Unsupervised space-time network for temporally-consistent segmentation of multiple motions

Etienne Meunier
Inria, Rennes, France
etienne.meunier@inria.fr

Patrick Bouthemy
Inria, Rennes, France
patrick.bouthemy@inria.fr

Abstract

Motion segmentation is one of the main tasks in computer vision and is relevant for many applications. The optical flow (OF) is the input generally used to segment every frame of a video sequence into regions of coherent motion. Temporal consistency is a key feature of motion segmentation, but it is often neglected. In this paper, we propose an original unsupervised spatio-temporal framework for motion segmentation from optical flow that fully investigates the temporal dimension of the problem. More specifically, we have defined a 3D network for multiple motion segmentation that takes as input a sub-volume of successive optical flows and delivers accordingly a sub-volume of coherent segmentation maps. Our network is trained in a fully unsupervised way, and the loss function combines a flow reconstruction term involving spatio-temporal parametric motion models, and a regularization term enforcing temporal consistency on the masks. We have specified an easy temporal linkage of the predicted segments. Besides, we have proposed a flexible and efficient way of coding U-nets. We report experiments on several VOS benchmarks with convincing quantitative results, while not using appearance and not training with any ground-truth data. We also highlight through visual results the distinctive contribution of the short- and long-term temporal consistency brought by our OF segmentation method.

1. Introduction

Motion segmentation is a key topic in computer vision that arises as soon as videos are processed. It may be a goal in itself. More frequently, it is a prerequisite for different objectives as independent moving object detection, object tracking, or motion recognition, to name a few. It is also widely leveraged in video object segmentation (VOS), but most often coupled with appearance. Motion segmentation is supposed to rely on optical flow as input. Indeed, the optical flow carries all the information on the movement between two successive images of the video.

Clearly, the motion segmentation problem has a strong temporal dimension, as motion is generally consistent throughout the video, at least in part of it within video shots. The use of one optical flow field at each given time instant may be sufficient to get the segmentation at frame t of the video. However, extending the temporal processing window can be beneficial. Introducing temporal consistency in the motion segmentation framework is certainly useful from an algorithmic perspective: it may allow to correct local errors or to predict the segmentation map at the next time instant. Beyond that, temporal consistency is an intrinsic property of motion that is essential to involve in the formulation of the motion segmentation problem.

In this paper, we propose an original method for multiple motion segmentation from optical flow, exhibiting temporal consistency, while ensuring accuracy and robustness. To the best of our knowledge, our optical flow segmentation (OFS) method is the first one to involve short- and long-term temporal consistency. We are considering a fully unsupervised method, which overcomes tedious or even unfeasible manual annotation and provides a better generalization power to any type of video sequences.

The main contributions of our work are as follows. We adopt an explicit space-time approach. More specifically, our network takes as input a sub-volume of successive optical flows and delivers accordingly a sub-volume of coherent segmentation maps. Our network is trained in a completely unsupervised manner, without any manual annotation or ground truth data of any kind. The loss function combines a flow reconstruction term involving spatio-temporal parametric motion models defined over the flow sub-volume, and a regularization term enforcing temporal consistency on the masks of the sub-volume. Our method also introduces a latent represention of each segment motion and enables an easy temporal linkage between predictions. In addition, we have designed a flexible and efficient coding of U-nets.

The rest of the paper is organized as follows. Section 2 is devoted to related work. In Section 3, we describe our unsupervised 3D network for multiple motion segmentation embedding temporal consistency. Section 4 collects details on our implementation. In Section 5, we report results on several VOS benchmarks with a comparison to several existing methods. Finally, Section 6 contains concluding remarks.

2. Related work

Motion segmentation can be addressed in different ways. It can be 2D in the image sequence, or 3D in the scene. Here, we will deal with 2D motion segmentation from optical flow. Motion segmentation can be understood from a general perspective as the motion-based partitioning of a video frame, or according to more dedicated goals as the detection of independently moving objects.

Motion segmentation has been investigated for decades [23, 25, 48]. Very first attempts in the 90's took two successive images as input. Consequently, they involved both the estimation of parametric motion models from images for each segment, usually affine motion models, and the segmentation of the image, which was a difficult egg-and-chicken problem. It was solved using either clustering [41], Markov random fields (MRF) and robust estimation [28], maximum likelihood estimation and MDL criterion [2], or later, level-set formulation [7, 39].

Since then, accurate and efficient methods for estimating optical flow have become available. As a consequence, motion segmentation methods can consider optical flow as a reliable input. Another change is nowadays the obvious supremacy of deep learning methods in computer vision.

Different kinds of neural networks have been considered for motion segmentation, but often with a two-mask segmentation objective only. An adversarial architecture is designed in [46] to generate a hiding mask on the input optical flow, while an inpainter network attempts to recover the flow within the mask. The idea is that the optical flow cannot be reconstructed from the surrounding optical flow, if the mask corresponds to an independent motion, and consequently, constitutes a different segment. In [45], the authors used a transformer module, more specifically, the slot attention mechanism introduced in [20]. Also, the loss function comprises a flow reconstruction term and an entropy term to make masks as binary as possible. A different approach was followed in [44] that can address multiple motion segmentation. Stacked deep multi-layer perceptrons were designed to learn nonlinear subspace filters, the motion segmentation problem being solved at inference by applying K-means to the output embeddings. Recently, we derived the network loss function and the training procedure, for unsupervised multiple motion segmentation, from the Expectation-Maximization (EM) framework, while using spatial quadratic motion models [24].

