Unsupervised video object segmentation: An affinity and edge learning approach

Sundaram Muthu [1], Ruwan Tennakoon [1], Reza Hoseinnezhad [1], and Alireza Bab-Hadiashar [1]

[1]Affiliation not available

October 30, 2023

Abstract

This paper presents a new approach to solve unsupervised video object segmentation˜(UVOS) problem (called TMNet). The UVOS is still a challenging problem as prior methods suffer from issues like generalization errors to segment multiple objects in unseen test videos (category agnostic), over reliance on inaccurate optic flow, and problem towards capturing fine details at object boundaries. These issues make the UVOS, particularly in presence of multiple objects, an ill-defined problem. Our focus is to constrain the problem and improve the segmentation results by inclusion of multiple available cues such as appearance, motion, image edge, flow edge and tracking information through neural attention. To solve the challenging category agnostic multiple object UVOS, our model is designed to predict neighbourhood affinities for being part of the same object and cluster those to obtain accurate segmentation. To achieve multi cue based neural attention, we designed a Temporal Motion Attention module, as part of our segmentation framework, to learn the spatio-temporal features. To refine and improve the accuracy of object segmentation boundaries, an edge refinement module (using image and optic flow edges) and a geometry based loss function are incorporated. The overall framework is capable of segmenting and finding accurate objects' boundaries without any heuristic post processing. This enables the method to be used for unseen videos. Experimental results on challenging DAVIS16 and multi object DAVIS17 datasets shows that our proposed TMNet performs favourably compared to the state-of-the-art methods without post processing.

Unsupervised video object segmentation: An affinity and edge learning approach.

Sundaram Muthu[1], Ruwan Tennakoon[2], Reza Hoseinnezhad[1], Alireza Bab-Hadiashar[1]

***Abstract*—This paper presents a new approach to solve unsupervised video object segmentation (UVOS) problem (called TMNet). The UVOS is still a challenging problem as prior methods suffer from issues like generalization errors to segment multiple objects in unseen test videos (category agnostic), over reliance on inaccurate optic flow, and problem towards capturing fine details at object boundaries. These issues make the UVOS, particularly in presence of multiple objects, an ill-defined problem. Our focus is to constrain the problem and improve the segmentation results by inclusion of multiple available cues such as appearance, motion, image edge, flow edge and tracking information through neural attention. To solve the challenging category agnostic multiple object UVOS, our model is designed to predict neighbourhood affinities for being part of the same object and cluster those to obtain accurate segmentation. To achieve multi cue based neural attention, we designed a Temporal Motion Attention module, as part of our segmentation framework, to learn the spatio-temporal features. To refine and improve the accuracy of object segmentation boundaries, an edge refinement module (using image and optic flow edges) and a geometry based loss function are incorporated. The overall framework is capable of segmenting and finding accurate objects' boundaries without any heuristic post processing. This enables the method to be used for unseen videos. Experimental results on challenging DAVIS16 and multi object DAVIS17 datasets shows that our proposed TMNet performs favourably compared to the state-of-the-art methods without post processing.**

***Index Terms*—Video object segmentation, Neural Attention, Edge refinement, Affinity learning, Multi-cue segmentation, Correlation clustering.**

I. Introduction

Video object segmentation (VOS) is one of the important tasks in computer vision. It involves pixel level Segmentation of independent moving objects across frames in a given video sequence. Solving the VOS problem is essential for the development of video analysis and scene understanding tools. Applications of VOS include video editing [1], autonomous driving [2], robotics, surveillance and tracking [3].

Traditionally, VOS (or motion segmentation as it was called) was solved via fitting geometric models to matched key-points in adjacent frames [4]. More recently, deep learning based methods have become prominent. Existing deep learning based VOS methods can be classified as semi-supervised, interactive and unsupervised methods based on the amount of human involvement in the segmentation process. The semi-supervised methods (SVOS) require annotations of objects of interest in the first frame. The interactive methods requires user interactions like scribbles, to guide and correct the segmentation. Recently introduced unsupervised VOS methods (UVOS) are expected to identify all moving objects in the scene, with no prior information about the number of objects or manual annotation of the first frame.

Unsupervised video object segmentation is challenging since the segments or the number of segments is unknown at start. Other challenges include dynamically varying number of objects, camouflaged object motions, occlusions, articulated non-rigid object motions, background motion, etc.

Prior works follow three main strategies to solve the UVOS problem. The first strategy is based on using motion and appearance features together for video object segmentation. The second strategy is to use semantic segmentation of objects in a video frame, followed by tracking of the detected objects using temporal information in subsequent video frames. The third strategy relies on neural attention leveraging temporal information from optic flow to focus on obtaining better features to represent moving objects.

Motion and appearance feature based methods [6]–[9] generally rely on a two stream architecture. i.e. processing image (appearance features), and optic flow (motion features) independently. The two stream architecture works well only if the data correspondences provided by optic flow are accurate. However, optic flow estimates are often inaccurate around: object boundaries [10], non-textured regions, and fast-moving objects [11]. In such cases, motion features would not be reliable to complement the appearance features for accurate segmentation. This leads to an over reliance of these methods towards the appearance features of the objects.

Object detection and tracking methods [12]–[14] use state-of-the-art object detection methods like Mask-RCNN [15] to detect foreground objects and track all detected objects using tracking algorithms. Tracking has its own challenges: it often suffers from drift, and it relies heavily on object re-identification to track missed or re-appearing objects. These methods usually suffer from generalization errors, when applied to larger test videos containing objects that do not appear in the training data.

To overcome these challenges, Motion-Attentive Transition Network (MATNet) [5] introduced a neural attention mechanism similar to how humans perform motion segmentation. A motion-attentive module uses optic flow to focus on moving objects and to obtain better features. Even though this method overcomes the problems associated with the first two strategies for performing UVOS, they still have several drawbacks.

[1] School of Engineering, RMIT University, Victoria, Australia.
[2] School of Science, RMIT University, Victoria, Australia.

 This work was supported by the Australian Research Council through an ARC Linkage Project grant (LP160100662).

Fig. 1: Qualitative comparisons for UVOS on the DAVIS17 dataset. a) Our method is capable of identifying multiple moving objects due to affinity learning (*pigs* sequence), b) Our temporal motion attention module improves segmentation results of articulated objects (*camels* sequence) c) Our edge refinement module refines object boundaries (*blackswan* sequence). The columns from left to right are the input frame, the optical flow, the segmentation ground-truth, the results of MATNet [5], and TMNet, respectively.

1) This approach mainly focus on foreground/background segmentation, not useful for practical real world applications that has multiple moving objects. 2) Neural attention is based on optic flow information between two consecutive frames that is ambiguous, leading to a bias towards semantic object classes due to the lack of temporal tracking information. 3) The method does not capture fine details of objects, leading to poor object boundaries.

