Scalable Learning Framework for Traversable Region Detection Fusing With Appearance and Geometrical Information

Yue Wang, *Member, IEEE*, Yong Liu, *Member, IEEE*, Yiyi Liao, *Member, IEEE*, and Rong Xiong, *Member, IEEE*

***Abstract*— In this paper, we present an online learning framework for traversable region detection fusing both appearance and geometry information. Our framework proposes an appearance classifier supervised by the sparse geometric clues to capture the variation in online data, yielding dense detection result in real time. It provides superior detection performance using appearance information with weak geometric prior and can be further improved with more geometry from external sensors. The learning process is divided into three steps: First, we construct features from the super-pixel level, which reduces the computational cost compared with the pixel level processing. Then we classify the multi-scale super-pixels to vote the label of each pixel. Second, we use weighted extreme learning machine as our classifier to deal with the imbalanced data distribution since the weak geometric prior only initializes the labels in a small region. Finally, we employ the online learning process so that our framework can be adaptive to the changing scenes. Experimental results on three different styles of image sequences, i.e., shadow road, rain sequence, and variational sequence, demonstrate the adaptability, stability, and parameter insensitivity of our weak geometry motivated method. We further demonstrate the performance of learning framework on additional five challenging data sets captured by Kinect V2 and stereo camera, validating the method's effectiveness and efficiency.**

***Index Terms*— Traversable region detection, fundamental mask, multi-scale classification, online learning, dynamic dataset.**

I. Introduction

THE traversable region detection is one of the most fundamental problems in autonomous navigation systems and driver assistance systems. Vision-based traversable region detection has especially gained much attentions in the community of robotics and computer vision. Although a number of approaches have been proposed to solve the traversable region detection,[1] there is still a huge gap between laboratory experiments and real applications due to the large diversity of scenes, the dramatic variation of illuminations and the lack of geometrical constraints.

There are two major limitations of the traditional approaches on traversable region detection, the weak robustness of parameters and the poor universality of models. These problems are revealed in two popular kinds of approaches respectively, i.e. the appearance-oriented approach and the geometry-oriented approach. More specifically, appearance-oriented methods usually build a classifier to identify traversable regions taking the knowledge of appearance characteristics such as textures and colors, which meant to be applied to various scenes without considering the geometrical constraints. However, with the large variation of textures and dramatic changes of illuminations on scene images, the insensitiveness of parameters is relatively weak and the parameters should be carefully selected to satisfy the varied conditions. For geometry-oriented methods, the common ways are to estimate the vanishing points and parallel edges from images, or estimate the plane from depth data to extract traversable region. Although it is a non-parametric method with no need of training classifies, their performances highly rely on the geometrical conditions of the environment and the quality of the available geometric data.

To address the two limitations of traditional approaches, this paper explores a unified detection model taking advantage of both appearance and geometry information, which aims for robustly detecting traversable regions with insensitive parameters against varied conditions. To loosen the strict constraints required by geometry-oriented methods, the basic framework of our method is to learn a predictive model guided by the target derived from the geometry using the appearance-oriented features, and this framework can also be extended to fusion with richer geometrical information for further promotion.

To enhance the adaptiveness of the model and promote the parameters insensitiveness of our approach, we construct an online learning framework to update the training dataset after receiving each incoming frame, the corresponding classifier is then updated as well. Instead of training with prior labeled instances or co-training with partially labeled instances [1], our model does not require manually labeled images for training as we assume that the front area of the robot can be safely passable. Integrating the previous windowed information, the assumption [2] is self-validated and can be further refined with more geometrical information from the common sensors equipped on the mobile platforms like depth sensor or stereo camera. With the initialized label information based on the geometric assumption, the weighted Extreme

Manuscript received April 26, 2016; revised January 20, 2017; accepted March 7, 2017. Date of publication April 3, 2017; date of current version December 7, 2017. This work was supported by the National Natural Science Foundation of China under Grant U1509210 and Grant U1609210. The Associate Editor for this paper was J. M. Alvarez. *(Corresponding author: Yong Liu.)*

The authors are with the Department of Control Science, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China (e-mail: wangyue@iipc.zju.edu.cn; cckaffe@hotmail.com; yyliao@iipc.zju.edu.cn; rxiong@iipc.zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2682218

[1] In this paper, traversable region is not limited to the highway with parallel edges and other structural geometrical characteristics.

Learning Machine (ELM) [3], which can be trained in closed-form solution and thus avoids the time-consuming iterative optimization, is used as our classifier. By the weighted ELM, our classifier not only supports the imbalanced class distribution, but also allows reducing the effects of those historical training samples.

The contributions of our approach can be concluded as follows:

- We construct an online learning framework that achieves effective ensembles of both the appearance information and the geometrical information.
- Our approach implements an empirical geometry motivated assumption which can be satisfied in most cases, thus our approach can also avoid the requirement of pre-labeled training images by that weak assumption, which is also able to be conveniently refined with richer geometrical information.
- Our traversable region detection framework is trained in online process. The training samples and classifier will be updated after each new coming frame, so our method can be insensitive to the initial parameters. The method can also be adapted (or robust) to the new emerging scenes under varied changes efficiently, without manually resetting of the parameters.
- Taking the weight ELM as the classifier, our model can sustain varied initializations with the imbalance distributions on classes, and is also easy to support the weighting of historical samples in real-time.

II. Related Works and Basic Idea

A. Related Works

Generally speaking, the detection of traversable region is based on two kinds of information, i.e. appearance information and geometrical information. A number of traversable region detection methods have been proposed based on these two kinds of information, which are named as appearance-oriented method and geometry-oriented method in this paper.

In geometry-oriented methods [4]–[7], a common way is detecting the vanishing point as well as estimating the road edges. To obtain more stable detection results, extending methods [8]–[10] are proposed to incorporate more complex geometry models, e.g. passable region's geometrical model. However, the performances of these methods will degenerate severely once these geometric constraints, such as vanishing points, side edges etc., are not satisfied in the image. Aside from detecting points and lines from monocular images, stereo vision is also considered in the traversable detection problem for estimation of ground plane using the homography matrix [11], [12]. Kinect v2 is also verified to be effective in outdoor environment [13], [14] and could be employed for plane estimation. As the estimation of ground plane assumes that the traversable region is flat, the accuracy of these methods will decreases when the environment changing from urban area to rural space.

The appearance-oriented methods have also been intensively explored to address the traversable region detection [2], [15]–[19]. Some early works detected the traversable region directly by training a classifier based on color features [15] or the combination of color and texture features [16]. Rather than classification with only color and texture information, the appearance feature based probabilistic models, e.g. the road density probability model in pixel space [17] and super-pixel based Gaussian model [2] etc., are also used to represent the region's traversable possibility, these models are employed as the initializations of the further classification. However, the parameters of probabilistic models are sensitive to changeable environments, which might limit their adaptiveness.

In recent years, combining of the appearance information and the geometrical information has attracted much attention. A typical work presented by Dahlkamp *et al.* [20] obtained the nearby passable region with a hybrid of the laser and RGB feature modeling with a mixture of Gaussians, the far region can be predicted by this model and the model will be updated once the environment changes. A recent work [21], which built a classifier based on monocular image using the appearance information, it then estimated the ground plan from the consecutive frames as the initialization of the model. Combining the appearance information and the geometrical information to detect the traversable region is also close to the cognition processing of human beings, as we human beings detect the traversable region according to both appearance and disparity. More generally, Lee *et al.* [22] used both geometric and appearance information to segment planar building facades (PBF) and uses geometric constraints to refine the 3D PBF mapping. Inspired by this issue, we improve our previous appearance-oriented approach [19] to an appearance-oriented and geometry-aided (or geometric constrained) traversable region detection framework, which can provide stable performances with only appearance and also can be conveniently extended with additional geometry information.

To enhance the model robustness to the illumination variance, some approaches [23]–[25] have be proposed to implement illumination-invariant feature transforming and then obtain the traversable region in that transformed feature space. However, this approach may be sensitive to the over-exposed or under-exposed [24] in the single RGB channel. We demonstrate that our online learning approach is able to fast adapt to the illumination changes without transform of input space.

B. Fundamental Assumption in Our Approach

As shown in figure 1(a), in our method, we adopt a weak geometric assumption that regards the small area in front of the robot as passable and two small areas on the top of the image as impassable. At the perspective of a temporal window, the passable areas in the historical images have been actually traversed through by the robot at the current instant, therefore this assumption is self-validated with probably very limited mistakes. To model the assumption, a Fundamental Mask (FM) is proposed, shown in figure 1(b).

We then formally define the traversable region detection as a binary classification problem, where each pixel is represented as passable or impassable . Considering the local consistency of the labels, we segment the image into multiple sub-regions

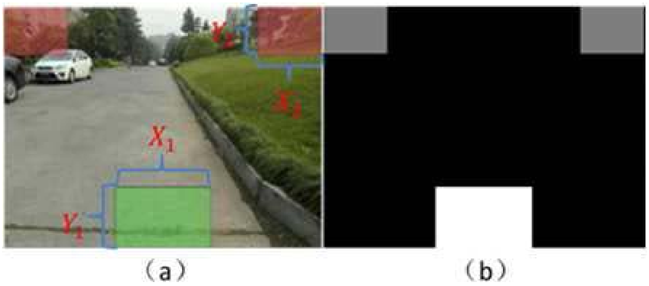


Fig. 1. A sample of our weak assumption. (a) the bottom middle green region is assumed traversable, the up left and right red regions are assumed impassable. (b) Fundamental mask generated from the weak assumption, the traversable, impassable and unknown regions are denoted as white, gray and black respectively.

