# Unsupervised computation of salient motion maps from the interpretation of a frame-based classification network

Etienne Meunier
etienne.meunier@inria.fr
Patrick Bouthemy
patrick.bouthemy@inria.fr

Inria
Centre Rennes - Bretagne Atlantique
Rennes, France

**Abstract**

We introduce a new paradigm for motion saliency (MS) which is an important issue in dynamic scene analysis. We formulate MS as a meta-task that can be instantiated for different tasks usually handled independently. To support this claim, we have addressed two important computer-vision problems with this MS paradigm: independent motion segmentation and anomalous motion detection in videos. We estimate MS from the interpretation of a frame-based saliency classification network with optical flow (OF) as input. Our paradigm can accommodate a given form of motion saliency by simply training the frame-based classification network on the corresponding task. Moreover, our MS estimation is unsupervised, as it does not require any ground-truth saliency maps for training. In addition, we have designed an original two-step network interpretation method, which supplies the binary salient motion segmentation. Finally, we recover the valued motion saliency map using a parametric flow inpainting method. Experimental results on real videos and comparison with existing methods assess the performance of our method.

## 1 Introduction and related work

In this paper, we introduce a novel paradigm to compute salient motions in video frames based on optical flow. Motion saliency (MS) aims to highlight local motions departing from their surrounding context, thus prone to reveal a significant event. MS has many applications in computer vision. It can be useful in the navigation of autonomous vehicles to anticipate moving obstacles, or for public safety to trigger alert in abnormal situations. It can facilitate the subsequent analysis of videos where motion may play the key discriminative role.

We formulate MS estimation as a meta-task that can encompass different tasks usually handled independently. More specifically, we have addressed two important computer-vision problems within this same MS paradigm: independent motion segmentation and anomalous motion detection in videos. Existing supervised methods implies heavy manual annotation. If unsupervised, they need several kinds of input including object appearance, motion, depth or camera parameters. Our method is unsupervised and estimates motion saliency maps from the interpretation of a frame-based classification network with optical flow as only input.

Several topics can be related to MS. Video saliency (VS) highlights moving objects of interest throughout a video. VS is by definition object-oriented, and therefore mixes appearance and motion characteristics. Past approaches such as [14] exploit image motion boundaries and appearance models to recognize salient moving objects throughout the video. In [4], the authors assume that salient objects have distinctive low-level appearance and motion features (orientation and magnitude for flow vectors) than the background, and use a Tukey-inspired measure to detect outlier pixels in the images, and label them as belonging to moving objects. Deep learning approaches have also been investigated. In [24], Wang et al. have developed a CNN-based method explicitly exploiting the spatial and temporal dimensions, yet without computing any optical flow. In [8], spatio-temporal deep features are learned to predict dynamic saliency in videos within a multi-scale segmentation strategy.

Video object segmentation (VOS) is another related topic. Both object appearance and motion are again important factors for VOS, but, the focus is put here on segmenting a moving object of prime interest in the foreground. The availability of large annotated datasets [16], make possible the use of supervised deep-learning techniques for VOS. In [18], one convolutional and one recurrent network are jointly trained to segment moving objects. A close formulation is proposed in [7] with a double supervision combining the saliency module with an optical flow module. In [26], the authors set up an adversarial framework between a generator network producing a hiding mask on the optical flow, and an impainter network trying to inpaint the flow inside the mask. The rationale is that independent motion cannot be predicted by the surrounding motion.

In this paper, we are concerned with the computation of salient motion maps from optical flow. Motion saliency (MS) cannot be formalized as an absolute notion. A motion is not salient in itself, but only with respect to its context, i.e., its surrounding motions or sometimes a predefined normal motion, and to a given application. A motion salient in a given context might not be salient in other contexts. In addition, the moving entity may not necessarily be a specific object in a scene, and its appearance may not be distinctive from the background content, e.g., someone in a crowd, a cell among others, a cloud among others.

As aforementioned, we will handle two different cases of salient motion. The first one is the salient motion produced by independently moving objects in a scene observed by a moving camera. It is obviously related to moving object segmentation [22, 26]. Since we will rely on the optical flow only, a specific problem arises with a mobile camera: motion parallax attached to static objects being at the forefront of the scene and image motion of independently moving objects in the viewed scene both generate distinguishing patterns in the computed optical flow. Indeed, image motion depends on both the relative 3D motion between scene objects and camera, and on the object depth.

