CNN-based temporal detection of motion saliency in videos

Léo Maczyta, Patrick Bouthemy, Olivier Le Meur

CNN-based temporal detection of motion saliency in videos

Léo Maczyta[1], Patrick Bouthemy[1], and Olivier Le Meur[2]

[1] *Inria, Centre Rennes - Bretagne Atlantique, Campus de Beaulieu, Rennes, 35042, France*

[2] *Univ Rennes, IRISA Rennes, Campus de Beaulieu, Rennes, 35042, France*

Abstract

The problem addressed in this paper appertains to the domain of motion saliency in videos. However this is a new problem since we aim to extract the temporal segments of the video where motion saliency is present. It turns out to be a frame-based classification problem. A frame will be classified as dynamically salient if it contains local motion departing from its context. Temporal motion saliency detection is relevant for applications where one needs to trigger alerts or to monitor dynamic behaviours from videos. It can also be viewed as a prerequisite before computing motion saliency maps. The proposed approach handles situations with a mobile camera. It involves two main stages consisting first in cancelling the global motion due to the camera movement, then in applying a deep learning classification framework. We have investigated two ways of implementing the first stage, based on image warping, and on residual flow respectively. Experiments on real videos demonstrate that we can obtain accurate classification in highly challenging situations.

1 Introduction

This paper is concerned with motion saliency in videos. Motion saliency information allows one to highlight objects undergoing separate, singular or unexpected motion in the video sequence. The specific problem addressed in this paper is the temporal detection of motion saliency in videos, i.e., determining throughout the video which frames contain motion saliency. This enables to recover time intervals for which motion saliency is present in the video.

To our knowledge, this problem has not been investigated so far, but it is crucial for numerous applications. It can help detecting obstacle irruption for mobile robotics or autonomous vehicles, raising alert for video-surveillance, triggering attention for video analysis, or highlighting relevant information for video summary. This frame-based classification acts as a pre-attention mechanism.

More specifically, our method can be viewed as a prerequisite to the computation of dynamic saliency maps. The existing methods extract salient moving objects, while implicitly assuming that the frame is dynamically salient. Our method precisely addresses the latter issue.

As aforementioned, temporal detection of motion saliency in videos is a classification problem. It consists in deciding for every frame of a video sequence whether it should be labelled as dynamically salient or not, that is, whether it contains elements whose local motion departs from their context. By context, we mean global (or dominant) motion. We will adopt to solve this classification problem the convolutional neural network (CNN) framework. Before classifying each frame of the video, we cancel the global motion usually due to the camera movement. We have investigated two ways of implementing it, based on image warping and on residual flow respectively. This leads to two main variants of our motion saliency detection approach.

The paper is organised as follows. Section 2 contains a brief review of motion saliency methods. Deep learning methods relevant to this problem are also discussed. In Section 3, we present the synthetic and real video datasets built to evaluate our classification scheme. In Section 4, we describe our approach and the two variants based on image warping and on residual flow respectively. Experimental results are reported in Section 5. Section 6 includes concluding comments.

2 Related work

To the best of our knowledge, there is no previous work that focuses on temporal motion saliency detection. Then, we will mainly browse contributions related to the computation of dynamic saliency maps in videos.

Research on dynamic saliency has mainly dealt with highlighting foreground objects moving in front of a static scene ([36, 38]). The scene may nevertheless be not fully static, but may comprise dynamic textures ([5]), such as flags in the wind or ripples on the water, that should not be considered as salient ([36]). Salient motion may be embedded in a global motion, when the camera is moving. A first category of methods compensate the camera motion in the image ([18, 10]). A second class of approaches combines spatial and temporal information without first cancelling the camera motion ([8, 14, 21, 37]). [13] use spatio-temporal cues and represent videos as spatio-temporal graphs with the aim of minimizing a global function. Related problems are the detection of anomalies in crowded scenes ([27, 39]), and the use of eye-fixation maps for visual saliency ([4, 20, 28]).

