Segmenting the motion components of a video: A long-term unsupervised model

Etienne Meunier and Patrick Bouthemy

Inria, Centre Rennes - Bretagne Atlantique, France

April 18, 2024

Abstract

Human beings have the ability to continuously analyze a video and immediately extract the motion components. We want to adopt this paradigm to provide a coherent and stable motion segmentation over the video sequence. In this perspective, we propose a novel long-term spatio-temporal model operating in a totally unsupervised way. It takes as input the volume of consecutive optical flow (OF) fields, and delivers a volume of segments of coherent motion over the video. More specifically, we have designed a transformer-based network, where we leverage a mathematically well-founded framework, the Evidence Lower Bound (ELBO), to derive the loss function. The loss function combines a flow reconstruction term involving spatio-temporal parametric motion models combining, in a novel way, polynomial (quadratic) motion models for the spatial dimensions and B-splines for the time dimension of the video sequence, and a regularization term enforcing temporal consistency on the segments. We report experiments on four VOS benchmarks, demonstrating competitive quantitative results, while performing motion segmentation on a whole sequence in one go. We also highlight through visual results the key contributions on temporal consistency brought by our method.

1 Introduction

When dealing with videos, motion segmentation is one of the key issues. Human beings have the ability to continuously analyze a video and immediately extract the diverse motion components. We want to design a similar paradigm to provide a coherent and stable motion segmentation over the video sequence. Since optical flow (OF) yields all the information on the movement between two images of the video sequence, it is natural to base motion segmentation on optical flow. Motion segmentation is a computer vision task in its own, but it is also useful for downstream tasks. Indeed, it is essential to have an accurate and reliable motion segmentation for independent moving object detection, multiple

object tracking, or video understanding, to name a few. Motion segmentation based on optic flow has recently received renewed attention in the context of deep learning [29, 47, 49, 50], but the temporal extension remains to be further investigated.

In this paper, we propose an original long-term unsupervised model for segmenting all the motion components of a video, while taking optical flow as input. It must therefore be considered first and foremost as an unsupervised method performing optical flow segmentation (OFS) over an entire sequence. We assume that optical flow is readily available, computed by one of the recent efficient optical flow methods [43]. The optical flow field at time t enables to get the motion segmentation map at frame t of the video. However, taking a large time window, and even the whole video sequence, is beneficial, since motion entities are generally consistent throughout a video sequence (or at least within every video shot for long videos). Indeed, temporal coherence is inherent to motion.

Motion components (or segments) observed in an image sequence may come from a variety of sources. The first that springs to mind is the case of one or several independent moving objects in the viewed scene. In case of articulated motion, a motion segment will ensue from each part of the articulated body. If the camera is itself moving, the static scene exhibits an apparent motion in the video, which may induce several motion segments according to the depth and/or orientation of the static objects in the scene. In particular, a static object in the foreground of the scene will have an apparent motion different than the scene background one, which is usually referred to as motion parallax [15].

Even if we take an object-oriented view of video content, motion appears as an intrinsic, globalizing property of an object. It is more immediate to segment an object based on its motion than on its appearance. Indeed, motion has been used in the training stage to facilitate image segmentation at inference time [7], or as a criterion to validate image segments [25]. Motion segmentation is also frequently involved in video object segmentation (VOS).

Admittedly, better performance is obtained in VOS, if appearance and motion are combined, compared to the single use of motion. However, there is a bias in the VOS datasets: the primary object to be detected is not only salient by its motion compared to the background, but also by its appearance and even its location, the object being generally placed in the image center. Yet, combining the two modalities may confine their use to videos where the appearance of the object of interest can be easily represented by deep features, like DINO features [4] that are trained on a huge set of natural images from the Internet. Indeed, such algorithms could not be easily applied to videos depicting other types of objects and contents, such as satellite images, biological images, underwater or industrial videos. Models trained on optical flow have a much better generalization and domain transfer ability. Moreover, relying on motion allows to deal with contents where appearance is not discriminating, for instance for crowd scenes [28], or for moving camouflaged animals in natural scenes [18].

Our long-term unsupervised spatio-temporal model is a significant extension of the model we introduced in [2]. For the sake of clarity, from now on, we

will name the method presented in this paper LT-MS (for long-term motion segmentation) and the previous one described in [2] ST-MS (for short-term motion segmentation).

First, we have designed a fully end-to-end method, whereas ST-MS required an *ad-hoc* post-processing step to achieve the temporal linkage beyond three successive frames. This was made possible by defining a new space-time parametric motion model to handle more elaborated temporal evolutions. The spatial part of the motion model remains a polynomial model (e.g., a quadratic one), while the temporal part results from a B-spline approximation. Besides, we have added a transformer (only the decoder part) on the latent space of the 3D U-net, allowing for long-range interactions. While ST-MS takes only three consecutive flows as input, LT-MS can accommodate at inference time any length of series of flow fields, ten, twenty, one hundred, or even the whole sequence. Second, the LT-MS loss function is similar to the ST-MS one, but it is now inferred in a mathematical well-founded manner, leveraging the Evidence Lower Bound (ELBO) framework. We also propose a mathematical development to motivate the temporal consistency term of the loss function. Third, we have added a fourth dataset in the experimental evaluation, the DAVIS2017-motion dataset introduced in [47].

The rest of the paper is organized as follows. Section 2 describes related work regarding motion segmentation. Section 3 presents our unsupervised network for long-term multiple motion segmentation. In Section 4, we provide implementation details. Section 5 reports results on four VOS benchmarks with a comparison to state-of-the-art unsupervised motion segmentation methods. Finally, Section 6 contains concluding remarks.

2 Related work

Motion segmentation aims to break down each frame of a video sequence into components (or segments) of coherent motion. Usually, each motion segment is identified by a motion model, which can be hand-crafted such as affine or quadratic polynomial models, or sometimes learned [42]. Motion segmentation has been investigated for decades [8, 26, 31, 40, 45, 46, 53]. However, we will focus on recent methods in this section. We refer the reader to [29, 31, 53] for a review of older works. Since accurate and efficient methods for estimating optical flow are now available, motion segmentation methods can leverage optical flow as a reliable input. The advent of deep learning, and more recently the use of transformers with attention mechanisms [14], has also encompassed motion segmentation.

In [49], a transformer module, more specifically, the slot attention mechanism introduced in [24], is leveraged to perform motion segmentation from optical flow. As a matter of fact, it addresses a binary segmentation problem, foreground moving object *vs* background. The loss function is composed of two terms, a flow reconstruction term and an entropy term to make masks as binary as possible. In [2], it is assumed that camera rotation can be removed in the opti-

cal flow field, which leads to a simplified scheme to detect independently moving objects in a self-supervised way. Another approach is adopted in [48] that is able to cope with multiple motion segmentation. Nonlinear subspace filters are learned from stacked deep multi-layer perceptrons. Then, motion segmentation is inferred at inference by applying K-means to the output embeddings. In [29], we leveraged the Expectation-Maximization (EM) framework to define the loss function and the training procedure for an unsupervised frame-by-frame motion segmentation, where 12-parameter quadratic motion models are involved. In [50], the authors developed an adversarial method whose aim is to generate a mask hiding the input optical flow, where an inpainter network attempts to recover the flow within the mask. The rationale is that no correct reconstruction of the inside flow can be achieved from the outside flow, if the hidden area exhibits an independent motion and then constitutes a motion segment.

The temporal dimension of motion segmentation has already been considered in different ways. Regarding classical approaches, in [33] 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 [40] 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 [32], large time windows were taken into account, but point trajectories were used as input within a spectral clustering method.
Regarding deep learning approaches, motion segmentation at time t is improved at training time in [49], by taking, as input, flows between t and several time instants before and after t. The authors of [10] consider several consecutive RGB frames as input of their self-supervised method. Optical flow is only computed at training time, and the loss function also comprises a temporal consistency term. However, the latter is not applied to two consecutive segmentation masks, but for pairings between the same frame t and another (more or less distant) one. In [11], spatio-temporal transformers were designed for video object segmentation involving temporal feature propagation. In [52], the authors introduce deformable sprites with a video auto-encoder model, but optimized at test time, to segment scene elements through the video. It leverages texture and non-rigid deformations. The work [36] involves pretrained appearance features (using DINO) and optical flow. The objective function is derived from a form of spectral clustering and applied to the entire video. However, optimization is required at test time. In [2], we explicitly handled the temporal dimension of the problem by considering triplets of consecutive flows as input, a space-time polynomial motion model, and a temporal consistency term in the loss function.

Since motion segmentation can be used in the context of VOS and that we will comparatively evaluate our method on VOS benchmarks, we briefly review VOS methods. The latter often combine appearance and motion features [6, 12]. It was addressed with supervised or semi-supervised methods as in [9, 11, 16, 51], but also with unsupervised methods [29, 49, 50]. VOS is concerned with the segmentation of primary objects, i.e., object moving in the foreground of a scene and possibly tracked by the camera [54]. Thus, VOS generally involves

4

binary ground truth segregating primary moving object against background. This is for instance the case for the DAVIS2016 benchmark [35]. Recent works have revisited the way of coupling appearance and motion for VOS. The AMD method [23] includes two pathways, the appearance one and the motion one. If it does not use optical flow as input and brings out the objectness concept, it nevertheless relies on a coarse motion representation. The RCF method [21] involves learnable motion models, and is structured in two stages, a motion-supervised object discovery stage, and a refinement stage with residual motion prediction and high-level appearance supervision. However, the method cannot distinguish objects undergoing different motions. In [7], the prediction of motion patterns is used at training time to learn objectness from videos. The DivA method [1] promotes divided attention. It is based on the same principle as in [50] that motion segments are mutually uninformative, but it is not limited to binary segmentation. It can segment a variable number of moving objects from optical flow, by leveraging a slot attention mechanism guided by the image content through a cross-modal conditional slot decoder.

