Long-Range Outdoor Depth estimation Using Perspective Information

Yifan Zhu
TMCC, Nankai University
zhuyifan@mail.nankai.edu.cn

Bo Ren†
TMCC, Nankai University
rb@nankai.edu.cn

Abstract

Learning-based monocular depth estimation algorithms have achieved good performance in indoor scenes and outdoor close-up scenes. However, such algorithms cannot distinguish the depth of distant objects from the background in outdoor scenes. This paper proposes a novel monocular depth estimation method to predict the depth of distant objects in outdoor scenes. Our algorithm can be divided into three parts. First, the object-level depth is calculate through semantic information and the perspective relationship of the scene. Subsequently, the object-level depth will be merged into the depth map generated by the monocular depth estimation network(MDE network). Finally, the semantic segmentation results is used to conduct guided filtering on the merged depth map, to enhance its edge information, and smooth the depth of the object mask. The experiment's visual results and quantitative analysis show our algorithm can recover the depth of distant objects in outdoor scenes compared with the existing methods.

1. Introduction

Depth information and RGB information are essential for humans to understand 3D space. Moreover, depth information plays an essential role in computer vision fields, such as augmented reality, autonomous driving cars, and robotics navigation. The core algorithm of such applications needs to take depth information of the scene as input.

Many devices can obtain the scene's depth information directly from the physical measurement. The most commonly used devices are lidar and RGB-D cameras. However, these devices are usually expensive and difficult to equip. Binocular cameras can also be used for depth estimation. However, the calculation complexity of the binocular image algorithm is high, and the matching effect for low-texture scenes is poor.

Using the monocular camera to estimate the environment's depth is more convenient and cheaper. With these advantages, the interest in monocular depth estimation has been significantly increased in recent years.

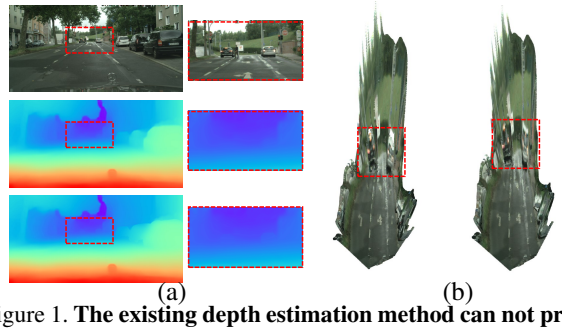

Figure 1. **The existing depth estimation method can not predict the proper depth of distant objects in outdoor scenes.** (a)The image from top to bottom is the input RGB image, the result of [23], and the result of our algorithm. (b)The left image is the 3D view result of our method, and the right image is the 3D view result of the [23]. The existing methods have poor results when estimating the depth of distant objects, but our method can solve this problem well.

The existing monocular depth estimation methods show good performances on indoor and small-range outdoor scenes. However, these methods can only provide depth estimates up to approximately 50 meters in outdoor environments, which will limit its application in outdoor scenes, as shown in figure 1. The long-range depth information is useful for outdoor applications, such as auto-driving cars. For example, high-speed self-driving cars rely heavily on the data transmitted by sensors to perceive and model the surrounding. If the depth of distant objects can be obtained, self-driving cars will be safer.

To address this problem, we propose a novel monocular depth estimation approach that can obtain the depth of the objects at far distances in an outdoor scene. The apparent ease at which humans can roughly estimate depth motivates us to extend single-view depth estimation limitation. As shown in figure 2, the human can distinguish the depth distribution of the scene from a sketch without any texture information. However, This is very difficult for existing deep learning depth estimation algorithms to reconstruct such depth information. Humans learn to estimate depth by using a variety of monocular depth cues [8], including per-

†Bo Ren is the corresponding author

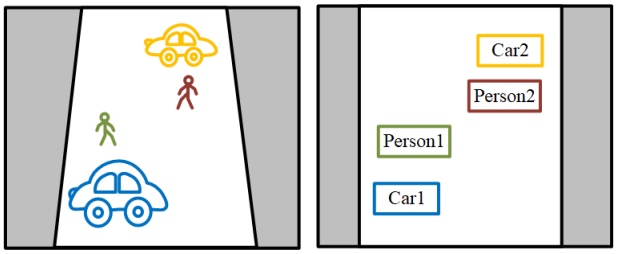

Figure 2. **Picture with only size information and Perspective relationship.**(a) is a simple sketch. (b) is its rough depth distribution. Humans can use the perspective information and object size information of the scene to get the object distribution of the scene without the texture information of the scene.

spective, absolute and relative image size of known objects, and semantic information of the scene. Inspired by this, semantic information, scale prior information, and perspective relationships are used to enhance depth.

Our method has three main stages: The vanishing point and semantic depth estimation(VSDE) stage, the depth merge stage, and the semantic filter stage.

