Self-supervised Video Object Segmentation by Motion Grouping

Charig Yang Hala Lamdouar Erika Lu Andrew Zisserman Weidi Xie
Visual Geometry Group, University of Oxford
{charig,lamdouar,erika,az,weidi}@robots.ox.ac.uk
https://charigyang.github.io/motiongroup/

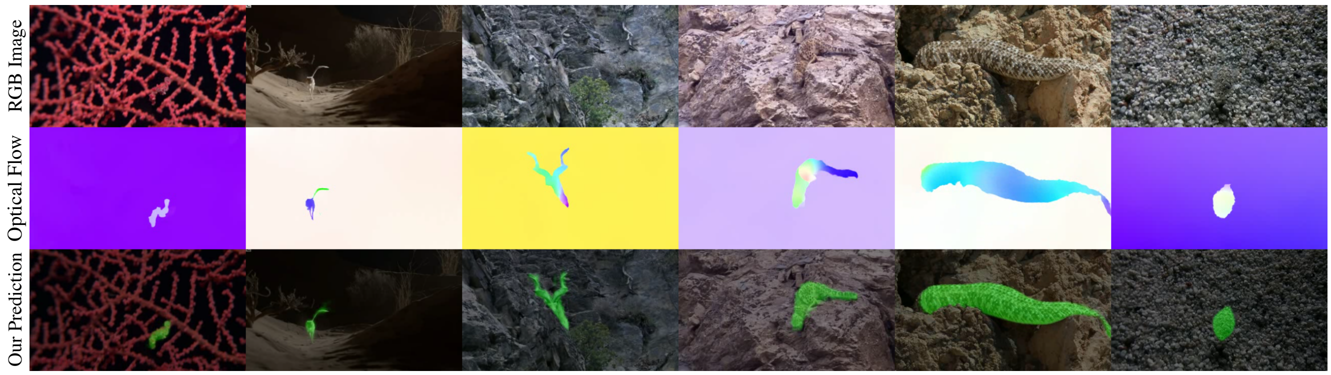

Figure 1: **Segmenting camouflaged animals.** Motion plays a critical role in augmenting the capability of our visual system for perceptual grouping in complex scenes – for example, in these sequences (MoCA dataset [39]), the visual appearance (RGB images) is clearly uninformative. In this paper, we propose a self-supervised approach to segment objects using *only* motion, *i.e*. optical flow. From top to bottom rows, we show the video frames, optical flow between consecutive frames, and the segmentation produced by our approach.

Abstract

Animals have evolved highly functional visual systems to understand motion, assisting perception even under complex environments. In this paper, we work towards developing a computer vision system able to segment objects by exploiting motion cues, i.e. *motion segmentation. To achieve this, we introduce a simple variant of the Transformer to segment optical flow frames into primary objects and the background, which can be trained in a self-supervised manner,* i.e. *without using any manual annotations. Despite using only optical flow, and no appearance information, as input, our approach achieves superior results compared to previous state-of-the-art self-supervised methods on public benchmarks (DAVIS2016, SegTrackv2, FBMS59), while being an order of magnitude faster. On a challenging camouflage dataset (MoCA), we significantly outperform other self-supervised approaches, and are competitive with the top supervised approach, highlighting the importance of motion cues and the potential bias towards appearance in existing video segmentation models.*

1. Introduction

When looking around the world, we effortlessly perceive a complex scene as a set of distinct objects. This phenomenon is referred to as *perceptual grouping* – the process of organizing the incoming visual information – and is usually considered a fundamental cognitive ability that enables understanding and interacting with the world efficiently. How do we accomplish such a remarkable perceptual achievement, given that the visual input is, in a sense, just a spatial distribution of variously colored individual points/pixels? In 1923, Wertheimer [80] first introduced the *Gestalt principles* with the goal of formulating the underlying causes by which sensory data is organized into groups, or Gestalten. The principles are much like heuristics with "a bag of tricks" [58] that the visual system may exploit for grouping, for example, proximity, similarity, closure, continuation, common fate, *etc*.

In computer vision, *perceptual grouping* is often closely related to the problem of segmentation, *i.e*. extracting the objects with arbitrary shape (pixel-wise labels) from cluttered scenes. In the recent literature of semantic or instance segmentation, tremendous progress has been made by training deep neural networks on image or video datasets. While it is exciting to see machines with the ability to detect, segment, and classify objects in images or video frames, training such segmentation models through supervised learning requires massive human annotation, and consequently limits their scalability. Even more importantly, the assumption that objects can be well-identified by their appearance alone in static frames is often an oversimplification – objects are not always visually distinguishable from their background environment. For instance when trying to discover camouflaged animals/objects from the background (Figure 1), extra cues, such as motion or sound, are usually required.

Among the numerous cues, motion is usually simple to obtain as it can be generated from unlabeled videos. In this paper, we aim to exploit such cues for object segmentation in a self-supervised manner, *i.e*. *zero* human annotation is required for training. At a high level, we aim to exploit the common fate principle, with the basic assumption being that **elements tend to be perceived as a group if they move in the same direction at the same rate (have similar optical flow)**. Specifically, we tackle the problem by training a generative model that decomposes the optical flow into foreground (object) and background layers, describing each as a homogeneous field, with discontinuities occurring only between layers. We adopt a variant of the Transformer [72], with the self-attention being replaced by slot attention [44], where iterative grouping and binding have been built into the architecture. With some critical architectural changes, we show that pixels undergoing similar motion are grouped together and assigned to the same layer.

