Automatic Creation of Path Information on Digital Map

Haruka Iesaki[1*], Shuhei Naruse[1], Tsubasa Hirakawa[1], Takayoshi Yamashita[1], and Hironobu Fujiyoshi[1]

***Abstract*— Digital maps are data that numerically represent map information necessary for autonomous driving. The digital map is used for estimating vehicle positions, surrounding environments, and determining moving paths to a destination. Because it includes important information such as surrounding environments and road signs. Path information is also included for safe autonomous driving by following traffic rules. However, such path data is annotated manually and it is costly. Therefore, we aim to create path-planning data automatically on digital maps. Then, we propose a path-planning approach for vehicle movement. Our approach defines a cost function based on semantic scene labels and creates a minimum and optimal path. To estimate such a path, we use optimal rapidly-exploring random trees. Thus, it is possible to estimate an optimal path efficiently. Optimal cost is determined by learning with vehicle moving-path data. We also made a dataset for creating paths at intersections for quantitative evaluation. The results indicate that our proposed approach can create a optimal moving path.**

I. INTRODUCTION

Digital maps (High Definition Maps) have high precision data, numerically map information, is necessary for autonomous driving. They include various information to assist autonomous driving such as the surrounding environment, road signs, appropriate driving paths, and so on. Especially, paths information are important data for determining the first vehicle position on maps, and it is good for self-localization system. Paths are consist of many positions for precise autonomous driving by maps information such as traffic rule signs. Generally, paths are annotated manually and cost a lot to made. If it creates paths automatically, a straight path is able to be determined because it just calculates an intermediate position between lane markings. However, creating paths at an intersection is more difficult because there is no exact standard rules. (i.e., many types of lane markings and intersection shapes, and determined path).

Our goal is an automatic paths planning creation. In the paper, we focus attention on intersections maps because they do not have clear lane markings. We develop a path-planning approach and create a dataset at intersections to this end. We collected aerial images captured by Google Maps and annotated semantic scene labels and paths. Our path-planning approach defines a cost function that is based on scene context representing the surrounding environment, and regularizers with respect to path length and traffic rules. We, then, find a path, whose cost will be minimum by using optimal rapidly-exploring random trees (RRT*) algorithm. We evaluated our approach with the created dataset and we show that our approach can create optimal paths for self-driving vehicles.

* Corresponding author iesaki@mprg.cs.chubu.ac.jp
[1] Authors are with Chubu University, Kasugai, Japan

Our contributions are as follows:

- This is the first attempt to automatically create paths for self-driving cars. Automatic creations approaches have been proposed with other data, such as lane and/or traffic signs, though.
- We propose a path-planning approach to create optimal path data at intersections without any exact standard rules such as lane marking. The proposed approach creates paths based on start-goal positions and surrounding semantic label information. It automatically creates path's data for any shape of intersections.
- We create a dataset with respect to moving paths at intersections for quantitative evaluation. This dataset includes aerial images of intersections and annotated path data.

II. RELATED WORK

A. *Digital maps for self-driving*

Digital maps are used for enabling safe and highly accurate automatic driving. In particular, they use a own self-position estimation by using information from digital maps and surrounding information obtained from Light Detection and Ranging (LiDAR) or drive recorders [1], [2], [3], [4], [5], [6]. To achieve accurate vehicle localization, Tao *et al.* [4] proposed combining an extended Kalman filter with lane marking. Obst *et al.* [2] proposed a localization approach by using a global positioning system (GPS) to correct measurement errors which is caused by weather conditions. This approached method uses a three-dimensional digital map to achieve higher localization accuracy. These approaches have object information for landmarks such as line markers and stop lines on maps.

Several autonomous driving experiments were conducted involved these localization approaches in real scenes [7], [8]. In these experiments, the location of a self-driving car was estimated, and the route was determined for appropriate self-driving. In contrast, our approach creates positions where a vehicle should drive and add the information to a digital map. It was not determined the route in a real time, but we propose path creation using the stored route information.

