Unsupervised Moving Object Detection via Contextual Information Separation

Yanchao Yang* Antonio Loquercio*
UCLA Vision Lab University of Zurich
Davide Scaramuzza Stefano Soatto
University of Zurich UCLA Vision Lab

Abstract

We propose an adversarial contextual model for detecting moving objects in images. A deep neural network is trained to predict the optical flow in a region using information from everywhere else but that region (context), while another network attempts to make such context as uninformative as possible. The result is a model where hypotheses naturally compete with no need for explicit regularization or hyper-parameter tuning. Although our method requires no supervision whatsoever, it outperforms several methods that are pre-trained on large annotated datasets. Our model can be thought of as a generalization of classical variational generative region-based segmentation, but in a way that avoids explicit regularization or solution of partial differential equations at run-time. We publicly release all our code and trained networks.

1. Introduction

Consider Fig. 1: Even relatively simple objects, when moving in the scene, cause complex discontinuous changes in the image. Being able to rapidly detect independently moving objects in a wide variety of scenes from images is functional to the survival of animals and autonomous vehicles alike. We wish to endow artificial systems with similar capabilities, without the need to pre-condition or learn similar-looking backgrounds. This problem relates to motion segmentation, foreground/background separation, visual attention, video object segmentation as we discuss in Sect. 3. For now, we use the words “object” or “foreground” informally\(^2\) to mean (possibly multiple) connected regions of the image domain, to be distinguished from their surrounding, which we call “background” or “context,” according to some criterion.

Since objects exist in the scene, not in the image, a method to infer them from the latter rests on an operational definition based on measurable image correlates. We call moving objects regions of the image whose motion cannot be explained by that of their surroundings. In other words, the motion of the background is uninformative of the motion of the foreground and vice-versa. The “information separation” can be quantified by the information reduction rate (IRR) between the two as defined in Sect. 2. This naturally translates into an adversarial inference criterion that has close connections with classical variational region-based segmentation, but with a twist: Instead of learning a generative model of a region that explains the image in that region as well as possible, our approach yields a model that tries to explain it as poorly as possible using measurements from everywhere else but that region.

In generative model-based segmentation, one can always explain the image with a trivial model, the image itself. To avoid that, one has to impose model complexity bounds, bottlenecks or regularization. Our model does not have access to trivial solutions, as it is forced to predict a region without looking at it. What we learn instead is a contextual adversarial model, without the need for explicit regularization, where foreground and background hypotheses

\(^1\)http://rpg.ifi.uzh.ch/unsupervised_detection.html
\(^2\)The precise meaning of these terms will be formalized in Sect. 2.
we describe typical failure modes that arise in (or previous) image analysis methods. This is a key strength of our approach, that outperforms the IRR. As one would expect, based on this definition, if the domain of an object is included in another, then they inform each other (see appendix [40]).

2.1. Loss function

We now operationalize the definition of foreground into a criterion to infer it. We use the information reduction rate (IRR) $\gamma$, which takes two subsets $x, y \subset D$ as input and returns a non-negative scalar:

$$
\gamma(x|y; I) = \frac{\mathbb{H}(u_x, u_y|I)}{\mathbb{H}(u_x|I)} = 1 - \frac{\mathbb{H}(u_x|u_y, I)}{\mathbb{H}(u_x|I)}
$$

where $\mathbb{H}$ denotes (Shannon) entropy. It is zero when the two variables are independent, but the normalization prevents the trivial solution (empty set). As proven in the appendix [40], objects as we defined them are the regions that minimize the following loss function

$$
\mathcal{L}(\Omega; I) = \gamma(\Omega|\Omega^c; I) + \gamma(\Omega^c|\Omega; I).
$$

Note that $\mathcal{L}$ does not have a complexity term, or regularizer, as one would expect in most region-based segmentation methods. This is a key strength of our approach, that involves no modeling hyperparameters, as we elaborate on in Sect. 3.

Tame as it may look, (3) is intractable in general. For simplicity we indicate the flow inside the region(s) $\Omega$ (foreground) with $u^{\text{in}} = \{u_i, i \in \Omega\}$, and similarly for $u^{\text{out}}$, the flow in the background $\Omega^c$. The only term that matters in the IRR is the ratio $\mathbb{H}(u^{\text{in}}|u^{\text{out}}, I)/\mathbb{H}(u^{\text{in}}|I)$, which is

$$
\frac{\int \log P(u^{\text{in}}|u^{\text{out}}, I) dP(u^{\text{in}}|u^{\text{out}}, I)}{\int \log P(u^{\text{in}}|I) dP(u^{\text{in}}|I)}
$$

that measures the information transfer from the background to the foreground. This is minimized when knowledge of

---

A small constant $0 < \epsilon \ll 1$ is added to the denominator to avoid singularities, and whenever $x \neq \emptyset$, $\mathbb{H}(u_x|I) \approx \epsilon$, thus we will omit $\epsilon$ from now on.
Figure 2: During training, our method entails two modules. One is the generator (G) which produces a mask of the object by looking at the image and the associated optical flow. The other module is the inpainter (I) which tries to inpaint back the optical flow masked out by the corresponding mask. Both modules employ the encoder-decoder structure with skip connections. However, the inpainter (I) is equipped with two separate encoding branches. See Sect. 4.1 for network details.

![Diagram of Figure 2](image)

Figure 3: The two diagrams illustrate the learning process of the mask generator (G), after the inpainter (I) has learned how to accurately inpaint a masked flow. The upper diagram shows a poorly trained mask generator which does not precisely detect the object. Due to the imprecise detection, the inpainter can observe part of the object’s flow, and perform an accurate reconstruction. At the same time, the inpainter partially observes the background’s flow in the complementary mask. Consequently, it can precisely predict missing parts of the background’s flow. In contrast, the lower diagram shows a fully trained mask generator which can precisely tell apart the object from the background. In this case, the inpainter observes the flow only outside the object and has no information to predict the flow inside it. At initialization time the inpainter does not know the conditionals to inpaint masked flows. Therefore, we propose to train both the generator and the inpainter jointly in an adversarial manner (see Sect. 2).

