Robust Ground Plane Detection from 3D Point Clouds

Sunglok Choi*, Jaehyun Park, Jaemin Byun, and Wonpil Yu

Intelligent Cognitive Technology Research Dept., ETRI, South Korea
*Corresponding author (E-mail: sunglok@etri.re.kr, Web: http://sites.google.com/site/sunglok)

Abstract: Ground provides useful and basic information such as traversal regions and location of 3D objects. The given point cloud may contain a point not only from ground, but also from other objects such as walls and people. Those points from other objects can disturb to find and identify a ground plane. In this paper, we propose robust and fast ground plane detection with an asymmetric kernel and RANSAC. We derive a probabilistic model of a 3D point based on an observation that a point from other objects is always above the ground. The asymmetric kernel is its approximation for fast computation, which is incorporated with RANSAC as a score function. We demonstrate effectiveness of our proposed method as quantitative experiments with our on-road 3D LiDAR dataset. The experimental result presents that our method was sufficiently accurate with slightly more computation. Finally, we also show our ground detection's application to augmented perception and visualization for drivers and remote operators.

Keywords: ground plane detection, 3D point cloud, asymmetric kernel, RANSAC, traversable region, obstacle detection

1. INTRODUCTION

Ground detection is essential and necessary for many applications in robotics, intelligent vehicles, and computer vision. Ground is regarded as a traversal region for mobile robots and vehicles so that it is required for autonomous navigation, especially obstacle detection and avoidance. It is also quite useful for remote operators to drive robots or vehicles with restricted visual information. In addition, road detection is a specialized form of ground detection with considering additional visual and structural characteristics of roads. In computer vision, ground enables 3D inference from 2D images. With a known ground plane, it is possible to reconstruct 3D structures on the ground and estimate 3D position of objects on the ground. It is also a popular and common preprocessing to split and label 3D point clouds.

Ground detection has been heavily investigated due to its necessity and importance. It is effective to analyze the previous works according to *local* and *global* approaches. The local approaches start from locally extracted features and their clustering and classification lead ground regions. It is a bottom-up approach. In contrast, assuming a globally consistent ground model, global approaches search and find the model from raw data or features, which is top-down. Generally, the local approaches can deal with more general and flexible situations such as rough terrain and irregular structures, but they require higher spatial and time complexity. In contrast, the global approaches need less computation and memory, but they cannot detect ground beyond their defined ground model. *For 2D images*, the local approaches

mostly pursued classical image segmentation with additional features for ground [2]. The popular features were color and texture, and their segmentation was done by Markov random field (MRF) or graph cuts or others. The global approaches commonly adopted a plane and planar homography as their model and update rule [1]. Planar homography can describe geometric relationship of a ground plane observed on a pair of images, and its incremental update enables to track the ground plane. RANSAC, filtering, and local optimization were applied to estimate planar homography. *For 3D data*, the local approaches were more common and popular [3], [4], [5], [6], [7]. Due to additional information, various geometric local features have been proposed such as vertical height, gradient, normal vector, convexity and others. MRF and support vector machine (SVM) were also popular tools to segment and classify local features toward ground. Similarly to 2D images, the global approaches also utilized a flat 3D plane as their ground models, and RANSAC was a popular tool to detect the ground plane [8]. As mentioned before, the local and global approaches have their respective strength and weakness, which is flexibility and time/spatial complexity.

In this paper, we propose a novel approach to detect a ground plane quickly and accurately from 3D point clouds. The given sensor acquires 3D points not only from ground, but also from other surroundings such as buildings, pedestrians, and vehicles. Therefore, it is important to identify a ground plane regardless of the points from such other objects. We basically pursue RANSAC approach to overcome such harmful points, but with our proposed asymmetric kernel. The asymmetric kernel is inspired from an observation that the harmful points exist always over the ground, not under the ground. From the observation, we model its probabilistic likelihood and approximate it as the asymmetric kernel for faster compu-

tation. Its preliminary idea was introduced in our previous work [9] for resolving scale ambiguity in monocular visual odometry. The asymmetric kernel and overall procedure are explained in Section 2. Its effectiveness was demonstrated by experiments and quantitative analysis in Section 3. We also present its application to recognize spatial information (e.g. traversable regions and obstacles) and provide it to drivers and remote operators for easier control and operation. The application is briefly presented in Section 4.