It costs to create a wide range of digital maps. Therefore, extracting information approach, on digital maps from aerial and/or satellite images, have been widely studied [9], [10], [11], [12], [13]. Thanks to the development of deep neural networks, a number of approaches based on convolutional neural networks (CNNs) have been developed. Azimi *et al.*

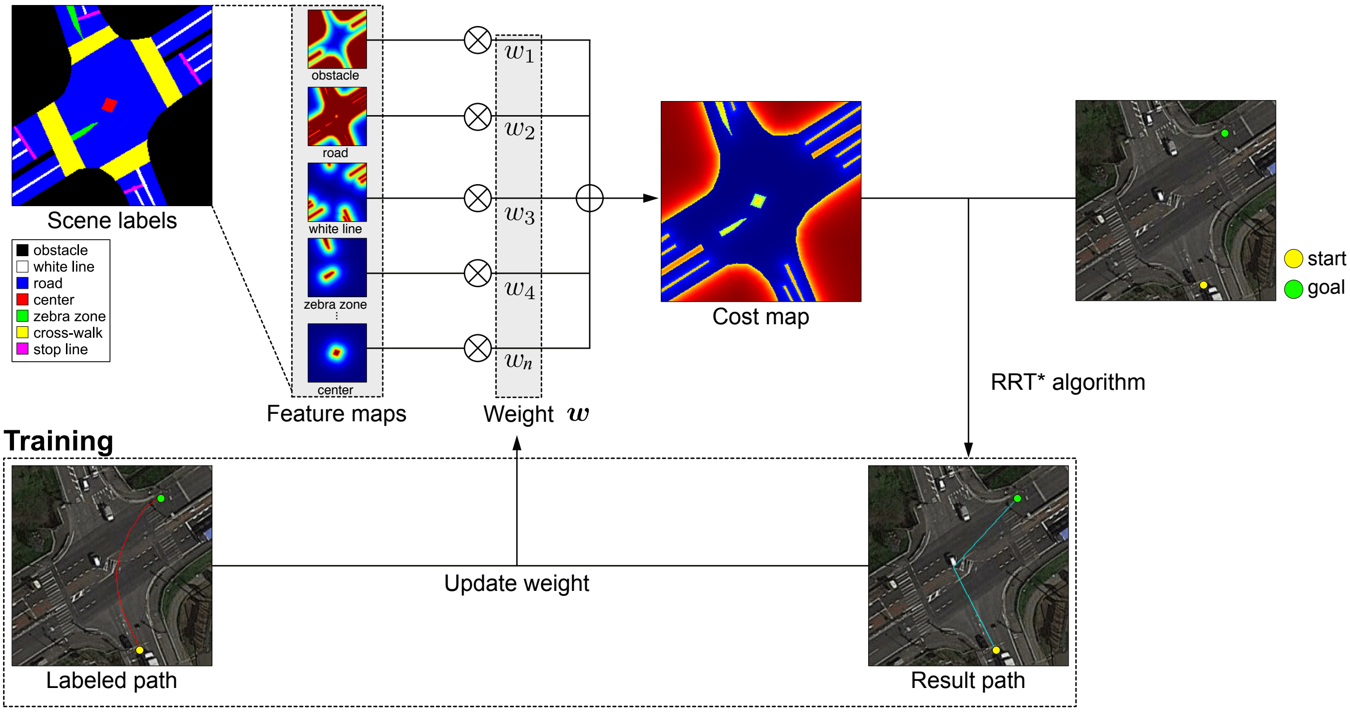

Fig. 1. Process flow of the proposed method

[11] proposed a CNN-based approach for detecting lane markings. They developed an encoder-decoder framework based on a fully convolutional network (FCN) [14], and created a benchmark dataset for lane-marking detection. Kampffeyer *et al.* [12] also proposed a CNN-based approach estimating the regions of each semantic object such as roads, cars, or buildings. They introduced a novel analysis of segmentation results which uses an uncertainty model. Many studies estimate object's categories and positions in aerial and/or satellite images. We estimate positions of vehicles that should drive with acquired semantic scene labels from a CNN or manual annotations.

