Small Obstacle Avoidance Based on RGB-D Semantic Segmentation

Minjie Hua*
CloudMinds Technologies
michael.hua@cloudminds.com

Yibing Nan*
CloudMinds Technologies
charlie.nan@cloudminds.com

Shiguo Lian
CloudMinds Technologies
sg_lian@163.com

Abstract

This paper presents a novel obstacle avoidance system for road robots equipped with RGB-D sensor that captures scenes of its way forward. The purpose of the system is to have road robots move around autonomously and constantly without any collision even with small obstacles, which are often missed by existing solutions. For each input RGB-D image, the system uses a new two-stage semantic segmentation network followed by the morphological processing to generate the accurate semantic map containing road and obstacles. Based on the map, the local path planning is applied to avoid possible collision. Additionally, optical flow supervision and motion blurring augmented training scheme is applied to improve temporal consistency between adjacent frames and overcome the disturbance caused by camera shake. Various experiments are conducted to show that the proposed architecture obtains high performance both in indoor and outdoor scenarios.

1. Introduction

Obstacle avoidance is a fundamental component for intelligent mobile robots, the core of which is to have an environmental perception module that helps identify the possible hindrance that can block the way of the robots. Especially, for some scenarios, small obstacles should be carefully considered, *e.g.*, autonomous driving, patrol robot and blind guidance, as shown in Fig. 1. In autonomous driving, while the car running at a high speed, such small obstacle as a brick on road may cause the car to turn over. In blind guidance, visually impaired people are fragile to such small object even only 3cm higher on road. For the patrol robot working as a community policing, some remnants scattered on road should be detected either as obstacles to avoid or spilled garbage to alert, such as garbage cans and stone blocks.

Range-based and appearance-based methods are two major approaches to perform the task of obstacle detection.

(a) (b) (c)

Figure 1. Some cases with small obstacles in autonomous driving (a), patrol robot (b) and blind guidance (c).

But the former share disadvantages, where it is difficult to detect small obstacles and distinguish between diverse types of road surfaces, making it hard to identify the sidewalk pavement from adjacent grassy area, a combination commonly seen under urban circumstances.

The appearance-based methods [19, 10, 23, 17, 24], on the other hand, are not subjected to the above problems. Since they define obstacles as objects that differ in appearance, which is more essential a character in our study, from the road, while the former only define obstacles as objects that rise at a certain height from the road. One of the main procedures of appearance-based obstacle avoidance methods can be identified as semantic segmentation of accessible area and other objects such that the robots can make decisions of obstacle avoidance and path planning.

For appearance-based sensors, we have both the monocular camera and the stereo camera. The former outputs pure visual cues (*i.e.* RGB information), where it is difficult for semantic segmentation networks to accurate distinguish real obstacles and appearance changes such as different road colors and road markings. The latter provides additional depth channel to RGB images. Incorporating RGB and depth information could potentially improve the performance of semantic segmentation networks, which is suggested by recent efforts [12, 9, 13].

Utilizing semantic segmentation to obstacle avoidance is a new topic in recent years [15, 20, 8], but these methods are applied to limited scene or not sensitive to small obstacles. In this paper, we propose a novel small obstacle avoidance system based on RGB-D semantic segmentation that can be widely used in indoor and outdoor scenarios.

* indicates equal contribution

The main contributions of this work are as follows: 1). An two-stage RGB-D encoder-decoder network is proposed for obstacle segmentation, which segments the image to get the road mask first and then gets more accurate obstacle region even small obstacle region from the extracted road area. 2) An optical flow supervision between adjacent frames is proposed for the segmentation network to keep the temporal consistency, which is critical for stable obstacle detection when camera moves. 3) A motion blur based data augmentation scheme is proposed to suppress segmentation errors introduced by camera shake. 4) A small obstacle avoidance system based on RGB-D semantic segmentation is proposed and evaluated based on collected practical datasets both of indoor and outdoor scenarios.

2. Related Work

2.1. Obstacle Detection and Avoidance

The first autonomous mobile robot used the appearance-based method for obstacle detection [19], named Shakey, developed by Nilsson. It can detect obstacles by edge detection on the monochrome image. Following Nilsson's step, Horswill also applied edge detection method to his mobile robot named Polly [10], which was operated in real-life environment. The edge detection method that Shakey and Polly used only performed well when the floor had little texture and the environment was uniformly illuminated. It's very sensitive to floor color variation and lighting conditions.

Besides the edge information, some other information have also been used for obstacle detection. Turk and Marra made use of the color information [23] to detect obstacles by subtracting consecutive frames of a video. Lorigo proposed an algorithm using both color information and edge information that worked on texture floor [17]. Lorigo assumed there is no obstacle right in front of the robot and used this area as reference area to find obstacle. Ulrich improved Lorigo's algorithm with a candidate queue and a reference queue [24]. However, before the mobile robot can move autonomously, they need to steer it first for several meters to form a correct reference area. And if the road outside the reference area differs from the reference area due to unexpected shadow or reflections, it will be incorrectly classified as an obstacle as well.