First, the semantic segmentation network is used to obtain the scene's instance-level semantic information. At the same time, vanishing point detection is also performed to gain the scene's perspective information. With the size information of the common objects, the object-level depth can be computed through semantic information and perspective information. The object-level depth means that the depth value is the same in the area covered by the same object, equal to the whole object's distance to the camera. In this stage, the absolute depth scale factor can also be obtained. In the calculation process, the human body and the size of the car in the real world are used as prior information.

Then, the object-level depth map is merged with the monocular depth estimation(MDE) network's result to get the merged depth. The object-level depth measurement range is farther, but detailed information about the scene cannot be recovered. The results of MDE have rich details, but the depth of distant objects cannot be obtained.

Finally, the semantic segmentation results are used to conduct guided filtering on the merged depth to enhance its edge information and smooth the internal depth of the object. The experiment's visual results and quantitative analysis show that our method can achieve good results in outdoor long-distance depth estimation.

In summary, the main contributions of our paper are:

- We propose a new method to compute the object-level depth with semantic information, scale prior information, and perspective relationship for single vanishing point and double vanishing point cases.
- We propose a new method to enhance the depth map by using the semantic segmentation result with the guide filter algorithm.

2. Related Works

2.1. Monocular Depth Estimation

Monocular depth estimation can be divided into supervised depth estimation and unsupervised depth estimation according to whether it uses supervised information for training. Next, we will introduce these two methods separately.

Supervised depth estimation takes RGB images as input and predicts the depth information of the scene based on the prior knowledge in the neural network. Because the training data includes ground-truth depth information, the model can learn absolute depth information, but due to internal camera parameters and other reasons, the model cannot be generalized to different source images and different scenes.

In [6], the author uses two network stacks for depth estimation, and the two networks are responsible for estimating global depth information and local depth information. In [12], the author proposes to use the ubiquitous planar structure in the indoor environment as a guide for depth estimation. The model assumes a linear correlation between pixels in the same plane and uses plane parameterization for training to model the relative position relationship of the scene for depth estimation. In [22], the author proposes to use three-dimensional geometric consistency constraints to train the depth estimation network. The model uses surface normal constraints and virtual normal constraints to model three-dimensional geometric consistency information.

In order to overcome the lack of depth annotation in monocular camera data, many recent works have proposed a variety of unsupervised methods to extract differences and depth cues from image pairs or monocular videos to predict depth.

Garg et al. [7] proposed an unsupervised framework based on auto-encoders for single-view depth prediction. The author considers a pair of images, the source image and the target image, and the relatively small motion between the two images during the training time. Guizilini et al. [9] proposed a self-supervised method to estimate the depth map by combining the geometry of PackNet.

None of these methods considers the problem of outdoor long-distance depth estimation. Their estimated depth range is very limited, and it is impossible to obtain the depth information of objects at a long distance. For depth estimation algorithms based on supervised learning, ground truth data are difficult to obtain, especially depth information at long distances. For unsupervised learning methods, epipolar constraints are often used as training constraints during training, and distant objects are often regarded as backgrounds because of their little position change. These