Deep learning has greatly improved performances in many image processing tasks, notably for image classification ([16, 34, 9]). Video processing applications such as optical flow computation ([11, 33]), moving object segmentation ([36]), automated description of videos ([40]), or action recognition in videos ([31, 30, 19]) showed that convolutional neural networks can also be successfully applied to videos. In particular, deep learning was recently applied to

video saliency tasks. This is for instance the case for [38], who estimate spatiotemporal saliency maps with motion and appearance information. They design an architecture comprising two successive CNNs, the first one dealing with spatial saliency and the second one refining the saliency map with temporal information. [17] use spatiotemporal deep features and define a spatiotemporal conditional random field to estimate saliency maps.

Most of the existing methods assume that motion saliency is attached to a foreground object moving in front of a (mostly) static scene. Then, appearance may play a key role and is consequently exploited. We take a more general definition of motion saliency. It involves all situations where local motion departs from its context with differences or not in appearance. Moreover, existing methods supply valued motion saliency maps. One could consider that they could recognize a dynamically non-salient frame, by providing a "practically" empty output. Yet, it would require designing an additional stage to reliably decide it. Anyway, we experienced that motion saliency maps are hallucinated by existing methods on non-salient frames. This shows that existing methods are not able to address the frame-based motion saliency detection problem.

3 Datasets

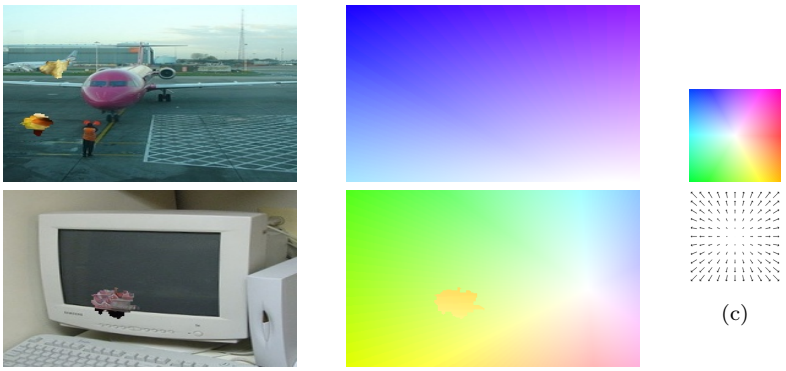
We start by describing the synthetic and real datasets used in our experiments. The dataset is indeed a key point for learning-based approaches, especially in the case of deep learning methods such as the ones we propose in this paper.

3.1 Synthetic dataset

Machine learning methods, and especially deep learning methods, require a training set large enough for successful learning and generalisation ([32]). Recently it was demonstrated for computer vision applications ([6, 22]), that learning can be efficiently achieved on synthetic datasets. We have then built a synthetic dataset for a first training stage of the networks involved in our methods.

Each element of the synthetic dataset consists of a frame pair. The second frame of the pair is obtained by applying to the first frame a parametric motion model. We take an affine motion model. Motion saliency is attached to added patches taken from other images and undergoing a different affine motion. Images of PascalVOC 2012 ([7]) were used to generate this dataset. Samples are provided in Fig. 1. Frames are generated with a probability of 0.5 for absence of motion saliency, 0.25 for presence of one salient moving patch, and 0.25 for presence of two salient moving patches. Added patches have a limited size, of approximately 0.5% to 1.5% of the frame. We deliberately chose to include small-size patches to produce motion saliency examples hard to detect.

By generating salient examples in this manner, the risk could be that the network manages to detect salient frames thanks to the appearance of the added patches generally different from the content of the reference frame. To avoid this, non salient frames can also contain up to two added patches coming from



(a) (b) (c)

Figure 1: Samples of the synthetic video dataset. (a) The top frame is not salient while the bottom frame is salient. Both include added patches cropped from another image, but only the one in the second example undergoes an independent motion. (b) The corresponding motion fields are represented with the colour code depicted in c). The optical flow is globally smooth in the top row, while it includes an outlier patch in the bottom row. (c) Colour code for the flow field represented below.

unrelated images, but still undergoing the global motion. The shape of the patches extracted from other images was generated randomly, as illustrated in Fig. 1.