Recently, some other appearance based obstacle avoidance schemes have been proposed. Ghosh and Wei proposed stereo vision based methods to detect obstacles [7, 26]. In these methods, the disparity map was used to analyze surroundings and determine obstacles. However, the disparity map computed by stereo images is not robust enough in complex environments especially in the presence of small obstacles. Yang proposed a blind guiding system based on both RGB and depth images [27]. In this method, the RGB image was used to get semantic map of the environment, and then combined with depth image to percept the terrain. However, this method focuses on the segmentation of transitable areas including pedestrians and vehicles, while it is incapable of detecting small obstacles. For small obstacles, [20, 8] explored the obstacle detection issue focusing on autonomous driving scenarios. In these methods, pre-defined obstacle categories that are common in driving scenarios were placed on road to initialize the obstacle dataset first,and then RGB and depth information were both used to train the obstacle segmentation models. However, the method supports only some normal obstacles predefined on traffic road while not extensible for arbitrary obstacles on road.

2.2. Semantic Segmentation

Recently, there have been some advances on deep neural networks based semantic segmentation. The task of semantic segmentation is to label each pixel of an image with a semantic class. FCN [16], the pioneer work in the exploration of CNN-based semantic segmentation methods, adapts classifiers for dense prediction by replacing the last fully-connected layer with convolution layers. SegNet [1] is an classical encoder-decoder architecture and reuses the pooling indices from the encoder to decrease parameters. DeepLab [2, 3, 4] proposes atrous spatial pyramid pooling (ASPP) for exploiting multi-scale information and then augments the ASPP module with image-level feature to capture longer range information. PSPNet [29] performs spatial pooling at several grid scales and demonstrates excellent performance. ICNet [28] achieves great balance between efficiency and accuracy by using a hierarchical structure to save time on high-resolution feature maps.

As regards RGB-D semantic segmentation, some stud-

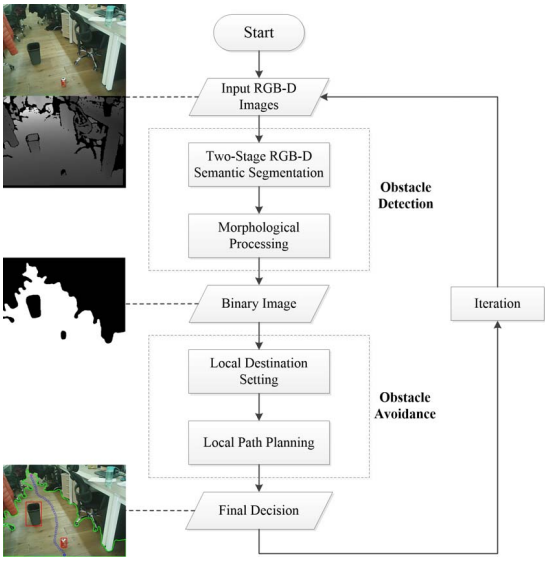


Figure 2. The flow chart of proposed system.

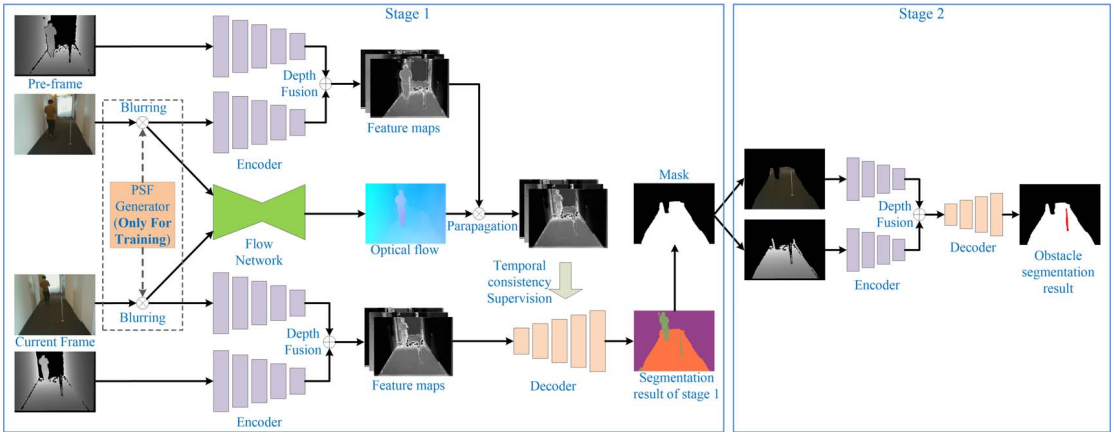


Figure 3. The proposed two-stage RGB-D semantic segmentation network architecture.

ies have tried to utilizing the depth information to achieve better segmentation accuracy [9, 13, 18, 25]. Hazirbas [9] presented a fusion-based CNN architecture which is consisted of two encoder branches for RGB and depth channel. The features of two branches are fused on different layers. Based on the fusion-based encoder-decoder architecture, Jiang [13] applied pyramid supervision training scheme and got a good result.