Our fully unsupervised long-term approach differs from these previous works in several respects as outlined in the introduction.

3   Long-term Motion Segmentation Method

We have designed a transformer-based network for multiple motion segmentation from optical flow. It is inspired from the MaskFormer architecture [5], but it only comprises one head corresponding to the mask prediction, as described in Fig.1. The network takes as input a volume, of flexible temporal length, comprising several consecutive optical flow fields. Temporal consistency is expressed in two main ways at the training stage. Firstly, we associate a space-time motion model with each segment to characterize its motion along with the evolution of the latter over time. Secondly, the loss function comprises an additional term enforcing consistent labeling of the motion segments over the volume. Adding the transformer to the latent space of the 3D U-net allows for interaction between distant features in the input, and thus for long-term segmentation. We only incorporate a transformer decoder on the downsampled latent space, which induces a limited additional computational load.

3.1   Spatio-temporal parametric motion model

The set of T consecutive flows will be designated as a space-time volume (or volume, to make it short). The volume could even be the whole video sequence. Our space-time motion model is estimated through B-spline functions [44] for its temporal dimension. More precisely, we consider a B-spline function of order n in the variable t over the space-time volume. For the spatial extension, we assign a parametric motion model $\tilde{f}_{\theta_k}$ to each motion segment $k, k = 1, .., K$, where θ_k specifies the spatial motion model $\tilde{f}$ for segment k. In practice, we take $n = 3$, which proved sufficient to handle motion evolution. The number
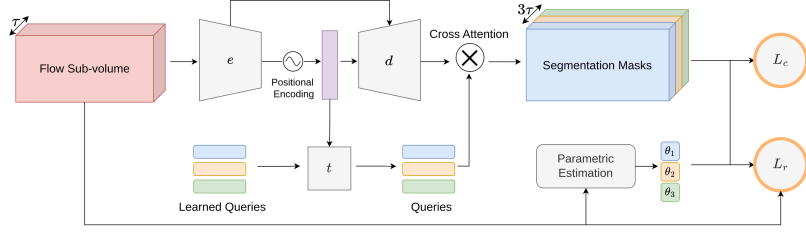
Figure 1: Overall architecture of our multiple motion segmentation method ensuring temporal consistency with the loss term $\mathcal{L}_c$ and the B-spline space-time motion models θ_k (for $k = 1, .., K$). It takes as input a volume of T flow fields. It comprises a 3D U-net (e and d boxes) and a transformer decoder (t box). It also involves positional encoding. A cross-attention product yields the K segmentation masks corresponding to the input volume For the sake of clarity, the block diagram is represented for three motion segments ($K = 3$). $\mathcal{L}_r$ is the flow-reconstruction loss term.

L of control points is given by $L = 2 + \lfloor \frac{T-2}{\nu} \rfloor$, where ν manages the temporal frequency of control points. We put a control point at both ends of the volume ($t = 1$, $t = T$), and the other control points are equidistantly placed between the two. The control points are often not located at time points of the video.

The space-time spline-based motion model is illustrated in Fig.2. Just for this illustration and for the sake of clarity, the motion models are computed within the two segments, foreground moving object and background, provided by the ground truth. The estimated motion model for the foreground moving object is able to capture the periodic nature of the swing motion as demonstrated by the plots of the computed motion model parameters. Also, the motion model computed in the background segment perfectly fits the camera motion. Articulated motion (woman's legs) would require multiple-motion segmentation.

Any parametric motion model could be considered. We take a polynomial expansion in the coordinates x and y of any point in the image, with one polynomial for each component of the 2D flow. The use of polynomial is a usual choice for 2D motion modelling [31], since it is easy to estimate and enables a straightforward physical interpretation. More specifically, we select the 12-parameter quadratic motion model to be able to account for continuously varying depth surface of the objects in the scene, especially for the whole background, and for complex object or camera motions. In contrast, the affine and the 8-parameter quadratic motion models assume a planar object surface. Indeed, the latter exactly corresponds to the projection in the image of the 3D rigid motion of a planar surface. It is equivalent for velocity fields to the homography transform for points. However, in presence of severe depth effects (strong depth discontinuities) and camera motion, the static scene cannot be represented by a single motion model due to motion parallax produced by static objects located in the foreground.
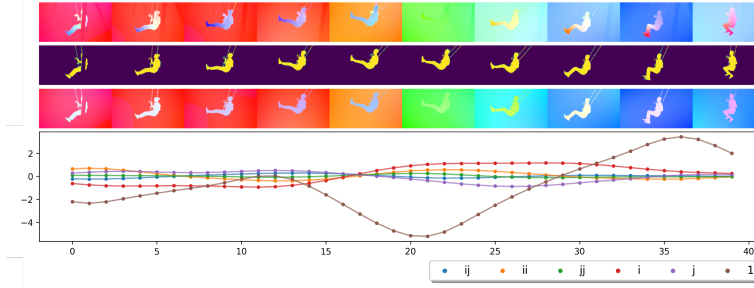
6

Figure 2: Illustration of the spatiotemporal spline-based motion model. Top group: respectively, input flows displayed with the HSV code for the *swing* video of DAVIS2016, binary segmentation ground truth, flows generated by the estimated spline-based motion models for the two segments. Bottom row: plot of the temporal evolution of the six estimated motion model parameters corresponding to the horizontal flow component for the foreground moving object. The x-axis is the video frame index.

Regarding first the spatial dimension of the motion model, the 2D flow vector yielded by the full quadratic motion model at point (x, y) writes:

$$\tilde{f}_\theta(x, y) = (\theta_1 + \theta_2 x + \theta_3 y + \theta_7 x^2 + \theta_8 xy + \theta_9 y^2,$$
$$\theta_4 + \theta_5 x + \theta_6 y + \theta_{10} x^2 + \theta_{11} xy + \theta_{12} y^2)^T. \tag{1}$$

B-Splines are modelling the evolution of the parameters of the motion model over time. The resulting motion model at a given time is then a weighted average of the close control ones, each control one being a quadratic model as in eq.(1). By denoting now $S_n(\theta_k)$ the subscript of $\tilde{f}$, we emphasize that the motion model parameters for each segment k are estimated through the B-spline functions:

$$\tilde{f}_{S_n(\theta_k)}(x, y, t) = \sum_{l=1}^{L} \tilde{f}_{\theta_{k,l}}(x, y) B_{n,l}(t), \tag{2}$$

where $B_{n,l}$ is the lth B-spline, $\theta_{k,l}$ corresponds to the lth control point of the spline, and $\tilde{f}_{\theta_{k,l}}$ is defined by eq.(1).

3.2 Loss function

We consider a volume of optical flow fields $f \in \mathbb{R}^{2 \times T \times W \times H}$ on the image grid $\Omega \in \mathbb{R}^{W \times H}$ of width W and height H, and over T temporal steps (or, in other words, a sequence of length T). We denote $f(i, t) \in \mathbb{R}^2$ the flow associated to site (or point) $i \in \Omega$ ($i = (x, y)$) at time t. We assume that we can decompose the flow as a set of K segments, each one exhibiting a coherent motion. Flow vectors within a given segment k are represented by a smooth parametric motion model parametrized by ϑ_k. Variable $z_{i,t}$ conveys the motion segmentation, $z_{i,t}^k = 1$

7

if site (i,t) belongs to segment k. $z = \{z_{i,t}^k, i \in \Omega, t = 1, \cdots T, k = 1 \cdots K\}$ and $\vartheta = \{\vartheta_k, k = 1 \cdots K\}$ are the latent variables, and f is the observed data. Following [17], we introduce an approximate distribution over segmentation and motion model parameters:

$$q(z, \vartheta|f) = q(z|f)q(\vartheta) = (\prod_{t=1}^{T} \prod_{i \in \Omega} \prod_{k=1}^{K} q(z_{i,t}^k|f))(\prod_{k}^{K} q(\vartheta_k)), \qquad (3)$$

where $q(\vartheta_k) \triangleq \delta(\theta_k)$ with δ the Dirac distribution and $q(z_{i,t}^k|f) = g_\phi(f)_{i,t}^k$. g_ϕ is our network model taking as input the optical flow volume and returning a probabilistic segmentation volume. We can also write the data-likelihood over a dataset $\mathcal{D}$ of optical flow volumes f as:

$$\log(\mathcal{D}) = \sum_{f \in D} \log p(f)$$

$$= \sum_{f \in D} \mathbb{E}_{q(z,\vartheta|f)}[\log \frac{p(f,z,\vartheta)}{q(z,\vartheta|f)}] + KL(q(z,\vartheta|f)||p(z,\vartheta|f))$$

$$\geq \sum_{f \in D} \mathbb{E}_{q(z,\vartheta|f)}[\log p(f|z,\vartheta)] - KL(q(z,\vartheta|f)||p(z,\vartheta)), \qquad (4)$$

where we get the Evidence Lower Bound (ELBO) [17].