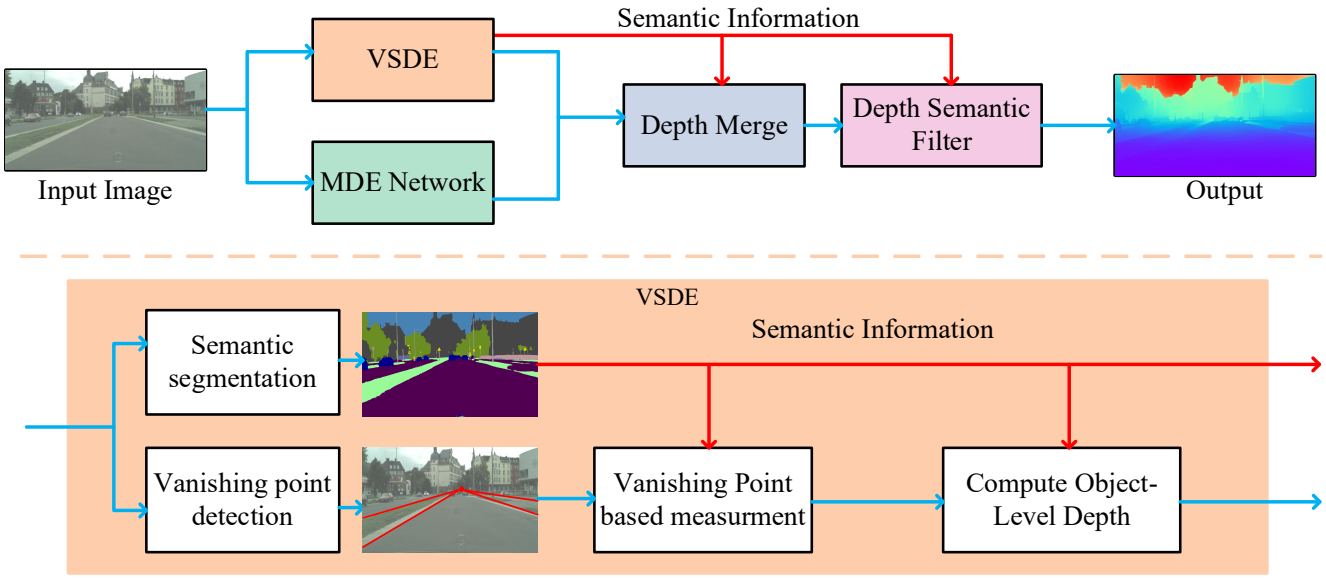

Figure 3. **System Overview.**Our algorithm can be divided into three main steps. First, calculate the object-level depth, and restore the absolute depth factor of the scene. Then we fuse the obtained distance information and the depth information estimated by the MDE network to obtain the depth estimation result enhanced by the perspective information. Finally, we perform guided filter on the merged depth map to enhance its expression of scene details. The VSDE means Vanishing point and semantic depth estimation. The MDE means monocular depth estimation.

problems limit the practical application of depth estimation algorithms.

2.2. Image-based Measurement

In [10] they proposed a method that models the 3D image by specifying the vanishing point in the 2D image. The background in the scene model then consists of at most five rectangles, whereas hierarchical polygons are used as a model for each foreground object. In [14], they propose a new modeling scheme based on a single vanishing line instead of a vanishing point, and this method can be naturally extended to a panoramic image. [16] formulate a non-linear optimization problem to find the 3D scene parameters with respect to the camera position, to automatically construct a reasonable 3D scene model, provided with a set of points and their corresponding points on the water surface.

In [5], They used a known reference plane, a vanishing line on the known reference plane, and a vanishing point perpendicular to the reference plane to measure image information. In [17] describe how 3D metric measurements can be determined from a single uncalibrated image when only minimal geometric information is available in the image. The minimal information just is orthogonal vanishing points.

These methods do not use the relative relationship between the actual object size and the object size on the image and do not make full use of the semantic information existing in the scene. And this method can't get detailed information of scene depth.

3. Method

We propose a new depth estimation method to solve the outdoor long-distance depth estimation problem. Our method can be easily integrated into existing depth estimation algorithms.

As shown in figure 3 Our method can be divided into three main parts: The depth estimation using vanishing point and semantic information(VSDE), the depth merge stage, and the semantic filter stage.

First, the object scale information and computational photography is used to compute the object-level depth. Our method can calculate the distant object's depth in long-distance. The existing methods can obtain the vanishing point and the instance level semantic segmentation information used in our method. Furthermore, in the calculation process, the size of the human body and car in the real world is used as the prior information. Using the prior information, the absolute depth scale factor can be obtained.

Then we merge the object-level depth map with the result of MDE networks. The depth scale information is used to align the scale of the result of MDE networks. Then the object-level depth is used to adjust the relevant regions in the aligned depth map.

Finally, the semantic segmentation result is used as the guided image to conduct guided filtering on the adjusted depth map, enhance its edge information, and smooth the internal depth of each object.

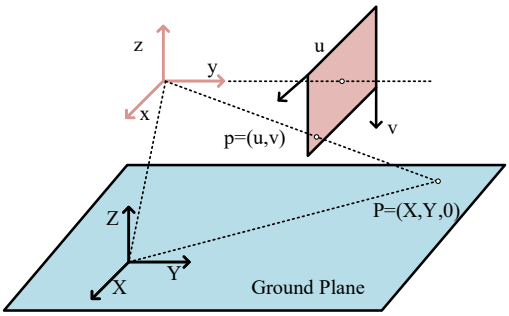

Figure 4. **Outdoor Scene model.**The outdoor pavement can be regarded as the supporting plane of the outdoor 3D model, and other objects are arranged on the pavement according to the perspective relationship.

3.1. Vanishing Point and Semantics Depth Estimation

First, we perform instance-level semantic segmentation on the input image. The vanishing point detecting is also performed to gain the scene's perspective information at the same time. Because the camera's internal parameters can easily be obtained through calibration or camera parameter estimation algorithms, here, we directly use the camera parameters provided by the data set. In addition, since the camera orientation in most autonomous driving technologies is horizontal forward, we also directly use the assumption that the camera orientation is perpendicular to the image plane. Moreover, we assume that the objects are vertically distributed on the ground, which is the general situation of outdoor scenes.

3.1.1 Vanishing Point-Based Measurement

The vanishing point can provide structural information and direction information for scene analysis. Theoretically, the outdoor objects lying on the same ray originating from the vanishing points are perceptively ordered in distance. Here, the detection result of the vanishing point and the Bertozzi formula [1] is used to perform the inverse perspective transformation on the image to measure the distance from the object to the camera plane.

