Scene Context-aware Rapidly-exploring Random Trees for Global Path Planning

Tsubasa Hirakawa
Chubu University
Kasugai, Japan
hirakawa@mprg.cs.chubu.ac.jp

Takayoshi Yamashita
Chubu University
Kasugai, Japan
takayoshi@cs.chubu.ac.jp

Hironobu Fujiyoshi
Chubu University
Kasugai, Japan
fujiyoshi@cs.chubu.ac.jp

***Abstract*—This paper introduces a global path planning method for autonomous systems. Global path planning finds a feasible and collision-free path in an environment in which various kinds of regions and objects exist. However, the most planning methods use information such as collision-free space and obstacles in the environment. Interactions at each region (e.g., sidewalk and pavement) would be different. In this paper, we propose a method for global path planning taking semantic scene context into account. In contrast to conventional path planning methods which use collision-free and obstacle regions, the proposed method represents an environment as a cost map. The cost map is estimated from demonstrated human behaviors and feature maps derived from semantic scene context. To find a path on the cost map, we define a path cost and leverage an optimal rapidly-exploring random tree (RRT*) algorithm. We evaluate the proposed method regarding accuracy and computational efficiency with two public datasets and our contributed dataset. Experimental results show that our method successfully reproduces paths like human behaviors in short computational time.**

***Index Terms*—path planning, rapidly-exploring random tree, semantic scene context**

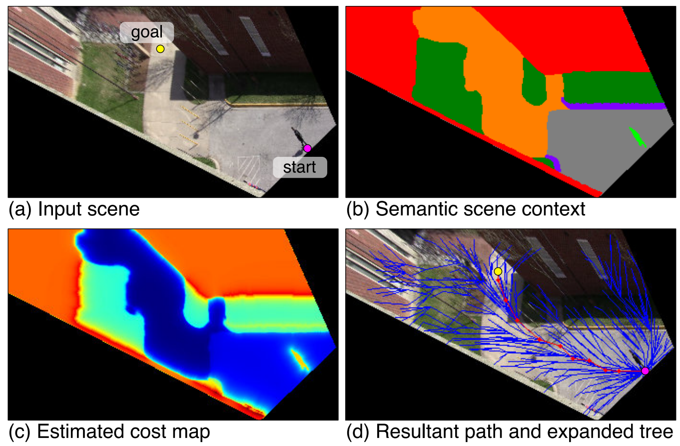


Fig. 1. Our approach extends RRT* algorithm. Given (a) a scene and (b) the semantic scene context, (c) we estimates cost of each location from demonstrated path. (d) A path is produced by randomly extending tree nodes. Our method enables us to find a path of behaving like a human by considering semantic scene context.

I. Introduction

Humans have a latent ability to understand their surrounding environment and to decide upon future actions appropriately and immediately. For instance, when a pedestrian walks in a public place, an office, or a house, they take a look around and understand the location of obstacles and other pedestrians. We then find a smooth route that will enable them to reach their destination without encountering collisions. Those human abilities are much helpful to develop autonomous systems [1], [2]. In autonomous systems (e.g., human support robots) that work with their surrounding person cooperatively, it would be ideal to move without disturbing actions of pedestrians and avoiding collisions.

Path planning, finding a feasible path in an environment, is a fundamental problem in robotics. Many planning approaches have been focused on avoiding collision with obstacles and on interactions between pedestrians [3]–[7]. These approaches leverage primitive information existing in an environment. One is collision-free regions where autonomous systems can move. The other is obstacle regions such as a wall, table, and the surrounding pedestrians. However, those regions can be categorized into several classes based on the semantics. For instance, in outdoor scenes, collision-free regions includes sidewalk and pavement. Based on this semantics, pedestrians move on a sidewalk and avoid walking on pavement, but they occasionally take a shortcut across a pavement. In indoor scenes such as an office or a house, there a lot of furniture (e.g., table and chair). Although pedestrian naturally walks avoiding furniture, the actions of autonomous systems obtained by the above planning methods are different from those of human. This difference would lead to confusion between human-robot interactions. Therefore, it would be better for autonomous systems to behave like a human.