To solve the UVOS problem, we propose a new method that uses temporal neural attention and edge refinement modules to predict segmentation affinities, that are later clustered to obtain the required segmentation. This approach overcomes the issues mentioned above. For example, Fig 1 compares the segmentation results of our method with the segmentation results of MATNet [5]. The method is capable of detecting multiple-objects (Fig 1a), articulate objects (Fig 1b) with accurate object boundaries (Fig 1c).

To address issue 1 (detecting multiple-objects), we propose to predict affinities instead of predicting segmentation masks directly. The predicted affinities represent the relationship between neighbouring pixels. The neighbouring pixel can either belong to the same object (pixels lie inside an object) or belong to different objects (pixels straddle object boundaries). This relationship motivates the need for predicting the probability that neighbouring pixels belong to the same object.

To address issues 2 and 3, we designed temporal motion attention and edge refinement modules. Humans are attracted first to anything that moves before learning to map objects to semantic object classes [16]. Inspired by this fact, the temporal motion attention module performs temporal neural attention to focus on the moving objects. This enables us to deal with highly complex object motions, and to resolve the issue of dynamic appearance changes of objects as shown in Fig 1. Edge refinement module uses the combination of image and flow edges to refine the edges of the segmentation, correcting errors occurring due to inaccurate motion information at object boundaries. Addition of this boundary information also captures fine details of objects.

The main contributions of this paper include:

- The introduction of temporal motion attention module, where tracking information is used as an attention mechanism to focus on segmenting moving objects.
- Predicting affinities instead of predicting the foreground/background segmentation masks directly to overcome the problem of segmenting multiple moving objects.
- Presenting a UVOS framework with results comparable to the state-of-the-art for UVOS task on single object DAVIS16 and multi-object DAVIS17 datasets.

This paper is organized as follows: Section II introduces the related work. Section III describes the network architecture of our TMNet deep learning model, implementation details and the loss function used. Ablation studies on DAVIS16 dataset is presented in Section IV, which show the performance improvement due to the use of temporal neural attention and edge refinement modules. Experimental results on DAVIS17 dataset show the effectiveness of the method to perform multiple object UVOS by predicting affinities instead of directly predicting a foreground/background segmentation. Section V concludes the paper and discusses future work.

II. Related Works

A. VOS Definition

According to Gestalt "common fate" principle [17], VOS is the grouping of pixels with the same motion. According to [18], VOS is defined as segmenting all objects that move relative to the background. Since the same motion grouping is not the same as same object grouping, ambiguities arise for some scenarios. For example, intermittent object motions (objects static for few frames in sequence), articulated objects (only

part of the object moves), similarly moving objects (different objects with same motion), etc. are analysed in [19] to propose an acceptable definition to resolve the ambiguities.

The following summarizes the definitions commonly used [19]: a) Entire object must be segmented even if only part of the object moves. b) Static objects must not be segmented even if they have moved before or could move later (to maintain causality). c) Similarly moving objects must be segmented separately only if they are not connected in 3D. Also, VOS differs from tracking by assigning precise masks to the tracked individual objects. Based on the amount of human involvement, VOS is classified as semi-supervised, interactive and unsupervised methods. Unlike semi-supervised methods that require first frame ground-truth of objects to be segmented, unsupervised or zero shot VOS methods should identify all moving object(s) without any prior information. *Our method also solves the challenging UVOS problem that requires the model to have generalizing capability.*

B. UVOS based on multiple cues

Appearance based methods fail to segment texture-less objects. They also segment static background objects as they lack the motion information. Flow-based techniques using optic flow based on brightness constancy assumption generally leads to over-segmentation of non-rigid objects, and often fail when there are occlusions, camouflaged objects, or degenerate motions. To resolve these issues, two stream architecture based methods [20]–[22] were introduced that combine both image-based appearance features and optic flow-based motion features. The inaccuracy of optic flow at object boundaries further leads to poor segmentation. Methods such as ARP [23] and [24] refine optic flow-based motion boundaries using edge cues to obtain better performance at object boundaries. Joint estimation of optic flow and UVOS [9], [25] has also shown to be somewhat effective in overcoming the problem of over-segmentation.

Advances in deep learning for object recognition [15], [26] have enabled the use of temporal information to track object proposals and generate consistent segmentation for the entire video [12], [14], [27]. Zhao et al. [28] perform UVOS for multiple objects by detecting and tracking objects using human-centric re-identification. UnOVOST [12] generate tracklets for object proposal masks and merge long-term consistent tracklets to perform segmentation. Object based detection methods use Siamese Re-ID networks for association. This leads to failure for fast moving objects, occlusions, and non-rigid motions. In contrast to above methods, our approach effectively combines all the cues discussed above in our TMNet framework to produce category agnostic multi-object UVOS. Most methods discussed also perform only binary foreground background segmentation, thereby limiting their applications scenes with only one moving object. In contrast, we perform multi-object UVOS by predicting affinities instead of predicting the segmentation directly. These are explained in section III.

C. Clustering methods

Clustering is an important part of bottom-up UVOS methods. Bottom-up based methods obtain low level point trajectories from appearance and motion information, and merge them to form high level object segmentation [29]. In [30], key-segments are identified with persistent appearance and motion cues. The key segments are later clustered to obtain the foreground object segmentation. Recently, STEm-Seg [31] extend the clustering across both spatial and temporal domains to model videos to segment and track all moving objects. These methods have the advantage of generating category-independent object segmentation. But segmenting multiple moving objects remains a challenge if the number of clusters are not known a priori and keep changing dynamically in a video sequence. To overcome this problem, Keuper at al. [32] formulated trajectory clustering problem as a correlation clustering problem. Correlation clustering or minimum cost multicuts finds the optimal number of segments automatically [33]. Keuper et al. [34] used heuristic algorithms to solve the correlation clustering efficiently. Different from the methods discussed above, our method leverages the benefits of both correlation clustering and advances in deep neural networks in a single TMNet framework that improves the generalization capability without suffering from over-segmentation problems faced by the bottom-up methods. The overall framework is explained in section III.

D. Attention in neural networks

Inspired by human perception, attention mechanism is recently used in deep neural networks to improve the performance of various applications like Attention guided object segmentation [35], Dynamic visual attention prediction [5], Visual question answering [36], etc. Attention helps the network to form effective feature representations from the data. The effectiveness is achieved by focusing on only the relevant informative regions of interest (avoiding unnecessary information). Neural attention has also been used for improving the performance of the UVOS problem [5], [37]–[39].