B. Path planning

Path planning is an extensively studied problem in the robotics, and many approaches have been proposed. One of the conventional approach is rapidly-exploring random trees (RRT) [15], which enables us a feasible path to find quickly by randomly exploring paths from a start position to a destination. Optimal RRT (RRT*) [16], [17] is proposed finding an optimal path based on RRT. This approach introduces a rewire process for minimizing cost between nodes extend to the exploration using RRT algorithm. Therefore, an optimal path can be determined with enough sampling, and the shortest paths to destinations can be created.

Scene context-aware rapidly-exploring random tree (SC-RRT*) [18], which takes scene context into account and observed paths was proposed. Scene context means the surrounding environment such as roads, sidewalks, and buildings. With SC-RRT*, a weight vector $\boldsymbol{w}$ is trained to lower the cost of pedestrian paths by using scene context and these paths. It, then, creates an optimal cost map using a acquired weight obtained by learning and creates optimal paths by RRT* [16], [17]. However, it is focused attention on pedestrian moving paths in scenes. Therefore, such paths are taken the cost into account creating short moving paths to avoid defined obstacles. On the other hand, our approach automatically creates a moving path on a digital map for autonomous driving, and defines an optimal cost function for vehicles.

III. PROPOSED METHOD

SC-RRT* defines a cost function to create a pedestrian path. This cost function is for the path will be a minimum cost. However, our approach is for vehicles, then need to create swelling curving paths depending on the shape of intersections or traffic rules. Therefore, we propose a path-planning approach defined a cost function for vehicles.

Fig. 1 gives an overview of the proposed method. First, we create feature maps representing each semantic label of a scene. These are made up of each semantic label, which means high cost (red) is each label and others are low cost (blue). We, then, define a weight vector $\boldsymbol{w} = (w_1, \dots, w_n)^{\mathrm{T}}$, where n is the number of created feature maps from the scene labels. Next, we compute a cost map from feature maps and weight vectors. After calculating the cost map, we estimate a path being a minimum cost between set the start and goal positions. To efficiently estimate the optimal path, we use RRT* [16], [17].

A. Cost function

In our path-planning problem, the target object is a vehicle on maps. In the case of going straight through an intersection, conventional approaches (e.g., RRT* or SC-RRT*) would be sufficient to generate a path. However, in other cases (e.g., turning rights), the path should take a roundabout route. Therefore, we focused attention on a path for turning right and use our method to define a cost function for vehicles. Let $X \in \mathcal{R}^2$ is a scene and $\boldsymbol{x} = (x_1, \dots, x_T)$ is a path, where $x_n \in X$ is a coordinate. The cost function is defined by

$$\begin{aligned} c_p(\boldsymbol{x}, \boldsymbol{w}) = & \sum_{t=1}^{T} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{f}(x_t) \\ & + \theta \sum_{t=1}^{T-1} \|x_t - x_{t+1}\|_2 \\ & + c\alpha \sum_{t=1}^{T} \|x_t - x_{center}\|_2 . \end{aligned} \tag{1}$$

The first term computes a cost by accumulating the cost acquired position at x_t. This term is derived from a feature vector representing the acquired scene context $\boldsymbol{f}(x_t)$ at a location x_t and weight vector $\boldsymbol{w}$. The second term regularizes the path length, where θ is a scale parameter. This term controls the smoothness and shortness of a path. The third term is a regularizer for a roundabout route, where c is a binary parameter and α is a scale parameter. This binary parameter c is set 1 for right turn or 0 for others. We calculate the distance between x_t and the center of intersection x_{center}. For turning right, our approach creates a path close to the center of the intersection. This term is only supporting creation in case of the turning right.

B. Algorithm

The algorithm of the proposed path-planning approach is shown in Algorithm 1. This is based on RRT* [16], [17].