A solution could be to compute the static scene flow, that is, the image motion of the whole static scene induced by the camera motion as in [17]. Then, independently moving objects can be identified against this static scene flow. The authors of [2] only compensate the rotational component of the camera motion. Those techniques enable the extraction of moving objects and are robust to motion parallax. However, they imply that the intrinsic parameters of the camera are available and an accurate estimation of depth and ego-motion is provided. In addition, they are dedicated to this problem. We will solve the problem of motion parallax with unknown scene geometry and unknown camera motion.

The second case that we will investigate is salient motion issued from a distinctive motion within a coherently moving set. The set can be typically human crowd, animal herd, bird flock, vehicle traffic, or cell set. An example is the detection of crowd anomaly, where motion saliency comes from a person moving differently than the surrounding ones [3, 9, 15].

Our main contributions are the following. We estimate MS from the interpretation of a frame-based saliency classification network that takes optical flow (OF) as input. Our paradigm can accommodate a given form of motion saliency by simply training the frame-based classification network on the corresponding task. Moreover, our MS estimation is unsupervised, since it does not require any ground-truth saliency maps for training. In addition, we have designed an original two-step network interpretation method, which supplies the binary salient motion segmentation. Finally, we compute the motion saliency map using a parametric flow inpainting method.

The rest of the paper is organized as follows. In Section 2, we will present our original framework involving several stages. We report experimental results with comparison to recent methods in Section 3. Section 4 will contain discussion and concluding remarks.

# 2 Method description

We proceed in two main stages to estimate motion saliency. First, we segment the salient motions from the two-step interpretation of the classification network that predicts the presence or the absence of salient motion in every frame of a video. Then, we compute the salient motion maps using a parametric flow inpainting technique. In the Results section, we will designate our method by the acronym NIMS (Network Interpretation for Motion Saliency).

## 2.1 Frame-based motion-saliency classification

We describe our frame-based classification network inspired from [10]. Basically, it involves the same layers comprising three convolutional blocks and a fully connected layer. However, it takes as input the optical flow field computed between two successive frames, and not a residual flow as in [10]. Thus, it is not necessary to compute any dominant motion resulting from the camera motion, which may introduce bias in case of complex background.

The classification network being shallow, the training is fast. For the first studied case of salient motion, it is trained to predict the presence of independent motion in a frame. It is thus implicitly able to discard motion parallax. In practice, we fine-tuned the network of [10] using the same training set as in [10]. For the second studied case, our network is trained from scratch to detect the presence of distinctive motion in crowds. We used the crowd synthetic dataset introduced in Section 3.3.

Our method is versatile in the sense that we just need to re-train the classification network on the task under study to deal with the salient motion of interest. The classification itself is supervised but the annotation process is extremely light. It boils down to labeling at the frame level, and in practice, it can be almost automatically completed, since the training dataset is composed of videos that were either fully dynamically salient or fully non salient.

We further modify the classification network for the interpretation. Following advice on applying interpretation techniques to deep neural networks [6], we modify the ordering of the inner layers as explained in Fig.1, which results in a functionally identical network but able to provide more operable interpretation maps.

## 2.2 Inference of interpretation maps for salient motion segmentation

Since the classification network predicts correctly the dynamically salient frames and the non salient ones (with an aerage precision of about 90% in our experiments), we can deduce that

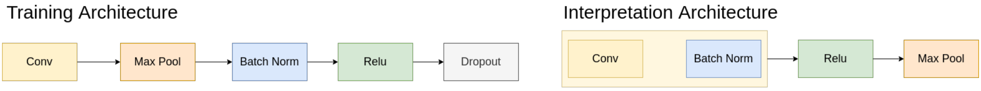

Figure 1: Modification of the ordering of the network inner layers. Left: Order for training and test. Right: Order for interpretation. Weights of each layer remain unchanged. For interpretation, Convolution and Batch Norm layers are merged into a functionally equivalent convolutional layer. A mathematical justification is given in the supplementary material.

it has learned to distinguish salient moving entities. Then, we can exploit this knowledge to locate the salient motions in the frame. Specifically, interpretation techniques allow one to generate attribution maps from the trained neural network. Those maps outline which parts of the input really contribute to the prediction. However, we are not dealing with images as usual but with 2D optical flow fields. Our goal is to determine for segmentation purpose which vectors of the optical flow field induce the prediction as dynamically salient frame.

