Multi-foreground objects segmentation based on RGB-D Image

Yan Li[1], Di Zhu[1], Hui Chen*, Haikun Li
School of Information Science and Engineering,
Shandong University,
QingDao, China
huichen@sdu.edu.cn

Changhe Tu
School of Computer Science and Technology ,
Shandong University,
QingDao, China

Abstract

Silhouette extraction of foreground objects appears frequently in various real-world applications, such as Advanced Driving Assistant System, Intelligent Monitoring System, and movie production. Plenty of solutions have been developed to extract silhouette in RGB image with only color information. Since those color based silhouette extraction methods still have difficulties to separate overlapping foreground objects and eliminate excessive segmentation, this paper proposes a novel object segmentation method using color and depth information in RGB-D images. Firstly, we remove the ground plane using the normal map of depth image. Secondly, to separate foreground objects at different distances completely and correctly, the deep Residual Network (ResNets) and Otsu's multi-thresholding method are combined to divide the depth image into multiple layers. Each depth layer contains only one foreground object or objects at same distance. Finally the outline of foreground object is extracted directly from its depth layer, and refined with color information. Experimental results demonstrate that our method has a better performance than those using color or depth information only, and extracts more types of objects than neural networks.

1. Introduction

In the field of computer vision, foreground object segmentation is the basis of many multimedia applications, such as image 3D reconstruction [10], 3D TV display [12], multi-vision naked eye 3D display [15], etc. There have been a lot of researches on segmenting foreground objects in RGB image [5, 23]. Since extracting foreground objects with color information needs to overcome the interference of shadows, color similarity, object occlusion and overlap in application, it is hardly to obtain accurate contour edges. In RGB-D image, objects at different distance has clearly different depth, which can be utilized to solve many problems in RGB image segmentation, such as object occlusion and overlap, etc.

The method presented in [4] segments the color and depth images individually and then merges regions by applying an iterative cooperative scheme. Bleiweiss et al. [3] extend the mean-shift segmentation algorithm by adding depth values to the feature space. Fermandez-Sanchez et al. [11] present a background subtraction technique that fuses depth and color information. Mutto et al. [24] describe scene point with a 6D vector containing geometry and color information; then perform spectral clustering. Hickson et al. [19] propose a multistage graph-based segmentation method, mainly based on depth values. Cong et al. [6] and Jung et al. [22] apply visual attention model to the segmentation task, take salient areas as object seeds. But in cluttered or low visual contrast scenes, the saliency-detection based methods may fail to separate multiple objects from the background. In recent years, neural-network based RGB-D image segmentation methods have received more and more attention. Gupta et al. [14] and He et al. [18] take depth image as a channel into the neural-network used in RGB image segmentation, to realize RGB-D image semantic segmentation. Wang et al. [29] improved the traditional Convolutional Neural Network based on depth value. A novel neural network with two input channels is designed in [20], the input is color and depth information. A graph-based neural-network is proposed in [26] to replace the traditional Convolutional Neural Network, which can make full use of the geometric information of scene. These neural networks designed for RGB-D image segmentation have achieved a breakthrough, but there are still some shortcomings, such as the need for a large number of training datasets to ensure the accuracy of the algorithm, higher requirements for computing capabilities, and can not extract objects that not be trained. In summary, these methods have great limitations in extracting all kinds of foreground objects accurately.

With the development of neural network, the recognition accuracy of current neural-network based method has exceeded the human eye's on ImageNet dataset [9]. Inspired by various image classification networks [17, 21, 28], this paper proposes an adaptive multi-thresholding segmentation method, combining ResNets and extended Otsu method [25], to extract foreground objects of various types completely. The main contributions of this paper are as follows:

- We propose an adaptive multi-thresholding method to divide depth image into multiple layers, each depth layer contains a foreground object or objects at similar distance, except for the depth layer of background;
- As one of the main components of background, the ground plane brings a lot of noise to foreground object segmentation. To improve silhouette extraction of foreground objects, we also propose a method to remove the ground plane;
- To evaluate the performance of current silhouette extraction methods, detailed comparison between state-of-the-art neural-network-based methods and the proposed method are performed on three real-world datasets.

This paper is organized as follows. Section 2 describes the proposed method. Section 3 reports on our experimental results. Section 4 concludes.

2. The proposed method

The workflow of the proposed method for foreground object segmentation is shown in Fig. 1. The whole process is divided into four steps; the first step removes the ground plane based on the depth image (see Section 2.1); the second step divides depth image into multiple depth layers with the proposed adaptive multi-thresholding method (see Section 2.2); the third step uses Mean Shift [5] algorithm to segment color images (see Section 2.3); finally, the over-segmented color regions are merged, under the constraint of depth stratification, to obtain more accurate foreground object contours (see Section 2.4).

2.1. Ground plane removal

Since the color difference between the foreground object and the background is inapparent, the segmentation result usually contains part of the ground plane, especially in the scene that contains most of the ground area such as the stage. With the ground plane being removed, the accuracy of segmentation result will be significantly improved.

We observe that when the camera is parallel to ground plane, pixels at the same row in depth image have the same depth value. Therefore, a perfect ground depth image can be fitted according to the original depth image. Firstly, we project the three-dimensional space (X, Y, Z) to (Y, Z) plane, where x, y are the horizontal and longitudinal ordinate of image coordinate system respectively, z shows the

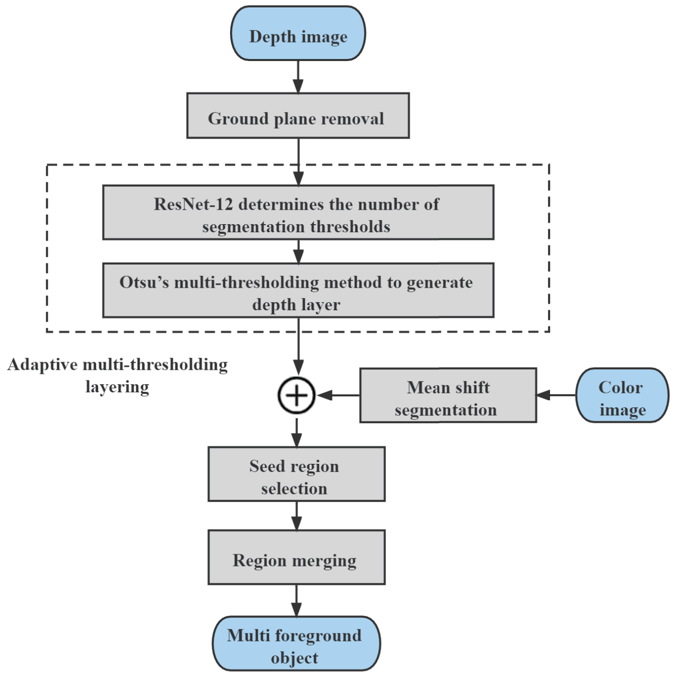
Depth image
Ground plane removal
ResNet-12 determines the number of segmentation thresholds
Otsu's multi-thresholding method to generate depth layer
Adaptive multi-thresholding layering
Mean shift segmentation
Color image
Seed region selection
Region merging
Multi foreground object

Figure 1. The proposed method.