![Diagram of Figure 3](image)

the background flow is sufficient to predict the foreground. To enable computation, we have to make draconian, yet common, assumptions on the underlying probability model, namely that

\[
P(u_i^\text{in} = x | I) \propto \exp \left(-\frac{||x||^2}{\sigma^2}\right) \tag{5}
\]

\[
P(u_i^\text{in} = x | u_i^\text{out} = y, I) \propto \exp \left(-\frac{||x - \phi(\Omega, y, I)||^2}{\sigma^2}\right)
\]

where \(\phi(\Omega, y, I) = \int u_i^\text{in} dP(u_i^\text{in} | u_i^\text{out}, I)\) is the conditional mean given the image and the complementary observation. Here we assume \(\phi(\Omega, y, I) = 0\), since given a single image the most probable guess of the flow is zeros. With these assumptions, (4) can be simplified, to

\[
\frac{\int ||u_i^\text{in} - \phi(\Omega, u_i^\text{out}, I)||^2 dP(u_i^\text{in} | u_i^\text{out}, I)}{\int ||u_i^\text{in}||^2 dP(u_i^\text{in} | I)} \approx \frac{\sum_{i=1}^{N} ||u_i^\text{in} - \phi(\Omega, u_i^\text{out}, I)||^2}{\sum_{i=1}^{N} ||u_i^\text{in}||^2} \tag{6}
\]
where $N = |D|$ is the cardinality of $D$, or the number of flow samples available. Finally, our loss (3) to be minimized can be approximated as

$$
\mathcal{L}(\Omega; I) = 1 - \frac{\sum_{i=1}^{N} \|u_i^\text{in} - \phi(\Omega, u_i^\text{out}, I)\|^2}{\sum_{i=1}^{N} \|u_i^\text{in}\|^2 + \epsilon} + 1 - \frac{\sum_{i=1}^{N} \|u_i^\text{out} - \phi(\Omega^c, u_i^\text{in}, I)\|^2}{\sum_{i=1}^{N} \|u_i^\text{out}\|^2 + \epsilon}.
$$

(7)

In order to minimize this loss, we have to choose a representation for the unknown region $\Omega$ and the function $\phi$.

### 2.2. Function class

The region $\Omega$ that minimizes (7) belongs to the power set of $D$, that is the set of all possible subsets of the image domain, which has exponential complexity.\(^4\) We represent it with the indicator function

$$
\chi : D \rightarrow \{0, 1\}
$$

\(i \mapsto 1\) if \(i \in \Omega\); 0 otherwise

so that the flow inside the region $\Omega$ can be written as $u_i^\text{in} = \chi u_i$, and outside as $u_i^\text{out} = (1 - \chi) u_i$.

Similarly, the function $\phi$ is non-linear, non-local, and high-dimensional, as it has to predict the flow in a region of the image of varying size and shape, given the flow in a different region. In other words, $\phi$ has to capture the context of a region to recover its flow.

Characteristically for the ages, we choose both $\phi$ and $\chi$ to be in the parametric function class of deep convolutional neural networks, as shown in Fig. 2, the specifics of which are in Sect. 4.1. We indicate the parameters with $w$, and the corresponding functions $\phi_w$, and $\chi_w$. Accordingly, after discarding the constants, the negative loss (7) can be written as a function of the parameters

$$
\mathcal{L}(w_1, w_2; I) = \frac{\sum_{i} \|\chi_w(u_i - \phi_w(\chi_w, u_i^\text{out}, I))\|^2}{\sum_{i} \|u_i^\text{in}\|^2} + \frac{\sum_{i} \|(1 - \chi_w)(u_i - \phi_w(1 - \chi_w, u_i^\text{in}, I))\|^2}{\sum_{i} \|u_i^\text{out}\|^2}
$$

(9)

$\phi_w$ is called the inpainter network, and must be chosen to minimize the loss above. At the same time, the region $\Omega$, represented by the parameters $w_2$ of its indicator function $\chi_w$ called mask generator network, should be chosen so that $u^\text{out}$ is as uninformative as possible of $u^\text{in}$, and therefore the same loss is maximized with respect to $w_2$. This naturally gives rise to a minmax problem:

$$
\hat{w} = \arg \min_{w_1} \max_{w_2} \mathcal{L}(w_1, w_2; I).
$$

(10)

This loss has interesting connections to classical region-based segmentation, but with a twist as we discuss next.

### 3. Related Work

To understand the relation of our approach to classical methods, consider the simplest model for region-based segmentation\(^8\)

$$
L(\Omega, c_i, c_o) = \int_{\Omega} |u^\text{in}(x) - c_i|^2 dx + \int_{\Omega^c} |u^\text{out}(x) - c_o|^2 dx
$$

(11)

typically combined with a regularizing term, for instance the length of the boundary of $\Omega$. This is a convex infinite-dimensional optimization problem that can be solved by numerically integrating a partial differential equation (PDE). The result enjoys significant robustness to noise, provided the underlying scene has piecewise constant radiance and is measured by image irradiance, to which it is related by a simple “signal-plus-noise” model. Not many scenes of interest have piecewise constant radiance, although this method has enjoyed a long career in medical image analysis. If we enrich the model by replacing the constants $c_i$ with smooth functions, $\phi_i(x)$, we obtain the celebrated Mumford-Shah functional\(^25\), also optimized by integrating a PDE. Since smooth functions are an infinite-dimensional space, regularization is needed, which opens the Pandora box of regularization criteria, not to mention hyperparameters: Too much regularization and details are missed; too little and the model gets stuck in noise-induced minima. A modern version of this program would replace $\phi(x)$ with a parametrized model $\phi_w(x)$, for instance a deep neural network with weights $w$ pre-trained on a dataset $D$.