2. ROBUST GROUND PLANE DETECTION

2.1 Problem Formulation

Ground plane detection is a process to find parameters describing a ground plane from the given 3D points. The set of parameters is written as Θ and 3D points are represented as

$$X = \{x_1, x_2, \cdots, x_N\}, \quad x_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}. \qquad (1)$$

The detection problem can be formulated as optimization of the parameters with points on ground as follows:

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{x_i \in X_G} f(x_i; \Theta), \qquad (2)$$

where X_G is a set of the given points on the ground surface and f is a function to calculate error between the given point and ground plane. However, the problem is not simple because X_G is unknown. In this paper, we define the problem as *regression analysis*, estimating unknown parameters from the given data, but it can be also formulated as *classification* of the given points into two groups, ground and others. If the ground parameter $\hat{\Theta}$ is known from the optimization, classification is also simply resolved. For example, the label of each point can be identified (1: ground, 0: others) as

$$l(x_i; \hat{\Theta}) = \begin{cases} 1 & \text{if } |f(x_i; \hat{\Theta})| < \tau \\ 0 & \text{otherwise} \end{cases}, \qquad (3)$$

where τ is a threshold to decide whether a point belongs to ground or not.

2.2 Ground Models

Ground can be described in various ways. One of the most general representations is a point cloud, a set of 3D points laying on a ground surface. It can represent diverse terrain and slopes with an enough number of points. However, such non-parametric expression requires high computation burden for further operations such as checking contact or collision. In many computer animations and games, ground is depicted by a polygon mesh, a collection of connected polygons. These polygons can attain generality with sufficient quantization level, but still
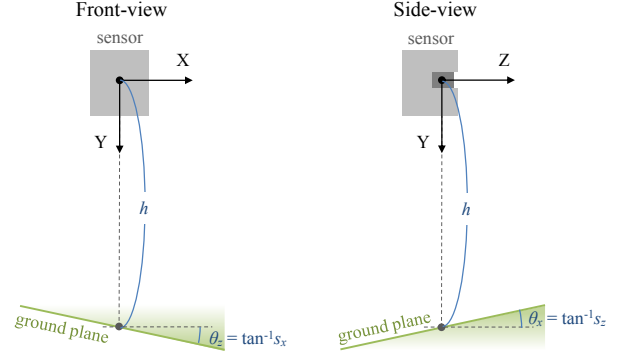

Fig. 1: The sensor and ground plane are presented based on the sensor coordinate. We use a conventional camera coordinate, whose leftward and downward axes are x- and y-directional axes, respectively.

needs a bunch of parameters to describe them. Similarly, a grid map such as an elevation map and voxel map is another alternative of a point cloud, but still requires a lot of memory to keep it.

In this paper, we model ground as a 3D plane. Even though a plane does not present various terrain and slopes, it is sufficiently effective for many applications. Most indoor environments contain flat ground, and roads can be modeled as locally planar. Moreover, the (locally) planar assumption is valid for many outdoor fields such as playgrounds and farms. Generally, a 3D plane can be represented by four parameters (3 DoF) as follows:

$$ax + by + cz + d = 0 \quad \text{such that} \quad a^2 + b^2 + c^2 = 1, \quad (4)$$

whose ground parameter is $\Theta = [a, b, c, d]$. In contrast to point clouds and polygon meshes, quite few parameters are enough to describe ground. In addition to planar assumption, we also utilize other prior that a sensor (attached to robots or vehicles) keeps almost constant interval from ground during its moving. The ground plane with a fixed sensor (2 DoF) is written as

$$y = s_x x + s_z z + h, \qquad (5)$$

where h is the constant height between the sensor and ground ($\Theta = [s_x, s_z]$). Figure 1 presents the sensor and ground plane with parameters. Moreover, a slope along with x-axis, s_x, is often fixed or zero when ground is not tilted from side to side (e.g. indoor and on-road) and the sensor is aligned well. In those cases, the ground is represented with only one parameter (1 DoF, $\Theta = [s_z]$). From the most general 3 DoF plane to well-aligned 1 DoF plane, we have enumerate three different degree-of-freedom of 3D planes for modeling ground. In this paper, we adopt 1 DoF ground model (fixed h and s_x) because it is sufficient for many applications and needs only one point to estimate the ground parameter. Such 1-point plane estimation can exponentially speed up RANSAC

[10] in contrast to the general 3-point estimation. Moreover, our experiments presented that 2 and 3 DoF ground models are too flexible to be deteriorated by outliers.

