Reciprocal Transformations for Unsupervised Video Object Segmentation

Sucheng Ren\textsuperscript{1}, Wenxi Liu\textsuperscript{2}, Yong tuo Liu\textsuperscript{1}, Haoxin Chen\textsuperscript{1}, Guoqiang Han\textsuperscript{1}, Shengfeng He\textsuperscript{1,}\textsuperscript{*}

\textsuperscript{1} School of Computer Science and Engineering, South China University of Technology
\textsuperscript{2} College of Mathematics and Computer Science, Fuzhou University

Abstract

Unsupervised video object segmentation (UVOS) aims at segmenting the primary objects in videos without any human intervention. Due to the lack of prior knowledge about the primary objects, identifying them from videos is the major challenge of UVOS. Previous methods often regard the moving objects as primary ones and rely on optical flow to capture the motion cues in videos, but the flow information alone is insufficient to distinguish the primary objects from the background objects that move together. This is because, when the noisy motion features are combined with the appearance features, the localization of the primary objects is misguided. To address this problem, we propose a novel reciprocal transformation network to discover primary objects by correlating three key factors: the intra-frame contrast, the motion cues, and temporal coherence of recurring objects. Each corresponds to a representative type of primary object, and our reciprocal mechanism enables an organic coordination of them to effectively remove ambiguous distractions from videos. Additionally, to exclude the information of the moving background objects from motion features, our transformation module enables to reciprocally transform the appearance features to enhance the motion features, so as to focus on the moving objects with salient appearance while removing the co-moving outliers. Experiments on the public benchmarks demonstrate that our model significantly outperforms the state-of-the-art methods. Code is available at \url{https://github.com/OliverRensu/RTNet}.

1. Introduction

Video object segmentation (VOS) aims at localizing and segmenting objects in videos. As one of the fundamental tasks in computer vision, VOS has many applications, \textit{e.g.}, object tracking \cite{22,30,51} autonomous driving \cite{5,13}, video surveillance \cite{45}. In specific, the existing techniques of VOS can be roughly categorized into: semi-supervised video object segmentation \cite{14,32,56} in which the segmentation mask of the primary object(s) is given at the first frame, and \textit{unsupervised video object segmentation} (UVOS) \cite{31,47,48,60} that aims to extract the mask of the primary object(s) without any prior knowledge. In this paper, we focus on the task of UVOS.

Due to the lack of prior knowledge, the UVOS models have to handle the major concern for identifying the sources of primary objects in the videos. We observe that there are three types of candidate primary objects: the salient objects in a single frame, the moving objects, and recurring objects in the video. In general, human attention will be drawn on the salient objects \cite{18,28,52} within an image, thus these visually distinct objects may be the candidate primary objects. However, these methods may not be applicable for identifying primary objects in videos, as human attention will naturally shift to various patterns of dynamics or motions in video \cite{9,37}. Thus, the objects that are indistinct in a single frame but moving in video may be the primary objects according to motion cues, yet ignored by the image-based models. Besides, people also tend to memorize the objects that appear repeatedly in the video, therefore these objects may be treated as another type of primary ones.

Previous methods \cite{24,65} apply optical flow to capture motion information. However, the optical flow can hardly
distinguish the dynamic background objects from the foreground objects. For instance, in Fig. 1, it is ambiguous to categorize the car and the billboard, or the human and the spray, into foreground and background by optical flow only. Therefore, directly mapping these motion cues to the appearance features [24, 65] may misguide UVOS models when localizing the primary objects.

To address the aforementioned limitation, we propose a unified framework, Reciprocal Transformation Network (RTNet), to identify primary objects beyond the distraction of co-mov ing outliers. Our idea is to mutually evolve and integrate the appearance and motion representations in the network such that all three types of candidate primary objects can be taken into consideration and produces a holistic decision. To this end, we propose a Reciprocal Transformation Module (RTM) within the network to enable in-domain and cross-domain feature interactions. In particular, the proposed reciprocal transformation scheme computes similarities for all the pairwise features including motion-motion, appearance-appearance, and appearance-motion pairs of features. The underlying information will be transformed to each other in order to replenish the appearance/motion object representation and remove the ambiguity from the inconsistent appearance or the inaccurate optical flow.

Applying the proposed RTM on different source features results in different types of primary object properties, i.e., 1) self-similarities of appearance and motion features lead to intra-frame contrast; 2) appearance-motion similarity produces motion cues; and 3) cross-frame appearance-appearance and motion-motion feature similarities yield temporal coherence. Each corresponds to one of the three types of primary objects. Besides, instead of simply skip connecting the encoder and decoder as FCN [29] does, we propose a Spatial Temporal Attentive Fusion Module (STAFM) to leverage the appearance and motion features from the corresponding encoder stage, and segment spatio-temporally consistent primary objects. In experiments, we evaluate our approach against the state-of-the-art methods on public benchmarks DAVIS [34] and achieve the performance gain of 4% on region similarity $\mathcal{J}$ and 5% on boundary accuracy $\mathcal{F}$ over the second best method [65].