In-order to avoid the use of computationally expensive optic flow for UVOS, AGNN [37] first used an attention mechanism to capture the higher order relationships in a message passing graph neural network framework. Differently, COSNet [38] solved UVOS using co-attention between frames in a video sequence by a Siamese neural network to learn the global context. MATNet [5] introduced a motion attentive two stream interleaved encoder to learn powerful spatio-temporal features for UVOS by an attention mechanism using optic flow to focus on only the moving objects. FEM-Net [39] extended MATNet [5] by additionally using the optic flow edge information in a Flow edge connect module that helps in segmenting the foreground salient objects, and their boundaries accurately. Our method also extends the motion based attention mechanism proposed by MATNet [5] to learn powerful spatio-temporal features by additionally incorporating temporal context from several frames in the video into the attention mechanism, which helps to resolve ambiguities in optic flow information of complex dynamic scenes in two

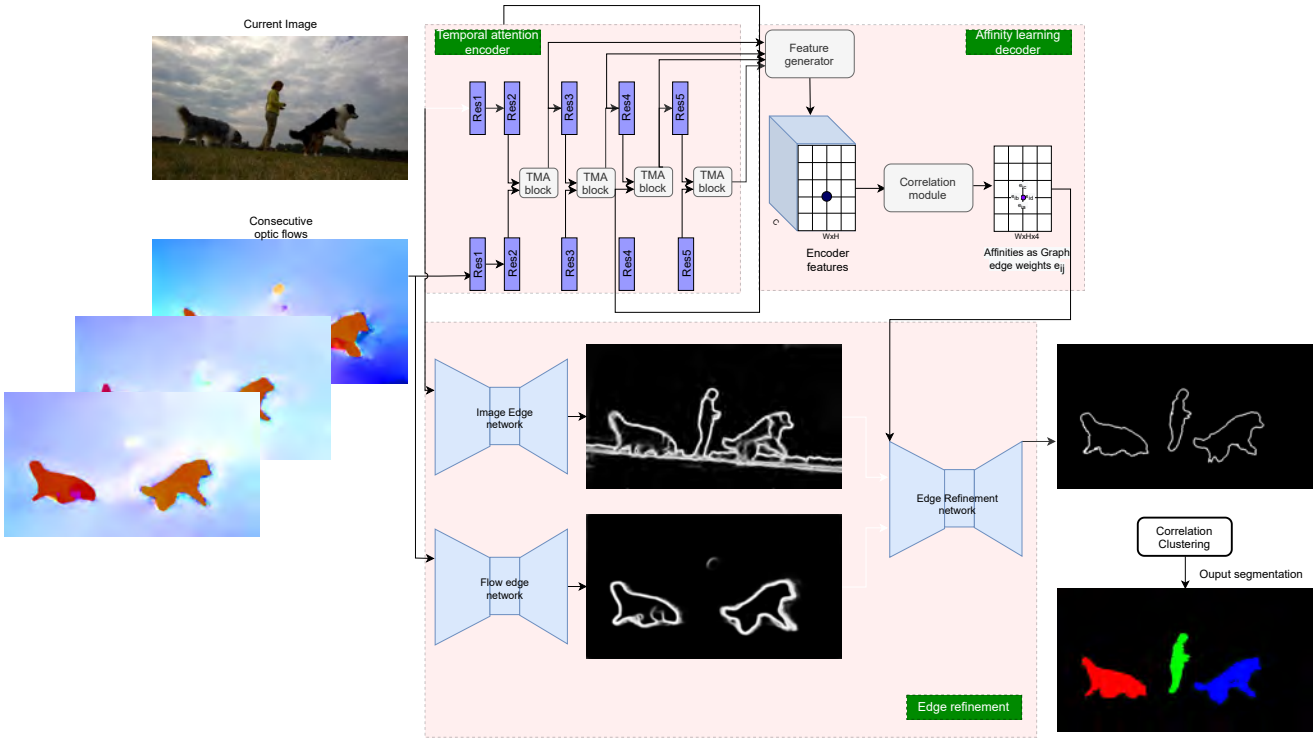

Fig. 2: Overview of the network architecture of our TMNet method. The temoral attention encoder uses both image and optic flow to generate robust features. The affinity learning decoder produces a neighbourhood affinity matrix. The edge refinement stage uses both image and flow edges to refine the boundaries of the affinity matrix. The affinity is later clustered to obtain the required segmentation.

consecutive frames. Different from the previous methods, our method also predicts affinities instead of predicting the output segmentation directly. This enables our method to be used for category agnostic multiple object segmentation.

III. Proposed Method

A. Problem Statement

Given a video sequence $I = \{I_1, I_2, \cdots, I_{t-1}, I_T\} \in \mathbb{R}^{W\times H\times 3\times T}$, and the forward optic flow $OF = \{OF_1, OF_2, \cdots, OF_{t-1}\} \in \mathbb{R}^{W\times H\times 2\times T}$, the objective of VOS is to generate multi-object segmentation $S = \{S_1, S_2 \cdots, S_{t-1}, S_t\}$. Here, each video frame, I_t, denotes an RGB image with width W and height H. The segmentation at each time frame, S_t, consists of a set of binary object segmentation masks $\{M_1, M_2, \cdots, M_k\} \in \mathbb{R}^{W\times H}$. k denotes the number of independent moving objects and this value can vary dynamically in the video sequence due to new objects entering the scene or objects leaving the scene.

B. Network Architecture

We propose an end-to-end deep neural network TMNet for category agnostic multiple object UVOS, to predict affinities by learning powerful spatio-temporal features through neural attention. Our TMNet framework consists of three main stages: 1) Temporal motion attentive encoder, 2) Affinity learning decoder, and 3) Edge refinement network. The framework of the proposed TMNet framework is illustrated in Fig 2.

1) Temporal motion attentive encoder: This encoder uses a motion attentive two stream interleaved architecture to learn robust feature representations for moving object(s). We can achieve a degree of robustness due to the use of our Temporal motion attention (TMA block) described in section III-D that uses a neural attention mechanism to focus on only the moving object(s).

For the appearance stream, we use the image $I_t \in \mathbb{R}^{W\times H\times 3}$ and for the motion stream, we use the optic flows calculated between the current frame t and δ previous frames $OF = \{OF_{t-1}, \cdots, OF_{t-\delta}\} \in \mathbb{R}^{W\times H\times 2\times \delta}$. The optic flows are converted to RGB domain before passing through the convolutional layers ($\hat{OF} \in \mathbb{R}^{W\times H\times 3\times \delta}$). The encoder processes the multi-cue information from the inputs and produces a robust feature representation $F_t \in \mathbb{R}^{W\times H\times C}$:

$$F_t = \textsc{Encoder}(I_t, \{OF_{t-1}, \cdots, OF_{t-\delta}\}). \quad (1)$$

Different from the previous neural attention based interleaved encoders for UVOS [5], our method uses additional temporal information from the previous δ frames. This additional information helps the network to detect objects temporarily stopping for a few frames. It also improves the detection capability of non-rigid objects with articulated motions.