2.3 Asymmetric Kernels

Error between a point and ground plane is commonly defined as their geometric distance,

$$f(\mathrm{x_i};\Theta) = \frac{y_i - s_x x_i - s_z z_i - h}{\sqrt{1 + s_x^2 + s_z^2 + h^2}}. \tag{6}$$

The distance is signed so that it is positive when a point is above the plane and negative when a point is below the plane. As mentioned before, the given 3D points can come not only from ground, but also from surrounding objects. We will call a point from ground as *inliers* and others as *outliers*. If we know the true ground plane, error between the truth and given point is a quite useful clue to identify whether the point is an inlier or outlier. The huge amount of error means the point far from the ground plane. We can also decide that the point is probably from other objects, that is an outlier. Similarly, we can consider a point as an inlier when the point is close to the ground plane.

Likelihood of a point is derived from such reasonable observation on the magnitude of error. We can model error from an inlier as Gaussian distribution as follows:

$$\mathrm{p}(f_i|l_i = 1) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{f_i^2}{2\sigma^2}\right), \tag{7}$$

where f_i and l_i are short notation for error and label of i-th point, and σ is standard deviation, that is the magnitude of inlier noise. All objects exist above the ground so that outliers are also extracted from somewhere above the ground, and its distance from the ground is unknown. We can model error from an outlier as uniform distribution as follows:

$$\mathrm{p}(f_i|l_i = 0) = \begin{cases} 1/h_{\max} & \text{if } -h_{\max} \leq f_i \leq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{8}$$

where $h_{\max}$ is the maximum height that object can exist above the ground. Finally, from total probability theorem, we can represent overall likelihood as

$$\begin{aligned} \mathrm{p}(f_i) &= \mathrm{p}(l_i = 1)\,\mathrm{p}(f_i|l_i = 1) + \mathrm{p}(l_i = 0)\,\mathrm{p}(f_i|l_i = 0) \\ &= \gamma_i\,\mathrm{p}(f_i|l_i = 1) + (1 - \gamma_i)\,\mathrm{p}(f_i|l_i = 0), \end{aligned} \tag{9}$$

where γ_i is short notation for prior probability of i-th point being an inlier, $\mathrm{p}(l_i = 1)$.

Maximum likelihood estimation (MLE) is adopted to find the ground parameter. Under independent and identically distributed (i.i.d.) assumption, it is mathematically represented as

$$\begin{aligned} \hat{\Theta} &= \arg\max_{\Theta} \mathrm{p}(\mathrm{X}|\Theta) \\ &= \arg\max_{\Theta} \prod_{i=1}^{N} \mathrm{p}(f_i) \\ &= \arg\max_{\Theta} \sum_{i=1}^{N} \log(\mathrm{p}(f_i)) \\ &= \arg\max_{\Theta} \sum_{i=1}^{N} \kappa(f_i), \end{aligned} \tag{10}$$

where a kernel function, κ, is a short notation for log-likelihood of each point. To reduce computational burden for calculating log-likelihood, we approximate the log-likelihood as asymmetric Gaussian kernel as follows:

$$\kappa(f_i) \simeq \begin{cases} \exp(-0.5 f_i^2/\sigma_+^2) & \text{if } f_i > 0 \\ \exp(-0.5 f_i^2/\sigma_-^2) & \text{otherwise} \end{cases}, \tag{11}$$

where σ_+ and σ_- are standard deviation of each side ($\sigma_+ < \sigma_-$), respectively. Two different standard deviations impose asymmetry to the kernel. Figure 2 shows an example of log likelihood and its approximated kernel.

