remote sensing

MDPI

Article

U^2-ONet: A Two-Level Nested Octave U-Structure Network with a Multi-Scale Attention Mechanism for Moving Object Segmentation

Chenjie Wang [1], Chengyuan Li [1], Jun Liu [1], Bin Luo [1,*], Xin Su [2], Yajun Wang [1] and Yan Gao [3]

[1] State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; wangchenjie@whu.edu.cn (C.W.); lichengyuan@whu.edu.cn (C.L.); liujunand@whu.edu.cn (J.L.); yjwangisu@whu.edu.cn (Y.W.)
[2] School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; xinsu.rs@whu.edu.cn
[3] Zhuhai Da Hengqin Science and Technology Development Co., Ltd., Unit 1, 33 Haihe Street, Hengqin New Area, Zhuhai 519031, China; 16425707@life.hkbu.edu.hk
* Correspondence: luob@whu.edu.cn

Abstract: Most scenes in practical applications are dynamic scenes containing moving objects, so accurately segmenting moving objects is crucial for many computer vision applications. In order to efficiently segment all the moving objects in the scene, regardless of whether the object has a predefined semantic label, we propose a two-level nested octave U-structure network with a multi-scale attention mechanism, called U^2-ONet. U^2-ONet takes two RGB frames, the optical flow between these frames, and the instance segmentation of the frames as inputs. Each stage of U^2-ONet is filled with the newly designed octave residual U-block (ORSU block) to enhance the ability to obtain more contextual information at different scales while reducing the spatial redundancy of the feature maps. In order to efficiently train the multi-scale deep network, we introduce a hierarchical training supervision strategy that calculates the loss at each level while adding knowledge-matching loss to keep the optimization consistent. The experimental results show that the proposed U^2-ONet method can achieve a state-of-the-art performance in several general moving object segmentation datasets.

Keywords: moving object segmentation; octave convolution; nested U-structure; hierarchical supervision

Citation: Wang, C.; Li, C.; Liu, J.; Luo, B.; Su, X.; Wang, Y.; Gao, Y. U^2-ONet: A Two-Level Nested Octave U-structure Network with Multi-scale Attention Mechanism for Moving Object Segmentation. *Remote Sens.* **2021**, *13*, 60. https://doi.org/10.3390/rs13010060

Received: 15November 2020
Accepted: 21 December 2020
Published: 25 December 2020

1. Introduction

Moving object segmentation is a critical technology in computer vision tasks, and is directly related to the effects of the subsequent work, such as object tracking, visual simultaneous localization and mapping (SLAM), image recognition, etc. Being able to accurately segment moving objects from a video sequence can greatly improve the effects of many visual tasks, such as dynamic visual SLAM [1–4], visual object tracking [5], dynamic object obstacle avoidance, autonomous navigation [6], autonomous vehicles [7], human activity analysis [8], video surveillance [9,10], and dynamic object modeling [11]. For example, in an autonomous driving scene, the segmentation of moving objects can help the vehicle to understand the surrounding motion information, which is the basis for avoiding collision, braking operations, and smooth maneuvering. Most of the current methods are designed to segment N predefined classes in the training set. However, in a practical environment, many applications, such as autonomous driving and intelligent robots, need to achieve robust perception in the open world. These applications must discover and segment never-before-seen moving objects in the new environment, regardless of whether they are associated with a particular semantic class.

The segmentation of the different motions in dynamic scenes has been studied for decades. The traditional methods of motion segmentation use powerful geometric con-

straints to cluster the points in the scene into a model parameter instance, thereby segmenting the moving objects into different motions [12,13], which is called multi-motion segmentation. This kind of method realizes the motion segmentation of feature points instead of working pixel by pixel. However, since the results of these methods are strongly dependent on the motion model, they are not robust enough in complex scenes. In addition, these methods can only segment the more salient moving objects and only simultaneously fit a small number of motion models in scenes. With the development of deep learning, instance/semantic segmentation and object detection in videos have been well studied [14–17]. These methods are used to segment specific labeled object categories in annotated data, so that the main focus is on predefined semantic category segmentation through appearance rather than segmentation of all the moving objects. Meanwhile, these methods are not able to segment new objects that have not been labeled in the training data. More recent approaches combine instance/semantic segmentation results with motion information from optical flow to segment moving object instances in dynamic scenes, as in [18–21]. These methods [19,20] can segment never-before-seen objects that have not been predefined in the training set based on their motion.

However, the networks used in these methods are usually not deep enough due to the fact that deeper networks usually lead to increased computational burden and greater training difficulty. The use of deeper architectures has proved successful in many artificial intelligence tasks. Therefore, in this paper, we use a much deeper architecture, which can greatly improve the effectiveness of the moving object segmentation. In order to avoid the increase in spatial redundancy of the feature maps, the increase in computational burden and memory cost, and the greater difficulty of training supervision, we integrate octave convolution (OctConv) [22] to improve the residual U-block (RSU block) in [23] and propose the novel octave residual U-block (ORSU block). We take advantage of OctConv to reduce the spatial redundancy and further improve the accuracy. We also propose a hierarchical training supervision strategy to improve the training effect of the deep network optimization in order to improve the segmentation accuracy.

In this paper, we propose a novel two-level nested U-structure network with a multi-scale attention mechanism to learn to segment pixels belonging to foreground moving objects from the background, called U^2-ONet, whose inputs consist of two RGB frames, the optical flow between between the pair of RGB frames, and the instance segmentation of the frames. We combine the convolutional block attention module (CBAM) [24] and OctConv [22] with the U^2-Net [23] network originally used for salient object detection to propose this network for moving object segmentation. Due to the continuation of U^2-Net's main architecture, the proposed U^2-ONet is a nested U-structure network that is designed without using any pre-trained backbones from image classification for training. On the bottom level, the novel octave residual U-block (ORSU block) is proposed, which is based on the residual U-block (RSU block) in [23], and uses octave convolution (OctConv) [22], factorizing the mixed feature maps by their frequencies, instead of vanilla convolution. With the advantages of OctConv and the structure of U-blocks, the ORSU blocks can extract intra-stage multi-scale features without degrading the feature map resolution, with a smaller computational cost than the RSU. At the top level, there is a U-Net-like structure, in which each stage is filled by an ORSU block and each scale contains an attention block. By adding attention blocks at different scales, we introduce spatial and channel attention into the network and eliminate aliasing effects. For the training strategy, we propose a hierarchical training supervision strategy instead of using the standard top-most supervised training or a deeply supervised training scheme. We calculate the loss at each level and add a probability-matching loss called the Kullback–Leibler divergence loss (KLloss) to promote the supervision interactions among the different levels in order to guarantee a more robust optimization process and better representation ability. An illustration of U^2-ONet is provided in Figure 1.

In summary, this work makes the following key contributions:

1. We propose the U^2-ONet, which is a two-level nested U-structure network with a multi-scale attention mechanism to efficiently segment all the moving object instances in a dynamic scene, regardless of whether they are associated with a particular semantic class.
2. We propose the novel octave residual U-block (ORSU block) with octave convolution to fill each stage of the U^2-ONet. The ORSU blocks extract intra-stage multi-scale features while adopting more efficient inter-frequency information exchange, as well as reducing the spatial redundancy and memory cost in the convolutional neural network (CNN).
3. We propose a hierarchical training supervision strategy that calculates both the standard binary cross-entropy loss (BCEloss) and KLloss at each level, and uses the KLloss implicit constraint gradient to enhance the opportunity of knowledge sharing in order to improve the training effect of this deep network.
4. In the task of moving object segmentation, the results have proved that the U^2-ONet is efficient, and the hierarchical training supervision strategy improves the accuracy of the deep network. The experimental results show that the proposed U^2-ONet achieves a state-of-the-art performance in some challenging datasets, which include camouflaged objects, tiny objects, and fast camera motion.