In this case, the loss is a function of $w$, with natural model complexity bounds. Evaluating $\phi_w$ at a point inside, $x \in \Omega$, requires knowledge of the entire function $u$ inside $\Omega$, which we indicate with $\phi_w(x, u^\text{in})$:

$$
\int_{\Omega} |u^\text{in}(x) - \phi_w(x, u^\text{in})|^2 dx + \int_{\Omega^c} |u^\text{out}(x) - \phi_w(x, u^\text{out})|^2 dx.
$$

(12)

Here, a network can just map $\phi_w(x, u^\text{in}) = u^\text{in}$ providing a trivial solution, avoided by introducing (architectural or information) bottlenecks, akin to explicit regularizers. We turn the table around and use the outside to predict the inside and vice-versa:

$$
\int_{\Omega} |u^\text{in}(x) - \phi_w(x, u^\text{out})|^2 dx + \int_{\Omega^c} |u^\text{out}(x) - \phi_w(x, u^\text{in})|^2 dx.
$$

(13)

After normalization and discretization, this leads to our loss function (7). The two regions compete: for one to grow, the other has to shrink. In this sense, our approach relates to region competition methods, and specifically Motion Competition\(^12\), but also to adversarial training, since we can

\(^{4}\)In the continuum, it belongs to the infinite-dimensional set of compact and multiply-connected regions of the unit square.
think of $\phi$ as the “discriminator” presented in a classification problem (GAN [1]), reflected in the loss function we use. This also relates to what is called “self-supervised learning,” a misnomer since there is no supervision, just a loss function that does not involve externally annotated data. Several variants of our approach can be constructed by using different norms, or correspondingly different models for the joint and marginal distributions (5).

More broadly, the ability to detect independently moving objects is primal, so there is a long history of motion-based segmentation, or moving object detection. Early attempts to explicitly model occlusions include the layer model [38] with piecewise affine regions, with computational complexity improvements using graph-based methods [30] and variational inference [11, 6, 32, 43] to jointly optimize for motion estimation and segmentation; [26] use of long-term temporal consistency and color constancy, making however the optimization more difficult and sensitive to parameter choices. Similar ideas were applied to motion detection in crowds [5], traffic monitoring [4] and medical image analysis [14]. Our work also related to the literature on visual attention [16, 7].

More recent data-driven methods [36, 35, 9, 31] learn discriminative spatio-temporal features and differ mainly for the type of inputs and architectures. Inputs can be either image pairs [31, 9] or image plus dense optical flow [36, 35]. Architectures can be either time-independent [35], or with recurrent memory [36, 31]. Overall, those methods outperform traditional ones on benchmark datasets [26, 29], but at the cost of requiring a large amount of labeled training data and with evidence of poor generalization to previously unseen data.

It must be noted that, unlike in Machine Learning at large, it is customary in video object segmentation to call “unsupervised” methods that do rely on massive amounts of manually annotated data, so long as they do not require manual annotation at run-time. We adopt the broader use of the term where unsupervised means that there is no supervision of any kind both at training and test time.

Like classical variational methods, our approach does not need any annotated training data. However, like modern learning methods, our approach learns a contextual model, which would be impossible to engineer given the complexity of image formation and scene dynamics.

4. Experiments

We compare our approach to a set of state-of-the-art baselines on the task of video object segmentation to evaluate the accuracy of detection. We first present experiments on a controlled toy-example, where the assumptions of our model are perfectly satisfied. The aim of this experiment is to get a sense of the capabilities of the presented approach in ideal conditions. In the second set of experiments, we evaluate the effectiveness of the proposed model on three public, widely used datasets: Densely Annotated Video Segmentation (DAVIS) [29], Freiburg-Berkeley Motion Segmentation (FBMS59) [26], and SegTrackV2 [37]. Provided the high degree of appearance and resolution differences between them, these datasets represent a challenging benchmark for any moving object segmentation method. While the DAVIS dataset has always a single object per scene, FBMS and SegTrackV2 scenes can contain multiple objects per frame. We show that our method not only outperforms the unsupervised approaches, but even edges out other supervised algorithms that, in contrast to ours, have access to a large amount of labeled data with precise manual segmentation at training time. For quantitative evaluation, we employ the most common metric for video object segmentation, i.e. the mean Jaccard score, a.k.a. intersection-over-union score, $J$. Given space constraints, we add additional evaluation metrics in the appendix [40].

4.1. Implementation and Networks Details

**Generator, $G$:** Depicted on the left of Fig. 3, the generator architecture is a shrunken version of SegNet [2]. Its encoder part consists of 5 convolutional layers each followed by batch normalization, reducing the input image to $\frac{1}{4}$ of its original dimensions. The encoder is followed by a set of 4 atrous convolutions with increasing radius (2,4,8,16). The decoder part consists of 5 convolutional layers, that, with upsampling, generate an output with the same size of the input image. As in SegNet [2], a final softmax layer generates the probabilities for each pixel to be foreground or background. The generator input consists of an RGB image $I_t$ and the optical flow $\psi_{t:t+\delta T}$ between $I_t$ and $I_{t+\delta T}$, to introduce more variations in the optical flows conditioned on image $I_t$. At training time, $\delta T$ is randomly sampled from the uniform distribution $\delta T \sim [-5, 5]$, with $\delta T \neq 0$. The optical flow $\psi_{t:t+\delta T}$ is generated with the pretrained PWC network [33], given its state-of-the-art accuracy and efficiency. The generator network has a total of 3.4M parameters.

**Inpainter, $I$:** We adapt the architecture of CPN [41] to build our inpainter network. Its structure is depicted on the right of Fig. 3. The input to this network consists of the input image $I_t$ and the flow masked according to the generator output, $\chi u$, the latter concatenated with $\chi$, to make the inpainter aware of the region to look for context. Differently from the CPN, these two branches are balanced, and have the same number of parameters. The encoded features are then concatenated and passed to the CPN decoder, that outputs an optical flow $\hat{u} = \phi(\chi, (1 - \chi) u, I_t)$ of the same size of the input image, whose inside is going to be used for the difference between $u^{\text{inv}}$ and the recovered flow inside. Similarly, we can run the same procedure for the complement part. Our inpainter network has a total of 1.5M parameters.