To sum up, the contributions of our paper are three-fold:

- We delve into three types of primary objects in videos, and present a novel reciprocal transformation network (RTNet), which is able to effectively exploit the intra-frame contrast, motion cues, and temporal coherence of recurring objects to identify and segment primary objects from the videos.

- To eliminate the co-occurring moving outliers from the optical flow and extract the moving objects with salient appearance, we propose a new reciprocal transformation approach that mutually evolves the appearance features to the motion features.

- We propose a Spatial Temporal Attentive Fusion Module (STAFM) to selectively integrate the appearance and motion features.

- Our method significantly outperforms state-of-the-art methods in the public benchmark. Even if we use a much smaller backbone and less training data, our lightweight model can still achieve comparable performance against latest competitors.

2. Related Work

In this section, we will survey the works on video object segmentation as well as attention mechanism.

2.1. Video Object Segmentation

Prior methods on video object segmentation can be divided into two categories: semi-supervised video object segmentation and unsupervised video object segmentation.

Semi-supervised Video Object Segmentation. The semi-supervised VOS methods assume that the ground-truth mask of the target object(s) is provided at the first frame. These methods can be further categorized into two types, i.e., online-learning methods [1,4,7,26] and offline-learning methods [6,20]. Online-learning methods fine-tune the pre-trained model based on the given ground-truth mask and predict the segmentation results at the cost of the inference time. On the contrary, offline-learning methods utilize the given mask as the guidance to update the pretrained model at the inference time. Despite the superior performance of semi-supervised VOS methods, the annotating ground-truth masks involves manual efforts and may introduce bias, which limits its application in real-world scenarios.

Unsupervised Video Object Segmentation. Compared with semi-supervised VOS methods, UVOS methods do not require any manual annotations. Early UVOS methods are mainly based on object proposal [21,23], temporal trajectory [3,11,33] and saliency prior [19,53,54]. With the development of deep convolutional neural networks and the establishment of large-scale datasets [34], deep learning based methods are proposed for modeling the spatio-temporal information. To capture the motion cues, MPNet [46] focuses on optical flow only, but it is difficult to segment the static objects, due to the insufficient appearance information. In addition, several methods apply two stream networks to capture and then fuse the appearance and motion features. Fragkiadaki et al. [10] use fully-connected layers to integrate optical flow and static boundaries to rank the segment proposals. MBN [25] includes a bilateral network for background estimation and integrates it with the appearance features into a graph. Besides, to capture long-term temporal information, several methods [43] process
Figure 2: Illustration of Reciprocal Transformation Network which is composed of two network streams: the appearance stream and the motion stream. A pair of frames \( \{ I_{a1}, I_{a2} \} \) and their corresponding optical flows \( \{ I_{m1}, I_{m2} \} \) are fed into two streams, respectively. To correlate the features of the spatial and temporal domains, we propose the Reciprocal Transformation Modules (RTM) to transform the motion features, \( F_{m1}^i \) and \( F_{m2}^i \) \((i = \{1, 2\})\), and the appearance features, \( F_{a1}^i \) and \( F_{a2}^i \). Given pairwise features, RTM enables in-domain and cross-domain feature interactions for identifying salient objects in a single frame, moving objects, and recurring objects, respectively. In the end, the appearance features and motion features are fused by the Spatial Temporal Attentive Fusion Module (STAFM) and fed to the decoder of the appearance stream to generate the final mask.

video in RNN-based architecture. Song et al. [43] introduce a bi-directional ConvLSTM [41] for extracting features for multiple frames. However, RNN-based models suffer from the gradient vanishing problem and they are difficult to run in parallel. Furthermore, the attention-based methods are proposed for capturing long range dependency. COSNet [31] propose a co-attention layer to extract the discriminative foreground in a short video. ADNet [60] introduces an anchor frame to model the long-term dependency. MATNet [65] uses a motion-attentive transition to model motion information and spatio-temporal representation. Different from prior works, our reciprocal transformations leverage long range intra-frame contrast, temporal coherence, and motion-appearance similarity to enhance the appearance feature representation.

**Attention Mechanism.** Attention mechanism has been demonstrated effective and efficiency in many tasks due to its flexibility [27, 40, 49, 63]. The core idea of attention mechanisms is to highlight the task-specific discriminative regions in features. The attention scheme for videos has been explored in many aspects, including gating or pooling [14, 32], pose primitives [2, 15], graph representations [17, 57], recurrent memory models [31, 39], and self-attention [56]. In contrast to previous works, our proposed model measures the similarity for any pair of feature maps of frames and flows, and reciprocally transform the similarity to each other.