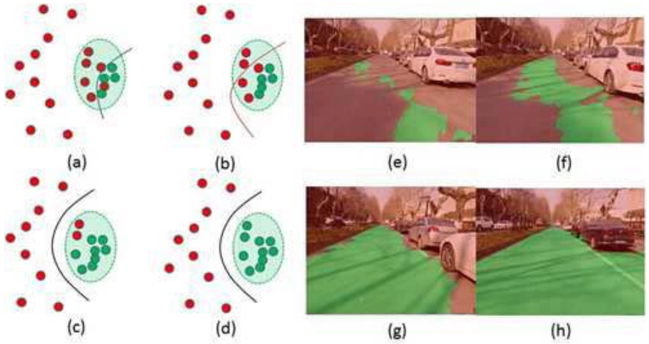


Fig. 2. A sample online training processing on the sequenced images. sub-figure (a)-(d) show classification boundaries and labels for each instance corresponding to real image sequence (e)-(h). With the varied misclassification cost functions, the boundary will be pushed toward the lower risk negative region during the iterative training.

with the super-pixel segmentation, and then each image can be represented as the following set:

$$\{(f_1, L_1), (f_2, L_2), \cdots, (f_i, L_i), \cdots, (f_n, L_n)\}$$

where n is the number of the super-pixels, $i = 1 \cdots n$, $f_i = [f_{i1}, f_{i2}, \cdots, f_{iD}]$ is the ith sub-region's feature vector with D dimensions, and $L_i \in \{-1, +1\}$, which labels f_i, $+1$ for traversable region, -1 for impassable region. With the Fundamental mask, the traversable region detection task can be formulated as to train a separation boundary which can correctly label the unknown sub-regions based on those traversable and impassable regions. Those unknown sub-regions are all denoted as impassable for the safety consideration in the initialization. As a result, the training set contains much more negative samples than the positive samples at the beginning. So we introduce imbalanced ELM method [3], [26], [27], which uses varied costs for different categories. The adaptive online training model is also implemented in our approach. In each frame, we will update the training dataset and retrain the classifier based on the classification results of previous frame. With the imbalanced ELM method and the adaptive online training model, the classifier can achieve a dynamic balance. For the condition that there are much more negative instances, we will set a larger misclassification cost for positive instances. Then the separation boundary will be pushed toward the negative instances in the online training process until achieving a balance, of which the process is shown in figure 2.

III. Fundamental Learning Framework for Traversable Region Detection With Appearance Information

There are three main steps in our traversable region detection approach, shown in figure 3, i.e. feature construction, classification in multi-scale and online training with dynamic dataset, in addition with a configurable initializer for sub-regions' labels. The feature space is constructed on the appearance clues including texture and color, which are also employed in previous works [28], [29], formulating the input space of the subsequent modules. While the labels' initialization module, which formulates the label space of the classifier, is investigated using the partial/sparse geometric clues, introduced by fundamental mask (FM), stereo vision and RGBD sensors which is Kinect V2 in this paper. Therefore the core idea of the fusion in the proposed framework is to develop an appearance classifier to follow the geometric indications, but capture more intrinsic variance with dense data and yield dense detection result, through the online learning.

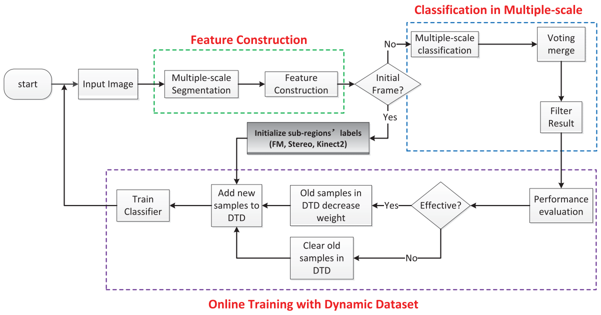

Classification in Multiple-scale
Feature Construction
start
Input Image
Multiple-scale Segmentation
Feature Construction
Initial Frame?
No
Yes
Multiple-scale classification
Voting merge
Filter Result
Initialize sub-regions' labels (FM, Stereo, Kinect2)
Train Classifier
Add new samples to DTD
Old samples in DTD decrease weight
Yes
Effective?
Performance evaluation
Clear old samples in DTD
No
Online Training with Dynamic Dataset

Fig. 3. The framework of our traversable region detection approach.

A. Discriminative Feature Construction

Compared with pixel based feature extraction, features constructed from super-pixel are based on the statistical information from a couple of closed pixels and thus more stable. It also reduces the computation complexity due to the much less amount of samples, and may achieve real-time performance.

To achieve real-time detection, the image is segmented with SLIC [30]. And each super-pixel is corresponding to a feature vector constructing by its color and texture distribution. The color based sub-vectors come from the HSV histogram and the texture based sub-vectors come from the uniform LBP [31] descriptor. The sub-vectors from both the color channels and the LBP descriptor are then concatenated and normalized as the feature vector for each super-pixel. More specifically, the color spaces of H channel and S channel are divided into 18 intervals, and the V channel is divided into 9 intervals, thus the dimension of the HSV's feature vector is 45. The uniform LBP [31] feature vector is encoded, with 8 pixel neighborhoods on a circle of radius of 1, which means the

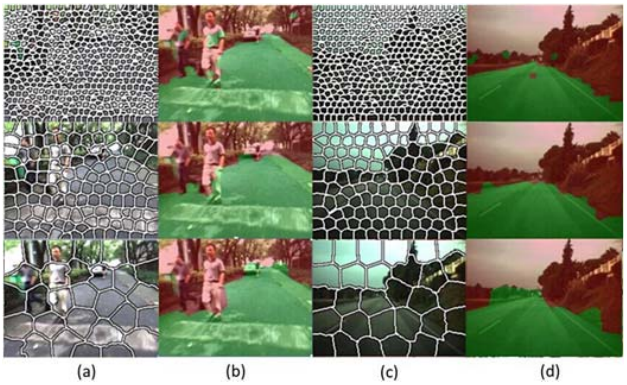


Fig. 4. Two sample images and their labeling results under three different segmentation scales. The passable regions are denoted with green color.

dimension of the LBP feature vector is 10. Then the total dimension of the super-pixel's feature vector is 55.

B. Multi-Scale Super-Pixels Based Classification

As shown in figure 4, the super-pixel with smaller size contains more detailed information but are more sensitive to noises, while the super-pixel with large size can be robust to noises but lacks of details. Then an ensemble of multiple-scale super-pixels based classification is desired.

Specifically, we segment the image into θ (θ is an odd number) scales, where there are M_i super-pixels in the ith scale, for each scale

$$M_{i+1} < M_i <, \cdots, < M_1$$

Then we define K as the amount ratio between two adjacent scales.

$$K = \frac{M_i}{M_{i+1}}$$

K can be regarded as the size gap between two adjacent scales and is used to control the sample size of the super-pixels in each scale. By setting an appropriate K, the super-pixel in multiple-scales can be robust to the noises as well as preserving the details.

Lets denote the feature vector of the mth super-pixel in ith layer (scale) as f_i^m. We train a classifier at each layer to get f_i^m's label L_i^m. Once the mth super-pixel in ith level is labelled as L_i^m, all the pixels in that super-pixel will be given the same labels L_i^m. Based on the labeled results of all the layers' super-pixels, we can get θ binary labeled maps of the image in pixel level.

The results of θ layers can be denoted as $R = [r_1, r_2, r_3, \cdots, r_\theta]$, where $r_i(h \times w)$ is the ith labeled map of the image contains only $+1$ and -1. Then the fused results can be obtained by merging all the levels as follow:

$$R_{sum} = sign(\sum_{i=1}^{\theta} r_i)$$

R_{sum} is the results fusing with all the levels' labels in the same pixel. As the result of R_{sum} is fused from the decisions in multiple scales, it can provide accurate edges and details via those small scale super-pixels while improve the robust capability against the noises via the large scale super-pixels.

The labeled results after fusing with all layers may still contain some noises or mislabeled patches due to the similarity of the colors and textures in the complex environment, we then use a post-filter processing to delete those isolated incorrect small patches. The online training will employ the detection results after post-filtering, thus the training data will contain less mislabeled samples and can converge faster.

C. Online Training Model

As the scenes change continuously during the robot navigation, we design an online training model, which is trained with the classification results of those recent previous image sequences, to overcome the problem of scenes changing. As the data in the training model are updated online, we call the training dataset as dynamic training database (DTD) [17], and the classifier will be retrained once the DTD is updated.

To obtain the real-time performance in our online training model, the weighted ELM (extreme learning machine) [3] is used as our classifier since the weighted ELM can be trained efficiently with its closed-form solution. Furthermore, the weighted ELM can also be helpful to handle the imbalanced instances in our problem as it minimizes the weighted cost function for different classes and maximize the margin for the boundary. For the feature f_i, $i = 1, \cdots, N$ and its label L_i, we use diagonal matrix W ($W \in N \times N$) to represent the weight, then f_i' weight is w_{ii}. Then we can minimize the following cost function:

$$L_{WELM} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{N} w_{ii}\|\xi_i\|^2$$

Subject to:

$$h(f_i)\beta = L_i - \xi_i, \quad i = 1, \cdots, N$$

Here $h(f_i)$ is the nonlinear representation of the original feature which can be randomly generated, and β is the output parameter of the classifier, ξ_i is f_i's misclassification cost. Then the β can be quickly calculated by thoore-Penrose "generalized" inverse. And C is a constant to adjust the weight between the margin maximization and cost function (training errors).