Given an initial state x_{init}, SC-RRT* extend the trees $T = (V, E)$ with N iterations, where V is a set of nodes, and E is a set of edges. First, we set x_{init} as the root node of T. The random state x_{rand} is sampled in X (`Sample`) and the nearest node $x_{nearest}$ of T from x_{rand} is selected (`Nearest`). Then, the `Steer` process creates a point x_{new}. A `Steer` determines a node by extending in a straight line to x_{rand} from $x_{nearest}$ with the length of η. Second, if the path from $x_{nearest}$ to x_{new} does not interfere with X_{obs} (`ObstacleFree`), node x_{new} and edges $(x_{nearest}, x_{new})$ are added into each V and E. Third, a set of near nodes X_{near} is selected by the `Near` procedure. The radius r to select X_{near} is defined as follows:

$$r = \gamma \left(\frac{\log |V|}{|V|} \right)^{-1/d}, \tag{2}$$

where $|V|$ is the number of nodes in T, d is the dimension of the space, and γ is constant. Then, x_{new} is connected with a node x_{parent}, which minimizes the accumulated cost $\texttt{Cost}(x_{new})$. To compute $\texttt{Cost}(x_{new})$, we use the defined cost function shown in Eq. (1). RRT* also extends the connection from x_{new} to other nodes in $x' \in X_{near} \setminus \{x_{parent}\}$ if the cost of the path from x_{init} to x' passing through x_{new} becomes smaller, which is called *rewire*. Because x_{new} improves the local connection within r, sufficient iteration provides an approximately optimal path.

Algorithm 1 Algorithm of proposed approach

$V \leftarrow \{x_{\mathrm{init}}\}; E \leftarrow \emptyset;$
for $n = 0$ to N **do**
  $T \leftarrow (V, E);$
  $x_{rand} \leftarrow \texttt{Sample}(n);$
  $x_{nearest} \leftarrow \texttt{Nearest}(T, x_{rand});$
  $x_{new} \leftarrow \texttt{Steer}(x_{nearest}, x_{rand}, \eta);$
  if $\texttt{ObstacleFree}(x_{nearest}, x_{new})$ **then**
    $V \leftarrow V \cup x_{new};$
    $x_{parent} \leftarrow x_{nearest}$
    $X_{near} \leftarrow \texttt{Near}(T, x_{new}, |V|)$
    for all $x_{near} \in X_{near}$ **do**
      if $\texttt{ObstacleFree}(x_{near}, x_{new})$ **then**
        $c' \leftarrow \texttt{Cost}(x_{near}) + c_p(x_{new}, x_{near})$
        if $c' < \texttt{Cost}(x_{new})$ **then**
          $x_{parent} \leftarrow x_{near}$
        end if
      end if
    end for
    $E \leftarrow E \cup \{x_{nearest}, x_{new}\};$
    for all $x' \in X_{near} \setminus \{x_{parent}\}$ **do**
      if $\texttt{ObstacleFree}(x_{new}, x')$ and $\texttt{Cost}(x') > \texttt{Cost}(x') + c_p(x_{new}, x')$ **then**
        $E \leftarrow E \setminus \{(\texttt{Parent}(x'), x')\}$
        $E \leftarrow E \cup \{(x_{new}, x')\}$
      end if
    end for
  end if
end for
return $T = (V, E);$

C. Interpolation by clothoid curve

we introduce the regularizer with respect to path length because we need to create smoother path considering vehicle kinematics. Our method explores paths by using the RRT*-based algorithm. It, however, creates an unsmooth path. To make a smoother path, we further interpolate the acquired path by applying clothoid curve which is used for many kinds of road shapes. Specifically, we interpolate the start to middle of acquired path by using a curve fitting method [19].

D. Learning weight vector $\boldsymbol{w}$

During the learning phase, we estimate an optimal weight vector $\hat{\boldsymbol{w}}$ from training data. We apply the feature-matching approach as standard in a previous study [18]. In feature matching, we update $\boldsymbol{w}$ which is the acquired features from the proposed approach with the current weight, corresponds to the features of the training samples.

A average of features computes $\bar{\boldsymbol{f}}$ calculated from K paths is defined by

$$\bar{\boldsymbol{f}} = \frac{1}{K} \sum_{k}^{K} \boldsymbol{f}(\boldsymbol{x}_k). \tag{3}$$

A gradient of the cost function is defined as

$$\nabla c_{\boldsymbol{w}} = \bar{\boldsymbol{f}} - \bar{\boldsymbol{f}}_{\boldsymbol{w}}, \tag{4}$$

where $\bar{\boldsymbol{f}}$ is the average of aomputed features, and $\bar{\boldsymbol{f}}_w$ is also the average of expected feature count. The $\bar{\boldsymbol{f}}$ is calculated from training samples, and $\bar{\boldsymbol{f}}_w$ is from created samples using the proposed approach with the current $\boldsymbol{w}$. That, then, updates the parameters by using

$$\boldsymbol{w} \leftarrow \boldsymbol{w} e^{(-\lambda \nabla c_{\boldsymbol{w}})}, \tag{5}$$

where λ is a step size.