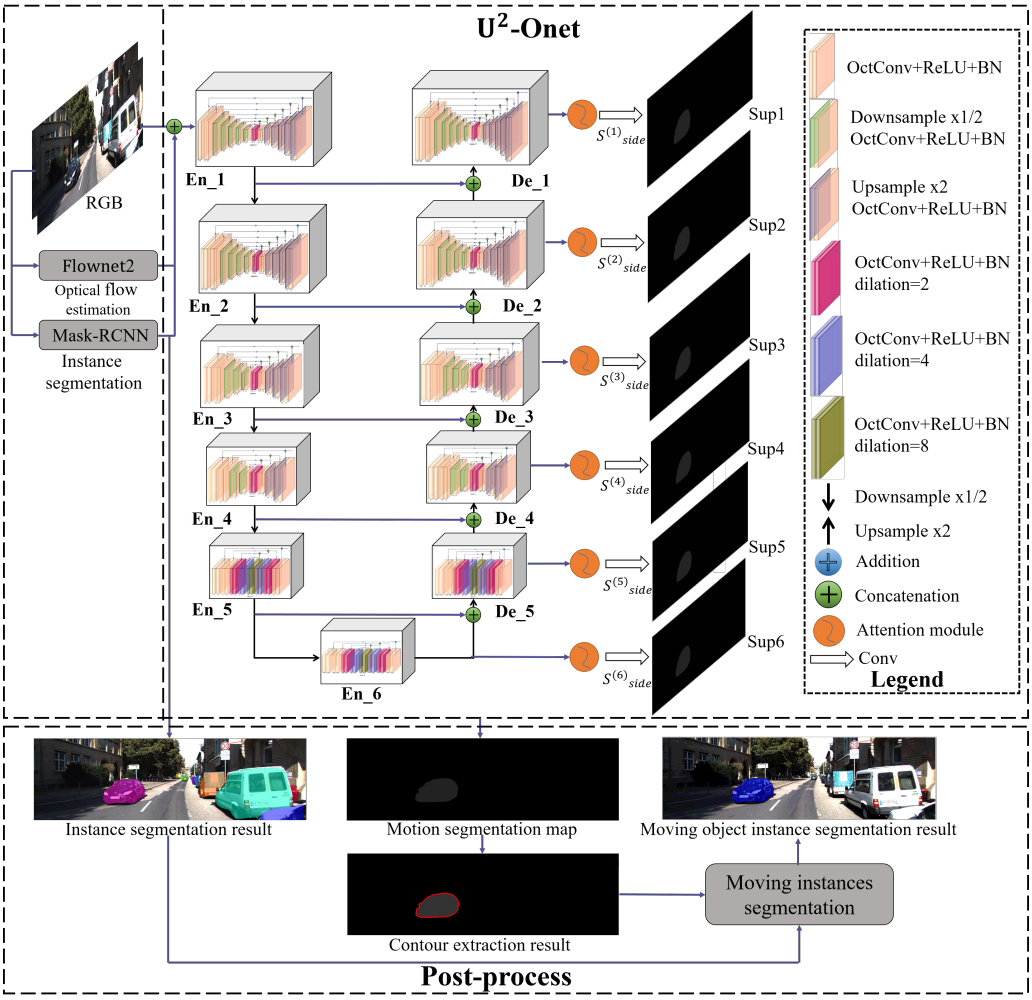


Figure 1. Illustration of the two-level nested octave U-structure network with a multi-scale attention mechanism (U^2-ONet).

2. Related Work

2.1. Video Foreground Segmentation

Video foreground segmentation is focused on classifying every pixel in a video as either foreground or background. Early methods [25–27] relied on heuristics in the optical flow field, such as spatial edges and temporal motion boundaries in [27], to identify moving objects. With the introduction of a standard benchmark, the Densely Annotated Video Segmentation (DAVIS) 2016 dataset [28], there has been much related research on video object segmentation [29–35]. Some methods [29,31–34] only complete segmentation of the foreground objects and the background, without segmenting individual instances. Among the instance-level methods, the attentive graph neural network (AGNN) method [30] is based on a novel neural network, and the collaborative video object segmentation using the foreground–background integration (CFBI) method [35] imposes the feature embedding from both the foreground and background to perform the matching process from both the pixel and instance levels. However, video object segmentation usually involves segmenting the most salient and critical objects in the video, not just moving objects. The proposed U^2-ONet method focuses on motion information and is used to segment all the moving objects in the scene, regardless of whether they are salient.

2.2. Instance Segmentation

Instance segmentation not only needs to assign class labels to pixels, but also to segment individual object instances in the images. Methods based on R-CNN (Region-CNN) [36] are popular and widely used at present. Mask-RCNN [14] uses the object bounding box obtained by Faster RCNN [37] to distinguish each instance, and then segments the instances in each bounding box. PANet [38] improves Mask-RCNN by adding bottom-up path augmentation that enhances the entire feature hierarchy with accurate localization signals in earlier layers of the network. Subsequently, BshapeNet [39] utilizes an extended framework by adding a bounding box mask branch that provides additional information about the object positions and coordinates to Faster RCNN to enhance the performance of the instance segmentation. More recently, a few novel contour-based approaches for real-time instance segmentation have been proposed [40,41]. The Deep Snake method [40] uses circular convolution for feature learning on the contours, and uses a two-stage pipeline including initial contour proposal and contour deformation for the instance segmentation. The Poly-YOLO algorithm [41] increases the detection accuracy of YOLOv3 and realizes instance segmentation using tight polygon-based contours. More recent approaches consider the instance segmentation problem as a pixel-wise labeling problem by learning pixel embeddings. The method proposed in [42] unrolls mean shift clustering as a neural network, and the method proposed in [43] introduces a new loss function optimizing the intersection over union of each object's mask to cluster pixels into instances. EmbedMask [44] is built on top of the one-stage detection models and applies proposal embedding and pixel embedding for instance segmentation. The method proposed in [45] uses a deep neural network to assign each pixel an embedding vector and groups pixels into instances based on object-aware embedding. However, most of these methods focus on appearance cues and can only segment objects that have been labeled as a specific category in a training set. In the proposed approach, we leverage the semantic instance masks from Mask-RCNN with motion cues from optical flow to segment moving object instances, whether or not they are associated with a particular semantic category.

2.3. Motion Segmentation

The multi-motion segmentation method [12,13,46–48] based on geometric methods involves clustering points of the same motion into a motion model parameter instance to segment the multiple motion models of the scene, which can be utilized to discover new objects based on their motion. This kind of method obtains the results at the feature point level, instead of pixel by pixel, and the application conditions and scenarios are limited. For example, these methods only segment the more salient moving objects, there is

a limited model number of segmentations, and they come with a high computational cost. Some deep-learning-based methods segment foreground moving object regions from the scene. The method proposed in [9] uses an analysis-based radial basis function network for motion detection in variable bit-rate video streams. The method proposed in [49] proposes a novel GAN (Generative Adversarial Networks) network based on unsupervised training and a dataset containing images of outdoor all-day illumination changes for detecting moving objects. However, these methods cannot segment each moving object instance. More recent approaches have used optical-flow-based methods for the instance-level moving object segmentation, including a hierarchical motion segmentation system that combines geometric knowledge with a modern CNN for appearance modeling [18], a novel pixel-trajectory recurrent neural network to cluster foreground pixels in videos into different objects [19], a two-stream architecture to separately process motion and appearance [20], a new submodular optimization process to achieve trajectory clustering [50], and a statistical-inference-based method for the combination of motion and semantic cues [21]. In comparison, we propose a two-level nested U-structure deep network with octave convolution to segment each moving object instance while reducing the spatial redundancy and memory cost in the CNN.