Considering that the flow field can be represented by segment-wise parametric motion models, and assuming conditional independence, we have:

$$p(f|z,\vartheta) = \prod_{t=1}^{T} \prod_{k=1}^{K} p(f(t)|\vartheta_k, z)$$

$$\propto \prod_{t=1}^{T} \prod_{k=1}^{K} \prod_{i \in \Omega} \exp(-||f(i,t) - \tilde{f}_{S_n(\vartheta_k)}(i,t)||_1/\xi_{f,t})^{z_{i,t}^k}, \qquad (5)$$

where $\tilde{f}_{S_n(\vartheta_k)}(i,t)$ is the flow vector given by the parametric motion model of parameters ϑ_k at point (i,t) as defined in Section 3.1 through the spline approximation and $\xi_{f,t} = \sum_{i \in \Omega} ||f(i,t)||_1$ is a normalizing factor. From this likelihood, we can write the left term of the ELBO as:

$$\mathcal{L}_r(f) = -\mathbb{E}_{q(z,\vartheta|f)}[\log p(f|z,\vartheta)] = -\sum_{t,k} \mathbb{E}_q[\log p(f(t)|\vartheta_k, z)]$$

$$= \sum_{t=1}^{T} \sum_{i \in \Omega} \sum_{k=1}^{K} g_\phi(f)_{i,t}^k ||f(i,t) - \tilde{f}_{S_n(\theta_k)}(i,t)||_1/\xi_{f,t} \qquad (6)$$

The term $KL(q(z,\vartheta|f)||p(z,\vartheta)))$ in eq.(4) allows us to define a prior over the segmentation. We want to enforce the property that the segmentation labels are temporally consistent, i.e., similar between neighbouring frames. More formally,

we express a prior on z as follows, assuming point independence:

$$\forall i \in \Omega, \forall t \in \{1 \cdots T\}, z_{i,t} \in \{1 \cdots K\}$$

$$p(z) = \prod_{i \in \Omega} p(z_{i,1}) \prod_{t=2}^{T} p(z_{i,t}|z_{i,t-1}), \text{ with}$$

$$p(z_{i,t} = k | z_{i,t-1} = l) = \frac{1}{M} \exp(\alpha \mathbb{1}[k=l] + \beta \mathbb{1}[k \neq l]) \tag{7}$$

$$\text{and } p(z_{i,1} = k) = \frac{1}{K}, \tag{8}$$

where M is a normalizing factor. We introduce the more compact notation $g_{i,t}^k = g_\phi(f)_{i,t}^k$. Using an uninformative prior on θ, the KL term is given by:

$$KL[q(z,\theta|f)||p(z,\theta)] = KL[q(z|f)||p(z)]$$
$$= \mathbb{E}_q[\log q(z|f)] - \mathbb{E}_q[\log p(z)]. \tag{9}$$

We then infer the two terms of eq.(9)

$$\mathbb{E}_q[\log q(z|f)] =$$
$$\sum_{i \in \Omega} \sum_{t=2}^{T} \sum_{k=1}^{K} \sum_{l=1}^{L} g_{i,t}^k g_{i,t-1}^l (\alpha \mathbb{1}[k=l] + \beta \mathbb{1}[k \neq l]) + c, \tag{10}$$

where $c = -|\Omega|(\log K + T \log M)$ is a constant. We set $\beta = 0$, and knowing that $\alpha > 0$, we have:

$$\mathbb{E}_q[\log p(z)] = \alpha \sum_{i \in \Omega} \sum_{t=2}^{T} \sum_{k=1}^{K} g_{i,t}^k g_{i,t-1}^k + c. \tag{11}$$

Since $2 * g_{i,t}^k g_{i,t-1}^k = -(g_{i,t}^k - g_{i,t-1}^k)^2 + (g_{i,t}^k)^2 + (g_{i,t-1}^k)^2$, after simple calculations, we get:

$$KL[q(z|f)||p(z)] = \frac{\alpha}{2} \sum_{i \in \Omega} \sum_{t=2}^{T} \sum_{k=1}^{K} (g_{i,t}^k - g_{i,t-1}^k)^2$$

$$+ \sum_{i \in \Omega} \sum_{t=1}^{T} \sum_{k=1}^{K} g_{i,t}^k * (\log(g_{i,t}^k) - \frac{\alpha}{2} * g_{i,t}^k)$$

$$- \frac{\alpha}{2} \sum_{i \in \Omega} \sum_{t=2}^{T-1} \sum_{k=1}^{K} (g_{i,t}^k)^2) + c. \tag{12}$$

Since $\forall x \in [0,1], -\frac{\alpha}{2} \leq x(\log x - \frac{\alpha}{2}x) \leq 0$, we can infer:

$$KL[q(z|f)||p(z)] \leq \frac{\alpha}{2} \sum_{i,t,k} (g_{i,t}^k - g_{i,t-1}^k)^2 + c$$

As $\forall x, y \in [0,1]^2, |x - y| \geq (x - y)^2$, we end up with the term of temporal consistency in the loss function we introduced in ST-MS [2]. Thus, we have provided a mathematical development to motivate it. More precisely, it writes:

$$\mathcal{L}_c(f) = \frac{1}{2K|\Omega|} \sum_{i \in \Omega} \sum_{t=2}^{T} \sum_{k=1}^{K} |g_\phi(f)_{i,t}^k - g_\phi(f)_{i,t-1}^k|. \tag{13}$$

The Eulerian temporal constraint expressed by eq.(13) does not mean that the site should be static between two successive frames, but it holds as long as the site lies within the successive projections of a given moving part so that it remains in the same mask. However, it does not make sense on occluded or disoccluded areas. This justifies the use of the L1 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.(13) that exhibit a large temporal flow difference. More precisely, for each input flow sequence, we set a threshold λ so that a quantile η of sites i is discarded as follows:

$$p(\|f(i, t + 1) - f(i, t)\|_1 \geq \lambda) \leq \eta. \tag{14}$$

In practice, we take $\eta = 1\%$ with an empirical distribution for p in eq.(14). 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 finally defined by:

$$\sum_{f \in \mathcal{D}} \mathcal{L}(f, \phi, \theta) = \sum_{f \in \mathcal{D}} \mathcal{L}_r(f, \phi, \theta) + \gamma \mathcal{L}_c(f, \phi). \tag{15}$$

We set $\gamma = 1$. The training procedure alternatively updates θ and ϕ with ϵ as learning rate:

$$\text{for } f \in D : \ \theta^* = \arg\min_\theta \mathcal{L}(f, \phi, \theta)(f) \ ; \tag{16}$$

$$\phi_{t+1} = \phi_t - \epsilon \nabla_\phi \mathcal{L}(f, \phi, \theta^*) \tag{17}$$

3.3   Network architecture

The overall architecture of our unsupervised multiple motion segmentation framework is illustrated in Fig.1. It includes two main modules. The first one, taking the flow volume as input, is a 3D U-net [39]. The latent code augmented with embedding positions is directed to the transformer decoder. Then, by cross-attention, the output formed by the volume of segmentation masks is produced. The training of the overall architecture is based on the minimization of the loss function defined in eq.(15), while the motion model parameters of the segments are given by the B-spline estimation. Temporal consistency is provided by the loss function and the space-time motion models.

4 Implementation

4.1 Implementation details

Following [7, 21, 29, 2, 49], we adopt the RAFT method [41] to compute the optical flow fields. More specifically, we use the RAFT version trained on the MPI Sintel dataset [3]. We downsample the computed flow fields to feed the network with $128{\times}224$ vector fields. The output segmentation maps are upsampled to the initial frame size for evaluation. Thus, we can perform efficient training and inference stages. We typically take flow volumes of temporal length $T = 9$ at training time. However, at test time, we can process flow volumes of larger temporal length, and even the flow volume of the whole sequence. To compute the spatio-temporal parametric motion models, x and y coordinates are normalized within $[-1, 1]$, and a similar normalization is applied to the t coordinate. The full 12-parameter quadratic motion model is used in all the experiments. We take $\nu = 3$ for the frequency factor in the B-spline approximation. We select for each point (x, y) the segment $\hat{k}$ with the highest prediction. In eq.(13), we set λ as the 99th quantile of the flow differences over the flow volume.

We use Adam optimizer with the following strategy on the learning rate ϵ to train the network (3D U-net and transformer), inspired from the warmup-decay strategy [47]. We linearly increase it from 0 to $1e - 4$ for 20 epochs, then, we divide it by two every 40 epochs. The estimation of the motion model parameters through the B-spline approximation at training time is achieved with the Pytorch implementation of L-BGFS [22]. Indeed, the parametric models induced over time by the splines are a linear combination of quadratic parametric models (see eq.(2)). Thus, we can optimize eq.(6) to obtain the argmin in eq.(16), which is a convex problem.

Our LT-MS method is very efficient at test time (code made available upon publication). The computational time amounts on average to 210 *fps* on a GPU P100, that is, twice as fast as the ST-MS method [2]. It is due to the long flow sequence given as input to the network, which allows for parallelisation of some heavy computations. In addition, our LT-MS architecture remains lightweight, since it combines only three U-net layers and a transformer decoder on the downsampled feature space. Let us also stress that our LT-MS method does not involve any post-processing at all.