### 3. Proposed Method
#### 3.1. System Overview

Given a pair of frames \( \{ I_{a1}, I_{a2} \} \) in a video and their corresponding optical flow \( \{ I_{m1}, I_{m2} \} \) computed by [44], we aims at segmenting the primary objects within \( I_{a1} \) and \( I_{a2} \). Fig. 2 shows the pipeline of our method, which is composed of two main streams: the appearance stream with \( \{ I_{a1}, I_{a2} \} \) as inputs and the motion stream with \( \{ I_{m1}, I_{m2} \} \) as inputs. Each stream is an encoder-decoder architecture with skip connections [38]. In specific, we adopt ResNet [16] with dilated convolution [61] as backbone.

In each stage of the encoder, we introduce a Reciprocal Transformation Module (RTM), \( F_{RTM} \). As the main component of our framework, RTM is consisted of three sub-modules: reciprocal scaling, reciprocal transformation, and reciprocal gating. We will elaborate it in Sec. 3.2. Hence, we leverage RTM to enhance the pairwise appearance and
motion features for identifying three types of candidate primary objects, i.e., salient objects within a frame, moving objects, and recurring objects in the video, which will be described in Sec. 3.3.

Last, in the decoder network of the motion stream, the skip connections bridge the low-level encoded features with the corresponding decoding layer, while the decoder of the appearance stream leverages the Spatial Temporal Attentive Fusion Module (STAFM) (Sec. 3.4) to fuse appearance and motion features to yield the final results.

3.2. Reciprocal Transformation Module

In this section, we will describe the structure of Reciprocal Transformation Module (RTM) (see Fig. 3). Given two features (denoted as $F_a$ and $F_b$ which could be any pair from the appearance or motion features), we can mutually evolve and integrate the pairwise features via RTM. As described in the following subsections, our RTM includes three sub-modules: reciprocal scaling for adjusting the weights of different semantics, reciprocal transformation for measuring the similarity between feature maps, and reciprocal gating for balancing the transformed features.

3.2.1 Reciprocal Scaling

In deep neural networks, the channels of features represent different semantic meanings [12]. Thus, the values of channels can be accordingly scaled so as to bridge the attention of different primary objects for the features of different sources. In particular, as the motion features mainly focus on moving objects while the appearance features focus on salient objects, the reciprocal scaling enables to bridge their attention gap. Similarly, for the appearance features of two different frames, the reciprocal scaling may enable them to cast attention on different objects. To do so, we squeeze the combined feature maps to generate the representative value $A_c$ for each channel and estimate the scaling factors $w_1$ and $w_2$ that indicate the importance of semantics:

$$A_c = \mathcal{F}_c(F_a; F_b),$$

$$w_1 = \sigma(FC(\phi(FC(A_c; \theta_1); \theta_21))),$$

$$w_2 = \sigma(FC(\phi(FC(A_c; \theta_1); \theta_22))),$$

where $\mathcal{F}_c$ refers to the concatenation and squeezing (i.e., global average pooling) operation. $FC(\cdot; \theta)$ refers to the fully connected layer and $\phi$ is a ReLU activation function.

Thus, we scale the channels of the features $F_a$ and $F_b$ according to $w_1$ and $w_2$:

$$F'_a = w_1 \odot F_a + F_a,$$

$$F'_b = w_2 \odot F_b + F_b,$$

where $\odot$ is the element-wise multiplication.

3.2.2 Reciprocal Transformation

In this sub-module, we aim at measuring the similarity between feature maps, and transform both features to enhance the representation. Before explaining our reciprocal transformation, we introduce the principle of the vanilla self-transformation and nonreciprocal transformation.

**Vanilla self-transformation.** The local receptive fields of standard fully convolution network (FCN) based methods [29] confines the capability of networks in segmentation tasks that require rich context information. To capture long range dependency, Wang et al. [56] propose the self-attention scheme in which the feature maps are applied to measure the holistic similarity with itself to estimate its attention.

**Nonreciprocal transformation.** Nonreciprocal transformation is designed to compute the holistic similarity across different feature maps and manages to deliver the information from one feature to another. The nonreciprocal transformation is computed based on the similarity of the features from different sources. Given the feature maps for query $F'_a \in \mathcal{R}^{c \times w \times h}$ and the target feature map $F'_b \in \mathcal{R}^{c \times w \times h}$, it measures the holistic positional similarity matrix $S$ using a non-local network structure:

$$S = Softmax(F'_b^TWF'_a),$$

where $W \in \mathcal{R}^{c \times c}$ is the similarity matrix. In practice, $W$ contains a large amount of trainable parameters. To reduce the network complexity, we approximate $W$ via two separate convolution operations:

$$\hat{F}'_a = Conv(F'_a; \theta_a),$$

$$\hat{F}'_b = Conv(F'_b; \theta_b),$$

$$S_{a \rightarrow b} = Softmax(\hat{F}'_b^T \times \hat{F}'_a),$$

Figure 3: Our Reciprocal Transformation Module (RTM) includes reciprocal scaling, reciprocal transformation and reciprocal gating. $\odot$ and $\otimes$ indicate element-wise multiplication and matrix multiplication respectively.
where $\text{Conv}(F; \theta)$ is the convolution layer with the parameters $\theta$ on the feature map $F$ and $\times$ refer to the matrix multiplication operator. Then, we derive the features according to the holistic positional similarity matrix $S$:

$$F''_{a \rightarrow b} = \text{Conv}(F'_{a}, \theta_{x_a}) \times S_{a \rightarrow b},$$