In the weighted ELM, we set a large weight for the feature vector of the minority class. Assuming there are n_{pos} passable regions and n_{neg} impassable regions, we then calculate the weights as follow.

$$\begin{aligned} C_d &= \frac{(n_{pos} - n_{neg})}{N} \in (-1, +1) \\ C_b &= \lambda sign(C_d)|C_d|^\sigma \\ w_{ii} &= w_0 - sign(L_i)(C_b - \phi) \end{aligned}$$

Here C_d denotes the different ratio between two categories, λ and σ are two constant parameters. If $\sigma > 1$, w_{ii} changes slightly when the numbers of two classes are closed. We use ϕ as the threshold to prioritize the passable region, so the classifier will tend to label a region as passable when it closes to the separation boundary. w_0 is the initial weight and can be set to a constant. Besides, the w_{ii} is always set as non-negative with appropriate λ and ϕ.

In our online learning model, the classification is executed in each image, then we evaluate the new labeled data and use them to update the DTD. Similar to [17], the item s_i in DTD is consisted with three elements that are feature vector, label and weight, and can be denoted as:

$$s_i = [f_i, L_i, w_{ii}]$$

Before we can add the classification results into the DTD, a confidence evaluation should be carried out based on the weak assumption. Specifically, we measure the accuracy ratio for the super-pixels lying in the FM as:

$$Acc = \frac{N_{WTP} + N_{GFN}}{N_{FM}}$$

where N_{FM} is the number of super-pixels lying in the FM on all the layers, N_{WTP} is the number of true positive super-pixels contained in the bottom middle region (passable region in FM), and N_{GFN} is the number of true negative super-pixels contained in the up left and right regions (impassable region in FM).

In our approach, the DTD of current image can be added to our training set only when $Acc > 0.9$, otherwise, the previous classifier is regarded out of date due to the scene changing, and all the items in the current training set will be discarded, a new classifier will be retrained based on the FM.

When adding to the DTD, the new samples' weight W is also computed based on their distribution on two classes. And we also decrease the pre-existing samples' weight as follow.

$$\hat{W} = W - \Delta W$$

where W is the weight matrix before updating and $\hat{W}$ is the updated weight matrix, ΔW denotes a constant decrement matrix. With the frame increasing, the previous samples' weights on training errors will be also decreased. This is reasonable as the newest samples will be more valuable to the next classification comparing with the pre-existing samples when the scene is changing. Once the weights of the samples reach zero or less than zero, they will be removed from the DTD. Then the DTD can be maintained to a limited scale, which allows real-time calculation and also keeps the reasonable generalization capability.

IV. Fusion With Information From Multiple Sensors

Although the assumption of *FM* can handle most of the conditions in traversable region detection only with the appearance information from the monocular vision and an empirical geometric prior, it may suffer from the inconsistent color and texture features distributed in varied sub-regions of the traversable region. These inconsistent distributions will lead to insufficient sampling in the previous learning framework, when initializing the sub-regions' labels with the *FM*. Thus some traversable sub-regions, which are not correctly sampled in the *FM*, may be classified incorrectly.

Fortunately, our framework intrinsically supports the integration of the external geometrical information, which can introduce additional discriminative information to solve the problem of inconsistent distribution in color and texture features. The external geometrical information can be easily obtained from multiple sensors such as stereo vision and depth sensors, where these sensors are common configurations on todays mobile platform. By utilizing these multiple sensors to obtain additional geometrical information, we can fuse the 3D information to initialize the labels of the sub-regions, which is able to refine the *FM* used in the monocular vision system. With those enhanced labeled samples fusing from multiple sensors, our learning framework can overcome the problem of inconsistent distribution in color and texture features. The following section will present two approaches fusing with stereo vision system and Kinect V2 system respectively.

A. Fusion With Stereo Vision

The stereo camera is a common sensor amounted on the mobile robots, it can obtain the disparity map from the two views with stereo matching algorithms [32], then the corresponding dense depth information can be constructed. Although the nearby depth information obtained by the stereo vision system can be quite accurate, the accuracy for far depth information is quite unstable and there are also many unmatched regions which are represented as black holes in the disparity map. An intuitive approach to detect the traversable region with the depth information is to extract the largest plane from the current view by RANSAC algorithm [33]. However, the results show that the stereo vision system does fail to recognize the traversable region in the far area due to the inaccurate or missing depth information in the far area. Therefore, the available partial depth information is utilized to initialize the labels.

We present a stereo vision based mask generation algorithm as follows:

Algorithm 1 Stereo Vision Based Mask Generation Algorithm

```
Input: Image $I_L$, $I_R$, distance threshold $\tilde{d}$
Output: Traversable 3D point set $M$ and impassable
        3D point set $\bar{M}$
Applying dense stereo matching for left image $I_L$ and
  right image $I_R$;
Calculate the 3D coordinate $p(x, y, z)$ for each matched
  pairs (points) in the disparity map $I_D$;
Extract the maximal plane $Ax + By + Cz + D = 0$ from
  all the 3D points with RANSAC;
for each $p_i(x_i, y_i, z_i) \in I_D$ do
    if $|Ax_i + By_i + Cz_i + D| < \tilde{d}$ then
        $M = M \cup \{p_i\}$;
    end
    else
        $\bar{M} = \bar{M} \cup \{p_i\}$;
    end
end
```

The algorithm 1 classifies the 3D points in the disparity map I_D into two categories, i.e. traversable point set M (green

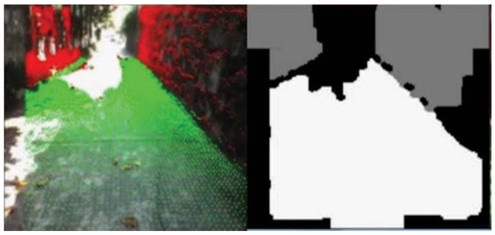


Fig. 5. The traversable region obtained by stereo vision system (left) and its corresponding mask (right).

points in figures) and impassable point set $\bar{M}$ (red points in figures). And if the distance from 3D point p_i to the extract plane is less than the distance threshold $\tilde{d}$, then p_i is categorized to M, otherwise p_i is categorized to $\bar{M}$.

After obtaining the M and $\bar{M}$, we re-project all the 3D points into the 2D images and construct an image only consisting of three kinds of points, i.e. traversable region points, impassable region points and unknown region points. We then implement an image expansion operation in the constructed image to connect those isolated points into regions, following by an image corrosion operation to obtain the mask. The final mask is then obtained by overlapping both the fundamental mask and the mask generated by the stereo vision system, the example is shown in figure 5.

B. Fusion With Kinect-2

The Kinect-2 depth sensor is based on the time-of-flight measurement principle. A strobed infrared light illuminates the scene, the light is reflected by obstacles, and the time of flight for each pixel is registered by the infrared camera. Internally, wave modulation and phase detection is used to estimate the distance to obstacles [34].

The Kinect-2 provides a big improvement over the original Kinect for ourdoor/sunlight situations. Whereas the original version is not suited to outdoor usage, the Kinect-2 can measure depth at ranges below 2m [13]. In [14], it has also been shown that the Kinect v2 is able to capture data for shadow and direct sunlight situations.

The Kinect-2 features a higher resolution of 512 × 424 pixels. In our practical outdoor experiments, we found that ranges between 1.0 to 2.8m and the pixels $(u, v), u \in [100, 412], v \in [270, 424]$, could reliably be measured, whereas measurements outside this range were considered as unreliable and therefore omitted from the traversable region mask model fitting. We denote the 3D point set corresponding to the valid depth pixels as S, and then the Kinect-2 mask generation algorithm can be constructed similar to the stereo vision system as follows.

Algorithm 2 Kinect-2 Based Mask Generation Algorithm

```
Input: Valid depth point patch $S$, $k$, angle threshold $\tilde{\theta}$,
       distance threshold $\tilde{d}$
Output: Traversable 3D point set $M$ and impassable 3D
        point set $\bar{M}$
1  Divide $S$ into $k$ adjacent sub-regions, $S_1, S_2, ..., S_k$, along
   the horizontal coordinate of the depth image from bottom
   to top of $S$, the bottom is $S_1$ and the top is $S_k$;
2  for each $S_i$ do
3      Fitting a maximal plane, $A_i x + B_i y + C_i z + D = 0$,
       and its corresponding normal vector $\vec{n_i}$ with
       RANSAC;
4  end
5  for $i$=1 to $k$ do
6      if $i$==1 or $\theta(\vec{n_i}, \vec{n_{i-1}}) < \tilde{\theta}$ then
7          for each point $p_j(x_j, y_j, z_j) \in S_i$ do
8              if $|A_i x_j + B_i y_j + C_i z_j + Di| < \tilde{d}$ then
9                  $M = M \cup \{p_i\}$;
10             end
11             else
12                 $\bar{M} = \bar{M} \cup \{p_i\}$;
13             end
14         end
15     end
16     else
17         for each point $p_j(x_j, y_j, z_j) \in S_i$ do
18             if $|A_1 x_j + B_1 y_j + C_1 z_j + D1| < \tilde{d}$ then
19                 $M = M \cup \{p_i\}$;
20             end
21             else
22                 $\bar{M} = \bar{M} \cup \{p_i\}$;
23             end
24         end
25     end
26 end
```

Different with the mask generation algorithm 1, the algorithm 2 used for Kinect-2 does not fit only one plane as the Kinect-2 cannot provide consistent depth data in outdoor environments. We try to fit k planes for the valid depth data patch in algorithm 2 in step 2-4, and regard the maximal plane in S_1, which is the bottom of the valid patch, as the traversable region. If the adjacent planes have an angle less the threshold $\tilde{\theta}$ (step 6), the algorithm 2 will use its own plane formula to label the points (step 7-12), otherwise the algorithm 2 will use the S_1's plane formula to label the points (step 17-23). The examples of the algorithm 2's results are shown in figure 6.

After obtaining the M and $\bar{M}$, we also need to re-project all the 3D points into the 2D images[2] and construct an image only consisting of three kinds of points, i.e. traversable region points, impassable region points and unknown region points. Following the same post-process methods as we use with the stereo vision, image expansion and image corrosion are applied to refine the mask. The final mask is also the overlap of both the fundamental mask and the mask generated by the Kinect-2 system.