Most of existing semantic segmentation schemes do not consider the condition of practical applications. For example, the camera often shakes during robot moving and thus captures the image sequence with shaking and/or blurs, which may lead to wrong segmentation results. And in actual application, pixel values and segmentation results of adjacent frames often vary greatly even though the camera and target don't move at all. This is determined by the hardware characteristics of imaging sensor and nonlinearity of deep neural networks. What's more, the original image contains abundant semantic information, making it hard to give specific definition of road and obstacle directly for one-stage segmentation models. In this paper, we adapt the RGB-D fusion-based structure, and introduce optical flow supervision and motion blurring training scheme to improve temporal consistency between adjacent frames. Moreover, a two-stage semantic segmentation architecture is proposed to get accurate semantic result for road and obstacle.

3. Approach

As illustrated in Fig. 2, the proposed obstacle detection and avoidance system contains several steps: RGB-D based two-stage semantic segmentation, morphological processing, local destination setting and path planning. The two-stage semantic segmentation transforms the input RGB-D image to a raw binary image, which is then smoothed in morphological processing. As a result, the module generates a calibrated binary image where every pixel is labelled as either road or obstacle. Then the binary image is passed to the obstacle avoidance module to determine a destination and a walkable path. The whole process works repeatedly during robot's moving.

3.1. Two-Stage RGB-D Semantic Segmentation

The architecture of proposed two-stage RGB-D semantic segmentation is shown in Fig. 3. For the first stage, the RGB-D encoder part of the network is similar with previous work of RedNet [13], which has two convolutional branches. Both of the RGB and depth branches adopt ResNet architecture that with global average pooling layer and fully-convolutional layer removed. Different scale of features are extracted for RGB and depth channel respectively. For each convolutional operation, batch normalization is performed before ReLU function. Feature maps from two branches are fused at each scale. The fusion operation is denoted as $f_{RGB} \oplus f_D$, where f_{RGB} denotes feature maps of the RGB branch and f_D denotes feature maps of the depth branch, $\oplus$ denotes direct element-wise summation. Two consecutive video frames I_p and I_c are input to the encoder at the same time, where I_p and I_c denotes the pre-frame and the current frame. The RGB channel of I_p and I_c are fed to a deep flow network to estimate their optical flow field $F_{c \to p}$ in the first stage.

Semantic information and their spatial location relationship is encoded in the feature maps. Consecutive video frames have highly similarity, so we can propagate feature maps of the pre-frame back to current frame through their flow field [30]. And the propagated feature maps should be also highly similar with feature maps generated by original current frame. In this work, we choose the FlowNet [6] inception architecture which is modified in [30] to meet the tradeoff between accuracy and speed. Because the feature maps have different spatial resolution, the flow field is bilinearly resized to the same resolution of the feature maps for propagation. So, given a position x on a pre-frame fea-

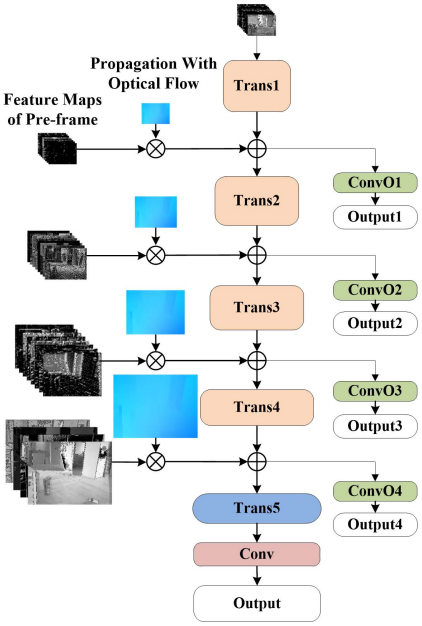


Figure 4. Decoder architecture of the first stage.

ture map f_p, the propagated value can be represented as:

$$f_c(x) = S_{c\to p}(x) \sum_i B(i, x + F_{c\to p}(x)) f_p(i) \quad (1)$$

where B denotes the bilinear interpolation kernel, $S_{c\to p}$ denotes the pixel-wise scale function introduced in [30].

For the decoder, different with the original RedNet, the inputs of skip connection are replaced by the propagated feature maps to keep temporal consistency of segmentation results between adjacent frames. As illustrated in Fig. 4, each scale of feature map, except the last layer of encoder of the I_p branch, is propagated to current frame and fused to the transpose convolution layer of I_c. The fused method is also element-wise summation. Then the fused feature maps are fed into convolution layers with 1×1 kernel size and stride one to generate score maps, *i.e*. output1, output2, output3, and output4. These four side outputs together with the final output are fed into a softmax layer to get their classification score of semantic classes. Loss function for each output is formulated by calculating cross entropy loss of the classification score map.

$$Loss(W_1) = -\frac{1}{N} \sum_i \log \left(\frac{\exp(s_{g_i}(i; W_1))}{\sum_k \exp(s_k(i; W_1))} \right) \quad (2)$$

where W_1 denotes the network parameters of stage one, i is the pixel location, and N denotes the spatial resolution of corresponding output, s denotes the score map, and g_i is the ground truth of class number on location i. The overall loss of the stage one network is calculated by adding the five losses together. For the ground truth has full resolution, it's downsampled to adapt the resolution of side outputs.