We aim to develop a global path planning method that finding a path behaving like a human. In this paper, we propose a global path planning method introducing semantic scene context in an environment (see Fig. 1). The proposed method uses feature maps derived from semantic scene context instead of collision-free and obstacle regions. We estimate a cost map from the feature maps and observed human behaviors (i.e., tracked paths of pedestrians) by feature matching approach. Based on a cost map, we find a path by using an optimal rapidly-exploring random tree (RRT*) [8], [9] that guarantees path optimality.

Our contribution is two-fold. 1) To the best of our knowledge, we first introduce semantic scene context into path plan-

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

ning problem in robotics. The existing path planning methods use only collision-free and obstacle regions. Meanwhile, our approach considers interactions between each scene context. 2) We contribute a new path planning dataset that collects human behaviors in an indoor environment. Our dataset consists of approximately 200 walking paths. Moreover, this dataset contains the detailed object informations in the scene.

II. Related work

Path planning is a widely investigated problem in the field of robotics and can be further categorized into two problems. One is *local path planning*, which finds a collision-free path by considering dynamic environmental changes or interactions between the surrounding pedestrians. Social force model is used as a model-based approach [3]–[5] to consider interactions among pedestrians. As a learning-based approach, Chen et al. [6], [7] proposed a planning method by using reinforcement learning (RL). They model deep reinforcement learning architecture and learn agent's policy to interact with dynamic environmental changes. Thus, these works try to find a relatively short-distance path when interactions happen, while our approach deals with long-term paths.

The other is *global path planning*, which finds long-distance paths from start to distant goal. Many approaches have been proposed over several decades such as artificial potential field (APF) [10], cell decomposition [11], and probabilistic road map (PRM) [12], [13]. These approaches find merely a feasible and collision-free path, while our method finds paths considering semantic scene context.

One of the typical planning approaches is inverse optimal control (IOC) or inverse reinforcement learning (IRL) framework. IRL estimates a reward function from demonstrated human behaviors (i.e., expert actions) by feature matching approach. As features, obstacles and paths of the surrounding pedestrians are used. In the field of computer vision, Kitani et al. [14] proposed a path prediction method by IRL framework. They predict a sequence of actions from stat to the goal using Markov decision process (MDP) and a static physical environment. Our method involves problem settings similar to [14] because they also predict a path from state to goal in a static scene environment. This method predicts future locations as a probability distribution, while our method finds a deterministic path. Moreover, these IRL-based approaches are applied in relatively small state space because of the high computational cost. Our RRT*-based method can find a path in short computational time.

Another approach is rapidly-exploring random tree (RRT) [15]–[17], which is one of the most popular approaches in path planning and is widely used because of its lower computational cost. RRT explores a path from start to the goal by randomly extending tree nodes. The random sampling regardless of path optimality and kinematics makes it possible to provide a path in shorter computational time. An improved RRT algorithm, RRT* [8], [9], has also been proposed, which mainly focus on guaranteeing the optimality of explored path. Other improved RRT-based algorithms have also been proposed [18]–[21]. Pérez-Higueras et al. [22] proposed RRT-based IRL path planning method (RRT*-IRL). They estimate an optimal reward function by feature matching approach as with IRL-based method. These RRT-based methods use information about obstacles and pedestrian locations as feature maps, while our approaches leverage semantic scene context. Moreover, in [22], they initially manually modeled reward function and generate expert's behaviors by operating robots based on the reward function. They reproduced the reward function from the collected behaviors. In contrast, our approach uses observed human walking path as expert's behaviors. Therefore, our approach provides a smoother path.

III. Method

We formulate the proposed method. As mentioned previously, our path planning method is based on RRT* algorithm. The basic RRT* explores tree nodes in a scene and finds a feasible path by using obstacle information. In contrast, our method defines a cost map taken for passing each state. The cost map is defined by linear combination with feature maps derived from semantic scene context and a weight vector. Accumulated cost on each location over a path is used for the cost of the path. The proposed method learns an optimal cost map (i.e., an optimal weight vector) to reproduce observed human behaviors by feature matching approach. To avoid confusion, we will refer to the cost at each state in a scene as *state cost* and the accumulated cost for a path as *path cost*.

