2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

S2P2: Self-Supervised Goal-Directed Path Planning Using RGB-D Data for Robotic Wheelchairs

Hengli Wang, Yuxiang Sun, Rui Fan, and Ming Liu, *Senior Member, IEEE*

***Abstract*— Path planning is a fundamental capability for autonomous navigation of robotic wheelchairs. With the impressive development of deep-learning technologies, imitation learning-based path planning approaches have achieved effective results in recent years. However, the disadvantages of these approaches are twofold: 1) they may need extensive time and labor to record expert demonstrations as training data; and 2) existing approaches could only receive high-level commands, such as turning left/right. These commands could be less sufficient for the navigation of mobile robots (*e.g.,* robotic wheelchairs), which usually require exact poses of goals. We contribute a solution to this problem by proposing S2P2, a self-supervised goal-directed path planning approach. Specifically, we develop a pipeline to automatically generate planned path labels given as input RGB-D images and poses of goals. Then, we present a best-fit regression plane loss to train our data-driven path planning model based on the generated labels. Our S2P2 does not need pre-built maps, but it can be integrated into existing map-based navigation systems through our framework. Experimental results show that our S2P2 outperforms traditional path planning algorithms, and increases the robustness of existing map-based navigation systems. Our project page is available at `https://sites.google.com/view/s2p2`.**

I. Introduction

Robotic wheelchairs play a significant role in improving the mobility of disabled and elderly people. Autonomous navigation is an essential capability for robotic wheelchairs [1]. To achieve it, traditional approaches typically consist of three modules: perception, path planning and control [2]–[4]. Although decomposing the whole system into individual modules allows each module to be developed independently, it may suffer from the accumulation of the uncertainties of each module. For example, inaccurate perception results may lead to unsafe paths or detours.

To address this problem, Levine *et al.* [5] showed that training the perception and control modules jointly could al-

This work was supported in part by the National Natural Science Foundation of China under grant U1713211, in part by the Collaborative Research Fund by Research Grants Council Hong Kong under Project C4063-18G, in part by the HKUST-SJTU Joint Research Collaboration Fund under project SJTU20EG03, in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 62003286, and in part by the Start-up Fund of HK PolyU under Grant P0034801. *(Corresponding author: Ming Liu.)*

Hengli Wang and Ming Liu are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China (email: hwangdf@connect.ust.hk; eelium@ust.hk).

Yuxiang Sun is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: yx.sun@polyu.edu.hk, sun.yuxiang@outlook.com).

Rui Fan is with the Department of Computer Science and Engineering, and the Department of Ophthalmology, the University of California San Diego, La Jolla, CA 92093, United States (email: rui.fan@ieee.org).

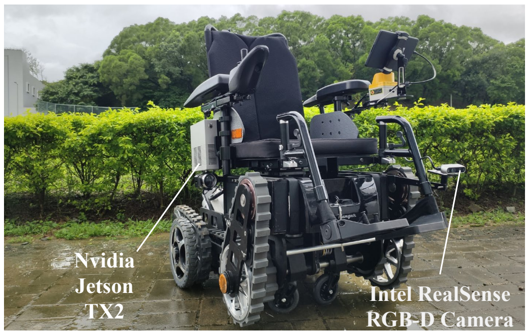


Fig. 1: The robotic wheelchair used in this work. It is equipped with an Intel RealSense RGB-D camera to collect data and an NVIDIA Jetson TX2 to run our S2P2 model.

low individual modules to cooperatively improve the overall performance. Recently, many researchers have resorted to the end-to-end control paradigm [6]–[8], which integrates perception, path planning and control into one module. However, Xu *et al.* [9] argued that the learned control policy would be limited to specific actuation setups or the simulation environments in which the training was performed.

Different from the above-mentioned approaches, many end-to-end path planning approaches based on imitation learning have been developed, which take as input raw sensor data and output planned paths instead of control signals to make the model more generic [10]–[14]. However, these approaches generally have two disadvantages: 1) experts are often required to drive robots many times in various scenes, so extensive time and labor may be needed to record expert demonstrations as training data; and 2) existing imitation learning-based approaches could only receive high-level commands at runtime, such as turning left/right or going straight [12]. These commands could be less sufficient for the navigation of mobile robots (*e.g.,* our robotic wheelchair shown in Fig. 1), which usually require exact poses of goals. For example, given a right-turning command, a robotic wheelchair in a large free space could have many options to go to the right side. The simple high-level commands may not accurately direct the robot to the goal.

To tackle the above issues, we present S2P2, a Self-Supervised goal-directed Path Planning approach for robotic wheelchairs, using an Intel RealSense RGB-D camera. The main difference between our S2P2 and other networks is that our S2P2 does not need manual demonstrations and can directly take as input poses of goals instead of high-level commands. Specifically, we adopt the end-to-end path

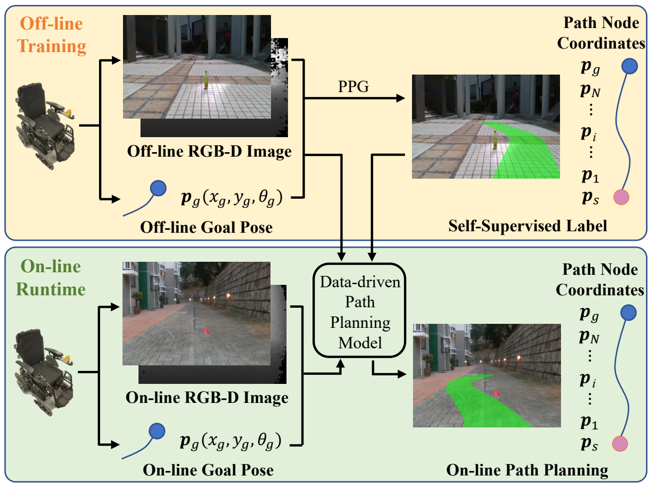


Fig. 2: An overview of our S2P2. We first use our proposed PPG to generate self-supervised labels (top), which are then used to train the data-driven path planning model based on our best-fit regression plane loss. At runtime, a robotic wheelchair equipped with an RGB-D camera can perform the on-line path planning (bottom).

planning paradigm, which directly takes as input RGB-D images as well as poses of goals, and outputs planned paths.