2.4 RANSAC-based Ground Detection

RANSAC framework is utilized to find the ground parameter with unknown inlier/outlier labels. To estimate the ground parameter, we need to rely on 3D points measured from a ground surface, but we don't know which point is an inlier. RANSAC is one of the most popular solutions to resolve such outlier problem. RANSAC is a kind of random search and composed of two steps, hypothesis generation and evaluation [10]. Its overall procedure is presented in Algorithm 1. A function *EstimatePlane* calculates the ground plane using the given points. It is simply done by solving Equation 5. A function *AdaptiveTerm* calculates the necessary number of iterations considering the ratio of inliers [10]. Therefore, our RANSAC can adaptively adjusts its number of iterations based on a probabilistic condition of confidence and inlier ratio. After RANSAC iterations, we get the best estimate Θ_{best}, and it is further refined as Θ_{final} in the least-squares sense.

Our approach is novel because we apply our asymmetric kernel to RANSAC. The conventional RANSAC uses a rectangular and symmetric kernel similar to Equation 3, and M-estimator SAC (MSAC) is also based on a smooth but symmetric kernel. Our asymmetric kernel is inspired from log-likelihood of a 3D point with its unknown label. Since outliers exist always above the ground plane, our kernel has longer tail in negative domain. Such asymmetry enables RANSAC more robust to outliers, which was shown in our experiments and application.
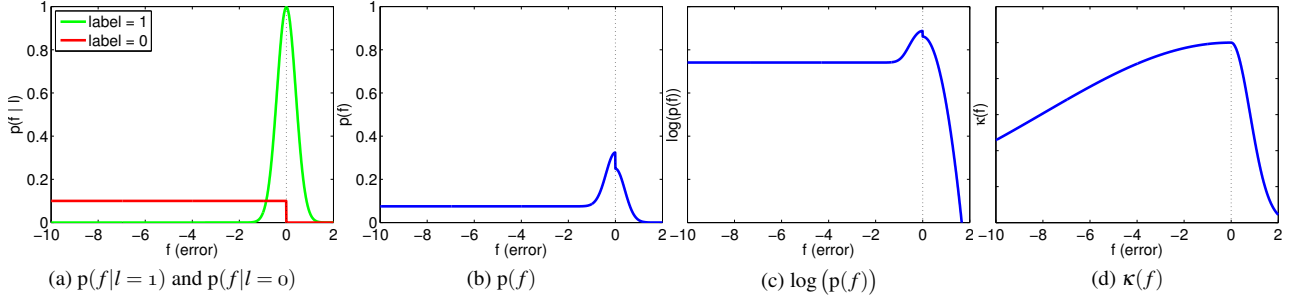
Fig. 2: Error likelihood is drawn in (a) inlier/outlier and (b) mixture, and its log likelihood is depicted in (c) in case of $\sigma = 0.4$, $h_{\max} = 10$ and $\gamma = 0.2$. The asymmetric kernel is presented in (d) with $\sigma_+ = 0.8$ and $\sigma_- = 8$.

Algorithm 1 Ground plane detection with RANSAC

Require: 3D points $X = \{x_1, x_2, \cdots, x_N\}$

1:    $i := 0$
2:    $T := T_{\max}$
3:    $(L_{\text{best}}, \Theta_{\text{best}}) := (0, 0)$
4: **while** $i < T$ **do**
5:      $i := i + 1$
      // 1. Hypothesis generation
6:      $q := \texttt{RandomNumber}(1, N)$
7:      $\Theta := \texttt{EstimatePlane}(x_q)$
      // 2. Hypothesis evaluation
8:      $L := \sum_{i=1}^{N} \kappa(f(x_i; \Theta))$
9:      **if** $L > L_{\text{best}}$ **then**
10:        $(L_{\text{best}}, \Theta_{\text{best}}) := (L, \Theta)$
11:        $\gamma := \frac{1}{N} \sum_{i=1}^{N} l(x_i; \Theta)$
12:        $T := \min(\texttt{AdaptiveTerm}(\gamma), T_{\max})$
13:      **end if**
14: **end while**
      // 3. Final refinement
15: $X_G := \{x | x \in X \text{ and } l(x; \Theta_{\text{best}}) = 1\}$
16: **return** $\texttt{EstimatePlane}(X_G)$