After semantic segmentation of stage one, we can extract outer contour of road area by pixel labels. Then the contour *i.e*. ROI is mapped to the original RGB-D image. Pixels out of the contour are set to zeros to eliminate the influence of these object semantic information out of the contour during training phase. It should be noted that non-road pixels within the contour are remain unchanged. The processed ROI of RGB-D image is then fed to the second stage of semantic segmentation. We define this ROI as:

$$Mask_{seg\to RGB-D}(i) = \begin{cases} 1, & \text{if } i \in \text{ROI} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

In the second stage, there are three categories of semantic concept: road, obstacle on the road and others. Objects within road contours are all mapped to obstacle class during training. The last output is:

$$Obstacle_{seg} = W_2(I_c \otimes Mask_{seg\to RGB-D}) \quad (4)$$

where W_2 denotes network parameters of stage two, $Obstacle_{seg}(i) \in C$, $C = \{road, obstacle, others\}$.

The network architecture adopts original RedNet. Considering the tradeoff of speed and accuracy, we use ResNet-34 as the feature extractor in stage two and ResNet-50 in stage one. These two networks are trained individually. After the two stages of semantic segmentation, the predicted class map is converted to a binary image by labelling all road pixels as 1 and non-road (obstacle and others) pixels as 0. It should be noted that, the semantic segmentation results will be presented by their binarized version in the following section, *i.e*. black and white images.

3.2. Motion Blurring for Data Augmentation

In actual application, as the robotic platform walks on the road or camera rotates, information coming from different sub-areas of the scene will move on the detector which will cause image blur. This phenomenon is more serious when the lighting condition is not good enough. In order to suppress segmentation errors introduced by motion blur, random motion blurring scheme is employed in this work for data augmentation. The motion blurring is commonly modeled as in [11]:

$$g(x) = (y \otimes psf)(x) \quad (5)$$

where $y\,(x)$ denotes the original image and $g\,(x)$ denotes the blurred image. psf means point spread function, and $\otimes$ denotes convolution operation. For the exposure time of the proposed system is relatively short, we simplify the motion as uniform linear motion. Then psf can be represented as

$$psf(x, y) = \begin{cases} \frac{1}{L}, \text{if } x = y \tan\theta, \sqrt{x^2 + y^2} \le \frac{L}{2} \\ 0\ , \text{otherwise} \end{cases} \quad (6)$$

where L and θ denote scale and angle of motion.

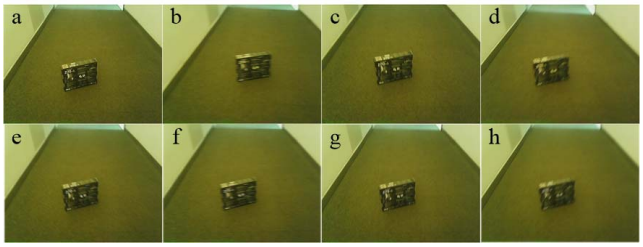


Figure 5. The real clear image (a) and blurred images (b-d) and generated blur images (e-h).

The motion blurring strategy is only implemented during training phase to make training data closer to actual imaging conditions. This data augmentation strategy is beneficial to obtain accurate semantic information when motion blur exists. Fig. 5 shows the comparison of real blurred images and generated images with random motion blur.

3.3. Morphological Processing

We implemented morphological processing to deal with the potential imperfections in the raw binary image. Suppose that the size of binary image is $w \times h$ and all the structuring elements are square. First, the closing of the binary image by a structuring element with size $a_1 \times a_1$ is performed. Then, our system performs an erosion with $a_2 \times a_2$ structuring element followed by a dilation with $a_3 \times a_3$ structuring element. a_i is calculated by the following formula:

$$a_i = f(k_i \cdot \min(w, h)) \quad i = 1, 2, 3 \tag{7}$$

where

$$f(x) = 2 \cdot \left\lfloor \frac{x}{2} + 1 \right\rfloor - 1 \tag{8}$$

The function $f(x)$ finds the closest odd integer to x. Assigning an odd value to a_i makes it easier to define the origin as the center of the structuring element. The selection of k_1 depends on the largest size of obstacle we can tolerate. In our implementation, $k_1 = 1/80$. A smaller k_1 allows the module to detect smaller obstacles but results in the increase of misdetection rate. A larger k_1 filters more misdetections and tiny obstacles out, but those not-so-small obstacles that may cause collision will also be neglected by mistake.

The motivation for performing the following erosion and dilation is to group adjacent obstacles together so that they could be regarded as a single one. There are two reasons for this: 1) Computational complexity is reduced because the number of obstacles decreases; 2) The risk of collision caused by narrow gaps between obstacles is minimized.

The selection of k_2 depends on the maximum distance of obstacles that the method should group, and the value of k_3 is set smaller than k_2 for expanding obstacles in the binary image in consideration of the robot's size. In our experiments, the combination of $k_2 = 1/48$ and $k_3 = 1/64$ shows the best performance.

3.4. Obstacle Avoidance

The obstacle detection module provides the system with a smooth binary obstacle image, which is then passed to the obstacle avoidance module to determine a destination and plan a collision-free path from the current position to the destination.