(5)

where $F''_{a \rightarrow b}$ is the enhanced features of $F'_b$ from interacting with the feature $F'_a$.

**Reciprocal Transformation.** Unlike nonreciprocal transformation, our reciprocal transformation use two features to mutually compensate each other, i.e.,

$$F''_{a \rightarrow b} = \text{Conv}(F'_{a}, \theta_{x_a}) \times S_{a \rightarrow b},$$

$$F''_{b \rightarrow a} = \text{Conv}(F'_{b}, \theta_{x_b}) \times S_{b \rightarrow a}. \tag{6}$$

Using reciprocal transformation, the motion cues can be strengthened across the features of different sources. In specific, we can first transform the motion features to the appearance features to improve the segmentation ability for moving objects, and then the appearance features are reciprocally transformed to the motion features to eliminate the co-moving outliers.

### 3.2.3 Reciprocal Gating

Transformed features often have the different extents of importance. For instance, the appearance or motion noise like background variation or motion blur should be regarded as distractors with less importance. Therefore, we design a reciprocal gating mechanism to balance different transformed features:

$$G_a = \sigma(\text{Conv}(F''_{a \rightarrow b} \oplus F''_{b \rightarrow a}; \theta_1)),$$

$$G_b = \sigma(\text{Conv}(F''_{a \rightarrow b} \oplus F''_{b \rightarrow a}; \theta_2)), \tag{7}$$

where $\oplus$ is the concatenation operation and $\sigma$ is the sigmoid function. $G_a, G_b \in (0, 1)$ are the reciprocal gates to balance the transformed features. Thus, we apply these gates to the original features:

$$X_{a \rightarrow b} = G_b \odot F''_{a \rightarrow b} + F''_{a \rightarrow b},$$

$$X_{b \rightarrow a} = G_a \odot F''_{b \rightarrow a} + F''_{b \rightarrow a}, \tag{8}$$

where $X_{a \rightarrow b}, X_{b \rightarrow a}$ are the final feature maps.

### 3.3. RTM-based Video Object Segmentation

Depending on the input features, RTM is enabling to interact and enhance the features of different sources for identifying the salient objects in a single frame, the moving objects, and the recurring objects, respectively. In general, on the $i^{th}$ stage of the encoder network, we obtain the appearance features $F_i^{a_1}$ and $F_i^{a_2}$ from the input frames $\{I_{a_1}, I_{a_2}\}$, as well as the motion features $F_i^{m_1}$ and $F_i^{m_2}$ from the corresponding optical flows $\{I_{m_1}, I_{m_2}\}$.

**Salient objects.** To identify the salient objects in a single frame, RTM is utilized to obtain the intra-frame contrast by measuring the self-similarity of the appearance features or the motion features, which can be expressed as below:

$$X_{a_1 \rightarrow a_1} = F_{RTM}(F_i^{a_1}, F_i^{a_1}; \theta_a),$$

$$X_{a_2 \rightarrow a_2} = F_{RTM}(F_i^{a_2}, F_i^{a_2}; \theta_a),$$

$$X_{m_1 \rightarrow m_1} = F_{RTM}(F_i^{m_1}, F_i^{m_1}; \theta_m),$$

$$X_{m_2 \rightarrow m_2} = F_{RTM}(F_i^{m_2}, F_i^{m_2}; \theta_m), \tag{9}$$

where $X_{a_1 \rightarrow a_1}$ and $X_{a_2 \rightarrow a_2}$ represent the self-similarity of the appearance features $F_i^{a_1}$ and $F_i^{a_2}$, $X_{m_1 \rightarrow m_1}$ and $X_{m_2 \rightarrow m_2}$ are the self-similarity of the motion features $F_i^{m_1}$ and $F_i^{m_2}$, $\theta_a, \theta_m$ are the parameters of the corresponding RTMs.

