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/

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. We introduce a simple variant of the Transformer to segment optical flow frames into primary objects and the background, and train the architecture in a self-supervised manner,* i.e. *without using any manual annotations. Despite using only optical flow as input, our approach achieves superior or comparable results to previous state-of-the-art self-supervised methods, while being an order of magnitude faster. We additionally evaluate on a challenging camouflage dataset (MoCA), significantly outperforming the other self-supervised approaches, and comparing favourably to the top supervised approach, highlighting the importance of motion cues, and the potential bias towards visual appearance in existing video segmentation models.*

1. Introduction

In computer vision, *perceptual grouping* is often closely related to the problem of segmentation. However, 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, such as in camouflage detection.

In this paper, we aim to exploit motion cues for object segmentation in a self-supervised manner. 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 [14], with the self-attention being replaced by slot attention [8], 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.

In terms of related work, our work focuses on demonstrating work on object-centric representations [8] towards video object segmentation [16], in particular motion segmentation. In our architecture, we also use a variant of Transformer [14] to create a layered representation [15].

2. Method

Our goal is to take an input optical flow field (computed using RAFT [12]) and predict a segment containing the moving object. We train this model in a self-supervised manner; specifically, we adopt an autoencoder-like framework. 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, we have:

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

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 $I_{t\to t+n}$:

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

2.1. Flow Segmentation Architecture

For simplicity, we first consider the case of a single flow field as input (depicted in the top half of Figure 1). The entire model consists of:

CNN Encoder. We first pass the precomputed optical flow field 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

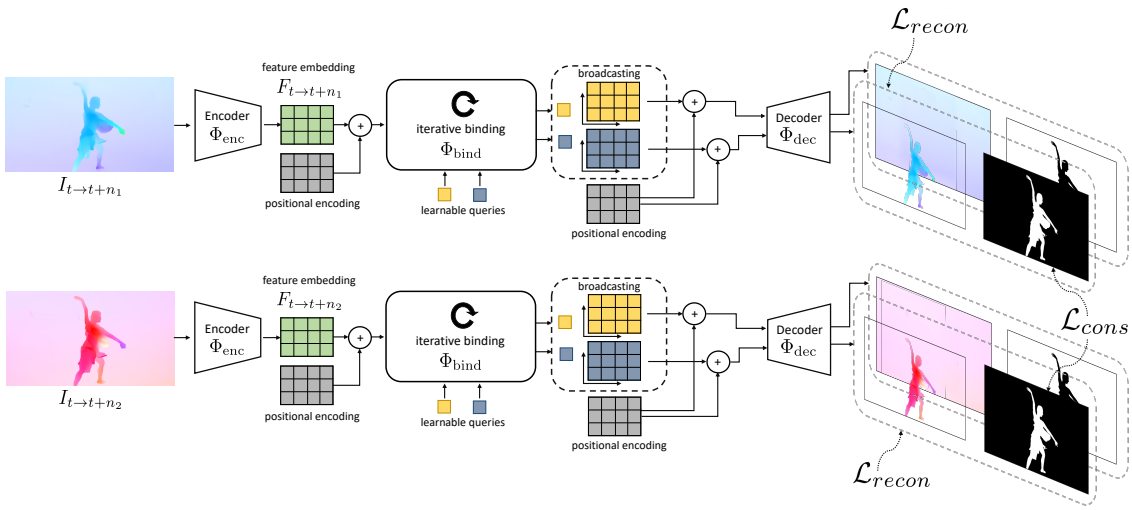

Figure 1. **Pipeline.** Our model takes optical flow $I_{t\to t+n}$ as input, and outputs a set of reconstruction and opacity layers. Specifically, it consists of three components: feature encoding, iterative binding, and decoding to layers, which are combined to reconstruct the input flow. To resolve 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.

feature map $F_{t\to t+n} \in \mathcal{R}^{D\times H\times W}$. Note that, we convert the optical flow field into a three-channel image according to the traditional method in the optical flow literature.