At test time, only the generator $G$ is used. Given $I_t$,
shows the performance of our method in the Appendix. As it is possible to observe in Table 1, the datasets and the visual results can be found in the Appendix. More details about the generation of those datasets and the visual results can be found in the appendix.

4.2. Experiments in Ideal Conditions

Our method relies on basic, fundamental assumptions: The optical flow of the foreground and of the background are independent. To get a sense of the capabilities of our approach in ideal conditions, we artificially produce datasets where this assumption is fully satisfied. The datasets are generated as a modification of DAVIS2016 [29], FMBS [26], and SegTrackV2 [37]. While images are kept unchanged, ground truth masks are used to artificially perturb the optical flow generated by PWC [33] such that foreground and background are statistically independent. More specifically, a different (constant) optical flow field is sampled from a uniform distribution independently at each frame, and associated to the foreground and the background, respectively. More details about the generation of those datasets and the visual results can be found in the Appendix. As it is possible to observe in Table 1, our method reaches very high performance in all considered datasets. This validates the correctness of our algorithm and that our loss function (10) is a valid and tractable approximation of the functional (3).

4.3. Performance on Video Object Segmentation

As previously stated, we use the term Unsupervised with a different meaning with respect to its definition in literature of video object segmentation. In our definition and for what follows, the supervision refers to the algorithm’s usage of ground truth object annotations at training time. In contrast, the literature usually defines methods as semi-supervised, if at test time they assume the ground-truth segmentation of the first frame to be known [5, 24]. This could be posed as tracking problem [42] since the detection of the target is human generated. Instead, here we focus on moving object detection and thus we compare our approach to the methods that are usually referred to as “unsupervised” in the video object segmentation domain. However, we make further differentiation on whether the ground truth object segmentation is needed (supervised) or not (truly unsupervised) during training.

In this section we compare our method with other 8 methods that represent the state of the art for moving object segmentation. For comparison, we use the same metric defined above, which is the Jaccard score $J$ between the real and predicted masks.

Table 2 shows the performance of our method and the baseline methods on three popular datasets, DAVIS2016 [29], FMBS59 [26] and SegTrackV2 [37]. Our approach is top-two in each of the considered datasets, and even outperforms baselines that need a large amount of labelled data at training time, i.e. FSEG [17].

As can be observed in Table 2, unsupervised baselines typically perform well in one dataset but significantly worse in others. For example, despite being the best performing unsupervised method on DAVIS2016, the performance of ARP [20] drops significantly in the FMBS59 [26] and SegTrackV2 [26] datasets. ARP outperforms our method by 6.5% on DAVIS, however, our method outperforms ARP by 6.3% and 8.4%, on FMBS59 and SegTrackV2 respectively. Similarly, NLC [15] and SAGE [39] are extremely competitive in the Segtrack and FMBS59 benchmarks, respectively, but not in others. NLC outperforms us on SegTrackV2 by 8.4%, however we outperform NLC by 29.8% and 24.7%, on DAVIS and FMBS respectively.

It has been established that being second-best in multiple benchmarks is more indicative of robust performance than being best in one [27]. Indeed, existing unsupervised approaches for moving object segmentation are typically highly-engineered pipeline methods which are tuned on one dataset but do not necessarily generalize to others. Also, consisting of several computationally intensive steps, existing unsupervised methods are generally orders of magnitude slower than our method (Table 3).

Interestingly, a similar pattern is observable for supervised methods. This is particularly evident on the SegTrackV2 dataset [37], which is particularly challenging since several frames have very low resolution and are motion blurred. Indeed, supervised methods have difficulties with the covariate shift due to changes in the distribution between training and testing data. Generally, supervised methods alleviate this problem by pre-training on image segmentation datasets, but this solution clearly does not scale to every possible case. In contrast, our method can be finetuned on any data without the need for the latter to be annotated. As a result, our approach outperforms the majority of unsupervised methods as well as all the supervised ones, in

<table>
<thead>
<tr>
<th>Method</th>
<th>DAVIS [29]</th>
<th>FMBS59 [26]</th>
<th>SegTrackV2 [37]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J$ (%)</td>
<td>92.5</td>
<td>88.5</td>
<td>92.1</td>
</tr>
</tbody>
</table>

Table 1: Performance under ideal conditions: When the assumptions made by our model are fully satisfied, our approach can successfully detect moving objects. Indeed, our model reaches near maximum Jaccard score in all considered datasets.
terms of segmentation quality and training efficiency.

4.4. Qualitative experiments and Failure Cases

In Fig. 4 we show a qualitative comparison of the detection generated by our and others’ methods on the DAVIS dataset. Our algorithm can segment precisely the moving object regardless of cluttered background, occlusions, or large depth discontinuities. The typical failure case of our method is the detection of objects whose motion is due to the primary object. An example is given in the last row of Fig. 4, where the water moved by the surfer is also classified as foreground by our algorithm.

4.5. Training and Runtime Analysis

The generator and inpainter network’s parameters are trained at the same time by minimizing the functional (10). The optimization time is approximately 6 hours on a single GPU Nvidia Titan XP 1080i. Since both our generator and inpainter networks are relatively small, we can afford very fast training/finetuning times. This stands in contrast to larger modules, e.g. PDB [31], that require up to 40 hrs of training.

At test time, predictions $\mathcal{T}_g$ (defined in eq. 14) are generated at 3.15 FPS, or with an average time of 320ms per frame, including the time to compute optical flow with PWC [33]. Excluding the time to generate optical flow, our model can generate predictions at 10.2 FPS, or 98ms per frame. All previous timings do not include the CRF post-processing step. Table 3 compares the inference time of our method with respect to other unsupervised methods. Since our method at test time requires only a pass through a relatively shallow network, it is orders of magnitude faster than other unsupervised approaches.