A. Problem formulation

Let $X \in \mathcal{R}^2$ be a state space where we find a path. We define a path $\boldsymbol{x} = (x_1, x_2, \dots, x_T)$ as a sequence of states in X, and $\boldsymbol{f}(x) = (f_1(x), f_2(x), \dots, f_m)$ be a feature vector at x. A regional cost c_r at x is defined as

$$c_r(x; \boldsymbol{w}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{f}(x), \tag{1}$$

where $\boldsymbol{w}$ is a weight vector. Given a path $\boldsymbol{x}$, we define a path cost $c_p(\boldsymbol{x}, \boldsymbol{w})$ as an accumulation of regional costs at each state in $\boldsymbol{x}$ as follows:

$$\begin{aligned} c_p(\boldsymbol{x}, \boldsymbol{w}) &= \sum_{t=1}^{T} c_r(x_t, \boldsymbol{w}) + \theta \sum_{t=1}^{T-1} \|x_t, x_{t+1}\|_2 \\ &= \sum_{t=1}^{T} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{f}(x_t) + \theta \sum_{t=1}^{T-1} \|x_t, x_{t+1}\|_2. \end{aligned} \tag{2}$$

The first term represents the cumulative cost at each state over the path, which represents feasibility of the path, that is, a path having a lower cost would be easy to move, and that a having higher cost would be difficult or impossible to move. The second term is a regularizer for the path length of $\boldsymbol{x}$, where θ is a scale parameter. If we only use the first term as the cost of the path, there is no constraint on the moving distance, and we could find a detour path (see Section IV). In [22], they define a similar cost function to our formulation, and a constraint for path lengths is embedded into the feature vector. However, such formulation yet generates a detour and unsmoothed path.

We, therefore, added the second term to find a path that keeps short length while minimizing the accumulated state cost.

Given feature vector and a weight vector, we can find a path by RRT* algorithm as shown in Algorithm 1. Initially, 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`). `Steer` returns a node x_{new}, which is decided by extending in the direction of x_{rand} from $x_{nearest}$ with the length of η. If the path from $x_{nearest}$ to x_{new} does not interfere with obstacles X_{obs} (`ObstacleFree`), x_{new} is added into V. Then, a set of near nodes X_{near} is selected (`Near`). The radius r to select X_{near} is defined as follows:

$$r = \gamma \left(\frac{\log |V|}{|V|} \right)^{-1/d}, \tag{3}$$

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 makes the path cost c' minimum. For the `Cost` operation, we compute a path cost from the root note x_{init} by using Eq. (2). Furthermore, RRT* extends the connection from x_{new} to other nodes in $x' \in X_{near} \backslash \{x_{parent}\}$ if the cost of the path from x_{init} to x' passed through x_{new} becomes smaller, which is called as *rewire*. After N iterations of RRT* are done, we find a node that makes the path cost from start to goal and obtains a path by connecting unique parent nodes to the root node.

Algorithm 1 SC-RRT(x_{init})

$V \leftarrow \{x_{\text{init}}\}; E \leftarrow \emptyset;$
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 $T = (V, E);$

B. Learning from demonstrated human behaviors

Given a feature vector and demonstrated human behaviors, we estimate an optimal weight vector $\hat{\boldsymbol{w}}$. For this optimization problem, we apply feature matching approach, that is, we update the cost $\boldsymbol{w}$ in which the empirical feature count of demonstrated paths and the expected feature count of planned paths with current $\boldsymbol{w}$ are equalized by using an exponentiated gradient descent. The gradient of path cost function $\nabla c_{\boldsymbol{w}}$ is defined as

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

where $\bar{\boldsymbol{f}}$ is the mean empirical feature count and $\bar{\boldsymbol{f}}_w$ is the mean expected feature count. Then, we update the parameters as

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

where λ is the learning rate (or step size). We repeat the update until $\boldsymbol{w}$ converges with the termination criterion of

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

IV. Results

We evaluate our method from the following points: accuracy of the planned paths, 2) convergence property over the different number of iterations, and 3) computational time.