V. Experiments

In this section, we carry out several experiments to evaluate our proposed approaches. To evaluate the effectiveness of

[2] The 2D image captured by Kinect-2 will be down-sampling to the resolution of the depth image.

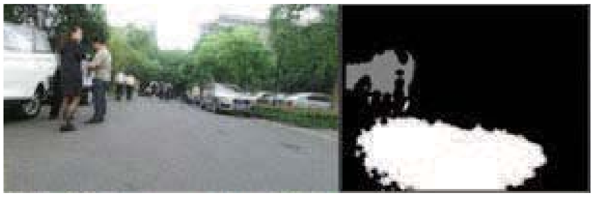


Fig. 6. The sample of the output results of algorithm 2, left sides are the color images and the right sides are the corresponding mask generated by algorithm 2, the white areas are the traversable regions, the grey areas are impassable regions, and the black areas are unknown regions.

the proposed framework, we make comparisons with three state-of-arts methods as follows:

A. Growcut

The first method is proposed by Lu *et al.* [2]. This algorithm assumes an elliptic area in front of the robot is traversable and uses GMM to model this region and a super-pixel level probability density map is built based on this model. Then the super-pixel level is used to select seeds of grow-cut, which is then used to label the image. However, this method may be sensitive to scene variation, and parameters need to be tuned when the scene changes. In our experiment, the optimal parameters for each dataset are obtained by an exhaustive search in the *Growcut* method.

B. VP

The second method is proposed by Kong *et al.* [4]. This method uses a voting strategy to select the vanishing point which is regarded as a strong cue to detect the edges of road. These edges are then used to update prior vanishing point. This method can work well on scenes have obvious vanishing point and the straight road edges. However, crossed and curved roads will affect the performance significantly. The code used in our experiment is downloaded from Kong's website.[3]

C. Gaussian

The third method is proposed by Dahlkamp *et al.* [20]. In this method, laser is used to detect nearby traversable region and a set of Gaussians are used to model the region in RGB color space. In the labeling process, the nearest Mahalanobis distance between Gaussians and sample pixel is set to decide whether the sample belongs to traversable regions or not. As the three datasets used in our experiments have no laser data, we use the assumption that the front of the robot is traversable to extract the information from those assumed traversable regions.

In the following experiments, all our approaches use the same parameters setting in all the datasets. We set the total amount of layer $\theta = 3$ and the amount ratio $K = 5$. The parameters in the ELM are assigned with $C = 1$, $w_0 = 1$, $\lambda = 0.8$, $\sigma = 3$ and $\phi = 0.05$.

Three pixel-wise quantitative metrics [35], [36], i.e. *FPR* (false positive rate), *FNR* (false negative rate) and *ErrorRate*, are implemented to evaluate the accuracy of the detection:

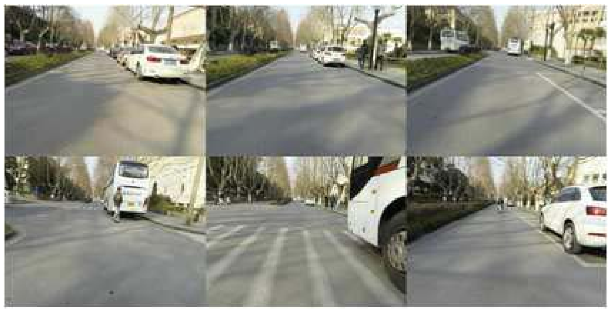


Fig. 7. *Shadow road* dataset containing shadows in the traversable regions. This dataset is a sequence containing 135 image frames, where all frames are labeled manually. It is captured by a monocular GoPro camera mounted on the robot with a frame rate of 5HZ. And the scene is a road lined with trees in our campus. There are many shadows of trees in sunny days.

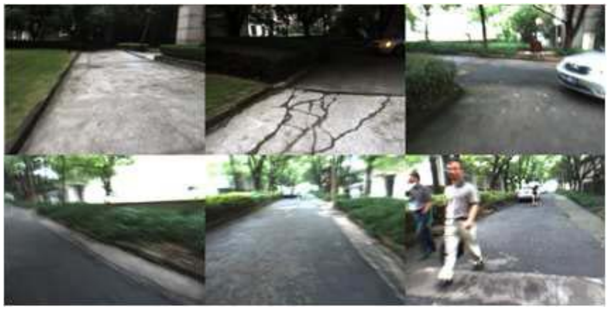


Fig. 8. *Variational road* dataset containing obviously texture and color changes during the sequent image frames. This dataset contains a sequence of 253 images, where 125 frames are labeled. It is captured by a Bumblebee stereo camera mounted on a four wheeled robot. In our experiments, we only use the image sequences from right camera. The scene is also a road in our campus which contains many challenges such as varied significant texture and color changes during the image sequence, image blur, barriers caused by moving pedestrians or vehicles, and varied illumination etc.

$$FPR = \frac{N_{FP}}{N_P} \times 100\% \quad FNR = \frac{N_{FN}}{N_N} \times 100\%$$

$$ErrorRate = \frac{N_{FP} + N_{FN}}{N_P + N_N} \times 100\%$$

Here N_{FP} and N_{FN} are the pixels being wrongly classified as passable or impassable regions respectively; N_P and N_N are the ground-truth pixels of passible and impassable regions respectively. We then can evaluate the performance with the average metrics on all the frame sequences of each dataset. Besides, the computational speed is also compared to evaluate the possibility of real-time applications.

1) Experiments on Monocular Camera: We employ three challenging datasets, *rain sequence* dataset,[4] *shadow road* and *variational road*, to evaluate our proposed basic monocular method (abbreviated as *LFTD* in the following experiments), *shadow road* and *variational road* shown in figure 7 and figure 8, are captured by our own vision system.[5]

These datasets cover typical daily life scenes. The *rain sequence* dataset consists of consistent appearances on the scenes, there are vanishing point and edges in almost

[3] http://web.mit.edu/huikong/www/index.htm

[4] https://rsu.forge.nicta.com.au/people/jalvarez/research_bbdd.php.

[5] Data available at http://www.csc.zju.edu.cn/yliu/index.html

TABLE I

QUANTITATIVE PERFORMANCE METRICS OF FOUR METHODS IN THREE VARIED DATASETS

	Metrics	*Growcut*	*VP*	*Gaussian*	*LFTD*
Shadow Road	ErrorRate	6.6	12.3	4.66	**3.93**
	FPR	4.09	2.48	4.61	**0.85**
	FNR	9.88	25.73	**4.7**	7.87
Rain Sequence	ErrorRate	7.2	**4.14**	12.36	5.46
	FPR	3.29	7.81	**0.99**	1.64
	FNR	9.61	**1.96**	19.21	7.77
Variational Road	ErrorRate	11.66	12.14	15.21	**7.8**
	FPR	13.62	9.64	22.51	**2.1**
	FNR	14.51	16.62	**13.08**	14.92
Overall performance	ErrorRate	8.19	8.55	10.91	**5.63**
	FPR	6.21	6.77	7.64	**1.53**
	FNR	10.97	12.55	13.47	**9.67**

every frames. The *shadow road* dataset has some shadows caused by the trees, then the shadows may lead to confusions during the edge detection or model based road representation. *Variational* road is a challenging dataset containing image blur, barriers, varied illumination and significant texture-color changes on road surfaces. In our experiments, we choose those three datasets with varied styles to evaluate the robustness and adaption our method, as our method uses the same parameters on all those three varied datasets.

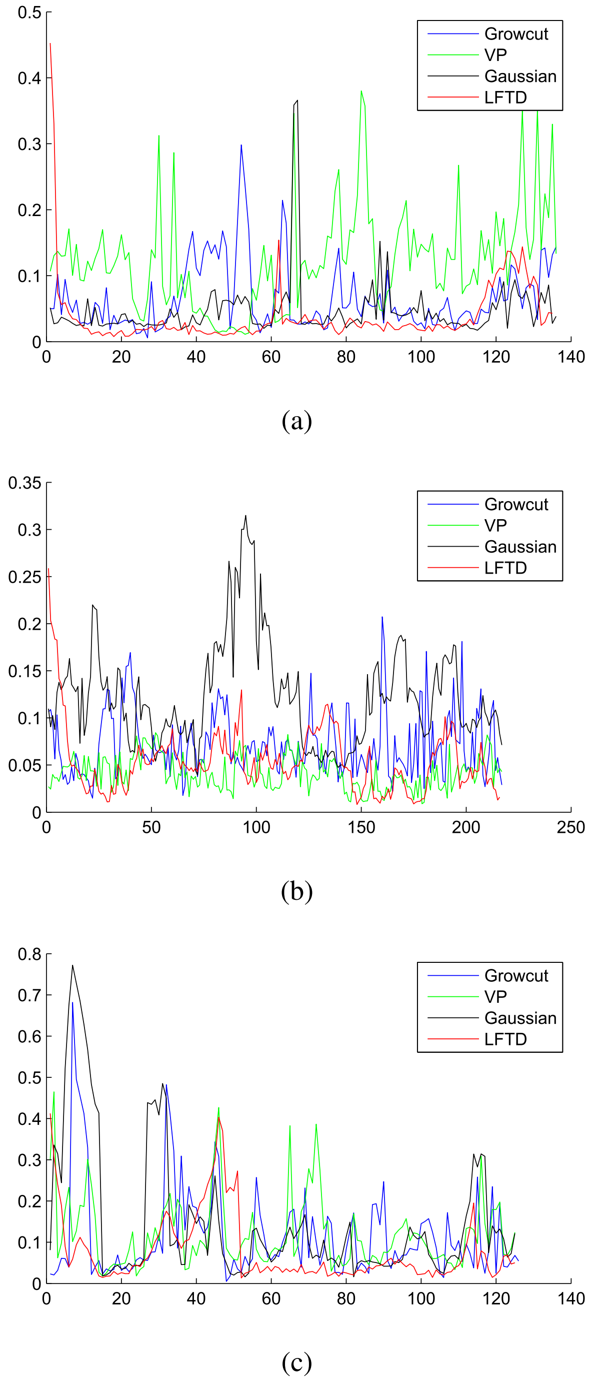
There are three state-of-the art approaches, i.e. *VP*, *Gaussian*, and *Growcut*, comparing with our *LFTD* in the experiments.