Fig. 2 illustrates the overview of our S2P2. We first develop a pipeline named the planned path generator (PPG) to automatically generate planned path labels given as input RGB-D images [15]–[17] and poses of goals. Then, we present a best-fit regression plane loss to train our proposed data-driven path planning model based on the generated labels. We also propose a framework that allows our mapless S2P2 to be integrated into existing map-based navigation systems. Experimental results show that our S2P2 outperforms traditional path planning algorithms, and increases the robustness of existing map-based navigation systems. The contributions of this paper are summarized as follows:

1) We develop S2P2, which contains an automatic labeling pipeline named PPG, a novel best-fit regression plane loss and a data-driven path planning model.
2) We propose a framework allowing our mapless S2P2 to be integrated with any map-based navigation system.
3) Experimental results demonstrate the superiority of both our S2P2 and the S2P2-integrated navigation system.

II. Related Work

A. Traditional Path Planning Approaches

Traditional path planning algorithms could be generally divided into two categories, complete algorithms and sampling-based algorithms. Complete algorithms, such as A* [18] and JPS [19], could always find a solution if one exists, but they are computationally intensive. Sampling-based algorithms, such as PRM [20] and RRT* [21], trade off between the quality of planned paths and efficiency. These algorithms may fail due to the uncertainties from the perception module.

To address this problem, many researchers have proposed to plan under obstacle uncertainties [22]–[24]. Kuwata *et al.* [22] formulated this problem as a chance-constrained dynamic programming problem, and solved it by performing cost analysis. Recently, Jasour *et al.* [24] proposed a novel risk-contour map and employed this map to obtain safe paths for robots with guaranteed bounded risks.

B. End-to-end Path Planning Approaches

There are many studies that could output the coordinates of the planned paths via imitation learning [10]–[14]. Bergqvist *et al.* [10] compared different combinations of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, and concluded that the path planned by LSTM or CNN-LSTM is smooth and feasible in many situations. Cai *et al.* [12] proposed a vision-based model, which receives camera images to plan a collision-free trajectory in the future. These imitation learning-based approaches, however, have two disadvantages as mentioned in Section I, which greatly limit their applications.

C. End-to-end Control Approaches

ALVINN [25] was the first attempt to implement end-to-end control for mobile robots by using a shallow neural network. Inspired by ALVINN, Bojarski *et al.* [7] proposed PilotNet, which utilizes CNNs to map front-view images directly to steering commands. There also exist studies that utilize other sensors (*e.g.,* LiDARs) besides cameras [6], [8]. However, these end-to-end control approaches exhibit low generalization capabilities as already mentioned [9].

