2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Self-Supervised Motion Segmentation with Confidence-Aware Loss Functions for Handling Occluded Pixels and Uncertain Optical Flow Predictions

Chung-Yu Chen[1], Bo-Yun Lai[1], Ying-Shiuan Huang[1], Wen-Chieh Lin[1] and Chieh-Chih Wang[2]

***Abstract*— In driving scenarios, motion segmentation is a crucial and fundamental component that is needed for many tasks. Recently, a self-supervised multitasking framework was proposed for driving scenarios. It simultaneously trains motion segmentation, optical flow, depth, and ego-motion models without annotated data. The self-supervised architecture derives training signals from training data via loss functions. If these loss functions lack robustness, they may result in model inaccuracies. To reduce the bad influences of occlusion and optical flow estimation errors on motion segmentation, we propose two loss functions: (1) Soft-Per-Pixel-Minimum (Soft-PPM) loss that excludes occluded pixels while balancing the contribution of each frame on the loss function temporally; (2) Flow difference loss that excludes pixels with unclear motion states to diminish the effect of optical flow estimation errors. Our loss function design is based on the key insight that information such as depth and optical flow can be used to train motion segmentation models and act as a reliable measure for pixels during training. Our approach can improve segmentation accuracy for both moving and static objects and has achieved IoU scores on moving and static classes comparable to the state-of-the-art methods on the KITTI dataset.**

***Index Terms*— Motion Segmentation, Self-driving, Deep Learning, Self-supervised Learning, Loss Function.**

I. Introduction

Motion segmentation is a fundamental problem in computer vision and robotics. It is a building block for numerous subsequent tasks such as monocular depth prediction [1]–[3], visual odometry [4], and traffic monitoring [5]. Among these tasks, motion segmentation for autonomous driving comes with additional challenges, including large depth variations, significant egomotion, and multiple moving objects. To address these problems, several neural models [6], [7] have been proposed. While these models exhibit the ability to capture high-level features effectively, they suffer from difficulties in gathering enough training data and pixel-wise labeling, hence impeding their scalability. This was resolved later with the introduction of unsupervised learning [8], [9].

Self-supervised methods extract training signals from the training data itself, commonly done by exploiting the interdependency of explicit information from multiple sub-tasks during training to obtain implicit information. In the context of autonomous driving, a self-supervised joint learning framework, *Competitive Collaboration (CC)* was proposed [10]. It consists of four modules: depth, ego-motion, optical flow, and motion segmentation. During training, these correlated sub-networks utilize the output of others to supervise each other, forming a self-supervised framework.

(a) Trained with all pixels (CC) [10] (b) Trained with "good" pixels (Ours)

Fig. 1: Motion segmentation results of CC and our method. Our method detects and excludes pixels that are occluded or with uncertain optical flow during training, improving the accuracy and robustness of results.

In CC [10], the Motion-Net (MN) that generates motion segmentation results acts as a moderator to guide the training of other sub-networks with two loss functions. First, the photometric loss is computed based on rigid body projection and image warping. By calculating the rigid flow for a static background using depth and ego-motion networks, all pixels in a sequence of images can be warped to a specific frame. This enables us to capture moving objects by identifying differences in the warped images. Second, the flow difference loss uses a flow difference mask to differentiate moving pixels from static ones by comparing the optical flow in a sequence of images with the rigid flow of the background.

Although the two loss functions are intuitive and efficient, we identify two key problems and propose the corresponding solutions: (1) Occlusion in scenes prohibits CC from synthesizing background appearance for the affected regions and causes it to falsely classify occluded pixels as moving. Therefore, we introduce the PPM loss [2] to CC and propose Soft-PPM, which reduces the weight assigned to the losses of occluded pixels and ensures more balanced gradient propagation across frames so that all frames contribute equally during training. (2) Errors in the rigid and optical flow predictions hinder the refinement of the MN. For this, we train the model exclusively on reliable pixels predicted by the differences between the two flows. With these modifications, our model improves the mean IoU by 10% compared to the baseline CC [10], and achieves better or comparable performance with state-of-the-art methods in the KITTI dataset [11].

[1]Chung-Yu Chen, Bo-Yun Lai, Ying-Shiuan Huang, and Wen-Chieh Lin are with the Dept. of Computer Science, National Yang Ming Chiao Tung Univ. c93294123@gmail.com; jonathanlai0717@gmail.com; s1042178@gmail.com; wclin@cs.nctu.edu.tw

[2]Chieh-Chih Wang is with the College of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, and with the Mechanical and Mechatronics Systems Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan. bobwang@ieee.org

II. Related Work

Motion segmentation has been extensively studied. In this section, we will only review learning-based approaches and introduce methods that address the occlusion problem. For a complete review of this topic, please see [12], [13].