3.4.1 Local Destination Setting

To avoid obstacles in the present field of view, the local destination is determined firstly. The setting of a destination tries to meet the following requirements: 1) The destination must be a road pixel; 2) The road on the same horizontal line with the destination should be wide enough for the robot to pass; 3) The destination should be as far as possible within the visual range of the robot to indicate the trend of road.

Based on these requirements, we proposed a progressive scanning method. The binary obstacle image is scanned from bottom to top. For each row, the system finds all disjoint road intervals, which are marked as their endpoints' column indices. Suppose that the module is now scanning the i-th row and there are n_i disjoint road intervals marked as (l_{i1}, r_{i1}), (l_{i2}, r_{i2}), , (l_{i,n_i}, r_{i,n_i}). It's satisfied that in the i-th row, all pixels included in these intervals are labeled as road. Conversely, all pixels excluded in these intervals are labeled as obstacle.

We define d_i as the road breadth of the i-th row, which is calculated by:

$$d_i = \max_{1 \le j \le n_i} |r_{ij} - l_{ij}| \tag{9}$$

For the second requirement, we introduce a threshold value $T = \alpha \cdot w$, where α is a coefficient determining the required width of road and w is the width of the binary obstacle image. In our implementation, $\alpha = 1/24$.

With the value of d_i and T, we can determine g_r, the row index of the destination. Recall that h is the height of binary obstacle image and its row index increases from top to bottom, then g_r is calculated as follows:

$$g_r = \min\{r | d_i \ge T, \forall i \in [r, h]\} \tag{10}$$

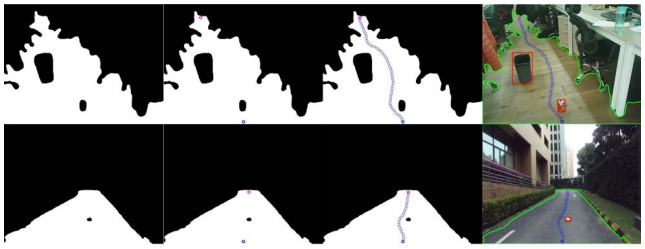
With g_r, the column index of the destination g_c is calculated by:

$$g_c = (l_{g_r,m} + r_{g_r,m})/2 \tag{11}$$

where

$$m = \arg\max_i |r_{g_r,i} - l_{g_r,i}| \tag{12}$$

Finally, the pixel at row g_r and column g_c is determined as the destination, which will be used for path planning in the next section. Fig. 6 gives some examples on destination settings. Furthermore, our destination setting algorithm is suitable for parallel computing thus implemented on GPU for real-time application.



(a) (b) (c) (d)

Figure 6. Path planning based on binary image. (a) Morphological processed image. (b) Destination setting (colored pink). (c) Path planning. (d) Mapping to original image.

3.4.2 Local Path Planning

With the local destination determined in the previous section, the obstacle avoidance module is now prepared to plan a path using Artificial Potential Field (APF) method [14].

APF, first used by Khatib, is an algorithm that can best realize our target of planning collision-free route for robots in real-time. It can construct an artificial potential field, where obstacles have repulsive potential fields repelling the robot, while the designated destination has attractive potential field pulling the robot. Consequently, our robot is under the resultant force and steered towards the destination.

Suppose that there are n obstacles in the binary image named o_1, o_2,..., o_n. The distance and angle between o_i and the robot are marked as d_i and θ_i. The distance and angle between the destination and the robot are marked as d_g and θ_g. And we introduce μ_r and μ_a as the repulsive and attractive scaling factors respectively. Then the repulsive force vector F_r and attractive force vector F_a on the robot is calculated as follows:

$$F_r = \mu_r \sum\nolimits_i \frac{1}{d_i^2} \cdot (\sin\theta_i, \cos\theta_i) \tag{13}$$

$$F_a = \mu_a \frac{1}{d_g^2} \cdot (\sin\theta_g, \cos\theta_g) \tag{14}$$

Therefore, the resultant force F on the robot equals to:

$$F = F_a - F_r \tag{15}$$

In each step, the algorithm calculates the resultant force F that affects the robot's forward direction, then steers the robot to the next position. This procedure will be repeated until the robot reaches its destination. If the path planning module cannot find a collision-free path to the destination, the robot will rotate itself by 15 degrees and reset a destination then try to plan a path to the new destination. Some results of path planning procedure are shown in Fig. 6.

4. Evaluation

In this section, we evaluate the proposed method both on SUN RGB-D dataset and Cityscapes dataset for indoor and outdoor scenarios respectively. Besides, we also evaluation the performance of obstacle avoidance on a new established small obstacle dataset.

SUN RGB-D [22] dataset consists of 10335 indoor RGB-D images with pixel-wise semantic annotations of 37 object classes. It has a default trainval-test split which is composed of 5285 images for training and validation and 5050 images for testing.

Cityscapes [5] is a large-scale dataset for urban scene semantic understanding. It contains footages of street scenes collected from 50 different cities, at a frame rate of 17 fps. The train, validation, and test sets contain 2975, 500, and 1525 footages, respectively. Each footage has 30 frames, where the 20th frame is annotated by 30 semantic classes with pixel-level ground-truth labels.