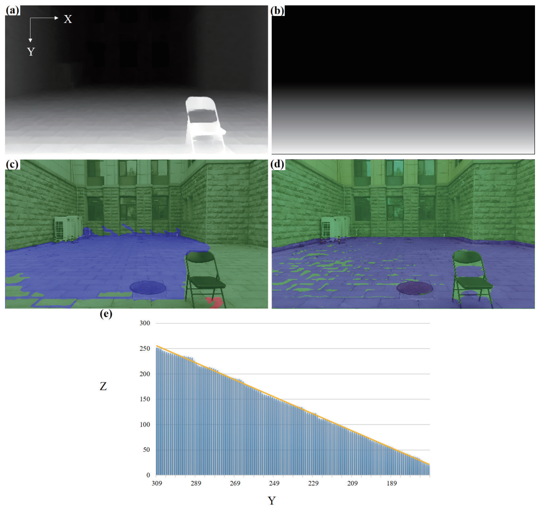

Figure 2. (a) Depth image; (b) The fitted ground plane image; (c) The result of removing the ground plane by using Yuan's [33] method; (d) The result of removing the ground plane by using our method; (e) $Y - Z$ plane projection and fitting curve.

depth. Secondly, select the smallest pixel value in each row as the depth value of this row, as shown in Fig. 2(e) (the value on each blue column represents the minimum depth value selected in each row). Thirdly, according to the linear relationship between the horizontal coordinate y of the pixel and the corresponding depth value z, the depth image of the fitted ground plane can be obtained (see the yellow line in Fig. 2(b)). Lastly, find the ground pixels in the original depth map. Our criterion is: if the difference of pixel value between the ground plane image and depth image within a fixed range, this pixel is considered to belong to the ground plane. Fig. 3 is the ground removal results under the fixed range of 10, 15, and 25, respectively. With a fixed range of 10, the depth image still contains large patches of ground, while with a fixed range of 25, too many pixels in the foreground objects (like chair legs and this person's feet) are removed. Experiments demonstrate that 15 is a suitable range, the fixed range is set to 15 in this paper, which can effectively remove most of the ground plane (see Fig. 3).

Fig. 2(c) is the results of the proposed ground plane removal method and Fig. 2(d) is the results of Yuan's [33] method. Yuan et al. [33] used the normal map of depth image to remove ground plane, as shown in Fig. 2(c), according to the fact that pixels belonging to the same plane have the same normal vector. We can see Yuan's method cannot remove the ground plane within the chair, which makes it difficult to accurately extract the contour of the chair. However, our method can remove most of the ground plane in the entire scene, especially within the chair, which improves the performance of the proposed object segmentation method significantly. Due to the complexity of image scenes, our method is not suitable for all scenes. The proposed method only works on relatively flat ground and the scene does not contain cluttered object arrangements.

2.2. Depth image layering

Depth image provides significant clues for image segmentation, and usually contains information about the contour or position of foreground objects. According to the depth image, we can effectively separate foreground objects at different distances and obtain the approximate contour of objects.

Depth image can be obtained by stereo vision [23] or time-of-flight (TOF) camera [7]. Fig. 4(a) is a depth image and its histogram, captured by ZED Stereo Camera 2, containing multiple objects. We can see that there are obvious peaks and valleys in the histogram, and we can use different thresholds to divide the depth image into different depth layers. Fig. 4(b), 4(c), 4(d) show the segmentation results using 1, 2, and 3 thresholds respectively. Fig. 4(b) shows that if the depth image is divided into a foreground depth layer and a background depth layer with one threshold, the person and the background are integrated and difficult to be separated. Using three thresholds to divide the depth map into four layers, the umbrella is separated into two depth layers, as shown in Fig. 4(d). Fig. 4(c) shows the result of using two segmentation thresholds, which not only divides multiple foreground objects into different depth layers, but also retains the complete edge information of foreground objects. Therefore, the number and value of segmentation thresholds have a great influence on foreground object segmentation. To accurately segment all the foreground objects, we propose an adaptive multi-thresholding segmentation method to divides the depth image into multiple layers. The proposed method is achieved by two steps; the first step uses Residual Networks (ResNets) to determine the number of segmentation thresholds of the depth image (see Section 2.2.1); the second step determines the values of segmentation thresholds with the extended Otsu's multi-thresholding method [8] (see Section 2.2.2). With the depth image divided into multiple layers, foreground objects at different distance and background are separated completely.

2.2.1 ResNets determine the number of thresholds

The neural network mainly learns features from the data to improve the prediction accuracy, so we first make a dataset according to the common thinking of human beings to layer the depth map. The training dataset of ResNets is classified according to the number of foreground objects as well as their position relationship. As shown in Fig. 5, although there are two chairs and one person in each depth image, the distance between them is different and the depth images are divided into different depth layers. In Fig. 5(c), the depth values of all foreground objects are similar, and the histogram also has only one peak (as shown in Fig. 5(f)). So the depth image can be divided into a foreground layer and a background layer, and the classification label of the image is 2. Fig. 5(b) shows two chairs are at the same distance from the camera, and the person is farther away from the camera. There are two obvious peaks in the histogram (as shown in Fig. 5(e)), and two segmentation thresholds are required to divide the depth image into three layers. So the classification label of Fig. 5(b) is 3. Similarly, the classification label of Fig. 5(a) is 4.