Several techniques exist for network interpretation [5]. However, to the best of our knowledge, such a network interpretation has not yet been investigated for an optical flow input. An important issue is to extract interpretation information linked to the inner workings of the trained network, rather than to extract instance-specific structures information [5]. Several works such as [1] showed that in some cases, interpretation maps produced with a trained model could be visually similar to the ones generated by a random model. In our case, we did not observe such a behaviour, as shown by experiments reported in the supplementary material. Integrated Gradients method [20] is a technique where maps are generated by cumulating gradients along a linear interpolation between a chosen baseline and a given input. Even though this technique delivers promising results on multiple datasets, results vary widely depending on the chosen baseline [19]. In the case of images, the default baseline is a black image that represents the absence of features. However, regarding optical flow, there is no obvious choice to represent the absence of salient motion. In our experiments, applying the Integrated Gradients method on optical flow fields mainly led to highlighting areas with large flow magnitude.

#### 2.2.1 First step: point-wise computation of attribution maps

Instead, we have adopted the layer-wise relevance propagation (LRP) method [11]. It supplies less noisy attribution maps and removes the need to choose a baseline. This technique propagates a relevance score from the output layer to the input of the convolutional network. It relies on purposely designed propagation rules [11].

We have followed the structure presented in [5], applying the $z^+$ rule to all convolutional and fully-connected layers except for the first convolutional layer were we apply the $z^\beta$ rule. The $z^+$ rule only takes into account positive weights during the propagation of relevance. The $z^\beta$ rule propagates positive relevance through layers that take as input negative values, by using an additive term for the extremal admissible values of the input space. The expression of these rules are given in the supplementary material. Max-pool layers are handled with a winner-takes-all strategy. In our case, we do not need to deal with normalisation layers as we merged them with convolutional layers.

We come out with attribution maps computed on a point-wise basis. However, these maps may be noisy, fragmented or even incomplete. To get an output of the network interpretation usable as salient motion segmentation maps, we need to introduce a second step in the network interpretation process.

### 2.2.2 Second step: transforming attribution maps into salient motion segments

Depending on the quality of the attribution maps and possibly the size of the salient moving objects, this second step may imply a light or a strong use of an additional representative flow information, but it is still driven by the attribution values. As described below, this additional flow information is straightforwardly derived from the input optical flow, and it corresponds to respectively a weighted mean flow vector or a hierarchy of affine motion models.

We proceed as follows for the light version of this second interpretation step. Using the square of the attribution values as weights, we compute over the whole frame the weighted average $\tilde{w}$ of the flow vectors, which is likely to adequately represent the salient motion in the frame. In case of several types of salient motion, several modes should be sought for. Then, for every point $p = (x,y)$ and its flow vector $w_p = (u_p, v_p)$, $p$ belongs to a candidate region if $\alpha_1 < (w_p.\tilde{w}/||\tilde{w}||_2^2) - 1 < \alpha_2$. In practice, we take $\alpha_1 = -0.01$ and $\alpha_2 = 4$. We set asymmetric threshold values because flow orientation and magnitude are both involved making the problem a bit tangled.

The other more elaborate version of the second interpretation step relies on a hierarchical parametric motion segmentation. Seminal works on image motion segmentation into layers or into regions [13, 23] took two successive images as input and estimated an affine motion model per layer or per region. Here, we have the optical flow as input, and we segment it by iteratively estimating affine motion models fitting the optical flow field, using a robust regression involving the Huber function $\mathcal{H}$. More precisely, we minimize the function:

$$\sum_{p \in \Omega_k} \mathcal{H}(a_1^k + a_2^k x + a_3^k y - u_p) + \mathcal{H}(a_4^k + a_5^k x + a_6^k y - v_p), \tag{1}$$