The perspective model of the camera can be simplified as shown in figure 4. In practical applications, We will adjust the position of the world coordinate system to ensure the origin of the camera coordinate system in the world coordinate system is $(0, 0, h)$. h is the camera's height, here we use the parameters provided by the data set. Under this model, the inverse perspective transformation can be defined by the following formula.

$$\begin{cases} \mathrm{rFactor} = \left(1 - \frac{2\mathrm{u}}{\mathrm{M}-1}\right) \times \tan\left(\alpha_{\mathrm{r}}\right) \\ \mathrm{cFactor} = \left(1 - \frac{2\mathrm{v}}{\mathrm{N}-1}\right) \times \tan\left(\alpha_{\mathrm{u}}\right) \\ \mathrm{X}_0(\mathrm{u},\mathrm{v}) = \mathrm{h} \times \frac{1+\mathrm{rFactor}\times\tan(\theta)}{\tan(\theta)-\mathrm{rFactor}} + \mathrm{C}_x \\ \mathrm{Y}_0(\mathrm{u},\mathrm{v}) = \mathrm{h} \times \frac{\mathrm{cFactor}/\cos(\theta)}{\tan(\theta)-\mathrm{rFactor}} + \mathrm{C}_y \end{cases} \tag{1}$$

Where $\mathrm{X}_0(\mathrm{u},\mathrm{v})$ and $\mathrm{Y}_0(\mathrm{u},\mathrm{v})$ Represents the road surface coordinates in the world coordinate system; u and v respectively represent the abscissa and ordinate values of the world coordinate system mapped to the image coordinate system; M and N represent the width and height of the image respectively; C_x,C_y represents the coordinate position of the camera in the world coordinate system. α_r indicates the range of the camera's vertical field of view; α_u indicates the range of the camera's horizontal field of view. θ represents the vertical pitch angle of the camera.

α_r and α_u can be calculated with the following equation: the L is the camera focal length.

$$\begin{cases} \alpha_r = \arctan(\frac{N}{2L}) \\ \alpha_u = \arctan(\frac{M}{2L}) \end{cases} \tag{2}$$

Suppose the coordinate of vanishing point is (x_c, y_c)The camera's vertical pitch angle can be calculated using the information of the vanishing point,

$$\theta = \arctan(\tan(\alpha_r) \times (1 - \frac{2(N - y_c)}{N})) \tag{3}$$

Through the result of semantic segmentation, the position of the ground in the image coordinate system can be obtained. Outdoor objects are often arranged from near to far around the vanishing point. Therefore, we can perform the inverse perspective transformation on the line of intersection between the object and the ground to obtain the distance of each point on the line of intersection with the ground relative to the camera coordinate system. Average these distances as the distance from the object to the camera.

3.1.2 Scences With Tow Vanishing Points

In daily life, there are often two vanishing points in the pictures of outdoor scenes. For example, at intersections, two different vanishing points are formed. As shown in 5, in this case, there are two distinct vanishing points in the image. The depth of the object on the other side cannot be accurately calculated from only one vanishing point. When only the right vanishing point is used, the depth information of the left object cannot be accurately calculated, and vice versa.

In the presence of two vanishing points, to address the above problem, a new computational strategy is preformed.

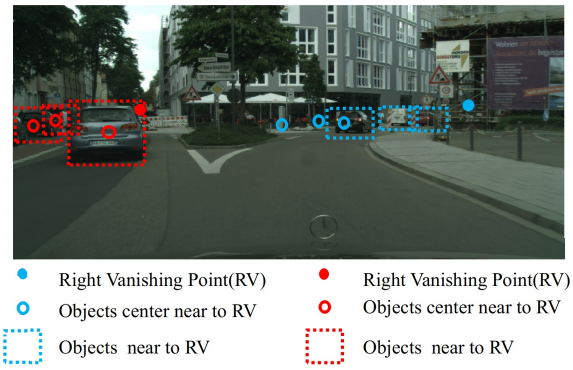

Figure 5. **A scene with two vanishing points.** At intersections,two different vanishing points are formed. In this case, use only one vanishing point cannot calculate the depth information of the objects accurately.

After calculating the coordinates of the vanishing point, the distance from each mask in the semantic segmentation result to the vanishing point can be calculated. This distance refers to the image distance from the centroid of each mask to the vanishing point. For each mask, the vanishing point closest to it is used as the vanishing point used to compute the inverse perspective transform.

The geometric moments of the image is used to calculate the centroid coordinates of each mask region. The formula for calculating the geometric moment is as follows.

$$M_{ij} = \sum_{x,y} I(x,y) * x^j * y^i \tag{4}$$

where $I(x,y)$ is the pixel value at pixel (x,y). When both i and j take the value 0, it is called the zeroth moment, and the zeroth moment can be used to calculate the centroid of a shape. When x and y take the value 0 and 1 respectively, it is called the first moment, and so on. The calculation formula of the image centroid is as follows:

$$\overline{x} = \frac{M_{10}}{M_{00}}, \overline{y} = \frac{M_{01}}{M_{00}} \tag{5}$$

After obtaining the centroid of each mask, the distance from each mask to the vanishing point is obtained by calculating the Euclidean distance from the vanishing point.