To summarize, we make the following contributions: *first*, we introduce a simple architecture for video object segmentation by exploiting motions, using only optical flow as input. *Second*, we propose a self-supervised proxy task that is used to train the architecture without any manual supervision. To validate these contributions, we conduct thorough ablation studies on the components that are key to the success of our architecture, such as a consistency loss on optical flow computed from various frame gaps. We evaluate the proposed architecture on public benchmarks (DAVIS2016 [55], SegTrackv2 [40], and FBMS59 [52]), outperforming previous state-of-the-art self-supervised models. Moreover, we also evaluate on a camouflage dataset (MoCA [39]), demonstrating a significant performance improvement over the other self-supervised approaches, with comparable performance to the best supervised approach, highlighting the importance of motion cues, and the potential bias towards visual appearance in existing video segmentation models.

2. Related Work

Video object segmentation has been a longstanding task in computer vision, which involves assigning pixels (or edges) of an image into groups (e.g. objects). In recent literature [4, 9, 11, 15, 23, 24, 30, 33, 34, 37, 38, 47, 50, 51, 53, 54, 56, 56, 70, 73, 74, 75, 76, 84, 87], two protocols have attracted increasing interest from the vision community, namely, semi-supervised video object segmentation (**semi-supervised VOS**), and unsupervised video object segmentation (**unsupervised VOS**). The former aims to re-localize one or multiple targets that are specified in the first frame of a video with pixel-wise masks, and the latter considers automatically segmenting the object of interest (usually the most salient one) from the background in a video sequence. Despite being called **unsupervised VOS**, in practice, the popular methods to address such problems extensively rely on supervised training, for example, by using two-stream networks [15, 30, 54, 70] trained on large-scale external datasets. As an alternative, in this work, we consider a *completely unsupervised* approach, where no manual annotation is used for training whatsoever.

Motion segmentation shares some similarity with unsupervised VOS, but focuses on discovering *moving* objects. In the literature, [9, 34, 51, 62, 83] consider clustering the pixels with similar motion patterns; [15, 69, 70] train deep networks to map the motions to segmentation masks. Another line of work has tackled the problem by explicitly leveraging the independence of motion between the moving object and its background. For instance, [86] proposes an adversarial setting, where a generator is trained to produce masks, altering the input flow, such that the inpainter fails to estimate the missing information. In [5, 6, 39], the authors propose to highlight the independently moving object by compensating for the background motion, either by registering consecutive frames, or explicitly estimating camera motion. In constrained scenarios, such as autonomous driving, [60] proposes to jointly optimize depth, camera motion, optical flow and motion segmentation.

Optical flow computation is one of the fundamental tasks in computer vision. Deep learning methods allow efficient computation of optical flow, both in training on synthetic data [67, 68], or learning with photometric loss in self-supervised [42, 43] setting. In practise, flow has been useful for a wide range of problems, for example, pose estimation [18], representation learning [29, 49], segmentation [9], and occasionally even used in lieu of appearance cues (RGB images) for tracking [61].

Transformer architectures have proven extremely adept at modelling long-term relationships within an input sequence via attention mechanisms. Originally used for language tasks [8, 17, 72], they have since been adapted to solve popular computer vision problems such as image classification [19], generation [13, 59], video understanding [3, 26, 79], object detection [12], and zero-shot classification [57]. In this work, we take inspiration from a specific variant of self-attention, namely slot attention [44], which was demonstrated to be effective for learning object-centric representations on synthetic data, *e.g*. CLEVR [31].

Layered representations were originally proposed by Wang and Adelson [77] to represent a video as a composition of layers with simpler motions. Since then, layered representations have been widely adopted in computer vision [7, 32, 36, 85, 90], often to estimate optical flow [65, 66, 81, 82]. More recently, deep learning-based layer decomposition methods have been used to infer depth for novel view synthesis [64, 88], separate reflections and other semi-transparent effects [1, 2, 25, 45], or perform foreground/background estimation [25]. These works operate on RGB inputs and produce RGB layers, whereas we propose a layered decomposition of optical flow inputs for unsupervised moving object discovery.

Object-centric representations interpret scenes with "objects" as the basic building blocks (instead of individual pixels), which is considered an essential step towards human-level generalization. There is a rich literature on this topic, for example, IODINE [28] uses iterative variational inference to infer a set of latent variables recurrently, with each representing one object in an image. Similarly, MONet [10] and GENESIS [20] also adopt multiple encoding-decoding steps. In contrast, [44] proposes Slot Attention, which enables single step encoding-decoding with iterative attention. However, all works mentioned above have only shown applications for synthetic datasets, *e.g*. CLEVR [31]. In this paper, we are the first to demonstrate its use for object segmentation of realistic videos by exploiting motion, where the challenging nuances in visual appearance (*e.g*. the complex background textures) have been removed.

3. Method

Our goal is to take an input optical flow frame and predict a segment containing the moving object. We propose to train this model in a self-supervised manner, with an autoencoder-type framework. Specifically, our model outputs two layers: one representing the background, and the other for one or more moving objects in the foreground, as well as their opacity layers (weighted masks). Formally:

$$\{\hat{I}^i_{t\to t+n}, \alpha^i_{t\to t+n}\}_{i=1}^N = \Phi(I_{t\to t+n}) \tag{1}$$

where $I_{t\to t+n}$ refers to the t to $t+n$ input flow (backward flow when $n < 0$), $\Phi(\cdot)$ is the parametrized model, $\hat{I}^i_{t\to t+n}$ is the ith layer reconstruction, $\alpha^i_{t\to t+n}$ is its mask, and $N = 2$ is the number of layers (foreground and background). These layers can then be composited linearly to reconstruct the input image $I_{t\to t+n}$:

$$\hat{I}_{t\to t+n} = \sum_{i=1}^{N} \alpha^i_{t\to t+n}\hat{I}^i_{t\to t+n} \tag{2}$$

3.1. Flow Segmentation Architecture

For simplicity, we first consider the case of a single flow field as input (depicted in the top part of Figure 2). The entire model consists of three components: (1) a CNN encoder to extract a compact feature representation, (2) an iterative binding module with learnable queries that plays a similar role as soft clustering, *i.e*. assigning each pixel to one of motion groups, and (3) a CNN decoder that individually decodes each query to full resolution layer outputs (where thresholding the alpha channel yields the predicted segment).