3. Method

Firstly, the overall structure of the network is introduced, including the network inputs. Next, the design of the proposed ORSU block is be introduced, and the structure of the U^2-ONet is described. Then, the hierarchical training supervision strategy and the training loss procedure are described. The post-processing used to obtain the instance-level moving object segmentation results is introduced at the end of this section.

3.1. Overall Structure

The proposed approach takes video frames, the instance segmentation of the frames, and the optical flow between pairs of frames as inputs, which are concatenated in the channel dimension and fed through U^2-ONet. We use the well-known FlowNet2 [51] and Mask-RCNN [14] methods, respectively, to obtain the results of the optical flow and instance segmentation as inputs. We use the public training models of FlowNet2 and Mask-RCNN. The FlowNet2 and Mask-RCNN networks only provide input data and do not participate in training. Before the optical flow is fed through the network, we undertake normalization to further highlight the moving objects. U^2-ONet is built with the ORSU blocks based on octave convolution and the multi-scale attention mechanism based on the convolutional block attention module (CBAM) [24]. Inspired by octave convolution [22] and U^2-Net [23], the octave U-block (ORSU block) is designed to capture intra-stage multi-scale features while reducing the spatial redundancy and computational cost in the CNN. For the motion segmentation map obtained from U^2-ONet, post-processing to combine it with the instance segmentation results is used to obtain the instance-level moving object segmentation results. The contours of the motion segmentation map are extracted, and each closed motion contour is used to determine whether each semantically labeled instance is moving and to find new moving instances.

3.2. ORSU Blocks

Inspired by U^2-Net [23], we propose the novel octave residual U-block (ORSU block) in order to make good use of both local and global contextual information to improve the segmentation effect. As shown in Figure 2, ORSU-$L(C_{in},M,C_{out})$ follows the main structure of the RSU block in U^2-Net [23]. Therefore, the proposed ORSU block is composed of three main parts:

1. An input convolutional layer, which uses octave convolution (OctConv) for the local feature extraction instead of vanilla convolution. Compared with RSU blocks, ORSU blocks using OctConv further reduce the computation and memory consumption while boosting the accuracy of the segmentation. This layer transforms the input

feature map $X(H \times W \times C_{in})$ into an intermediate map $F_1(x)$ with the output channel of C_{out}.

2. A U-Net-like symmetric encoder–decoder structure with a height of L, which is deeper with a larger value of L. It takes $F_1(x)$ from the input convolutional layer as input and learns to extract and encode the multi-scale contextual information $\mu(F_1(x))$, where μ denotes the U-Net-like structure, as shown in Figure 2.

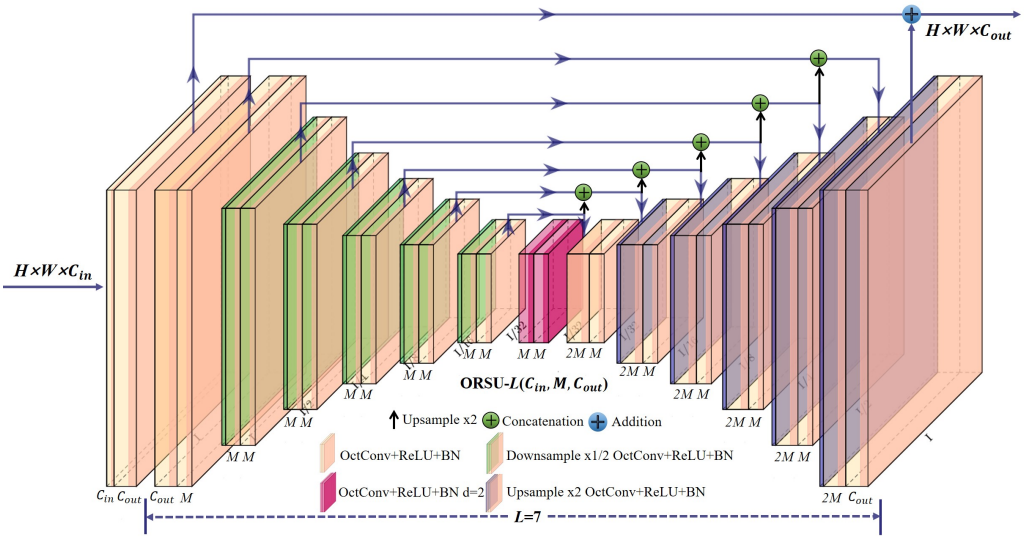


Figure 2. The designed octave residual U-block (ORSU block). L is the number of encoder layers, C_{in} and C_{out} denote the input and output channels, and M is the number of channels in the inner layer of the ORSU.

3. A residual connection for fusing local features and the multi-scale features through the summation of: $F_1(x) + \mu(F_1(x))$.

Like the RSU block, the ORSU block can capture intra-stage multi-scale features without degradation of the high-resolution features. The main difference between the design of the ORSU and RSU blocks is that the ORSU block replaces vanilla convolution with octave convolution (OctConv). CNNs have achieved outstanding achievements in many computer vision tasks. However, behind the high accuracy, there is a lot of spatial redundancy that cannot be ignored [22]. As with the decomposition of the spatial frequency components of natural images, OctConv decomposes the output feature maps of a convolutional layer into high- and low-frequency feature maps stored in different groups (see Figure 3). Therefore, through the information sharing between neighboring locations, the spatial resolution of the low-frequency group can be safely reduced and the spatial redundancy can also be reduced. In addition, OctConv performs the corresponding (low-frequency) convolution on the low-frequency information, effectively enlarging the receptive field in the pixel space. Therefore, the use of OctConv empowers the network to further reduce the computational and memory overheads while retaining the designed advantages of the RSU block. A computational cost comparison between the proposed ORSU block and the RSU block is proposed in Table 1.

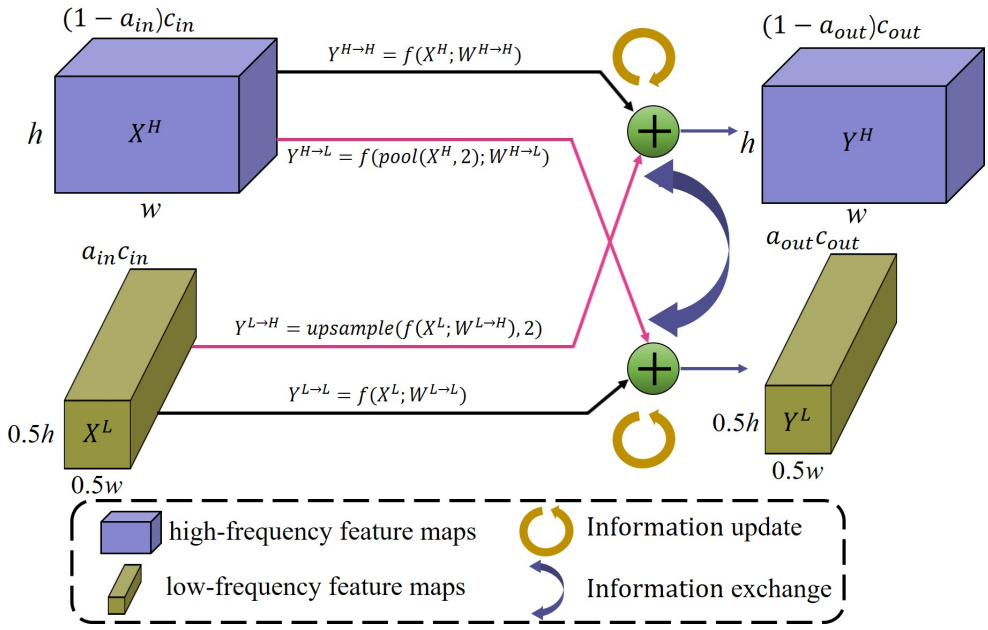


Figure 3. The concept of octave convolution [22]. $f(X;W)$ denotes the convolution function with weight parameters W, $pool(X,2)$ indicates spatial average pooling with kernel size 2×2 and stride 2, and $upsample(X)$ indicates an up-sampling operation by a factor of 2.