Firstly, we use the initial 5 convolutional blocks of the standard ResNet backbone to extract appearance feature F_a from the image I_t, and temporal features $F_m = \{F_{m1}, \cdots, F_{m\delta}\}$ from the optic flows $\hat{OF}$ at various residual stages with

different spatial resolutions (1/2$^{\text{th}}$, 1/4$^{\text{th}}$, 1/8$^{\text{th}}$, 1/16$^{\text{th}}$) of the original image size.

Next, we use the temporal features F_m to update the appearance features F_a at all intermediate stages, and obtain the enhanced appearance features $\hat{F_a}$. This enhancement is performed using the TMA block for each stage $i \in \{2, 3, 4, 5\}$ as follows:

$$\hat{F_a}^i = \text{TMA}(F_a{}^i, \{F_{m1}, \cdots, F_{m\delta}\}^i). \tag{2}$$

The enhanced appearance features $\hat{F_a}$ and the motion features F_m are concatenated to form the combined features. The combined features of all four residual stages are scaled to the size of the image, and combined to obtain the final robust feature representation F_t. Finally, the features F_t are fed to the affinity learning decoder stage.

2) Affinity learning decoder: The affinity learning decoder is designed to use the temporal motion attentive features $F_t \in \mathbb{R}^{W \times H \times C}$, and the predict affinities $A_t \in \mathbb{R}^{W \times H \times 4}$:

$$A_t = \textsc{Decoder}(F_t). \tag{3}$$

The segmentation affinity is defined for every pixel $u \in I_t$, and one of its neighbouring pixels $v \in N(u)$. The affinity describes the probability that the selected pixel, and its neighbour belongs to the same motion. The value varies from 0 to 1. The affinity is 1 if the label of the pixel and its neighbour matches (high affinity for edges between pixels within the same segmentation). The affinity is 0 if the labels do not match (low affinity for edges between pixels that belong to different segments). This representation has the advantages of permutation invariance and fixed size, making it easy to use for training purposes. Here we face the trade-off between accuracy and memory/time. Increasing the neighbourhood size will give us more accurate results at the expense of the large memory footprint of segmentation affinity leading to more prediction time. Hence we restrict our model to predict affinities for only the four immediate neighbouring pixels.

We utilize the well-known cosine similarity function to learn affinities from temporal motion attentive features. Cosine similarity is calculated between the spatio-temporal features F_t of the image I_t, and warped features F_t^{shifted} of the same image. We use the simple cosine similarity function instead of learning from a cost volume constructed from the features (since constructing the full cost volume makes model large in-terms of memory).

3) Edge refinement network: The edge refinement network is designed to refine the object boundaries by updating the predicted affinities A_t obtained from the decoder. The edge maps $I_e \in \mathbb{R}^{W \times H}$ of the image I_t, and the edge maps $OF_e \in \mathbb{R}^{W \times H}$ of the optic flow OF_{t-1} are both used to update the predicted affinities $A_t \in \mathbb{R}^{W \times H \times 4}$:

$$\hat{A} = \textsc{Refine}(A_t, I_e, OF_e). \tag{4}$$

The edge refinement block is described in detail in section III-E. The refined edges are finally clustered to obtain the required segmentation as described in III-C

C. Predicted Affinity to video object segmentation

Our network is not limited the specific set of object classes present in the training class since we predict affinities. But in-order to obtain the required segmentation S from the affinity $\hat{A}$ predicted by our TMNet model, we need to perform a clustering step. Unlike other clustering methods, correlation clustering finds the optimal number of clusters automatically. So we apply correlation clustering on a pixel grid graph that uses the predicted affinities.

Firstly, we create a pixel grid graph $G = (V, E, W)$ for the image I_t from predicted affinities $\hat{A}$ as follows:

- Nodes V: Set of $N = W \times H$ vertices for every pixel in the image I_t. W and H denote the width and height of the image I_t.
- Edges E: Set of edges $e_{uv} \in \mathbb{R}^{N \times 4}$ connect four neighbouring nodes v (left, right, top and bottom nodes) of every node u that form the pixel grid.
- Weights W: The affinities $\hat{A} \in \mathbb{R}^{N \times 4}$ predicted by our model are used as weights w_{uv} for every edge defined in the graph. It is the cost associated with assigning the two nodes u and v of the edge e_{uv} to distinct components.

Next, the segmentation is performed by solving the optimization problem defined on the pixel grid graph created using the predicted affinities as edge weights. The correlation clustering or the graph multicut optimization problem is solved using the method described in [34]. The output is a unique decomposition of the graph G, which assigns $0/1$ labels to all the edges. Edges labelled 1 straddle distinct clusters.

Finally, once edges straddling distinct clusters are identified, the clusters can be separated to obtain the output segmentation S_t as the set of binary object segmentation masks $\{M_1, M_2, \cdots, M_k\} \in \mathbb{R}^{W \times H}$. Our method automatically assigns the optimal number of independently moving objects k which can vary dynamically in the video sequence.

D. TMA block

The Temporal Motion Attention (TMA block) uses the temporal features $F_{mi} \in \mathbb{R}^{W \times H \times C_m}$ to update the appearance feature $F_a \in \mathbb{R}^{W \times H \times C_a}$ at all intermediate stages. The output of the block is the enhanced appearance features $\hat{F_a} \in \mathbb{R}^{W \times H \times C_a}$.