The new small obstacle dataset is captured in indoor and outdoor scenarios respectively with Orbbec Astra and ZED with a height of about 1.2m. The spatial resolutions of RGB-D are both 640×480. The captured data are pixel-wise labeled as road, obstacle, and others. Besides, for each image, walking routes are labeled by five people. The reasonable ground truth of path plan is obtained by their mean. Samples for indoor and outdoor scenarios are 2200 and 2000. The obstacles we choose are arbitrary objects with random color or shape and with the size ranging from $5cm \times 5cm \times 5cm$ to $50cm \times 50cm \times 50cm$ that may hinder walking of robots, such as trash can, carton, brick, and other small objects easily falls on the road.

4.1. Training Scheme

The key to obstacle avoidance based on semantic segmentation is precise segmentation result of road area. For the two public dataset, there is no concept of obstacle, so their original classes are mapped to a new list in both training and inference of stage one. For the SUN RGB-D dataset, the principles of mapping are similar to the structure class of NYUDv2 dataset [21], floor, wall, window, and ceiling are reserved as scene classes. A few classes that are adjacent to the scene classes and have similar depth with the former are merged, such as floor-mat and floor, or blinds and window. Other classes are divided to furniture and objects. Furniture are large objects that cannot be easily moved, objects are small objects that can be easily carried. For the Cityscapes dataset, the mapping principles are implemented as the original large categories, *i.e.* flat, construction, object, and so on. The new small obstacle dataset is only used during inference.

According to our statistics, most of blur kernel sizes are within scope of 5 pixels. In training, each sample is added by random linear blurring before fed to the model, with a kernel size of 3, 5, or 7 pixels and a blur direction from 0 to ± 180 degree. The original larger images are cropped to 640×480 randomly for the proposed network both in train-

Table 1. Semantic segmentation performance of indoor scenario

Model	$mIoU_1$	$mIoU_2$	ODR	NOFP
SegNet	58.1	–	68.4	9.6
FuseNet	70.2	–	75.7	7.7
RedNet	74.5	–	84.1	4.5
Ours	75.7	98.2	95.2	2.7
Ours(+blur)	76.2	98.5	95.8	2.3
Ours(+blur+flow)	**76.9**	**99.2**	**96.3**	**2.2**

Table 2. Semantic segmentation performance of outdoor scenario

Model	$mIoU_1$	$mIoU_2$	ODR	NOFP
PSPNet	87.8	–	71.4	8.2
DeepLabv3+	89.3	–	77.9	6.7
Ours	91.1	96.4	92.7	5.0
Ours(+blur)	91.5	96.6	92.9	4.7
Ours(+blur+flow)	**92.1**	**97.2**	**93.8**	**4.2**

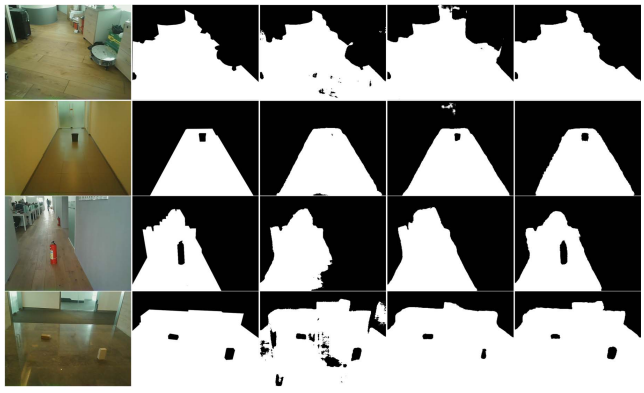
ing and inference. ResNet-50 and ResNet-34 pre-trained on ImageNet object classification dataset are used as the encoder for the two stages. Training for the two stages are performed individually. Images of SUN RGB-D dataset are translated to calculate optical flow with original images in training.

The proposed two-stage semantic segmentation architecture is implemented on the Pytorch deep learning framework. Stochastic gradient descent (SGD) is applied for both of the networks. The initial learning rate is set to 0.002 for stage one, and 0.0002 for stage two. Both the learning rates are decayed by a factor of 0.8 in every 50 epochs. The networks are trained on 4 Nvidia Tesla P100 GPUs with a batch size of 10 until the losses do not further decrease.