III. Methodology

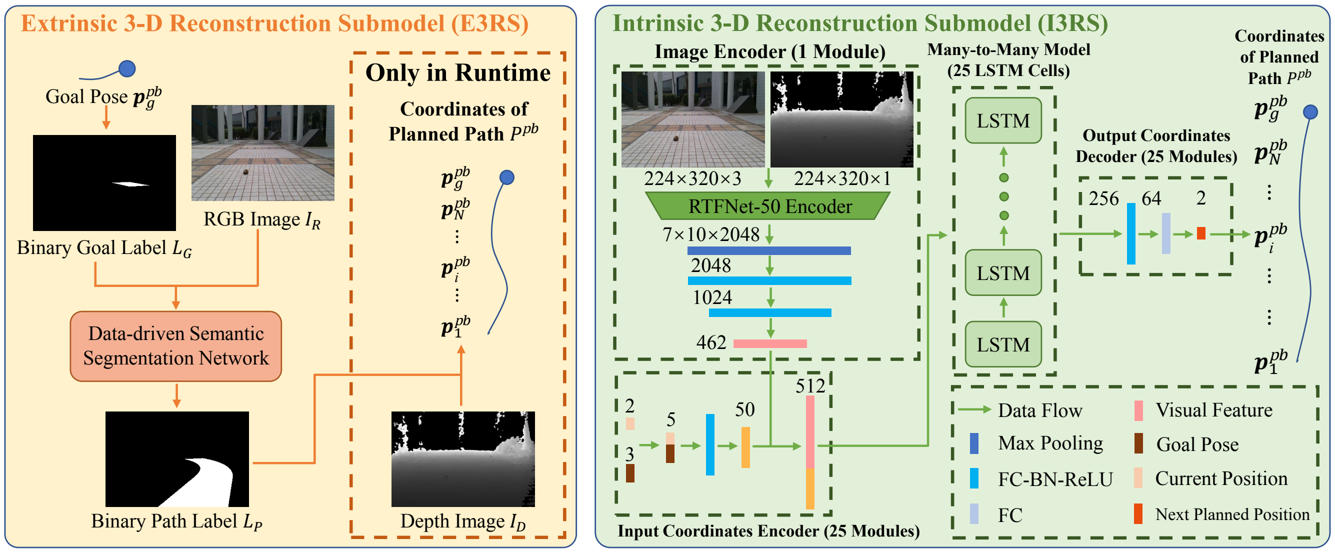
A. Problem Formulation

Let $(\cdot)^w$ denote the world frame, $(\cdot)^b$ denote the robot body frame and $(\cdot)^c$ denote the camera frame. $\mathbf{R}_b^w$ and $\mathbf{T}_b^w$ represent the rotation matrix and the translation matrix from the body frame to the world frame, respectively. Given an input registered front-view RGB image I_R and depth image I_D, we can construct a 3-D point cloud of the front scene, which contains the configuration space of the robotic wheelchair. To simplify the analysis, we set the configuration space $C \subset \mathbb{R}^2 \times \mathrm{SO}(2)$, where $\mathrm{SO}(2)$ denotes the 2-D rotation group. Since the z-axis value has no impact on the path planning of robotic wheelchairs, we consider a 2-D projected body frame denoted by $(\cdot)^{pb}$, which coincides with the $x - y$ plane of the body frame.

We consider the problem of generating a path given any arbitrary goal pose $\boldsymbol{p}_g^{pb} \in C$ without colliding with obstacles. Then, the generated path can be expressed as:

$$P^{pb} \triangleq \left\{ \boldsymbol{p}_1^{pb}, \ldots, \boldsymbol{p}_i^{pb}, \ldots, \boldsymbol{p}_N^{pb}, \boldsymbol{p}_g^{pb} \right\}, \tag{1}$$

where $\boldsymbol{p}_i^{pb} \in C$ denotes a node of the planned path. In this paper, we set $N = 24$. Note that we take the orientation of the goal pose θ_g^{pb} into consideration because it could affect the generated paths. However, our S2P2 does not output the orientation of the generated path nodes because the subsequent control module only needs the position information.



Extrinsic 3-D Reconstruction Submodel (E3RS)
Goal Pose p_g^{pb}
RGB Image I_R
Binary Goal Label L_G
Data-driven Semantic Segmentation Network
Binary Path Label L_P
Only in Runtime
Coordinates of Planned Path P^{pb}
Depth Image I_D
Intrinsic 3-D Reconstruction Submodel (I3RS)
Image Encoder (1 Module)
224×320×3
224×320×1
RTFNet-50 Encoder
7×10×2048
2048
1024
462
Many-to-Many Model (25 LSTM Cells)
LSTM
Output Coordinates Decoder (25 Modules)
256
64
2
Coordinates of Planned Path P^{pb}
Input Coordinates Encoder (25 Modules)
512
50
5
2
3
Data Flow
Max Pooling
FC-BN-ReLU
FC
Visual Feature
Goal Pose
Current Position
Next Planned Position

Fig. 3: An overview of our data-driven path planning model, which consists of two different submodels. E3RS (left) transforms the original path planning problem to a semantic segmentation problem. At runtime, the coordinates of the planned path P^{pb} can be computed by extrinsically reconstructing the binary path label L_P. I3RS (right) utilizes a CNN-LSTM neural network to output P^{pb} intrinsically. Each LSTM module takes as input the combined feature including the current position information, and outputs a feature that is then decoded to the next planned position.

B. Planned Path Generator

Our PPG is designed to automatically generate planned path labels P^{pb} given as input RGB-D images and poses of goals. Given I_R and I_D, we first use [26] to generate corresponding semantic segmentation image I_S, which can provide pixel-level predictions of the drivable area and obstacles. For any pixel $\boldsymbol{q}$ in the image, we calculate its 3-D coordinate in the camera frame $\boldsymbol{q}^c$ by using I_D. Then, we can obtain the point cloud in the camera frame PC^c by calculating the 3-D coordinates of all pixels. We further employ [27] to filter out outliers in PC^c. Afterwards, we build an occupancy grid map named costmap M^{pb} in the projected body frame. The initialization approach is described in Algorithm 1, where $\boldsymbol{q}_d^{pb}$ and $\boldsymbol{q}_o^{pb}$ denote the point belonging to the drivable area and obstacles, respectively. For any point $\boldsymbol{q}^c \in PC^c$, we calculate its coordinate in the projected body frame $\boldsymbol{q}^{pb}$ via $\mathbf{R}_c^b$ and $\mathbf{T}_c^b$ (line 1). We then constrict the free area (line 7) and inflate the occupied area (line 8). Since the size of our robotic wheelchair is $1m \times 0.5m$, we accordingly set two radiuses both as $0.5m$. The size of each cell in M^{pb} is $0.1m \times 0.1m$.

Now, given a goal pose $\boldsymbol{p}_g^{pb}$, we can plan a path P^{pb} in M^{pb} by using traditional path planning algorithms. If $\boldsymbol{p}_g^{pb}$ lies outside the free area due to perception or localization errors, we will take the closest point to $\boldsymbol{p}_g^{pb}$ in the free area as the new goal and replan the path. After obtaining P^{pb} from the above steps, we project the input goal pose and the planned path to the original image and obtain the binary path label L_P and binary goal label L_G, as presented in Fig. 3. By sampling goal poses randomly, our proposed PPG can generate a large number of planned path labels given as input RGB-D images, which saves much time and labor.