To calculate the expected features for the current $\boldsymbol{w}$, we apply the Algorithm. 1 and create a path. We, then, acquire proper weights $\hat{\boldsymbol{w}}$ by repeating the update (Eq. (5)) until $\boldsymbol{w}$ does not change, which is defined by

$$\|\boldsymbol{w}_k - \boldsymbol{w}_{k-1}\| = \epsilon. \tag{6}$$

IV. EXPERIMENTAL RESULTS

A. Dataset

We create a moving-path dataset of vehicles at intersections for learning samples and quantitative evaluation. The dataset includes intersection images, scene labels, paths, and coordinates of intersection centers.

The dataset contains 100 aerial images at intersections collected using Google Static Maps API. Each image size is 640×640 pixels. We, then, crop these images to include only the intersection area without changing the aspect ratio. Therefore, each image size differe depending on the scene. We annotate scene labels is classified seven categories, such as roads and white lines, which would affect vehicle driving.

We collected path information of vehicles driving in those scenes. We collected a total of 1,417 moving-path samples; 472 straights, 471 turning lefts and 474 turning rights. Furthermore, we collected the center coordinates center of intersections. Examples from the dataset are shown in Fig. 2.

B. Experiment settings

We evaluate the difference of our approach's accuracy by changing its parameters (i.e., θ and α). Each parameter is changed by 0.1. Because the number of possible patterns of parameters is a lot, we estimate optimal parameters in the following steps:

1) We evaluate the error of conventional SC-RRT* and the proposed method while changing θ. At that time, α of the proposed method is fixed to 1.0.
2) Then, we seek appropriate α of the proposed method. θ is fixed to the best acquired value by the first step experiment.

We use SC-RRT* [18] as the conventional method, whose cost function is defined with the first and second terms in Eq. (1).

As parameters of RRT* in proposed and conventional methods, we set the tree length to 10 pixels, the number of random samples to 1,000, and the goal sampling rate to 5 %.

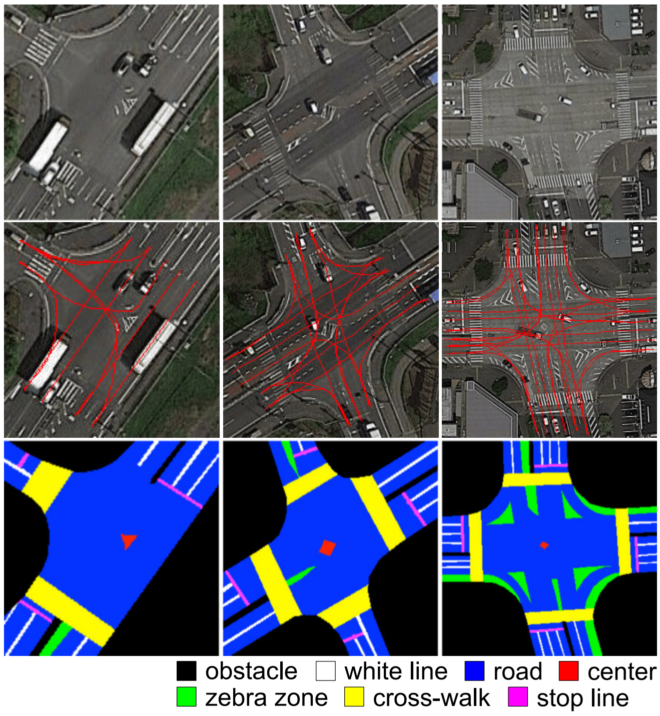

Fig. 2. Examples of created dataset. Top: collected intersection images. Middle: collected moving paths. Bottom: annotated scene labels in each scene.