CNN encoder. We first pass the precomputed optical flow between two frames, $I_{t\to t+n} \in \mathcal{R}^{3\times H_0\times W_0}$, to a CNN encoder Φ_{enc}, which outputs a lower-resolution feature map:

$$F_{t\to t+n} = \Phi_{\text{enc}}(I_{t\to t+n}) \in \mathcal{R}^{D\times H\times W} \tag{3}$$

where H_0, W_0 and H, W refer to the spatial dimensions of the input and output feature maps respectively. Note that, we convert the flow into a three-channel image using the traditional method in the optical flow literature [67].

Iterative binding. The iterative binding module Φ_{bind} aims to group image regions into single entities based on their similarities in motion, *i.e*. pixels moving in the same direction at the same rate should be grouped together. Intuitively, such a binding process requires a data-dependent parameter updating mechanism, iteratively enriching the model, gradually including more pixels undergoing similar motions.

To accomplish this task, we adopt a simple variant of slot attention [44], where instead of Gaussian-initialized slots, we use learnable query vectors. Slot attention has recently shown remarkable performance for object-centric representation learning, where the query vectors compete to explain parts of the inputs via a softmax-based attention mechanism, and the representations of these slots are iteratively updated with a recurrent update function. In our case of motion segmentation, ideally, the final representation in each query vector separately encodes the moving object or the background, which can then be decoded and combined to reconstruct the input flow fields.

Formally, our inputs to Φ_{bind} are feature maps $F_{t\to t+n}$ and two learnable queries (representing foreground and background) $Q \in \mathcal{R}^{D\times 2}$. Learnable spatial positional encodings are summed with $F_{t\to t+n}$; with some abuse of notation, we still refer to this sum as $F_{t\to t+n}$. We use *three* different linear transformations to generate the $query$, key and $value$: $q \in \mathcal{R}^{D\times 2}$, $k, v \in \mathcal{R}^{D\times HW}$,

$$q,\ k,\ v = W^Q{\cdot}Q,\ W^K{\cdot}F_{t\to t+n},\ W^V{\cdot}F_{t\to t+n} \tag{4}$$

where $W^Q, W^K, W^V \in \mathcal{R}^{D\times D}$.

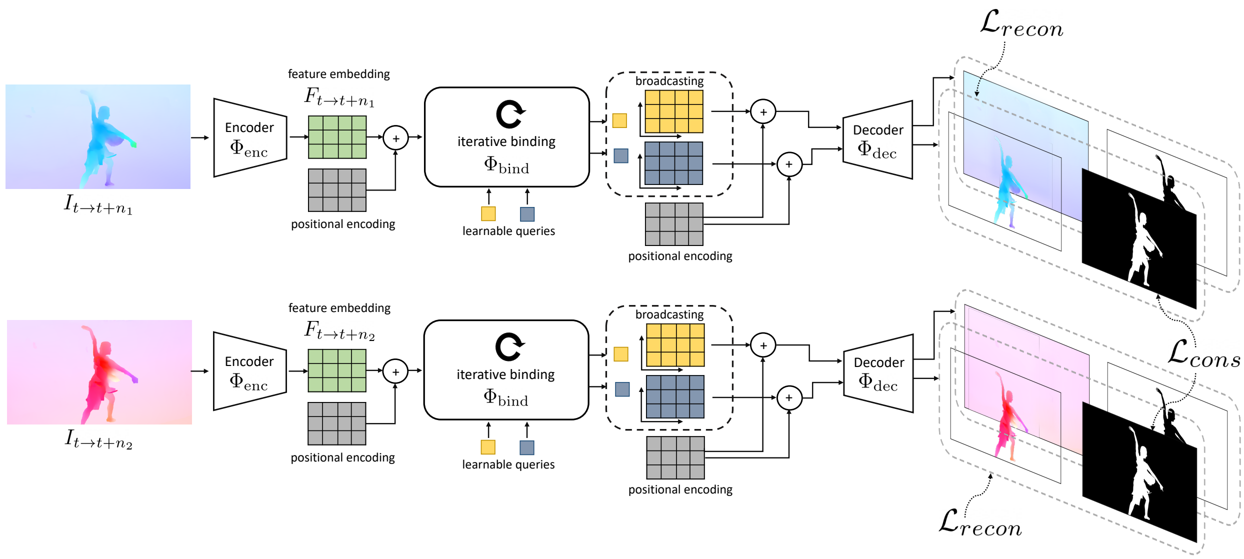

Figure 2: **Pipeline.** Our model takes optical flow $I_{t\to t+n}$ as input, and outputs a set of reconstruction and opacity layers. Specifically, it consists of three components: feature encoding, iterative binding, and decoding to layers, which are combined to reconstruct the input flow. To resolve motion ambiguities (small motion), or noise in optical flow, consistency between two flow fields computed under different frame gaps is enforced during training. At inference time, only the top half of the figure is used to predict masks from a single-step flow.