4.2 Data augmentation and network training

We introduced two types of data augmentation dedicated to optical flow. First, we add a 2D global flow to the input flow similarily as we did in the EM [29] and ST-MS [2] methods. However, for LT-MS, the 2D global flow is given by a full spline-based spatio-temporal motion model whose parameters are chosen at random. The same 2D global flow is added to the flow fields of a given input volume. It allows us to mimic diverse camera motions, enforcing that the motion segments are independent of the 2D added global flow, as justified in [29]. In addition, we corrupt a few flows out of the nine input ones. Thus, we

Ablation / Dataset	DAVIS2016	FBMS59	SegTrackV2
Full Model LT-MS-K4	72.4	58.2	60.4
Unet3D only	71.3	55.5	57.3
No consistency term $\mathcal{L}_c$	71.0	55.5	57.5
Polynomial space-time model	69.8	54.5	56.6

Table 1: Ablation study for three main components of our method LT-MS ($K = 4$) on DAVIS2016, FBMS59 and SegTrackV2. Only one model component is modified at a time. Input sequence length is 120 (in practice, it comprises the whole video for the DAVIS2016 dataset). The metrics used is the Jaccard index $\mathcal{J}$. All evaluations are performed on the binary ground truth for the three datasets as explained in Section 5.5.

simulate poorly estimated flow fields at some time instants, and the temporal consistency should overcome it.

Our motion segmentation method is fully unsupervised. We never use any manual segmentation annotation. We train our model using the FlyingThings3D (FT3D) dataset [27]. Our (demanding) objective is to demonstrate the high generalization power of our model. We train our model once for all on the optical flow fields computed on FT3D, and then, apply it on new unseen datasets. For hyperparameter setting, we select the stopping epoch from the loss function evaluated on the DAVIS2016 training set. We trained on a single Quadro RTX 6000 for 77h with a batch size of 5 on 32k flow fields computed on FT3D.

5 Experimental Results

Our LT-MS method is above all a long-term multiple motion segmentation method. However, in order to quantitatively assess its performance and in absence of OFS benchmarks, we need to resort to VOS benchmarks. We have carried out several types of experiments on four datasets: DAVIS2016[1] [35], SegTrackV2[2] [20], FBMS59 [32], and DAVIS2017-motion [47]. However, most VOS benchmarks are concerned with the segmentation of the primary moving object in the foreground, which means a binary ground truth. This led us first to test our LT-MS method in a two-mask ($K = 2$) configuration, as discussed in Section 5.3, and reported in Table 2. Then, we have performed multiple motion segmentation evaluation, especially on DAVIS2017-motion, as commented in Section 5.4 and Table 3. Finally, we have assessed the long-term impact of LT-MS both on a quantitative (Table 4, Section 5.5) and qualitative (Section 5.6) basis.

---

[1]https://davischallenge.org/index.html
[2]https://paperswithcode.com/dataset/segtrack-v2-1

Figure 3: Three groups of qualitative results regarding the ablation of the temporal-consistency loss term ($K = 4$). They respectively correspond to the *worm* video of SegTrackV2, the *dogs01* video of FBMS59, and the *car-roundabout* videos of DAVIS2016. For each group, the first row contains sample images with the segmentation ground-truth overlaid in yellow, the second row displays the input flows, the third and fourth rows show the predicted motion segmentations, respectively without and with the temporal consistency loss term. Clearly, this model component allows us to get far more consistent segments over time.

5.1 Datasets

DAVIS2016 comprises 50 videos (for a total of 3455 frames), depicting diverse moving objects. It is split in a training set of 30 videos and a validation set of 20 videos. Only the primary moving object is annotated in the ground truth. The criteria for evaluation on this dataset are the Jaccard score $\mathcal{J}$ (intersection-over-union), and the contour accuracy score $\mathcal{F}$, the higher, the better for both. FBMS59 includes 59 videos (720 annotated frames), and SegTrackV2 14 videos (1066 annotated frames). Both mostly involve one foreground moving object, but sometimes a couple of moving objects. For FBMS59, we use the 30 sequences of the validation set. Although annotations may comprise multiple objects, the VOS community exploits SegTrackV2 similarily as DAVIS2016, considering a binary ground-truth by grouping moving objects in the foreground. We follow this practice. However, we provide a multi-mask evaluation for FBMS59 as

13

other works did.

DAVIS2017 is an extension of DAVIS2016, containing 90 videos. It includes additional videos with multiple moving objects, and it provides multiple-segment annotations for the ground truth. It is split into 60 videos for training and 30 for evaluation. DAVIS2017-motion is a curated version of the DAVIS2017 dataset proposed in [47] for a fair evaluation of methods based on optical flow only. Connected objects exhibiting the same motion are merged in the ground truth for evaluation. We proceed to the evaluation with the official algorithm that involves a Hungarian matching process.

5.2 Ablation study

We have conducted an ablation study to assess three main components of our long-term multiple-motion segmentation method LT-MS. To this end, we took four masks ($K = 4$), and call LT-MS-K4 this version of our LT-MS method. These three components are mainly related to the temporal dimension of the problem. We have changed one component at a time as specified hereafter. We use the polynomial space-time quadratic motion model of ST-MS [2] instead of the space-time motion model based on B-splines over the input sequence. We omit the consistency term $\mathcal{L}_c$ in the loss function. We just take the convnet without the transformer decoder. All the ablation experiments were run on the three datasets, DAVIS2016, FBMS59, SegTrackV2. Results are collected in Table 1. We performed them for the input sequence length of 120 (in practice, the whole video for the DAVIS2016 dataset).

We can observe that the three ablations have almost the same impact on the performance. The three corresponding model components, i.e., the spline-based motion model, the temporal-consistency loss term, and the transformer decoder are thus all beneficial in similar proportions. They are able to handle the temporal dimension of the problem and the temporal motion evolution along the sequence in a compelling way. The contributions of these three components are even more significant for the FBMS59 and the SegTrackV2 datasets. In fact, the dynamic content of the majority of DAVIS2016 videos, along with performance scores calculated on a frame basis on these videos, do not allow to fully appreciate the contributions of the three model components. Yet, they can be highlighted by visualizing results on some videos of DAVIS2016, as shown in Fig.3.

5.3 Quantitative and comparative binary evaluation

We first report in Table 2 the quantitative evaluation on DAVIS2016 and Seg-TrackV2, performed on the binary ground truth (foreground moving object *vs* background) for the two datasets. This is the most frequent evaluation reported in the VOS literature. Therefore, we first apply our LT-MS method with two masks ($K = 2$), denoted LT-MS-K2. Since our OFS does not establish the assignment of the two segments to respectively the background and the primary

14

Methods	Training	Input	DAVIS2016 $\mathcal{J}\uparrow$	SegTrackV2 $\mathcal{J}\uparrow$
Ours LT-MS-K2	Unsupervised	Flow	70.7	58.9
EM [29]			69.3	55.5
MoSeg [49]			68.3	58.6
FTS [34]			55.8	47.8
TIS$_0$ [12]			56.2	-
OCLR [47] (flow only)	Additional Supervision		72.1	67.6
DyStab - Dyn [51]			62.4	40.0
GWM [7]	Unsupervised	RGB (Flow in loss)	79.5	78.3
RCF [21]			80.9	76.7
AMD [23]			57.8	57.0
MOD [10]			73.9	62.2
DivA(4)† [1]		RGB & Flow	72.4	64.6
IKE [13]			76.0	66.4
TIS$_s$ [12]			62.6	-
CIS - No Post [50]			59.2	45.6
DS [52]	Optimisation at test time		79.1	72.1
GO [36]	Optimisation at test time & Additional Supervision		80.2	74.9

Table 2: Results obtained with our LT-MS-K2 method (with two masks) on DAVIS2016 and SegTrackV2. An input flow sequence length of 120 is considered. We include comparison with different categories of methods (scores from cited articles). The most relevant comparison is with the category of unsupervised methods taking only optical flow as input. All scores correspond to evaluation on the binary ground-truth. The Jaccard index $\mathcal{J}$ is the intersection over union between the extracted segments and the ground truth, the higher the better. Performance is assessed by the average score over all samples for SegTrackV2. For DAVIS2016, the overall score is given by the average of sequence scores. †DivA somehow uses RGB input since its conditional decoder leverages input images.

moving object, we rely on a simple heuristic designating the biggest segment as the background one.

Let us recall that our actual objective is the optical flow segmentation (OFS) task, even if we evaluate our method on VOS benchmarks in absence of OFS benchmarks. Since the two VOS benchmarks deal with the segmentation of the primary object moving in the foreground, it may occur discrepancies with OFS, which negatively impacts the scores for OFS methods. If wrong w.r.t. VOS ground truth, the segmentation of additional moving parts actually makes sense from the OFS standpoint, such as moving objects in the background, subparts of moving articulated bodies, or static foreground objects exhibiting motion parallax with a moving camera. Therefore, the comparative evaluation regarding OFS must be restricted to unsupervised methods taking only optical flow as input, namely, the top part of Table 2. Nevertheless, for a more comprehensive view, we also include results of VOS methods that leverage both RGB images and flow as input, in the bottom part of Table 2.

We evaluate LT-MS-K2 on a sequence basis, not on a frame basis as done by the other methods. This is more demanding, but we have done so to be consis-

tent with the long-term intrinsic property of our method. We took a sequence length $T = 120$ for DAVIS2016 and SegTrackV2. Table 2 also collects results obtained by other existing unsupervised methods also with two masks, when available on these benchmarks. We keep the categorization that we proposed in [29] regarding input and training. Nevertheless, we have added a category w.r.t. the network input for four recent methods, [7, 10, 21, 23], that only use RGB images as imput at test time, the optical flow being only involved in the loss function. OCLR leverages additional supervision, since it relies on human-annotated sprites to include realistic shapes in the computer-generated data used at training time. We consider the OCLR version taking only optical flow as input. The same holds for the DyStab method, the DyStab-Dyn version is considered. The post-processing added to the CIS method [50], based on Conditional Random Fields (CRF), is an heavy one, which has led most authors to retain only the version without post-processing for a fair comparison. We also include a separate category for two recent unsupervised methods, DS [52] and GO [36], where optimisation is performed at test time.