Table 1. Comparison of the computational costs for blocks and networks. All results were obtained using an open-source neural network analyzer (https://github.com/Swall0w/torchstat).

Blocks	**FLOPS**	**Memory**	**MAdd**
RSU-7	4.39 GFLOPS	138.67MB	8.74 MAdd
ORSU-7	**2.41 GFLOPS**	**123.19MB**	**4.79 MAdd**
Networks	**FLOPS**	**Memory**	**MAdd**
U^2-Net	37.67 GFLOPS	444.25MB	75.20 MAdd
U^2-ONet	**21.88 GFLOPS**	**418.56MB**	**43.64 MAdd**

The results of both blocks and networks are calculated based on an input feature map of dimension $256 \times 256 \times 3$. The dimension of the output feature map of the blocks is $256 \times 256 \times 64$. FLOPS shows the theoretical number of floating point operations. GFLOPS is the giga floating point operations. Memory denotes memory usage. MAdd is the theoretical amount of multiply–adds.

3.3. U^2-ONet

Inspired by U^2-Net [23], we propose the novel U^2-ONet, whose exponential notation is level 2 of the nested U-structure. As shown in Figure 1, each stage of U^2-ONet is filled by a well-configured ORSU block, and there are 11 stages that form a large U-structure. In general, U^2-ONet consists of three main parts:

1. The six-stage encoder. Detailed configurations are presented in Table 2. The number "L" behind the "ORSU-" denotes the height of the blocks. C_{in}, M, and C_{out} represent the input channels, middle channels, and output channels of each block, respectively. A larger value of L is used to capture more large-scale information of the feature map, with larger height and width. In both the *En*_5 and *En*_6 stages, ORSU-4F blocks are used, which are the dilated version of the ORSU blocks using dilated convolution (see Figure 1), because the resolution of the feature maps in these two stages is relatively low.

Table 2. Detailed configuration of the ORSU block at each stage. Except for C_{in} of En_1, the encoders and decoders adopt the same parameter configuration as in U^2-Net [23].

	Stages					
	En_1	**En_2**	**En_3**	**En_4**	**En_5**	**En_6**
Block	ORSU-7	ORSU-6	ORSU-5	ORSU-4	ORSU-4F	ORSU-4F
C_{in}	15	64	128	256	512	512
M	32	32	64	128	256	256
C_{out}	64	128	256	512	512	512
	De_5	**De_4**	**De_3**	**De_2**	**De_1**	
Block	ORSU-4F	ORSU-4	ORSU-5	ORSU-6	ORSU-7	
C_{in}	1024	1024	512	256	128	
M	256	128	64	32	16	
C_{out}	512	256	128	64	64	

2. The five-stage decoder has a similar structure to the symmetrical encoder stages (see Figure 1 and Table 2). The concatenation of the upsampling feature map of the previous stage and the upsampling feature map of the symmetric encoder stage is the input for each decoder stage.
3. The last part is a multi-scale attention mechanism attached to the decoder stages and the last encoder stage. At each level of the network, we add an attention module including channel and spatial attention mechanisms to eliminate the aliasing effect that should be eliminated by 3×3 convolution, inspired by [24] (see Figure 4) and [52]. At the same time, the channel attention mechanism is used to assign different significances to the channels of the feature map, and the spatial attention mechanism is used to discover which parts of the feature map are more important, so that the saliency of the spatial dimension of the moving objects is enhanced. Compared to U^2-Net for salient object detection, we maintain a deep architecture with high resolution for moving object segmentation while further enhancing the effect, reducing the computational and memory costs (see Tables 1 and 3).

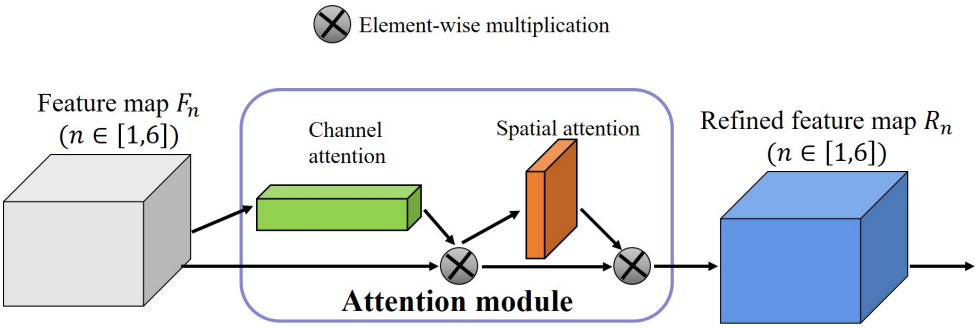


Figure 4. Overview of the convolutional block attention module (CBAM) [24].

3.4. Training Supervision Strategy

Generally speaking, the standard top-most supervised training is not a problem for relatively shallow networks. However, for extremely deep networks, the network will slow down, not converge, or converge to a local optimum due to the vanishing gradient problem during gradient back-propagation. The deeply supervised network (DSN) [53] was proposed to alleviate the optimization difficulties caused by gradient flows through long chains. However, it is still susceptible to problems, including interference of the hierarchical representation generation process and the inconsistency of the optimization goals. In the training process, we used a hierarchical training supervision strategy instead of using the standard top-most supervised training and a deep supervision scheme. For each level, we used both the standard binary cross-entropy loss (BCEloss) and Kullback–Leibler

divergence loss (KLloss), inspired by [54], to calculate the loss. By adding a pairwise probability prediction-matching loss (KLloss) between any two levels, we promote multi-level interaction between the different levels. The optimization objectives of the losses in the different levels are consistent, thus ensuring the robustness and generalization performance of the model. The ablation study in Section 4.1.3 proves the effectiveness of the hierarchical training supervision strategy. The binary cross-entropy loss is defined as:

$$l_{BCE} = -\sum_{(i,j)}^{(M,N)} [G_{(i,j)} \log S_{(i,j)} + (1 - G_{(i,j)}) \log (1 - S_{(i,j)})], \quad (1)$$

and the Kullback–Leibler divergence loss is defined as:

$$l_{KL} = \sum_{(i,j)}^{(M,N)} G_{(i,j)} \log \frac{G_{(i,j)}}{S_{(i,j)}}, \quad (2)$$

where (i, j) are the pixel coordinates and (M, N) are the height and width of the image. $G_{(i,j)}$ and $S_{(i,j)}$ denote the pixel values of the ground truth and the predicted moving object segmentation result, respectively. The proposed training loss function is defined as:

$$\zeta = \sum_{h=1}^{H} w_{BCE}^{(h)} l_{BCE}^{(h)} + \sum_{h=1}^{H} w_{KL}^{(h)} l_{KL}^{(h)}, \quad (3)$$

where $l_{BCE}^{(h)}$ and $l_{KL}^{(h)}$ (M = 6, as the Sup1, Sup2, ..., Sup6 in Figure 1) respectively denote the binary cross-entropy loss and the Kullback–Leibler divergence loss of the side output moving object segmentation result $S_{side}^{(h)}$. $w_{BCE}^{(h)}$ and $w_{KL}^{(h)}$ are the weights of each loss term. We try to minimize the overall loss ζ of Equation (3) in the training process. In the test process, we choose the $S_{side}^{(h)}$ (M = 1) as the final moving object segmentation result.