5. Discussion

Our definition of objects and the resulting inference criterion are related to generative model-based segmentation and region-based methods popular in the nineties. However, there is an important difference: Instead of using the evidence inside a region to infer a model of that region which is as accurate as possible, we use evidence everywhere else but that region to infer a model within the region, and we seek the model to be as bad as possible. This relation, explored in detail in Sect. 3, forces learning a contextual model of the image, which is not otherwise the outcome of a generative model in region-based segmentation. For instance, if we choose a rich enough model class, we can trivially model the appearance of an object inside an image region as the image itself. This is not an option in our model: We can only predict the inside of a region by looking outside of it. This frees us from having to impose modeling assumptions to avoid trivial solutions, but requires a much richer class of function to harvest contextual information.

This naturally gives rise to an adversarial (min-max) optimization: An inpainter network, as a discriminator, tries to hallucinate the flow inside from the outside, with the reconstruction error as a quality measure of the generator network, which tries to force the inpainter network to do the lousiest possible job.

The strengths of our approach relate to its ability to learn complex relations between foreground and background without any annotation. This is made possible by using modern deep neural network architectures like Seg-Net [2] and CPN [41] as function approximators.

Not using ground-truth annotations can be seen as a strength but also a limitation: If massive datasets are available, why not use them? In part because even massive is not large enough: We have shown that models trained on large amount of data still suffer performance drops whenever tested on a new benchmark significantly different from the training ones. Moreover, our method does not require any pre-training on large image segmentation datasets, and it can adapt to any new data, since it does not require any supervision. This adaptation ability is not only important for computer vision tasks, but can also benefit other applications, e.g. robotic navigation [23, 13] or manipulation [19].

Another limitation of our approach is that, for the task of

## Table 2: Moving Object Segmentation Benchmarks

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DAVIS2016 [29]</td>
<td>77.2</td>
<td>70.7</td>
<td>75.9</td>
<td>76.2</td>
<td>55.8</td>
<td>55.1</td>
<td>42.6</td>
<td>55.2</td>
<td>71.5</td>
</tr>
<tr>
<td>FBMS59 [26]</td>
<td>74.0</td>
<td>68.4</td>
<td>65.1</td>
<td>59.8</td>
<td>47.7</td>
<td>51.5</td>
<td>61.2</td>
<td>57.2</td>
<td>63.6</td>
</tr>
<tr>
<td>SegTrackV2 [37]</td>
<td>60.9</td>
<td>61.4</td>
<td>57.3</td>
<td>57.2</td>
<td>47.8</td>
<td>67.2</td>
<td>57.6</td>
<td>54.3</td>
<td>62.0</td>
</tr>
<tr>
<td>DNN-Based</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Pre-Training Required</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

We compare our approach with 8 different baselines on the task of moving object segmentation. In order to do so, we use three popular datasets, i.e. DAVIS2016 [29], FBMS59 [26], and SegTrackV2 [37]. Methods in blue require ground truth annotations at training time and are pre-trained on image segmentation datasets. In contrast, methods in red are unsupervised and not require any ground-truth annotation. Our approach is top-two in all the considered benchmarks, comparing to the other unsupervised methods. **Bold** indicates best among all methods, while **Bold Red** and red represent the best and second best for unsupervised methods, respectively.
Table 3: **Run-time analysis**: Our method is not only effective (top-two in each considered dataset), but also orders of magnitude faster than other unsupervised methods. All timings are indicated without optical flow computation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime(s)</td>
<td>74.5</td>
<td>0.5</td>
<td>11.0</td>
<td>0.88</td>
<td>103.0</td>
<td><strong>0.098</strong></td>
</tr>
<tr>
<td>DNN-based</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Figure 4: **Qualitative Results**: We qualitatively compare the performance of our approach with several state-of-the-art baselines as well as the Ground-Truth (GT) mask. Our prediction are robust to background clutter, large depth discontinuities and occlusions. The last row shows a typical failure case of our method, *i.e.* objects which are moved by the primary objects are detected as foreground (water is moved by the surfer in this case).

motion-based segmentation, we require the optical flow between subsequent frames. One could argue that optical flow is costly, local, and error-prone. However, our method is general and could be applied to other statistics than optical flow. Such extensions are part of our future work agenda. In addition, our approach does not fully exploit the intensity image, although we use it as a conditioning factor for the inpainter network. An optical flow or an image can be ambiguous in some cases, but the combination of the two is rarely insufficient for recognition [43]. Again, our framework allows in theory exploitation of both, and in future work we intend to expand in this direction.

6. **Appendix**

**Notation**

- $\Omega \subset D$: $\Omega$ is a subset of the image domain $D$.
- $i \in \Omega$: pixel $i$ in region $\Omega$.
- $u_{\Omega}$: optical flow $u$ restricted to region $\Omega$.
- $\mathbb{I}(u_i, u_j | I)$: conditional mutual information between $u_i$ and $u_j$ given image $I$.

6.1. More on the definition of objects

Our definition Eq. (1) of object is in terms of mutual information between optical flows restricted to different regions. It is helpful to analyze some simple cases that follow directly from the definition.
statement 1: a subset of an object informs the remaining part of this object. If the object is $\Omega$, and there is a subset $\hat{\Omega} \subseteq \Omega$, suppose $i \in \hat{\Omega}, j \in \Omega \setminus \hat{\Omega}$ respectively, then: $I(u_{\hat{\Omega}}, u_{\Omega \setminus \hat{\Omega}}) \geq I(u_i, u_j) > 0$ by Eq. (1).

statement 2: a subset of the foreground does not inform a subset of the background. Suppose $\Omega$ is the foreground, if $\hat{\Omega} \subseteq \Omega$, and $\hat{\Omega}' \subseteq \Omega'$ then $I(u_{\hat{\Omega}}, u_{\hat{\Omega}'}) = 0$. Otherwise, we can find at least two pixels $i \in \hat{\Omega}, j \in \Omega'$ such that $I(u_i, u_j) > 0$, which is contradictory to definition Eq. (1).