As shown in Table 2, LT-MS-K2 provides the best results on the two datasets in the category of unsupervised methods based on optical flow only. Besides, we report a repeatability study in the Appendix. OCLR and DivA demonstrate better performance, especially on the SegTrackV2 dataset. However, as afore-mentioned, OCLR involves additional supervision, while DivA leverages RGB images in its conditional decoder. In addition, DivA, along with MoSeg and CIS methods, takes multi-step flows as input between t and in turn $t+1$, $t+2$, $t-1$, $t-2$, and averages the four corresponding predictions to get the final result.

Failure cases may occur for very complex motion (e.g., ripples on the water), or a particularly strong temporal evolution (as an object approaching the camera from afar quickly and closely). Overall, temporal consistency is properly handled over long periods by our LT-MS method. We want to stress that our method is the only one providing *by design* a coherent segmentation over the sequence, which is a significant added value for downstream tasks, like tracking or dynamic scene interpretation.

5.4 Multi-segment evaluation

We now focus on the multiple motion segmentation for the DAVIS2017-motion and FBMS59 datasets. Since multiple-motion segmentation is harder than the binary motion segmentation that only involves moving foreground *vs* back-ground, accuracy scores are expected to decrease for all methods. In Table 3, we report comparative results on the DAVIS2017-motion dataset using the official evaluation procedure, and on FBMS59. As for the other methods, we performed the multiple motion segmentation with three masks ($K = 3$), and we name this version LT-MS-K3. To this end, we finetuned the LT-MS-K4 network on the DAVIS2016 training set with now three masks ($K = 3$). The resulting performance on DAVIS2017-motion is slightly better than ST-MS and far better than MoSeg. Let us recall that OCLR involves additional supervision. Regarding FBMS59, we report multimask segmentation results with two

Methods	DAVIS2017-motion			FBMS59	
	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	bIoU	Linear Assign.
Ours LT-MS-K3	42.2	39.3	45.0	58.4	47.2
ST-MS [2]	42.0	38.8	45.2	-	-
MoSeg [49]	35.8	38.4	33.2	-	-
OCLR [47]	55.1	54.5	55.7	-	-
DivA [1]	-	-	-	42.0	-

Table 3: Multi-segment evaluation, all compared methods using three masks. Evaluation is performed on the video as a whole following the official DAVIS2017 scheme. Reported scores are the average of the individual video scores. The $\mathcal{F}$ metrics focuses on segment boundary accuracy. $\mathcal{J}\&\mathcal{F}$ is the mean of the two. For FBMS59, the evaluation metrics are bootstrap IoU (bIoU) [1] where each ground-truth mask is mapped to the most likely predicted segment (performed at the frame level), and linear assignment that is a bilinear mapping between the ground-truth and the predicted segments at the sequence level (as in the official DAVIS2017 evaluation).
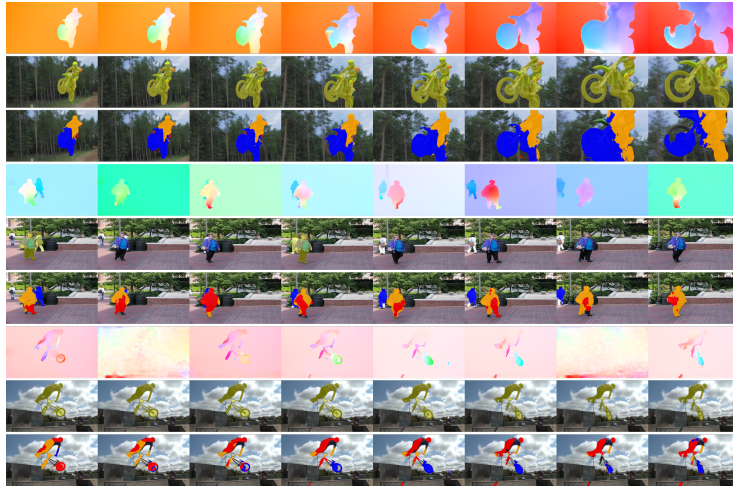

Figure 4: Results obtained with our LT-MS-K4 method for *motocross-jump* from DAVIS2016, *people02* from FBMS59 and *bmx* from SegTrackV2. For each group, the first row samples flow fields (HSV color code) corresponding to the processed video. The second row contains the corresponding images of the video, where ground-truth of the moving object is overlaid in yellow. The third row shows the motion segments provided by our LT-MS-K4 method with one colour per segment. We constantly adopt the same color set for the three masks corresponding to moving objects (blue, red and orange), and we let the image for the background mask.

different metrics. Our LT-MS method outperforms by a large margin DivA [1] that also attempted this multi-segment evaluation.

Methods	DAVIS2016		FBMS59		SegTrackV2	
Evaluation basis	f	s	f	s	f	s
LT-MS-K4	75.0	72.4	60.1	58.2	63.4	60.4
ST-MS [2]	71.4	70.2	56.3	55.2	55.3	53.7
EM [29]	77.1	68.0	66.1	51.9	62.5	50.6

Table 4: Accuracy and stability in the evaluation, on DAVIS2016, FBMS59 and SegTrackV2, of our three multiple motion segmentation models, LT-MS, ST-MS and EM, all with $K = 4$, on a frame-based (f) basis and on a sequence-based (s) basis for a sequence length of 120. Reported scores correspond to the $\mathcal{J}$ metrics.

5.5 Accuracy *vs* stability

Since the frame-based quantitative evaluation reported in Section 5.3 cannot fully reflect the contributions of our long-term motion segmentation model, we have carried out complementary experiments. The idea is to somehow cross-validate accuracy and stability. To this end, we compare our three models, the frame-based EM model, the short-term ST-MS model, and the long-term LT-MS model. We proceed in two ways. First, we evaluate all three models on a frame-based basis, then, on a sequence-based basis for a sequence length of 120. By sequence length, we mean the length of the flow sequence taken as input for LT-MS or the output length of the temporal linkage (post-processing stage) for ST-MS.

Results obtained on DAVIS2016, FBMS59 and SegTrackV2 are shown in Table 4 for a binary ground truth. In the Appendix, we explain how we select the segments for the binary evaluation from the set of segments delivered by our multiple motion segmentation methods, when using a number of masks $K > 2$. It is not the same procedure as the one used at the end of Section 5.3, which explains the scores obtained for LT-MS-K4 compared to Table 2. Here, we can select the right segments by simply comparing with the ground-truth segments, since it is a question of internal comparison. As expected, the frame-based evaluation favours a frame-based multiple motion segmentation, and overall, the accuracy slightly decreases with the sequence length since it is more demanding to maintain the same labelling, and then to preserve accuracy, over a sequence. In contrast, the sequence-based evaluation highlights the temporal stability brought by ST-MS and with a significant larger margin by LT-MS, in contrast to EM whose performance significantly drops when evaluated on a sequence basis.

5.6 Qualitative visual evaluation

Visual results are provided for LT-MS-K2 in the Appendix. Fig.4 contains a set of visual results to demonstrate how our LT-MS-K4 method behaves on different situations. We display three result samples obtained on different videos of the

18

benchmarks. We can observe that the motion segments are globally accurate. Since our method involves several masks, we can properly handle articulated motions (*people02, bmx*), deal with the presence of several moving objects in the scene (*people02*), separate the rider from the bike or motorbike (*bmx, motocross-jump*). Since our method enforces temporal consistency, it can deal with errors in optical flow estimation or with objects that momentarily stop moving (*bmx*).

Additional visual results of our LT-MS-K4 are collected in Fig.5. Again, we can properly cope with articulated motions (*monkey*), with the presence of several moving objects in the scene (*hummingbird, goats01*), or accomodate motion parallax (*libby*). Finally, in Fig.6, we collect visual results on two videos of the SegTrackV2 dataset, from top to bottom, *birdfall* and *bird-of-paradise* videos. Results demontrate the ability of our long-term segmentation method to recover correct segmentation even in case of flow fields wrongly estimated for some frames. By the way, this ability was also demonstrated by the greater accuracy (Tables 2 and 4) obtained on SegTrackV2 dataset by LT-MS compared to EM that operates on a frame basis.

6   Conclusion

We have designed an original transformer-based unsupervised method for segmenting multiple motions over a sequence, without any postprocessing. It performs it in one go, which is a more demanding goal, but provides decisive added value for downstream tasks like tracking or dynamic scene interpretation. Our LT-MS method leverages the ELBO framework for the loss function, and fully acknowledges the temporal dimension of the motion segmentation problem. Our method delivers a stable and consistent OF segmentation throughout long video sequences. It introduces at training time B-splines spatio-temporal parametric motion models over space-time volumes, and a loss term expressing temporal consistency over successive masks while taking care of occlusions. Our transformer-based network can be applied at test time to input volumes of any time length. It can accomodate different choices on the number K of masks with a simple fine-tuning step. Experimental results on several datasets demonstrate the efficiency and the accuracy of our LT-MS method by providing competitive results on several datasets. In addition, it is very fast at test time.

References

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

[2] P. Bideau, R. Menon, and E. Learned-Miller. MoA-Net: Self-supervised motion segmentation. In *European Conference on Computer Vision Workshops (EC-CVW)*, 2018.