3.5. Post-Processing

Through the output of the network, the result of the moving object segmentation can be obtained, which is that the foreground moving objects and background are separated. However the instance-level moving object segmentation result cannot be obtained. In order to obtain instance-level results, the semantic instance label mask from Mask-RCNN is fused with the contour extraction results of the motion segmentation map. The geometric contours of the motion segmentation map can improve the quality of the semantic instance mask boundaries, determine whether the instance object is moving, and find new moving objects that are not associated with a particular semantic class. Meanwhile, the semantic instance mask can provide the category label of some moving objects and accurate boundaries to distinguish overlapping objects for the motion segmentation map.

The contour extraction method follows the approach proposed in [55], which utilizes topological structural analysis of digitized binary images to obtain the multiple closed contours of the motion segmentation map. For each motion contour C_i, we calculate the overlap of each semantic instance mask m_j and C_i to associate m_j and C_i. Only if this overlap is greater than a threshold—in our experiments, $80\% \cdot |m_j|$, where $\cdot|m_j|$ denotes the number of pixels belonging to the mask m_j—is m_j associated with C_i. Finally, we obtain the instance-level moving object segmentation result and segment new objects according to the number of semantic instance masks associated with each C_i (see Algorithm 1).

Algorithm 1 Instance-level moving object segmentation

Require: Each motion contour C_i;
Each instance semantic mask m_j;
Results of the semantic instance masks associated with C_i;
Judgment threshold t (t was usually 200 in our experiments);

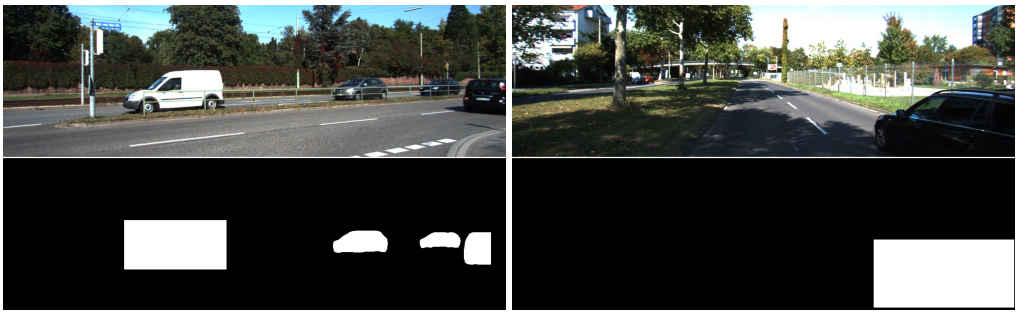
Ensure: Each moving object instance and its mask;

for each motion contour $C_i \in$ the current motion segmentation map **do**
  if the number of semantic instance masks associated with the motion contour $C_i >$ 1 **then**
    for each semantic instance mask m_j associated with the motion contour C_i **do**
      m_j is output as the mask for a moving object instance.
      the number of moving object instances ← the number of moving object instances + 1.
    end for
  end if
  if the number of semantic instance masks associated with the motion contour C_i == 1 **then**
    The area contained in motion contour C_i is assigned as the mask for associated semantic instance j.
    m_j is output as the mask for a moving object instance.
    the number of moving object instances ← the number of moving object instances + 1.
  end if
  if the number of semantic instance masks associated with the motion contour $C_i <$ 1 **then**
    if the length of the motion contour $C_i > t$ **then**
      The area contained in motion contour C_i is output as the mask for a new moving object instance.
      the number of moving object instances ← the number of moving object instances + 1.
    end if
  end if
end for

4. Experiments

Datasets: We evaluated the proposed method on several commonly used benchmark datasets: the Freiburg–Berkeley Motion Segmentation (FBMS) dataset [56], the Densely Annotated Video Segmentation (DAVIS) dataset [28,57], the YouTube Video Object Segmentation (YTVOS) dataset [58], and the extended KittiMoSeg dataset [59] proposed in FuseMODNet [60]. For the FBMS, we evaluated on the test set using the model trained from the training set. However, the FBMS shows a large number of annotation errors. We therefore used a corrected version of the dataset linked from the original dataset's website [61]. For the DAVIS dataset, DAVIS 2016 [28] is made up of 50 sequences containing instance segmentation masks for only the moving objects. Unlike DAVIS 2016, the DAVIS 2017 dataset [57] contains sequences providing instance-level masks for both moving and static objects, but not all of its sequences are suitable for our model. Therefore, we trained

the proposed model on DAVIS 2016 and used a subset of the DAVIS 2017 dataset called DAVIS-Moving, as defined in [20], for the evaluation. For the YTVOS dataset containing both labeled static and moving objects, we also used the YTVOS-Moving dataset introduced by [20], which selects sequences where all the moving objects are labeled. For the extended KittiMoSeg dataset, there are many images in this dataset that do not contain moving objects, for which is there is no label, or for which some labels are ambiguous, as shown in Figure 5, where the (static) background is wrongly segmented into objects and the cars are labeled roughly with a square area. We manually selected 5315 images for training and 2116 images for evaluation from the extended KittiMoSeg dataset, where the moving objects are accurately labeled.



(**a**) (**b**)

Figure 5. Some ambiguous labels in the extended KittiMoSeg dataset. Like (**a**) and (**b**), some moving objects are labeled roughly with a square area, and some regions of the background are also wrongly labeled as moving objects.

Implementation Details: We trained the proposed network from scratch, and all of the convolutional layers were initialized by [62]. Stochastic gradient descent (SGD) with an initial learning rate of 4×10^{-2} was used for the optimization, and its hyperparameter settings were as follows: momentum = 0.9, weight_decay = 0.0001. We trained for 20 epochs using a batch size of 4. Both the training and testing were conducted on a single NVIDIA Tesla V100 GPU with 16 GB memory, along with the PyTorch 1.1.0 and Python 3.7 deep learning frameworks. The results use the precision (P), recall (R), and F-measure (F), as defined in [56], as well as the mean intersection over union (IoU) for the evaluation metrics.

4.1. Ablation Studies

4.1.1. ORSU Block Structure

An ablation study on the blocks was undertaken to verify the effectiveness of the proposed ORSU block structure. The attention mechanism was removed in the proposed network, and then the ORSU blocks were replaced with the RSU blocks from U^2-Net to obtain the network called U^2-Net$_{6bk}$(−a). U^2-Net$_{6bk}$(−a) is a network for moving object segmentation that uses the backbone of U^2-Net and calculates the BCEloss and KLloss of six levels, as we designed. The results are shown in Tables 1 and 3. After replacing the RSU blocks with the proposed ORSU blocks, the memory usage drops by 25.69 MB and the computational cost falls by nearly 40%. For both the video foreground segmentation and multi-object motion segmentation, the network with ORSU blocks improves the precision by over 2.4%, the recall by about 1.0%, the F-measure by over 1.55%, and the IoU by over 1.51%. At the same time, it can be noted that the increase in value of the evaluation metrics in the multi-object motion segmentation is higher than that in the video foreground segmentation. It is worth noting that the improvement in precision is the most obvious, indicating that the ORSU blocks help the network to better learn motion information and more accurately segment moving objects. Therefore, it can be proved that the designed ORSU blocks are superior to the RSU blocks in motion segmentation tasks.