4.2. Evaluation Methodology

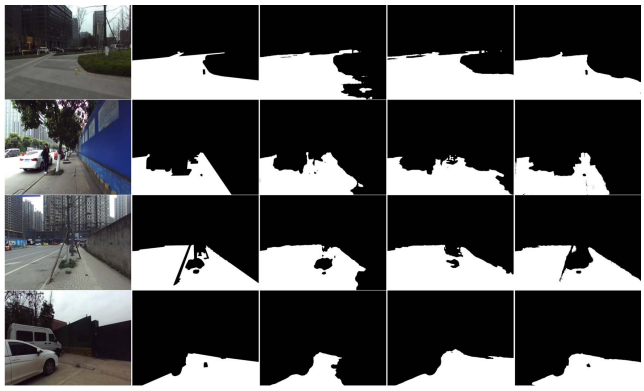
The evaluation consists of two parts: obstacle segmentation and obstacle avoidance. The obstacle segmentation is performed on the default testing set of SUN RGB-D dataset (indoor scenario) and validation set of Cityscapes dataset (outdoor scenario) for the mapped categories of stage one, and on the small obstacle dataset for road/obstacle. The obstacle avoidance is performed on the small obstacle dataset. We evaluate semantic segmentation accuracy on pixel-level, and instance-level for obstacles of stage two. We adopt mean intersection-over-union (mIoU) scorce as the criteria of pixel level. $mIoU_1$ and $mIoU_2$ denote the performance of stage one and stage two respectively. The instance-level accuracy is measured by obstacle detection rate (ODR) and non-obstacle false positives (NOFP). If more than 50% pixels of a predicted obstacle instance are overlapped with the ground truth, it's a success prediction. If there's no pixel of a predicted instance is overlapped with obstacle ground truth, it's a false prediction. ODR is defined as:

$$\text{ODR} = \text{SPI}_{obs}/\text{TI}_{obs} \tag{16}$$



(a) (b) (c) (d) (e)

Figure 7. Obstacle segmentation results of different models. (a) The input image. (b) Ground truth of semantic segmentation. (c) Result of Segnet. (d) Result of RedNet. (e) Result of ours.



(a) (b) (c) (d) (e)

Figure 8. Outdoor obstacle segmentation results of different models. (a) The input image. (b) Ground truth of semantic segmentation. (c) Result of PSPNet. (d) Result of DeepLabv3+. (e) Result of ours.

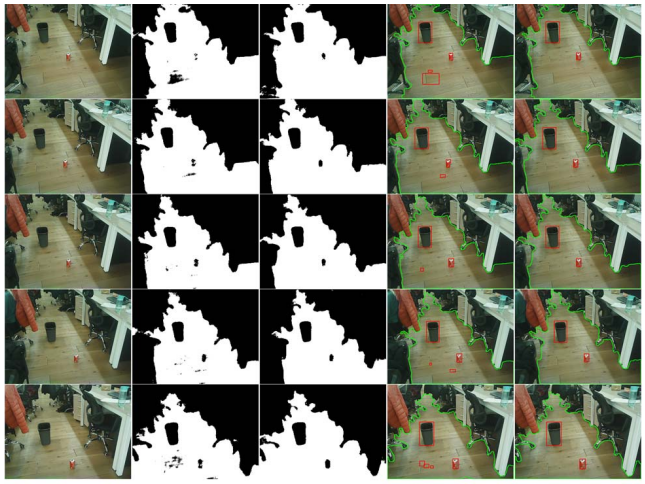
where SPI_{obs} is the quantity of success predictions, TI_{obs} the total obstacle instances. NOFP if defined as:

$$\text{NOFP} = \text{FPI}_{obs}/\text{TF} \tag{17}$$

where FPI_{obs} is the quantity of false prediction instances, and TF the total test frames. The obstacle avoidance quality is measured by Hausdorff distance of planned path with the ground truth path.

4.3. Evaluation of Obstacle Segmentation

Table 1 and 2 shows the performance comparison of our indoor and outdoor models with some state-of-art semantic segmentation based on RGB and RGB-D data. The proposed method outperforms other semantic segmentation architectures both in indoor and outdoor scenarios. After the augmentation of random motion blurring, the mIoU elevates 1.2% and 0.4% in two scenarios respectively. Furthermore, accuracy of instance-level is also improved. Due to the in-



(a) (b) (c) (d) (e)

Figure 9. Segmentation performance of our method on consecutive frames. (a) The input image. (b) and (c) are segmentation result without and with temporal consistency supervision. (d) and (e) are the obstacle detection result of (b) and (c).

Table 3. Path planning accuracy

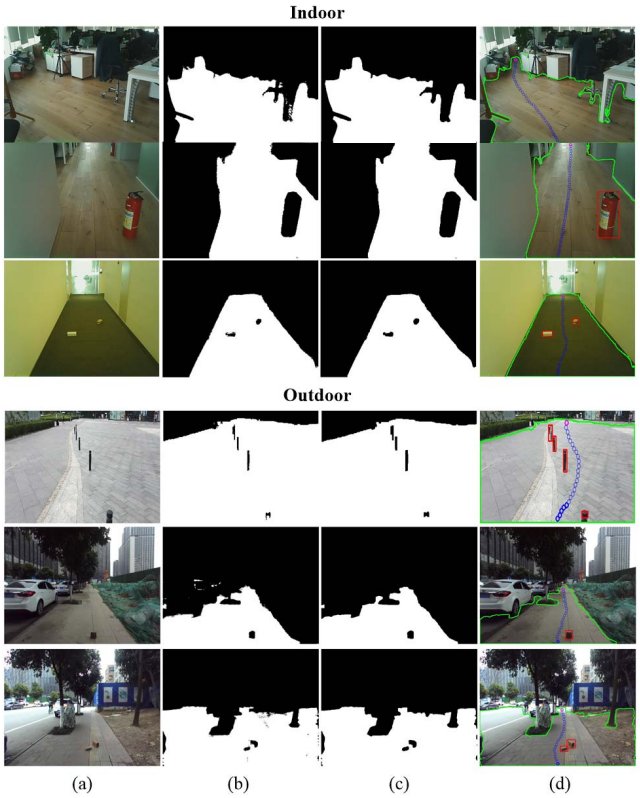
Model	Indoor	Outdoor
Stereo Method [7]	∼0.6m	–
Ours	0.15m	0.27m

troduce of optical flow supervision, the performance of obstacle segmentation is better and stable.