In contrast to the standard Transformer [72], the coefficients in slot attention are normalized over all slots. This choice of normalization introduces competition between the slots to explain parts of the input, and ensures each pixel is assigned to a query vector:

$$\text{attn}_{i,j} := \frac{e^{M_{i,j}}}{\sum_l e^{M_{i,l}}} \tag{5}$$
$$M := \frac{1}{\sqrt{D}} k^T \cdot q, \quad \text{attn} \in \mathcal{R}^{HW\times 2}$$

To aggregate the input values to their assigned query slot, a weighted mean is used as follows:

$$U := v \cdot A \quad \in \mathcal{R}^{D\times 2} \tag{6}$$
$$\text{where,} \quad A_{i,j} := \frac{\text{attn}_{i,j}}{\sum_l \text{attn}_{l,j}}$$

To maintain a smooth update of the query slots Q, the aggregated vectors U are fed into a recurrent function, parametrized with Gated Recurrent Units (GRU),

$$Q := \text{GRU}(\text{inputs} = U, \text{states} = Q) \tag{7}$$

This whole binding process is then iterated T times. The pseudocode can be found in the Supplementary Material.

CNN decoder. The CNN decoder Φ_{dec} individually decodes each of the slots to outputs of original resolution ($\{\hat{I}^i_{t\to t+n}, \alpha^i_{t\to t+n}\} \in \mathcal{R}^{4\times H_0 \times W_0}$), which includes an (unnormalized) single-channel alpha mask and the reconstructed flow fields. Specifically, the input to the decoder is the slot vector broadcasted onto a 2D grid augmented with a learnable spatial positional encoding.

Reconstruction. Once each slot has been decoded, we apply softmax to the alpha masks across the slot dimension, and use them as mixture weights to obtain the reconstruction $\hat{I}_{t\to t+n}$ (Eq. 2). Our reconstruction loss is an L2 loss between the input and reconstructed flow,

$$\mathcal{L}_{recon} = \frac{1}{\Omega} \sum_{p\in\Omega} |I_{t\to t+n}(p) - \hat{I}_{t\to t+n}(p)|^2 \tag{8}$$

where p is the pixel index, and Ω is the entire spatial grid.

Entropy regularization. We impose a pixel-wise entropy regularisation on inferred masks:

$$\mathcal{L}_{entr} = \frac{1}{\Omega} \sum_{p\in\Omega} (-\alpha^1_{t\to t+n}(p) \log \alpha^1_{t\to t+n}(p) \tag{9}$$
$$-\alpha^0_{t\to t+n}(p) \log \alpha^0_{t\to t+n}(p))$$

This loss is zero when the alpha channels are one-hot, and maximum when they are of equal probability. Intuitively, this helps encourage the masks to be binary, which aligns with our goal in obtaining segmentation masks.

Instance normalisation. In the case of motion segmentation, objects can only be detected if they undergo an independent motion from the camera; thus previous work attempts to compensate for camera motion [5, 39]. We are inspired by these ideas, but instead of explicitly estimating homography or camera motion, we take a poor-man's approach by simply using Instance Normalisation (IN) [71] in the CNN encoder and decoder, which normalizes each channel of the training sample independently. Intuitively, the *mean* activation tends to be dominated by the motions in the large homogeneous region, which is usually the background. This normalization, in combination with ReLU activations, helps gradually separate the background motion from the foreground motions. This is experimentally shown in Section 5.1

3.2. Self-supervised Temporal Consistency Loss

The segmentation computed for the current frame should be identical irrespective of whether the 'second' frame is consecutive, or earlier or later in time. We harness this constraint to form a self-supervised temporal consistency loss by first defining a set of 'second' frames, and then requiring consistency between their pairwise predictions. We describe the set first, followed by the loss.

Multi-step flow. As objects may be static for some frames, we make our predictions more robust by leveraging observations from multiple timesteps. We consider the flow fields computed from various temporal gaps as an input set, *i.e.* $\{I_{t\to t+n_1}, I_{t\to t+n_2}\}$, $n_1, n_2 \in \{-2,-1,1,2\}$, and use a permutation invariant consistency loss to encourage the model to predict the same foreground/background segmentation for all flow fields in the set.

Consistency loss. We randomly sample two flow fields from the input set and pass them through the model ($\Phi(\cdot)$), outputting the flow reconstruction and alpha masks for each. As the reconstruction loss is commutative, it is not guaranteed that the same slot will always output the background layer; therefore, we use a permutation-invariant consistency loss, *i.e.* only backpropagating through the lowest-error permutation:

$$\begin{aligned}\mathcal{L}_{cons} = \frac{1}{\Omega}\min(&\sum_{p\in\Omega}|\alpha^1_{t\to t+n_1}(p) - \alpha^1_{t\to t+n_2}(p)|^2, \\ &\sum_{p\in\Omega}|\alpha^1_{t\to t+n_1}(p) - \alpha^0_{t\to t+n_2}(p)|^2)\end{aligned}$$

Note that, this consistency enforcement only occurs during training. At inference time, a single-step flow is used, as shown in the top half of Figure 2.

Total loss. The total loss for training the architecture is:

$$\mathcal{L}_{total} = \gamma_r\mathcal{L}_{recon} + \gamma_c\mathcal{L}_{cons} + \gamma_e\mathcal{L}_{entr} \tag{10}$$

we use $\gamma_r = 10^2$, $\gamma_c = 10^{-2}$ and $\gamma_e = 10^{-2}$, but we found the model to be fairly robust to these hyperparameters.

3.3. Discussion

Differences from slot attention. Slot attention was originally introduced for self-supervised object segmentation for RGB images [44], and its usefulness was demonstrated on synthetic data (CLEVR [31]), where objects are made of primitive shapes with simple textures. However, this assumption is unlikely to hold in the case of natural images or videos, making it challenging to generalize such object-centric representations.