We experiment this quantitative evaluation by applying a modified Hausdorff distance (MHD) [20]. Given a ground truth of moving path $A = \{a_1, \ldots, a_{N_a}\}$ and created path $B = \{b_1, \ldots, b_{N_b}\}$, the MHD is formulated as

$$d(A, B) = \frac{1}{N_a} \sum_{a \in A} d(a, B) \tag{7}$$

$$d(B, A) = \frac{1}{N_b} \sum_{b \in B} d(b, A) \tag{8}$$

$$\mathcal{D}(A, B) = \max(d(A, B), d(B, A)), \tag{9}$$

where $d(a, B) = \min_{b \in B} \|a - b\|$. The MHD is used to measure the degree of similarity of an object's form. We quantitatively evaluate the degree of similarity between a ground truth and a created path regarded as a quadratic curve.

C. Experimental results

Table I shows the MHD of SC-RRT* and the proposed approach, where α was fixed to 1.0. SC-RRT* achieved the best result from $\theta = 1.0$. On the other hand, our approach is the most part better the MHD than SC-RRT* with $\theta = 0.5$-0.8. These results indicate the proposed approach is able to create a more proper moving path for vehicles. Therefore, we fixed $\theta = 0.8$ and evaluate the proposed approach with different α to decide the best parameter patterns.

Table II shows the MHD of the proposed approach with fixed $\theta = 0.8$ changing by $\alpha = 0.1$. These results indicate the result in the case of $\theta = 0.8$ and $\alpha = 0.1$ is the lowest error. It indicates this parameter pattern is the most proper.

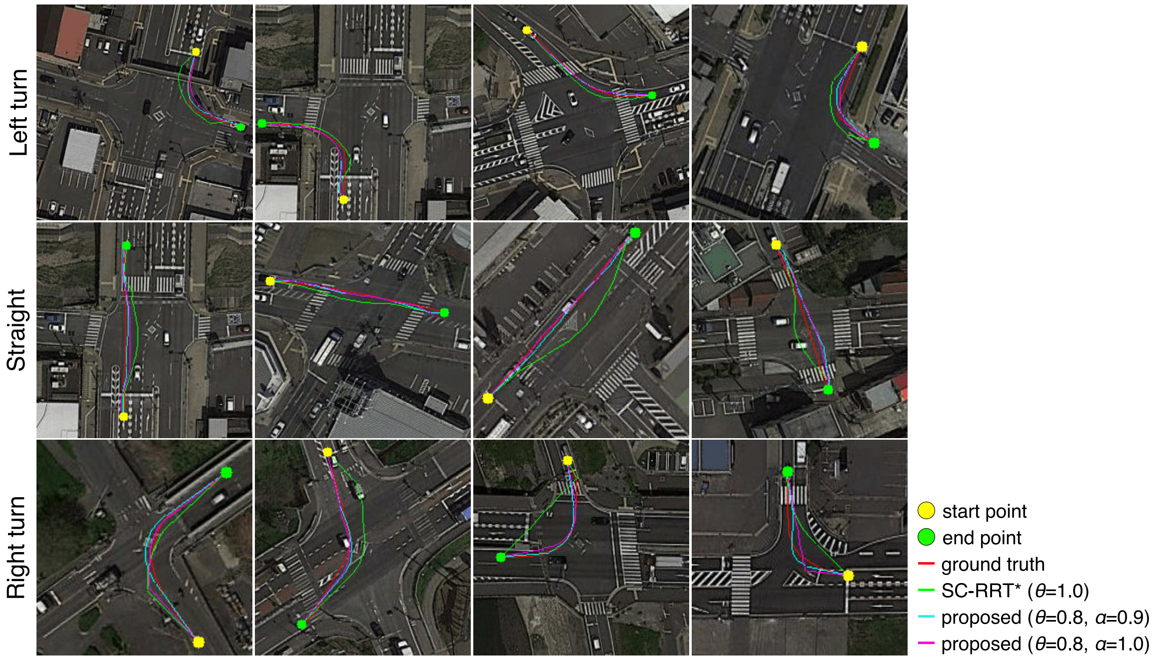

Fig. 3. Examples of generated paths

TABLE I
MHD OF SC-RRT* AND PROPOSED APPROACH (UNIT: PIXELS)

approach	θ	α	MHD
SC-RRT* [18]	0.5	–	6.275
	0.6	–	6.044
	0.7	–	6.100
	0.8	–	5.962
	0.9	–	5.843
	1.0	–	5.695
Proposed	0.5	1.0	4.273
	0.6	1.0	6.930
	0.7	1.0	4.076
	0.8	1.0	**3.987**
	0.9	1.0	5.887
	1.0	1.0	5.883

