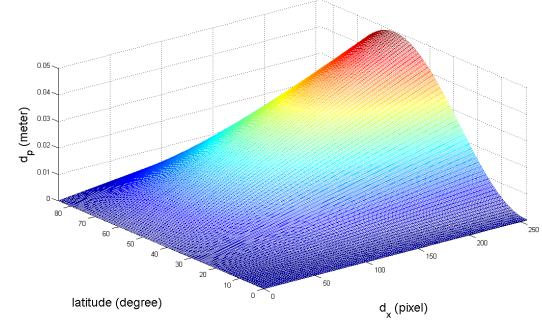


Figure 10: Illustration of $d_p(\phi, d_x, d_y, l)$ when $l = 20$ and $d_y = 0$.

5 EXPERIMENTS

In the proposed methodology, we test numerous patch configurations (e.g. patch feature representation, size, etc.) and machine learning techniques to determine the best classifier. We then run our end-to-end program to extract lane boundary geometry using this classifier. In this section, we present both patch level accuracy and final extracted model accuracy by using the metrics described in Section 5.1.

5.1 Accuracy Definition

Lane boundaries are a collection of polylines in control point representation [23], therefore we cannot simply report the accuracy in pixel-wise representation as [26] proposes.

Considering the misalignment between ground truth and modeling coordinates, we propose two metrics for a performance score: **function level** and **geometry level** accuracy. Theoretically, the transform to align the two coordinate systems is affine (translation, scale, rotation, and skew). However, in our task, scaling, rotating, and skewness are unnoticeable so they can be ignored, so only translation is considered.

Given ground truth lines $L_i = \{l_{i,1}, l_{i,2}, \dots, l_{i,n}\}$ and predicted lines $L'_i = \{l'_{i,1}, l'_{i,2}, \dots, l'_{i,m}\}$ from i th chunk, the first step is to match $l \in L_i$ and $l' \in L'_i$. Let $d(l, l')$ denote the distance between lines l and l' with sign (for example, left for negative) and $Pair(l_i, L'_i)$ denotes the paired line in L'_i of l_i . To align two models we use each l_i and each l'_i , to find

$$\operatorname{argmin}_{\text{all pairs}} \left(\sum_{l_{i,j} \in L_i} (d(l_{i,j}, \operatorname{Pair}(l_{i,j}, L'_i))) \right)$$

Given a distance threshold T_d , we consider a pairing of l'_i and l_i to be correct if their functions are matched and $d(l'_i, l_i) < T_d$. Then, the accuracy of the predicted model compared to ground truth can be represented in

$$\text{precision}_{\text{function}} = \frac{\text{number of correct detections}}{(\|L\|)}$$

$$\text{recall}_{\text{function}} = \frac{\text{number of correct detections}}{(\|L'\|)}$$

$$L = \{L_i\}, L' = \{L'_i\}, i \in \text{road}$$



Figure 9: Illustration of our ground truth portions highlighted on aerial imagery, red for training and yellow for testing.

To calculate geometry accuracy, for each pair $\{l_{i,j} : \text{Pair}(l_{i,j}, L'_i)\}$ in L_i , the *shift* is defined as $\text{AVG}_i(d(l_i : \text{Pair}(l_{i,j}, L'_i)))$, L'_i and $L_i \in \mathbf{L}$, and the *performance_{geometry}* is defined as $\text{AVG}_i(1 - \frac{\sigma(d(l_i : \text{Pair}(l_{i,j}, L'_i)))}{\sigma_{\max}(\|L_i\|)})$, while $\sigma_{\max} = \sigma(L_{\max})$ is used to normalize precision for each chunk, where $L_{\max} = M \cup N$, set M contains $\text{floor}(\frac{\|L_i\|}{2})$'s T_d and set N contains $\text{ceiling}(\frac{\|L_i\|}{2})$'s $-T_d$ (for example, if $\|L_i\|$ equals 4, then $\sigma_{\max}(\|L_i\|) = \sigma([T_d, T_d, -T_d, -T_d])$). With this $\{\text{perfomance} : \text{shift}\}$ metric, we can present the lane boundary geometry reasonably if the alignment between two source coordinates is unknown. Also, we can tweak the parameters of end-to-end program to generate expected model depends on the project requirement. For example, if alignment is not a requirement and we want to address the LiDAR shadow issue from point cloud, we would need to tweak the configurations with the lowest *performance*.