4.1.2. Attention Mechanism

As mentioned above, the addition of the attention mechanism introduces spatial and channel attention, making the moving objects more salient in the segmentation result. This ablation study was conducted to validate the effectiveness of adding the attention mechanism. U^2-$ONet_{6bk}$(-a) without the attention mechanism is compared with the complete network, called U^2-$ONet_{6bk}$. Table 3 shows that adding the attention mechanism improves the precision by about 1%, the recall by over 2.37%, the F-measure by over 1.64%, and the IoU by over 0.87%. Differing from the ablation study for the blocks, the increase in the value of the evaluation metrics after adding the attention mechanism in the video foreground segmentation is higher than that in the multi-object motion segmentation. Meanwhile, it can be noted that the improvement in recall is the most obvious, indicating that the designed multi-scale attention mechanism helps the network to better discover more moving objects. It is done by introducing global contextual information and capturing the spatial details around moving objects to enhance the saliency of the spatial dimension of the moving objects.

Table 3. Results of the ablation studies on the blocks and attention mechanism.

	Video Foreground Segmentation				**Multi-Object Motion Segmentation**			
	P (Precision)	**R (Recall)**	**F (F-Measure)**	**IoU**	**P (Precision)**	**R (Recall)**	**F (F-Measure)**	**IoU**
U^2-$Net_{6bk}(-a)$	84.6748	78.7881	80.2232	70.9541	79.8170	74.4494	75.7438	70.3260
U^2-$ONet_{6bk}(-a)$	87.0821	79.7173	81.6738	72.4611	82.3621	75.5477	77.3492	72.1066
U^2-$ONet_{6bk}$	88.3126	82.5591	83.7723	74.1656	83.1169	77.9250	78.9910	72.9785

Video foreground segmentation denotes segmenting the scene into static background or foreground moving objects without distinguishing object instances. Multi-object motion segmentation is segmenting each moving object instance in the scene.

4.1.3. Training Supervision

In order to prove the effectiveness of the multi-level loss calculation strategy, we evaluated the U^2-$ONet_{bk}$ network when calculating the BCEloss and KLloss from one to six levels. Table 4 shows that the overall effect of the network in calculating the loss improves from two to four levels as the number of levels increases. The overall effect when using from four to six levels appears to decline first, and then increases as the number of levels increases. In summary, it can be proved that the proposed multi-level loss calculation training strategy further improves the effect of the deep network, but it is not that more levels result in a better effect. For the proposed network, it works best when using six levels or four levels.

To further demonstrate the superiority of the calculation of both the BCEloss and KLloss, multiple ablation experiments were conducted. From one to six levels, we compared the network only calculating the BCEloss, called U^2-$ONet_b$, and the network calculating both the BCEloss and KLloss, called U^2-$ONet_{bk}$. The results are listed in Table 4. From the results, when calculating the loss for three and four levels, the addition of KLloss improves the effect of the network, in general, although the improvement in each metric is not obvious, unlike in the previous ablation studies. Overall, we think that the addition of KLloss has the potential to improve the effect of the multi-level network to a certain extent. At the same time, the addition of KLloss improves the precision and makes the network more accurately segment moving objects, which is what is needed for some practical applications.

Table 4. Results of the ablation study on the training supervision.

	Video Foreground Segmentation				Multi-object Motion Segmentation			
	P (Precision)	**R (Recall)**	**F (F-Measure)**	**IoU**	**P (Precision)**	**R (Recall)**	**F (F-Measure)**	**IoU**
U^2-ONet$_{1bk}$	87.1959	82.9170	83.3599	73.7483	82.0288	78.2415	78.5725	72.7251
U^2-ONet$_{1b}$	87.6808	82.4263	83.2078	73.4435	82.4819	77.7830	78.4259	72.9288
U^2-ONet$_{2bk}$	86.5249	83.6013	83.2855	73.5262	81.3052	78.8389	78.4271	73.2597
U^2-ONet$_{2b}$	87.5558	83.3884	83.7176	74.0066	82.2579	78.6416	78.8341	73.2570
U^2-ONet$_{3bk}$	88.0516	82.7291	83.6525	74.0434	82.9008	78.0600	78.9054	73.2325
U^2-ONet$_{3b}$	88.2510	82.3962	83.5610	73.9233	83.0433	77.7766	78.7818	73.2416
U^2-ONet$_{4bk}$	88.1412	83.1088	83.9566	74.4435	82.9590	78.4306	79.1666	73.2620
U^2-ONet$_{4b}$	88.2951	82.4503	83.7305	74.1254	83.1775	77.8654	79.0048	73.1279
U^2-ONet$_{5bk}$	88.2397	82.2916	83.4473	73.7114	83.0693	77.6426	78.6916	72.8023
U^2-ONet$_{5b}$	88.8535	82.2117	83.7729	74.2275	83.6419	77.6087	79.0023	72.7234
U^2-ONet$_{6bk}$	88.3126	82.5591	83.7723	74.1656	83.1169	77.9250	78.9910	72.9785
U^2-ONet$_{6b}$	88.1056	82.9051	83.8449	74.1089	82.9020	78.2278	79.0357	73.3622

Best results are highlighted in red with second best in blue.

4.2. Comparison with Prior Work

Official FBMS: The proposed method was evaluated against prior works on the standard FBMS test set using the model trained from the FBMS training set. The input image size was (512, 640). Since some of the methods compared only provide metrics for the multi-object motion segmentation task, only the metrics for this task are compared. In order to compare with the other methods, to indicate the accuracy of the segmented object count, the ΔObj metric was added, as defined in [19]. As shown in Table 5, the proposed model performs the best in recall. In terms of recall and F-measure, it outperforms CCG [18], OBV [19], and STB [50] by over 16.5% and 6.4%, respectively. The qualitative results are shown in Figure 6.

Table 5. FBMS results using the official metric.

	Multi-Object Motion Segmentation				
	P	**R**	**F**	**IoU**	**ΔObj**
CCG [18]	74.23	63.07	64.97	–	4
OBV [19]	75.90	66.60	67.30	–	4.9
STB [50]	87.11	66.53	75.44	–	–
TSA [20]	88.60	80.40	84.30	–	–
Ours	84.80	83.10	81.84	79.70	4.9

The best results are highlighted in red, with the second best in blue.

Figure 6. Qualitative results for the Freiburg–Berkeley Motion Segmentation (FBMS) dataset.

DAVIS and YTVOS: The proposed method was further evaluated on the DAVIS-Moving dataset using the model trained from DAVIS 2016 and evaluated on the YTVOS-Moving test set using the model trained from the YTVOS-Moving training set, as defined in [20]. The input image size for both datasets was (512, 896). The results are listed in Table 6. For DAVIS-Moving, the proposed approach outperforms TSA [20] by about 4.8% in precision and about 0.8% in F-measure. Unlike the FBMS and DAVIS datasets, the YTVOS-Moving dataset contains many moving objects that are difficult to segment, such as snakes, octopuses, and camouflaged objects. Therefore, the metrics for YTVOS-Moving are much lower than the metrics for the previous datasets. However, in YTVOS-Moving, the proposed method still outperforms TSA by about 4.2% in recall and about 1.6% in F-measure. The qualitative results are shown in Figures 7 and 8.

Table 6. Results for the Densely Annotated Video Segmentation (DAVIS)-Moving and YouTube Video Object Segmentation (YTVOS)-Moving datasets, as defined in [20].

Multi-Object Motion Segmentation					
Dataset		**P**	**R**	**F**	**IoU**
DAVIS-Moving	TSA [20]	78.30	**78.80**	78.10	–
	Ours	**83.12**	77.93	**78.99**	72.98
YTVOS-Moving	TSA [20]	74.50	66.40	68.30	–
	Ours	**74.64**	**70.56**	**69.93**	65.67

The best results are highlighted in bold.

Figure 7. Qualitative results for the DAVIS-Moving dataset.

Figure 8. Qualitative results for the YTVOS-Moving dataset.