Algorithm 1: Costmap Initialization

```
   Input: $PC^c$.
   Output: $M^{pb}$.
1  Compute $\boldsymbol{q}^{pb}$ for every point $\boldsymbol{q}^c \in PC^c$
2  Initialize $M^{pb}$ as unknown area
3  Compute region $R_d^{pb} \leftarrow$ findConvexHull $\left(\boldsymbol{q}_d^{pb}\right)$
4  Set $M^{pb}\left(R_d^{pb}\right)$ as free area
5  Compute region $R_o^{pb} \leftarrow$ findConvexHull $\left(\boldsymbol{q}_o^{pb}\right)$
6  Set $M^{pb}\left(R_o^{pb}\right)$ as occupied area
7  ConstrictFreeArea $\left(M^{pb}\right)$
8  InflateOccupiedArea $\left(M^{pb}\right)$
```

C. Data-driven Path Planning Model

Although our PPG can generate the coordinates of the planned paths, the generated paths often present detours or unsafe routes due to perception errors. Therefore, we propose a data-driven path planning model to provide better planned paths given as input RGB-D images and poses of goals. The overview of our model is illustrated in Fig. 3.

Since the nodes of the planned paths are in sequential order, our model should be able to model this relationship. To this end, we propose two different submodels. The extrinsic 3-D reconstruction submodel (E3RS) analogizes this sequential relationship to the spatial continuity of an image and transforms this problem to a semantic segmentation problem, while the intrinsic 3-D reconstruction submodel (I3RS) utilizes LSTM to model this sequential relationship.

1) Extrinsic 3-D Reconstruction Submodel (E3RS): We first project the input goal pose $\boldsymbol{p}_g^{pb}$ to the image frame and

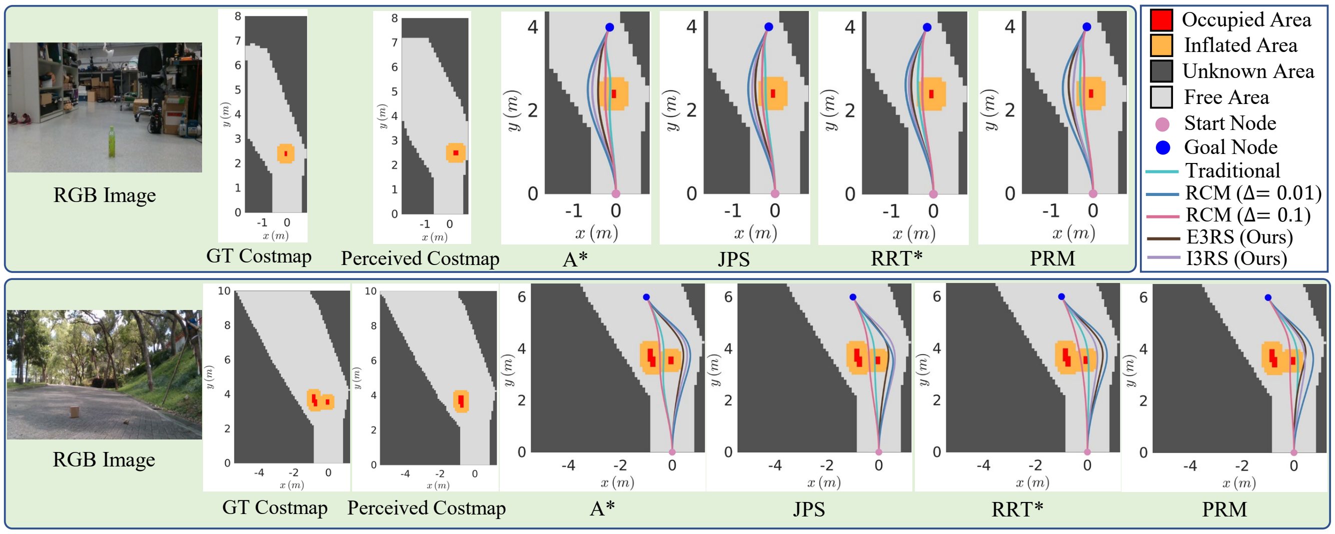


Fig. 4: Two examples (indoor and outdoor) with inaccurate perception results for the comparison between traditional algorithms, RCM [24] with two different risk bounds, and our proposed E3RS and I3RS. Perceived costmaps and ground-truth costmaps are constructed by the semantic predictions and ground-truth semantic segmentation labels, respectively. The paths planned by different approaches are all displayed on the ground-truth costmaps. Although the perception results are not entirely accurate, our proposed E3RS and I3RS can still present a better performance than the other approaches.

obtain the binary goal label L_G. Then, we train an existing data-driven semantic segmentation network, RTFNet-50 [28], that takes as input the binary goal label L_G and RGB image I_R, and outputs binary path label L_P. At runtime, we use I_D to compute P^{pb} for the subsequent control module.

2) Intrinsic 3-D Reconstruction Submodel (I3RS): We construct a CNN-LSTM network to model the sequential relationship between each node of the planned path. We first use the encoder of RTFNet-50 [28] to extract the visual features of a given RGB-D image. To balance the dimensional difference between the coordinate vector and visual features, we encode the input coordinate vector to a feature that has a higher dimension, which is then concatenated with the visual features and fed to the LSTM cell. Each LSTM cell takes the combined feature as input, and outputs a feature that is then decoded to the next planned position. The current position feature fed to one LSTM cell comes from the next planned position feature given by the previous LSTM cell, and the first LSTM cell takes $(0, 0)$ as the current position. Since P^{pb} contains 25 nodes including the input goal node, there are a total of 25 LSTM cells that constitute our many-to-many LSTM model.