6.2. The optimality and uniqueness of objects

In the main paper, we formalize the notion of foreground as a region $\Omega$ that is uninformed by the background, see Eq. (1). Objects as we defined them in Eq. (1) are the regions that minimize the loss function Eq. (3).

Proof: First we show that the estimate $\hat{\Omega}$ right on the object achieves the minimum value of the loss function, since:

$$
\mathcal{L}(\hat{\Omega}; I) = \gamma(\hat{\Omega}; D(\hat{\Omega}; I) + \gamma(D(\hat{\Omega}; | \hat{\Omega}; I) = H(u_{\hat{\Omega}}) + H(u_{\hat{\Omega}'}) = 0 \quad (15)
$$

by statement (2) above. Thus $\hat{\Omega}$ achieves the minimum value of the loss Eq. (3). Now we need to show that $\hat{\Omega}$ is unique, for which, we just need to check the following two mutually exclusive and collectively inclusive cases for $\hat{\Omega} \neq \hat{\Omega}$ (note that $\mathcal{L}(\hat{\Omega}; I) = \mathcal{L}(D; I) = 1.0$ as $0 < \epsilon \ll 1$ is added to the denominator):

- $\hat{\Omega}$ is either a subset of foreground or a subset of background: $\hat{\Omega} \cap D \backslash \hat{\Omega} = \emptyset$ or $\hat{\Omega} \cap \hat{\Omega} = \emptyset$.

- $\hat{\Omega}$ is neither a subset of foreground nor a subset of background: $\hat{\Omega} \cap D \backslash \hat{\Omega} \neq \emptyset$ and $\hat{\Omega} \cap \hat{\Omega} \neq \emptyset$.

In both cases $\mathcal{L}(\hat{\Omega}; I)$ is strictly larger than 0 with some set operations under statements (1,2) above. Thus the object satisfies the definition Eq. (1) is a unique optima of the loss Eq. (3).

A3

The performance of our algorithm compared with state-of-the-art systems on DAVIS2016 [29] can be found in Table 4. The metrics used to perform the quantitative evaluation are the Jaccard score $J$, and the mean boundary measure $J$. For more details see [29]. According to the same metrics, we also provide the per-category score for DAVIS and FBMS59 in Table 5 and 6.

A4

We use the online implementation\(^5\) of the CRF algorithm [21] for post-processing the mask $\hat{F}$. We only use the pairwise bilateral potential with the parameters: $s_{xy}=25$, $s_{rgb}=5$, $com=5$ as defined in the corresponding function.

A5

Experiments in ideal conditions The experiments under ideal conditions have been specifically designed to test the performance of our algorithm when its assumptions are fully satisfied. In particular, the derivation presented in Section 2 assumes that the foreground and the background motion are completely independent given the image. In real datasets, this conditions is not always true: during the generation of the benchmark datasets, many times the camera tracks the moving object, creating a relation between the two motions. To analyze the performance of our method when background and foreground are completely independent, we artificially modify our considered benchmark datasets. To generate an ideal sample, we proceed as follows: We first take two image, $I_t$ and $I_{t+\delta T}$, with $\delta T \in U[-5,5]$. Then, we generate the optical flow between the two frame $u_{t:t+\delta T}$ using PWC Net [33]. Now, using the ground-truth mask, we artificially add to the latter a random optical flow, different between the foreground and background. The random optical flows are generated from a rotation $r \in U[-1,1]$ radians and translations $t_x, t_y \in U[-30,30]$ pixels. Since the ground-truth mask is used to generate the flow, in several case a solution is easy to find (last column of Fig. 5 and Fig. 6). However, as the original optical flow can be complex, it is not always possible to easily observe the mask in the flow (first three columns of Fig. 5 and Fig. 6). Nonetheless, since the background and foreground are fully independent, our algorithm can accurately segment the objects.

\(^5\)https://github.com/lucasb-eyer/pydensecrf
Table 4: **Performance on DAVIS2016** [29]: Methods in blue require ground truth annotations at training time, while the ones in red are fully unsupervised. Our approach reaches comparable performance to the state-of-the-art, outperforming several supervised methods.