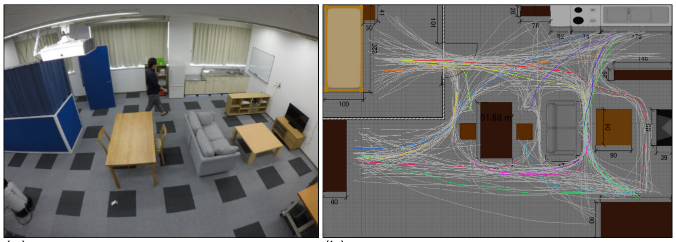
A. Datasets

For quantitative evaluation, we used three datasets collecting pedestrian walking paths.

VIRAT Video Dataset [23]: VIRAT Video Dataset is a surveillance camera video dataset collected in 11 outdoor scenes including the locations of any objects (e.g., pedestrians, cars, buildings, and carts). We collected 168 pedestrian walking paths samples that are available for our experiment. To estimate semantic scene context from a video frame, we labeled 180 video frames with the following ten classes: pavement, sidewalk, curb, building, person, car, grass, tree, gravel, and fence. These annotated scene labels were used to train a fully convolutional network (FCN) [24], [25] and the trained network was used to estimate semantic scene labels of video frames of the path samples.

Stanford Drone Dataset (SDD) [26]: SDD consists of large-scale aerial videos taken at Stanford University. A large number of objects exist in a scene, and SDD has more than 20,000 examples in total. SDD also includes annotations of object attributes such as pedestrian, car, and bike. Among them, we select only paths of pedestrians, and approximately 3,000 pedestrian path samples are used for our evaluation. As semantic scene context, we used a manually annotated semantic scene labels: pavement, sidewalk, grass, tree, building, and roundabout. We resize the dataset to 1/4 size.

Living Space Path Planning Dataset: For further evaluation, we built a new dataset for path planning. In this dataset, we collected approximately 200 walking paths in a living



(a) (b)

Fig. 2. An example of Living Space Path Planning Dataset. (a) A snapshot that a pedestrian is walking. (b) Annotated path. Different colors indicate different pedestrian paths.

space as shown in Fig. 2. Because the area of this room and the location and size of furniture items such as tables and chairs are known, we can evaluate the planned results under the complete observable environment. The dataset is going to be publicly available.

For the following experimental results, we used 80% of the data as training samples, and the rest is used for tests.

B. Metrics and Baselines

As an evaluation metrics, we adopted modified Hausdorff distance (MHD) [27].

We compared our method with the following baselines:

- **Markov decision process (MDP)** [14]: This approach outputs a probability distribution as a prediction results while our method provides a deterministic path. For a fair evaluation, we estimate a deterministic path by selecting actions having the highest probability at each state.
- **RRT*** [8]: For VIRAT Video Dataset and SDD, obstacles are not given. Therefore, we find paths without any obstacle regions. For Living Space Path Planning Dataset, since we built the dataset and various information were known (e.g., the area of the space and the size and layout of furniture), we find paths with obstacle regions.
- **RRT*-IRL** [22]: Instead of feature vectors of obstacles and pedestrians, we leverage feature vectors derived from semantic scene context. Moreover, we introduce a feature vector about a distance to goal as with [22].
- **Scene Context-aware RRT* (SC-RRT*)**: The proposed method. We denote our model without the regularization term as SC-RRT*. Our model with regularization term is denoted as SC-RRT*-L2-θ, where θ is a scale parameter of the regularization term. SC-RRT* contains a few hyperparameters, i.e. η, r, and θ. We used a grid search to select η and r. Because θ affects the performance, we confirm the performance by changing the value of θ.