TABLE II
MHD VALUES WITH FIXED $\theta = 0.8$ AND CHANGING α (UNIT: PIXEL)

approach	θ	α	MHD
Proposed	0.8	0.1	4.882
		0.2	4.793
		0.3	4.499
		0.4	4.515
		0.5	4.365
		0.6	4.328
		0.7	4.160
		0.8	4.250
		0.9	**3.967**
		1.0	3.987

Table III shows MHD focus attention on the different courses which is right turns, left turns and going straights. The MHD of straights and left turns are smaller than right turns. The created paths for straights and left turns are shorter to avoid collisions between a vehicle and obstacles. On the other hand, moving paths for right turns needed to move toward the center of the interaction, then move to turn the right. These paths are comparatively difficult problems because it just create the shortest paths such as for straights or left turns.

TABLE III
MHD ON THE RIGHT OR LEFT TURNS OR GOING STRAIGHTS (UNIT: PIXELS (METERS))

	straight	left turn	right turn
SC-RRT* ($\theta = 1.0$) [18]	4.268 (1.067)	3.119 (0.780)	9.744 (2.436)
Proposed ($\theta = 0.8, \alpha = 0.9$)	1.901 (0.475)	1.074 (0.269)	4.177 (1.044)

In our dataset, 1 pixel in the image plane is corresponding to about 0.25 meters. From Tab. III, our method decreases error by over 1 meter in case of right turn.

D. Qualitative evaluations

The results of created paths are shown in Fig. 3. Each approach in left turns is able to create similarity paths. In straights and right turns, SC-RRT* create square or improper paths from start to end point. But our approach is able to take traffic rules into, and also account the center of intersections. These results show our approach creates proper moving paths at interactions for vehicles.

The RRT-based approach has features in which the resultant paths are not quite smooth because it constantly extends nodes. In fact, the moving paths of vehicles should be more smooth like the real driving paths. Therefore, we need to consider creating more smoothly moving paths.

Examples of failure results are shown in Fig. 4. The top-left of the figure shows the straight course which should be just created straight paths, but, the created paths are not

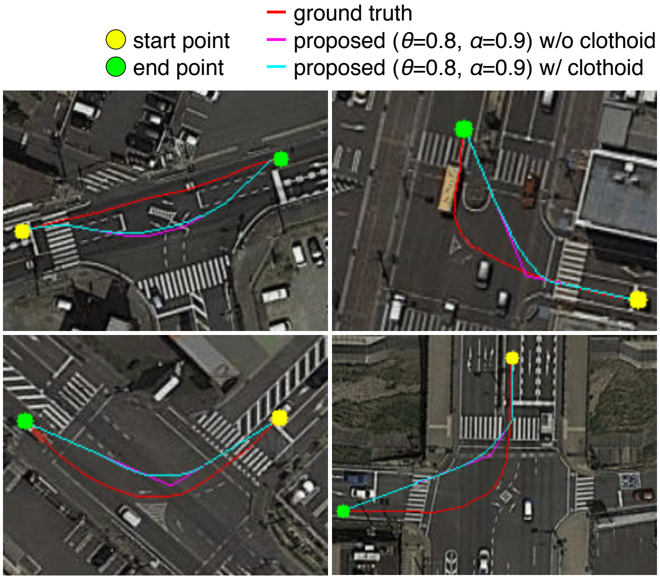

Fig. 4. Examples of failure results

straights. That is why the clothoid fitting does not interpolate sufficiently. The top-right and bottom of figures show the results in right turn. In these cases, we need to create paths that go straight toward the center of intersection and then turn to right. However, these results just start to turn right before entering the intersection. The third term of our cost function does not work effectively. In these results, the clothoid fitting also does not interpolate in these results. Therefore, we need to set more appropriate scale parameters or acquire optimal parameter by training.