5.2 Patch level accuracy

To find the best patch level classifier, we crop the tile image with given ground truth data as described in Section 3.1, with configurable variables such as as patch size (8, 12, 16, 24 pixels), patch features (pixel representation and gradient based features such as Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP)), and numerous machine learning techniques (Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN) and Convolutional Neural Network (CNN)) to evaluate their precision and recall. Figure 11 shows the performances of different training configurations of 10-fold Cross Validation.

According to our evaluation result, the change in patch size does not have a big impact on patch level performance. Considering the computational cost, a patch size of 12 is used in our final patch level classifier to generate a dense/smooth lane marking probability map shown in Figure 4 (b).

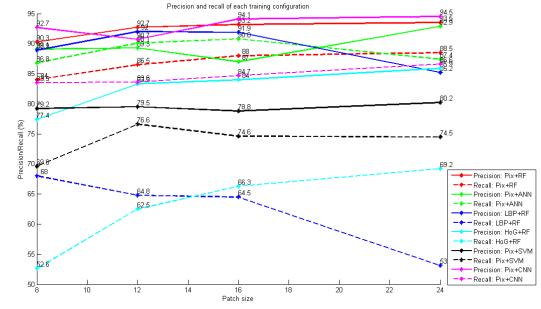


Figure 11: Patch level precision and recall of each training configuration.

5.3 Lane Boundary Geometry Accuracy

With the pre-trained classifier and our end-to-end solution, we experimented with parameters and thresholds as mentioned in Section 3.3 to evaluate the performance of our approach on the testing set. The function level and geometry performances are shown in Table 1 below.

Table 1: Model level performance with different conditions.

| Stage | Model Performance | | | | |
|----------------|-------------------|------------|----------------|------------|-------------------|
| | Function Level | | Geometry Level | | |
| | Precision (%) | Recall (%) | Performance | Shift (cm) | Median Error (cm) |
| Before linking | 84.96 | 95.58 | 76.01 | 0.37 | 4.26 |
| After linking | 95.79 | 95.79 | 74.38 | 0.87 | 4.24 |

Sample of ground truth lane boundaries and final extracted lane boundaries rendered on satellite imagery are shown in Figure 12 (a). Each chunk is bounded by a blue rectangle, yellow stars denote the road trajectory points, the ground truth model is rendered in red, and the resulting lane boundary geometry is rendered in green.

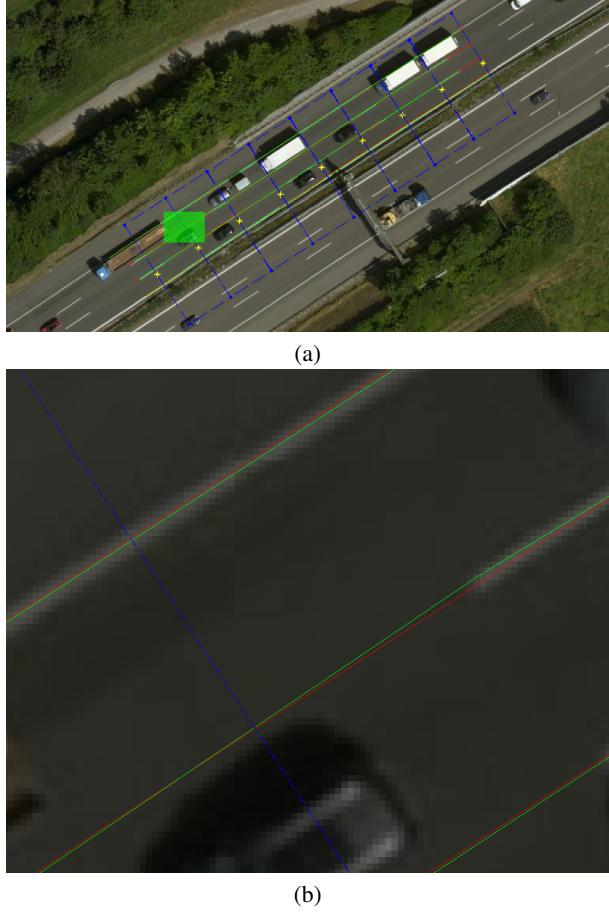


Figure 12: a) Result visualization overview and b) zoom view of the green box in (a).