where $\Omega_k$ is the image part intervening at iteration $k$, and the $a_i^k$'s the six parameters of the affine motion model computed at iteration $k$. At each iteration $k$, inliers are selected to form a layer. Inliers are pixels satisfying the constraint $\frac{1}{2}(\gamma_u^k |u_p - u_p^k| + \gamma_v^k |v_p - v_p^k|) < \beta$, where $(u_p^k, v_p^k)$ are the coordinates of $w_p^k$, the flow vector given at $p$ by the affine motion model estimated at iteration $k$. $\gamma_u^k$ and $\gamma_v^k$ are the standard deviations over the differences of the $u$ and $v$ components at iteration $k$. Those factors were introduced to avoid being affected by diverse motion magnitudes. In addition, it allows us to set the threshold $\beta$ once and for all, and we take $\beta = 1.6$. Outliers are kept for the next iterations. We continue this procedure until only a small number of outliers remain (in practice 50 points), and we group them together to form the last layer. Following this iterative decomposition, each vector in the initial optical flow field belongs to one unique layer. Finally, we split each layer into connected components to form the candidate regions.

We still will have to select the salient moving regions among the candidate regions, whether it is for the light version or the elaborated one. This will be driven again by the attribution values as follows. As suggested in [5], the network interpretation must be primarily driven by the strength of attribution values rather than their spatial pattern. Therefore, we precisely take into account the attribution values to assign an attribution score to each candidate regions (i.e., segments of the optical flow). It is given by the ratio between the sum of the attribution values in the segment, and the size of the segment. Afterwards, we compute an adaptive threshold on the attribution score to filter out segments with the lowest scores considered as not salient. In practice, we use the score of the biggest region, assumed to correspond to the background, as threshold. This procedure allows us to select several salient segments if needed. Finally, we get a set $\mathcal{R}_{sal}$ of salient moving regions $R_j$.

## 2.3 Estimation of the motion saliency maps

We want now to estimate the motion saliency map. To this end, we will adopt a parametric flow inpainting. More specifically, we will extend within each segmented region of the set $\mathcal{R}_{sal}$ the optical flow surrounding this region, and compare it to the initially computed optical flow inside the region. To achieve it, it is easier to follow a parametric approach. First, we estimate an affine motion model fitting the external optical flow, i.e., the flow outside $\mathcal{R}_{sal}$. We use again the robust estimation of eq.(1), but here, it is applied to $\Omega \setminus \mathcal{R}_{sal}$ where $\Omega$ is the whole image. Then, we leverage this estimated affine motion model to inpaint the flow within all the regions $R_j$ of $\mathcal{R}_{sal}$. The inpainted flow within each $R_j$ is given by: $\forall p = (x,y) \in R_j, (u_p^{imp}, v_p^{imp}) = (\hat{a}_1^{sal} + \hat{a}_2^{sal} x + \hat{a}_3^{sal} y, \hat{a}_4^{sal} + \hat{a}_5^{sal} x + \hat{a}_6^{sal} y)$, where $\hat{a}_i^{sal}$ designates each of the parameters of the affine motion model estimated outside $\mathcal{R}_{sal}$.

The parametric inpainted flow allows us to infer the motion saliency map $\phi$ inside $R_j$ from the flow gap $w_p^{imp} - w_p$, where $w_p$ is the flow vector initially computed at $p$:

$$\text{for each } R_j,\ \forall p \in R_j, \quad \phi(p) = 1 - \exp(-\lambda ||w_p^{imp} - w_p||_2), \tag{2}$$

where $\phi(p)$ values lie within $[0,1]$ and $\lambda > 0$ modulates the visualization of the motion saliency map.$\phi(p)$ is set to 0 for all the pixels that do not belong to the set $\mathcal{R}_{sal} = \{R_j\}$. We use $\lambda = 0.15$ for all presented visualisations.

# 3 Experimental results

## 3.1 Implementation details

Optical flow fields are computed using the RAFT method [21]. For LRP, we adopted a modified version of the implementation described in [5]. We used Captum to test Integrated Gradients and Scikit Learn to implement the Huber robust regression. For the first case study, we transferred the model and weights from [10] onto Pytorch framework, and fine-tuned the last fully connected layer using the original training dataset. We trained for 2 epochs, requiring approximately 20 min on a GPU GeForce MX150. For the second case study, we trained the network form scratch, which took 4.5 hours on a GPU GeForce RTX2080Ti. The average runtime per frame is 0.70s on a CPU 1.90GHz.