<table>
<thead>
<tr>
<th>Category</th>
<th>( J ) mean ↑</th>
<th>( J ) recall ↑</th>
<th>( J ) decay ↓</th>
<th>( F ) mean ↑</th>
<th>( F ) recall ↑</th>
<th>( F ) decay ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>blackswan</td>
<td>69.1</td>
<td>100.0</td>
<td>25.2</td>
<td>72.4</td>
<td>100.0</td>
<td>23.6</td>
</tr>
<tr>
<td>bmx-trees</td>
<td>59.2</td>
<td>74.4</td>
<td>12.6</td>
<td>84.1</td>
<td>100.0</td>
<td>-7.4</td>
</tr>
<tr>
<td>breakdance</td>
<td>82.4</td>
<td>100.0</td>
<td>-0.6</td>
<td>84.1</td>
<td>100.0</td>
<td>-3.3</td>
</tr>
<tr>
<td>camel</td>
<td>83.0</td>
<td>100.0</td>
<td>4.6</td>
<td>83.0</td>
<td>100.0</td>
<td>0.7</td>
</tr>
<tr>
<td>car-roundabout</td>
<td>87.6</td>
<td>100.0</td>
<td>-0.8</td>
<td>76.5</td>
<td>98.6</td>
<td>-5.5</td>
</tr>
<tr>
<td>car-shadow</td>
<td>78.6</td>
<td>100.0</td>
<td>15.4</td>
<td>74.3</td>
<td>100.0</td>
<td>6.2</td>
</tr>
<tr>
<td>cows</td>
<td>85.4</td>
<td>100.0</td>
<td>0.6</td>
<td>79.6</td>
<td>98.0</td>
<td>-1.3</td>
</tr>
<tr>
<td>dance-twirl</td>
<td>79.0</td>
<td>95.5</td>
<td>1.2</td>
<td>82.4</td>
<td>100.0</td>
<td>9.9</td>
</tr>
<tr>
<td>dog</td>
<td>80.0</td>
<td>100.0</td>
<td>10.2</td>
<td>76.1</td>
<td>94.8</td>
<td>17.8</td>
</tr>
<tr>
<td>drift-chicane</td>
<td>62.0</td>
<td>80.0</td>
<td>10.9</td>
<td>76.1</td>
<td>90.0</td>
<td>18.4</td>
</tr>
<tr>
<td>drift-straight</td>
<td>67.9</td>
<td>87.5</td>
<td>16.9</td>
<td>57.7</td>
<td>54.2</td>
<td>43.0</td>
</tr>
<tr>
<td>goat</td>
<td>26.9</td>
<td>17.0</td>
<td>-17.7</td>
<td>34.4</td>
<td>17.0</td>
<td>-14.7</td>
</tr>
<tr>
<td>horsejump-high</td>
<td>79.6</td>
<td>100.0</td>
<td>15.6</td>
<td>87.6</td>
<td>100.0</td>
<td>9.4</td>
</tr>
<tr>
<td>kite-surf</td>
<td>23.9</td>
<td>0.0</td>
<td>6.7</td>
<td>44.0</td>
<td>20.8</td>
<td>-5.2</td>
</tr>
<tr>
<td>libby</td>
<td>76.5</td>
<td>91.5</td>
<td>19.1</td>
<td>92.2</td>
<td>100.0</td>
<td>2.7</td>
</tr>
<tr>
<td>motocross-jump</td>
<td>70.6</td>
<td>78.9</td>
<td>-1.6</td>
<td>53.4</td>
<td>57.9</td>
<td>-7.0</td>
</tr>
<tr>
<td>paragliding-launch</td>
<td>72.7</td>
<td>100.0</td>
<td>25.0</td>
<td>44.3</td>
<td>43.6</td>
<td>33.3</td>
</tr>
<tr>
<td>parkour</td>
<td>83.4</td>
<td>98.0</td>
<td>5.4</td>
<td>88.9</td>
<td>100.0</td>
<td>7.9</td>
</tr>
<tr>
<td>scooter-black</td>
<td>76.5</td>
<td>95.1</td>
<td>-3.5</td>
<td>70.2</td>
<td>95.1</td>
<td>-2.0</td>
</tr>
<tr>
<td>soapbox</td>
<td>77.0</td>
<td>95.9</td>
<td>16.7</td>
<td>74.5</td>
<td>100.0</td>
<td>18.0</td>
</tr>
</tbody>
</table>

Table 5: **Performance on DAVIS2016** [29]: Per category performance on the DAVIS2016 dataset.

<table>
<thead>
<tr>
<th>Category</th>
<th>( J ) mean ↑</th>
<th>( F ) mean ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>blackswan</td>
<td>71.5</td>
<td>72.4</td>
</tr>
<tr>
<td>bmx-trees</td>
<td>86.5</td>
<td>100.0</td>
</tr>
<tr>
<td>breakdance</td>
<td>9.5</td>
<td>100.0</td>
</tr>
<tr>
<td>car-roundabout</td>
<td>70.5</td>
<td>74.3</td>
</tr>
<tr>
<td>car-shadow</td>
<td>83.5</td>
<td>100.0</td>
</tr>
<tr>
<td>cows</td>
<td>5.1</td>
<td>98.6</td>
</tr>
<tr>
<td>dance-twirl</td>
<td>2.9</td>
<td>98.0</td>
</tr>
<tr>
<td>dog</td>
<td>11.4</td>
<td>94.8</td>
</tr>
<tr>
<td>drift-chicane</td>
<td>7.0</td>
<td>90.0</td>
</tr>
<tr>
<td>drift-straight</td>
<td></td>
<td>18.4</td>
</tr>
<tr>
<td>goat</td>
<td></td>
<td>14.7</td>
</tr>
<tr>
<td>horsejump-high</td>
<td></td>
<td>9.4</td>
</tr>
<tr>
<td>kite-surf</td>
<td></td>
<td>5.2</td>
</tr>
<tr>
<td>libby</td>
<td></td>
<td>2.7</td>
</tr>
<tr>
<td>motocross-jump</td>
<td></td>
<td>7.0</td>
</tr>
<tr>
<td>paragliding-launch</td>
<td></td>
<td>33.3</td>
</tr>
<tr>
<td>parkour</td>
<td></td>
<td>7.9</td>
</tr>
<tr>
<td>scooter-black</td>
<td></td>
<td>2.0</td>
</tr>
<tr>
<td>soapbox</td>
<td></td>
<td>18.0</td>
</tr>
</tbody>
</table>