Supervised Motion Segmentation methods acquire high-level motion information via training. The class-agnostic motion features can enhance a model's robustness to handle unseen objects. Cheng et al. proposed a two-branch architecture that jointly predicts object segmentation and optical flow [14]. Tokmakov et al. used a two-stream neural network and proposed the convolutional gated unit (ConvGRU) to explicitly model visual memory [7]. Dave et al. utilized motion cues from optical flow as a bottom-up signal to separate objects [15]. Shen et al. used specific architectures to extract patterns for motion segmentation [6]. These methods usually have higher accuracy compared to unsupervised methods but require a substantial amount of annotated training data.

Unsupervised Motion Segmentation aims to segment moving regions without human supervision. Yang et al. proposed an adversarial framework to detect moving objects [8]. The generator was trained to minimize information transfer from the background to the foreground. Lian et al. proposed a visual grouping method that is capable of handling articulation and reflection [9]. These methods eliminate the need for labeling data but do not consider significant depth variations. Our method models the background as a rigid body, making it more suitable for driving scenarios.

Self-Supervised Joint Learning Framework provides multiple types of information, including depth, pose, optical flow, and motion segmentation. These methods are designed primarily for driving scenes and can be trained without explicit labels. A pioneering work by Zhou et al. introduced an unsupervised learning framework to estimate depth and egomotion [1] . Ranjan et al. further extended the architecture and proposed *Competitive Collaboration* [10]. They formulated this problem as a three-player game. The MN plays the role of moderator and collaborates with other models to accomplish a meta-task. Jiao et al. introduced a "Rigidity from Motion" layer to estimate pixel rigidity by modeling the correlation between rigid flow and optical flow [3]. Hui et al. proposed an object motion decoder that combines predictions of optical flow and camera poses to mitigate artifacts caused by moving objects. A Recurrent Modulation Unit was also introduced to iteratively refine the fused features [16]. We adopted an architecture similar to CC [10] and proposed enhancements for the MN to improve the robustness of our model against occlusion and optical flow errors.

Occlusion-aware Methods. Occlusion is a common challenge in driving scenes. Godard et al. argued that 'good' correspondences should have low losses and proposed a PPM reprojection loss that integrates losses from multiple frames to exclude occluded pixels [2]. Janai et al. assumed that the motion of objects is approximately linear and proposed an optical flow model with explicit reasoning about occlusion [17]. Similarly to these methods, our occlusion-aware loss function utilizes the assumption of a stable motion state to integrate information from different frames.

III. Proposed Method

Our framework is based on the self-supervised joint learning framework, CC [10]. It includes a Motion-Net (MN) that collaborates with other sub-networks. The sub-networks and computational pipelines of the loss functions in our framework are shown in Fig. 2. The MN in CC often underperforms when there are occluded pixels or optical flow estimation errors. To resolve this, we made two modifications to the loss functions. First, we propose *Soft-PPM* to select non-occluded pixels for training. Soft-PPM is specifically designed for MN to ensure that the contributions of all frames to the training loss are balanced. Second, we propose *Conservative Flow Difference Mask (CFD-Mask)*, which doesn't enforce the algorithm to make definitive decisions for all pixels, thus reducing the number of classification errors.

A. Motion-Net with Self-supervised Learning

The training of the MN involves two loss functions: (1) the projection loss produced by the collaboration of multiple sub-networks, and (2) the Flow Difference Mask (FD-Mask) loss which combines rigid and optical flows to generate another mask distinct from the MN and brings additional information to improve both training and post-processing.

The projection loss uses optical and rigid flows to warp images in all reference frames to the target frame. By minimizing this loss, the MN and all sub-networks can be trained simultaneously. The warping function and photometric loss are defined as:

$$I'_t = W(I_t, F_t) \tag{1}$$

$$L_t^{pho} = \lambda L_1(I'_t, I_0) + (1-\lambda)(1 - SSIM(I'_t, I_0)) \tag{2}$$

Our framework operates on a 5-frame sequence with images $\{I_{-2}, I_{-1}, I_0, I_{+1}, I_{+2}\}$, where I_0 is the image of the target frame and I_t with $t \in T = \{-2, -1, +1, +2\}$ are images of

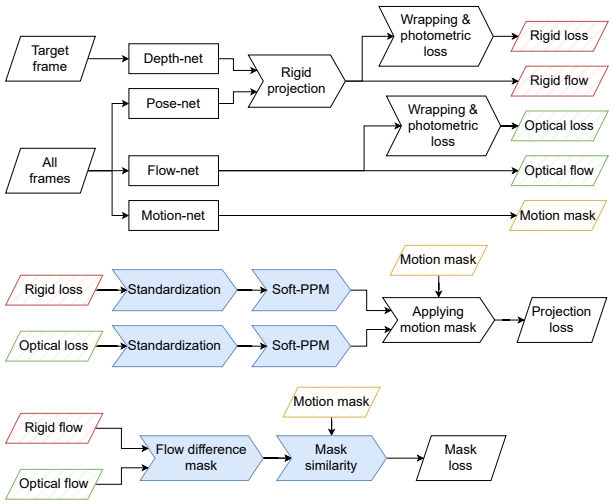


Fig. 2: Our framework consists of four sub-networks. The Expectation-Maximization (EM) strategy is used to iteratively refine each sub-network until convergence is achieved across all sub-networks. In this architecture, Motion-Net is influenced by the projection loss and mask loss.