Let us also mention the competitive collaboration scheme between several networks, proposed in [33]. The goal was to segment independent moving objects from the estimation of optical flow, depth and camera pose, achieved by dedicated networks, and the computation of the resulting static scene flow, i.e., the apparent motion of the static parts of the scene viewed by a moving camera.

The VOS problem has also leveraged motion segmentation. However, VOS focuses on the segmentation of primary objects, moving in the foreground of a scene throughout a video and possibly tracked by the camera [42]. VOS ground truth is defined as a binary segmentation, i.e., (single) primary moving object versus background, in the DAVIS2016 benchmark that is representative of the VOS task [32]. The common approach for VOS is to jointly take into account appearance and motion. Best performing methods are supervised or semi-supervised CNN-based ones, as proposed in [5, 8, 10, 15, 21, 35, 38, 49]. Unsupervised methods have also been considered in [24, 45–47], the two first ones using motion only. Classical approaches were previously designed for that task, for instance in [14, 31].

The temporal dimension of the motion segmentation problem has been somewhat considered in various ways. First, regarding classical approaches, in [28] the motion partition at time t was predicted from the one obtained at time $t-1$ using the affine motion models estimated between images $t-1$ and t for each segment, within a robust MRF-based method. The authors in [36] showed that it was beneficial to introduce temporal layer constancy over several frames to perform motion segmentation in a MRF-based and graph-cut optimization framework. In [27], large time windows were taken into account, allowing the use of point trajectories within a spectral clustering method but resulting in sparse displacement fields.

More recently, regarding deep learning approaches, segmentation at a given time instant t is enforced during the training phase of the network designed in [45], by considering several time pairs involving instants before and after t and their corresponding optical flow fields. In [17], the scope is a bit different, since the authors deal with amodal segmentation, i.e., the recovery of the whole object even in case of occlusion or temporary static state. To this end, they introduce a multi-frame analysis comprising a transformer encoder, while using synthetic ground truth for training involving human annotation. In the same vein, multiple object segmentation is addressed in [43] with the addition of depth-ordered layer representation. A self-supervised model for VOS has been very recently proposed in [9], taking several consecutive RGB frames as input. Optical flow is computed at training time. Furthermore, a temporal consistency term is added in the loss function. However, the temporal consistency is not applied to two consecutive segmentation maps, but for different pairings between frame t and another (more or less distant) frame. In [10], temporal feature propagation is an important component of the framework of spatio-temporal transformers designed for video object segmentation.

Another way to integrate the evolution of the video is to involve memory modules, as in [38] where a two-stream neural network, encompassing spatial and temporal features, is equipped with an explicit memory designed with convolutional gated recurrent units. Memory networks that can be trained end-to-end are leveraged in [29] for semi-supervised VOS. The memory is fed by frames with object masks and can be dynamically updated. Memory is introduced through a recurrent network for zero-shot VOS in [40]. It is fully end-to-end trainable and involves both the spatial and temporal domains.

Video prediction, a topic of growing importance surveyed in [30], can also be mentioned in this discussion. The objective is to forecast future frames of a video sequence from several past frames. The authors of [1] concentrate on the prediction of the transformations between successive images, represented by affine models, to generate the next frame of the sequence. In [13], the proposed framework handles in different ways predictable moving regions and disoccluded regions, from a confidence factor evaluated after warping. The latter regions are predicted by a dedicated inpainting network. Motion related to actions is predicted from previous frames in [12] in the context of robot interactions, which enables to be partially invariant to object appearance. In [22], convolutional features of the Mask R-CNN instance segmentation model are predicted to produce the segmentation of future frames. Other works proposed LSTM networks for semantic segmentation [26], and local frequency domain transformer networks [11]. In a different perspective, the prediction of probable motion patterns is used at the training stage in [6], as a cue to learn objectness from videos.

Our fully unsupervised approach differs from these previous works in several respects. We rely only on optical flow and take a sub-volume of OF fields as input, providing a sub-volume of consistent segmentation maps. Moreover, we introduce space-time parametric motion models and temporal consistency between consecutive masks.

3. Motion segmentation method

We have designed a 3D convolutional network for multiple motion segmentation from optical flow. The network takes a sub-volume of several successive optical flow fields as input. Temporal consistency is expressed in two main ways at the training stage. Firstly, we introduce space-time parametric motion models to represent the flow in each segment over the space-time sub-volume. Secondly, we define a regularization term in the loss function enforcing stable labeling of the motion segments over the sub-volume.

3.1. Spatio-temporal parametric motion model

For each motion segment k from a set of K segments, we define a spatio-temporal parametric motion model $\tilde{f}_{\theta_k,\alpha_k}$ in the (x,y,t) volume of the sequence, introducing a temporal variation of each spatial parameter of the model. For instance, if we consider an affine spatio-temporal motion model, it is given by:

$$\tilde{f}_{\theta_k,\alpha_k}(x,y,t) = (\theta_{k_1} + \alpha_{k_1}t + (\theta_{k_2} + \alpha_{k_2}t)x + (\theta_{k_3} + \alpha_{k_3}t)y, \\ \theta_{k_4} + \alpha_{k_4}t + (\theta_{k_5} + \alpha_{k_5}t)x + (\theta_{k_6} + \alpha_{k_6}t)y)^T, \quad (1)$$

where $\theta_k = (\theta_{k_1}, .., \theta_{k_6})$ corresponds to the spatial part of the motion model, and $\alpha_k = (\alpha_{k_1}, .., \alpha_{k_6})$ its temporal extension to account for possible variations of the flow over time. By developing eq.(1), an equivalent formulation is:

$$\tilde{f}_{\theta_k,\alpha_k}(x,y,t) = (\theta_{k_1} + \theta_{k_2}x + \theta_{k_3}y + \alpha_{k_1}t + \alpha_{k_2}xt + \alpha_{k_3}yt, \\ \theta_{k_4} + \theta_{k_5}x + \theta_{k_6}y + \alpha_{k_4}t + \alpha_{k_5}xt + \alpha_{k_6}yt)^T. \quad (2)$$

Similar expressions can be straightforwardly defined for any spatio-temporal quadratic motion model.