Deep Convolutional Neural Networks have made a series of breakthroughs in image classification. However, with the number of layers of Convolutional Neural Network growing, the gradient disappears and explodes. For this reason, He et al. [17] proposed a Residual Network (ResNets). ResNets has a good performance in the field of image classification. The classic structure of ResNet includes ResNet18, ResNet50 and ResNet101. Our training dataset consists of 4,099 depth images, and is divided into five categories, each of which has about 800 depth images. Since over-fitting occurs in RestNet18 due to the insufficient datasets, we change the number of network layers and propose RestNet12 model. The schematic diagram of RestNet12 structure is shown in Fig. 6.

In order to improve the accuracy and reduce over-fitting, a regularization is added after each convolutional layer, the

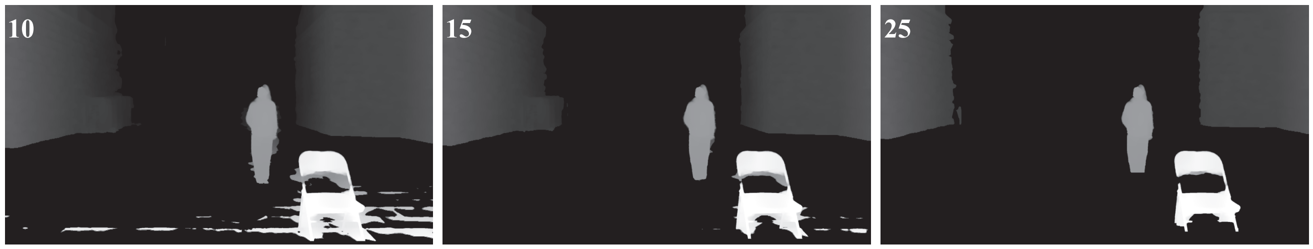

Figure 3. The ground removal results under the fixed range of 10, 15, and 25

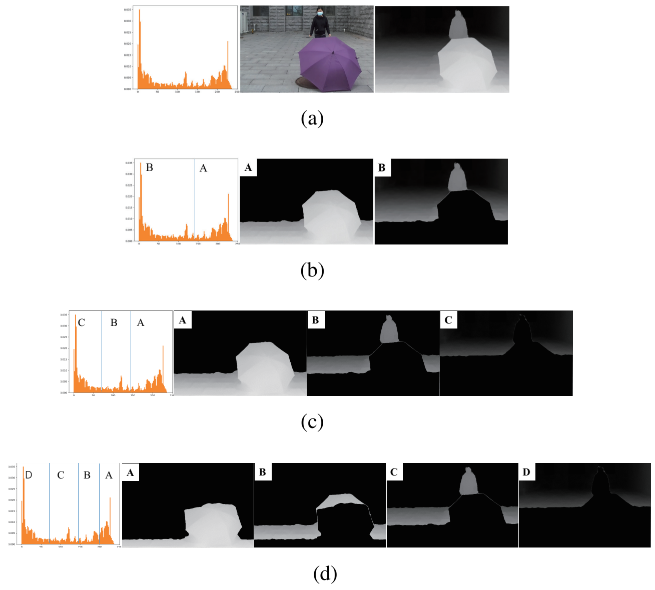

Figure 4. Multiple depth layers of depth image obtained by extended Otsu's multi-thresholding method [8]. (a) Depth image and its histogram; (b) Two depth layers generated with one threshold; (c) Three depth layers with two thresholds; (d) Four depth layers with three thresholds.

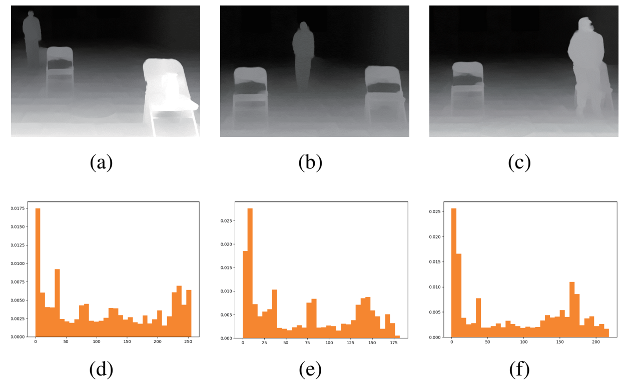

Figure 5. Depth image and histogram. (a)∼ (d) depth map; (d)∼(f) histogram.

value is 0.0005, and the dropout is increased after the fully connected layer, the value is 0.0005. The experimental re-

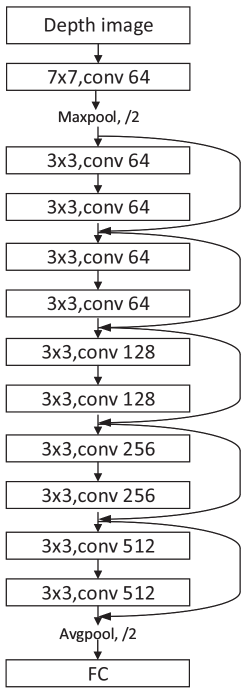

Figure 6. RestNet12 network structure diagram.

sults and more experimental details will be illustrated in Section 3.

2.2.2 Otsu's determine the value of thresholds

The essence of layering the depth image is to divide the depth value into multiple intervals based on different thresholds. In the field of computer vision and image segmentation, the Otsu method is often used to find the best segmentation threshold of an image. We use the extended Otsu method [8] to divide depth image into multiple layers, the implementation of the extended algorithm is as follows.

Assuming that a depth image of size $W * H$ contains L-level pixel gray values, we count the number of pixels with gray value $i \in [0, L-1]$, and calculate the distribution probability $p(i)$ of pixels with gray value equal to i, the cumulative distribution function of the gray value i is obtained:

$$F(i) = \sum_{k=0}^{i} p(i) \tag{1}$$

Assuming that the depth image is classified as N, the gray values needs to be divided into N intervals, where the pixel value of the nth interval should belong to the closed space $[T_{n-1},T_n]$, where $n \in [1, N]$, $T_0 = 0$ and $T_n = L$, the weight of the nth interval is:

$$\omega(T_n) = \begin{cases} F(T_n) - F(T_{n-1}) & 0 < T_n < L \\ F(0) & T_n = 0 \end{cases} \tag{2}$$

The mean value of each category can be calculated:

$$\mu(T_n) = \frac{1}{\omega(T_n)} \sum_{i=T_{n-1}}^{T_n} ip(i) \tag{3}$$

Finally calculate the variance between classes:

$$\sigma^2(N) = \sum_{n=1}^{N} \omega(T_n)(\mu(T_n) - \mu)^2 \tag{4}$$

Where u is the global mean of the image, defined as:

$$\mu = \sum_{i=0}^{L-1} ip(i) \tag{5}$$

Calculate the final optimal segmentation threshold set by maximizing the variance between classes:

$$\sigma^2_{max}(N) = \max(\sum_{n=1}^{N} \omega(T^*_n)(\mu(T^*_n) - \mu)^2) \tag{6}$$

Eq. 7 shows how to segment a depth map into layers with the optimal segmentation threshold:

$$L_m(i,j) = \begin{cases} F(T_n) - F(T_{n-1}) & \text{if } T_m < D(i,j) < T_{m+1} \\ F(0) & \text{otherwise} \end{cases} \tag{7}$$

where $D(i, j)$ is the depth value at pixel (i, j), and T_m, for $0 \leq m \leq n-1$, is the mth threshold computed by an extended Otsu's multi-threshold method.