reference frames. The warping function W takes an image I_t in reference frame t and the flow F_t to perform an inverse warping, where F_t is the flow from the target frame to the reference frame t. The resulting warped image I'_t should have an appearance similar to I_0 if the pixels are not occluded. L_1 is the per-pixel L1 loss between two images. The structural similarity index (SSIM) is used to measure image similarity [18]. λ is set to 0.003 as suggested in CC.

Our framework consists of a moderating MN, and two competitors each generating a flow field. One competitor is Flow-net that predicts the optical flow F^{opt} of each frame, and the other, comprising the Depth-net and Pose-net, generates the rigid flow F^{rig}. The rigid flow of a pixel at location x at frame t is calculated by $f_t^{rig} = x'_t - x$, where $x'_t = K \cdot P_t(D(x)K^{-1}x)$ represents the pixel's coordinate after projection. K is the camera's intrinsic matrix, P_t is the projection matrix of ego-motion from the target frame to frame t, and $D(x)$ is the depth of coordinate x in the target frame. P_t and $D(x)$ are the outputs of Pose-Net and Depth-Net, respectively. We compute the flow for all pixels and consolidate that information into a flow field F_t^{rig}.

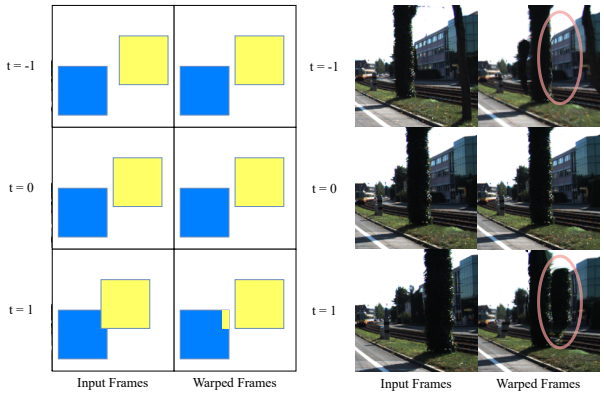
With F_t^{rig} and F_t^{opt}, we can calculate the photometric loss of each competitor using Eq. 2. The resulting $L_t^{pho,rig}$ is the rigid loss and $L_t^{pho,opt}$ is the optical loss. A motion mask M_t generated by the MN is applied to ensure that the Depth-net and Pose-net focus on static pixels while the Flow-net focuses on moving pixels. Mathematically, the overall photometric loss is calculated by $L_t^{pho,all} = M_t \odot L_t^{pho,rig} + (1 - M_t) \odot L_t^{pho,opt}$, where $\odot$ denotes element-wise multiplication. Each pixel in M_t has a continuous value within the range of $[0, 1]$. If the value is close to 1, the pixel is likely to be static.

Flow difference is another loss in the framework (the bottom pipeline in Fig. 2). It feeds information from both competitors back to the MN to refine MN performance. If a pixel is static, we can estimate its rigid flow by employing rigid projection. The resulting value should be similar to its optical flow. In contrast, if a pixel is moving, the values should be different. In CC [10], the FD-mask Ψ_t^f uses a threshold δ to determine whether a pixel located at x is moving (Eq. 3). This is a simple, yet efficient way for the MN to utilize information from other sub-networks. The FD-mask loss L^{Mask} can then be computed using Eq. 4.

$$\Psi_t^f(x) = \begin{cases} 1, & \text{if } |f_t^{opt}(x) - f_t^{rig}(x)| > \delta \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$L^{Mask} = \sum_{t \in T} H(\Psi_t^f, M_t) \tag{4}$$

M_t is the motion mask predicted by MN. H is cross-entropy to measure the difference between two masks. This measure can also be performed using L1 loss. During the training phase of MN, the values of the loss functions on static pixels of both competitors are similar. To enhance MN's ability to capture static elements, a bias is added to MN towards predicting static pixels: $L^{Bias} = \sum_{t \in T} H(1, M_t)$.



(a) Example of occlusion (b) Occlusion on images

Fig. 3: This figure illustrates the ghosting effect of image warping. When an object is occluded, warping it back to the target frame may capture part of the occluder. (a) shows that the upper right corner of the blue square is occluded by the yellow square. If the occluded image is warped back to the target frame, a "ghost" of the foreground object appears in the occluded region. (b) shows the ghosting effect on real images. The circled region is occluded by the tree. Typically, a pixel is only occluded before or after it appears unobstructed, not both.