The spatio-temporal motion model encompasses the successive locations of segment k in the sub-volume. In practice, we take a sub-volume of three flow fields sampled every τ time instants, i.e., at $t-\tau, t$, and $t+\tau$. A typical value for τ is 1, that is, a triplet of three consecutive flows, but, other values can be chosen. If τ is positive, respectively negative, it means that we proceed forward, respectively backward, in time. We can cope with a single τ value, or with several ones jointly, all the triplets being centered on t. Other types of sub-volumes could be handled as well.

3.2. Network architecture

The overall principle of our unsupervised 3D multiple motion segmentation network is illustrated in Fig.1. It is based on the U-net architecture [34]. The network takes a sub-volume of flow fields as input around time t and jointly predicts the sub-volume of segmentation maps, while enforcing temporal consistency on them. One possibility is to keep only the segmentation map m_t, and the process is performed again at the next time instant $t+1$. However, alternatives can be considered, as keeping the three maps.

3.3. Loss function

The loss function of our 3D motion segmentation network is composed of two terms: a segment-wise flow reconstruction term and a temporal consistency one on the predictions. The first term, denoted $\mathcal{L}_r$, expresses how the estimated parametric motion models fit the input optical flow within each segment. It writes:

$$\mathcal{L}_r = \frac{1}{3} \sum_{s\in\{-\tau,0,\tau\}} \frac{\sum\limits_{i\in\mathcal{I}} \sum\limits_{k=1}^{K} m_k(i,t+s) \| f(i,t+s) - \tilde{f}_{\theta_k,\alpha_k}(i,t+s) \|_1}{\sum\limits_{i\in\mathcal{I}} \| f(i,t+s) \|_1}, \quad (3)$$

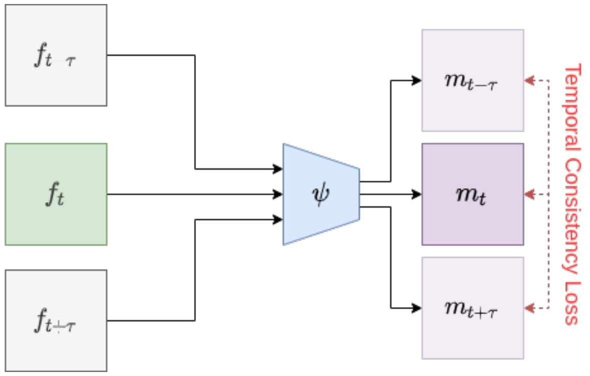

Figure 1. Principle diagram of our space-time multiple motion segmentation network ψ, taking as input a space-time sub-volume composed of three flow fields sampled every τ time instants and delivering a sub-volume of three coherent segmentation maps.

where $i = (x, y)$ is a site of the image grid $\mathcal{I}$, K is the number of motion layers or segments, $f(i, t)$ is the flow vector at site i and time instant t, and $m_k(i, t)$ denotes the probability of site i to belong to motion segment k at time t, that is, the prediction (or output) of the motion segmentation network. f_t will designate the optical flow field at time t, $f_t = \{f(i, t), i \in \mathcal{I}\}$. We use the robust norm L_1 to overcome the presence of outliers in the motion segment, especially at the beginning of the training when segments are not yet well extracted, and to mitigate possibly wrong flow vectors, around motion discontinuities in particular.

The second term, denoted $\mathcal{L}_c$, enforces temporal consistency of the motion segments. To do this, the probability of site i to belong to segment k is assumed to be stable over time within the considered triplet. We have:

$$\mathcal{L}_c = \frac{1}{2K|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{k=1}^{K} \big(|m_k(i, t-\tau) - m_k(i, t)| \\ + |m_k(i, t) - m_k(i, t+\tau)|\big), \quad (4)$$

where $|\mathcal{I}|$ designates the number of sites over the image grid $\mathcal{I}$. For the sake of simplicity, we have adopted an Eulerian standpoint [3], that is, we compare segment labels over time at any given site i of the image grid. In fact, every point is likely to move and a Lagrangian standpoint [3] would be more appropriate. It would require the use of the optical flow vectors to track every point over time. However, the computed flow field may be imprecise or even erroneous at some points, and besides, interpolation operations would be necessary since the flow components take on real values. The Eulerian temporal constraint does not mean that the site should be static within the triplets, but it works as long as the site lies on the overlap of the successive positions of the moving parts. However, it does not make sense on occluded or disoccluded parts. This justifies the use of the L_1 norm to deal with the latter configuration.

We further prevent from enforcing the temporal consistency over occlusion areas by ignoring sites i in the summation over $\mathcal{I}$ in eq.(4) that exhibit a large temporal flow difference. More precisely, we set a threshold λ so that a quantile η of sites i is discarded as follows:

$$p(\|f(i, t+\tau) - f(i, t)\|_1 \geq \lambda) \leq \eta. \quad (5)$$

In practice, we take $\eta = 1\%$. In doing so, we make an implicit assumption on the overall surface of the occlusion areas, but it seems reasonable for the datasets we deal with. The loss function is the sum of the two terms:

$$\mathcal{L} = \mathcal{L}_r + \beta \mathcal{L}_c. \quad (6)$$

We simply set $\beta = 1$, since the two terms of the loss function are properly normalized.