C. Planning accuracy

We show the quantitative evaluation with MHDs and examples of results in Table I and Fig. 3, respectively. Results of RRT*-based methods (i.e., RRT*, SC-RRT*, and SC-RRT*-L2-θ) are obtained by 1,000 times iterations. In the results of MDP, although the results follow the scene context, these paths are produced by taking a straight line to a certain location and turn around repeatedly as shown in Fig. 3(a, e, h). As the mentioned above, obstacles are unknown, and our method estimates obstacle regions from scene context and demonstrated paths. Therefore, RRT* provides roughly straight path on VIRAT and SDD (see Fig. 3(a-f)). In the Living Room Path Planning Dataset, obstacle regions are given and RRT* finds paths without collision with furniture. However, these paths move very close to wall or tables. We can see that SC-RRT* makes detour paths. Because SC-RRT* does not consider the length of a path, it randomly chooses points which have smaller costs. For every dataset, RRT*-IRL provides detour paths. These results are similar to our method without a regularizer for the path length (SC-RRT*). Although RRT*-IRL includes the distance from goal as a feature element, the learned weight for this feature is rather small, and the shortness of the planned path is not ensured. Our method, SC-RRT*-L2-θ, improves SC-RRT* by adding regularizer and outperform these methods. And the accuracies are varied by changing the parameter of the regularizer θ in Eq. (2). Choosing smaller θ (e.g., $\theta = 0.1$) loses the smoothness of the path and makes slightly poor while increasing parameter of the regularizer in Eq. (2), the planning results become better.

TABLE I
MHD ON EACH DATASET

	VIRAT	SDD	Living Space
MDP [14]	11.413	19.047	30.109
RRT* [8]	7.327	14.119	24.676
RRT*-IRL [22]	16.845	31.015	36.119
SC-RRT*	25.220	34.579	33.764
SC-RRT*-L2-0.1	8.126	15.825	29.402
SC-RRT*-L2-0.5	7.708	13.609	**23.193**
SC-RRT*-L2-1.0	**6.176**	**13.099**	24.991

TABLE II
MHD OVER THE DIFFERENT NUMBER OF ITERATIONS ON SC-RRT*-L2-1.0

	VIRAT	SDD	Living Space
0.5k iterations	6.344	13.290	27.142
1.0k iterations	6.176	13.099	24.991
2.0k iterations	6.132	12.981	22.648

D. Convergence property

Table II shows MHDs over the different number of iterations on Living Space Path Planning Dataset. Examples of the planned path over the different number of iterations are shown in Fig. 4. As we can see, even 500 iterations provide the optimal results, and the later iterations provide only a few improvements of the cost and path. Therefore, 1,000 iterations are sufficient to obtain practical results.

E. Computational cost

Because our method is based on a random sampling approach, our approach reaches an optimal solution by iterating exploration of the tree, but there is a trade-off due to increasing computational cost. Herein, we show the relationship between