B. Soft Per-Pixel-Minimum Loss (Soft-PPM)

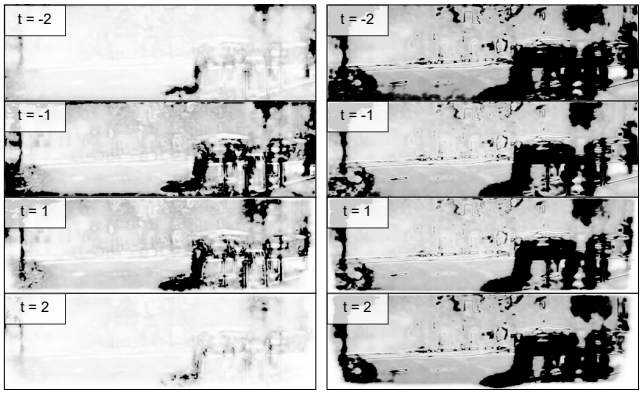
While the projection losses can train a basic MN, the model is prone to errors on occluded pixels, which is a common issue in complex driving scenes. In this section, we elaborate on the problem and show how Per-Pixel-Minimum (PPM) [2], originally designed to improve Depth-Net performance, excludes occluded pixels. Then, we introduce Soft-PPM, an approach that builds on PPM's strengths in pixel filtering but overcomes its shortcomings when applied to our framework.

Within the scope of this paper, a pixel is considered occluded if it is visible in the target frame but hidden in any of the reference frames. Fig. 3 illustrates how occlusion occurs and how PPM handles occlusion in principle. The application of photometric losses hinges on the assumption of appearance consistency. That is, the same object should have a consistent appearance across all frames. While this assumption holds in most cases, occlusion breaks this assumption because the correct match of the occluded pixel in that frame cannot be found. Even when warped with the ground truth flow, it is still impossible to restore the appearance of those pixels. This leads to a failure of the photometric loss for that particular pixel. However, the appearance consistency of those pixels can still hold in other frames. In [2], the authors argued that pixels with smaller training losses are more likely to be non-occluded pixels. Hence, we can utilize the photometric loss to determine which pixels are non-occluded and more reliable for MN training. This method is based on the assumption that the cases where pixels are visible only in the target frame but occluded in all other frames are rare [2]. Under this assumption, the minimum loss value in all frames is likely to be the loss when that pixel is not occluded. Choosing the minimum loss instead of averaging losses in all frames can dramatically reduce the interference of occlusion.

We observe that applying PPM directly to our framework

causes the frames to contribute unequally to training loss (Fig. 4(b)) due to a phenomenon we call "gradient unbalanced problem". PPM preferentially selects pixels with minimal loss, which are often pixels in temporally close frames $\{I_{-1}, I_{+1}\}$ with low inter-frame motion due to their high correspondence to pixels in the target frame. Training a model mainly on these pixels from temporally nearer frames makes it insensitive to moving objects in the farther frames $\{I_{-2}, I_{+2}\}$. Our experiment found that only 35% of the loss propagates to farther frames. This problem is exacerbated with the addition of more reference frames, and results in the MN predicting most pixels in far-frames as static because of the small bias described in Section III-A.

(a) Target frame t: A cyclist on road.



(b) Motion masks by PPM (c) Motion masks by Soft-PPM

Fig. 4: The gradient unbalanced problem. (b) Motion masks produced by MN with PPM. The MN's training loss is dominated by the temporally nearer frames. In contrast, (c) illustrates that our Soft-PPM allows more equal training in all frames.

To address this issue, we propose a method to balance the losses temporally. We generalize the concept of PPM and apply a weighted sum over the temporal axis to select the minimum softly. We first obtain the photometric loss for all pixels in all frames, including occluded pixels. We then assign smaller weights to pixels with larger photometric losses, as they are more likely to be occluded. Similarly to the soft-max function used in classification, we determine the weights and select the minimum as follows:

$$p_t = L_t^{pho}(x) \tag{5}$$

$$\sigma_j(\mathbf{p}) = \frac{e^{-p_j}}{\sum_{t \in T} e^{-p_t}} \tag{6}$$

$$\omega_j(\mathbf{p}) = p_j \sigma_j(\mathbf{p}), \tag{7}$$

where p_t is the photometric loss of the pixel at coordinate x at frame t. $L_t^{pho}(x)$ is the rigid loss or the optical loss defined in Eq. 2. $\sigma_j(\mathbf{p})$ is the weight of the pixel at coordinate x in frame j, where $\mathbf{p} = [p_{-2}, p_{-1}, p_1, p_2] \in \mathcal{R}^4$ contains the photometric loss of the 4 frames. The proposed Soft-PPM $\omega_j(\mathbf{p})$ is the weighted sum of the loss in the four frames.