References


<table>
<thead>
<tr>
<th>Category</th>
<th>$\mathcal{J}$ mean</th>
<th>$\mathcal{J}$ recall</th>
<th>$\mathcal{J}$ decay</th>
<th>$\mathcal{F}$ mean</th>
<th>$\mathcal{F}$ recall</th>
<th>$\mathcal{F}$ decay</th>
</tr>
</thead>
<tbody>
<tr>
<td>camel01</td>
<td>78.3</td>
<td>100.0</td>
<td>3.3</td>
<td>83.1</td>
<td>100.0</td>
<td>4.9</td>
</tr>
<tr>
<td>cars1</td>
<td>61.4</td>
<td>100.0</td>
<td>0.0</td>
<td>37.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>cars10</td>
<td>31.1</td>
<td>0.0</td>
<td>-9.4</td>
<td>22.9</td>
<td>0.0</td>
<td>0.8</td>
</tr>
<tr>
<td>cars4</td>
<td>83.6</td>
<td>100.0</td>
<td>7.6</td>
<td>79.5</td>
<td>100.0</td>
<td>11.6</td>
</tr>
<tr>
<td>cats01</td>
<td>79.0</td>
<td>100.0</td>
<td>9.2</td>
<td>78.5</td>
<td>100.0</td>
<td>14.1</td>
</tr>
<tr>
<td>cats03</td>
<td>75.8</td>
<td>100.0</td>
<td>-12.6</td>
<td>73.8</td>
<td>100.0</td>
<td>0.8</td>
</tr>
<tr>
<td>cats06</td>
<td>61.3</td>
<td>81.2</td>
<td>27.3</td>
<td>75.6</td>
<td>87.5</td>
<td>17.5</td>
</tr>
<tr>
<td>dogs01</td>
<td>73.4</td>
<td>77.8</td>
<td>35.9</td>
<td>74.3</td>
<td>88.9</td>
<td>21.3</td>
</tr>
<tr>
<td>dogs02</td>
<td>73.8</td>
<td>90.0</td>
<td>-9.7</td>
<td>74.4</td>
<td>90.0</td>
<td>10.9</td>
</tr>
<tr>
<td>farm01</td>
<td>82.4</td>
<td>91.7</td>
<td>-24.6</td>
<td>71.0</td>
<td>61.5</td>
<td>27.8</td>
</tr>
<tr>
<td>giraffes01</td>
<td>38.5</td>
<td>46.7</td>
<td>-25.4</td>
<td>44.2</td>
<td>33.3</td>
<td>-3.0</td>
</tr>
<tr>
<td>goats01</td>
<td>47.3</td>
<td>61.5</td>
<td>55.8</td>
<td>61.0</td>
<td>61.5</td>
<td>27.8</td>
</tr>
<tr>
<td>horses02</td>
<td>63.1</td>
<td>81.8</td>
<td>18.3</td>
<td>67.3</td>
<td>90.9</td>
<td>36.1</td>
</tr>
<tr>
<td>horses04</td>
<td>59.3</td>
<td>82.1</td>
<td>1.5</td>
<td>60.0</td>
<td>87.2</td>
<td>-5.9</td>
</tr>
<tr>
<td>horses05</td>
<td>41.0</td>
<td>28.6</td>
<td>-10.3</td>
<td>38.7</td>
<td>19.0</td>
<td>11.6</td>
</tr>
<tr>
<td>lion01</td>
<td>53.5</td>
<td>57.1</td>
<td>12.7</td>
<td>63.3</td>
<td>100.0</td>
<td>-3.4</td>
</tr>
<tr>
<td>marple12</td>
<td>68.6</td>
<td>100.0</td>
<td>-15.6</td>
<td>58.6</td>
<td>83.3</td>
<td>-0.9</td>
</tr>
<tr>
<td>marple2</td>
<td>67.7</td>
<td>66.7</td>
<td>33.5</td>
<td>55.6</td>
<td>55.6</td>
<td>47.3</td>
</tr>
<tr>
<td>marple4</td>
<td>68.2</td>
<td>100.0</td>
<td>-3.9</td>
<td>58.9</td>
<td>100.0</td>
<td>3.0</td>
</tr>
<tr>
<td>marple6</td>
<td>53.7</td>
<td>57.9</td>
<td>17.6</td>
<td>41.8</td>
<td>21.1</td>
<td>22.0</td>
</tr>
<tr>
<td>marple7</td>
<td>58.1</td>
<td>78.6</td>
<td>-1.8</td>
<td>35.6</td>
<td>21.4</td>
<td>-17.5</td>
</tr>
<tr>
<td>marple9</td>
<td>66.0</td>
<td>100.0</td>
<td>2.7</td>
<td>41.5</td>
<td>16.7</td>
<td>-3.9</td>
</tr>
<tr>
<td>people03</td>
<td>62.4</td>
<td>87.5</td>
<td>23.3</td>
<td>56.2</td>
<td>100.0</td>
<td>7.8</td>
</tr>
<tr>
<td>people1</td>
<td>88.2</td>
<td>100.0</td>
<td>1.6</td>
<td>96.7</td>
<td>100.0</td>
<td>2.6</td>
</tr>
<tr>
<td>people2</td>
<td>82.1</td>
<td>100.0</td>
<td>-7.2</td>
<td>81.3</td>
<td>100.0</td>
<td>-0.6</td>
</tr>
<tr>
<td>rabbits02</td>
<td>50.2</td>
<td>66.7</td>
<td>-5.9</td>
<td>55.5</td>
<td>75.0</td>
<td>-4.9</td>
</tr>
<tr>
<td>rabbits03</td>
<td>45.4</td>
<td>40.0</td>
<td>-7.7</td>
<td>59.7</td>
<td>100.0</td>
<td>-3.5</td>
</tr>
<tr>
<td>rabbits04</td>
<td>44.0</td>
<td>50.0</td>
<td>31.2</td>
<td>50.9</td>
<td>57.1</td>
<td>26.6</td>
</tr>
<tr>
<td>tennis</td>
<td>70.9</td>
<td>100.0</td>
<td>-4.9</td>
<td>78.1</td>
<td>100.0</td>
<td>-2.4</td>
</tr>
<tr>
<td>Mean</td>
<td>63.6</td>
<td>78.2</td>
<td>4.9</td>
<td>62.2</td>
<td>72.4</td>
<td>7.02</td>
</tr>
</tbody>
</table>

Table 6: Performance on FBMS59 [26]: Per category performance on FBMS59.

IEEE TRANSACTIONS ON IMAGE PROCESSING, 10(2), 2001. 4


Figure 5: **Experiment on FBMS in ideal conditions**: The first row shows some samples of input images, the second row their idealized optical flows, and the latter row the segmentation generated by our algorithm.

Figure 6: **Experiment on SegTrackV2 in ideal conditions**: The first row shows samples of input images, the second row the idealized optical flows, and the latter row the segmentation generated by our algorithm.

---