3.1.3 Convert Size to Depth

Size information is useful in the depth estimation [2] [20]. In this step, we use the scale information of common objects as a priori information to adjust the object-level depth and compute the absolute depth scale. The prior information of the object scale can be obtained through the statistics website. We select people and cars as reference objects in the experiment.

Because our camera model is a small hole imaging model, for the objects on the image, the following relationship holds:

$$d = kL\frac{A_{ws}}{A_{is}} \tag{6}$$

d is the distance of the object from the camera plane,A_{ws},A_{is} respectively represent the size of the object in the world coordinate system and the image coordinate system. k is the depth factor,L is the camera focal length.

For each object, the relationship is established. For the human body and cars, height is used as the scale information for judging its depth. In real life, we can assume that cars and people are distributed perpendicular to the ground, so the height of these objects in the image coordinate system can be calculated by calculating the pixel distance in the vertical direction on the image.

Each set of actual size and image size are satisfied with equation 6. From each equation, the scale factor k of the image can be solved . Due to the existence of measurement noise, the value of k may not be consistent.The final scale factor k_n can be calculated from the least-squares solution of these equations.

3.2. Depth Merge

In the depth map merging stage, object-level depth calculated in the previous step is merged with the depth estimated by the monocular depth estimation network.

Similar to humans' understanding of the scene's depth, the first thing to judge is the distance of a single object, rather than calculating the distance of each point on the object. After the previous calculation stage, the depth inside the object is the same, representing the distribution of the entire object in the scene. However, the depth inside the object obtained by the depth estimation network is different and has richer depth information.

In the deep merging stage, the object-level depth calculated in the previous step is merged with the result of the depth estimation network.

3.2.1 Depth Align

Due to the different calculation methods, the two types of depth information scales are also different. Before these two depth maps be merged, them need to be normalized to the same scale. Let C_i be the area enclosed by the i-th contour in the segmentation results. The area's distance can be generate through the previous step, donated as $distance(C_i)$. Then calculate the average depth covered by the contour on the depth map, donate as $depth(C_i)$. Note that the depth estimated by the depth estimation network often has a significant deviation at a long distance, which can be observed from figure 6. Here the area in which the depth

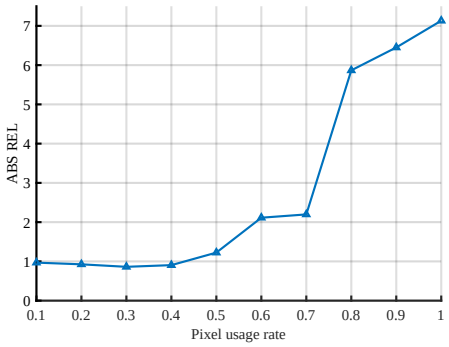

Figure 6. **The relationship between using pixel range and absolute error.**The depth estimated by the depth estimation network often has a significant deviation at a long distance.

value is small than the top 70% is used to estimate the scale factor. Then object-level scale factor can be calculated from the equation as follows:

$$factor_i = \frac{distance(C_i)}{depth(C_i)} \tag{7}$$

By calculating the value of each mask that meets the standard, a factor sequence $factor_i \in F$ can be calculated and then the weighted average of the sequence can be obtained from follow equation:

$$factor = \frac{\sum(w_i f_i)}{\sum w_i}, for \quad factor_i \in F \tag{8}$$

Among them, $w_i = 1/depth_i$. Because as the distance increases, the error of depth estimation will gradually become larger, the weight of the scale factor calculated at a far distance is relatively small. Then this factor is used to normalize the depth map generate from the MDE network.

3.2.2 Depth Merge

After the depth alignment, the object-level depth and depth from MDE will be merged. The following two situations will be encountered when performing deep merging:

- The distance information of the object is calculated above, but there is no such information in the estimated depth map. This situation often occurs at the far position of the depth map.
- The depth information of the object exists in both the object-level depth map and MDE's result. This situation often occurs in nearby objects or objects with apparent structures.

We adopt different strategies to deal with these two situations. First, the edge detection operator is used to perform edge detection on the depth map. If there is no edge near the object, it belongs to case 1. Otherwise, it belongs to case 2.

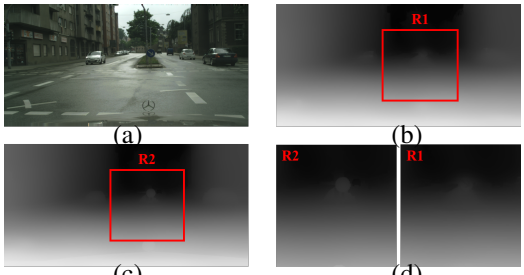

Figure 7. **Visual comparison of two filtering methods.**(a) is the input image,(b) is the result of the guided filter using semantic result,(c) is the result of the guided filter using the RGB image,(d) is the compare.