Table I gives the four methods' experimental results on three varied datasets. The overall performance is calculated as the weighted average on three datasets based on the number of frame in each dataset.

The experimental results in Table I show that *LFTD* can achieve better performances on all the datasets comparing with other algorithms. As our *LFTD* employs the same parameters in all the three datasets, which also indicates the adaptability and robustness of our *LFTD*.

In Table I, our *LFTD* outperforms the other three state-of-the-art methods on the dataset of *shadow road*, which validates that our *LFTD* can be adaptive on the conditions with complex illumination and shadow. The results in Table I indicate that *VP* can achieve best performance in the dataset of *rain sequence*, as there are stable geometrical features, e.g. vanish points and edges, in that dataset. However, *VP* performs much worse in the other datasets due to its strong dependency on scenes' geometrical constrains, thus its overall performance is also relatively poor. In dataset *rain sequence*, our *LFTD* has slightly larger *FPR* than the *Gaussian*, as the *Gaussian* method may tend to overoptimize the *FPR* and then achieve worst *FNR* in all the methods, thus the *Gaussian*'s *ErrorRate* is also worst in the dataset of *rain sequence*. The *variational road* dataset contains frequently changed scenes, when the scenes changing, our *LFTD* will empty the DTD in previous and retrain the classifier from the *FM*. In the retraining frame, many passable regions are still regarded as impassable, which will produce a very high *FNR* and increase the average *FNR* of our method. That's why our *LFTD* outputs a slight higher *FNR* compared to the best method in *variational road* dataset.

We then present the *ErrorRate* of each continuous frame on all datasets in figure 9. The results show that the *ErrorRate* of our *LFTD* may be slightly high at the beginning, as only few regions are correct labeled based on the *FM*. However, the *LFTD*'s error rate will quickly decrease to the lowest one comparing to other methods with the frame adding. As shown in figure 9(c), there is also a high error rate in the middle of the *LFTD*'s curve, which is caused by the significant changing on the scene, and our *LFTD* can also quickly decrease to the lowest one with the frame adding. Thus our method can be insensitive and adaptive to the significant changing of scenes.



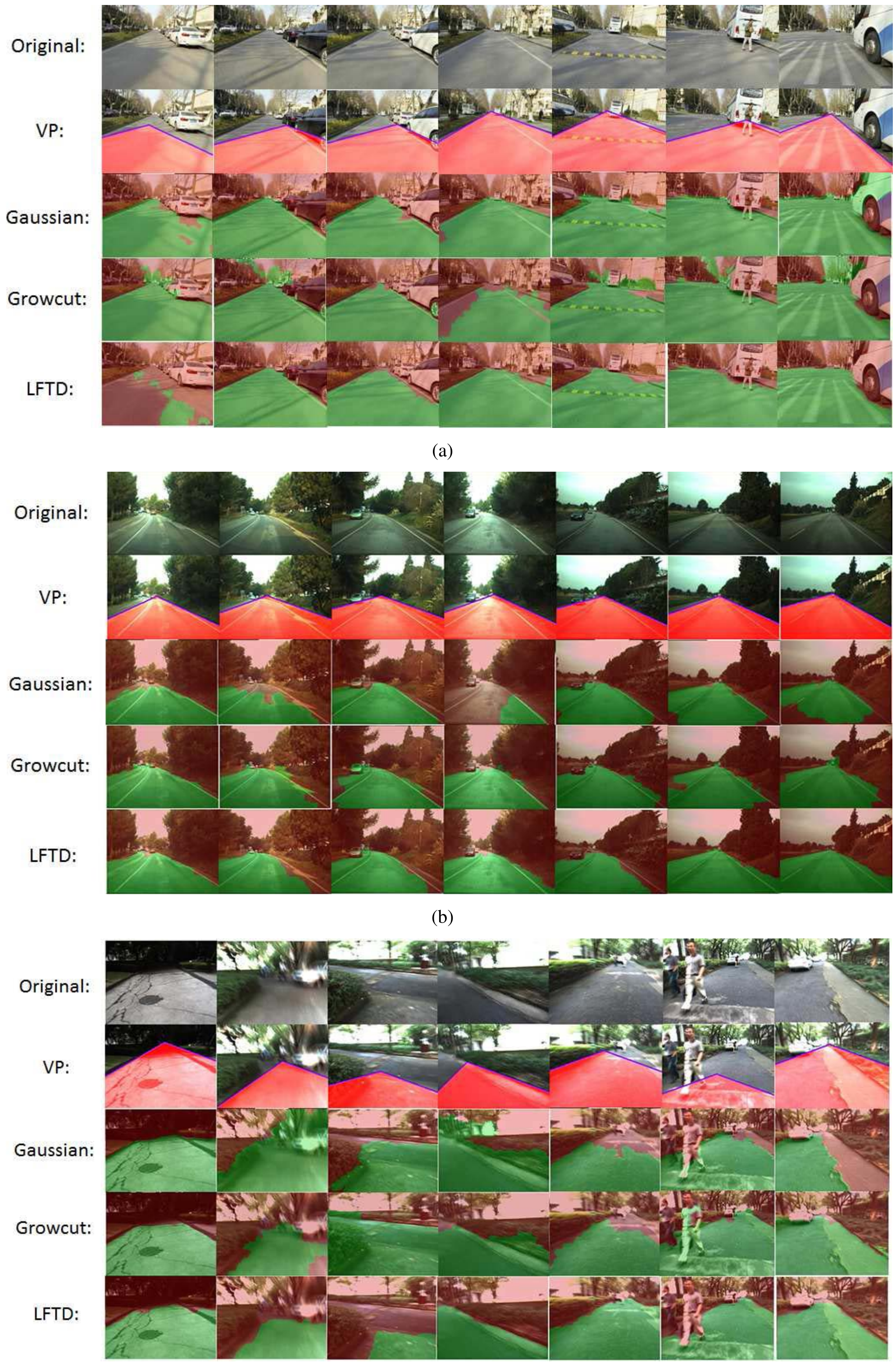
(a)

(b)

(c)

Fig. 9. The *ErrorRate* on the datasets of *shadow road*, *rain sequence* and *variation road*. The horizontal axis and vertical axis are the frame number and the average *ErrorRate* respectively. (a) *Shadow road*. (b) *Rain sequence*. (c) *Variational road*.

Figure 10 gives the detection results of each method in three datasets, the results are plotted with different colors in



(a)

(b)

(c)

Fig. 10. Detection results in consecutive frames. The row denoted as *original* present the original images, the other rows present the corresponding results on the four methods. The passable areas in second row are represented with red color following the setting of Kong's code, the passable areas in other rows are represented with green color. (a) *Shadow road*. (b) *Rain sequence*. (c) *Variational road*.

the sequenced frames. From figure 10(a), we can find the performance of *LFTD* will quickly increase to best with the frame increasing. While *VP* is always misled by the straight shadow projected by the tree trunk, and regards them as the edges of road, thus achieves quite low performance. In the sixth frame of figure 10(c), there is a pedestrian and only *LFTD* can detect the passable region correctly. The results also indicate *LFTD* can deal with the condition with moving objects by the online training model. The above experiments prove that our *LFTD* is able to process varied complex conditions, such

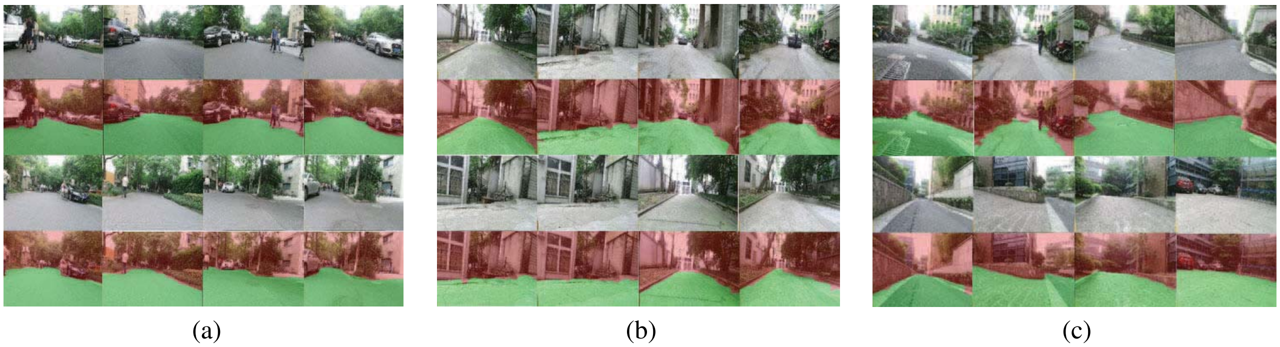


Fig. 11. Three Kinect-2 datasets used in our comparable experiments, the first and third rows are the original images and the second and the fourth rows are the labeled results with our KLFTD method, the results show that our KLFTD can label those complex scenes in these three datasets. (a) Kinect Dataset1. (b) Kinect Dataset2. (c) Kinect Dataset3.

as changed illumination, dynamic objects, shadows etc., and achieve well performance with only one set of parameters.