3) Training Loss for Our E3RS and I3RS: The training loss for our E3RS $\mathcal{L}_E$ consists of two terms:

$$\mathcal{L}_E = \mathcal{L}_{EP} + \lambda_{ER}\mathcal{L}_{ER}, \tag{2}$$

where $\mathcal{L}_{EP}$ and $\mathcal{L}_{ER}$ denote the PPG guiding loss and the best-fit regression plane loss, respectively. $\mathcal{L}_{EP}$ takes the binary path label generated by PPG $\widehat{L}_P$ as the supervision, and is defined as the cross entropy between $\widehat{L}_P$ and the E3RS prediction L_P. Moreover, since we regard the desired path as a plane, we design $\mathcal{L}_{ER}$ to penalize off-plane pixels with high probability in L_P. Referring to the generalized v-

Algorithm 2: Intermediate Goal Pose Generator

```
Input: $P^w, r$.
Output: $G^w$.
1 $G^w \leftarrow \emptyset$
2 $cur \leftarrow \boldsymbol{p}_s^w$
3 for $i = 1 \rightarrow M$ do
4     if $\left(\textbf{Visible}\left(\boldsymbol{p}_i^w, cur\right) \wedge \neg\textbf{Visible}\left(\boldsymbol{p}_{i+1}^w, cur\right)\right) \vee$
        $\left(\textbf{Dist}\left(\boldsymbol{p}_{i+1}^w, cur\right) > r\right)$ then
5         $G^w$.insert $\left(\boldsymbol{p}_i^w\right)$
6         $cur \leftarrow \boldsymbol{p}_i^w$
7     end
8 end
9 $G^w$.insert $\left(\boldsymbol{p}_g^w\right)$
```

disparity analysis discussed in [29], [30], the inverse depth (or disparity) pixels $1/I_D(\boldsymbol{q})$ are typically projected as a non-linear pattern $f(\boldsymbol{a}, \boldsymbol{q}, \phi) = a_0 + a_1(-u\sin\phi + v\cos\phi)$ in the v (vertical) direction [31], [32], where $\boldsymbol{q} = [u, v]^T$ denotes the pixel and ϕ denotes the RGB-D camera roll angle. $\boldsymbol{a} = [a_0, a_1]^T$ and ϕ can be yielded by finding the best-fit regression plane, which corresponds to the minimum of the mean of squared residuals between the non-linear pattern and the pixels $\boldsymbol{q}$ with high probability in L_P [33], [34]:

$$\mathcal{L}_{ER} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(1/I_D(\boldsymbol{q}_i) - f(\hat{\boldsymbol{a}}, \boldsymbol{q}_i, \hat{\phi})\right)^2, \tag{3}$$

where N_p represents the number of pixels with a probability larger than 0.5 in L_P; and $\hat{\boldsymbol{a}}$ and $\hat{\phi}$ separately denote the optimum $\boldsymbol{a}$ and ϕ. Their closed-form solutions are provided in [33]. The off-plane pixels with high probability in L_P can

TABLE I: Performance comparison between traditional algorithms, RCM [24] with two different risk bounds, and our E3RS and I3RS. ↑ means higher numbers are better, and ↓ means lower numbers are better. The best results are bolded.

Evaluation Metrics	Traditional (PPG)				E3RS (**Ours**)				I3RS (**Ours**)				RCM ($\Delta = 0.01$)				RCM ($\Delta = 0.1$)			
	A*	JPS	RRT*	PRM	A*	JPS	RRT*	PRM	A*	JPS	RRT*	PRM	A*	JPS	RRT*	PRM	A*	JPS	RRT*	PRM
SR ↑	70.7%	65.6%	70.4%	68.6%	89.2%	86.9%	**91.7%**	88.1%	90.2%	87.9%	89.6%	86.4%	87.8%	85.7%	86.5%	83.6%	69.6%	67.3%	70.2%	65.9%
TC ↓	0.132	0.151	0.127	0.187	0.115	0.116	0.089	0.096	0.091	0.121	**0.082**	0.125	0.159	0.167	0.131	0.192	0.137	0.152	0.125	0.179

produce a relatively high $\mathcal{L}_{ER}$, and vice versa.

Similarly, the training loss for our I3RS $\mathcal{L}_I$ also consists of two terms:

$$\mathcal{L}_I = \mathcal{L}_{IP} + \lambda_{IR}\mathcal{L}_{IR}, \tag{4}$$

where $\mathcal{L}_{IP}$ and $\mathcal{L}_{IR}$ denote the PPG guiding loss and the best-fit regression plane loss, respectively. $\mathcal{L}_{IP}$ takes the planned path generated by PPG $\widehat{P}^{pb}$ as the supervision:

$$\mathcal{L}_{IP} = \frac{1}{25}\sum_{i=1}^{25}\left\|\widehat{\boldsymbol{p}}_i^{pb} - \boldsymbol{p}_i^{pb}\right\|_2, \tag{5}$$

where $\widehat{\boldsymbol{p}}_i^{pb}$ and $\boldsymbol{p}_i^{pb}$ denote the planned path node via PPG and I3RS, respectively; and $||\cdot||$ denotes the $L2$-Norm. Moreover, we use the I3RS prediction P^{pb} and I_D to compute $\mathcal{L}_{IR}$ based on (3). $\mathcal{L}_{IR}$ is also employed to penalize the on-path but off-plane points.

D. The Proposed Framework for Integrating S2P2 into Existing Map-based Navigation Systems

For long-range autonomous navigation tasks, the input goals are often outside the field of view (FOV) of the front-view camera. To address this problem, we propose a framework allowing our mapless S2P2 to be integrated into existing map-based navigation systems.