If the object belongs to case 1, the MDE network cannot obtain the object's depth here. In this case, the object level depth of the area is directly used to fill it. This depth value represents the position of the entire object in the scene.

If the object belongs to case 2, the MDE network also gives an estimate to this object. So the following formula is used to adjust the object's depth in this area while retaining the position information of the object and its internal depth distribution.

$$depth_m(C_i) = depth(C_i) + offset \tag{9}$$

$$offset = distance(C_i) - avg(depth(C_i)) \tag{10}$$

$depth_m(C_i)$ is the depth of the object after merge. It is equivalent to the original depth of the area plus the offset between the depth value of the area and the object level.

3.3. Depth Filter

After the depth merging stage, the object-level depth information calculated by the vanishing point and the depth obtained from the MDE network are merged. Then the guided filter that uses the semantic segmentation map as the guidance is used to enhance the result.

The object's depth should be as different as possible from the depth of the surrounding environment, and the depth inside the object should be as consistent as possible. This is the effect we expect to obtain after filtering. The nature of guided filtering can satisfy this need well. The guided filtering can be formulated as follows :

$$q_i = a_k I_i + b_k, \forall i \in \omega_k \tag{11}$$

$$q_i = pi - n_i \tag{12}$$

In these equations, q is the output image, I is the guided image, and P is the output image. The meaning of this formula is that the pixels on the output image can be regarded as the linear transformation of neighboring pixels on the guide map. The local linear model ensures that the edge of the result is consistent with the edge of the guided image.

	Cityscape				SYNTHIA			
	LeReS	Midas	Ours$_{Midas}$	Ours$_{LeReS}$	LeReS	Midas	Ours$_{Midas}$	Ours$_{LeReS}$
ORD	0.1761	0.2087	0.1988	**0.1246**	0.1073	0.0865	**0.0862**	0.1057
D^3R	0.3047	0.2220	**0.2124**	0.2839	0.2397	0.1375	**0.1364**	0.2357
RMSE	0.3791	0.3849	0.3829	**0.3565**	0.2819	0.6890	0.6776	**0.2616**
SQ REL	0.3367	0.2901	0.2899	**0.2898**	0.3179	0.3691	0.3686	**0.3077**
$\delta_{1.25}$	0.9872	0.9574	0.9733	**0.9910**	0.9985	0.9957	0.9964	**0.9996**

Table 1. Quantitative Results

The next is to solve such coefficients so that the difference between p and q is as small as possible, and the local linear model can also be maintained. Linear ridge regression with the standard term is used as follows:

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2) \tag{13}$$

Through the above definition, it can be seen that the choice of the guide image is essential. [11] directly use the RGB image of the input depth map for depth filtering, but there is a lot of redundant edge information in the RGB image. This will cause artifacts on the filter result.

Here, the results obtained by segmentation is used as the guided image. The edges on the semantic segmentation results represent the boundaries between different objects. After filtering, the depths between different objects will be distinguished according to the boundaries of the objects. The depth inside the same object will be smoothed. Using segmentation results as the guided image can achieve the goal mentioned above.

The visible result of using semantic segmentation result and RGB image is shown in figure 7. From the result, we can observe that direct use of the RGB images as the guided image will cause a lot of artifacts on the depth map, but using segmentation results does not have such problems.

4. Experiments and Results

4.1. Dataset and Evaluation Criteria

We evaluate our algorithm on the Cityscape [4] and SYNTHIA [13] datasets. The Cityscape dataset contains a large number of RGB images of outdoor scenes and provides corresponding ground truth depth information.SYNTHIA consists of a collection of frames rendered from a virtual city and comes with depth information.

A set of standard depth evaluation metrics are used as suggested in recent work [18, 21] to evaluate our method's performance. The metrics root mean squared error in disparity space (RMSE), square relative error(SQ REL),percentage of pixels with $\delta_{1.25}$, and ordinal error (ORD) from [21] in depth space. Additionally, the depth discontinuity disagreement ratio (D^3R) in [15] is used to evaluate the quality of high frequencies in depth estimates.

4.2. Results And Analysis

We evaluate how much our method can improve upon pre-trained monocular depth estimation models using Midas [18], and LeRes [24]. Furthermore, the whole experiment is running on one Nvidia 3080 GPU, and we use the default size of the pretrain model as the input size. The guided filter uses a radius of 12 and an accuracy of 0.001 in our experiments, and the depth map is scaled to one-half the original size before performing the guided filtering process.

The quantitative results are listed in Table 1, and the visualization result is shown in figure 8. From the quantitative results, we can see that our method has a significant improvement in ORD and D^3R, and the RMSE, SQ REL also have improved. This shows that our method can well recover the high-frequency information on the image. Our method can retain more detailed information that exists in the original image. At the same time, the error of depth estimation is also lower than the original algorithm.