With our approach, we can easily infer a temporal linkage between successive predictions, as explained below.

3.4. Segment selection for evaluation

To evaluate our method and compare it with similar unsupervised methods, we use VOS benchmarks as a substitute for optical flow segmentation (OFS) benchmarks, as no such benchmarks are available. The two tasks are close. However, the VOS one is attached to the notion of a primary object of interest moving in the foreground (sometimes, a couple of objects). As a consequence, we have to select the right segments to cope with the binary ground truth of the VOS benchmarks, as described below.

First, we link throughout the video the K segments obtained at each instant t. The fact that we proceed with sub-volumes imposing common segment labels within the sub-volume, and that consecutive sub-volumes share two masks, helps us establish the temporal linkage. We link segments from one time instant to the next one, using the IoU measure (intersection over union) as linkage criterion. This is achieved throughout the video by sub-sequences, and then, the comparison of the K segments with the ground truth is done at this sub-sequence level, enforcing the temporal dimension of our approach.

This procedure is applied on the three datasets DAVIS2016, SegTrackV2 and FBMS59. In practice, we take sub-sequences of 10 frames. The segment association, required to compute the Jaccard score, leverages the ground truth, but with the notable fact that it is done only once at the sub-sequence level, which shows the ability of our method to supply long-term stability.

Let us mention that we are able to infer another information related to motion. We can generate a latent representation of the segment motion. More specifically, this latent representation $\chi(S_i)$ of segment S_i is defined as the average value of the latent vectors, normalized in mean and variance, of the segment sites. In future work, we could establish a motion similarity measure between two segments

S_i and S_j, given by the dot product of their respective latent representations χ_j and χ_j. This motion similarity could be used in the temporal linkage of the segments in addition to IoU, or to merge segments, especially if we take a large value for the mask number K.

More details on the different items presented in this subsection are provided in the supplementary material.

4. Implementation

4.1. Network coding

In this work, we explored different ways of dealing with the input optical flow volume and especially different interactions between time steps. We also wanted to use a multi-resolution structure to segment large inputs while preserving fine-grain details. In order to easily handle these different options, we developed an original and flexible implementation[1] of a U-net based on the decomposition of its structure into five modules as described in Fig.2.

The skeleton of our network is a version of the classical U-net calling abstract instances of these modules and applying error checks to control their output. Using our implementation, one can easily instantiate a novel architecture by solely implementing desired modules without getting involved in the core steps of the U-net. Since all U-net blocks are composed of the same modules, we can stack them making the code needed to implement a new architecture minimal. The input and bottleneck layers of the U-net are handled seamlessly by using a part of provided modules.

This framework makes it straightforward to implement a multidimensional U-net, to incorporate various transformations in the transit layer (e.g., transformer as in [43] or recurrent CNN as in [26]), or to change the sampling or upsampling steps, keeping the same general skeleton between all these different architectures. For example, in our work, we implemented a version of downsampling and upsampling that is applied only on the spatial dimension, while double convolutions are applied on both the spatial and temporal dimensions. The proposed implementation encompasses many of the solutions described in Section 2, and allows new ideas to be tested quickly and in a standardized way. It is also applicable beyond motion segmentation. We will make the code available in an open source repository.

4.2. Implementation details

As in [24, 45], we adopt the RAFT method [37] to compute the optical flow fields. More specifically, we use the RAFT implementation[2] with network weights fine-tuned on the MPI Sintel dataset [4]. We downsample them to feed the network with 128×224 vector fields as input. Thus, we achieve much more efficient training and inference stages.

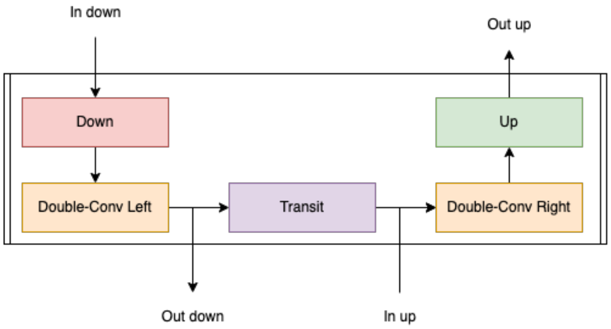

Figure 2. Diagram of the prototype layer in our implementation of the U-net. "Double-Conv Left" and "Double-Conv Right" are applying several transformations of the feature maps after downscaling and before upscaling respectively. In classical U-net architecture, it is a 2D convolution kernel applied on the spatial dimension followed with a batch norm. "Down" is a downscaling layer that reduces the spatial dimension of the feature map. In classical U-net architecture, it is implemented using max pooling 2D. "Up" is a block that increases the dimension of the feature map usually implemented with bilinear interpolation or transposed convolution. "Transit" is the connection between the down path and the up path. In classical U-net architecture, it is a skip connection.

The output segmentation maps are then upsampled to the original frame size for evaluation w.r.t. the ground truth.

Regarding the estimation of the spatio-temporal parametric motion model, since the x and y coordinates are normalized within $[0, 1]$, we apply a similar normalization for the t coordinate. For instance, if we set $\tau = 1$, we get as normalized time values: $t - 1 = -0.33$, $t = 0$, and $t + 1 = 0.33$. We use the full quadratic motion model, with 12 spatial parameters and 12 temporal ones, in all the reported experiments. This type of motion model enables to account for complex depth surfaces and movements.