## 3.2 First case study: independent scene motion

First, we want to objectively evaluate the accuracy of our method regarding the salient motion case of independent scene motion. Due to the lack of dedicated benchmarks, we make do with the VOS DAVIS2016 dataset[1] as is also done by VS methods. The DAVIS2016 videos involve one single independently moving object in the foreground.

Let us stress that the DAVIS2016 training set has not been used for training our classification network. In this experiment, we use the more elaborated second step of the interpretation

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

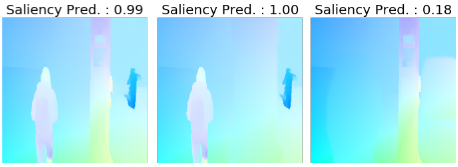

Figure 2: Starting from the optical flow (left), we replace the flow in the selected regions by inpainting. When the tree-trunk region in the foreground is ablated, the frame is still classified as dynamically salient (middle). In contrast, ablating regions of the two moving people switches the frame as non salient (right). See the prediction score above each frame.

process described in subsection 2.2.2. The bitmap of the salient moving regions $R_j$ extracted by our method is compared to the ground truth. Results obtained with our method and other unsupervised methods are collected in Table 1. Our method outperforms comparable unsupervised methods on the VOS benchmark as BGM [25], TIS$_0$ [4], and FTS [14]. On our side, we do not resort to any postprocessing step and do not rely on appearance contrary to the CIS and TIS$_s$ methods. Indeed, the CIS method involves an important post-processing stage and its performance without the postprocessing, evaluated with the publicly available code, drops to $\mathcal{J}$Mean $= 59.7$ on the complete DAVIS2016 dataset and to 59.2 on the validation set, that is, both lower than our scores. Besides, our method is penalized by the ground truth of DAVIS2016 focused on the primary object. For instance, the ripples on the water in the flamingo sequence and the foam under the kytesurfer (Fig.4) are recognized as salient motions by our method, which is correct.

Table 1: Results on the complete DAVIS2016 dataset for several unsupervised methods (scores taken from [4] and [26]). In brackets on the DAVIS2016 validation set only. J is the Jaccard index (region similarity) and F accounts for contour accuracy. For further explanation on the evaluation metrics, we refer the reader to the DAVIS2016 website.

| | CIS [26] | TIS$_s$[4] | TIS$_0$[4] | BGM[25] | FTS [14] | NIMS (Ours) |
|---|---|---|---|---|---|---|
| $\mathcal{J}$ Mean ↑ | - (71.5) | 67.6 (62.6) | 58.6 (56.2) | 62.5 (-) | 57.5 (55.8) | 63.0 (60.2) |
| $\mathcal{F}$ Mean ↑ | - (70.5) | 63.9 (59.6) | 47.5 (45.6) | 59.3 (-) | 53.6 (51.1) | 68.2 (65.8) |
| $\mathcal{J}$ Recall ↑ | - (86.5) | 84.7 (-) | 75.9 (-) | 70.0 (-) | 65.2 (64.9) | 76.7 (73.2) |
| $\mathcal{F}$ Recall ↑ | - (83.5) | 78.5 (-) | 48.8 (-) | 66.2 (-) | 57.9 (51.6) | 80.1 (75.6) |

We want to further demonstrate the ability of our method to distinguish between independent motion and motion parallax. The latter being rarely present in DAVIS2016, we built a dataset of videos taken with a hand-held moving camera.Those videos are taken in an urban environment with complex backgrounds, where at the same time static objects in foreground induce motion parallax and other objects undergo independent motions. Examples of those videos are given in Figs.2-3-4 and in the supplementary material.