We develop a temporal aggregation block within our TMA block to extend the Motion Attentive Transition block developed by MATNet [5] in-order to accommodate the temporal information from previous δ frames. Firstly, there is a soft attention that weights each of feature maps ($F = F_a, \{F_{m1}, \cdots, F_{m\delta}\}$) at every pixel location. This is performed by a 1x1 conv function that learns the probability that a particular region of the feature map is important. The probability is then normalized using a softmax function to obtain the normalized importance weights $I \in \mathbb{R}^{W \times H}$. The feature maps in F are all converted separately to obtain the spatial attentive features Z by using a channel-wise Hadamard product:

$$Z_a^c = \textsc{Softmax}(W_a(F_a)) \odot F_a^c. \tag{5}$$

Next, we find the correlation between the spatial attentive appearance features Z_a, and each of the spatial attentive

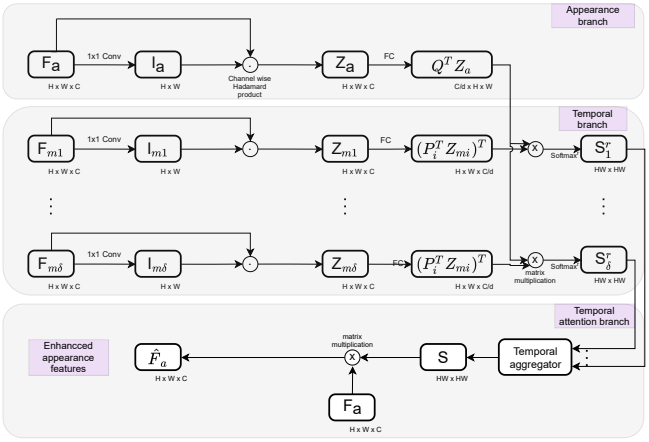

Fig. 3: Computational graph of our TMA block.

motion features Z_{mi}. This correlation S_i, or the non-linear affinity, is learnt to find the relationship between the two feature spaces. The affinity is high (both appearance and temporal motion features are similar) for regions of the image where moving object(s) are present:

$$\begin{aligned} S_i &= Z_{mi}^T(W_i)Z_a & S_i \in \mathbb{R}^{WH\times WH} \\ &= Z_{mi}^T(P_iQ^T)Z_a & P_i, Q \in \mathbb{R}^{C\times\frac{C}{d}} \\ &= (P_i^T Z_{mi})^T(Q^T Z_a) & i \in \{1,\cdots,\delta\}, \end{aligned} \tag{6}$$

where P_i and Q are trainable weights learnt during training to compress the size of the model and avoid the overfitting problem. Q matrix which is learnt to compress the appearance features, are shared in the calculation of all S_i.

In the third step, we normalize the affinity matrix S_i along the row dimension to ensure that the sum of the contributions of all channels is 1. Finally we aggregate the normalized affinities S_i^r for all $delta$ optic flows to obtain the temporal motion attention factor S.

$$\begin{aligned} S_i^r &= \textsc{Softmax}(S_i) \\ S &= \textsc{Aggregation}(S_1^r,\cdots,S_\delta^r). \end{aligned} \tag{7}$$

The aggregation function can be implemented by simple maximum, minumum, average or median values. Experiments in section IV indicate that the average performs better than other functions.

Finally, the enhanced appearance features $\hat{F}_a \in \mathbb{R}^{W\times H\times C_a}$ are obtained from the temporal motion attention factor S by

$$\hat{F}_a = F_a \times S. \tag{8}$$

E. Edge Refinement module

Fig (4) shows the motivation for using both the image edge I_e, and the flow edge OF_e information jointly in the edge refinement module of our TMNet model. First row shows failure of using optic flow edges OF_e to detect the swan (inaccurate boundaries, background noise due to moving water). As our proposed method uses additional image edge cues, it can overcome this issue and segment the swan correctly. Second row shows failure of using image edges I_e to segment the moving car (due to the object having no texture). Again, our proposed method has been able to detect the car correctly using the flow edge information. The above two cases highlight the fact that both image and flow edges act as complementary information aiding the segmentation together to refine the boundaries. Third row also shows the improvement due to the complementary information even though both flow and image edges are accurate.

After predicting the affinities $A_t \in \mathbb{R}^{W\times H\times 4}$, we concatenate it with the edge maps $I_e \in \mathbb{R}^{W\times H}$ of the image I_t, and the edge maps $OF_e \in \mathbb{R}^{W\times H}$ of the optic flow OF_{t-1} to form the input $R_i \in \mathbb{R}^{W\times H\times 6}$ to the convolutional modules ($R_i = \textsc{concat}(A_t, I_e, OF_e)$). We capture the fine details of the boundaries by using the edge features in three consecutinve $conv$ blocks. The last convolutional stage reduces the feature dimension from six back to four. The output refined affinity $\hat{A}$ at the end of the block is clustered to obtain the required boundary aware segmentation.

F. Loss Function

Since our network predicts affinities instead of the segmentation directly, we use a loss function based on the predicted affinities. For the training, we formulate our loss between the predicted affinities and the ground-truth affinity (instead of the usually used binary cross entropy loss between the predicted and the ground-truth segmentation).

Firstly, in-order to define a loss term, we need to convert the labelled ground-truth segmentation to ground-truth affinities. The ground-truth affinity is 1 if the label of the pixel in the image, and its neighbour are the same. Consider an image $I_t \in \mathbb{R}^{W\times H\times 3}$, and its segmentation ground-truth $L_t \in \mathbb{R}^{W\times H}$. The ground-truth affinity matrix $A \in \mathbb{R}^{W\times H\times M}$ is then defined for each pixel $u \in I_t$ and one if its neighbouring pixels $v \in N(u)$, as follows:

$$A_{uv} = \begin{cases} 1, & \text{if } L(u) = L(v), \\ 0. & \text{otherwise,} \end{cases} \tag{9}$$

where, W & H are the width and height of the image, M is the number of pixels v in the neighbourhood of a pixel u.

For the loss function, we use the mean square error (MSE) function between the ground-truth affinities A defined previously, and the affinities $\hat{A}$ predicted by our TMNet model given by:

$$\begin{aligned} L(A,\hat{A}) &= \tfrac{1}{\alpha_e}\sum_{u\in I}\sum_{v\in N(u)}(1-A_{uv})\times L_e(A_{uv}-\hat{A}_{uv}) \\ &\quad +\tfrac{1}{\alpha_{ne}}\sum_{u\in I}\sum_{v\in N(u)}A_{uv}\times L_{ne}(A_{uv}-\hat{A}_{uv},w). \end{aligned} \tag{10}$$

We split the loss into two parts to overcome the imbalance problem in the ground-truth affinity as number of $0's$ (edges - labels of the compared pixels do not match each other) $\ll$ number of $1's$ (non-edges - labels of the compared pixels match each other) in A. The first term is the normal mean square error loss for $0's$ (edges). The second term is the weighted mean square error loss for $1's$ (non-edges). The equations for those losses are as follows:

$$L_e(x) = x^2 \tag{11}$$

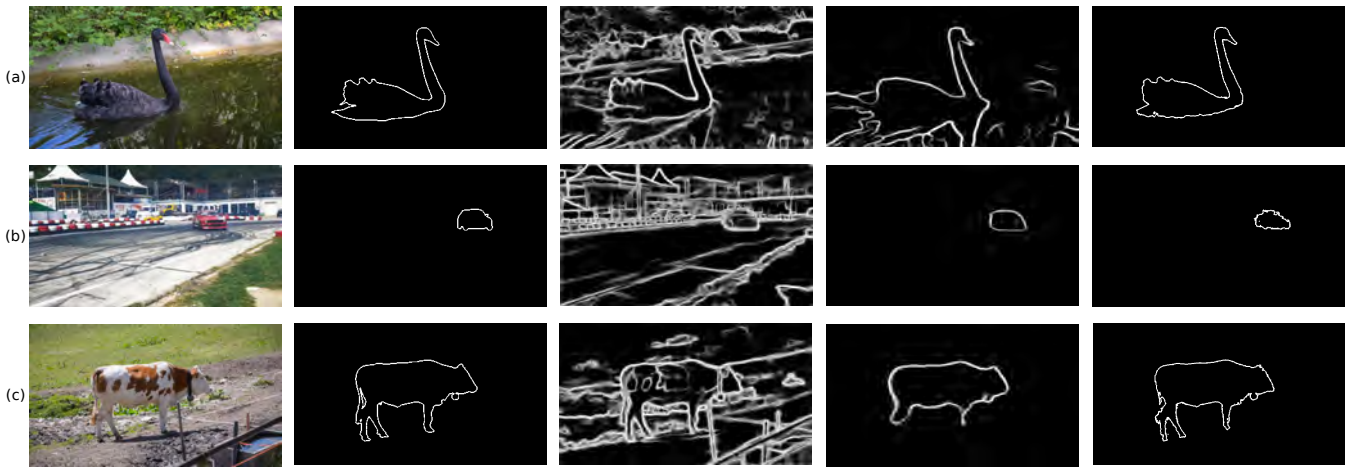

Fig. 4: Usage of both image and flow edges together in our edge refinement module. a) Flow edges not reliable in comparison to image edges (*blackswan* sequence), b) Image edges not reliable in comparison to flow edges (*drift-straight* sequence) c) Both image and flow edges complement each other (*cows* sequence). The columns from left to right are the input frame, ground-truth edge annotation, image edge, flow edge, and the results of our TMNet edge refinement module, respectively.

$$L_{ne}(x, w) \;=\; w \;*\; x^2. \tag{12}$$

The weights $w \in \mathbb{R}^{W \times H}$ are the edge weights generated using the normalized gradient magnitude of the image. We incorporate the use of geometry to control the importance of non-edge pixels during training. This loss penalizes edges in an image, that are non-edges in A (penalize static object boundaries that do not appear in the motion boundaries).

Normalization terms α_e and α_{ne} count the number of the edges and weighted sum of the non-edges in A, respectively.

$$\begin{aligned} \alpha_e &= \sum_{u \in I} \sum_{v \in N(u)} (1 - A_{uv}) \text{ and} \\ \alpha_{ne} &= \sum_{u \in I} \sum_{v \in N(u)} A_{uv} \times W_u. \end{aligned} \tag{13}$$

G. Implementation Details

Our TMNet model is end-to-end trainable to predict affinities that are clustered to obtain the required segmentation.

Training: For pre-processing, the images are scaled to 384x512x3. We also augment training data to prevent overfitting. We use the open-source Flownet2 [41] for optic flow estimation. We also use RCF [42] for obtaining image and flow edge maps. For fair comparison, we adopt the same method for generating optic flow and edge maps in all of our experiments. We train the model only on the 30 training set video sequences of the DAVIS17 dataset [43] without the use of any additional training data. The model is trained from the scratch using the loss term and affinity ground-truth as explained in eqn (10) in a supervised manner using random initial weights. We train the network with a batch size of 2. We follow a learning rate of 10^{-4} for pre-training, and 10^{-5} for fine-tuning as the training schedule using the ADAM optimizer. The number of previous frames for extracting the temporal attention δ is chosen to be 3. The number of neighbours M for every pixel used to calculate the affinity is chosen to be the 4 immediate neighbours. We choose the values to be as low as possible since increasing both δ and M, results in increased accuracy in the output segmentation at the cost of increased model capacity and runtime.

Testing: For testing, we apply our trained TMNet model to the unseen videos. We use the current image and the optic flow of the previous δ frames to produce the output segmentation. It is to be noted that for obtaining the segmentation of the first δ frames, we craete copies of the initial optic flow.

Run-time: We implement our TMNet method in Pytorch on a NVidida TitanX GPU with 12GB memory for both training and testing. For our trained TMNet model, preprocessing steps like optic flow estimation takes around 0.07 s/frame, prediction of segmentation affinity takes 0.28 s/frame . Additionally, the clustering and tracking takes 2.17 s/frame. Our method has timing comparable to other UVOS methods but is also capable of performing category agnostic multi-object segmentation (segmentation of objects not seen by the training data).

IV. Experimental Results

We investigate the performance of our method on standard benchmarks for UVOS: DAVIS16 [40], and multiple object DAVIS17 [43] datasets. We compare with the state-of-the-art methods, and also perform ablation studies to understand the main advantages of specific components of our TMNet model.

A. Dataset and Evaluation metrics

We report results on two widely used benchmarks: single object DAVIS16 dataset [40] and multi-object DAVIS17 dataset [43]. The datasets contain many challenging video sequences with multiple objects, occlusion, fast moving objects, background clutter, articulated motion, etc.

DAVIS16 [40] contains 50 HD video sequences, 3455 manual instance segmentation ground-truths. DAVIS17 [43] is a more challenging benchmark extending DAVIS16 [40] to multiple moving objects, and contains 120 HD video sequences (60 for train, 30 for val, 30 for test-dev) and

Variant	J&F↑	ΔJ&F	J mean↑	ΔJ mean	J recall↑	ΔJ recall	F mean↑	ΔF mean	F recall↑	ΔF recall
w/o Temporal attention module	0.758	-5.01	0.774	-3.61	0.923	-2.73	0.742	-6.54	0.844	-4.52
w/o Edge refinement module	0.779	-2.38	0.792	-1.36	0.948	-0.01	0.766	-3.52	0.880	-0.45
Full TMNet model	**0.798**		**0.803**		**0.949**		**0.794**		**0.884**	

TABLE I: Ablation study: Key component analysis of proposed TMNet on DAVIS16 dataset [40].

10K manual instance segmentation ground-truths. The task is more challenging due to the inclusion of multiple objects that additionally create occlusions, background clutter, etc.

We use the following performance measures described in DAVIS challenge [43] to evaluate the performance of our method:

- Region similarity metric (J) : Intersection over union between predicted and ground-truth segmentations.
- Contour accuracy metric (F) : Boundary accuracy of predicted boundaries against the ground-truth.
- Overall global metric ($J\&F$) : Average of Jmean metric and Fmean.
- Temporal decay metric (T) : Measure of consistancy in labelling accross the video sequence.