We use only the prediction $m_k(i, t)$ of the network obtained when considering the flow triplet $(f_{t-\tau}, f_t, f_{t+\tau})$ to decide to which segment k site i belongs to at time instant t. More precisely, we select for each point i the segment $\hat{k}$ with the highest probability. In all the experiments reported in Section 5, we simply use a single value for τ, $\tau = 1$. We refer the reader to the supplementary material for possible alternatives. Let us recall that negative values of τ mean that we proceed backward in time. A combinaison of several τ values could also be used (see the supplementary material).

4.3. Data augmentation and network training

We perform two types of data augmentation. The first one consists in adding a global flow to the input flow as done in [24]. The global flow is given by a full spatio-temporal motion model whose parameters are chosen at random. We just make sure that the added global flow is roughly equivalent in magnitude to the initial flow. The same global flow

[1] https://github.com/Etienne-Meunier-Inria/GeneralUnet
[2] https://github.com/princeton-vl/RAFT

Network modification	Spatial quadratic motion model	Loss without $\mathcal{L}_c$	Our full method
$\mathcal{J} \uparrow$	70.5	33.1	73.2

Table 1. Ablation study for two main components of our method. Only one component is modified at a time. $\mathcal{J}$ is computed on the DAVIS2016 validation set.

is added to the three flow fields of the input sub-volume. This type of data augmentation allows us to mimic different camera motions, enforcing that the motion segments are independent of it. For the second type of data augmentation, we corrupt one input flow out of the three ones. The idea is to simulate a poorly estimated flow field and to compel the temporal consistency to compensate.

Our motion segmentation method is entirely unsupervised. We do not perform any manual annotation in all the experiments. We train the 3D motion segmentation network on the training set of DAVIS2016, once for all. Moreover, the stopping epoch is selected from the loss function evaluated on the DAVIS2016 validation set. We use Adam optimizer with a learning rate of 10^{-4} to train the 3D network. The estimation of the parameters $\{\theta_k, \alpha_k, k = 1, K\}$ of the motion models is achieved with the Pytorch implementation of L-BGFS [19]. Let us recall that we estimate the parametric motion models only at training time.

Our method is very efficient at test time. For the model (small U-Net 3D), the computational time amounts on average to 114 *fps* on a P100. The impact is negligible regarding the number K of masks used since only the final layer is modified, and it is proportional to the frame size $|\mathcal{I}|$.

5. Experimental results

5.1. Datasets

We have carried out experiments on three VOS datasets: DAVIS2016[3] [32], SegTrackV2[4] [18], and FBMS59 [27].

DAVIS2016 consists of 50 videos (and 3455 frames) that are split in a training set of 30 videos and a validation set of 20 videos. They contain diverse moving objects. Only the primary moving object is annotated in the ground truth. The criteria for evaluation on this dataset are the Jaccard score (denoted $\mathcal{J}$), and the contour accuracy score (denoted $\mathcal{F}$).

SegTrackV2 includes 14 videos (with a total of 1066 annotated frames), and FBMS59 contains 59 videos (720 annotated frames), both involving one moving object but sometimes a couple of moving objects. For FBMS59, we use the 30 sequences of the validation set for evaluation. As done in [45], if there are several moving objects, we group them into a single foreground mask for evaluation.

5.2. Ablation study

We have conducted an ablation study to assess two main components of our method related to the temporal dimension. We proceeded by modifying only one component at a time. The two network components concerned are: *i)* use of a spatial quadratic motion model per frame instead of the spatio-temporal one, *ii)* specification of the loss function without the consistency term $\mathcal{L}_c$. All the ablation experiments were run on the DAVIS2016 validation set. Results are collected in Table 1. We can observe that the spatio-temporal motion model improves the performance of the method, by taking into account the possible motion evolution within the sub-volume. Above all, the introduction of the temporal consistency term $\mathcal{L}_c$ in the loss function is drastically beneficial. The ablation study demonstrates the pivotal role of the two components acting at two levels of temporal consistency in the flow segmentation

5.3. Quantitative and comparative evaluation

We report in Table 2 the results obtained by our method on the three datasets DAVIS2016, SegTrackV2 and FBMS59, along with those obtained by other existing methods. Since our method is fully unsupervised and only uses optical flow as input, we focus on similar methods for a fair comparison. We consider the method categories that we proposed in [24] regarding input and training, by the way very close to other propositions. We have added a category w.r.t. the network input for two very recent methods, [6, 9], that only use RGB images as imput at test time, the optical flow being only involved in the loss function. Additionally, the OCLR method [43] exploits human-annotated sprites to generate realistic shapes in the synthetic data used in the training. We consider the OCLR version taking only optical flow as input. The post-processing added to the CIS method [46], based on Conditional Random Fields (CRF), is an heavy one, which leads most authors to retain only the version without post-processing for a fair comparison.

Overall, our method shows convincing performance w.r.t. comparable methods, namely, unsupervised methods taking optical flow as input. Temporal consistency was properly handled by our method and gave quite satisfying results. More specifically, our method shows an excellent performance on DAVIS2016 and a very good performance on FBMS59. Regarding SegTrackV2, this dataset includes sequences filmed with a poorly controlled handheld camera, which leads to unstable sequences where the contribution of our method cannot be as significant.