The weight of a pixel in frame j, $\sigma_j(\mathbf{p})$, decreases exponentially as the photometric loss of the pixel increases, and the sum of the weights is normalized to 1. This design prevents an occluded pixel from inducing an abnormally large photometric loss and dominating the training, because the same pixel, if not occluded, might have a small loss in other frames. This results in a high denominator value in Eq. 6 and dilutes the impact of the occluded pixel by setting a small weight $\sigma_j(\mathbf{p})$. For moving pixels, the losses of that pixel in all frames will be large. In such cases, Soft-PPM acts similarly to averaging. In summary, Soft-PPM selectively reduces the weights of occluded pixels in the loss function while having a small impact on moving pixels. Fig. 5 illustrates the effect of Soft-PPM.

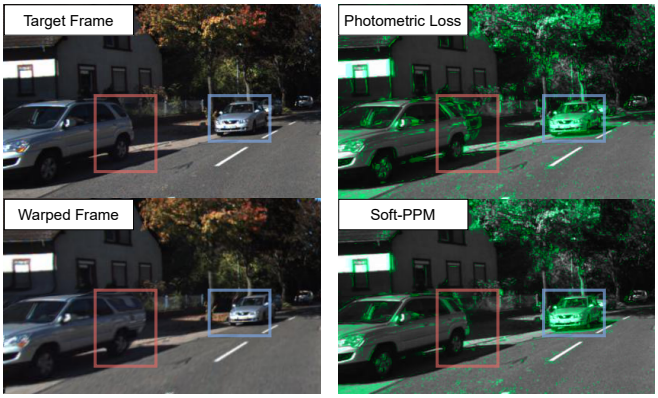


Fig. 5: In this scene, there are a parked car on the left and a moving car on the right. The warped frame is affected by occlusion. The pixels in green have larger photometric losses or Soft-PPM losses. In the image of photometric loss (used in CC), both occluded regions and moving objects have large losses. Our Soft-PPM can selectively decrease the losses for occluded pixels (in the red box) while keeping the loss values on moving objects very close to the photometric loss.

C. Conservative Flow Difference Mask (CFD-Mask)

The proposed CFD-Mask is a variant of the FD-Mask that utilizes information from Depth-Net, Pose-Net, and Flow-Net to train and refine the MN. It acts both as a loss for the MN during training and as a post-process for refining the MN's output during inference. When a pixel is static, the values of both F^{rig} and F^{opt} for that pixel should be the same. A motion mask can be generated using the difference between two flows. However, since the predictions of sub-networks may be incorrect, the original FD mask that uses a simple threshold to separate moving and static regions usually results in many incorrect predictions.

To refine the training loss, we utilize the difference between optical and rigid flows as a measure of the confidence of a pixel's motion status. A large difference implies a high likelihood that the pixel is truly moving, rather than due to a flow prediction error. We set a larger threshold for the moving mask and a smaller threshold for the static mask. With these two thresholds, we can identify pixels with a clearer moving state and rely more on these pixels in both the training and inference stages. Specifically,

$$\Psi_t^d(x) = \begin{cases} 1, & \text{if } \frac{\left\| F_t^{opt}(x) - F_t^{rig}(x) \right\|_2}{\left\| F_t^{rig}(x) \right\|_2 + \gamma_m} > \alpha \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\Psi_t^s(x) = \begin{cases} 1, & \text{if } \frac{\left\|F_t^{opt}(x) - F_t^{rig}(x)\right\|_2}{\left\|F_t^{rig}(x)\right\|_2 + \gamma_s} < \beta \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$\Psi_t^u = XOR(\Psi_t^d, \Psi_t^s) \quad (10)$$

Ψ_t^d, Ψ_t^s, and Ψ_t^u are binary masks of dynamic (moving), static, and unknown pixels in frame t, respectively. $F_t^{opt}(x)$ and $F_t^{rig}(x)$ denote the optical and rigid flows of the pixel at the coordinate x, respextively. Euclidean distance is used to measure the flow differences because it is superior to the inner product [3]. Two thresholds, $\alpha > \beta$, are used to classify pixels as moving, static, and unknown. Only pixels with normalized flow differences that exceed the upper threshold α or below the lower threshold β are used in computing the flow difference loss.

γ_m and γ_s are biases to avoid the masks from becoming oversensitive to small $\left\|F_t^{rig}(x)\right\|_2$. A false static pixel in the static mask harms performance more than a pixel marked as unknown, so γ_s should be much smaller than γ_m because a larger denominator would lead the model to falsely identify pixels as static. We can set a larger value for γ_m in the motion mask because false static in the moving mask does not affect the MN. Minimizing false moving in the moving mask is more crucial; hence we assign it a higher bias. We set γ_s as 10% of the value of γ_m. Our experiments show that minor deviations from this value do not drastically affect the results. α, β, and γ_m are empirically determined using a small dataset of 35 images we collected, in which moving vehicles are labeled.