The *shift* from experiment result shows the misalignment between ground truth coordinate system (from the point cloud) and the testing data coordinate system (from satellite imagery) was negligible on testing data. Due to the above-average conditions of the testing road (good paint quality, light traffic, etc.), the extracted lane boundary geometry achieved impressive results when measured against the ground truth before and after the linking stage at geometry level. Function level precision improves 10.83% by the linking stage while geometry level performance only dropped 1.63% due to the interpolated, synthetic lane boundaries. Even though the median error of the results is lower than 5 cm, limited by original satellite imagery resolution, we claim that our lane boundary geometry accuracy is 30 cm.

6 CONCLUDING REMARKS AND FUTURE DIRECTIONS

In this paper we present a novel approach to automatically extract lane boundary geometry from reasonably high resolution satellite imagery. It complements the existing ground-level based methods with advantages such as cost-effectiveness, wider coverage, and

better updatability. We also designed a comprehensive lane boundary geometry evaluation metric and published our lane boundary geometry dataset.

The following areas will be investigated for further improvement:

- (1) Elevation information is not available in satellite imagery. This can be solved with an additional data source such as the High Definition Digital Elevation Model (HD DEM) database [33].
- (2) Combine both ground level and airborne data to extract lane boundary geometry. Alignment of these two data sets is critical. Some feature points visible in both modalities could register the two data sets. For example, road surface markings and pole like objects are good correspondence candidates.
- (3) By adding absolute/relative spatial information to each pixel [21], a machine learning based (especially Recurrent Convolutional Neural Networks) lane marking pixel classification may substitute the approach of combining machine learning based lane marking patch classification and pixel-wise segmentation proposed in this paper.
- (4) This work has focused on highway lane boundary extraction. However, there is a large number of urban road networks that also need to be modeled in high definition for fully automated driving. Occlusions and shadows of urban roads in satellite imagery make it very challenging to apply our approach. In addition, the complex geometry and topology of urban road may require a different lane model for automation.

REFERENCES

- [1] Federal Highway Administration. 2013. Highway Statistics 2013. (2013). <https://www.fhwa.dot.gov/policyinformation/statistics/2013/hm220.cfm>
- [2] Gerardo Atanacio-Jiménez, José-Joel González-Barbosa, Juan B Hurtado-Ramos, Francisco J Ornelas-Rodríguez, Hugo Jiménez-Hernández, Teresa García-Ramírez, and Ricardo González-Barbosa. 2011. LIDAR velodyne HDL-64E calibration using pattern planes. *International Journal of Advanced Robotic Systems* 8, 5 (2011), 59.
- [3] Ruzena Bajcsy and Mohamad Tavakoli. 1976. Computer recognition of roads from satellite pictures. *IEEE Transactions on Systems, Man, and Cybernetics* 9 (1976), 623–637.
- [4] Bing. 2012. Bing Maps Tile System. (2012). <https://msdn.microsoft.com/en-us/library/bb259689.aspx>
- [5] Sebastian Bittel, Vitali Kaiser, Marvin Teichmann, and Martin Thoma. 2015. Pixel-wise segmentation of street with neural networks. *arXiv preprint arXiv:1511.00513* (2015).
- [6] D Chaudhuri, NK Kushwaha, and A Samal. 2012. Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques. *IEEE journal of selected topics in applied earth observations and remote sensing* 5, 5 (2012), 1538–1544.
- [7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* (2016).
- [8] Dragos Costea and Marius Leordeanu. 2016. Aerial image geolocalization from recognition and matching of roads and intersections. *arXiv preprint arXiv:1605.08323* (2016).
- [9] David Freedman. 2017. *10 Breakthrough Technologies*. MIT Technology Review.
- [10] Xiaodong Gu, Andi Zang, Xinyu Huang, Alade Tokuta, and Xin Chen. 2015. Fusion of color images and LiDAR data for lane classification. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 69.
- [11] Jeremy Heitz and Daphne Koller. 2008. Learning spatial context: Using stuff to find things. In *European conference on computer vision*. Springer, 30–43.
- [12] HERE. 2015. HERE introduces HD maps for highly automated vehicle testing. (2015). <http://360.here.com/2015/07/20/here-introduces-hd-maps-for-highly-automated-vehicle-testing/>