5.4. Qualitative visual evaluation

Fig.3 contains several visual results to demonstrate how our method behaves on different situations. We display result samples obtained on different videos of the benchmarks. We can observe that the segmentation are globally

[3] https://davischallenge.org/index.html

[4] https://paperswithcode.com/dataset/segtrack-v2-1

Method	Training	Input	DAVIS2016		SegTrack V2	FBMS59
			$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{J}\uparrow$
Ours	Unsupervised	Flow	73.2	70.3	55.0	59.4
EM [24]			69.3	70.7	55.5	57.8
MoSeg [45]			68.3	66.1	58.6	53.1
FTS [31]			55.8		47.8	47.7
TIS$_0$ [14]			56.2	45.6	-	-
OCLR* [43] (flow-only)			72.1	-	67.6	65.4
GWM [6]		RGB (Flow in loss)	79.5	-	78.3	77.4
MOD [9]			73.9	-	62.2	61.3
TIS$_s$ [14]		RGB & Flow	62.6	59.6	-	-
CIS - No Post [46]			59.2		45.6	36.8
CIS - With Post [46]			71.5		62.0	63.6
DyStab - Dyn [47]	Supervised Features	Flow	62.4		40.0	49.1
DyStab - Stat&Dyn [47]		RGB & Flow	80.0		73.2	74.2
ARP [16]			76.2	70.6	57.2	59.8
MATNet [49]	Supervised	Flow	82.4	80.7		
COSNet [21]		RGB	80.5	79.5	-	75.6

Table 2. Results obtained with our method ($K = 4$) on DAVIS2016, SegTrackV2, and FBMS59, including comparison with unsupervised and supervised methods (scores from cited articles). The Jaccard index $\mathcal{J}$ expresses the correct overlap (intersection over union) between the extracted segments and the ground truth, while $\mathcal{F}$ focuses on segment boundary accuracy (the higher the better). Performance is assessed by the average score over all samples, for all datasets but DAVIS2016. For the latter, the overall score is given by the average of sequence scores. *OCLR is not a truly unsupervised method since it relies on human-annotated sprites to get realistic shapes in the synthetic data used in the training.

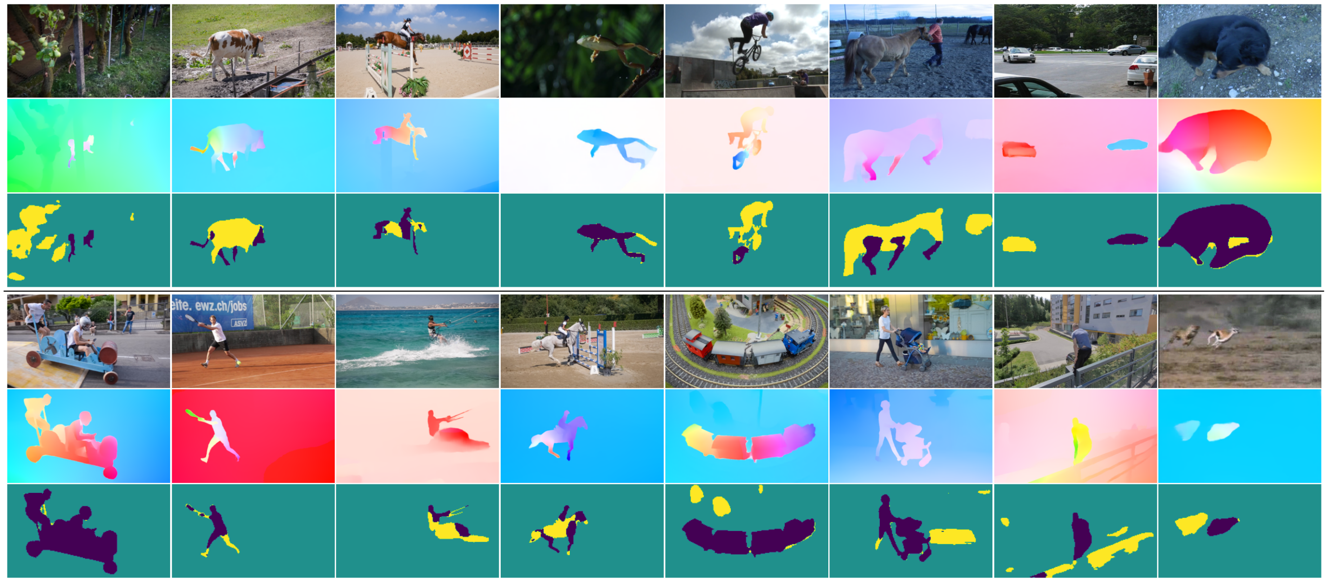

Figure 3. Results obtained with our method using four masks ($K = 4$), but the network may not necessarily use all the four masks. Two groups of results. For each group, the first row depicts one image of the video, the second row contains the optical flow input represented with the usual HSV color code, the third row displays motion segments (given by layers that are not necessarily connected) with one colour per segment. Samples are drawn from the different datasets.

accurate. Since our method can involve K masks, we can properly handle articulated motions, or the presence of several moving objects in the scene, as illustrated in Fig.3. We must keep in mind that our actual target is the OFS task,