Extended KittiMoSeg: Finally, the proposed approach was evaluated on the extended KittiMoSeg dataset using our split training set and test set. The proposed method was evaluated on the test set using the model trained with the training set of the extended

KittiMoSeg. The input image size for this dataset was (384, 1280). The extended KittiMoSeg dataset is based on subsets of the Kitti dataset and is for real autonomous driving scenarios. Therefore, this dataset includes continuous and fast-forward moving cameras and multiple fast-moving vehicles of different sizes. These pose different challenges to the generic segmentation of moving objects. As the annotations in this dataset are all binary annotations, i.e., only the static background and moving objects are segmented, without distinguishing object instances, only the results for the video foreground segmentation are compared. Since the complete FuseMODNet combines RGB and LiDAR (Light Detection and Ranging) data, the proposed approach is compared with FuseMODNet using RGB and rgbFlow without using LiDAR. As shown in Table 7, the proposed method significantly outperforms FuseMODNet by 13.29% IoU. Since MODNet uses the unexpanded KittiMoSeg, including about 1950 frames only, the model from the training set of KittiMoSeg was used to evaluate on the testing set of KittiMoSeg. As shown in Table 8, the proposed method outperforms MODNet in all metrics. In term of precision and IoU, the proposed method outperforms MODNet by 11.9% and 10.26%, respectively. Therefore, it can be proved that the proposed approach is likely to have good application prospects in the field of autonomous driving. The qualitative results are shown in Figure 9.

Table 7. Quantitative evaluation on the extended KittiMoSeg dataset.

	Video Foreground Segmentation			
	P	**R**	**F**	**IoU**
FuseMODNet (RGB + rgbFlow) [60]	–	–	–	49.36
Ours	74.57	67.62	68.88	**62.65**

The best results are highlighted in bold.

Table 8. Quantitative evaluation on the unexpanded KittiMoSeg dataset.

	Video Foreground Segmentation			
	P	**R**	**F**	**IoU**
MODNet [59]	56.18	70.32	62.46	45.41
Ours	**68.08**	**72.36**	**64.23**	**55.67**

The best results are highlighted in bold.

Figure 9. Qualitative results for the extended KittiMoSeg dataset.

New robot dataset: In order to prove that the proposed method has the ability to discover moving objects that are not considered in the trained model, qualitative experiments were conducted using our own dataset, which is called the new robot dataset. This dataset is of a mobile robot, without semantic labels (see Figure 10). The models trained using the DAVIS dataset were used for testing, without any training on the new robot dataset. For quantitative evaluation, we acquired a 900-frame-long sequence and manually provided 2D ground-truth annotations for the masks of the moving objects. The qualitative and quantitative results are shown in Figure 11 and Table 9. Our method was compared with the only method with open source code—TSA [20]. TSA was also not trained with this

dataset. We used the TSA's open-source model directly for evaluation and this open-source model cannot segment the moving object without semantic labels in the new robot dataset. It can be seen that the proposed method can segment the moving object without semantic labels in the new robot dataset better, proving the efficiency of the proposed method.

(**a**) (**b**) (**c**)

Figure 10. (**a**) The mobile robot without semantic labels in the new robot dataset. (**b**) Instance segmentation result from Mask-RCNN. (**c**) Semantic segmentation result from PSPnet [63]. Neither method can segment this object well.

(**a**) (**b**) (**c**)

Figure 11. Qualitative results of the proposed method with the new robot dataset are shown in (**a**–**c**).

Table 9. Quantitative evaluation on the new robot dataset.

	Multi-Object Motion Segmentation			
	P	**R**	**F**	**IoU**
TSA [20]	–	–	–	–
Ours	63.79	50.62	56.28	46.79

"–" indicates that this method can not segment this moving object, so this metric cannot be obtained.

5. Conclusions

In this paper, we proposed a two-level nested U-structure network with a multi-scale attention mechanism, called U^2-ONet, for moving object segmentation. Each stage of U^2-ONet is filled with the newly designed octave residual U-blocks (ORSU blocks) based on octave convolution, which enable U^2-ONet to capture both local and global information with a high resolution while reducing the spatial redundancy and computational burden in the CNN. We also designed a hierarchical training supervision strategy that calculates both the BCEloss and KLloss at all six levels to improve the effectiveness of the deep network. The experimental results obtained on several general moving object segmentation datasets show that the proposed approach is a state-of-the-art method. Even in some challenging datasets, such as YTVOS-Moving (which includes camouflaged objects and tiny objects) and extended KittiMoSeg (which includes fast camera motion and non-salient moving cars), the proposed method still achieves a good performance. Experiments on our own new robot dataset also proved that this approach has the ability to segment new objects.

In the near future, we will further leverage the anti-noise ability of octave convolution to combine with the ability of the ORSU blocks to extract multi-scale features. We will also introduce a network for image dehazing and detraining in order to enhance the images before being fed through the network. Finally, we will attempt to enable U^2-ONet to still achieve a good performance under extremely complex conditions, including rain, fog, snow, and motion blur, as well as in high-noise scenarios.

Author Contributions: B.L. guided the algorithm design. C.W. wrote the paper. C.W. designed the whole experiments. C.L. designed the whole framework. J.L. helped organize the paper. X.S., Y.W.

and Y.G. provided advice for the preparation of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (2019YFC0121502), National Key R&D Program of China (2017YFB1302400), and "the Fundamental Research Funds for the Central Universities".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available because the data have not been sorted out.

Acknowledgments: The numerical calculations in this paper were done on the supercomputing system in the Supercomputing Center of Wuhan University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 37. [CrossRef]
2. Runz, M.; Buffier, M.; Agapito, L. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 16–20 October 2018; pp. 10–20.
3. Wang, R.; Wan, W.; Wang, Y.; Di, K. A New RGB-D SLAM Method with Moving Object Detection for Dynamic Indoor Scenes. *Remote. Sens.* **2019**, *11*, 1143. [CrossRef]
4. Wang, Z.; Zhang, Q.; Li, J.; Zhang, S.; Liu, J. A Computationally Efficient Semantic SLAM Solution for Dynamic Scenes. *Remote Sens.* **2019**, *11*, 1363. [CrossRef]
5. Zha, Y.; Wu, M.; Qiu, Z.; Dong, S.; Yang, F.; Zhang, P. Distractor-Aware Visual Tracking by Online Siamese Network. *IEEE Access* **2019**, *7*, 89777–89788. [CrossRef]
6. Amiranashvili, A.; Dosovitskiy, A.; Koltun, V.; Brox, T. Motion Perception in Reinforcement Learning with Dynamic Objects. *Conf. Robot. Learn. (CoRL)* **2018**, *87*, 156–168.
7. Shah, S.; Dey, D.; Lovett, C.; Kapoor, A. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*; Springer: Cham, Switzerland, 2017.
8. Baradel, F.; Wolf, C.; Mille, J.; Taylor, G.W. Glimpse Clouds: Human Activity Recognition From Unstructured Feature Points. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
9. Chen, B.; Huang, S. An Advanced Moving Object Detection Algorithm for Automatic Traffic Monitoring in Real-World Limited Bandwidth Networks. *IEEE Trans. Multimed.* **2014**, *16*, 837–847. [CrossRef]
10. Bouwmans, T.; Zahzah, E.H. Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance. *Comput. Vis. Image Underst.* **2014**, *122*, 22–34. [CrossRef]
11. Wang, C.; Luo, B.; Zhang, Y.; Zhao, Q.; Yin, L.; Wang, W.; Su, X.; Wang, Y.; Li, C. DymSLAM:4D Dynamic Scene Reconstruction Based on Geometrical Motion Segmentation. *arXiv* **2020**, arXiv:cs.CV/2003.04569.
12. Zhao, X.; Qin, Q.; Luo, B. Motion Segmentation Based on Model Selection in Permutation Space for RGB Sensors. *Sensors* **2019**, *19*, 2936. [CrossRef]
13. Zhang, Y.; Luo, B.; Zhang, L. Permutation preference based alternate sampling and clustering for motion segmentation. *IEEE Signal Process. Lett.* **2017**, *25*, 432–436. [CrossRef]
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
15. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
17. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Bideau, P.; RoyChowdhury, A.; Menon, R.R.; Learned-Miller, E. The best of both worlds: Combining cnns and geometric constraints for hierarchical motion segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 508–517.
19. Xie, C.; Xiang, Y.; Harchaoui, Z.; Fox, D. Object discovery in videos as foreground motion clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9994–10003.