The results of adaptive multi-thresholding algorithm is shown in Fig. 7(c)(d), and the white areas represent the grouped pixels. In the depth image, it is difficult to obtain accurate contour information only by segmenting the depth image. Although the foreground object is not completely extracted in this step, the result provides a rough outline and provides a basis for merging over-segmented color regions.

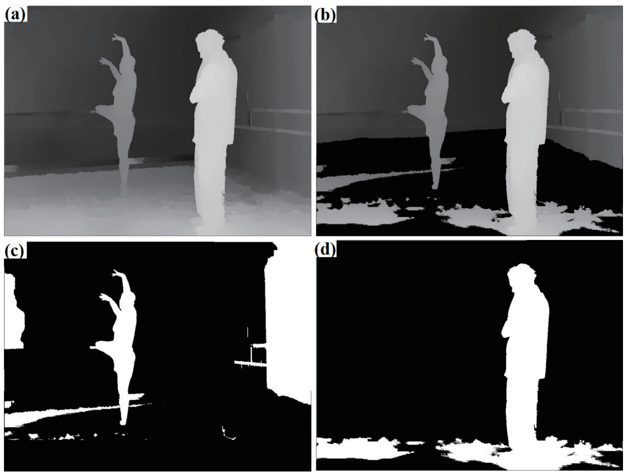

Figure 7. The result image of adaptive segmentation. (a) Depth image; (b) Depth image after removing the ground plane; (c) First layer image; (d) Second layer image.

The results of using other methods to divide the depth layer are shown in Fig. 8; the left image is the result of using the Mean shift algorithm [5], and the right image is the result of using the method [10]. In both cases, the depth image is divided into more segments than expected. Compared with Fig. 7, our method provides a better initial contour than the other two methods.

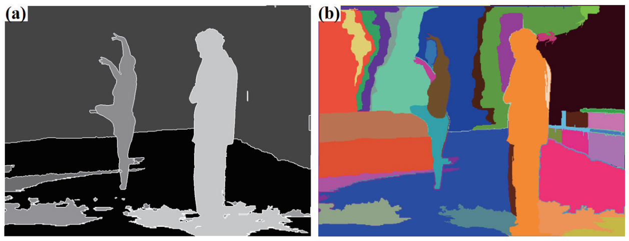

Figure 8. (a) Mean shift segmentation; (b) Graph-based segmentation [10].

2.3. Color image segmentation

We use Mean shift [5] to segment color image in $L^*u^*v^*$ color space. Among various segmentation algorithms, Mean shift is widely used due to its performance. Fig. 9 shows the regions of similar color obtained by Mean shift.

In Fig. 9, chairs, kettles and people are foreground objects. The parts with similar colors are divided into the same area (see Fig. 9(b)), and there are obvious boundaries between the areas of different colors (such as the handle and the body of the kettle). Although the color of the chair itself is similar, the edges and some small areas are still divided into several small areas. Generally, a foreground object does not contain only one color, so an object in the

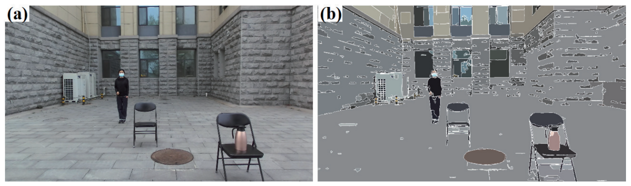

Figure 9. Color image segmentation with Mean shift. (a) original color image; (b) segmentation result.

Mean shift segmentation result is often divided into multiple regions, which makes it difficult to completely extract a foreground object with an accurate contour using only the Mean shift algorithm.

2.4. Region Merging

The segmentation of the color image suffers from over-segmentation, and it is difficult to fully identify semantic objects. Region merging is a common way to solve this problem. Merging criterions typically define some kind of criterion for adjacent regions to determine whether to merge adjacent regions[32].

In recorded 3D scenes, the background and foreground have basically different colors. We usually observed that foreground objects are usually comprised of smaller segments obtained by segmenting the color image, while there are larger segments in the background. Based on this observation, we propose an intuitive merging rule which uses segment size and depth grouping. Adjacent color segments contained in the same depth grouping are assigned to the same seed if their area is smaller than a threshold. This avoids merging of relatively large segments (such as the floor and wall) into a foreground object. The goal is to ensure that the merging process results in a semantic object.

Let $D_i, i = 1, 2, \dots, m$, be m grouped regions of the depth image,and $C_j, j = 1, 2, \dots, n$,n segments of the color image. Segments C_i which are selected as being seed regions, denoted by $H_k, k = 1, 2, \dots, s$. Note that each H_k is also a C_j, for some j.

Our merging algorithm is summarised in Algorithm. 1. Threshold T is determined by statistical analysis of the area of each segment in a given color image; in order to avoid merging large background segments with the foreground objects, we use the average of the five largest regions as the threshold T. Each seed region H_k will generate a corresponding object region F_k. The final output is the family of all these sets F_k.