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 similar rate and direction should be grouped together. Intuitively, such a binding process requires a data-dependent parameter updating mechanism, iteratively adapting and enriching the model, gradually including more pixels undergoing similar motions.

To accomplish this task, we adopt a simple variant of slot attention [8], 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.

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,

$$\mathcal{L}_{recon} = \frac{1}{\Omega}\sum_{p\in\Omega} |I_{t\to t+n}(p) - \hat{I}_{t\to t+n}(p)|^2 \qquad (3)$$

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

Temporal Consistency. 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 consider the flow fields computed from various temporal gaps as an input set, *i.e.* $\{I_{t\to t+n_1}, I_{t\to t+n_2}\}$, $n_1, n_2 \in \{-2, -1, 1, 2\}$. We randomly sample two flow fields from this set and pass them through the model ($\Phi(\cdot)$), outputting the flow reconstruction and alpha masks for each. As the reconstruction is commutative, it is not guaranteed that the same slot will always output the background layer; therefore, we use a permutation-invariant consistency loss:

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

We refer readers to the arXiv version for more details (https://arxiv.org/abs/2104.07658).

3. Experimental Setup

We benchmark on four different datasets that are commonly used for unsupervised video object segmentation, namely DAVIS2016 [10], SegTrackv2 [7], FBMS59 [11]

Model	Sup.	RGB	Flow	Res.	DAVIS16 ($\mathcal{J}\uparrow$)	STv2 ($\mathcal{J}\uparrow$)	FBMS59 ($\mathcal{J}\uparrow$)	Runtime (sec $\downarrow$)
NLC [2]	✗	✓	✓	–	55.1	**67.2**	51.5	11s
CUT [5]	✗	✓	✓	–	55.2	54.3	**57.2**	103s
CIS [16]	✗	✓	✓	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 [1]	✓	✓	✓	854×480	67.4	–	–	7.9s
LVO [13]	✓	✓	✓	–	75.9	57.3	65.1	–
COSNet [9]	✓	✓	✗	473×473	80.5	–	75.6	–
MATNet [17]	✓	✓	✓	473×473	82.4	–	–	0.55s

Table 1. **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 are pre-trained on image or video segmentation datasets, *e.g.* DAVIS, YouTube-VOS. **Note:** the numbers in parentheses denote the additional usage of significant post-processing, *e.g.* multi-step flow, multi-crop, temporal smoothing, CRFs.

and Moving Camouflaged Animals (MoCA) [6].

We use two different evaluation metrics, depending on whether pixel-wise annotations or bounding boxes are provided. For DAVIS2016, SegTrackv2 and FBMS59, pixel-wise 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 [4, 16] and combine multiple objects as one single foreground. 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, for varying IoU thresholds ($\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$).

4. Results

4.1. Comparison with State-of-the-art

We show comparison results for different datasets in Table 1. On DAVIS2016, we improve upon the state-of-the-art for unsupervised methods (CIS) by a large margin ($+9.1\%$), approaching that of supervised methods. 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 at low-resolution (potentially sacrificing some accuracy) with over 80 fps. SegTrackv2 and FBMS59 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 still achieve competitive performance on these datasets.

4.2. Camouflage Breaking

In addition, we also benchmark the model on camouflaged object detection on the MoCA dataset, where visual cues are often less effective than motion cues. We report quantitative results in Table 2 and show qualitative results in Figure 2. Our model significantly outperforms CIS, and even COSNet [9] (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 MoCA) 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 [3] that both appearance and motion are essential to visual systems.

4.3. Limitations

We note the following limitations of the proposed approach and treat them as future work. *first*, we only use two slots in this paper *i.e.* foreground and background. In cases with multiple independently moving objects, it may be desirable to further separate these objects into different layers. *Second*, we have only explored motion-only (optical flow) as input, which significantly limits the model in segmenting objects when flow is uninformative or incomplete (as in Fig. 2, 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 (Fig. 2, left); jointly optimizing flow and segmentation is a possible way forward in this case.