The $mean$ of a metric is the average error measured across all objects in all video sequences. The $recall$ is the fraction of sequences scoring higher than a threshold of 0.5.

B. Ablation Study

To examine the effectiveness of the temporal neural attention and edge refinement components of our TMNet model individually, we performed an ablation study of our model on the DAVIS16 dataset [40].

The decrease in performance due to the removal of specific key components of our method is calculated as:

$$\Delta m = \left(1 - \frac{m_{partial}}{m_{partial}}\right) \times 100\% \tag{14}$$

where m represents the metric for which the decrease in performance is calculated ($J\&F$, Jmean, Jrecall, Fmean or Frecall). Δm represents the performance loss, $m_{partial}$ and m_{full} represent the metric values without and with the specific component of our model whose efficiency is studied. Table I shows the results of our key component analysis.

The first row in Table I shows the loss in performance of our model without the temporal attention module, compared to the full model performance described in the third row. There is a significant improvement of 5.01% in global $J\&F$ metric due to the inclusion of the temporal attention module. This shows that the addition of tracking information helps the networks performance by resolving ambiguities in appearance of objects for challenging scenarios (unseen objects, dynamically varying number of objects, occlusions, non-rigid motions, and noisy background).

Similarly, the second row in Table I shows the loss in performance of our model without the edge refinement module. There is a 2.38% improvement due to the inclusion of the edge refinement module. It is also seen that edge refine module improves the boundary metric Fmean by a large margin (3.52%). This performance gain is attributed to the edge refinement module as it improves the segmentation at object boundaries.

C. Evaluation Results

1) DAVIS16: We evaluated our method for UVOS on the single object DAVIS16 dataset [40]. Table II demonstrates that our method performs favourably compared to the state-of-the-art methods. Our method outperforms the state-of-the-art methods in one metric (Jrecall), and the second best performance in both boundary metrics (Fmean and Frecall) as highlighted in Table II. The overall performance indicates the robustness of the appearance features, and is attributed to the temporal attention module. The increased performance in the boundary metrics compared to the other methods is attributed to the use of edge refinement module that refines object boundaries.

Figure 5 shows qualitative results of our method. The results for the sequence 'camel' shows the capability of our model to work on non-rigid articulated motions. In sequence 'blackswan', our method robustly segments the object amongst the noisy background. Apart from the quantitative results, our method also has another advantage. Different from most of the existing methods which perform binary foreground/background segmentation, our method is able to perform segmentation and tracking for multiple moving objects as explained in the next section.

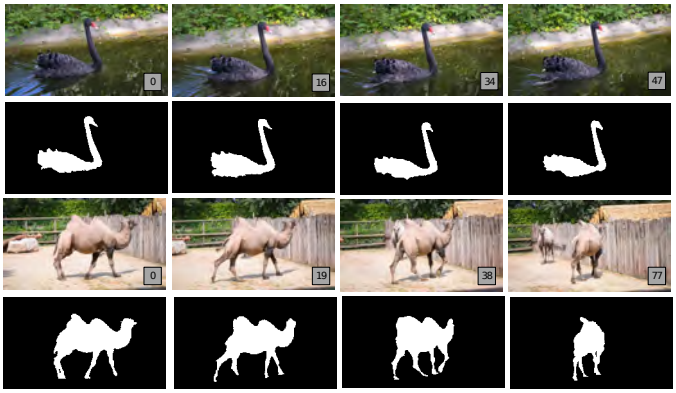

Fig. 5: Qualitative results of our method for UVOS on the single object DAVIS16 dataset. From top to bottom: *blackswan* sequence input and visual segmentation results, *camel* sequence input and visual segmentation results.

2) DAVIS17: To show that our method performs accurate segmentation for sequences with multiple moving objects, we applied our method to DAVIS17 dataset [43] and compared our results with existing methods. The performance of the

Method	J&F(Mean)↑	J(mean)↑	J(recall)↑	F(mean)↑	F(recall)↑
SFL [32]	67.0	67.4	81.4	66.7	77.1
LMP [22]	67.9	70.0	85.0	65.9	79.2
FSEG [44]	68	70.7	83.0	65.3	73.8
LVO [22]	73.9	75.9	89.1	72.1	83.4
ARP [23]	73.4	76.2	91.1	70.6	83.5
PDB [27]	75.8	77.2	90.1	74.5	84.4
MATNet [5]	76.4	78.0	92.1	74.8	87.5
AGS [35]	78.5	79.7	91.1	77.4	85.8
COSNet [38]	78.8	80.2	91.9	77.4	87.4
AGNN [37]	**79.9**	**80.7**	**94.0**	79.1	**90.5**
AnDiff [45]	**81.1**	**81.7**	90.9	**80.5**	85.1
FEMNet [39]	78.4	79.9	93.9	76.9	88.3
Ours	79.8	80.3	**94.9**	**79.4**	**88.4**

TABLE II: Comparison with the state-of-the-art methods for unsupervised video object segmentation on DAVIS-16 dataset [40].

Method	J&F(Mean)↑	J(mean)↑	F(mean)↑	F(decay)↓	Heuristic Post-processing
RVOS [21]	0.412	0.368	0.457	1.70	No
STEm-Seg [31]	0.647	0.615	0.678	1.20	No
MATNet [5]	0.586	0.567	0.604	1.80	Yes
UnOVOST [12]	**0.679**	**0.664**	**0.693**	**0.01**	Yes
Ours	0.461	0.427	0.496	0.03	No

TABLE III: Comparison with the state-of-the-art methods for unsupervised video object segmentation on multiple object DAVIS-17 dataset [43]. Our method, despite not using any post-processing, produces relatively accurate segmentation results.

proposed method is compared with several related state-of-the-art approaches (we have selected the top methods that do not use additional training data) including: (1) RVOS [21], (2) STEM-seg [31], (3) MATnet [5], (4) UnOVOST [12]. Table III compares the methods using the performance metrics described in the previous section.

The best results are obtained by UnOVOST [12]. This method is computationally expensive as it uses Mask-RCNN [15] for object proposal generation. It also has many heuristic post-processing steps requiring hyper-parameters tuning of multiple parameters (hyper-parameters are required for converting instance object mask proposals to short term tracklets, and merging short term tracklets to long term object trajectories), therefore limiting its use to specific datasets. Similarly MATnet [5] uses a CRF based dataset specific heuristic post-processing to convert the foreground/background saliency maps, to multi-object segmentation.

In contrast, our method uses no such post-processing, and shows comparable accuracy to other similar methods that perform multi-object segmentation directly without the requirement of any heuristic post-processing operations. Our method also has other advantages compared to the other methods. Unlike object detection and tracking methods [12], [21], [31], which require prior knowledge of known objects (object proposals from pre-trained Imagenet models) to solve UVOS accurately, our method does not depend on any prior knowledge of the segmented objects (as we only use affinities to perform UVOS). Hence our model is capable of generalizing well to unseen object classes.