[3] 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.
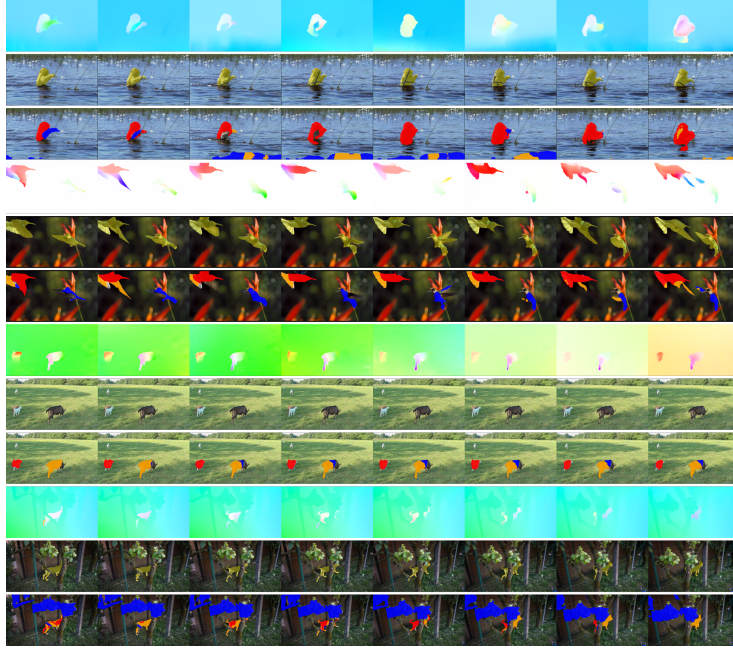
Figure 5: Results obtained with our LT-MS-K4 method ($K = 4$). Four groups of results are displayed: *monkey*, *hummingbird* from SegTrackV2, *goats01* from FBMS59 and *libby* from DAVIS2016. For each group, the first row samples successive flow fields (HSV color code) corresponding to the processed video. The second row contains the corresponding images of the video, where the ground-truth of the moving object is overlaid in yellow (when available at that frame). The third row shows the motion segments provided by our LT-MS-K4 method with one colour per segment. For all the results, we adopt the same color set for the three masks corresponding to the moving objects (blue, red and orange), and we let the background image for the background mask.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Int. Conf. on Computer Vision (ICCV)*, Oct. 2021.

[5] B. Cheng, A. G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

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

[7] 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 Conf. (BMVC)*, 2022.

[8] D. Cremers and S. Soatto. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, 2005.
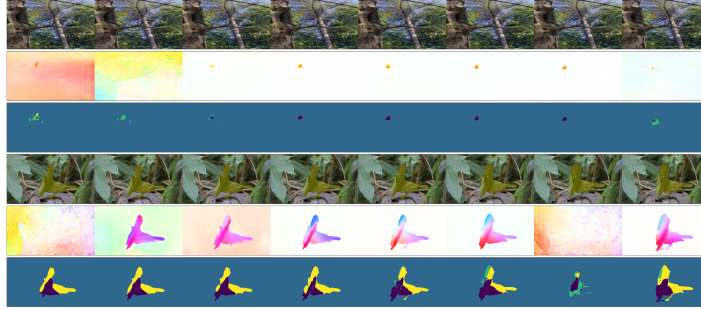
Figure 6: Illustration of the temporal consistency provided by our LT-MS-K4 method on two videos of SegTrackV2, *birdfall*, *bird-of-paradise*, For each group, the first row contains the video images with the ground truth overlaid in yellow; the second row depicts the corresponding flow fields represented with the HSV code while normalized independently from each other; the third row provides the predicted segmentation.

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

[10] 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.

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

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

[13] E. Haller, A. M. Florea, and M. Leordeanu. Iterative knowledge exchange between deep learning and space-time spectral clustering for unsupervised segmentation in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7638–7656, 2021.

[14] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87-110, January 2023.

[15] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998.

[16] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.

[17] D.P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307-392, 2019.

[18] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Asian Conf. on Computer Vision (ACCV)*, Kyoto, 2020.

[19] D. Lao, Z. Hu, F. Locatello, Y. Yang, and S. Soatto. Divided attention: Unsupervised multi-object discovery with contextually separated slots. *arXiv:2304.01430*, April 2023.

[20] 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 (ICCV)*, Sydney, 2013.

[21] L. Lian, Z. Wu, S. X. Yu. Bootstrapping objectness from videos by relaxed common fate and visual grouping *Conference on Computer Vision and Pattern Recognition*, Vancouver, June 2023.

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

[23] R. Liu, Z. Wu, S. X. Yu, and S. Lin. The Emergence of objectness: Learning zero-shot segmentation from videos. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[24] 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.

[25] A. Mahendran, J. Thewlis, and A. Vedaldi. Self-supervised segmentation by grouping optical flow. In *European Conf. on Computer Vision Workshops (ECCVW)*, Munich, 2018.

[26] 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.

[27] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] E. Meunier and P. Bouthemy. Unsupervised computation of salient motion maps from the interpretation of a frame-based classification network, In *British Machine Vision Conference (BMVC)*, 2021.

[29] E. Meunier, A. Badoual and P. Bouthemy. EM-driven unsupervised learning for efficient motion segmentation. In *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 45(4):4462-4473, April 2023.

[30] E. Meunier and P. Bouthemy. Unsupervised space-time network for temporally-consistent segmentation of multiple motions. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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

[32] 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.

[33] 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.

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

[35] F. Perazzi, J. Pont-Tuset, B. Mc Williams, 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.

[36] G. Ponimatkin, N. Samet, Y. Xiao, Y. Du, R. Marlet, and V. Lepetit. A simple and powerful global optimization for unsupervised video object segmentation. In *Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, Jan. 2023.

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

[38] 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)*, 2019.

[39] 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.

[40] 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.

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

[42] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2017.

[43] Z. Tu, W. Xie, D. Zhang, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan. A survey of variational and CNN-based optical flow techniques. *Signal Proc.: Image Communication*, 72:9-24, March 2019.

[44] M. Unser. Splines: A perfect fit for signal and image processing. *Signal Processing Magazine*, 16(6):22-38, November 1999.

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

[46] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1644-1659, October 2005.

[47] J. Xie, W. Xie, and A. Zisserman. Segmenting moving objects via an object-centric layered representation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[48] 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.

[49] 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)*, Oct. 2021.

[50] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto. Unsupervised moving object detection via contextual information separation. In *Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2019.

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

[52] V. Ye, Z. Li, R. Tucker, A. Kanazawa, and N. Snavely. Deformable sprites for unsupervised video decomposition. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, 2022.

[53] L. Zappella, X. Llado, and J. Salvi Motion segmentation: A review. *Frontiers in Artificial Intelligence and Applic.*, 184:398-407, Jan. 2008.

[54] T. Zhou, F. Porikli, D.J. Crandall, L. Van Gool, and W. Wang. A survey on deep learning technique for video segmentation. *IEEE Trans. on Pattern Anal. and Mach. Intelligence*, 45(6):7099-7122, 2023.

7 Appendix

7.1 Loss function

7.1.1 Experimental validation of the approximation of the temporal consistency loss term.

We have tried to assess the impact of the approximation made in the definition of the temporal consistency term of the loss function as developed in Section 3.2 of the main text. To this end, we have compared the full LT-MS-K4 model with a version where the temporal consistency term is issued from eq.(11) in the main text before the approximation. More specifically, we consider the alternate loss function given by:

$$
\mathcal{L}(f) = \sum_{t=1}^{T} \sum_{i \in \Omega} \sum_{k=1}^{K} g_\phi(f)_{i,t}^k ||f(i,t) - \tilde{f}_{S_n(\theta_k)}(i,t)||_1 / \xi_{f,t}
$$

$$
- \gamma_1 \sum_{i \in \Omega} \sum_{t=2}^{T} \sum_{k=1}^{K} g_{i,t}^k g_{i,t-1}^k + \gamma_2 \sum_{t=1}^{T} \sum_{i \in \Omega} \sum_{k=1}^{K} g_\phi(f)_{i,t}^k \log g_\phi(f)_{i,t}^k, \tag{18}
$$

where $\xi_{f,t} = \sum_{i \in \Omega} ||f(i,t)||_1$. In practice, we set $\gamma_1 = 0.5$ and $\gamma_2 = 0.01$.

We carried out the comparison with an input flow sequence length of 120 on th ethree datasets DAVIS2016, FBMS59 and SegTrackV2. We obtained the results reported in Table 7.1.1. Results are almost on par for DAVIS2016. The increased performance of the full model LT-MS-K4 is slightly higher for the other two datasets, most likely because the L1 norm used in our loss is more robust to error noise in the network predictions during training.

7.2 Addition to the ablation study

Fig.11 highlights the contribution of the spline-based motion model on the *dog* video, and its obvious ability to handle motions that do not vary uniformly, as the erratic movement of the dog.

7.3 Segment selection for binary evaluation

VOS and OFS are close but not identical tasks. However, we use VOS benchmarks, since no OFS benchmarks are available. Let us recall that 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 usually done for the DAVIS2016, SegTrackV2 and FBMS59 datasets.

Since we deal with multiple-motion segmentation, i.e., K segments with $k > 2$, we have to group them into two clusters corresponding to the foreground moving object on one side and the background on the other side. We proceed as in the ST-MS method [2] on a sequence basis. Note that this selection is used solely for results presented in Table 4. of the main paper.