The effect after region merging is shown in Fig. 10. Fig. 10(a) is the result of segmentation of the color image by Mean shift; Fig. 10(b) is the result of region merging using only area constraints, including many background regions. Fig. 10(c) shows the result of region merging using depth layering constraint (the depth image does not remove

Algorithm 1 Region merging algorithm.

```
Input: Depth regions $\{D_i \mid i = 1, 2, \dots, m\}$, color segments $\{C_j \mid j = 1, 2, \dots, n\}$, and seed regions $\{H_k \mid k = 1, 2, \dots, s\}$.
Output: Object regions $\{F_k \mid k = 1, 2, \dots, s\}$
1: for every seed region $H_k$ in the color image do
2:     Select the depth region $D_j$ in the depth image which contains $H_k$.
3:     Color segments contained in $D_j$ are selected as candidate regions for merging. These regions are denoted by $S_i$, for $i = 1, 2, \dots, u$.
4:     $R_{seed} = H_k, R_{set} = S_i :, i = 1, 2, \dots, u$.
5:     If $R_{set}$ contains a region $S_i$ which is adjacent to $R_{seed}$, and the area of $S_i$ is smaller than threshold $T$, then merge $S_i$ and $R_{seed}$ into a new seed region $R_{seed}$, and delete this $S_i$ from $R_{set}$.
6:     Repeat step 4 until there is no region in $R_{set}$ satisfying both conditions to be adjacent to $R_{seed}$ and area smaller than $T$.
7:     $F_k = R_{seed}$
8: end for
```

the road surface) and area constraint. The manhole cover is mistakenly regarded as the foreground object, and there is some noise at the bottom of the umbrella. Fig. 10(d) is the result of foreground object segmentation obtained by our method. Road areas such as manhole covers and umbrella bottoms have been removed, and the contour edges are accurate.

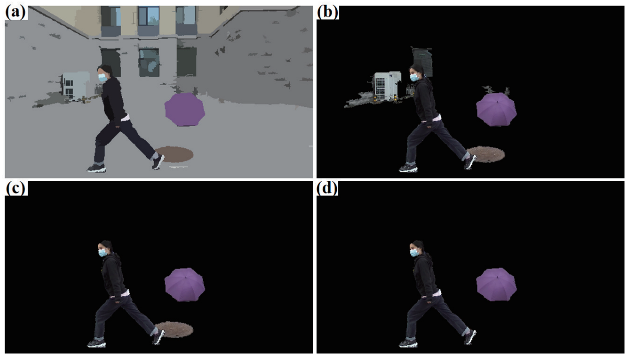

Figure 10. Comparison of region merging results. (a) RGB image after meanshift segmentation; (b) merged result using area constraint only; (c) merged result using area constraint and depth layering constraint (the depth image does not remove the road surface); (d) our result.

3. Experiments and discussion

We tested the proposed multi-foreground objects segmentation method on three real-world datasets, including a self-collected dataset, KITTI[13] and MSR 3D Dataset[2]. Foreground objects in the self-collected dataset include human being, umbrella, kettle, chair and bag. ZED Stereo Camera 2 is used to capture RGB-D images of the self-collected dataset, which can accurately obtain distance of objects within 0.5 to 20 meters. The proposed method is programmed in Matlab R2018a, on a PC with Intel Core i5-3770 3.40GHz CPU and 8GB RAM. Additionally, to demonstrate the performance of the proposed multi-foreground objects segmentation method, we compare the proposed method with Grabcut [27] and three neural-network based methods on three real-world datasets.

3.1. ResNet12 experiment results

The training dataset of ResNet12 contains 4,099 depth images from ApolloScape Stereo Dataset[1], KITTI Dataset[13], MSR 3D Dataset[2], and self-collected dataset. Our dataset contains 5 categories, where category 1 represents that there is no foreground object in the image, and category 5 represents that the depth image can be divided into five layers. Each category contains approximately 800 depth images. We use the histogram of the depth image as a reference, and use the number and position relationship of foreground objects to determine the classification criteria. We randomly generate training set, validation set, and test set from our dataset according to the ratio of $7 : 2 : 1$. Finally, the best performing model on the test set is selected to determine how many layers the depth image is divided into.

The comparison results of different ResNets models are shown in Fig. 11. On the validation set, the accuracy of all models is above 80%. The model with the highest accuracy on the test set is ResNet-12, with an accuracy rate of up to 83.15%. This shows that the ResNet-12 model can correctly predict how many layers the depth image can be divided into to a certain extent (only suitable for simple scenes, such as several cars far apart on the highway).

3.2. Qualitative analysis

Our approach was evaluated mainly on a large-scale hierarchical multi-view RGBD object dataset collected using a ZED camera. The MSR 3D Video Dataset (MSR 3D), and the public KITTI stereo 2015 Dataset, were also used to test our method. Due to the different sources of depth information in these datasets, all the original depth Images are uniformly normalized into the same range of $[0, 255]$.

To evaluate the performance of our method, we also compare it with some instance segmentation methods, such as Grabcut[27], Mask R-CNN[16], SOLO[30] and

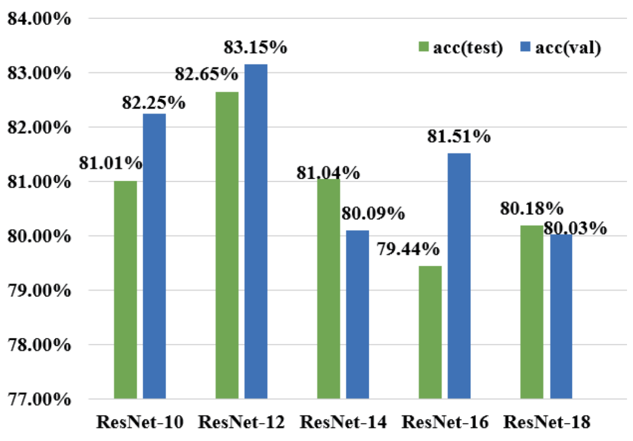

Figure 11. Experimental results of ResNets.

SOLOv2[31]. In the comparative experiments between SOLO and SOLOv2, we directly use the highest-precision models provided by the official: $solo-r101-fpn-8gpu-3x$ and $solov2-x101-dcn-fpn-8gpu-3x$. The comparison results on KITTI dataset and MSR dataset are shown in Fig. 12, and comparison results on the self-collected dataset are shown in Fig. 13.