Given a goal pose, we can first plan a global collision-free path in the world frame $P^w = \{\boldsymbol{p}_s^w, \boldsymbol{p}_1^w, \dots, \boldsymbol{p}_M^w, \boldsymbol{p}_g^w\}$ using global path planners (*e.g.*, PRM [20]), where $\boldsymbol{p}_s^w$ and $\boldsymbol{p}_g^w$ are start and goal poses, respectively. Then we use Algorithm 2 to generate an intermediate goal pose array G^w, where each pose can lie within the FOV of the camera. r is the distance measurement range of the camera, and we set $r = 10m$ in this paper. In Algorithm 2, **Visible** $(\boldsymbol{m}, \boldsymbol{n})$ determines whether pose $\boldsymbol{m}$ is within the FOV of the camera when the robot is located at $\boldsymbol{n}$. **Dist** $(\boldsymbol{m}, \boldsymbol{n})$ is the euclidean distance between $\boldsymbol{m}$ and $\boldsymbol{n}$. With G^w transformed from the world frame to the projected body frame, our mapless S2P2 can be used as a local planner given as input RGB-D images and the intermediate goal poses in G^{pb} incrementally until the robot reaches the input goal. Fig. 6 shows two example trajectories from real-world experiments by using an existing navigation system with S2P2 integrated, where orange points represent the intermediate goals generated by Algorithm 2.

IV. Experimental Results and Discussions

A. Datasets and Implementation Details

We use the RGB-D dataset from [26], which covers 30 common scenes where robotic wheelchairs usually work. The input images are downsampled to 224×320 for efficiency.

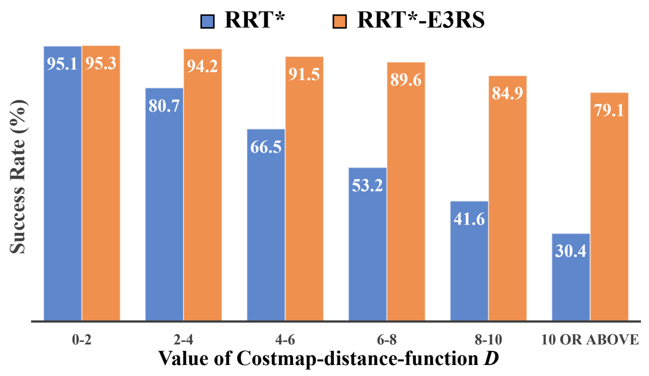


Fig. 5: Comparison of the SR between RRT* and RRT*-E3RS when the quality of the perceived costmaps changes.

For our PPG, we employ four different traditional path planning algorithms: two complete algorithms, A* [18] and JPS [19]; and two sampling-based algorithms, PRM [20] and RRT* [21]. Each PPG generates a total number of 26429 self-supervised planned path labels given as input RGB-D images and randomly sampled goal poses. The 26429 planned path labels are split into training, validation and test sets, which contain 15859, 5285 and 5285 samples, respectively. Moreover, we adopt $\lambda_{ER} = 0.10$ and $\lambda_{IR} = 0.15$ in our experiments. For all networks, we use the stochastic gradient descent (SGD) optimizer and adopt an initial learning rate of 10^{-4}.

We use two metrics, of which the most important is success rate (SR). SR is defined as the ratio of the number of successfully generated paths and total generated paths. A successfully generated path is defined as a path that reaches the input goal pose without colliding with obstacles. Inspired by [35], we also adopt turning cost (TC) to measure the smoothness of each path, as follows:

$$TC = \frac{\sum_{i=1}^{25}\left|\theta_i^{pb}\right|}{25 \times 90^\circ}, \tag{6}$$

where θ_i^{pb} denotes the turning angle at $\boldsymbol{p}_i^{pb}$.

B. Path Planning Results

Jasour *et al.* [24] proposed a novel risk-contour map (RCM), a state-of-the-art approach for path planning under obstacle uncertainties. We test this approach with two different risk bounds in our test set. Table I presents the evaluation results. It is evident that our E3RS and I3RS present significant improvements compared with the other approaches. Note that although RCM with a low risk bound

TABLE II: Experimental results of different variants. $\mathcal{L}_R$ denotes $\mathcal{L}_{ER}$ and $\mathcal{L}_{IR}$ for our E3RS and I3RS, respectively. ↑ and ↓ mean higher and lower numbers are better, respectively. The best results for our E3RS and I3RS are both bolded.

No.	Architecture	$\mathcal{L}_R$	SR ↑	TC ↓
(a)	A*-PPG	–	70.7%	0.132
(b)	A*-E3RS	–	68.6%	0.139
(c)	A*-E3RS (**Adopted**)	✓	**89.2%**	**0.115**
(d)	A*-I3RS	–	71.8%	0.130
(e)	A*-I3RS (**Adopted**)	✓	**90.2%**	**0.091**
(f)	A*-I3RS (Only RGB Images)	✓	83.1%	0.124
(g)	A*-I3RS (Only Depth Images)	✓	81.4%	0.119
(h)	A*-I3RS (One-to-Many LSTM)	✓	77.2%	0.128
(i)	A*-I3RS (FCN)	✓	76.3%	0.155

performs well in SR, it presents a much worse performance in TC than E3RS and I3RS because it can generate detours easily to meet the low-risk-bound requirement. To analyze why our E3RS and I3RS can perform better than the PPG, we present some experimental results in (a)–(e) of Table II. We can clearly observe that our best-fit regression plane loss $\mathcal{L}_R$ can effectively reduce the adverse impact of off-plane points and further improve the performance of planned paths.