For the sequence, we select the subset $\mathcal{S}^*$ of labels (that is, of segments) to constitute the predicted foreground, using the binary ground-truth g_t. The selected label subset $\mathcal{S}^*$ is given by:

$$\mathcal{S}^* = \arg \max_{\mathcal{S} \subset \mathcal{P}(\{1,..,K\})} \sum_{t=0}^{T} J(\bigcup_{l \in \mathcal{S}} l_t, g_t), \tag{19}$$

where J is the IoU score between two segments, $\mathcal{P}(\{1,..,K\})$ is the partition of the labels in the sequence, and the binary masks l_t correspond to the label mask m_t. Once we have selected the subset $\mathcal{S}^*$ of labels, we can use it to build our binary segmentation $\{s_t, t = 0,..,T\}$ on the whole sequence for evaluation.

Let us recall that for the case $K = 2$, we just pick the smallest of the two masks to compare it to the ground truth mask of the primary moving object.

7.4 Multi-mask evaluation

Regarding the evaluation on the DAVIS2017-motion dataset whose ground truth is not binary, we use the official evaluation script (which performs Hungarian matching over the sequence) to associate each predicted segment k with the ground-truth annotations, knowing that we have considered $K = 3$ masks for this experimentation as the other methods we compare with. We also apply it to FBMS59 for the multiple segment setting.

Prior / Dataset	DAVIS2016	FBMS59	SegTrackV2
Full Model LT-MS-K4	72.4	58.2	60.4
Modified model of eq.(18)	71.6	55.6	57.4

Table 5: Comparison of the two temporal consistency loss terms

```
sequences_iou = []
for sequence in dataset :
    objects_iou = []
    for frame in sequence :
      for x in
        gt_annotations_including_background :
            maxiou = 0
            for y in predicted_segments :
             maxiou = max(maxiou, iou(x, y))
            objects_iou.append(maxiou)
    sequence_iou.append(mean(objects_iou))
dataset_iou = mean(sequences_iou)
```

Figure 7: Evaluation Boostrap IoU

In addition, we implement the "Boostrap IoU" score to evaluate multi-mask segmentation in FBMS59, as described in [1][3] The core idea is to match each ground-truth segment to the most likely segment (i.e., with the one of highest IoU). Let us note that this evaluation includes the background segment, since background is not identified in the FBMS59 multi-mask annotation. The algorithm for evaluation we used is described in figure 7.

For the linear assignment score, we find the best bipartite sequence-level match between the ground truth and the predictions. Similar to the bIoU evaluation, we compute the score with all the labels of the ground truth. This evaluation is more demanding, because it forces a one-to-one match between the prediction and the masks at the sequence level.

7.5 Qualitative results for LT-MS-K2

We report in Fig.8 visual results obtained with our LT-MS-K2 method with two masks ($K = 2$). LT-MS-K2 is able to accurately segment the primary moving object throughout the sequence for the videos *motocross-jump, people2, monkey, hummingbird, goats01* and *libby*. In *bmx* video, it segments only part of the whole formed by the cyclist and his bike, but there are different motions within this articulated moving set, and one single motion model cannot handle all of them. These results can be compared with those obtained by LT-MS-K4 on the same videos and reported in Figures 4 and 5 of the main text.

7.6 Repeatability

With the introduction of the transformer decoder, we experimentally found that the convergence of the network depends on the weights initialization, and that the same network and loss configuration can yield different results at test time. In Fig.9, we show that our unsupervised loss on a held-out validation set is a good indicator of the network performance at test time. This is a critical point for model and hyperparameter selection, since we do not have access to the ground truth at training time (with our fully unsupervised scenario), and thus we cannot evaluate model performance. Fig.10 plots the model performance after training as a function of the initialization budget.

---

[3]We use the multi-label ground truth provided by Dong Lao, first author of [1].
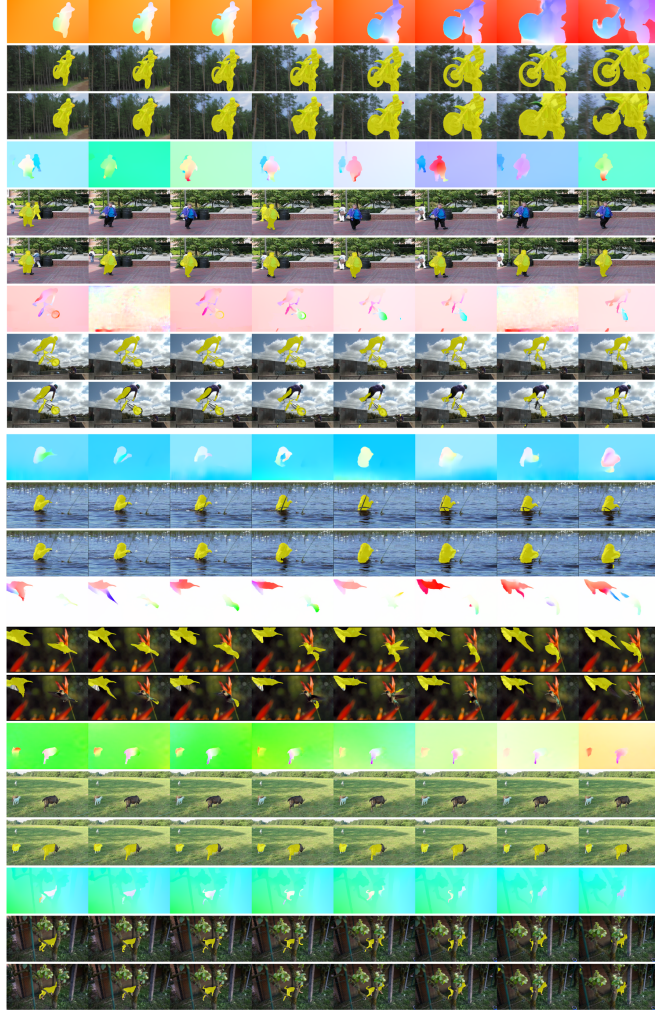
Figure 8: Results obtained with our LT-MS-K2 method for, from top to bottom, *motocross-jump* from DAVIS2016, *people02* from FBMS59, *bmx* from SegTrackV2, *monkey*, *hummingbird* both from SegTrackV2, *goats01* from FBMS59 and *libby* from DAVIS2016. For each group, the first row samples flow fields (HSV color code) corresponding to the processed video. The second row contains the corresponding images of the video, where the binary ground-truth of the primary moving object is overlaid in yellow (when available, for FBMS59 the ground truth is not available in every frame). The third row shows the binary segmentation provided by our LT-MS-K2 method. We let the image content for the background mask.
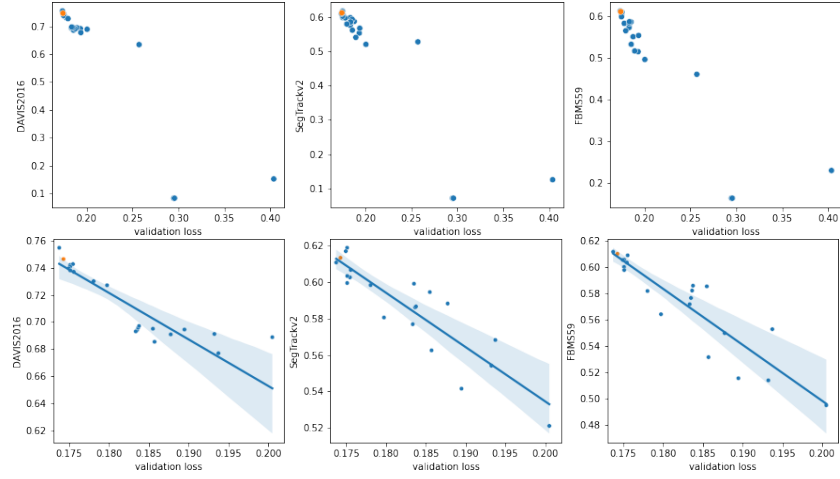
Figure 9: Model scores depending on initialization. Each dot represents a trained model. The x-axis represents the validation loss on the DAVIS2016 train dataset, and the y-axis the performance on the evaluation dataset. The top row includes all models and the bottom row excludes models that diverged during training (validation loss > 0.25). The bottom row displays the linear relationship between validation loss and evaluation score. The model whose results are reported in the main text is represented as an orange dot.
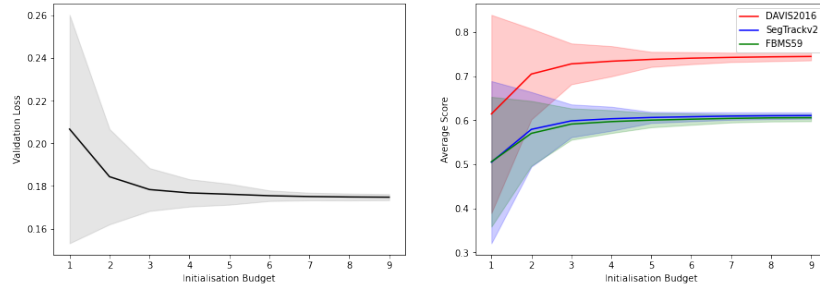

Figure 10: Evolution of the model performance as a function of the initialization budget. Left : the validation loss associated to the best network of each subset. Right : average performance on the test data set associated to this network for each different budget. The filled area correspond to $+/-$ the standard deviation for each curve.

In our evaluation experiments, we account for this randomness by training five models with the same set of seeds for all ablations and by reporting the score of the model with the lowest validation loss for each model.

28

Training mode / Dataset	DAVIS2016	FBMS59	SegTrackV2
LT-MS-K4 (FT3D)	72.4	58.2	60.4
LT-MS-K4 (DAVIS2016)	72.8	57.6	61.3

Table 6: Training on synthetic (FT3D) and on real (DAVIS2016) data. Input sequence length of 120.