The performance improvement provided by our method is more significant in qualitative comparisons shown in Figure 8. It can be seen from the figure that our method can recover the edge and detail information of the image very well. Especially at the area in the far distance, the original algorithm does not recover the details of the image very well. However, our method can recover the depth information of objects in these areas, which is very valuable for outdoor applications.

It can be seen from the above experimental results that our algorithm can achieve good results in both evaluation indicators and visualization results. It can handle the outdoor long-distance depth estimation problem that the traditional depth estimation algorithm cannot handle and get a more hierarchical outdoor depth distribution. This feature can well meet application scenarios that require outdoor long-distance depth, such as autonomous driving.

4.3. Ablation Study

We conduct ablation studies to investigate the individual contribution of our method's component. We use LeReS as the MDE network and test on the CityScape dataset. For the "+GR", we use the RGB image as our input guided image. For the "+GS", we use the semantic segmentation results as

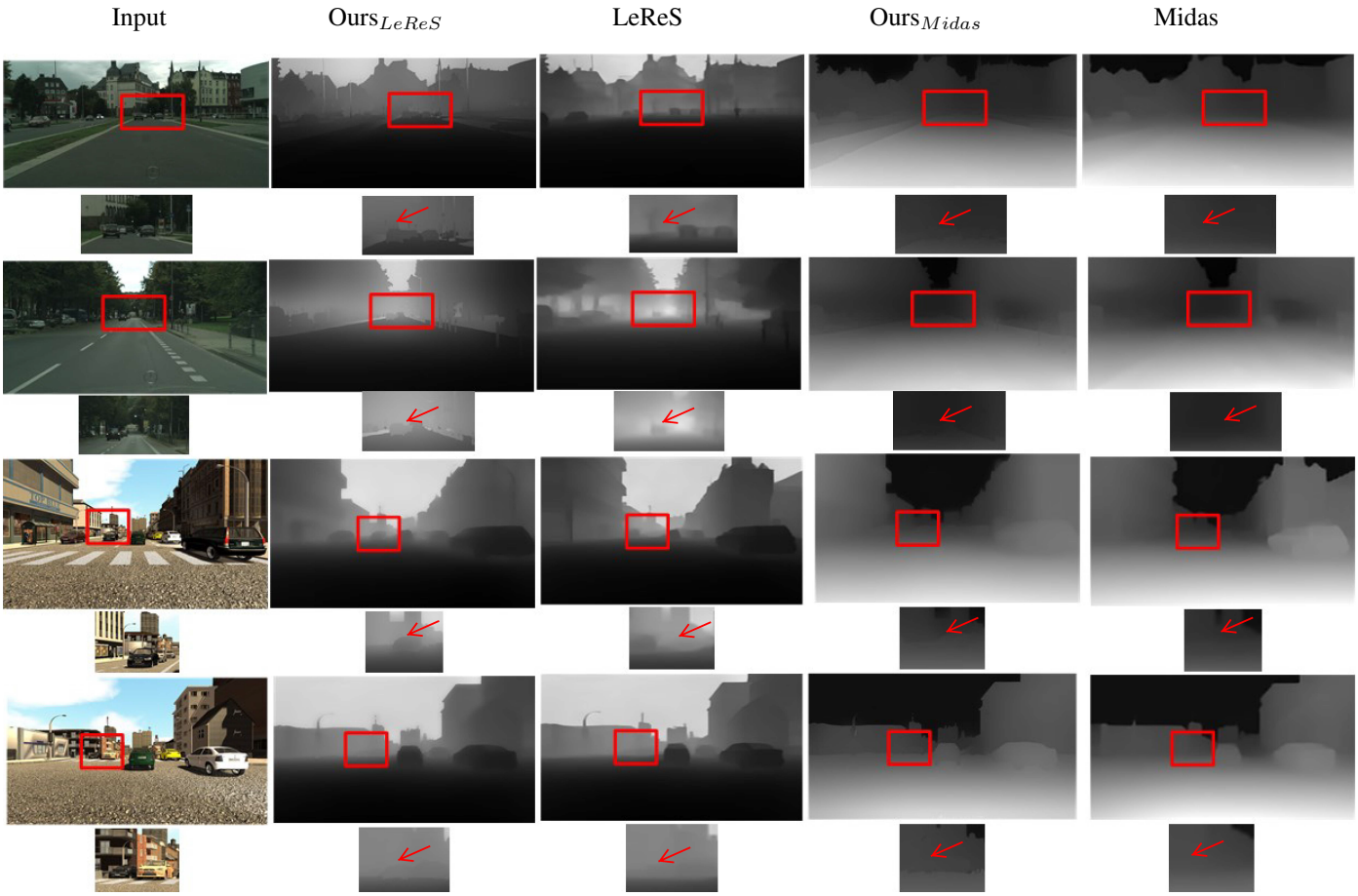

Figure 8. Visualization results

	Cityscape				SYNTHIA			
	Ours$_{LeReS}$+GR	Ours$_{Midas}$+GR	Ours$_{Midas}$+GS	Ours$_{LeReS}$+GS	Ours$_{LeReS}$+GR	Ours$_{Midas}$+GR	Ours$_{Midas}$+GS	Ours$_{LeReS}$+GS
ORD	0.1373	**0.1246**	0.2013	0.1988	0.1073	0.1057	**0.0825**	0.0862
D^3R	0.3186	0.2185	**0.2124**	0.2839	0.2397	0.1368	**0.1364**	0.2357
RMSE	0.3905	0.3844	0.3829	**0.3565**	0.2819	0.6897	0.6776	**0.2616**
SQ REL	0.2987	0.2902	0.2899	**0.2898**	0.3179	0.3686	0.3686	**0.3077**
$\delta_{1.25}$	0.9881	0.9602	0.9733	**0.9910**	0.9985	0.9959	0.9964	**0.9996**

Table 2. Ablation Study

our input guide image. When using RGB as the guide image for our depth filtering, the effect is not obvious. This is because the direct use of RGB images as the input for filtering will introduce too much depth information that should not exist in the depth map, resulting in the algorithm not obtaining the expected performance.

5. Conclusion and Limitation

We propose a new depth estimation algorithm to estimate the depth information of objects at a long distance outdoors. Our algorithm combines semantic sizes and perspective information and is able to recover absolute object-level depth in a far distance in outdoor scenes. A guided filtering method using semantic information is also proposed to refine the final result. Experimental results show that compared to existing methods, our results have a better preference in visual effect, and ORD,D^3R,$\delta_{1.25}$, RMSE are also improved.

We directly apply[19] in vanishing point detection, [3] in instance-level semantic segmentation , which possibly contain systematic errors. Development in vanishing point detection and semantic segmentation will reduce such errors. In addition, using the size of the object as a priori information will also bring errors in the calculation process of the algorithm. In the next work, the depth network can be used to measure the size of the object in the picture.