Comparison results show that all methods can roughly segment multiple foreground objects. Fig. 12(a4)−(d4) are the segmentation results of Grabcut method. When the color difference between the foreground object and the background area is obvious, the segmentation effect is well, but when the colors are similar, it is difficult to accurately segment the foreground object by using Grabcut. In Fig. 12(b4), there are holes in brighter colors such as the license plate, and the segmentation results in Fig. 12(c4) and (d4) contain some walls. In contrast, our method takes into account the depth-aware layered information, which is more effective for such easily-confused scenes. Furthermore, we implement the Mask-RCNN,SOLOand SOLOv2 (see Fig. 12(5)(6)(7)) for comparison, which are object detection and instance segmentation frameworks based on deep learning. Although Mask R-CNN preserves spatial locations by modifying RoIPool to RoIAlign layer, the mask misaligns the ground-true silhouette due to dimension reduction and information loss. As shown in Fig. 12(b7)(d7), a part of the tail light on the right side is missing and accurate information of the ballet dancer's arm silhouette cannot be obtained. Fig. 12(5)(6) show the segmentation results of SOLO and SOLOv2, the segmentation accuracy is significantly better than Mask R-CNN, especially for large objects. However, there are still obvious flaws in the tiny details. As shown in Fig. 12(c5)(c6), all the fingers of the ballet dancer are divided into a whole. And there are redundant background pixels between the arms and the head in Fig. 12(d5)(d6). Our method retains more precise and de-

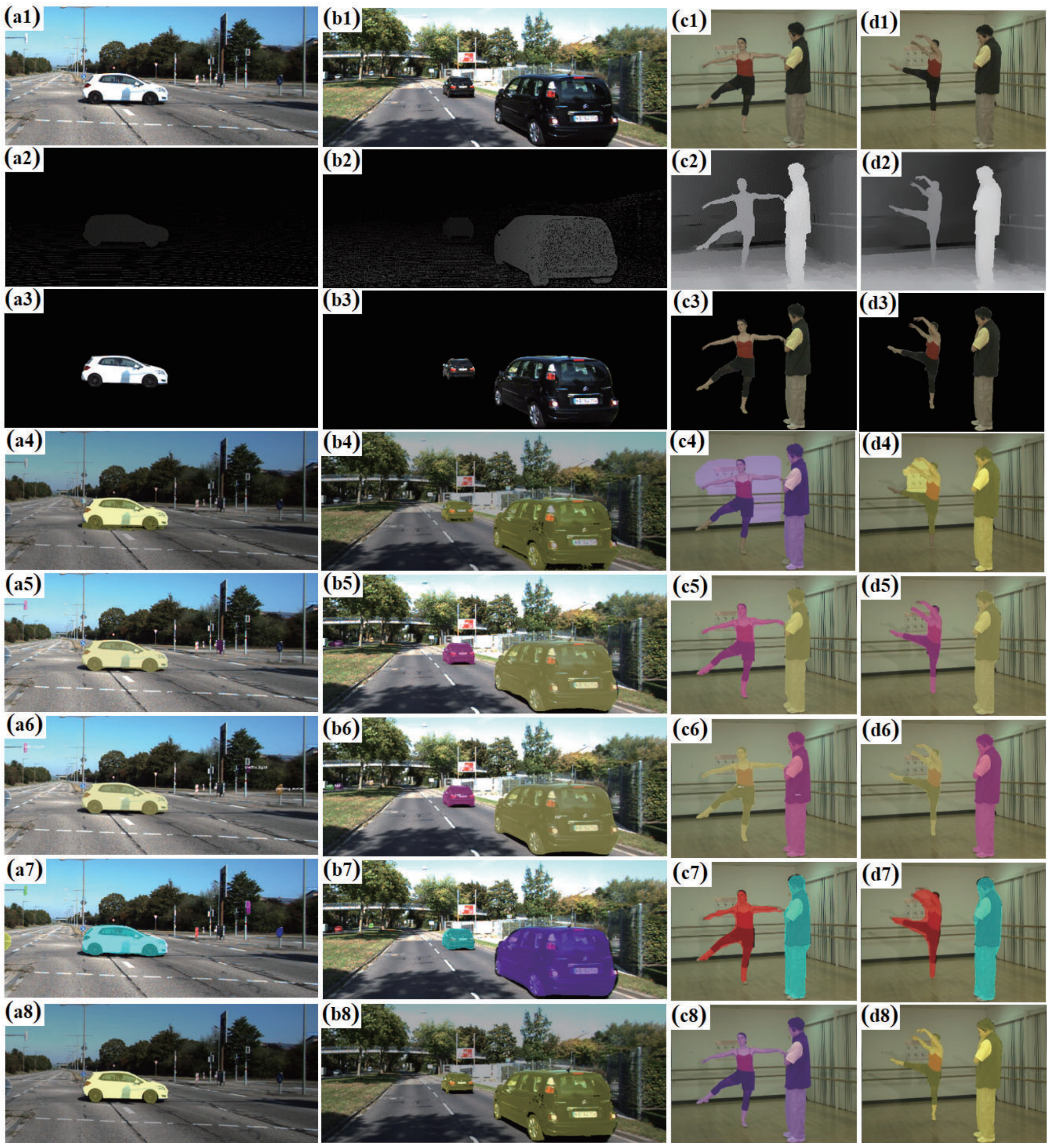

Figure 12. Foreground objects extracted by different methods on KIITTI dataset and MSR 3D datdaset. (1) Input color image, (2) Depth image, (3) Ground truth, (4) Grabcut[27], (5) SOLO, (6) SOLOv2 , (7) Mask R-CNN and (8) the proposed method.

tailed object silhouettes, such as the bottom edge of the car in Fig. 12(a8), the dancer's fingers and the coach's hair in Fig. 12(c8)(d8), etc. In small scenes, our method can still segment accurate results. In summary, our method can extract relatively accurate foreground object silhouettes from different background scenes.

Fig. 13 is a comparison of the segmentation results in the dataset taken by the ZED binocular camera. In

Fig. 13(a4)−(a7), the segmentation results of the chairs contain a lot of road surface information, and our method (Fig. 13(a8)) removed most of the road surface information through the step in Section 2.1, such as the road surface in the middle of the chair. Therefore, in this scenario, the segmentation results of our method are significantly better than the other four methods. Since an absolutely accurate depth image cannot be obtained from the binocular image, the foreground object in Fig. 13(a8) still contains a small area of the road surface. If the depth information is very accurate (for example, using a radar device to generate a depth image), the algorithm in this paper can completely remove the road surface. Fig. 13(c), SOLO and SOLOv2 failed to identify kettle without dedicated retraining. This is a major drawback of neural networks. If the training dataset does not contain this category, it cannot be detected. So we made a new dataset containing 100 images of kettle, and trained on the basis of the officially provided model to correctly segment. But because the dataset is too small, the segmentation edges are not very accurate. In addition, the Grabcut method is difficult to segment the foreground objects that are similar to the background color, such as the satchel in Fig. 13(c4). In Fig. 13(b)(d), all the methods have achieved good segmentation results, but our method is significantly better than other algorithms in detail processing.