**Recurring objects.** To identify the recurring objects, the spatio-temporal correlation between the input frames will be measured, so as to capture the long range dependency in two separate frames. Thus, we have:

$$X_{a_1 \rightarrow a_2}, X_{a_2 \rightarrow a_1} = F_{RTM}(F_i^{a_1}, F_i^{a_2}; \theta_{aa}),$$

$$X_{m_1 \rightarrow m_2}, X_{m_2 \rightarrow m_1} = F_{RTM}(F_i^{m_1}, F_i^{m_2}; \theta_{mm}), \tag{10}$$

where $X_{a_1 \rightarrow a_2}$ and $X_{a_2 \rightarrow a_1}$ are the similarity between the appearance features of two frames, $F_i^{a_1}$ and $F_i^{a_2}$, $X_{m_1 \rightarrow m_1}$ and $X_{m_2 \rightarrow m_2}$ refer to the similarity between the motion features of two frames, $F_i^{m_1}$ and $F_i^{m_2}$.

**Moving objects.** We associate the motion features with the appearance features for identifying the moving objects, by computing the similarity of salient appearance and motion cues. Likewise, by associating the motion features with the appearance features, we can eliminate the co-moving outliers:

$$X_{m_1 \rightarrow a_1}, X_{a_1 \rightarrow m_1} = F_{RTM}(F_i^{a_1}, F_i^{m_1}; \theta_{am}),$$

$$X_{m_2 \rightarrow a_2}, X_{a_2 \rightarrow m_2} = F_{RTM}(F_i^{a_2}, F_i^{m_2}; \theta_{am}), \tag{11}$$

where $X_{m_1 \rightarrow a_1}$ and $X_{a_1 \rightarrow m_1}$ are the similarity between the motion and appearance features, $F_i^{m_1}$ and $F_i^{a_1}$, $X_{m_2 \rightarrow a_2}$ and $X_{a_2 \rightarrow m_2}$ refer to the similarity between the motion and appearance features, $F_i^{m_2}$ and $F_i^{a_2}$.

Hence, the final appearance features are the combination of the intra-frame contrast, temporal coherence, and motion cues, as below:

$$X_{a_1} = F_{a_1} + X_{a_1 \rightarrow a_1} + X_{a_2 \rightarrow a_1} + X_{m_1 \rightarrow a_1},$$

$$X_{a_2} = F_{a_2} + X_{a_2 \rightarrow a_2} + X_{a_1 \rightarrow a_2} + X_{m_2 \rightarrow a_2}. \tag{12}$$

Similarly, the motion features are defined as below:

$$X_{m_1} = F_{m_1} + X_{m_1 \rightarrow m_1} + X_{m_2 \rightarrow m_1} + X_{a_1 \rightarrow m_1},$$

$$X_{m_2} = F_{m_2} + X_{m_2 \rightarrow m_2} + X_{m_1 \rightarrow m_2} + X_{a_2 \rightarrow m_2}. \tag{13}$$
3.4. Spatial Temporal Attentive Fusion Module

Inspired by CBAM [59], we aim at selectively fusing appearance features and motion features from each channel stage via channel attention and spatial attention. As illustrated in Fig. 4, with the appearance features $X_a \in \mathbb{R}^{c \times w \times h}$ and motion features $X_m \in \mathbb{R}^{c \times w \times h}$, we design a gating mechanism to adjust the weight of each channel for all the feature maps:

$$A_c = F_c(X_a; X_m),$$
$$w_c = \sigma(FC(\phi(FC(A_c; \theta_1); \theta_2))),$$
$$w^m_c = w_c, w^m_c = 1 - w_c,$$

where $F_c$ is the concatenation and squeezing operation. $FC(X; \theta)$ is a fully connected layer, $\phi$ is the ReLU activation function. The fully connected layers and these activation functions jointly serve as the excitation operation. $\{w_c \in (0, 1) \mid w_c \in \mathbb{R}^c\}$ refers to the channel gate. Thus, we use the channel gate to enhance motion and appearance features as below:

$$X'_a = w^a_c \odot X_a,$$
$$X'_m = w^m_c \odot X_m.$$  \hspace{1cm} (15)

In addition, we exploit the spatial relationship across motion and appearance features to infer the spatial attention, which can guide the adaptive fusion of the motion and appearance features. To do so, we first calculate the spatial features for the appearance features $X'_a$ and motion features $X'_m$ enhanced by channel attention:

$$P_a = \text{MaxPool}(X'_a) \oplus \text{AvgPool}(X'_a),$$
$$P_m = \text{MaxPool}(X'_m) \oplus \text{AvgPool}(X'_m),$$
$$P^{'} = \sigma(\text{Conv}(P_a \oplus P_m; \theta_p)),$$
$$P' = P^{'} + P'_m = 1 - P',$$

where $P_a$ and $P_m$ represent the spatial attention maps of $X'_a$ and $X'_m$. $\text{MaxPool}$ represents the max pooling layer and $\text{AvgPool}$ represents the average pooling layer. Then, we leverage the attention maps to fuse the appearance and motion features, as follows:

$$X''_a = P'_a \odot X'_a + P'_m \odot X'_m,$$  \hspace{1cm} (17)

where $X''_a$ is the fused features which will be passed to the decoder of the appearance network stream.