(a) Target frame

(b) $\gamma_m = 0$ (c) $\gamma_m = 5$

Fig. 6: Our method enhances the sensitivity of the FD-Mask in distant regions. This approach may generate many false moving predictions in pixels that are too far away. A gamma value in our CFD-Mask is used to eliminate these false moving predictions.

Design rationales. In Eq. 8 and Eq. 9, we use the ratio of the flow difference to the rigid flow as the metric. This decision is motivated by the observation that pixels located farther from the camera often have a smaller motion in the camera view. We can enhance the sensitivity of the mask for these distant pixels. With this step, we ensure that the method's sensitivity to moving objects is more consistent across the entire image. We use rigid flow rather than optical flow as the denominator because optical flow is influenced by the object's motion status. Different motion directions of an object can produce varying optical flow, complicating the problem. For example, in scenarios where a vehicle is approaching, the optical flow predictions for the vehicle are large. It makes the FD-Mask insensitive to this kind of motion. Dividing by the length of optical flow would not be effective.

(a) Target frame

(b) Single threshold (c) Conservative Mask

Fig. 7: CFD-Mask. (a) Target frame. (b) The resulting motion segmentation using a single threshold. The black area indicates where the pixels are moving. There are some incorrect predictions in this figure because the flow estimate could be inaccurate. (c) Our conservative mask takes into account the inaccurate estimate of the flow by denoting the pixels that are more difficult to classify as "unknown" (gray area). Additionally, we normalize the flow difference by F^{rig}, which reduces the chance that static pixels closer to the camera are misclassified as moving (false moving).

CFD-Mask Loss. During training, we compute the L1 loss between the CFD-Mask and the motion mask produced by MN. This CFD-Mask loss L_t^m is defined by

$$L_t^m = \sum_{\forall x \in \{x | \Psi_t^d(x) = 1\}} |0 - M_t(x)| + \sum_{\forall x \in \{x | \Psi_t^s(x) = 1\}} |1 - M_t(x)| \quad (11)$$

where M_t is the motion mask generated by MN at frame t and $M_t(x) \in [0, 1]$ denotes the motion state of a pixel at location x with higher value denoting static state. Minimizing this loss trains the MN to produce motion segmentation results close to the FD-Mask.

Motion Segmentation. During testing, the motion segmentation result Ψ_t^{comb} is calculated by

$$\Psi_t^{comb} = \Psi_t \cup \Psi_t^d \cap Not(\Psi_t^s), \quad (12)$$

where Ψ_t integrates the MN's outputs at a set of frames T centered at frame t using

$$\Psi_t = \prod_{t \in T} (1 - M_t) > 0.5 \quad (13)$$

The MN output M_t involves determining whether each pixel moves between the target frame and each reference frame. M_t is the motion mask of the frame t. T is the set of all reference frames. Note that in the context of the joint learning framework, mainstream methods often assign a higher value to "static" and a lower value to "moving" in this mask. The underlying concept of the integration formula can be understood as the "AND" operation applied to the "moving" predictions. Because pixels that meet rigid-body motion are constrained to a lower-dimensional manifold, noise and occlusion typically result in false moving rather than false static. We identify a pixel as moving only when it is determined to be moving in all frames. This operation further excludes false moving predictions. Similarly to our Soft-PPM loss, the combination also requires the object's motion to not change rapidly.

IV. Experiment

Our motion segmentation method is compared with the SOTA self-supervised joint learning frameworks on the

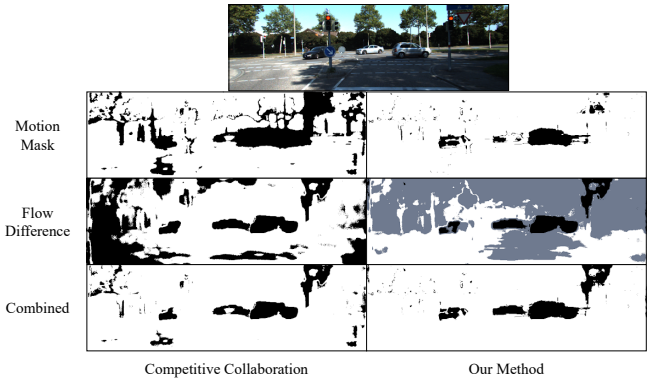


Fig. 8: Motion segmentation results of CC's and our method. The 2nd and 3rd rows compare the effects of FD-mask and CFD mask. Our approach ignores the unknown pixels and considers each pixel included in the moving or static mask as moving or static when generating the combined mask.

KITTI dataset. We also validate the effectiveness of Soft-PPM and CFD-Mask using an ablation study.

A. Dataset

The training set is derived from KITTI raw data [19] and follows the existing standard training protocol [3], [10], [20]. Static frames are removed, yielding 39,810 monocular sequences for training and 4,424 for validation. We use the KITTI 2015 Scene Flow dataset [21] and its motion segmentation ground truth for evaluation. This dataset comprises 200 training scenes and 200 testing scenes. We evaluated our model using 200 scenes from the training set that contain motion segmentation labels. Data in those scenes are excluded by the training protocol. Each scene is a 21-frame sequence with labels in the 11th frame. In contrast to KITTI 2012, which only contains camera movement with static scenes, the KITTI Scene Flow 2015 benchmark specifically focuses on dynamic scenes.