2) Experiments on Kinect-2: As there are no benchmark datasets for Kinect-2 system, we captured three datasets with the Kinect-2 to evaluate the performance of our Kinect-2 based learning framework for traversable region detection method (*KLFTD*).

The first dataset, shown in figure 11(a) and denoted as Kinect dataset1, includes 313 depth-2D image pairs, and there are many pedestrians and vehicles in this dataset, and the pedestrians and vehicles may burst in the views of the Kinect-2. There are also speed bumps in the dataset.

The second dataset, shown in figure 11(b) and denoted as Kinect dataset2, include 73 depth-2D image pairs, and there are various textures for the traversable regions, rough road surfaces, and barriers similar to the road surfaces in both colors and textures.

The third dataset, shown in figure 11(c) and denoted as Kinect dataset3, include 137 depth-2D image pairs, which contain both complex variational textures in the surfaces and walking pedestrians, vehicles in the image views. There are various conditions in that dataset, such as manhole covers, abrupt slops, steps and walls that are highly similar to the road surfaces in both colors and textures, cross roads and open areas etc.

The traversable regions in all these three datasets are manually labeled as the ground truths,[6] we also compare our *KLFTD* with three state-of-art methods introduced in previous as well as the standard monocular *LFTD* method in the evaluation experiments for Kinect-2 based traversable region detection. As the Kinect-2 sensor does not support stereo vision, the performance of the stereo vision based approach is not evaluated in the experiments of this subsection.

Several comparison labeling results of those five methods in typical scenes are given in figure 12. The results in this figure suggest that *KLFTD* can achieve much better performance than the other four methods, especially in the third and forth columns. The scene in the third column contains a cross road which consists of two significant varied road surfaces in both colors and textures, only the *KLFTD* can correctly recognize both two road surfaces in that scene. As there are only few correct samples for both road surfaces in that cross road scene based on the FM, the performance of the *LFTD* is relatively lower than the *KLFTD*. It also indicates that increasing the number of the correct samples can improve the performance of our learning framework significantly. The scene in the forth column contains some walls and windows whose colors and textures are highly similar to the road surfaces. In this condition, our *KLFTD* and *LFTD* can achieve much better performances than the *Gaussian* and *Growcut* methods, and the *KLFTD* can almost correctly recognize all the traversable regions in that scene.

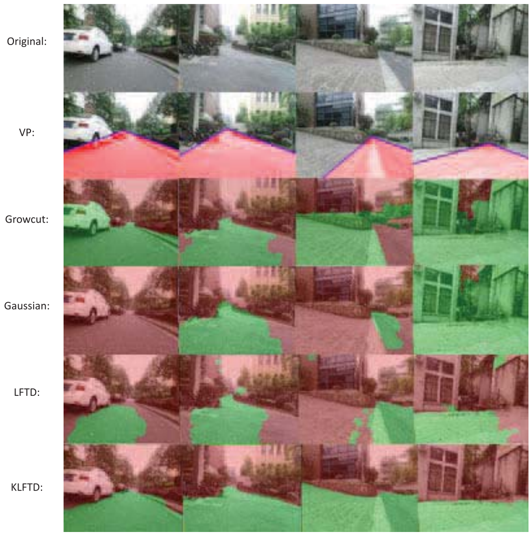


Fig. 12. Comparison results of *VP*, *Growcut*, *Gaussian*, *LFTD* and *KLFTD* in some challenging scenes extracted from the three benchmark Kinect datasets.

As these three Kinect-2 datasets are manually labeled, we can also use the metrics of *FPR*, *FNR* and *ErrorRate* to evaluate these five methods in all the three datasets quantitatively, the results are given in table II. The results show that our *KLFTD* can achieve the smallest *ErrorRate* in all the three datasets and the *FPR* and *FNR* of *KLFTD* are also controlled to relatively small among all the five methods. And our *LTFD* can also achieve more robust performances on *ErrorRate* (except *KLFTD*) and the *FPR* comparing with other state-of-art methods. It is also indicate that the method of *Growcut* may tend to over-optimize the *FNR* while the performances on *FPR* are not perfect in various datasets.

[6] The datasets can be download at, http://www.csc.zju.edu.cn/yliu/TVR/traverable_region.htm

TABLE II

QUANTITATIVE PERFORMANCE METRICS OF FIVE METHODS IN THREE KINECT DATASETS

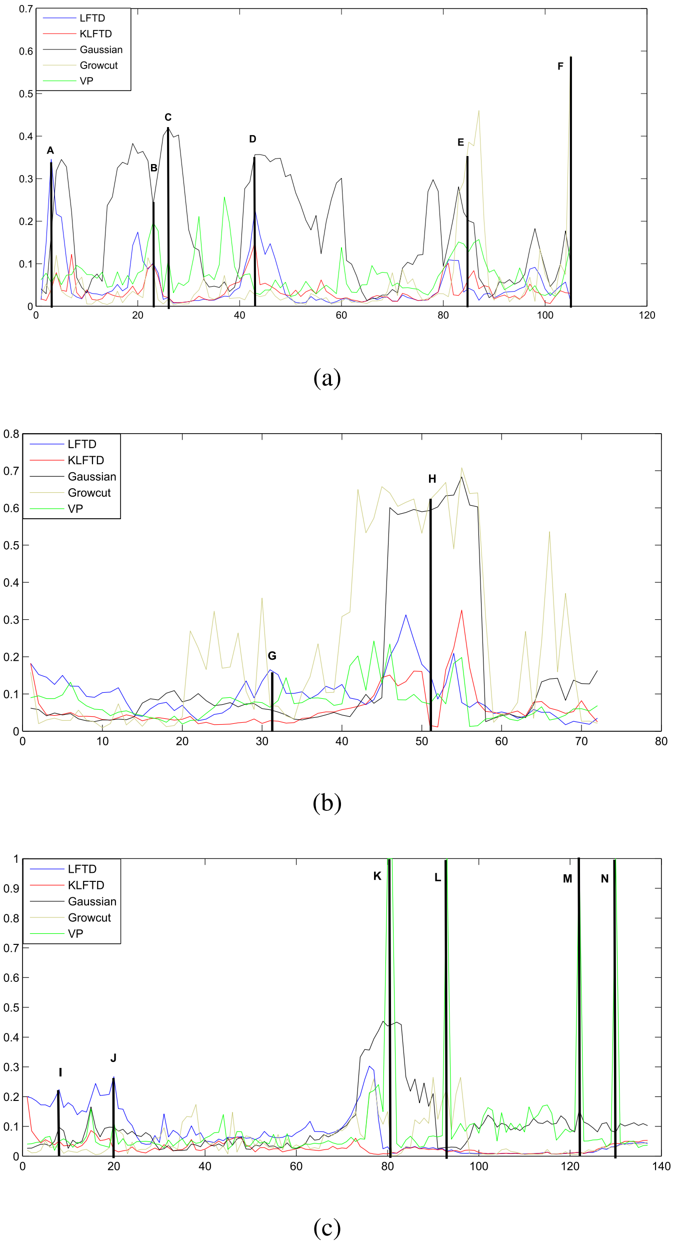
	Metrics	*Growcut*	*VP*	*Gaussian*	*LFTD*	*KLFTD*
Kinect Dataset1	ErrorRate	5.40	7.30	20.91	4.91	**3.24**
	FPR	9.98	9.00	**0.99**	1.53	2.90
	FNR	**3.30**	6.77	35.76	7.71	3.73
Kinect Dataset2	ErrorRate	22.52	7.57	15.98	8.59	**4.76**
	FPR	62.57	14.72	36.03	**1.56**	7.73
	FNR	**1.26**	4.10	5.16	12.34	3.33
Kinect Dataset3	ErrorRate	5.32	10.14	10.31	7.50	**2.84**
	FPR	8.47	8.86	5.32	**1.76**	2.03
	FNR	**3.44**	11.63	16.87	11.65	3.50
Overall performance	ErrorRate	11.08	8.34	15.73	7.00	**3.61**
	FPR	27.01	10.86	14.11	**1.62**	4.22
	FNR	**2.67**	7.50	19.26	10.57	3.52

In figure 13, we also present the *ErrorRate* curves of consecutive frames in all datasets. It also shows that both the *KLFTD* and *LFTD* may output higher error rates in the initial frames and then will quickly converge. The results in figure 13 also show that the *KLFTD* will converge faster than the *LFTD*, as the *KLFTD* can obtain more correct training samples in the initialization stage.

To further analyze the error rate curves, we will focus on several points in the error rate curves as shown in figure 13 and extract the labeled scenes output by different methods in those focus points.

The extracted scenes in Kinect dataset1 are shown in figure 14, which shows the focus points from *A* to *F*. From frame *A* and *B*, we can find that none of the methods can correctly label the sidewalk in both scenes except our *KLFTD*, as our *KLFTD* can fuse the depth information from Kinect-2 and recognize the impassable sidewalk although its texture and color are highly similar to the road surface. In frame *C* and *D*, the narrow scenes are suddenly changed to open areas, and the *KLFTD* can also achieve better performance than the *LFTD* as the *KLFTD* has more samples during the training process. In the complex scenes of frame *E* and *F*, only our *KLFTD* and *LFTD* can successfully label most of the traversable regions, the *Growcut* fails to label the trees in frame *E* and *F*, the *Gaussian* fails to label the manhole cover in frame *F* and the *VP* fails to find a wrong vanish point in frame *F*, as there is a car in the front of the scene. The results of frames *A* to *F* also indicate that the *Gaussian* method can only detect few traversable regions, which leads that the probability of labeling the impassable regions to the traversable regions is significant decreased, thus its *FNR* tends to quite low, however, its *FPR* will be excessive high. This condition for the *Gaussian* method often occurs in the following frames and the other two Kniect datasets, and it can explain why the *Gaussian* method's FNR in table II are all smallest in all these three datasets.