Fig 6 shows qualitative results for performing UVOS on multi-object sequences of DAVIS17 dataset [43]. The sequence 'pigs' demonstrates that the model can handle multiple moving objects under challenging scenarios like occlusions and similar objects. Sequence 'dogs-jump' also shows the ability to handle category agnostic fast moving objects with similar appearance.

Fig. 6: Qualitative results of our method for UVOS on multiple object sequences from the DAVIS17 dataset. From top to bottom: *pigs* sequence input and visual segmentation results, *dogs-jump* sequence input and visual segmentation results.

V. Conclusions

In this paper, we propose a new method (TMNet) to solve UVOS. Our model combines appearance, motion and edge cues. The motion cues from consecutive frames of video sequences help to find temporal connections, guide our model to learn powerful object representations, as they resolve ambiguities in appearance features through neural attention towards the moving object(s). The edge cues help to refine the errors at object boundaries where motion cues are inaccurate. Different from previous neural attention UVOS methods, our method predicts affinities instead of predicting binary segmentation masks, making the method capable of handling multiple moving objects in one forward pass. The model is efficiently optimized by a loss function on the predicted affinities using geometric constraints. Our experiments on two

popular benchmarks,i.e., DAVIS16 and DAVIS17 demonstrate that TMNet is capable of effectively handling unseen object categories, multiple moving objects, occlusions, articulated object motions, and cluttered background. Extensive experiments on the datasets also show that the improvement in performance is due to the addition of the temporal neural attention and edge refinement modules.

Our method fails when objects in the video temporarily stop for multiple frames. This failure occurs since we process only selected number of frames at a time. So, we plan to extend the work further by storing important features of objects seen in a memory. This will allow the model to merge the tracklets accurately once the objects are re-identified later. Another area for improvement is to make use of the 3D scene flow available to incorporate the additional depth change information(not available in 2D optic flow) to aid the segmentation.

REFERENCES

[1] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2013.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[3] S. Muthu, R. Tennakoon, T. Rathnayake, R. Hoseinnezhad, D. Suter, and A. Bab-Hadiashar, "Motion segmentation of rgb-d sequences: Combining semantic and motion information using statistical inference," *IEEE Transactions on Image Processing*, vol. 29, pp. 5557–5570, 2020.

[4] R. Tennakoon, A. Sadri, R. Hoseinnezhad, and A. Bab-Hadiashar, "Effective sampling: Fast segmentation using robust geometric model fitting," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4182–4194, 2018.

[5] T. Zhou, J. Li, S. Wang, R. Tao, and J. Shen, "Matnet: Motion-attentive transition network for zero-shot video object segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 8326–8338, 2020.

[6] K. Xu, L. Wen, G. Li, L. Bo, and Q. Huang, "Spatiotemporal cnn for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 1379–1388.

[7] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, Conference Proceedings, pp. 1738–1745.

[8] J. Chang and J. W. Fisher, "Topology-constrained layered tracking with latent flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, Conference Proceedings, pp. 161–168.

[9] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 12 240–12 249.

[10] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, Conference Proceedings, pp. 3899–3908.

[11] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2014.

[12] I. E. Zulfikar, J. Luiten, and B. Leibe, "Unovost: Unsupervised offline video object segmentation and tracking for the 2019 unsupervised davis challenge," in *Proceedings of the 2019 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, 2019, Conference Proceedings.

[13] S. Xu, D. Liu, L. Bao, W. Liu, and P. Zhou, "Mhp-vos: Multiple hypotheses propagation for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, Conference Proceedings, pp. 314–323.

[14] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, Conference Proceedings, pp. 2192–2199.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, Conference Proceedings, pp. 2961–2969.

[16] E. S. Spelke, "Principles of object perception," *Cognitive science*, vol. 14, no. 1, pp. 29–56, 1990.

[17] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013.

[18] P. H. Torr, "Geometric motion segmentation and model selection," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 356, no. 1740, pp. 1321–1340, 1998.

[19] P. Bideau and E. Learned-Miller, "A detailed rubric for motion segmentation," *arXiv preprint arXiv:1610.10033*, 2016.

[20] X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, X. Tang, and C. C. Loy, "Video object segmentation with re-identification," *arXiv preprint arXiv:1708.00197*, 2017.

[21] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i Nieto, "Rvos: End-to-end recurrent network for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5277–5286.

[22] P. Tokmakov, K. Alahari, and C. Schmid, "Learning video object segmentation with visual memory," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4481–4490.

[23] Y. J. Koh and C.-S. Kim, "Primary object segmentation in videos based on region augmentation and reduction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 7417–7425.

[24] A. Papazoglou and V. Ferrari, "Fast object segmentation in unconstrained video," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, Conference Proceedings, pp. 1777–1784.

[25] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *Proceedings of the IEEE international conference on computer vision*, 2017, Conference Proceedings, pp. 686–695.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, Conference Proceedings, pp. 91–99.

[27] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 715–731.

[28] Z. Yang, Q. Wang, S. Bai, W. Hu, and P. H. Torr, "Video segmentation by detection for the 2019 unsupervised davis challenge'," *arXiv:1905.00737*, 2019.

[29] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *European conference on computer vision*. Springer, 2010, pp. 282–295.

[30] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *2011 International conference on computer vision*. IEEE, 2011, Conference Proceedings, pp. 1995–2002.

[31] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixé, and B. Leibe, "Stemseg: Spatio-temporal embeddings for instance segmentation in videos," in *European Conference on Computer Vision*. Springer, 2020, pp. 158–177.

[32] M. Keuper, B. Andres, and T. Brox, "Motion trajectory segmentation via minimum cost multicuts," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3271–3279.

[33] M. M. Deza and M. Laurent, *Geometry of cuts and metrics*. Springer, 2009, vol. 15.

[34] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres, "Efficient decomposition of image and mesh graphs by lifted multicuts," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1751–1759.

[35] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. Hoi, and H. Ling, "Learning unsupervised video object segmentation through visual attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3064–3074.

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 21–29.

[37] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9236–9245.

[38] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention

siamese networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3623–3632.

[39] Y. Zhou, X. Xu, F. Shen, X. Zhu, and H. T. Shen, "Flow-edge guided unsupervised video object segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[40] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 724–732.

[41] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[42] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3000–3009.

[43] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[44] S. D. Jain, B. Xiong, and K. Grauman, "Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2017, pp. 2117–2126.

[45] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. H. Torr, "Anchor diffusion for unsupervised video object segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 931–940.