---

3. EXPERIMENTS

3.1 Configuration

We performed experiments with 3D LiDAR datasets [7] to quantify performance of our proposed RANSAC with the asymmetric kernel. The 3D point clouds were acquired on usual on-road situations using Velodyne HDL-64E. We manually constructed their ground truth so that each point's label l_i is available for evaluation. We used 13 datasets whose examples and ground truth are presented in Figure 4. We compared our proposed approach with the original RANSAC. We configured T_{max} as 100 and RANSAC's fixed number of iteration was also selected as 100. We measured each algorithm's averaged accuracy and computing time and calculated precision and recall for more detailed analysis. The evaluation was performed in MATLAB 2010a with Intel Core i7 2.80 GHz. For statistically meaningful results, we executed each algorithm 100 times and adopted their average.

Algorithm	Accuracy	Comp. Time
Original RANSAC	0.89	15.4 [msec]
Asym. RANSAC	0.90	17.6 [msec]

Table 1: Two algorithms were evaluated in the view of averaged accuracy and computing time. The computing time was measured in MATLAB 2010a using *tic* and *toc* functions
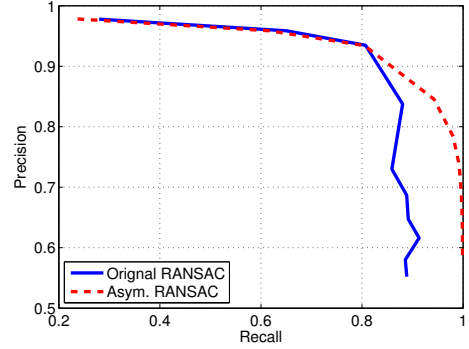

Fig. 3: ROC curves of two algorithms are presented. We retrieved the curves by varying the kernel width, σ, from 0.01 to 0.8. Our asymmetric kernel was assigned as $\sigma_+ = 0.1\sigma$ and $\sigma_- = 10\sigma$. RANSAC also adopted its cutoff threshold as 1.96σ.

3.2 Results and Discussion

The original and our asymmetric RANSAC had almost similar accuracy and computing time. The details are shown in Table 1. Their best accuracy was achieved at $\sigma = 0.09$ and $\sigma = 0.10$, respectively. Even though the asymmetric kernel involved more computation burden, its adaptive termination (Algorithm 1: Line 12) enabled less number of iterations than the original RANSAC so that their computing time was similar.

However, their precision-recall curves reveals robustness of our asymmetric RANSAC. The ROC curves are described in Figure 3. Two curves clearly show that the asymmetric RANSAC had higher recall rate at same precision, especially more than 0.9 recall rate. Figure
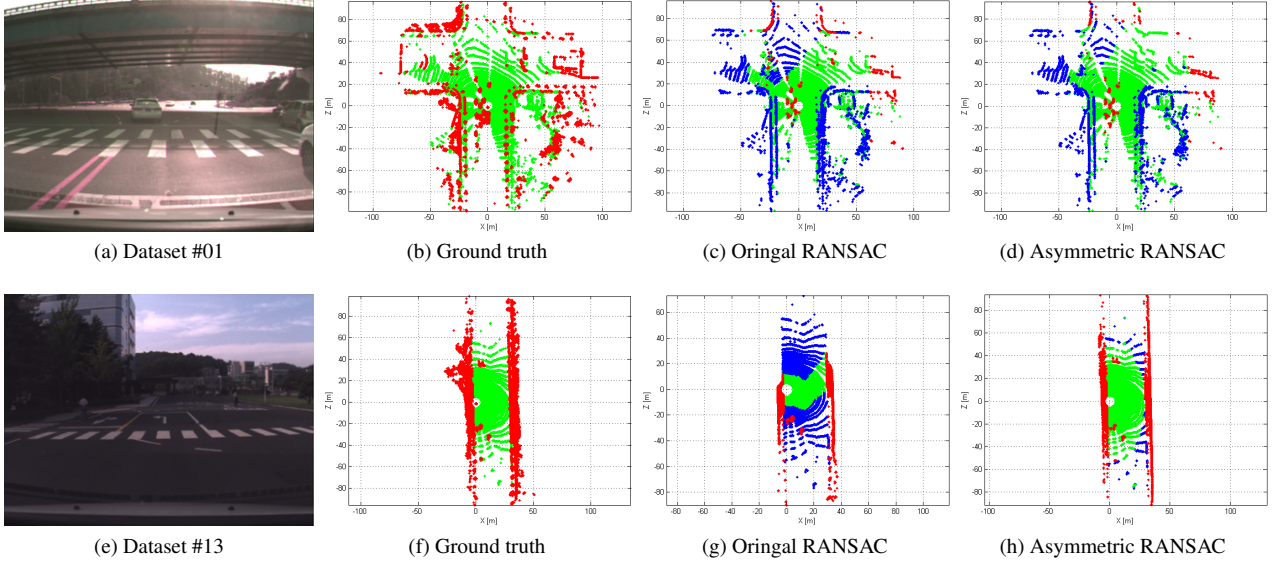
(a) Dataset #01	(b) Ground truth	(c) Oringal RANSAC	(d) Asymmetric RANSAC
(e) Dataset #13	(f) Ground truth	(g) Oringal RANSAC	(h) Asymmetric RANSAC