B. Implementation Details

The input data is a 5-frame sequence, including a target frame, four reference frames, and their corresponding camera intrinsics. All images are resized to 256 × 832. To enhance the robustness of the model, we apply data augmentation techniques, including random horizontal flips, color jitter, and random Gaussian noise with a variance of 0.1.

To prevent flow predictions from getting stuck at local minima, the photometric loss and flow difference loss are computed on a 6-layer image pyramid. Smoothness loss is applied to enable the derivation of gradients from larger regions. However, these steps may also amplify the edge regions affected by errors.

C. Training

Similarly to CC, we employed the EM strategy for training. We used pre-trained weights from CC for the Flow-net and only retrained the other three sub-networks. This is because that unlike Depth-Net and Pose-Net, Flow-Net is not limited to rigid body projection. However, in the early iterations, the photometric loss from a pre-trained Flow-Net tends to be significantly smaller than that in the non-trained Depth-Net and Pose-Net. This imbalance can influence the MN to predict all pixels as moving, and the Depth-Net and Pose-Net would be unable to obtain sufficient training because their gradients are filtered out by MN. To address this issue, we applied Z-score standardization to the photometric loss to balance the losses between rigid flow and optical flow.

Soft-PPM can reduce the weights of the losses on the occluded pixels, but we do not want to excessively decrease their weights. For example, the original PPM [2] assigns a weight of 0 to losses with a value other than the minimum, leading to a gradient unbalanced problem where frames other than the minimum value are not effectively trained. If the output of Soft-PPM decreases while its input increases, this adjustment of loss violates the basic idea of competitive training. In a more generalized sense, we aim for the Soft-PPM to be strictly increasing. It can be proved that the Soft-PPM is strictly increasing if its input values are smaller than 1. To ensure the efficiency of Soft-PPM, we scaled the standardized loss to have a mean of 0.5 and a variance of 0.2 and clamped all scaled values into the range $[0, 1]$.

D. Comparison with Other Methods

Table I compares the results of our method and other SOTA methods. We resize all testing images to the same size as the training images. The output mask is then resized to the original resolution to calculate the intersection-over-unit (IoU) score. Our objective is not to classify different semantic classes, but rather to mitigate the impact of occlusion and errors in motion segmentation estimation, with a specific focus on car pixels. Additionally, to ensure robustness and adaptability across various scenarios and to minimize false positives and false negatives in segmentation results, we concentrate on both static and moving cars.

Method	Moving Car	Static Car	Overall
CC [10]	0.5811	0.5577	0.5694
EffiScene [3]	-	-	0.6150
DS [22]	**0.6689**	0.5842	0.6266
RM-Depth [16]	0.6204	**0.6691**	0.6448
Ours	0.6682	0.6640	**0.6661**

TABLE I: Motion Segmentation Comparisons: Our IoU scores on moving and static cars are comparable to the SOTA methods'. Overall, our IoU is at least 2% higher. The IoU scores are retrieved from [3], [10], [16], [22].

E. Ablation Study

We validate the effectiveness of CFD-mask and Soft-PPM in the ablation study by treating CC as a baseline. We refer to the CC model that uses PPM loss but without FD masks as *CC Bare*, and our corresponding configuration (without any FD-masks but using Soft-PPM loss) is called *Soft-PPM*. That is, neither *CC bare* nor *Soft-PPM* uses flow difference loss in training. For flow difference masks, there are two settings: FD-mask and CFD-mask. FD-mask uses a single threshold and was originally adopted in CC. CFD-mask uses two thresholds and does not make judgments in uncertain regions.

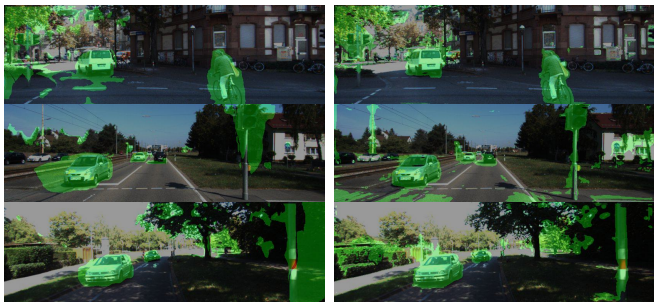
The results of different settings are listed in Table II. By comparing *CC Bare* and *Soft-PPM*, Soft-PPM significantly improves the IoU on static cars by about 23%. Although Soft-PPM slightly reduces the model's sensitivity to moving cars, the overall IoU is still improved by 11%. By comparing *CC Bare+FD-mask* and *CC Bare+CFD-mask*, CFD-Mask improves IoU by 8.8% for moving cars and 10% for static cars. Compared to *Soft-PPM+FD-mask*, Soft-PPM with CFD-mask increases IoU by 17% for moving cars and 7% for static cars. These comparisons validate the effectiveness of CFD-mask. Overall, our method (*Soft-PPM+CFD-mask*) improves the IoU by 10% compared to CC (*CC Bare+FD-mask*).