V. CONCLUSIONS

We propose the automatic path-planning approach on digital maps for autonomous driving. The proposed approach creates cost maps from feature maps by scene labels and weight vectors, and creates paths being a minimum cost from a start position to a destination. We estimate optimal weight vectors from learning data by calculating optimal cost maps. We also define the cost function for moving vehicles at interactions.

We create a dataset to evaluate the performance of the proposed approach by using a dataset we created, and our proposed approach is able to create paths suitable for vehicle movement at intersections. Because this is the first attempt of creating optimal vehicle paths for digital maps, our path-planning approach is a rather simplicity. But it will be further improved based on the following aspects. Our approach is estimating optimal values with parameters, and the other is considering vehicle kinematics and dynamics to create smooth paths. Simultaneously optimizing every paths at an intersection is also necessary to consider interactions between vehicle movements.

ACKNOWLEDGMENT

This work is supported in part by JSPS KAKENHI grant number JP16H06540.

REFERENCES

[1] P. Fischer, S. M. Azimi, R. Roschlaub, and T. Krauß, “Towards hd maps from aerial imagery: Robust lane marking segmentation using country-scale imagery,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 12, p. 458, 2018.

[2] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik, “Multipath detection with 3d digital maps for robust multi-constellation gnss/ins vehicle localization in urban areas,” in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 184–190.

[3] Z. Tao and P. Bonnifait, “Tightly coupling gps with lane markings for autonomous vehicle navigation,” in *International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 439–444.

[4] ——, “Road invariant extended kalman filter for an enhanced estimation of gps errors using lane markings,” in *International Conference on Intelligent Robots and Systems*, 2015, pp. 3119–3124.

[5] S. Bauer, Y. Alkhorshid, and G. Wanielik, “Using high-definition maps for precise urban vehicle localization,” in *International Conference on Intelligent Transportation Systems*, 2016, pp. 492–497.

[6] M. Schreiber, C. Knöppel, and U. Franke, “Laneloc: Lane marking based localization using highly accurate maps,” in *Intelligent Vehicles Symposium*, 2013, pp. 449–454.

[7] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, *et al.*, “Making bertha drive—an autonomous journey on a historic route,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[8] M. Aeberhard, S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen, “Experience, results and lessons learned from automated driving on germany’s highways,” *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 42–57, 2015.

[9] O. Tournaire, N. Paparoditis, and F. Lafarge, “Rectangular road marking detection with marked point processes,” in *Conference on Photogrammetric Image Analysis*, vol. 3, 2007, pp. 149–154.

[10] V. Mnih and G. E. Hinton, “Learning to detect roads in high-resolution aerial images,” in *European Conference on Computer Vision*, 2010, pp. 210–223.

[11] S. M. Azimi, P. Fischer, M. Körner, and P. Reinartz, “Aerial lanenet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–19, 2018.

[12] M. Kampffmeyer, A. Salberg, and R. Jenssen, “Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks,” in *Computer Vision and Pattern Recognition Workshops*, 2016, pp. 680–688.

[13] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “High-resolution aerial image labeling with convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7092–7103, 2017.

[14] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[15] S. M. LaValle and J. James J. Kuffner, “Randomized Kinodynamic Planning,” *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[16] S. Karaman and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning,” in *Robotics: Science and Systems VI*, vol. 104, 2010.

[17] ——, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[18] T. Hirakawa, T. Yamashita, and H. Fujiyoshi, “Scene Context-aware Rapidly-exploring Random Trees for Global Path Planning,” in *International Workshop on Behavior analysis and Recognition for knowledge Discovery*, 2019.

[19] E. Bertolazzi and M. Frego, “G1 fitting with clothoids,” *Mathematical Methods in the Applied Sciences*, vol. 38, no. 5, pp. 881–897, 2015.

[20] M. P. Dubuisson and A. K. Jain, “A modified Hausdorff distance for object matching,” in *International Conference on Pattern Recognition*, vol. 1, 1994, pp. 566–568.