First, we assess the performance of the classification network in itself by performing an ablation test. We draw a mask around an area depicting distinguishable motion pattern and inpaint the flow within the mask as done in Section 2.3, hiding this prominent flow from the network. The procedure is illustrated in Fig.2. We can observe that removing only moving objects lowers the saliency score, and makes the network predict the frame as "non salient". Conversely, when we remove the static object causing motion parallax, the prediction of the network remains "salient frame" due to the presence of moving objects. It shows that the frame-based classification network truly bases its prediction on the presence of independent

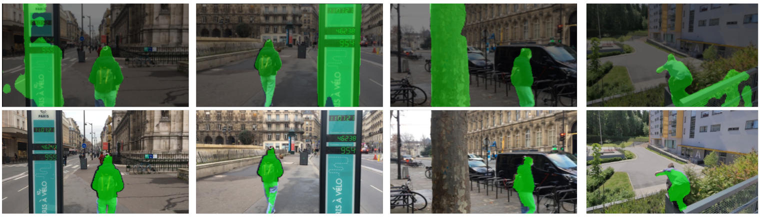

Figure 3: Extracted salient masks, superimposed on original images, on three examples from our dataset and one from DAVIS2016 (Parkour). Top row : results from [26] using the available code and model (without the postprocessing step). Bottom row : corresponding results using our method (NIMS). CIS [26] detects the foreground static object due to parallax motion, while our method succesfully discards it. The sequence of the third example is available in the Supplementary Material.

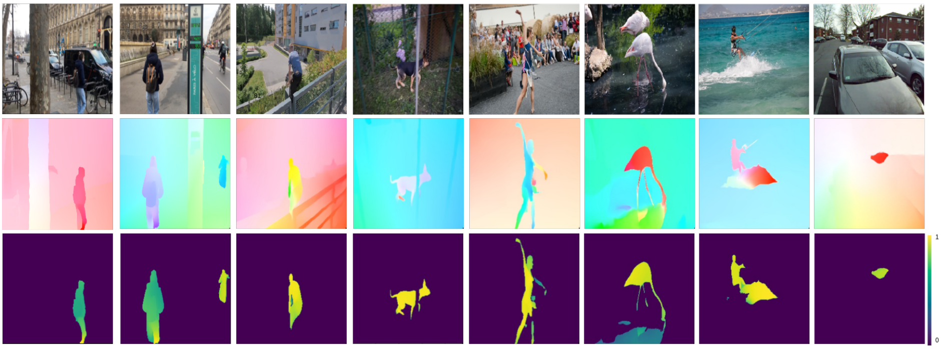

Figure 4: Results obtained with our method (NIMS). From left to right, two outdoor videos we acquired, five examples from the DAVIS2016 dataset, respectively, Parkour, Libby, Dance-Twirl, Flamingo and Kite-surf, and the Drive video of the Complex dataset. Top row: one image of the video. Middle row: Optical flow in HSV color code; Bottom row: Motion saliency maps. The closer to yellow, the higher the motion saliency degree.

motion. Secondly, we visually evaluate the localisation of the salient moving regions. Fig.3 contains a comparison between the motion saliency segmentation performed by our method, NIMS, and by the CIS method presented in [26]. We can note that our method is able to correctly extract the person moving as a salient moving region and not the pole, the tree trunk or the fence in the foreground. In contrast, the method [26] also segments them as independently moving objects.

Finally, we display samples of motion saliency maps in Fig.4. They were extracted from several datasets, respectively, our video dataset, the DAVIS2016 dataset and the Complex dataset [12]. The closer to yellow, the higher the motion saliency degree given by $\phi(p)$ in eq.(2). Let us stress that the static foreground objects are not found salient, whereas their optical flow is prominent since it conveys parallax motion. The computed salient motion maps are representative of the underlying motions.