Method	Moving Car	Static Car	Overall
CC Bare	0.5271	0.3056	0.4164
CC Bare + FD-mask [10]	0.5811	0.5577	0.5694
CC Bare + CFD-mask	**0.6699**	0.6574	0.6637
Soft-PPM	0.5174	0.5323	0.5248
Soft-PPM + FD-mask	0.4962	0.5916	0.5439
Soft-PPM + CFD-mask	0.6682	**0.6640**	**0.6661**

TABLE II: Ablation study on Soft-PPM and flow different masks.

F. Discussion

The KITTI 2015 benchmark only computes IoU scores for car pixels; however, Soft-PPM is proposed to handle occluded background pixels, which are not considered in the evaluation. Fig. 9 illustrates an example in which our method effectively addresses the occlusion issue and reduces false-moving predictions, especially in the background. The performance of our method could be much better than that of other methods if more object classes and background pixels were considered in the evaluation metric.



(a) Baseline (CC) (b) Ours

Fig. 9: Our method reduces a large amount of false moving predictions and produces more accurate motion segmentation that closely aligns with object boundaries.

Limitations. Soft-PPM is designed based on photometric loss and thus inherits several of its problems except for occlusion. It may not be able to handle challenges such as changes in lighting, different weather conditions, or the presence of transparent or reflective objects. Another limitation of Soft-PPM is its assumption that objects' motions should be stable. In the loss calculations, we integrate information from both preceding and succeeding frames to identify pixels that are not occluded. This process assumes a constant motion state of the object. Finally, Soft-PPM relies on information from future frames, which unavoidably introduces some latency.

V. Conclusion

We introduced two loss functions, Soft PPM and CFD-mask, which aim to refine the training signal and mitigate the influence of occlusion on motion segmentation. Soft-PPM utilizes photometric loss as a hint to identify and exclude pixels that are likely to be occluded, resulting in an overall 10% improvement in IoU compared to CC Bare. CFD-mask affects the MN only when a clear motion status is presented, which was validated with a comparison to the FD-mask. Our approach shows the efficacy of training the MN exclusively with high-reliability pixel data, resulting in a 10% improvement in overall IoU score compared to our baseline CC. Our methods are independent of the model architecture and can be applied to different MN architectures when having access to both past and future information.

References

[1] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.

[2] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019.

[3] Y. Jiao, T. D. Tran, and G. Shi, "Effiscene: Efficient per-pixel rigidity inference for unsupervised joint learning of optical flow, depth, camera pose and motion segmentation," in *CVPR*, 2021, pp. 5538–5547.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] S.-C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *VCIP*, 2004, pp. 881–892.

[6] S. Shen, Y. Cai, W. Wang, and S. Scherer, "DytanVO: Joint refinement of visual odometry and motion segmentation in dynamic environments," in *ICRA*, 2023, pp. 4048–4055.

[7] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *ICCV*, 2017, pp. 4481–4490.

[8] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto, "Unsupervised moving object detection via contextual information separation," in *CVPR*, 2019, pp. 879–888.

[9] L. Lian, Z. Wu, and S. X. Yu, "Bootstrapping objectness from videos by relaxed common fate and visual grouping," in *CVPR*, 2023.

[10] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *CVPR*, 2019, pp. 12 240–12 249.

[11] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *CVPR*, 2015.

[12] J. Mattheus, H. Grobler, and A. M. Abu-Mahfouz, "A review of motion segmentation: Approaches and major challenges," in *IMITEC*, 2020.

[13] L. Zappella, X. Lladó, and J. Salvi, "Motion segmentation: A review," *Artificial Intelligence Research and Development*, pp. 398–407, 2008.

[14] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "SegFlow: Joint learning for video object segmentation and optical flow," in *ICCV'17*.

[15] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting anything that moves," in *ICCVW*, 2019.

[16] T.-W. Hui, "RM-Depth: Unsupervised learning of recurrent monocular depth in dynamic scenes," in *CVPR*, 2022, pp. 1675–1684.

[17] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *ECCV*, 2018.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, 2013.

[20] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014.

[21] M. Menze, C. Heipke, and A. Geiger, "Joint 3D estimation of vehicles and scene flow," in *ISPRS Geospatial Week*, 2015.

[22] F. Tosi, F. Aleotti, P. Z. Ramirez, M. Poggi, S. Salti, L. D. Stefano, and S. Mattoccia, "Distilled semantics for comprehensive scene understanding from videos," in *CVPR*, 2020, pp. 4654–4665.