Fig. 4: Two examples in the 3D LiDAR datasets are presented. Their ground truth was labeled as ground and other objects. Each algorithm's result was described as true positive, false positive, and false negative, respectively. (Note: Images were not utilized in ground detection. They were captured for debugging and visualization.)

4 shows two examples among our 13 point clouds. In the point cloud #01, two algorithms had similar accuracy, precision, and recall. The original RANSAC had (0.91 accuracy, 0.96 precision, 0.89 recall), and asymmetric RANSAC had (0.92, 0.96, 0.92). In contrast, the point cloud #13 had more obstacles such as buildings and bushes, so the asymmetric RANSAC had better performance, especially recall rate. The original RANSAC had (0.76 accuracy, 0.68 precision, **0.77** recall), and our asymmetric RANSAC had (0.81, 0.69, **0.99**).

4. APPLICATION TO AUGMENTED PERCEPTION

We applied our ground plane detection to augmented perception for manual driving and remote control. Our application identifies ground (traversable region) and obstacles, and marks them on the given images. It aims for drivers or remote operators to understand their driving environment clearly and quickly. The augmented perception can make them drive or control their vehicles and robots easily and safely [11]. We applied the augmented perception to stereo images acquired during urban driving. 3D point cloud is generated from the given stereo images using a stereo matching library, LIBELAS [12]. At first, we identify a ground plane through our RANSAC-based detection with the asymmetric kernel. After finding the ground plane, obstacles are easily distinguished because they are a cluster of points on the ground plane. We only consider obstacles on region-of-interest (ROI) to ignore a number of unimportant obstacles. To visualize ground and obstacles, we mark the ground with green and obstacles with red on the images. Figure 5 shows an example of our visualization. Additionally, we present ROI
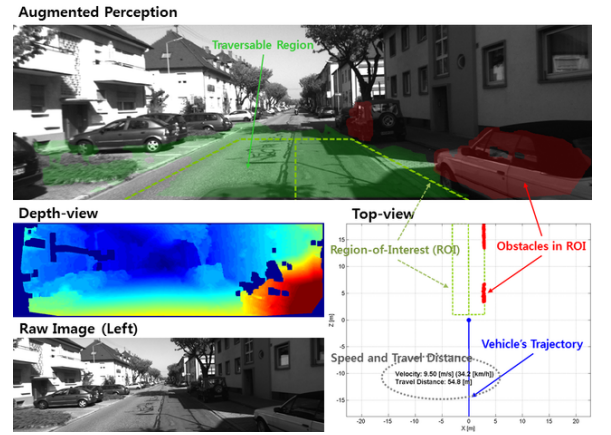

Fig. 5: Our augmented perception visualizes spatial recognition such as traversal regions, obstacles on rectangular ROI. They are also presented in top-view with camera trajectory and vehicle speed. We also show small raw (left) and depth images.

on the plane and provide its top-view with camera trajectory. We utilized a stereo odometry library, LIBVISO2 [13], to acquire camera trajectory. Figure 6 demonstrates a sequence of images on KITTI odometry dataset [14] with our ground plane detection. In the KITTI dataset, we used $\sigma_+ = 0.1$, $\sigma_- = 10\sigma_+$, $\tau = 1.96\sigma_+$, $s_x = 0$, $h = 1.5686$, and $T_{\max} = 100$.