- [13] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. 2014. Recent progress in road and lane detection: a survey. *Machine vision and applications* 25, 3 (2014), 727–745.
- [14] Jiaxiang Hu, Anshuman Razdan, John C Femiani, Ming Cui, and Peter Wonka. 2007. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Transactions on Geoscience and Remote Sensing* 45, 12 (2007), 4144–4157.
- [15] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, and others. 2015. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716* (2015).
- [16] Gellert Matyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. 2015. Enhancing Road Maps by Parsing Aerial Images Around the World. In *Proceedings of the IEEE International Conference on Computer Vision*. 1689–1697.
- [17] Gellért Mátyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. 2016. HD Maps: Fine-grained Road Segmentation by Parsing Ground and Aerial Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3611–3619.
- [18] Volodymyr Mnih and Geoffrey E Hinton. 2010. Learning to detect roads in high-resolution aerial images. In *European Conference on Computer Vision*. Springer, 210–223.
- [19] Mehdi Mokhtarzade and MJ Valadan Zoej. 2007. Road detection from high-resolution satellite images using artificial neural networks. *International journal of applied earth observation and geoinformation* 9, 1 (2007), 32–40.
- [20] Joel Pazhayampallil and Kai Yuan Kuan. 2013. Deep Learning Lane Detection for Autonomous Vehicle Localization. (2013).
- [21] Pedro Pinheiro and Ronan Collobert. 2014. Recurrent convolutional neural networks for scene labeling. In *International Conference on Machine Learning*. 82–90.
- [22] Oliver Pink and Christoph Stiller. 2010. Automated map generation from aerial images for precise vehicle localization. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 1517–1522.
- [23] Andreas Schindler, Georg Maier, and Florian Janda. 2012. Generation of high precision digital maps using circular arc splines. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 246–251.
- [24] Brent Schwarz. 2010. LIDAR: Mapping the world in 3D. *Nature Photonics* 4, 7 (2010), 429.
- [25] Young-Woo Seo. 2012. *Augmenting cartographic resources and assessing road-way state for vehicle navigation*. Carnegie Mellon University.
- [26] Young-Woo Seo, Chris Urmson, and David Wettergreen. 2012. Ortho-image analysis for producing lane-level highway maps. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 506–509.
- [27] Navigation Data Standard. 2016. *Navigation Data Standard - Open Lane Model Documentation*. Technical Report. Navigation Data Standard.
- [28] Roadtraffic Technology. 2014. The worlds biggest road networks. (2014). <http://www.roadtraffic-technology.com/features/feature-the-worlds-biggest-road-networks-4159235/>
- [29] TMTpost. 2016. Baidu Driverless Cars Run In Wuzhen, Powered By Four Leading Technologies. (2016). <https://medium.com/@TMTpost/baidu-driverless-cars-run-in-wuzhen-poweredby-four-leading-technologies\discretionary{-}{}{}-tmtpost-53c0b3072cec>
- [30] TomTom. 2015. HD MAP - HIGHLY ACCURATE BORDER-TO-BORDER MODEL OF THE ROAD. (2015). <http://automotive.tomtom.com/uploads/assets/972/1474451229-hd-map-product-info-sheet.pdf>
- [31] John C Trinder and Yandong Wang. 1998. Automatic road extraction from aerial images. *Digital Signal Processing* 8, 4 (1998), 215–224.
- [32] Bisheng Yang, Lina Fang, and Jonathan Li. 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 79 (2013), 80–93.
- [33] Andi Zang, Xin Chen, and Goce Trajcevski. 2017. High-Definition Digital Elevation Model System (Vision Paper). In *Proceedings of International Conference on Scientific and Statistical Database Management*. ACM.
- [34] J. Zoesch and T.J. Zoesch. 2014. New Guidelines for the Geometric Design of Rural Roads in Germany. (2014).