In this work, we build on the insight that although objects in images may not be naturally textureless, their motions typically are. Hence, we develop the self-supervised object segmentation model by exploiting their optical flows, where the nuance in visual appearance is discarded, thus not restricted to simple synthetic cases. As an initial trial, we experimented with the same setting as [44], where query vectors are sampled from a Gaussian distribution; however, we were unable to train it. Instead, we use learnable embeddings here, which we highlight as one of the architectural changes critical to our model's success. Other critical changes include instance normalization and temporal consistency, which we demonstrate in ablations in Section 5.1.

Why does it work for motion segmentation? The proposed idea can be seen as training a generative model to segment the flow fields. With the layered formulation, reconstruction is limited to be a simple *linear* composition of layer-wise flow, decoded from a single slot vector.

Conceptually, this design has effectively introduced a representational bottleneck, encouraging each slot vector to represent minimal information, *i.e.* homogeneous motion, and with minimal redundancy (mutual information) between slots. All these properties make such an architecture well-suited to the task of segmenting objects undergoing independent motions.

4. Experimental Setup

4.1. Datasets

DAVIS2016 [55] contains a total of 50 sequences (30 for training and 20 for validation), depicting diverse moving objects such as animals, people, and cars. The dataset contains 3455 1080p frames with pixel-wise annotations at 480p for the predominantly moving object.

SegTrackv2 [40] contains 14 sequences and 976 annotated frames. Each sequence contains 1-6 moving objects, and presents challenges including motion blur, appearance change, complex deformation, occlusion, slow motion, and interacting objects.

FBMS59 [52] consists of 59 sequences and 720 annotated frames (every 20th frame is annotated), which vary greatly in image resolution. Sequences involve multiple moving objects, some of which may be static for periods of time.

Moving Camouflaged Animals (MoCA) [39] contains 141 HD video sequences, depicting 67 kinds of camouflaged animals moving in natural scenes. Both temporal and spatial annotations are provided in the form of tight bounding boxes for every 5th frame. Using the provided motion labels (locomotion, deformation, static), we filter out videos with predominantly no locomotion, resulting in 88 video sequences and 4803 frames.

4.2. Evaluation Metrics

Segmentation (Jaccard). For DAVIS2016, SegTrackv2 and FBMS59, pixelwise segmentation is provided; thus we report the standard metric, region similarity ($\mathcal{J}$), computing the mean over the test set. For FBMS59 and SegTrackv2, we follow the common practice [30, 86] and combine multiple objects as one single foreground.

Localization (Jaccard & Success Rate). As the MoCA dataset provides only bounding box annotations, we evaluate for the detection task and report results in the form of detection success rate [21, 41], for varying IoU thresholds ($\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$).

4.3. Implementation Details

We evaluate three different approaches for computing optical flow, namely, PWC-Net [67], RAFT [68] and ARFlow [42]; the first two are supervised, while the latter is self-supervised. We extract the optical flow at the original resolution of the image pairs, with the frame gaps $n \in \{-2, -1, 1, 2\}$ for all datasets, except for FBMS59, where we use $n \in \{-6, -3, 3, 6\}$ to compensate for small motion. To generate inputs to the network for training, the flows are resized to 128×224 (and scaled accordingly), converted to 3-channel images with the standard visualization used for optical flow, and normalized to $[-1, 1]$.

In the iterative binding module ($\Phi_{\text{bind}}(\cdot)$), we use two learnable query vectors (as we consider the case of segmenting a single moving object from the background), and choose $T = 5$ iterations (as explained in Section 3.1). We adopt a simple VGG-style network for the CNN encoder and decoder with instance normalization. We train with a batch size of 64 images and use the Adam optimizer [35] with an initial learning rate of 5×10^{-4}, decreasing every 80k iterations. The exact architecture description and training schedule can be found in the Supplementary Material.

5. Results

In this section, we compare primarily with a top-performing approach trained without manual annotations – Contextual Information Separation (CIS [86]). However, as the architecture, input resolution, modality and post-processing are all different, we try our best to conduct the comparison as fairly as possible. Note that benchmarks are evaluated at full resolution by simply upsampling the predicted masks.

5.1. Ablation Studies

We conduct all ablation studies on DAVIS2016, and vary one variable each time, as shown in Table 1.

Choice of optical flow algorithm. With the same flow extraction method (PWC-Net), our proposed model (Ours-A) outperforms CIS by about 4.5 points on mean Jaccard ($\mathcal{J}$), and using improved optical flow (RAFT) provides further performance gains. We therefore use RAFT from hereon.

Instance normalization and grouping. We observe two phenomena: *first*, when holding constant on the number of grouping iterations T (3 or 5), models trained with instance normalization perform consistently better; *second*, iterative grouping with $T = 5$ is better than that trained with $T = 3$. However, at $T = 8$, the model did not converge in the same number of training steps, and thus we do not include it in the table. For the remainder of the experiments, we use instance normalization and $T = 5$.

Consistency and entropy regularization. While comparing Ours-B and Ours-I, we observe that the performance degrades significantly without the temporal consistency loss, and that the entropy regularization is also important, as shown by Ours-B and Ours-H.

5.2. Comparison with State-of-the-art

We show our results in Table 2. On DAVIS2016, we improve upon the state-of-the-art for unsupervised methods (CIS) by a large margin (+9.1%). As shown in Figure 3, despite not using any pixel-level annotations during training, our method is nearing the performance of supervised models trained on thousands of images.