7.7 Training on real-world data

We have trained our model LT-MS-K4 on the DAVIS2016 training set and evaluated it on DAVIS2016 validation set, FBMS59 and SegTrackV2. Results are collected in Table 7.7 and compared with those obtained when training our model on the synthetic FT3D dataset. We can observe that performance when training on DAVIS2016 remains globally on par with performance when training on FT3D. Let us note that the volume of data is much smaller when training on DAVIS 2016.

7.8 Influence of the input sequence length at inference

We have also tested how our LT-MS-K4 method behaves when varying the length of the input flow sequence. Results are plotted in Fig.12 for three datasets, DAVIS2016, FBMS59 and SegTrackV2.

As expected, the best performance is obtained for the smallest length (equal to 10) of the input flow sequence. However, performance decreases slowly when the length increases, and remains stable for larger ones. It demonstrates the intrinsic ability of the LT-MS method to achieve accurate and consistent motion segmentation over long periods of the video, which is a unique property.

7.9 Detailed results for DAVIS2016

References

[1] D. Lao, Z. Hu, F. Locatello, Y. Yang, and S. Soatto. Divided attention: Unsupervised multi-object discovery with contextually separated slots. *arXiv:2304.01430*, April 2023.

[2] E. Meunier and P. Bouthemy. Unsupervised space-time network for temporally-consistent segmentation of multiple motions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, June 2023.

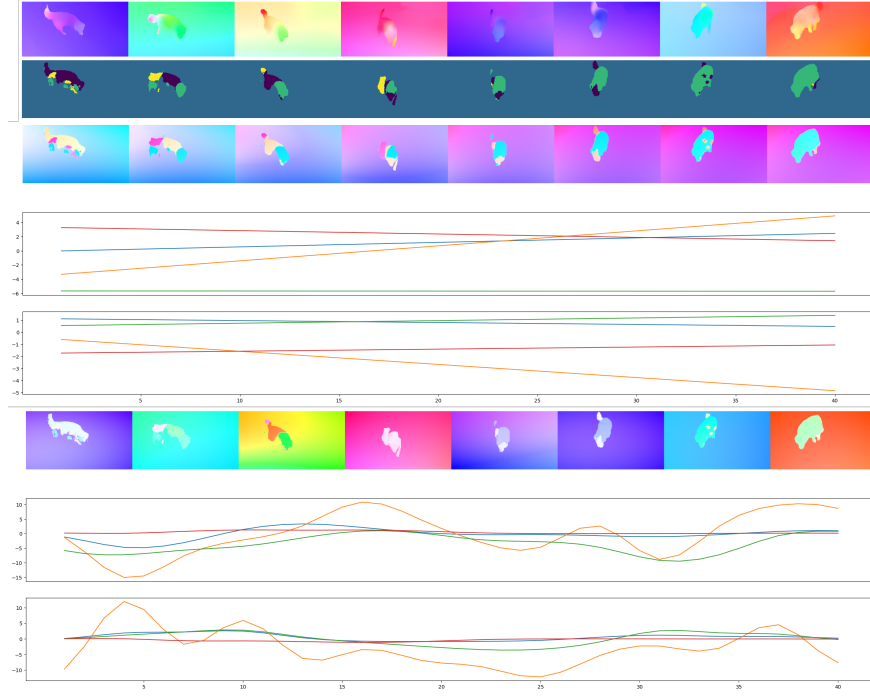## 7.10 Detailed results for DAVIS2017-motion

Figure 11: Visual assessment of the contribution of the spline-based motion model on the *dog* video of DAVIS2016 dataset (with $K = 4$). From top to bottom: input flows of the sequence; corresponding predicted motion segmentation maps; flows reconstructed with the space-time polynomial motion model; plots over time of the mean-value of the u-components of the flows provided per segment by the space-time polynomial motion models; plots over time of the mean-value of the v-components of the flows provided per segment by the space-time polynomial motion models; flows reconstructed with the space-time spline-based motion model; plots over time of the mean-value of the u-components of the flows provided per segment by the spline-based motion models; plots over time of the mean-value of the v-components of the flows provided per segment by the spline-based motion models. Clearly, the space-time polynomial model fails to handle non-uniformly varying motion, whereas the spline-based model does.
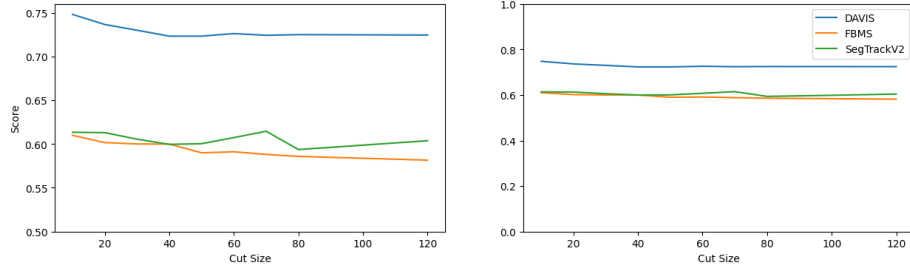
Figure 12: Influence of the length of the input flow sequence (horizontal axis of the plot) on the LT-MS-K4 method for three datasets, DAVIS2016, FBMS59 and SegTrackV2. As expected, the best performance is obtained for the smallest length (equal to 10), but performance decreases slowly when the sequence length increases and remains stable for larger ones. On the right, the plot is shown with the full range of values on the ordinate $[0, 100]$. On the left, a zoomed-in version of the plot.

Video	$\mathcal{J}$ (M)	$\mathcal{J}$ (O)	$\mathcal{J}$ (D)	$\mathcal{F}$ (M)	$\mathcal{F}$ (O)	$\mathcal{F}$ (D)
blackswan	0.468	0.333	-0.058	0.548	0.938	-0.008
bmx-trees	0.596	0.744	0.199	0.787	0.923	0.087
breakdance	0.758	1.000	0.019	0.763	1.000	0.010
camel	0.812	1.000	0.138	0.771	1.000	0.136
car-roundabout	0.926	1.000	0.006	0.816	1.000	-0.033
car-shadow	0.897	1.000	0.014	0.848	1.000	-0.020
cows	0.812	0.990	-0.012	0.716	0.961	-0.012
dance-twirl	0.748	1.000	0.021	0.758	1.000	0.104
dog	0.680	0.845	-0.287	0.575	0.655	-0.207
drift-chicane	0.608	0.680	-0.178	0.699	0.760	-0.055
drift-straight	0.883	1.000	0.065	0.825	0.979	0.290
goat	0.137	0.000	-0.157	0.242	0.000	-0.137
horsejump-high	0.841	1.000	0.060	0.892	1.000	-0.001
kite-surf	0.484	0.458	0.192	0.465	0.250	-0.003
libby	0.745	0.957	0.191	0.849	1.000	0.109
motocross-jump	0.682	0.763	0.142	0.527	0.579	0.176
paragliding-launch	0.615	0.667	0.283	0.275	0.051	0.274
parkour	0.703	0.816	-0.028	0.790	0.980	0.096
scooter-black	0.849	1.000	-0.044	0.716	1.000	0.062
soapbox	0.885	1.000	0.032	0.847	1.000	0.043
Average	0.707	0.813	0.030	0.686	0.804	0.046

Table 7: Results given for every video of DAVIS2016 dataset. Reported scores per video are the average Jaccard score over frames in the video. The very last row is the average over videos scores. $\mathcal{J}$ is the Jaccard index and $\mathcal{F}$ is the Countour Accuracy. The Mean (M) is the average of the scores, the Recall (O) is the fraction of frames per video with a score higher than 0.5, and the Decay (D) is the degradation of the score over time in the video.

Sequence	J-Mean	F-Mean
bike-packing_1	0.078	0.327
bike-packing_2	0.254	0.267
blackswan_1	0.477	0.575
bmx-trees_1	0.608	0.804
breakdance_1	0.450	0.526
camel_1	0.772	0.726
car-roundabout_1	0.913	0.810
car-shadow_1	0.896	0.845
cows_1	0.821	0.728
dance-twirl_1	0.444	0.576
dog_1	0.654	0.571
dogs-jump_1	0.019	0.147
dogs-jump_2	0.254	0.323
dogs-jump_3	0.303	0.351
drift-chicane_1	0.557	0.627
drift-straight_1	0.886	0.829
goat_1	0.220	0.300
gold-fish_1	0.018	0.240
gold-fish_2	0.282	0.336
gold-fish_3	0.357	0.356
gold-fish_4	0.000	0.000
gold-fish_5	0.000	0.000
horsejump-high_1	0.721	0.794
india_1	0.066	0.085
india_2	0.163	0.172
india_3	0.072	0.108
judo_1	0.225	0.397
judo_2	0.286	0.414
kite-surf_1	0.449	0.464
lab-coat_1	0.332	0.350
libby_1	0.746	0.853
loading_1	0.057	0.246
loading_2	0.128	0.259
loading_3	0.427	0.514
mbike-trick_1	0.444	0.436
motocross-jump_1	0.446	0.435
paragliding-launch_1	0.577	0.286
parkour_1	0.531	0.646
pigs_1	0.105	0.433
pigs_2	0.437	0.431
pigs_3	0.059	0.217
scooter-black_1	0.843	0.724
shooting_1	0.434	0.568
soapbox_1	0.487	0.696

J&FMean	JMean	JRecall	JDecay	FMean	FRecall	FDecay
0.422	0.393	0.387	0.004	0.450	0.437	0.024

Table 8: Results given for every video of DAVIS2017-motion dataset. The very last row is the average score over all the videos for the different criteria.