3.3. Quantitative analysis

The quantitative analysis of comparison results is shown in Table. 1 and Table. 2. The segmentation accuracy consists of three metrics including $Precision$ in Eq. 8, $Recall$ in Eq. 9, and $F-measure$ in Eq. 10, where T_P is the number of correctly detected foreground person pixels, F_P denotes the background pixels which are wrongly classified as foreground pixels, and F_n indicates the undetected foreground pixels.F_β considers both the precision P and the recall R to compute the score. F1-score is the harmonic mean of precision and recall where β is set to 1.

$$P = \frac{T_P}{T_P + F_P} \tag{8}$$

$$R = \frac{T_P}{T_P + F_n} \tag{9}$$

$$F_\beta = \frac{P(\beta^2 + 1)R}{\beta^2 P + R} \tag{10}$$

We extract ground truth manually to evaluate the results.For the segmentation results of Mask-RCNN, SOLO, and SOLOv2, only the foreground objects we determined are extracted for comparison. Quantitative results in Table 1 and Tabel 2 show that the proposed method achieves more accurate extraction than other methods. The precision rate, recall rate, and F-measure value are 0.26%, 4.06%, and 2.03% higher than the most popular SOLOv2 method in Table 2. For the KITTI and MSR 3D dataset, the result of the SOLO method is approximate to ours, while the recall value in KITTI dataset is slightly higher, being 94.41%. And our method is significantly better than Grabcut and Mask R-CNN, both in the public dataset and our dataset.

3.4. Limitations of the proposed approach

Despite the proposed model reaching reliable performances in typical scenes, there are still some limitations. Firstly, the algorithm proposed in this paper relies on the depth map to remove the ground, but only works on relatively flat surfaces. And this step has high requirements on the quality of the depth map, but the development of stereo matching algorithm can facilitate this requirement. Secondly, the generation of color region using Mean shift consumes most of the time in the proposed algorithm and needs to be optimized and accelerated to save the computing resources. Thirdly, our method is based on the assumption that the background is of rather uniform texture, which means that it does not vary “very much” in color and that it can be segmented into large regions. In the case of scenes with a much-cluttered background, our method may fail to extract correct foreground objects. They are the challenges for the proposed algorithm, which are also the further work to improve and overcome.

4. Conclusions

This paper proposes a 2.5D multi-foreground object segmentation method. For a given color and depth image, first, we use the depth image road surface removal method, which effectively removes most of the road surface area. Then we proposed an adaptive multi-thresholding segmentation method to divide foreground objects at different distance into different depth layers. Finally, combined with color image, multiple foreground objects are extracted from the background at the pixel level.

A self-collected dataset, KITTI Dataset and MSR 3D Dataset have been used for testing the proposed method. Besides, we show the performance of the proposed method by comparing it with Grabcut and three state-of-the-art neural-networks. Comparison results demonstrate that our method has a better performance than Grabcut, Mask-RCNN, SOLO, SOLOv2, and can extract more types of objects.

Acknowledgment

This work was supported by the Key Project of Science and Technology Development Plan of Shandong Province of China under Grant No. 2019GGX101018. Thanks also goes to Dr. Weiping Huang and the Foundation of Hisense.