In addition, we argue that, motion segmentation in realistic scenarios, *e.g.* by predator or prey, is likely to require fast processing. Our model operates on small resolution (potentially sacrificing some accuracy) with over 80fps. Our method's efficiency gain mainly comes from two sources: first, our model is a lightweight VGG-style network with only 4.77**M** parameters; second, we disregard any post-processing used in previous approaches, e.g. averaging the prediction across multiple flow steps, across multiple crops, temporal smoothing, or CRFs, which cost over 10s in total.

For SegTrackv2 and FBMS59, they occasionally include multiple objects in a single video, and only a subset of them are moving, making it challenging to spot all objects using flow-only input, but we achieve competitive performance nonetheless. We discuss this limitation below.

5.3. Camouflage Breaking

In addition, we also benchmark the model on camouflaged object detection on MoCA dataset, where visual cues are often less effective than motion cues. To compare fairly with CIS [86], we use the code and model released by the authors, and fine-tune their model on MoCA in a

Model	Flow	IN	T	$\mathcal{L}_e$	$\mathcal{L}_c$	DAVIS ($\mathcal{J}\uparrow$)
CIS [86]	PWC-Net	–	–	–	–	59.2
Ours-A	PWC-Net	✓	5	✓	✓	63.7
Ours-B	**RAFT**	✓	**5**	✓	✓	**68.3**
Ours-C	ARFlow	✓	5	✓	✓	53.2
Ours-D	RAFT	✓	3	✓	✓	65.8
Ours-E	RAFT	✗	3	✓	✓	63.3
Ours-F	RAFT	✗	5	✓	✓	64.5
Ours-G	RAFT	✓	5	✗	✗	48.0
Ours-H	RAFT	✓	5	✗	✓	60.3
Ours-I	RAFT	✓	5	✓	✗	51.2

Table 1: **Ablation studies** on flow extraction methods, instance normalization (IN), grouping iterations (T), entropy regularization ($\mathcal{L}_e$) and set consistency ($\mathcal{L}_c$).

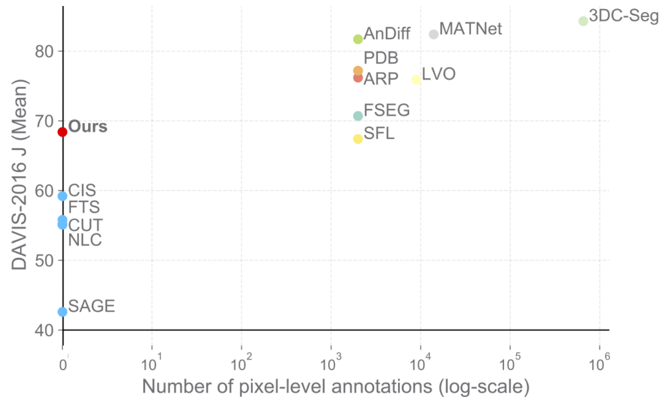

Figure 3: **Comparison on DAVIS2016.** Note that, supervised approaches may use ImageNet pretraining [16], but here we only count images with pixel-wise annotations.

Model	Sup.	RGB	Flow	Res.	DAVIS16 ($\mathcal{J}\uparrow$)	STv2 ($\mathcal{J}\uparrow$)	FBMS59 ($\mathcal{J}\uparrow$)	Runtime (sec $\downarrow$)
SAGE [78]	✗	✓	✓	–	42.6	57.6	**61.2**	0.9s
NLC [22]	✗	✓	✓	–	55.1	**67.2**	51.5	11s
CUT [34]	✗	✓	✓	–	55.2	54.3	57.2	103s
FTS [54]	✗	✓	✓	–	55.8	47.8	47.7	0.5s
CIS [86]	✗	✓	✓	192×384	59.2 (71.5)	45.6 (62.0)	36.8 (63.5)	0.1s (11s)
Ours	✗	✗	✓	128×224	**68.3**	58.6	53.1	**0.012s**
SFL [14]	✓	✓	✓	854×480	67.4	–	–	7.9s
FSEG [30]	✓	✓	✓	854×480	70.7	61.4	68.4	–
LVO [70]	✓	✓	✓	–	75.9	57.3	65.1	–
ARP [63]	✓	✓	✓	–	76.2	57.2	59.8	74.5s
COSNet [46]	✓	✓	✗	473×473	80.5	–	75.6	–
MATNet [89]	✓	✓	✓	473×473	82.4	–	–	0.55s
3DC-Seg [48]	✓	✓	✓	854×480	84.3	–	–	0.84s

Table 2: **Full comparison on moving object segmentation** (unsupervised video segmentation). We consider three popular datasets, DAVIS2016, SegTrack-v2 (STv2), and FBMS59. Models above the horizontal dividing line are trained without using *any* manual annotation, while models below require ground truth annotations at training time. Numbers in parentheses denote the additional usage of significant post-processing, *e.g.* multi-step flow, multi-crop, temporal smoothing, CRFs. Runtime excludes optical flow computation.

self-supervised manner. We convert the output segmentation mask into a bounding box by drawing a bounding box around the largest connected region in the predicted mask.

We report quantitative results in Table 3 and show qualitative results in Figure 4. Our model significantly outperforms CIS (14% when allowing no post-processing), previous supervised approaches *e.g.* COD [39] (18.5% on Jaccard), and even COSNet [46] (among the top supervised approaches on DAVIS). We conjecture that COSNet's weaker performance is due to its sole reliance on visual appearance (which is distracting for the MoCA data) rather than using motion inputs. This is particularly interesting, as it clearly indicates that no single information cue is able to do the task perfectly, echoing the two-stream hypothesis [27] that both appearance and motion are essential to visual systems.