The extracted scenes in Kinect dataset2 are shown in figure 15, which shows the focus points from *G* to *H*. These scenes are most consist of various barriers whose colors and textures are highly similar to the road surface. In the beginning frame *G*, the results of all the methods are still acceptable, while the next frame *H*, the *Gaussian* and *Growcut* methods almost totally fail to label the traversable regions. It can be also found from figure 13(b), both the *Gaussian* and *Growcut* methods appear severe bad error rates during the scenes close to frame *H*. While the *VP* also mislabels the window-wall in frame *G*, and finds wrong vanish point in frame *H*.



(a)

(b)

(c)

Fig. 13. The *ErrorRate* on the datasets of Kinect Dataset1, Kinect Dataset2 and Kinect Dataset3. The horizontal axis and vertical axis are the frame number and the average *ErrorRate* respectively. (a) *Kinect Dataset1*. (b) *Kinect Dataset2*. (c) *Kinect Dataset3*.

The extracted scenes in Kinect dataset3 are shown in figure 16, which shows the focus points from *I* to *N*. These focus scenes are most consist of open areas and unparallel roads. Thus the *VP* method will often fail to detect the vanish point and lead to poor performance, such as frame *K*, *L*, *M*, *N*. The *Gaussian* method still tends to label a shrinking traversable region. Although the *Growcut* method can relative better performance in most of the frames, it will fail to label the green vegetation in frame *L*. And our *KLFTD* and *LFTD* can achieve much better results in these focus scenes. As the scenes from frame *I* to *J* are changing suddenly, the results of *LFTD* are less complete due to its slower rate of convergence for training.

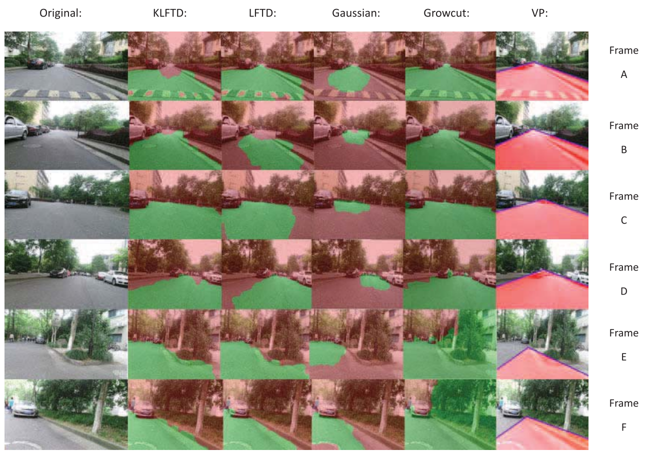


Fig. 14. Extracted scenes(frames) sampled from Kinect Dataset1, these sampled frames are corresponding to the focus points *A* to *F* in figure 13(a). The labeled results on these sampled frames output by the five methods are also shown for further comparison.

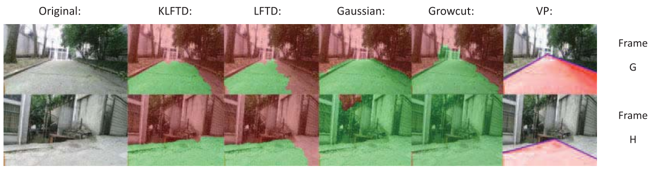


Fig. 15. Extracted scenes(frames) sampled from Kinect Dataset2, these sampled frames are corresponding to the focus points *G* to *H* in figure 13(b). The labeled results on these sampled frames output by the five methods are also shown for further comparison.

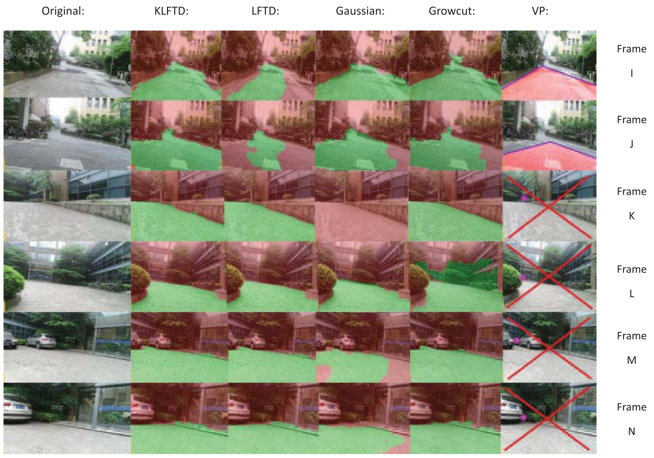


Fig. 16. Extracted scenes(frames) sampled from Kinect Dataset3, these sampled frames are corresponding to the focus points *I* to *N* in figure 13(c). The labeled results on these sampled frames output by the five methods are also shown for further comparison.

3) Experiments on Stereo Camera: In the experiments, a Bumblebee stereo is used to capture sequenced image-pairs which are used to evaluate our stereo vision based learning framework for traversable region detection (*SLFTD*). We capture two challenging datasets. The first dataset includes blurred images under various road surfaces, and the results are given in figure 17, and the results show our *SLFTD* can be robust against the blurring caused by the fast motion of the camera, as our approach is based on the statistical features on super-pixels. The second dataset includes varied and fast lighting changes, the lighting varies from overexposure to underexposure frequently when the robot goes through the forest road. And the results are given in figure 18, which shows the *SLFTD* can successfully handle that challengeing condition with frequent lighting changes. For more results with our *SLFTD* please refer to our attached demonstration video.

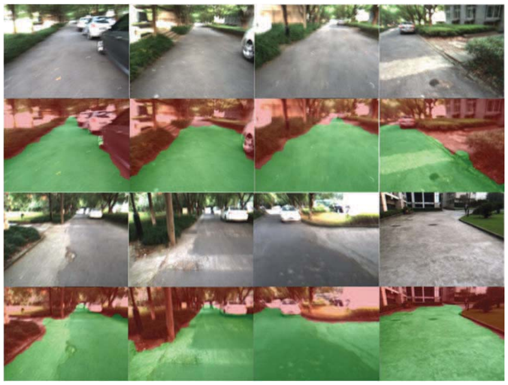


Fig. 17. Experimental dataset1 and its corresponding labeled results in stereo vision system. The first and third rows are the original images and the second and fourth rows are the labeled results.

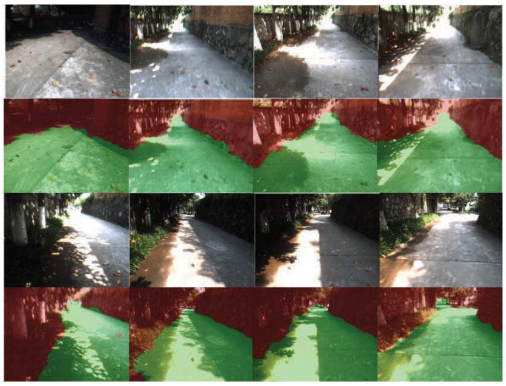


Fig. 18. Experimental dataset2 and its corresponding labeled results in stereo vision system. The first and third rows are the original images and the second and fourth rows are the labeled results.

We also compare our *SLFTD* with the *LFTD* on the manually labeled variational road dataset. The error curves of these two methods are shown in figure 19, and we can find that *SLFTD* performances much better than the *LFTD* on almost all the dataset. We will further focus on two points of the curves, one point is the 10*th* frame, where the performance of our *SLFTD* is worse than the *LFTD*, the detailed results on the 10*th* frame are shown in figure 20. The 10*th* frame contains a low separation between the grass and the road, and the *SLFTD* will tend to recognize both the grass and the road as one big plane based on the stereo information as the separation is too small to stride by robots, then *SLFTD* will labeled the grass as traversable region based on its initial input mask. While the *LFTD* only obtains few samples from the grass based on the FM, thus it will only label the road and less grass. The other focus point is the 40th frame, where our *SLFTD* is much better than the *LFTD*, the detailed results on the 40*th* frame are shown in figure 21. The 40*th* frame contains a cross road with

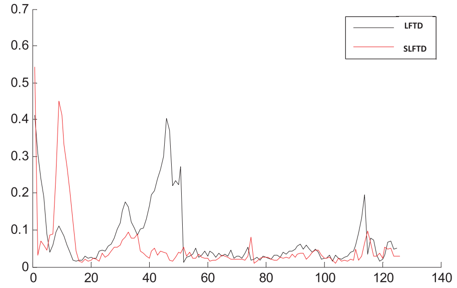


Fig. 19. The *ErrorRate* curves on *variational road* dataset of *SLFTD* and *LFTD*.

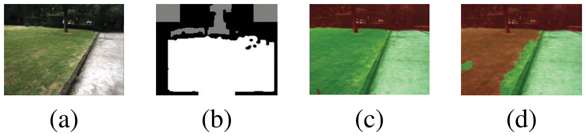


Fig. 20. (a) is the source image, (b) is the mask of *SLFTD*, (c) and (d) are the results of the *SLFTD* and *LFTD* in the 10*th* frame of *variational road* dataset.

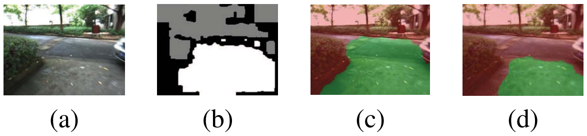


Fig. 21. (a) is the source image, (b) is the mask of *SLFTD*, (c) and (d) are the results of the *SLFTD* and *LFTD* in the 40*th* frame of *variational road* dataset.

two different surfaces, obviously, our *SLFTD* can correctly label both kinds of road surfaces as the mask provide enough information, while the *LFTD* can only label one kind of road surface. The error curves shown in figure 19 also indicate that the *SLFTD* will converge to the optimal results much faster than the *LFTD*.