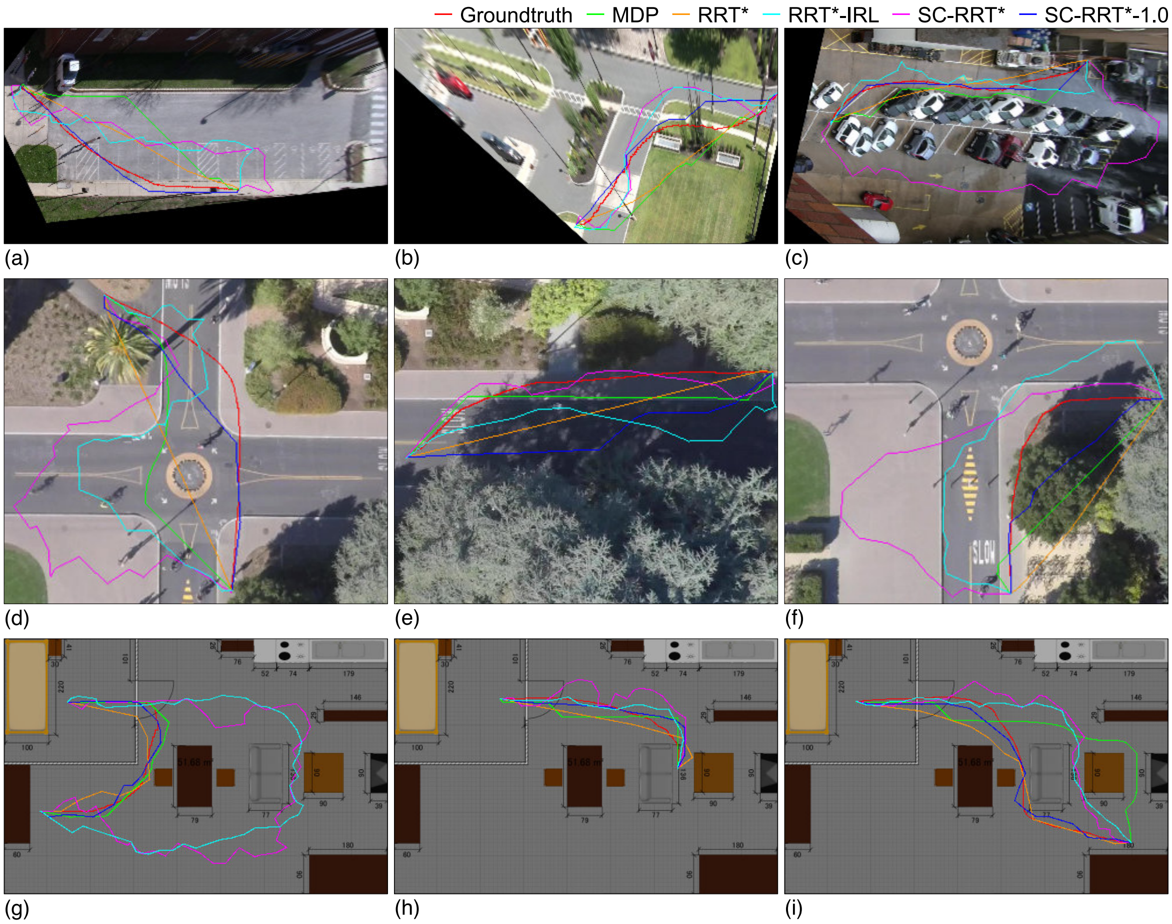


Fig. 3. Examples of the planned paths. From top to bottom row, results on VIRAT Video Dataset (a-c), SDD (d-f), and Living Space Path Planning Dataset (g-i). As a comparative methods, we also show the results of MDP [14], RRT* [8], RRT*-IRL [22], and SC-RRT*.

TABLE III
COMPUTATIONAL TIME ON EACH DATASET

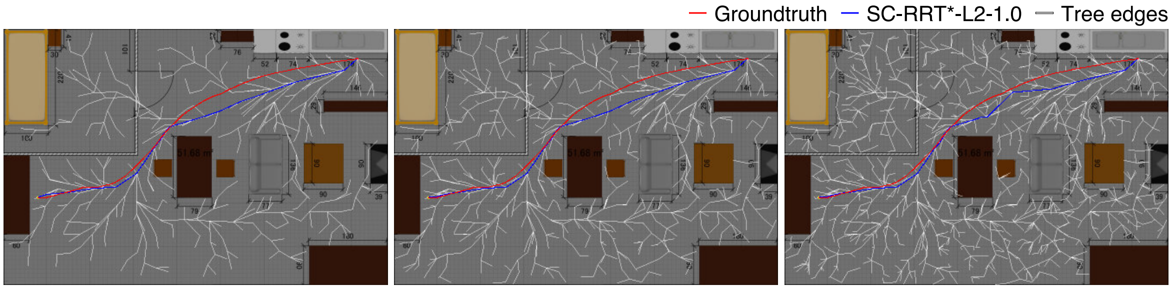
	VIRAT	SDD	Living Space
MDP [14]	491.896	1,144.605	2,894.500
RRT* (1.0k iter.) [8]	5.136	5.317	3.889
SC-RRT*-L2-1.0 (0.5k iter.)	5.512	7.664	8.540
SC-RRT*-L2-1.0 (1.0k iter.)	19.001	28.358	35.344
SC-RRT*-L2-1.0 (2.0k iter.)	74.593	118.664	132.724

the number of iteration and the path optimality. Table III shows the average computational time to find a path on each dataset. We can see that the computational time of MDP approach is much longer than the other methods because the MDP computes actions, i.e., move direction, at every states in a scene. The RRT* finds a path in short computational time, but our method is relatively slower than RRT*. The reason is that our method computes a path cost derived from semantic scene context in addition to Euclidean distance. As shown in Table II, since a planned path almost reaches near optimal ones by 1,000 iterations, our approach can find a path in efficient computational time. Currently, our unoptimized implementation written in Python uses a single thread on an Intel Core i5 (3.1 GHz) processor with 16 GB memory.