5. 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, 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.

Acknowledgements. This research is supported by Google-DeepMind Studentship, UK EPSRC CDT in AIMS, Schlumberger Studentship, and the UK EPSRC Programme

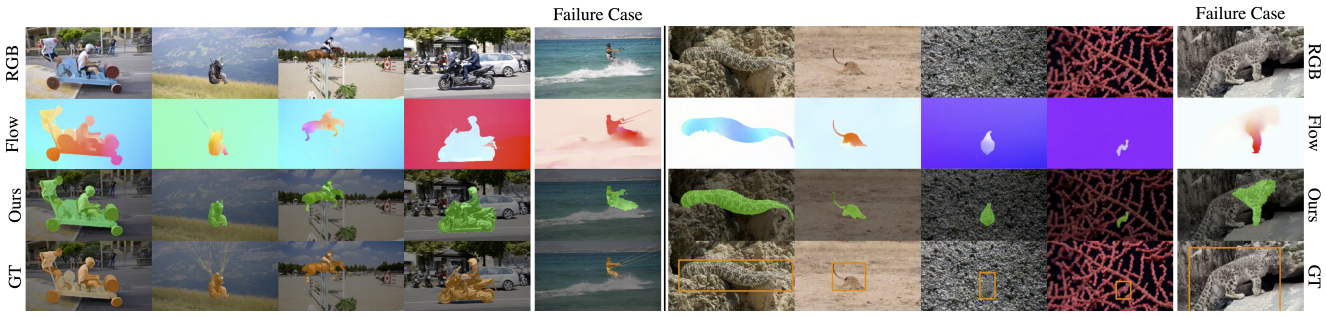

Figure 2. **Qualitative results.** On DAVIS2016 (left), our method is able to segment a variety of challenging objects such as people, animals, and vehicles, often on-par with top supervised approaches. On MoCA (right), our model is able to accurately segment well-camouflaged objects even when previous supervised methods fail completely. We show a failure case (left) where the splash created by the person is incorrectly included, 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 (two-stream) [6]	✓	✓	✓	55.3	0.602	0.523	0.413	0.267	0.088	0.379
COSNet [9]	✓	✓	✗	50.7	0.588	0.534	0.457	0.337	0.167	0.417
MATNet [17]	✓	✓	✓	**64.2**	**0.712**	**0.670**	**0.599**	**0.492**	**0.246**	**0.544**
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 2. **Comparison results on MoCA dataset.** We report the successful localization rate for various thresholds τ (see Section 3). Both CIS and Ours were pre-trained on DAVIS and finetuned on MoCA in a self-supervised manner. Note that, 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.

Grant Visual AI (EP/T028572/1).

References

[1] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proc. ICCV*, 2017.

[2] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *Proc. BMVC*, 2014.

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

[4] 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 *Proc. CVPR*, 2017.

[5] M. Keuper, B. Andres, and T. Brox. Motion trajectory segmentation via minimum cost multicuts. In *Proc. ICCV*, 2015.

[6] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. *Proc. ACCV*, 2020.

[7] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *Proc. ICCV*, 2013.

[8] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Proc. NeurIPS*, 2020.

[9] 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 *Proc. CVPR*, 2019.

[10] 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 *Proc. CVPR*, 2016.

[11] O. Peter, M. Jitendra, and B. Thomas. Segmentation of moving objects by long term video analysis. *TPAMI*, 2014.

[12] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, 2020.

[13] P. Tokmakov, C. Schmid, and K. Alahari. Learning to segment moving objects. *International Journal of Computer Vision*, 2019.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

[15] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *The IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[16] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised moving object detection via contextual information separation. In *Proc. CVPR*, 2019.

[17] T. Zhou, S. Wang, Y. Zhou, Y. Yao, J. Li, and L. Shao. Motion-attentive transition for zero-shot video object segmentation. In *AAAI*, volume 34, pages 13066–13073, 2020.