3.5. Loss Function

We adopt the same loss function as BASNet [36] to jointly measure the prediction in the pixel level by cross entropy loss [8], in the patch level by SSIM loss [58], as well as in the region level by IoU loss [62]:

$$L(M, G) = l_{ce}(M, G) + l_{ssim}(M, G) + l_{iou}(M, G),$$  \hspace{1cm} (18)

where $M$ denotes the segmentation mask and $G$ refers to the ground-truth.

Given an pair of image $I_{a_1}, I_{a_2}$ and their corresponding optical flow $I_{m_1}, I_{m_2}$, we have four output segmentation masks from four stages of the decoder. Hence, the total loss is defined as below:

$$\mathcal{L} = \sum_{i=1}^{2} \sum_{j=1}^{4} L(M^i_{a_i}, G^i_{a_i}) + L(M^i_{m_i}, G^i_{m_i}).$$  \hspace{1cm} (19)

4. Experiment

4.1. Implementation Details and Datasets

Implementation Details. In the following experiments, we test our model with ResNet-34 and ResNet-101 as backbone networks. We pre-train our appearance stream and the motion stream without the transformation and fusion modules on the saliency detection dataset DUTS [50] and video object segmentation dataset DAVIS-16 [34] dataset respectively. Then, we freeze the weights of the backbone networks and train the transformation modules and the feature fusion module on DAVIS-16 dataset. During above training period, the learning rate is set as 1e-3 and we fine-tune the whole model with the learning rate of 5e-4. We apply SGD as the optimizer with momentum of 5e-4. During training, we randomly sample frame pairs in the same video.

We adopt the data argumentation strategies including the vertical/horizontal flip and multi-scale training. On the test time, to produce the segmentation masks of primary objects in the target frame, we randomly select another frame in the same video as reference. Following the practice of [31], we also apply CRF-based post-processing technique.

Datasets. We evaluate our method on two public dataset: DAVIS-16 [34] contains 50 high-quality videos of 480p and 720p with high quality dense pixel-level annotations and YouTube-Objects [35] contains 126 videos over 20,000 frames and 10 semantic categories. We adopt region similarity $\mathcal{J}$ and boundary accuracy $\mathcal{F}$ as evaluation metrics.
### 4.2. Comparison with State-of-the-arts

We compare with the previous methods: LMP [29], LVO [47], PDB [43], LSMO [48], MOTAdapt [42], AGS [55], COSNet [31], ADNet [60], MATNet [65], DFNet [64].

**Evaluation on DAVIS-16.** We evaluate our RTNet with state-of-the-art unsupervised video object segmentation methods. The quantitative results are reported in Table 1. LMP [29] tries to prediction primary objects based on the optical flow only. However, due to the lack of appearance features, this method has the worst performance in localizing and segmenting the primary objects. Some methods [31, 43, 60] rely on the appearance features without the guidance of optical flow also achieves comparable performance, because these methods extract the appearance features in sequences based on the spatial-temporal architectures like ConvLSTM. Among them, COSNet [31] and ADNet [60] leverage the attention mechanism and show strong ability to model global sequence information. AGS [55] takes extra visual attention annotations and is more powerful to locate primary objects. More methods [47, 48, 65] including us take both appearance features and motion features into consideration. With the reciprocal transformation, our method is able to identify all kinds of candidate primary objects. In particular, our lightweight model using ResNet-34 as backbone (“Ours-Light” in Table 1) outperforms the both motion and appearance based method MATNet by 2.91% on $J$ and 3.47% on $F$. MATNet uses ResNet-101 as backbone and around 12k video frames for training, while our lightweight version model adopts ResNet-34 as backbone and around 2k frames for training. Our full model using ResNet-101 (“Ours” in Table 1) shows even better performance. It achieves 0.94% on $J$ over our lightweight weight model and 3.88% on $J$ over the second best model.

![Figure 5: Qualitative results on DAVIS-16: breakdance, horsejump-high and dance-twirl](image)