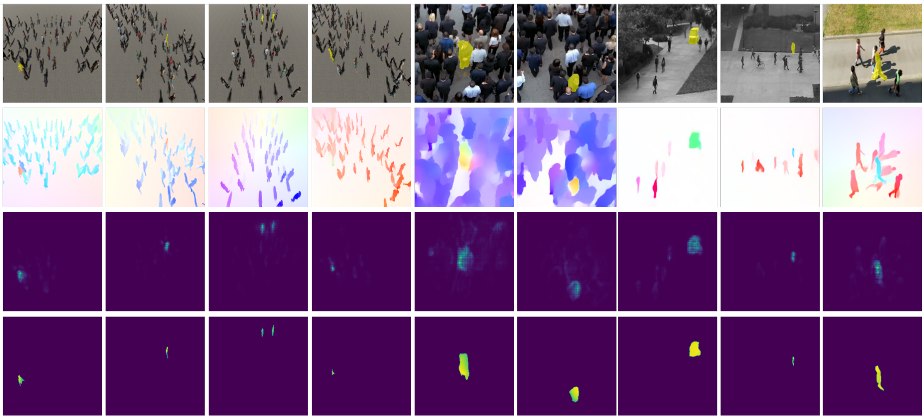

Figure 5: Results obtained with our method (NIMS). Top to bottom row : one image of the video with the salient-motion ground-truth superimposed in yellow, optical flow displayed with the HSV color code, LRP attribution map, computed salient motion map. Left to right: four examples from the synthetic dataset (4th column: the outlier pedestrian is strongly occluded and only the foot is consequently recovered), two samples from the Wrong Way video at two distant time instants, two examples from the UCSD Anomaly Detection dataset, and an example from an outdoor scene acquired for the experiment. Several videos are provided in the supplementary material.

## 3.3 Second case study: distinctive motion in a crowd

In order to train our network and to evaluate the ability of our method to detect distinctive motion within a crowd, we built a dataset made of computer-generated sequences from 3D scenes depicting crowds in motion, using the public software ChAOS [2]. The resulting dataset is not trivial, since the salient moving persons are small and not always easily discernible. It is composed of 100 videos of about 250 frames each, taken by a camera in motion. In each video, the crowd of people is walking in a common, randomly chosen direction. In order to simulate distinctive motion, one to three people walking in a different direction are randomly added to some videos. Frames including a distinctive motion are dynamically salient.

The first step is to train the frame-based saliency classification network on this dataset for the new task. This randomly initialized network follows the structure defined in subsection 2.1, except that we used InstanceNorm layers instead of BatchNorm to tackle changes in input flow magnitudes. After training, the network is able to recognize frames containing a distinctive motion with 90.6% of accuracy on the test set. We apply the LRP interpretation to the classification network and extract attribution maps. The latter exhibit a low level of noise and mostly highlight salient motions. Therefore, we use the light version of the second step of the network interpretation process described in subsection 2.2.2.

Figure 5 displays samples of attribution and saliency maps computed with our method on videos of the crowd dataset. We can observe that these maps correctly delineate the salient motion in the crowd despite the small size of the salient area and the global camera

[2] Crowd Animation Open Software : https://project.inria.fr/crowdscience/download/

motion. We have also processed several real videos of a similar kind. Visual results are again collected in Fig.5. We used the classification network trained on the synthetic dataset without any additional fine-tuning. We have processed the Wrong Way video that depicts a man walking against a crowd. As shown in the video frames displayed in the supplementary material, the salient motion is correctly segmented in the vast majority of them. The method fails in case of poorly visible salient motion. On the other hand, when a woman moves aside to avoid the opposite individual, she produces the main salient motion. We also processed videos from the UCSD Anomaly Detection dataset[3] fitting this second task. Again, correct salient motion masks are recovered, whether it be the vehicle or the pedestrian walking in the opposite direction. Let us note that in the later video, the anomaly regarding our task is precisely this pedestrian and not the cyclist as defined in the UCSD dataset benchmark.

In addition, we compared our method with a segmentation U-Net network trained in a supervised way using the same synthetic dataset and ground-truth masks corresponding to the salient moving objects. The objective is to build a gold standard, and to achieve a quantitative evaluation on the synthetic dataset. We use again as evaluation metric the Jaccard index $\mathcal{J}$Mean. Let us note that high evaluation scores cannot be expected, because salient moving objects are frequently occluded. For a fair but strict evaluation, we compute it only on salient frames correctly classified as dynamically salient (since for non-salient frames the Jaccard index is 1, the ground-truth being empty). We get $\mathcal{J}\text{Mean} = 67.1$ for the supervised network, and $\mathcal{J}\text{Mean} = 53.6$ for our unsupervised method, which demonstrates that our method performs well, given the relatively small gap between the two scores.