In order to maximise the variability of the training samples, frame pairs are generated on the fly during training. Approximately 4 million training samples are generated this way. The validation and test sets contain 2000 frame pairs each.

3.2 Dataset of real videos

A dataset with real videos is necessary for fine-tuning the networks and to assess the motion saliency classification methods. Since the proposed methods will have to handle videos with a mobile camera, the camera should be moving for a significant part of the dataset. Moreover, the ground truth has to supply the presence of motion saliency or not at the frame level. The dataset constructed by [2] meets these requirements. It gathers FBMS-59 ([24]), Complex Background ([23]) and Camouflaged Animals ([3]), all of them being specifically re-annotated for the problem of moving object segmentation. This dataset contains difficult examples, in particular the videos of Camouflaged Animals for which motion information strongly prevails in the perception of the salient moving animals. We label a frame as dynamically salient in the ground truth, if it contains at least one independently moving object in the ground truth of the original dataset.

Since our objective is to detect temporal intervals of motion saliency in

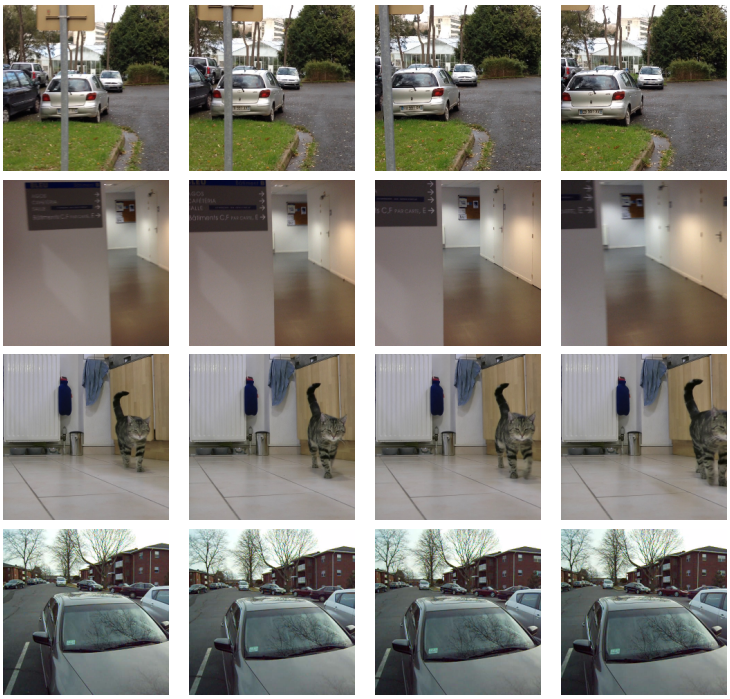


Figure 2: Four samples of the real dataset. The two first examples involve non salient frames. The two last ones comprise dynamically salient frames (respectively, including a moving cat, and a (small) moving car behind the parked cars). Four consecutive frames are displayed for each example.

videos, non salient frames are required to serve as negative examples. Non salient frames are rare in the dataset of [2]. Consequently, we acquired 71 additional videos with no salient frames, i.e., depicting static scenes, but acquired with a mobile camera. These videos depict indoor, urban and natural scenes. The ground-truth is by construction available for all the frames of the 71 additional non-salient videos.

The final dataset includes 144 videos, and is split in a training set of 94 videos, a validation set of 13 videos and a test set of 37 videos, for a total of 3451 labelled frames. The three sets contain approximately the same amount of salient and non salient frames. During training, data augmentation with resizing, cropping, temporal inversion and flipping around a vertical axis, was applied. The batches used to train our networks are built so that salient and non salient frames are correctly balanced. Samples are displayed in Fig. 2.