Table 1: Quantitative results on DAVIS-16. Res-m indicates the number of layers in the backbone. The top three performers are marked in **Red**, **Green**, and **Blue**, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>$J$ Mean</th>
<th>$J$ Recall</th>
<th>$F$ Mean</th>
<th>$F$ Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMP</td>
<td>FLowNet</td>
<td>70.0</td>
<td>85.0</td>
<td>65.9</td>
<td>79.2</td>
</tr>
<tr>
<td>LVO</td>
<td>DeepLab</td>
<td>75.9</td>
<td>89.1</td>
<td>72.1</td>
<td>83.4</td>
</tr>
<tr>
<td>PDB</td>
<td>ResNet-50</td>
<td>77.2</td>
<td>90.1</td>
<td>74.5</td>
<td>84.4</td>
</tr>
<tr>
<td>LMSO</td>
<td>ResNet-101</td>
<td>78.2</td>
<td>89.1</td>
<td>75.9</td>
<td>84.7</td>
</tr>
<tr>
<td>MOTAdapt</td>
<td>ResNet-101</td>
<td>77.2</td>
<td>87.8</td>
<td>77.4</td>
<td>84.4</td>
</tr>
<tr>
<td>AGS</td>
<td>ResNet-101</td>
<td>79.7</td>
<td>91.1</td>
<td>77.4</td>
<td>85.8</td>
</tr>
<tr>
<td>COSNet</td>
<td>ResNet-101</td>
<td>80.5</td>
<td>93.1</td>
<td>79.5</td>
<td>89.5</td>
</tr>
<tr>
<td>ADNet</td>
<td>ResNet-101</td>
<td>81.7</td>
<td>90.9</td>
<td>80.5</td>
<td>85.1</td>
</tr>
<tr>
<td>MATNet</td>
<td>ResNet-101</td>
<td>82.4</td>
<td>94.5</td>
<td>80.7</td>
<td>90.2</td>
</tr>
<tr>
<td>DFNet</td>
<td>DeepLab</td>
<td>83.4</td>
<td>-</td>
<td>81.8</td>
<td>-</td>
</tr>
<tr>
<td>Ours-Light</td>
<td>ResNet-34</td>
<td>84.8</td>
<td>95.8</td>
<td>83.5</td>
<td>93.1</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-101</td>
<td>85.6</td>
<td>96.1</td>
<td>84.7</td>
<td>93.8</td>
</tr>
</tbody>
</table>

Fig. 5 shows our qualitative results on DAVIS-16 which contains challenging scenarios like complex background and motion blur. Our RTNet precisely captures the location of primary objects and segments them with sharp boundaries, thanks to the transformed intra-frame contrast, moving cues and temporal coherence. The effectiveness can be specially observed in breakdance, where there are plenty of appearance-similar humans standing behind the dancing man. Besides, with our multi-stage reciprocal transformation, objects of different scales in horsejump-high can be accurately segmented.

**Evaluation on YouTube-Objects.** We report the performance of our RTNet on YouTube-Objects dataset in Table 2. Our method achieve the best performance over all the comparison methods under the region similarity $J$. For the slow moving objects (**i.e. Airplane and Boat**) with ambiguous background, it is difficult for the optical flow based model (**i.e. MATNet**) to capture the primary objects, while our model fuses temporal coherence and intra-frame contrast information and thus significantly outperform MATNet. For the moving objects with salient appearance (**i.e. Bird and Cat**), motion based method including ours and MATNet outperform the appearance based method (**i.e. COSNet**).
Table 2: Quantitative results of each category on YouTube-Objects dataset over regional similarity (mean $\bar{J}$).

<table>
<thead>
<tr>
<th>Category</th>
<th>LVO</th>
<th>PDB</th>
<th>MATNet AGS</th>
<th>COSNet</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>86.2</td>
<td>78.0</td>
<td>72.9</td>
<td>87.7</td>
<td>81.1</td>
</tr>
<tr>
<td>Bird</td>
<td>81.0</td>
<td>80.0</td>
<td>77.5</td>
<td>76.7</td>
<td>75.7</td>
</tr>
<tr>
<td>Boat</td>
<td>68.5</td>
<td>58.9</td>
<td>66.9</td>
<td>72.2</td>
<td>71.3</td>
</tr>
<tr>
<td>Car</td>
<td>69.3</td>
<td>76.5</td>
<td>79.0</td>
<td>78.6</td>
<td>77.6</td>
</tr>
<tr>
<td>Cat</td>
<td>58.8</td>
<td>63.0</td>
<td>73.7</td>
<td>69.2</td>
<td>66.5</td>
</tr>
<tr>
<td>Cow</td>
<td>68.5</td>
<td>64.1</td>
<td>67.4</td>
<td>64.6</td>
<td>69.8</td>
</tr>
<tr>
<td>Dog</td>
<td>61.7</td>
<td>70.1</td>
<td>75.9</td>
<td>73.3</td>
<td>76.8</td>
</tr>
<tr>
<td>Horse</td>
<td>53.9</td>
<td>67.6</td>
<td>63.2</td>
<td>64.4</td>
<td>67.4</td>
</tr>
<tr>
<td>Motorbike</td>
<td>60.8</td>
<td>58.3</td>
<td>62.6</td>
<td>62.1</td>
<td>67.7</td>
</tr>
<tr>
<td>Train</td>
<td>66.3</td>
<td>35.2</td>
<td>51.0</td>
<td>48.2</td>
<td>46.8</td>
</tr>
</tbody>
</table>