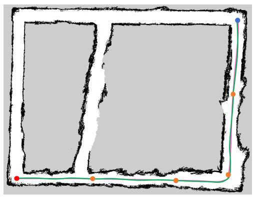
We also choose RRT* and our RRT*-E3RS for further analysis. We use the costmap-distance-function D in [36] to measure the difference between the perceived costmap constructed by the predicted semantic segmentation image and the ground-truth costmap constructed by the ground-truth semantic segmentation label. The larger the value of D, the worse the quality of the perceived costmap. Then, we divide the test set into six categories based on the quality of the perceived costmaps, and test the SR of RRT* and RRT*-E3RS on each category. Fig. 5 shows that the SR of RRT* decreases much more rapidly than our RRT*-E3RS when the quality of the perceived costmap drops, which is also confirmed by the qualitative results shown in Fig. 4. We can see that when the perception results are not very accurate, our E3RS and I3RS still present a better performance than the other approaches. The reason is that the other approaches depend on the perception results, while our S2P2 is an end-to-end approach that does not. In addition, our best-fit regression plane loss $\mathcal{L}_R$ can effectively improve the performance of planned paths.

C. Ablation Study

We perform ablation studies on our A*-I3RS, which presents the best performance in our I3RS. We first test the network structures with only one kind of input, either RGB or depth images. The results in (e)–(g) of Table II demonstrates the superiority of using RGB-D images [37]. We speculate that it is promising to fuse RGB images with other modalities of data, such as surface normal [38], [39] and optical flow [40], [41], for autonomous navigation. To show the effectiveness of our many-to-many LSTM model, we replace the many-to-many LSTM model with two different models, a one-to-many LSTM model and a fully connected network (FCN). We can see that our many-to-many LSTM model achieves the best results from (e), (h) and (i) of Table II.

TABLE III: Performance comparison between different navigation systems in real-world experiments with our robotic wheelchair. The best results are bolded.

Approaches	SR (Indoor)	SR (Outdoor)
PRM-APF [44]	60%	50%
PRM-DWA [45]	65%	55%
PRM-S2P2 (**Ours**)	**95%**	**90%**



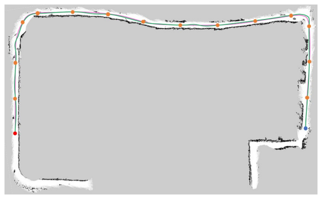


Fig. 6: Two examples from real-world experiments using PRM-S2P2. Green and magenta lines denote the actual robot path and the path planned by the global planner PRM, respectively. Red, blue and orange points denote the start, the goal and the intermediate goals generated by Algorithm 2, respectively.

D. Navigation Experiments with Our Robotic Wheelchair

To test the performance of the existing navigation system integrated with our S2P2, we use our robotic wheelchair to perform navigation tasks in one indoor environment and one outdoor environment, respectively. We choose our best approach RRT*-E3RS for our S2P2. The RGB-D SLAM system RTAB-Map [42] is adopted for mapping and localization. Additionally, we use PRM [20] as the global path planner and optimal time allocation [43] as the trajectory generator. The robotic wheelchair is commanded to track the trajectory. We call this navigation system PRM-S2P2. Fig. 6 shows two example trajectories in real-world environments using our PRM-S2P2. We can see that the robotic wheelchair can reach the goal successfully without colliding with obstacles in both the indoor and outdoor environments. As aforementioned, our S2P2 behaves as a local planner, and therefore we also compare the performance between our PRM-S2P2 and the same navigation system with other local planners integrated. The results in Table III demonstrate the superiority of our PRM-S2P2 over PRM-APF [44] and PRM-DWA [45].

V. Conclusions

In this paper, we proposed S2P2, a self-supervised goal-directed path planning approach for robotic wheelchairs. Experimental results have demonstrated that our S2P2 outperforms traditional path planning algorithms, and increases the robustness of existing map-based navigation systems. One limitation is that the proposed approach does not explicitly model moving obstacles. Therefore, in the future, we will incorporate the moving obstacle model into our S2P2, to enable the mobile robot to present more robust and accurate navigation performance in dynamic environments.

REFERENCES

[1] H. Wang, R. Fan, Y. Sun, and M. Liu, "Dynamic fusion module evolves drivable area and road anomaly detection: A benchmark and algorithms," *IEEE Trans. Cybern.*, 2021.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Comput.*, vol. 22, no. 6, pp. 46–57, 1989.

[3] R. Fan, H. Wang, P. Cai, and M. Liu, "SNE-RoadSeg: Incorporating surface normal information into semantic segmentation for accurate freespace detection," in *Eur. Conf. Comput. Vision*. Springer, 2020, pp. 340–356.

[4] T. Liu, Q. Liao, L. Gan, F. Ma, J. Cheng, X. Xie, Z. Wang, Y. Chen, Y. Zhu, S. Zhang, Z. Chen, Y. Liu, M. Xie, Y. Yu, Z. Guo, G. Li, P. Yuan, D. Han, Y. Chen, H. Ye, J. Jiao, P. Yun, Z. Xu, H. Wang, H. Huang, S. Wang, P. Cai, Y. Sun, Y. Liu, L. Wang, and M. Liu, "The role of the hercules autonomous vehicle during the covid-19 pandemic: An autonomous logistic vehicle for contactless goods transportation," *IEEE Robot. Automat. Mag.*, pp. 0–0, 2021.

[5] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2016.

[6] S. Hecker, D. Dai, and L. Van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 435–453.

[7] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017.

[8] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1527–1533.

[9] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2174–2182.

[10] M. Bergqvist and O. Rödholm, "Deep Path Planning Using Images and Object Data," Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2018.