Acknowledgments

This work is supported by the Natural Science Foundation of China,No.62132012.

References

[1] M. Bertozz, A. Broggi, and A. Fascioli. Stereo inverse perspective mapping: theory and applications. *Image and vision computing*, 16(8):585–590, 1998. 4

[2] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019. 5

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 8

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[5] A. Criminisi. Single-view metrology: Algorithms and applications. In *Joint Pattern Recognition Symposium*, pages 224–239. Springer, 2002. 3

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Neural Information Processing Systems (NeurIPS)*, 2014. 2

[7] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)*. Springer, 2016. 2

[8] E. B. Goldstein and J. Brockmole. *Sensation and perception*. Cengage Learning, 2016. 1

[9] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[10] Y. Horry, K.-I. Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232, 1997. 3

[11] T.-W. Hui and K. N. Ngan. Depth enhancement using rgb-d guided filtering. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3832–3836. IEEE, 2014. 7

[12] L. Huynh, P. Nguyen-Ha, J. Matas, E. Rahtu, and J. Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *European Conference on Computer Vision (ECCV)*, pages 581–597. Springer, 2020. 2

[13] D. H. Juárez, L. Schneider, A. Espinosa, D. Vázquez, A. M. López, U. Franke, M. Pollefeys, and J. C. Moure. Slanted stixels: Representing san francisco's steepest streets. *CoRR*, abs/1707.05397, 2017. 7

[14] H. W. Kang, S. H. Pyo, K.-i. Anjyo, and S. Y. Shin. Tour into the picture using a vanishing line and its extension to panoramic images. In *Computer Graphics Forum*, volume 20, pages 132–141. Wiley Online Library, 2001. 3

[15] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021. 7

[16] J. Park, N. Heo, S. Choi, and S. Y. Shin. Tour into the picture with water surface reflection and object movements. *Computer Animation and Virtual Worlds*, 17(3-4):315–324, 2006. 3

[17] K. Peng, L. Hou, R. Ren, X. Ying, and H. Zha. Single view metrology along orthogonal directions. In *2010 20th International Conference on Pattern Recognition*, pages 1658–1661. IEEE, 2010. 3

[18] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 7

[19] G. Simon, A. Fond, and M.-O. Berger. A simple and effective method to detect orthogonal vanishing points in uncalibrated images of man-made environments. In *Eurographics 2016*, 2016. 8

[20] Y. Wu, S. Ying, and L. Zheng. Size-to-depth: A new perspective for single image depth estimation. *ArXiv*, abs/1801.04461, 2018. 5

[21] K. Xian, J. Zhang, O. Wang, L. Mai, Z. Lin, and Z. Cao. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 611–620, 2020. 7

[22] W. Yin, Y. Liu, C. Shen, and Y. Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[23] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. Learning to Recover 3D Scene Shape from a Single Image. In *arXiv*, 2020. 1

[24] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. Learning to recover 3d scene shape from a single image. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR)*, 2021. 7