20. Dave, A.; Tokmakov, P.; Ramanan, D. Towards segmenting anything that moves. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019; pp. 1493–1502.
21. Muthu, S.; Tennakoon, R.; Rathnayake, T.; Hoseinnezhad, R.; Suter, D.; Bab-Hadiashar, A. Motion Segmentation of RGB-D Sequences: Combining Semantic and Motion Information Using Statistical Inference. *IEEE Trans. Image Process.* **2020**, *29*, 5557–5570. [CrossRef] [PubMed]
22. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; Feng, J. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3435–3444.
23. Qin, X.; Zhang, Z.; Huang, C.; Dehghan, M.; Zaiane, O.R.; Jagersand, M. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognit.* **2020**, *106*, 107404. [CrossRef]
24. Woo, S.; Park, J.; Lee, J.Y.; So Kweon, I. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
25. Papazoglou, A.; Ferrari, V. Fast Object Segmentation in Unconstrained Video. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Darling Harbour, Sydney, Australia, 2–8 December 2013; pp. 1777–1784. [CrossRef]
26. Faktor, A.; Irani, M. Video Segmentation by Non-Local Consensus voting. *BMVC* **2014**, *2*, 8.
27. Wang, W.; Shen, J.; Porikli, F. Saliency-aware geodesic video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3395–3402.
28. Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 724–732.
29. Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S.C.; Ling, H. Learning unsupervised video object segmentation through visual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3064–3074.
30. Wang, W.; Lu, X.; Shen, J.; Crandall, D.J.; Shao, L. Zero-shot video object segmentation via attentive graph neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9236–9245.
31. Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; Porikli, F. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3623–3632.
32. Peng, Q.; Cheung, Y. Automatic Video Object Segmentation Based on Visual and Motion Saliency. *IEEE Trans. Multimed.* **2019**, *21*, 3083–3094. [CrossRef]
33. Chen, Y.; Hao, C.; Liu, A.X.; Wu, E. Multilevel Model for Video Object Segmentation Based on Supervision Optimization. *IEEE Trans. Multimed.* **2019**, *21*, 1934–1945. [CrossRef]
34. Zhuo, T.; Cheng, Z.; Zhang, P.; Wong, Y.; Kankanhalli, M. Unsupervised online video object segmentation with motion property understanding. *IEEE Trans. Image Process.* **2019**, *29*, 237–249. [CrossRef]
35. Yang, Z.; Wei, Y.; Yang, Y. Collaborative video object segmentation by foreground-background integration. *arXiv* **2020**, arXiv:2003.08333.
36. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149.
38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
39. Kang, B.R.; Lee, H.; Park, K.; Ryu, H.; Kim, H.Y. BshapeNet: Object detection and instance segmentation with bounding shape masks. *Pattern Recognit. Lett.* **2020**, *131*, 449–455. [CrossRef]
40. Peng, S.; Jiang, W.; Pi, H.; Li, X.; Bao, H.; Zhou, X. Deep Snake for Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8533–8542.
41. Hurtik, P.; Molek, V.; Hula, J.; Vajgl, M.; Vlasanek, P.; Nejezchleba, T. Poly-YOLO: Higher speed, more precise detection and instance segmentation for YOLOv3. *arXiv* **2020**, arXiv:2005.13243.
42. Kong, S.; Fowlkes, C.C. Recurrent pixel embedding for instance grouping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9018–9028.
43. Neven, D.; Brabandere, B.D.; Proesmans, M.; Gool, L.V. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8837–8845.
44. Ying, H.; Huang, Z.; Liu, S.; Shao, T.; Zhou, K. Embedmask: Embedding coupling for one-stage instance segmentation. *arXiv* **2019**, arXiv:1912.01954.
45. Chen, L.; Strauch, M.; Merhof, D. Instance Segmentation of Biomedical Images with an Object-Aware Embedding Learned with Local Constraints. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Berlin/Heidelberg, Germany, 2019; pp. 451–459.

46. Xu, X.; Cheong, L.F.; Li, Z. 3D Rigid Motion Segmentation with Mixed and Unknown Number of Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]
47. Thakoor, N.; Gao, J.; Devarajan, V. Multibody structure-and-motion segmentation by branch-and-bound model selection. *IEEE Trans. Image Process.* **2010**, *19*, 1393–1402. [CrossRef]
48. Zhao, Q.; Zhang, Y.; Qin, Q.; Luo, B. Quantized Residual Preference Based Linkage Clustering for Model Selection and Inlier Segmentation in Geometric Multi-Model Fitting. *Sensors* **2020**, *20*, 3806. [CrossRef]
49. Sultana, M.; Mahmood, A.; Jung, S.K. Unsupervised Moving Object Detection in Complex Scenes Using Adversarial Regularizations. *IEEE Trans. Multimed.* **2020**, *1*. [CrossRef]
50. Shen, J.; Peng, J.; Shao, L. Submodular trajectories for better motion segmentation in videos. *IEEE Trans. Image Process.* **2018**, *27*, 2688–2700. [CrossRef]
51. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 2462–2470.
52. Li, C.; Luo, B.; Hong, H.; Su, X.; Wang, Y.; Liu, J.; Wang, C.; Zhang, J.; Wei, L. Object Detection Based on Global-Local Saliency Constraint in Aerial Images. *Remote Sens.* **2020**, *12*, 1435. [CrossRef]
53. Lee, C.Y.; Xie, S.; Gallagher, P.; Zhang, Z.; Tu, Z. Deeply-supervised nets. In *Artificial Intelligence and Statistics*; 2015; pp. 562–570.
54. Li, D.; Chen, Q. Dynamic Hierarchical Mimicking Towards Consistent Optimization Objectives. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7642–7651.
55. Suzuki, S.; Abe, K. Topological structural analysis of digitized binary images by border following. *Comput. Vis. Graph. Image Process.* **1985**, *30*, 32–46. [CrossRef]
56. Ochs, P.; Malik, J.; Brox, T. Segmentation of moving objects by long term video analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 1187–1200. [CrossRef] [PubMed]
57. Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; Van Gool, L. The 2017 davis challenge on video object segmentation. *arXiv* **2017**, arXiv:1704.00675.
58. Xu, N.; Yang, L.; Fan, Y.; Yue, D.; Liang, Y.; Yang, J.; Huang, T. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv* **2018**, arXiv:1809.03327.
59. Siam, M.; Mahgoub, H.; Zahran, M.; Yogamani, S.; Jagersand, M.; El-Sallab, A. Modnet: Motion and appearance based moving object detection network for autonomous driving. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2859–2864.
60. Rashed, H.; Ramzy, M.; Vaquero, V.; El Sallab, A.; Sistu, G.; Yogamani, S. Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
61. Bideau, P.; Learned-Miller, E. A detailed rubric for motion segmentation. *arXiv* **2016**, arXiv:1610.10033.
62. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
63. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 2881–2890.