# 4 Conclusion

We have defined an original, versatile and efficient method for the estimation of salient motion maps based on the interpretation of a simple frame-based classification network. It only requires the optical flow as input. The computation of the salient motion maps is unsupervised as it does not require any ground-truth on the segmentation maps. To tackle a given motion-saliency task related to a given application, it is sufficient to re-train the classification network, the overall framework remaining unchanged. We have demonstrated the performance of our method on two different case studies. Our method was thus able to disentangle parallax motion and independent scene motion without any 3D information. It can also handle distinctive motion in a crowd. Experimental results demonstrated that we are able to compute accurate and reliable salient motion maps which convey rich, readily available information on the nature of the salient motion, through the flow gap of the inpainting step, and not only the degree of saliency.

**Acknowledgements**
This work is financially supported by BpiFrance through the Lichie contract.

# References

[1] Julius Adebayo et al. Sanity checks for saliency maps. *arXiv:1810.03292*, Nov. 2020.

[2] Pia Bideau, Rakesh R. Menon, and Erik Learned-Miller. MoA-Net: Self-supervised motion segmentation. In *European Conference on Computer Vision Workshops (ECCVW)*, 2018.

[3] http://www.svcl.ucsd.edu/projects/anomaly/dataset.html

[3] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah. Anomaly detection in video via self-supervised and multi-task learning. *arXiv:2011.07491*, March 2021.

[4] Brent Griffin and Jason Corso. Tukey-inspired video object segmentation. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, Waikoloa Village, HI, USA, January 2019.

[5] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of CNNs via contrastive backpropagation. In *Asian Conf. on Computer Vision (ACCV)*, Perth, Australia, 2018.

[6] Mathilde Guillemot et al. Breaking batch normalization for better explainability of deep neural networks through layer-wise relevance propagation. *arXiv:2002.11018*, Feb. 2020.

[7] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.

[8] Trung-Nghia Le and Akihiro Sugimoto. Video salient object detection using spatiotemporal deep features. *IEEE Transactions on Image Processing*, 27(10):5002–5015, October 2018.

[9] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan. Crowded scene analysis: A survey. *IEEE Trans. on Circuits and Systems for Video Technology*, 25(3): 367–386, March 2015.

[10] Léo Maczyta, Patrick Bouthemy, and Olivier Le Meur. CNN-based temporal detection of motion saliency in videos. *Pattern Recognition Letters*, 128:298, December 2019.

[11] Grégoire Montavon et al. Layer-wise relevance propagation: An overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, vol.11700:193–209. Springer, 2019.

[12] Manjunath Narayana, Allen Hanson and Erik Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *Int. Conference on Computer Vision (ICCV)*, Sydney, 2013.

[13] Jean-Marc Odobez and Patrick Bouthemy. MRF-based motion segmentation exploiting a 2D motion model robust estimation. In *Int. Conf. on Image Processing (ICIP)*, Washington DC, 1995.

[14] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *IEEE Int. Conf. on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[15] Juan-Manuel Pérez-Rúa, Antoine Basset, and Patrick Bouthemy. Detection and localization of anomalous motion in video sequences from local histograms of labeled affine flows. *Frontiers in ICT: Computer Image Analysis*, May 2017. doi: 10.3389/fict.2017.00010.

[16] Federico Perazzi et al. A benchmark dataset and evaluation methodology for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, June 2016.

[17] Anurag Ranjan et al. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, 2019.

[18] Hongmei Song et al. Pyramid dilated deeper ConvLSTM for video salient object detection. In *European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018.

[19] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, January 2020.

[20] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *34th International Conference on Machine Learning (ICML)*, Sydney, Australia, August 2017.

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

[22] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.

[23] John YA Wang and Edward H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, Sept.1994.

[24] Wenguan Wang, Jianbing Shen, and Ling Shao. Video salient object detection via fully convolutional networks. *IEEE Transactions on Image Processing*, 27(1):38–49, January 2018.

[25] Scott Wehrwein and Richard Szeliski. Video segmentation with background motion models. In *British Machine Vision Conference (BMVC)*, London, UK, 2017.

[26] Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto. Unsupervised moving object detection via contextual information separation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA 2019.