5.4. Limitations

Despite showing remarkable improvements on motion segmentation in accuracy and runtime, we note the following limitations of the proposed approach (shown in Figure 4) and treat them as future work: *first*, the existing benchmarks are mostly limited to motion segmentation into foreground and background, thus, we choose to use two slots in this paper; however, in real scenarios, videos may contain multiple independently moving objects, which the current model will assign to a single layer. It may be desirable to further separate these objects into different layers. *Second*, we have only explored motion (optical flow)

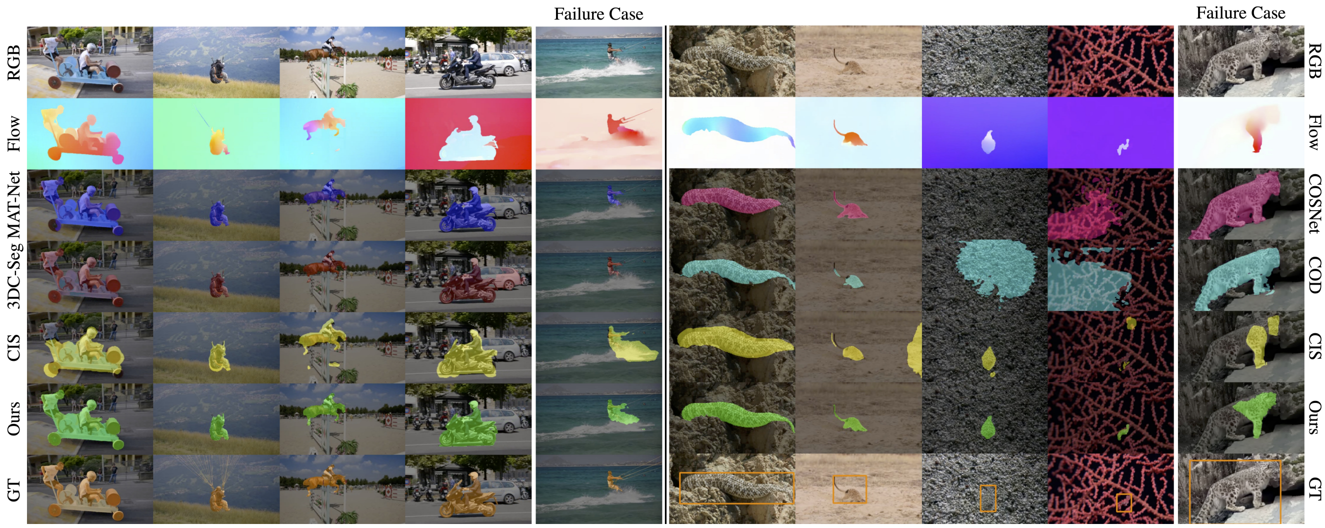

Figure 4: **Qualitative results.** On DAVIS2016 (left), our method is able to segment a variety of challenging objects, often on-par with top supervised approaches. On MoCA (right), our model is able to accurately segment well-camouflaged objects even when previous supervised methods fail completely (3rd, 4th columns). We show a failure case (left) where the splash created by the person is incorrectly included in our predicted segment, and another failure case (right) where the animal is only partially moving and thus partially segmented.

					Success Rate					
Model	Sup.	RGB	Flow	$\mathcal{J}\uparrow$	$\tau=0.5$	$\tau=0.6$	$\tau=0.7$	$\tau=0.8$	$\tau=0.9$	SR_{mean}
COD [39]	✓	✗	✓	44.9	0.414	0.330	0.235	0.140	0.059	0.236
COD (two-stream) [39]	✓	✓	✓	55.3	0.602	0.523	0.413	0.267	0.088	0.379
COSNet [46]	✓	✓	✗	50.7	0.588	0.534	0.457	0.337	0.167	0.417
MATNet [89]	✓	✓	✓	**64.2**	**0.712**	**0.670**	**0.599**	**0.492**	**0.246**	**0.544**
CIS	✗	✓	✓	49.4	0.556	0.463	0.329	0.176	0.030	0.311
CIS (post-processing)	✗	✓	✓	54.1	0.631	0.542	0.399	0.210	0.033	0.363
Ours	✗	✗	✓	**63.4**	**0.742**	**0.654**	**0.524**	**0.351**	**0.147**	**0.484**

Table 3: **Comparison results on MoCA dataset.** We report the successful localization rate for various thresholds τ (see Section 4.2). Both CIS and Ours were pre-trained on DAVIS and finetuned on MoCA in a self-supervised manner. Our method achieves comparable Jaccard ($\mathcal{J}$) to MATNet (2nd best model on DAVIS), without using RGB inputs and without any manual annotation for training.

as input, which significantly limits the model in segmenting objects when flow is uninformative or incomplete (as in Figure 4, right); however, the self-supervised video object segmentation objective is applicable also to a two-stream approach, and so RGB could be incorporated. *Third*, the current method may fail when optical flow is noisy or low-quality (Figure 4, left); jointly optimizing flow and segmentation is a possible way forward in this case.

6. Conclusion

In this paper, we present a self-supervised model for motion segmentation. The algorithm takes only flow as input, and is trained without any manual annotation, surpassing previous self-supervised methods on public benchmarks such as DAVIS2016, and narrowing the gap with supervised methods. On the more challenging camouflage dataset (MoCA), our model actually compares favourably to the top approaches in video object segmentation that are trained with heavy supervision. As computation power grows and more high quality videos become available, we believe that self-supervised learning algorithms can serve as a strong competitor to the supervised counterparts for their scalability and generalizability.

7. Acknowledgements

This research is supported by Google-DeepMind Studentship, UK EPSRC CDT in AIMS, Schlumberger Studentship, a Royal Society Research Professorship, and the UK EPSRC Programme Grant Visual AI (EP/T028572/1).