V. CONCLUSION

In this paper, we proposed a global path planning method that introduces semantic scene context. The proposed method estimates a cost map from feature vector generated from semantic scene context and observed human behaviors. We find paths on the estimated cost map by RRT*. Because of the property of RRT*, the optimality of the planned path are guaranteed. Experimental results demonstrate that our method outperforms other methods and can work in shorter computational time.

The proposed method can be further improved from the following two aspects. The first aspect is finding a path in a large and high dimensional space. Many path planning studies were applied to only relatively small environments. The main problem would be the higher computational cost. Our method showed computational efficiency, and we believe that it has the potential to be applied in a large and high dimensional space. Therefore, extending our method into a higher dimensional space is one of our future work. The second aspect is considering dynamic environmental changes



(a) 500 iterations
Cost: 112.606, MHD: 18.185

(b) 1,000 iterations
Cost: 112.519, MHD: 18.023

(c) 2,000 iterations
Cost: 111.642, MHD: 17.711

Fig. 4. Results over the different number of iterations. (a-c) The planned path at 500, 1,000, and 2,000. We can see that the number of tree edges increases by repeating extending procedure and the results slightly improve regarding the both of cost and MHD.

such as moving pedestrians. Our approach finds a path considering the static environment. However, autonomous systems would operate in a scene where a lot of pedestrians exists, and it is natural that autonomous systems interact with those environmental changes. Therefore, extending our method for dynamical environmental changes is also our future work.

ACKNOWLEDGMENT

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

REFERENCES

[1] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2616–2625.

[2] A. Vemula, K. Mülling, and J. Oh, "Social Attention: Modeling Attention in Human Crowds," *arXiv preprint arXiv:1710.04689*, 2017.

[3] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[4] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *International Conference on Intelligent Robots and Systems*, 2013, pp. 1688–1694.

[5] G. Ferrer and A. Sanfeliu, "Behavior estimation for a complete framework for human motion prediction in crowded environments," in *International Conference on Robotics and Automation*, 2014, pp. 5940–5945.

[6] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *International Conference on Robotics and Automation*, 2017, pp. 285–292.

[7] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *International Conference on Intelligent Robots and Systems*, 2017, pp. 1343–1350.

[8] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," in *Robotics: Science and Systems*, 2010.

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

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[11] R. A. Brooks and T. Lozano-Pérez, "A subdivision algorithm in configuration space for findpath with rotation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 2, pp. 224–233, 1985.

[12] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[13] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *International Conference on Robotics and Automation*, vol. 1, 1996, pp. 113–120.

[14] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *European Conference on Computer Vision*, 2012, pp. 201–214.

[15] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000, pp. 293–308.

[16] 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.

[17] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[18] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *International Conference on Robotics and Automation*, vol. 2, 2000, pp. 995–1001.

[19] M. Rickert, O. Brock, and A. Knoll, "Balancing exploration and exploitation in motion planning," in *International Conference on Robotics and Automation*, 2008, pp. 2812–2817.

[20] A. H. Qureshi, K. F. Iqbal, S. M. Qamar, F. Islam, Y. Ayaz, and N. Muhammad, "Potential guided directional-rrt* for accelerated motion planning in cluttered environments," in *International Conference on Mechatronics and Automation*, 2013, pp. 519–524.

[21] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," *Autonomous Robots*, vol. 40, no. 6, pp. 1079–1093, 2016.

[22] N. Pérez-Higueras, F. Caballero, and L. Merino, "Learning Robot Navigation Behaviors by Demonstration Using a RRT* Planner," in *International Conference on Social Robotics*, 2016, pp. 1–10.

[23] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *Computer Vision and Pattern Recognition*, 2011, pp. 3153–3160.

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

[25] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[26] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European Conference on Computer Vision*, 2016, pp. 549–565.

[27] 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.