4) Computational Time: In all experiments above, the input image is first resized to 320 × 240 and then fed into the algorithms respectively. The computational times for all algorithms are recorded using this configuration. The results are shown in table III. It can be found that the proposed framework is faster than the other methods owing to the very efficient closed form of ELM training. In real application, a relatively fast outdoor robot with 30km per hour can pass 1.25m during the computation of an image frame, which is generally guaranteed as the proposed framework yields a dense detection of traversable region even on the long-range pixels in the image.

5) Discussion: In real application, FPR we think is the most important indicator for traversable region detection, since misclassifying the obstacle as the passable region is seriously unacceptable due to the security. The learning framework plugged with FM achieves the lowest FPR in most datasets, followed by the Kinect2 initializer, and outperform than other methods. While for the error rate, the Kinect2 initializer is the best, followed by FM version. This fact demonstrates that the more geometric information incorporated, the better shaping of the online sample space, leading to a boosted overall performance. However, due to the slightly non-planar road and sensory noise, mislabeling can also be brought by the geometric assumption, especially in distant parts, which is also investigated in previous study [29]. When FM is applied, the assumption is self-validated, though introducing the other online training data, the method is still conservative due to the sample weighting, thus achieving a better FPR. From this result, and also a previous study [28], we consider the performance, when we only use the FM to collect the online training data, can be even more conservative. However, the performance on the Kinect Dataset3, shown in table IV, is with worse performance than the proposed framework, which is caused by the too sparse geometric initialization, insufficient to capture the intrinsic variations of the data. Therefore, our framework can be considered as a balance between the data variations, assumption validity and label density.

TABLE III
THE COMPUTATIONAL TIME FOR SIX METHODS

Methods	*Growcut*	*VP*	*Gaussian*	*LFTD*	*KLFTD*	*SLFTD*
Time (s)	1.27	72.60	3.24	**0.148**	0.152	0.156

TABLE IV
THE PERFORMANCE OF PURE FM INITIALIZER

	ErrorRate	FPR	FNR
Kinect Dataset1	6.18	3.05	9.20
Kinect Dataset2	11.23	2.29	16.04
Kinect Dataset3	7.76	2.11	11.89
Overall Performance	8.39	2.48	12.38

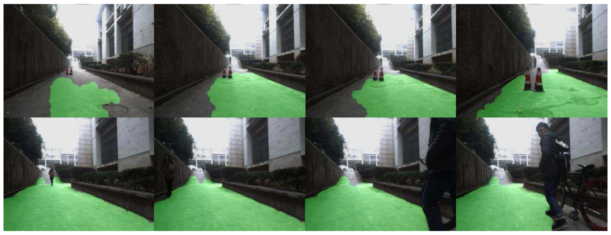


Fig. 22. Evaluation of the feasibility in robot navigation application when robot approaches an obstacle (top row), or moving obstacles approach the robot (bottom row).

To evaluate the feasibility of the FM in real robot navigation task, we test a case study in which a robot is moving toward an obstacle, i.e. a traffic cone, a pedestrian, and a cyclist. The results are shown in figure 22. When the robot approaches the cone, the cone is stably identified as an obstacle until 1.6m. The pedestrian walking toward the robot, and the appearing cyclist, are both detected too. These facts reflect that the robot is able to avoid the obstacles only if the robot can plan a correct path before the security range. This range can be set as the minimum distance to obstacle in a robot navigation planner. In general, a motion planner can always keep the robot from the obstacles in a minimum distance. Therefore, the framework is accepted when considering the navigation and passable region detection in the loop, which is always the case in mobile robots.

VI. CONCLUSION AND FUTURE WORK

This paper presents a novel and scalable travelable region detection framework fusing with both appearance and geometrical information, our approach can be insensitive to challenging environments with varied lightings, shadows, blurred images and dynamic objects. We also carried out quantitative experiments to evaluate our approach with state-of-the-art methods and proved that our approach can achieve superior performances on both adaptability and stableness in real traversable region detection.

As our approach only uses a weak assumed model of the traversable and impassible regions, which can be satisfied in most conditions, the model's universality of our method can be guaranteed. The *found mask* introduced in our approach can be also reinforced by the geometrical information, which will further improve the performance of our method. In additional,the online learning framework in our approach can update the internal parameters of our method and make the parameters adaptive (or robust) to the changes of scenes or illuminations.

In the future, we plan to improve the computation complexity of our approach with the GPU techniques, as most of the computation costs spend on the segmentation of superpixels and its corresponding feature construction, which may be easily parallelized to calculate.

References

[1] L. Shi, R. Khushaba, S. Kodagoda, and G. Dissanayake, "Application of CRF and SVM based semi-supervised learning for semantic labeling of environments," in *Proc. 12th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Guangzhou, China, Dec. 2012, pp. 835–840. [Online]. Available: http://dx.doi.org/10.1109/ICARCV.2012.6485266

[2] K. Lu, J. Li, X. An, and H. He, "A hierarchical approach for road detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 517–522.

[3] W. Zong, G.-B. Huang, and L. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, Feb. 2013.

[4] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2211–2220, Aug. 2010.

[5] C. Siagian, C.-K. Chang, and L. Itti, "Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 564–571.

[6] T.-T. Tran, H.-M. Cho, and S.-B. Cho, "A robust method for detecting lane boundary in challenging scenes," *Inf. Technol. J.*, vol. 10, no. 12, pp. 2300–2307, 2011.

[7] O. Miksik, "Rapid vanishing point estimation for general road detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 4844–4849.

[8] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3D scene priors for road detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 57–64.

[9] W.-H. Zuo and T.-Z. Yao, "Road model prediction based unstructured road detection," *J. Zhejiang Univ. Sci. C*, vol. 14, no. 11, pp. 822–834, 2013.

[10] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot monocular vision navigation based on road region and boundary estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 1043–1050.

[11] A. Seki and M. Okutomi, "Robust obstacle detection in general road environment based on road extraction and pose estimation," *Electron. Commun. Jpn. (II, Electron.)*, vol. 90, no. 12, pp. 12–22, 2007.

[12] P. Lombardi, M. Zanin, and S. Messelodi, "Unified stereovision for ground, road, and obstacle detection," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2005, pp. 783–788.

[13] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *Proc. IEEE Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 388–394.

[14] T. Butkiewicz, "Low-cost coastal mapping using Kinect v2 time-of-flight cameras," in *Proc. Oceans-St. John's*, Sep. 2014, pp. 1–9.

[15] C. Rotaru, T. Graf, and J. Zhang, "Color image segmentation in HSI space for automotive applications," *J. Real-Time Image Process.*, vol. 3, no. 4, pp. 311–322, 2008.

[16] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4. May 2002, pp. 4320–4325.

[17] S. Zhou and K. Iagnemma, "Self-supervised learning method for unstructured road detection using fuzzy support vector machines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2010, pp. 1183–1189.

[18] S. Zhou, J. Gong, G. Xiong, H. Chen, and K. Iagnemma, "Road detection using support vector machine based on online learning and evaluation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 256–261.

[19] Q. Zhang, Y. Liu, Y. Liao, and Y. Wang, "Traversable region detection with a learning framework," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1678–1683.

[20] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain," in *Proc. Robot., Sci. Syst.*, Philadelphia, PA, USA, 2006, pp. 1–7.

[21] L. M. Paz, P. Piniés, and P. Newman, "A variational approach to online road and path segmentation with monocular vision," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 1633–1639.

[22] J. Lee, Y. Lu, and D. Song, "Planar building facade segmentation and mapping using appearance and geometric constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 1060–1065.

[23] J. M. Alvarez, A. López, and R. Baldrich, "Illuminant-invariant model-based road segmentation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 1175–1180.

[24] B. Upcroft, C. McManus, W. Churchill, W. Maddern, and P. Newman, "Lighting invariant urban street classification," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 1712–1718.

[25] B. Wang and V. Frémont, "Fast road detection from color images," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1209–1214.

[26] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[27] R. Batuwita and V. Palade, "FSVM-CIL: Fuzzy support vector machines for class imbalance learning," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 558–571, Jun. 2010.

[28] J. M. Alvarez, M. Salzmann, and N. Barnes, "Learning appearance models for road detection," in *Proc. Intell. Vehicles Symp.*, 2013, pp. 423–429.

[29] W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Color-based free-space segmentation using online disparity-supervised learning," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 906–912.

[30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[31] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[32] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.

[33] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

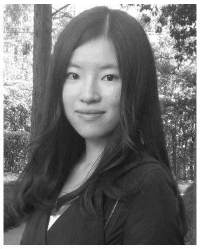
[34] J. Sell and P. O'Connor, "The Xbox one system on a chip and Kinect sensor," *IEEE Micro*, vol. 34, no. 2, pp. 44–53, Mar./Apr. 2014.

[35] J. Fritsch, T. Kühnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *Proc. Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1693–1700.

[36] C. Guo, S. Mita, and D. McAllester, "MRF-based road detection with unsupervised learning for autonomous driving in changing environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2010, pp. 361–368.

Yue Wang is a Research Fellow with the Department of Control Science and Engineering, Institute of Cyber-Systems and Control, Zhejiang University. His latest research interests include mobile robotics and robotic vision.

Yong Liu is a Professor with the Department of Control Science and Engineering, Institute of Cyber-Systems and Control, Zhejiang University. His latest research interests include machine learning, robotics vision.



Yiyi Liao is working toward the Ph.D. degree with the Department of Control Science and Engineering, Institute of Cyber-Systems and Control, Zhejiang University. Her latest research interests include deep neural network learning and robotics vision.

Rong Xiong is a Professor with the Department of Control Science and Engineering, Institute of Cyber-Systems and Control, Zhejiang University. Her latest research interests include robot planning and robotics vision.