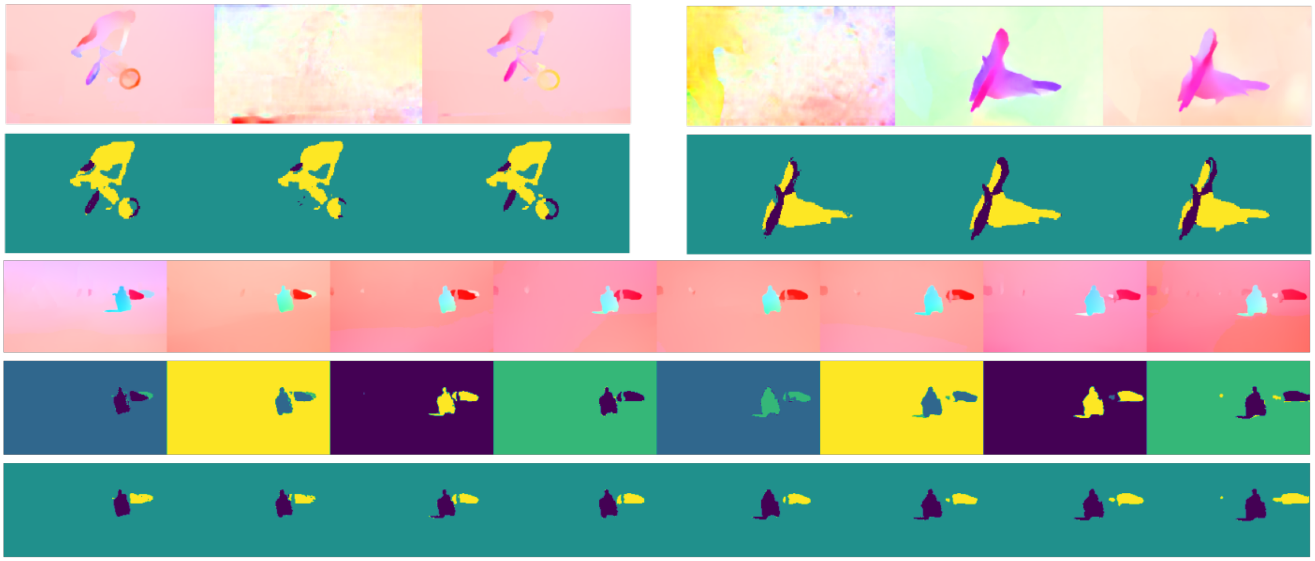

Figure 4. Impact of temporal consistency on several situations with our method ($K = 4$). Two groups of results from top to bottom. First group: two cases of amodal segmentation, in fact corresponding to a repeated image in the video file mimicking a stop (first row contains optical flow input, second row displays motion segments). Second group: one case exemplifying the action of the temporal consistency term $\mathcal{L}_c$ of the loss function (short term) and of the temporal linkage (long term) to maintain the same mask labels over time (first row contains optical flow input, second row displays motion segments obtained without $\mathcal{L}_c$, third row includes motion segments obtained with $\mathcal{L}_c$ and temporal linkage).

even if we evaluate our method on VOS benchmarks. Since the VOS benchmarks mainly deal with the segmentation of one primary object moving in the foreground, it may occur some discrepancies with OFS. For instance, the segmentation of additional parts w.r.t. VOS ground truth makes nonetheless sense from the OFS standpoint. Let us mention the cases of a moving car in the background, two animals running, ripples on the water, motion parallax due to static objects in the foreground, as illustrated in several examples of Fig.3. It can affect the overall scores reported in Table 2.

We gather in Fig.4 several result samples that demonstrate the benefit of the short-term and long-term temporal consistency provided by our method, with respectively the $\mathcal{L}_c$ term of the loss function defined in eq.(4) and the temporal linkage described in Subsection 3.4. Our method is able to recover the moving object segment when the object is temporarily static, showing its ability to segment amodally without any dedicated training, as shown in the first group of Fig.4. The second group highlights how our method can maintain the same mask labels in the video. Additional results can be found in the supplementary material.

6. Conclusion

We have designed an original unsupervised method[5] for the segmentation of multiple motions in a video. It fully leverages the temporal dimension of the motion segmentation problem. To the best of our knowledge, our method is the first unsupervised network-based OFS method involving short- and long-term temporal consistency, which leads to stable OF segmentation along the video. It introduces at training time spatio-temporal parametric motion models in sub-volumes, and a loss term expressing temporal consistency over consecutive masks while taking care of occlusions. In addition, the method allows for an easy temporal linkage of the motion segments throughout the video.

Our 3D network is flexible by design. It can straightforwardly handle different choices of mask number for the multiple motion segmentation. Different flow sub-volumes can be envisaged as input, including forward and backward in time. Besides, we have proposed an efficient way to code U-nets, which can be easily generalized beyond motion segmentation. Experimental results on several datasets demonstrate its efficiency and its accuracy by providing competitive results. Future work could leverage the latent representation of the segment motion over the video, which can contribute for example to a motion similarity measure.

Acknowledgements

This work was financially supported by BpiFrance through the LiChIE contract within a collaboration with Airbus Defence&Space. The authors thank Renaud Fraisse for valuable discussions.

[5] https://github.com/Etienne-Meunier-Inria/ST-Space-Time-Flow-Segmentation

References

[1] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala. Transformation-based models of video sequences. *arXiv: 1701.08435*, 2017. 3

[2] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *International Conference on Computer Vision (ICCV)*, Boston, June 1995. 2

[3] G.K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, 1967. 4

[4] D.J. Butler, J. Wulff, G.B. Stanley, and M.J. Black. A naturalistic open source movie for optical flow evaluation, In *European Conference on Computer Vision (ECCV)*, 2012. 5

[5] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Int. Conf. on Computer Vision (ICCV)*, Venice, 2017. 2

[6] S. Choudhury, L. Karazija, I. Laina, A. Vedaldi, and C. Rupprecht. Guess what moves: Unsupervised video and image segmentation by anticipating motion. *British Machine Vision Conference (BMVC)*, London, 2022. 3, 6, 7

[7] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *Int. Journal of Computer Vision*, 62(3):249–265, 2005. 2