5. CONCLUSION

In this paper, we propose ground detection with the asymmetric kernel. The asymmetric kernel is a fast approximation of likelihood considering inliers and out-
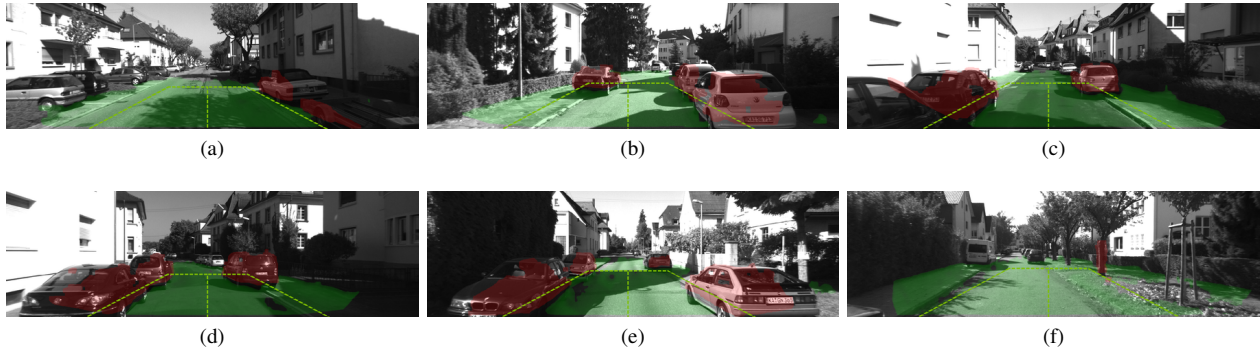
Fig. 6: Our augmented perception was applied to the KITTI odometry dataset. Traversable regions and obstacles on ROI were shown in each color scheme.

liers. It is incorporated with RANSAC as a score function. Compared to the previous symmetric kernels, our asymmetric kernel is more robust because it considers distribution of outliers which exist over the ground plane. We also quantitatively demonstrated that the proposed ground detection robust and fast. In our experiments with the 3D LiDAR datasets, our proposed method had higher recall rate and spent almost 18 milliseconds in MATLAB. Finally, we also applied our proposed ground detection to augmented perception for urban driving situation.

As further works, we will improve our ground detection, especially accuracy. We believe that we can achieve higher accuracy by incorporating a global ground model and local features, together. In the view of sensor fusion, additional visual data will be useful for further improvement.

REFERENCES

[1] D. Conrad and G. N. DeSouza, "Homography-based ground plane detection for mobile robot navigation using a modified EM algorithm," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[2] A. Cherian, V. Morellas, and N. Papanikolopoulos, "Accurate 3D ground plane estimation from a single image," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[3] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. S. C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, pp. 661–692, 2006.

[4] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3D Lidar data in non-flat urban environments using a local convexity criterion," in *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2009.

[5] M. W. McDaniel, T. Nishihata, C. A. Brooks, and K. Iagnemma, "Ground plane identification using LIDAR in forested environments," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[6] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *RoboCup 2011: Robot Soccer World Cup XV*, 2012.

[7] J. Byun, K. in Na, B. su Seo, and M. Roh, "Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle," in *Proceedings of International Conference on Field and Service Robots (FSR)*, 2013.

[8] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with RGB-D data," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[9] S. Choi, J. Park, and W. Yu, "Resolving scale ambiguity for monocular visual odometry," in *Proceedings of International Conference on Ubiquitous Robot and Ambient Intelligence (URAI)*, 2013.

[10] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family," in *Proceedings of British Machine Vision Conference (BMVC)*, 2009.

[11] H. S. Park, M. W. Park, K. H. Won, K.-H. Kim, and S. K. Jung, "In-vehicle ar-hud system to provide driving-safety information," *ETRI Journal*, vol. 35, pp. 1038–1047, 2013.

[12] A. Geiger, M. Roser, and R. Urtasun, "Effient large-scale stereo matching," in *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2010.

[13] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI Vision Benchmark Suite," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.