[11] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[12] P. Cai, Y. Sun, H. Wang, and M. Liu, "VTGNet: A vision-based trajectory generation network for autonomous vehicles in urban environments," *IEEE Trans. Intell. Vehicles*, 2020.

[13] Y. Sun, W. Zuo, and M. Liu, "See the future: A semantic segmentation network predicting ego-vehicle trajectory with a single monocular camera," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3066–3073, 2020.

[14] H. Wang, P. Cai, Y. Sun, L. Wang, and M. Liu, "Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation," in *IEEE Int. Conf. Robot. Autom.* IEEE, 2021.

[15] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D slam in dynamic environments: A motion removal approach," *Robot. Autom. Syst.*, vol. 89, pp. 110 – 122, 2017.

[16] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Autom. Syst.*, vol. 108, pp. 115 – 128, 2018.

[17] Y. Sun, M. Liu, and M. Q.-H. Meng, "Active Perception for Foreground Segmentation: An RGB-D Data-Based Background Modeling Method," *IEEE Trans. Automat. Sci. Eng.*, pp. 1–14, 2019.

[18] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cyber.*, vol. 4, no. 2, pp. 100–107, 1968.

[19] D. D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *25th AAAI Conf. Artif. Intell.*, 2011.

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

[21] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *arXiv preprint arXiv:1005.0416*, 2010.

[22] Y. Kuwata, M. Pavone, and J. Balaram, "A risk-constrained multi-stage decision making approach to the architectural analysis of planetary missions," in *IEEE Conf. Decis. Control.* IEEE, 2012, pp. 2102–2109.

[23] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, "Chance-constrained dynamic programming with application to risk-aware robotic space exploration," *Auton. Robots*, vol. 39, no. 4, pp. 555–571, 2015.

[24] A. M. Jasour and B. C. Williams, "Risk contours map for risk bounded motion planning under perception uncertainties," *Robot. Sci. and Syst.*, 2019.

[25] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances Neural Inf. Process. Syst.*, 1989, pp. 305–313.

[26] H. Wang, Y. Sun, and M. Liu, "Self-Supervised Drivable Area and Road Anomaly Segmentation Using RGB-D Data For Robotic Wheelchairs," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4386–4393, Oct 2019.

[27] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robot. Auton. Syst.*, vol. 56, no. 11, pp. 927–941, 2008.

[28] Y. Sun, W. Zuo, and M. Liu, "RTFNet: RGB-Thermal Fusion Network for Semantic Segmentation of Urban Scenes," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2576–2583, July 2019.

[29] R. Fan, J. Jiao, J. Pan, H. Huang, S. Shen, and M. Liu, "Real-time dense stereo embedded in a UAV for road inspection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*. IEEE Computer Society, 2019, pp. 535–543.

[30] R. Fan, U. Ozgunalp, B. Hosking, M. Liu, and I. Pitas, "Pothole detection based on disparity transformation and road surface modeling," *IEEE Trans. Image Process.*, vol. 29, pp. 897–908, 2019.

[31] R. Fan, H. Wang, P. Cai, J. Wu, M. J. Bocus, L. Qiao, and M. Liu, "Learning collision-free space detection from stereo images: Homography matrix brings better data augmentation," *IEEE/ASME Trans. Mechatronics*, 2021.

[32] R. Fan, U. Ozgunalp, Y. Wang, M. Liu, and I. Pitas, "Rethinking road surface 3d reconstruction and pothole detection: From perspective transformation to disparity map segmentation," *IEEE Trans. Cybern.*, 2021.

[33] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4906–4911, 2019.

[34] R. Fan, H. Wang, M. J. Bocus, and M. Liu, "We learn better road pothole detection: from attention aggregation to adversarial domain adaptation," in *Eur. Conf. Comput. Vision*. Springer, 2020, pp. 285–300.

[35] E. Tsardoulias, A. Iliakopoulou, A. Kargakos, and L. Petrou, "A review of global path planning methods for occupancy grid maps regardless of obstacle density," *J. Intell. Robot. Syst.*, vol. 84, no. 1-4, pp. 829–858, 2016.

[36] A. Birk, "Learning geometric concepts with an evolutionary algorithm," *Environ.*, vol. 4, p. 3, 1996.

[37] H. Wang, R. Fan, P. Cai, and M. Liu, "PVStereo: Pyramid voting module for end-to-end self-supervised stereo matching," *IEEE Robot. Autom. Lett.*, 2021.

[38] H. Wang, R. Fan, Y. Sun, and M. Liu, "Applying surface normal information in drivable area and road anomaly detection for ground mobile robots," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020.

[39] R. Fan, H. Wang, B. Xue, H. Huang, Y. Wang, M. Liu, and I. Pitas, "Three-filters-to-normal: An accurate and ultrafast surface normal estimator," *IEEE Robot. Autom. Lett.*, 2021.

[40] H. Wang, R. Fan, and M. Liu, "CoT-AMFlow: Adaptive modulation network with co-teaching strategy for unsupervised optical flow estimation," *arXiv preprint arXiv:2011.02156*, 2020.

[41] H. Wang, Y. Liu, H. Huang, Y. Pan, W. Yu, J. Jiang, D. Lyu, M. J. Bocus, M. Liu, I. Pitas *et al.*, "ATG-PVD: Ticketing parking violations on a drone," in *Eur. Conf. Comput. Vision*. Springer, 2020, pp. 541–557.

[42] M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 734–745, 2013.

[43] F. Gao, W. Wu, J. Pan, B. Zhou, and S. Shen, "Optimal time allocation for quadrotor trajectory generation," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4715–4722.

[44] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," in *IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2347–2354.

[45] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.