[8] A. Dave, P. Tokmakov, and D. Ramanan. Towards segmenting anything that moves. In *Int. Conference on Computer Vision Workhops (ICCVW)*, Seoul, 2019. 2

[9] S. Ding, W. Xie, Y. Chen, R. Qian, X. Zhang, H. Xiong, and Q. Tian. Motion-inductive self-supervised object discovery in videos. *arxiv.org/abs/2210.00221*, October 2022. 2, 6, 7

[10] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor. SSTVOS: Sparse spatiotemporal transformers for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[11] H Farazi, J. Nogga, and S. Behnke. Local frequency domain transformer networks for video prediction. In *International Joint Conference on Neural Networks (IJCNN)*, 2021. 3

[12] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Conference on Neural Information Processing Systems (NIPS)*, Barcelona, 2016. 3

[13] H. Gao, H. Xu, Q.-Z. Cai, R. Wang, F. Yu, and T. Darrell. Disentangling propagation and generation for video prediction. In *International Conference on Computer Vision (ICCV)*, Seoul, 2019. 3

[14] B. Griffin and J. Corso. Tukey-inspired video object segmentation. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Waikoloa Village, January 2019. 2, 7

[15] 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 *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017. 2

[16] Y. J. Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, July 2017. 7

[17] H. Lamdouar, W. Xie, and A. Zisserman. Segmenting invisible moving objects. In *British Machine Vision Conference (BMVC)*, November 2021. 2

[18] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *International Conference on Computer Vision*, Sydney, 2013. 6

[19] D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. In *Mathematical Programming*, 45(1-3):503-528, 1989. 6

[20] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Conference on Neural Information Processing Systems (NeuRIPS)*, 2020. 2

[21] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, June 2019. 2, 7

[22] P. Luc, C. Couprie, Y. LeCun, and J. Verbeek. Predicting future instance segmentation by forecasting convolutional features. In *European Conference on Computer Vision (ECCV)*, Munich, 2018. 3

[23] J. Mattheus, H. Grobler, and A. M. Abu-Mahfouzl. A review of motion segmentation: Approaches and major challenges. *International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, November 2020. 2

[24] E. Meunier, A. Badoual and P. Bouthemy. EM-driven unsupervised learning for efficient motion segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, early access, August 2022, doi:10.1109/TPAMI.2022.3198480. 2, 5, 6, 7

[25] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *Int. Journal of Computer Vision*, 19(1):29-55, 1996. 2

[26] S. Shahabeddin Nabavi, M. Rochan, and Y. Wang. Future semantic segmentation with convolutional LSTM. In *Bristish Machine Vision Conference*, Newcastle upon Tyne, 2018. 3, 5

[27] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187-1200, June 2014. 2, 6

[28] J.-M. Odobez and P. Bouthemy. MRF-based motion segmentation exploiting a 2D motion model robust estimation. In *International Conference on Image Processing (ICIP)*, Washington, October 1995. 2

[29] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim. Video object segmentation using space-time memory networks. In *Int. Conference on Computer Vision (ICCV)*, Seoul, 2019. 3

[30] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez and A. Argyros. A review on deep learning techniques for video prediction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806-2826, June 2022. 3

[31] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *IEEE International Conference on Computer Vision (ICCV)*, Sydney, December 2013. 2, 7

[32] 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 *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, June 2016. 2, 6

[33] 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 *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019. 2

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Munich, October 2015. 3

[35] H. Song, W. Wang, S. Zhao1, J. Shen, and K.-M. Lam. Pyramid dilated deeper ConvLSTM for video salient object detection. In *European Conference on Computer Vision (ECCV)*, Munich, 2018. 2

[36] D. Sun, E. B. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, June 2012. 2

[37] Z. Teed and J. Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020. 5

[38] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Int. Conf. on Computer Vision (ICCV)*, Venice, 2017. 2, 3

[39] C. Vazquez, A. Mitiche, and R. Laganière Joint multiregion segmentation and parametric estimation of image motion by basis function representation and level set evolution. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(5):782–793, May 2006. 2

[40] 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 *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019. 3

[41] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625-638, Sept.1994. 2

[42] W. Wang, T. Zhou, F. Porikli, D. Crandall, and L. Van Gool. A survey on deep learning technique for video segmentation. arXiv:2107.01153v1, Dec. 2021. 2

[43] J. Xie, W. Xie, and A. Zisserman. Segmenting moving objects via an object-centric layered representation. *arXiv:2207.02206*, July 2022. 2, 5, 6, 7

[44] X. Xu, L. Zhang, L.-F. Cheong, Z. Li, and C. Zhu. Learning clustering for motion segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3):908-919, March 2022. 2

[45] C. Yang, H. Lamdouar, E. Lu, A. Zisserman, and W. Xie. Self-supervised video object segmentation by motion grouping. In *International Conference on Computer Vision (ICCV)*, October 2021. 2, 5, 6, 7

[46] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised moving object detection via contextual information separation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019. 2, 6, 7

[47] Y. Yang, B. Lai, and S. Soatto, DyStaB: Unsupervised object segmentation via dynamic-static bootstrapping. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 7

[48] L. Zappella, X. Llado, and J. Salvi Motion segmentation: A review. *Frontiers in Artificial Intelligence and Applications*, 184:(398-407), January 2008. 2

[49] T. Zhou, J. Li, S. Wang, R. Tao, and J. Shen. MATNet: Motion-attentive transition network for zero-shot video object segmentation. *IEEE Trans. on Image Processing*, Volume 29, August 2020, doi:10.1109/TIP.2020.3013162 2, 7