Mean $\bar{J}$ | 67.5 | 65.4 | 69.0 | 69.7 | 70.5 | 71.0 |

Table 3: Ablation study for three types of primary objects. S, M, R indicate intra-frame salient objects, moving objects, and recurring objects, respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean $\bar{J}$</th>
<th>$\Delta \bar{J}$</th>
<th>Mean $\bar{F}$</th>
<th>$\Delta \bar{F}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>77.53</td>
<td>-</td>
<td>76.26</td>
<td>-</td>
</tr>
<tr>
<td>Baseline+S</td>
<td>78.19</td>
<td>0.66</td>
<td>77.30</td>
<td>1.04</td>
</tr>
<tr>
<td>Baseline+M</td>
<td>83.01</td>
<td>5.48</td>
<td>82.19</td>
<td>5.93</td>
</tr>
<tr>
<td>Baseline+R</td>
<td>79.57</td>
<td>2.04</td>
<td>79.22</td>
<td>2.96</td>
</tr>
<tr>
<td>Baseline+SM</td>
<td>83.51</td>
<td>5.98</td>
<td>81.97</td>
<td>5.71</td>
</tr>
<tr>
<td>Baseline+SR</td>
<td>80.11</td>
<td>2.58</td>
<td>79.95</td>
<td>3.69</td>
</tr>
<tr>
<td>Baseline+MR</td>
<td>83.43</td>
<td>5.90</td>
<td>82.44</td>
<td>6.18</td>
</tr>
<tr>
<td>Baseline+SMR</td>
<td>83.96</td>
<td>6.43</td>
<td>82.65</td>
<td>6.39</td>
</tr>
</tbody>
</table>

Table 4: Ablation study of reciprocal mechanism.

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean $\bar{J}$</th>
<th>$\Delta \bar{J}$</th>
<th>Mean $\bar{F}$</th>
<th>$\Delta \bar{F}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>77.53</td>
<td>-</td>
<td>76.26</td>
<td>-</td>
</tr>
<tr>
<td>Nonreciprocal</td>
<td>83.01</td>
<td>5.51</td>
<td>82.19</td>
<td>5.93</td>
</tr>
<tr>
<td>Reciprocal</td>
<td>83.74</td>
<td>6.21</td>
<td>82.57</td>
<td>6.31</td>
</tr>
</tbody>
</table>

Table 5: Ablation study for STAFM.

4.3. Ablation Study

Our ablation experiment is conducted based on our lightweight model without applying the CRF post-processing operation on DAVIS-16 dataset.

**Primary Objects.** We evaluate the effectiveness of our proposed model for transforming intra-frame saliency, moving cues, and temporal coherence. The results are reported in Table 3. We use the vanilla encoder-decoder architecture without any transformation or feature fusion module as Baseline. Then, we search three kinds of candidate primary objects and transform the intra-frame contrast features for local salient objects (denoted as $S$), the motion features for moving objects (denoted as $M$) and the temporal coherence for recurring objects (denoted as $R$). On the one hand, we find that identifying the three kinds of primary objects all contribute to the whole model according to performance gain while considering one more kind of primary objects. On the other hand, we find that the performance improvement is most obvious. Therefore, the moving objects play the most important roles in primary objects comparing with salient objects and the recurring objects.

**Reciprocal Mechanism.** We study the quality of the motion features transformed to the appearance to show the effectiveness of our reciprocal transformation between motion features and appearance features. We adopt the vanilla encoder-decoder architecture without transforming any motion features as Baseline. Then, we transform the motion features directly from the motion stream without salient appearance as Nonreciprocal. Furthermore, our reciprocal transformation for transforming between appearance and motion with moving primary objects as Reciprocal. The results are reported in Table 4.

**STAFM.** To evaluate the effectiveness of our STAFM, we compare the performance for our RTNet with STAFM ($Ours \ w/ \ STAFM$) and the model simply skip connect both motion and appearance features ($Ours \ w/o \ STAFM$). The results are reported in Table 5. The gain of our STAFM comes from the fusion of the crucial spatial and temporal features while removing the redundant features.

5. Conclusion

In this paper, we propose the reciprocal transformations to identify the three kind of primary objects: salient objects, recurring objects and moving objects by searching the intra-frame dependency, the correlation between motion and appearance features, and temporal coherence to the appearance features. Besides, to eliminate the moving background objects, the reciprocal scheme transform appearance features back to motion features to filter moving objects with distinct appearance. Finally, we propose a spatial temporal attentive feature fusion module to dynamically and selective fuse spatial and temporal features.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61972162, 62072110), the Natural Science Foundation of Fujian Province under Grant 2018J07005, and CCF-Tencent Open Research fund.
References


[37] Sucheng Ren, Chu Han, Xin Yang, Guoqiang Han, and Shengfeng He. Tenet: Triple excitation network for video salient object detection. 1
[50] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 6