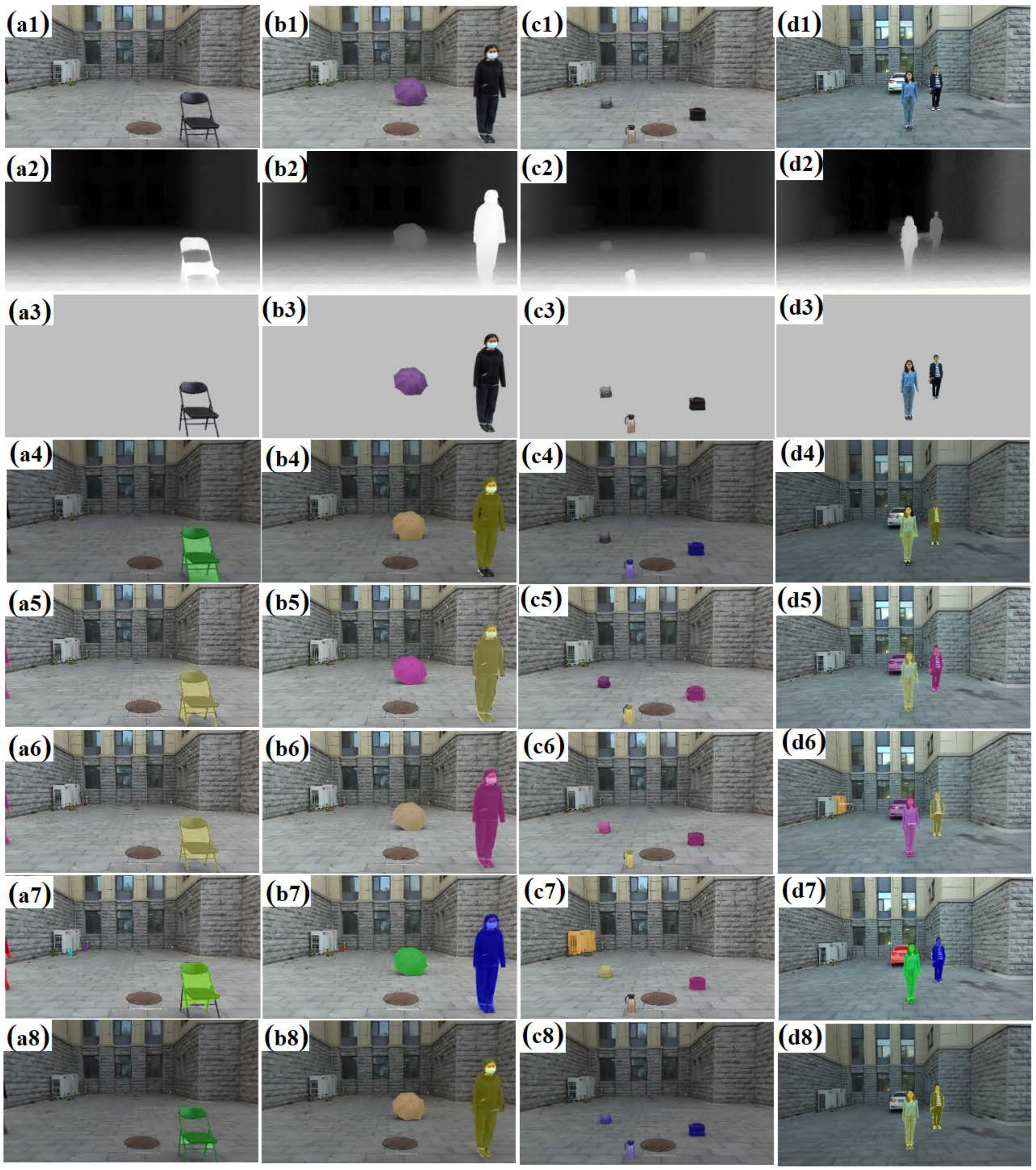

Figure 13. Foreground objects extracted by different methods on our datdaset. (1) Input color image, (2) Depth image, (3) Ground truth,(4) Grabcut[27], (5) SOLO, (6) SOLOv2, (7) Mask R-CNN and (8) the proposed method.

References

[1] Apolloscape stereo dataset. http://apolloscape.auto/stereo.html. 7

[2] Msr 3d video dataset. research.microsoft.com/en-us/downloads/

Table 1. Quantitative analysis results on KITTI and MSR 3D dataset.

Method	KITTI			MSR 3D		
	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Grabcut[27]	67.37%	70.56%	68.40%	73.1%	71.04%	71.90%
MASK R-CNN[16]	80.44%	88.19%	81.55%	86.95%	83.53%	85.20%
SOLO[30]	92.90%	94.24%	92.68%	86.34%	95.20%	90.14%
SOLOv2[31]	94.32%	**94.41**%	93.62%	88.53%	98.24%	92.81%
Proposed method	**94.62**%	93.97%	**93.99**%	**96.48**%	**99.68**%	**98.05**%

Table 2. Quantitative analysis results on Our dataset.

Method	Our dataset		
	Precision	Recall	F1-Measure
Grabcut[27]	70.53%	60.60%	63.88%
MASK R-CNN[16]	93.20%	73.03%	79.91%
SOLO[30]	93.46%	85.19%	88.67%
SOLOv2[31]	93.89%	91.21%	92.56%
Proposed method	**94.15**%	**95.27**%	**94.59**%

5e4675af-03f4-4b16-b3bc-a85c5bafb21d. 7

[3] A. Bleiweiss and M. Werman. Fusing time-of-flight depth and color for real-time segmentation and tracking. In *Dynamic 3D Imaging, DAGM 2009 Workshop, Dyn3D 2009, Jena, Germany, September 9, 2009. Proceedings*, 2009. 1

[4] F. Calderero and F. Marqués. Hierarchical fusion of color and depth information at partition level by cooperative region merging. In *IEEE International Conference on Acoustics*, 2009. 1

[5] D. Comaniciu and P. Meer. Meer, p.: Mean shift: A robust approach toward feature space analysis. ieee transactions on pattern analysis and machine intelligence 24(5), 603-619. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. 1, 2, 5

[6] R. Cong, J. Lei, H. Fu, Q. Huang, X. Cao, and C. Hou. Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation. *IEEE Trans Image Process*, PP(2):1–1, 2018. 1

[7] L. Cruz, D. Lucio, and L. Velho. Kinect and rgbd images: Challenges and applications. In *Graphics, Patterns & Images Tutorials*, 2012. 3

[8] S. Deepa and V. S. Bharathi. Efficient roi segmentation of digital mammogram images using otsu's n thresholding method. *Esrsa Publications*, 2013. 3, 4

[9] J. Deng. Imagenet : A large-scale hierarchical image database. *Proc. CVPR, 2009*, 2009. 1

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Effcient graph-based image segmentation. 2004. 1, 5

[11] E. J. Fernandez-Sanchez, L. Rubio, J. Diaz, and E. Ros. Background subtraction model based on color and depth cues. *Machine Vision & Applications*, 25(5):1211–1225, 2014. 1

[12] Y. Gao, C. Hui, W. Gao, and T. Vaudrey. Virtual view synthesis based on dibr and image inpainting. page 172, 2014. 1

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2012. 7

[14] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, 2014. 1

[15] D. Han, H. Chen, C. Tu, and Y. Xu. View synthesis using foreground object extraction for disparity control and image inpainting. *Journal of Visual Communication and Image Representation*, 56(OCT.):287–295, 2018. 1

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2017. 7, 11

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE*, 2016. 2, 3

[18] Y. He, W. C. Chiu, M. Keuper, and M. Fritz. Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. In *IEEE*, 2016. 1

[19] S. Hickson, S. Birchfield, I. Essa, and H. Christensen. Efficient hierarchical graph-based segmentation of rgbd videos. *IEEE*, 2018. 1

[20] J. Jiang, L. Zheng, F. Luo, and Z. Zhang. Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. 2018. 1

[21] H. Jie, S. Li, S. Gang, and S. Albanie. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2017. 2

[22] C. Jung and C. Kim. A unified spectral-domain approach for saliency detection and its application to automatic object segmentation. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 21(3):1272–83, 2012. 1

[23] R. Klette. *Concise Computer Vision - An Introduction into Theory and Algorithms*. Concise Computer Vision: An Introduction into Theory and Algorithms, 2014. 1, 3

[24] C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo. Fusion of geometry and color information for scene segmentation. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):505–521, 2012. 1

[25] N. Otsu. Threshold selection method from gray-level histograms, ieee transactions on systems man and cybernetics. *IEEE Trans.syst.man Cybern*, 9, 1979. 2

[26] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[27] C. Rother. Grabcut : interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23, 2004. 7, 8, 10, 11

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014. 2

[29] W. Wang and U. Neumann. Depth-aware cnn for rgb-d segmentation. 2018. 1

[30] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li. Solo: Segmenting objects by locations. 2020. 7, 11

[31] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen. Solov2: Dynamic and fast instance segmentation. 2020. 7, 11

[32] A. Wong and J. Scharcanski. Fisher-tippett region-merging approach to transrectal ultrasound prostate lesion segmentation. *IEEE Transactions on Information Technology in Biomedicine A Publication of the IEEE Engineering in Medicine & Biology Society*, 15(6):900–7, 2011. 6

[33] C. Yuan, H. Chen, J. Liu, D. Zhu, and Y. Xu. Robust lane detection for complicated road environment based on normal map. *IEEE Access*, PP(99):1–1, 2018. 2, 3