References

[1] Jean-Baptiste Alayrac, João Carreira, and Andrew Zisserman. The visual centrifuge: Model-free layered video representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] Jean-Baptiste Alayrac, Joao Carreira, Relja Arandjelovic, and Andrew Zisserman. Controllable attention for structured layered video decomposition. In *Proceedings of the International Conference on Computer Vision*, 2019.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning*, 2021.

[4] Pia Bideau and Erik Learned-Miller. A detailed rubric for motion segmentation. *arXiv preprint arXiv:1610.10033*, 2016.

[5] Pia Bideau and Erik Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *Proceedings of the European Conference on Computer Vision*, 2016.

[6] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moa-net: self-supervised motion segmentation. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018.

[7] Gabriel J Brostow and Irfan A Essa. Motion based decompositing of video. In *Proceedings of the International Conference on Computer Vision*, 1999.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[9] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the European Conference on Computer Vision*, 2010.

[10] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[11] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, 2020.

[13] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the International Conference on Machine Learning*, 2020.

[14] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the International Conference on Computer Vision*, 2017.

[15] Achal Dave, Pavel Tokmakov, and Deva Ramanan. Towards segmenting anything that moves. In *Proceedings of the International Conference on Computer Vision Workshops*, 2019.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL Human Language Technologies*, 2019.

[18] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. *Advances in Neural Information Processing Systems*, 2019.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2021.

[20] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, , and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *Proceedings of the International Conference on Learning Representations*, 2020.

[21] Mark Everingham, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[22] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *Proceedings of the British Machine Vision Conference*, 2014.

[23] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[24] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[25] Yossi Gandelsman, Assaf Shocher, and Michal Irani. "Double-DIP": Unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[26] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video Action Transformer Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[27] Melvyn A. Goodale and A. David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[28] Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey,

Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of the International Conference on Machine Learning*, 2019.
[29] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems*, 2020.
[30] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusion-seg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[31] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[32] Njegica Jojic and B.J. Frey. Learning flexible sprites in video layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
[33] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
[34] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proceedings of the International Conference on Computer Vision*, 2015.
[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.
[36] M. Pawan Kumar, Philip H. S. Torr, and Andrew Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76:301–319, 2008.
[37] Zihang Lai, Erika Lu, and Weidi Xie. MAST: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
[38] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *Proceedings of the British Machine Vision Conference*, 2019.
[39] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.
[40] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the International Conference on Computer Vision*, 2013.
[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014.
[42] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
[43] Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu. Self-low: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
[44] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.
[45] Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T Freeman, and Michael Rubinstein. Layered neural rendering for retiming people in video. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques in Asia*, 2020.
[46] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
[47] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
[48] Sabarinath Mahadevan, Ali Athar, Aljoša Ošep, Sebastian Hennen, Laura Leal-Taixé, and Bastian Leibe. Making a case for 3d convolutions for object segmentation in videos. In *Proceedings of the British Machine Vision Conference*, 2020.
[49] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Cross pixel optical-flow similarity for self-supervised learning. In *Proceedings of the Asian Conference on Computer Vision*, 2018.
[50] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
[51] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *Proceedings of the International Conference on Computer Vision*, 2011.
[52] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014.
[53] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the International Conference on Computer Vision*, 2019.
[54] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the International Conference on Computer Vision*, 2013.
[55] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander 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.

[56] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021.

[58] VS Ramachandran. Guest editorial: The neurobiology of perception. *Perception*, 14(97):103, 1985.

[59] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning*, 2021.

[60] Anurag Ranjan, Varun Jampani, Lukas Balles, Deqing Sun, Kihwan Kim, Jonas Wulff, and Michael 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.

[61] Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proceedings of the European Conference on Computer Vision*, 2000.

[62] Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman. Object level grouping for video shots. In *Proceedings of the European Conference on Computer Vision*, 2004.

[63] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European Conference on Computer Vision*, 2018.

[64] Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[65] Deqing Sun, Erik B. Sudderth, and Michael J. Black. Layered segmentation and optical flow estimation over time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[66] Deqing Sun, Jonas Wulff, Erik B. Sudderth, Hanspeter Pfister, and Michael J. Black. A fully-connected layered model of foreground and background flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[67] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[68] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, 2020.

[69] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[70] Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari. Learning to segment moving objects. *International Journal of Computer Vision*, 2019.

[71] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[72] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[73] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[74] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *Proceedings of the British Machine Vision Conference*, 2017.

[75] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European Conference on Computer Vision*, 2018.

[76] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[77] John YA Wang and Edward H Adelson. Representing moving images with layers. *The IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[78] Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli. Saliency-aware video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):20–33, 2018.

[79] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[80] Max Wertheimer. Untersuchungen zur lehre von der gestalt. ii. *Psychologische forschung*, 4(1):301–350, 1923.

[81] Jonas Wulff and Michael J. Black. Modeling blurred video with layers. In *Proceedings of the International Conference on Computer Vision*, 2014.

[82] Jonas Wulff and Michael J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proceedings of the European Conference on Computer Vision*, 2015.

[83] Christopher Xie, Yu Xiang, Zaid Harchaoui, and Dieter Fox. Object discovery in videos as foreground motion clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[84] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *Proceedings of the European Conference on Computer Vision*, 2018.

[85] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T.

Freeman. A computational approach for obstruction-free photography. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2015.

[86] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[87] Zhao Yang, Qiang Wang, Luca Bertinetto, Song Bai, Weiming Hu, and Philip H.S. Torr. Anchor diffusion for unsupervised video object segmentation. In *Proceedings of the International Conference on Computer Vision*, 2019.

[88] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2018.

[89] Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao. Motion-attentive transition for zero-shot video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[90] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004.