Some obstacle segmentation results are shown in Fig. 7 and 8. As can be seen, obstacles and road areas are both segmented more accurate compared with other architectures, which is crucial to the following obstacle detection and avoidance. Moreover, the proposed method can detect very small obstacles successfully, while other methods fail to. Fig. 9 presents the segmentation performance of our method on consecutive frames with and without optical flow supervision. As can be seen, with the supervision optical flow, the segmentation results are relatively stable, in other words, the temporal consistency is kept.

4.4. Evaluation of Obstacle Avoidance

The obstacle segmentation results are morphological processed to make path planning. Table 3 presents the path planning performance evaluated by Hausdorff distance. The paths made by the proposed method is very close to the ground truth and surpass stereo based method. Fig. 10 shows examples of our obstacle avoidance results. The first column shows the original input color images, the second the binary images calculated by the deep segmentation network, the third the binary images after morphological processing, and the last the result of our system with road area contoured by green line, obstacles marked by red bounding, destination marked by pink circle and path marked by several blue circles. Five thick blue circles, closest to the bottom of the images, indicate five steps calculated in one calculation, while the rest are for demonstration. Above evaluation results indicate the proposed method can effectively detect obstacles and make reasonable path planning for robots in various complicated scenarios.



(a) (b) (c) (d)

Figure 10. Obstacle avoidance results for indoor and outdoor scenarios. (a) image (b) segmentation (c) morphological processing (d) path planning

5. Conclusion

In this study, we proposed a new architecture to automatically create walking routes with RGB-D images for road robots based on deep semantic segmentation neural networks. Two-stage segmentation with motion blurring augmentation and optical flow supervision is presented to acquire more accurate and stable obstacle segmentation, morphological processing is applied to refine the obstacle segmentation, and based on the accurate obstacle map, the local path planning is done to produce the collision-free path. Experimental results show that the proposed method works well under various scenarios whether indoor or outdoor, even with small obstacles or capture blurs. It is the two-stage segmentation that improves small obstacle detection accuracy, and the blurring based data augmentation and optical flow supervision that improves the stability.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[3] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv preprint arXiv:1706.05587*, 2017.

[4] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[5] M. Cordts et al. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[6] A. Dosovitskiy et al. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[7] S. Ghosh and J. Biswas. Joint perception and planning for efficient obstacle avoidance using stereo vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1026–1031, 2017.

[8] K. Gupta, S. A. Javed, V. Gandhi, and K. M. Krishna. Mergenet: A deep net architecture for small obstacle discovery. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5856–5862, 2018.

[9] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian Conference on Computer Vision (ACCV)*, pages 213–228, 2016.

[10] I. Horswill. Visual collision avoidance by segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 87–99, 1995.

[11] J. Jia. Single image motion deblurring using transparency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[12] J. Jiang, Z. Zhang, Y. Huang, and L. Zheng. Incorporating depth into both cnn and crf for indoor semantic segmentation. In *IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 525–530, 2017.

[13] J. Jiang, L. Zheng, F. Luo, and Z. Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. In *arXiv preprint arXiv:1806.01054*, 2018.

[14] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous Robot Vehicles*, pages 396–404. 1986.

[15] M. Kristan, V. S. Kenk, S. Kovačič, and J. Perš. Fast image-based obstacle detection from unmanned surface vehicles. *IEEE Transactions on Cybernetics*, 46(3):641–654, 2016.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[17] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimsou. Visually-guided obstacle avoidance in unstructured environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 373–379, 1997.

[18] L. Ma, J. Stückler, C. Kerl, and D. Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 598–605, 2017.

[19] N. J. Nilsson. Shakey the robot. Technical report, SRI INTERNATIONAL MENLO PARK CA, 1984.

[20] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother. Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1025–1032, 2017.

[21] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pages 746–760, 2012.

[22] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015.

[23] M. A. Turk and M. Marra. Color road segmentation and video obstacle detection. In *Mobile Robots I*, volume 727, pages 136–143, 1987.

[24] I. Ulrich and I. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 866–871, 2000.

[25] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. In *arXiv preprint arXiv:1808.03833*, 2018.

[26] C. Wei, Q. Ge, S. Chattopadhyay, and E. Lobaton. Robust obstacle segmentation based on topological persistence in outdoor traffic scenes. In *IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*, pages 92–99, 2014.

[27] K. Yang et al. Unifying terrain awareness for the visually impaired through real-time semantic segmentation. *Sensors*, 18(5):1506, 2018.

[28] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia. Icnet for real-time semantic segmentation on high-resolution images. In *European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[30] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei. Deep feature flow for video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2349–2358, 2017.