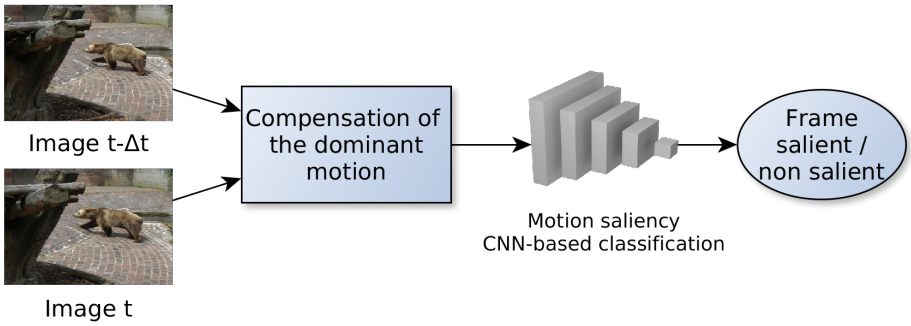


Figure 3: Our overall motion saliency detection framework

4 Temporal motion saliency detection

Our overall approach is summarized in Fig. 3. Its core module is the frame classification, which will be achieved with a convolutional neural network (Section 4.1). We have investigated two ways to implement the global motion compensation: image warping (Section 4.2), and residual flow (Section 4.3).

We assume that the dominant motion (or global motion) can be represented by a single affine motion model. We have considered two parametric dominant motion estimation algorithms, Motion2D and DeepDOM, described in Section 4.4.

In addition, we have designed two baselines for comparison purpose, since there are no existing methods available for temporal motion saliency detection. The first baseline merely thresholds the average residual flow magnitude (Section 4.5). It essentially allows us to assess the difficulty of the problem. The second baseline is defined in Section 4.6. It leverages the two-stream network introduced in [31].

4.1 Frame classification on motion saliency

Due to the clear superiority of CNN-based methods in any image classification problem to date, we adopt a CNN framework for the frame classification issue attached to the temporal detection of motion saliency. The input will be specified for each method of temporal motion saliency detection. The input will be formatted as 240x240 frames. We defined the structure of the CNN for classification (Fig. 4) from preliminary experiments on the synthetic dataset. We conclude from these experiments that a too deep network led to overfitting. Therefore, only three convolutional layers are kept, with 7x7 kernels to have large enough receptive fields. The convolutional layers involve respectively 64, 96 and 128 features maps, with a stride of 2. This architecture also comprises max pooling with a stride of 2, batch normalization, the ReLU non-linearity and dropout.

Our objective is to build a flexible framework. As a consequence, we keep

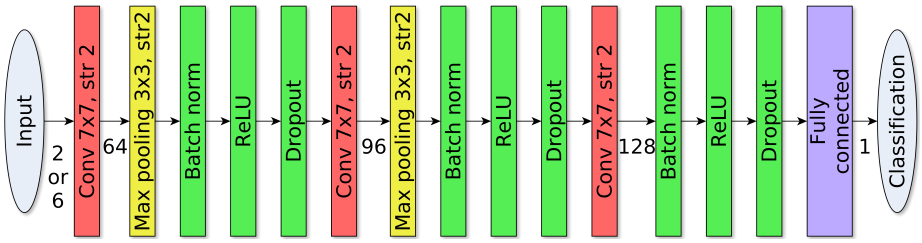


Figure 4: CNN for motion saliency classification ('str' stands for stride). The number of channels are mentioned at the relevant places. The input is either the 2-channel residual flow or the 6-channel concatenation of the two RGB images. The final output is the probabillity of motion saliency, computed with a sigmoid.

the same architecture for the classification in all the motion saliency detection methods, whatever the input data. With this specification, we can also draw a fair comparison of the methods. The CNN architecture was trained for each method with the cross-entropy loss.

The network only supplies the probability of the frame to be dynamically salient. A frame is then classified as dynamically salient if this probability is greater than 0.5, and non salient otherwise. Notwithstanding, we are not facing a threshold setting issue. Comparing to 0.5 is equivalent to taking the maximum of the two probabilities, when their sum is equal to one. We first designed a network with two output yielding probabilities for the dynamically salient and non-salient classes respectively. We experimentally noticed that the sum of the two probabilities was very close to one, without formally imposing this constraint in the network. More precisely, the mean of the absolute difference between 1 and the sum, on the overall test set, was $3.7\,10^{-5}$, and its maximum value amounts to $4.7\,10^{-4}$, knowing that the test set is well balanced between the two classes. Then, for the sake of simplicity and efficiency, we built a network with one single output.

4.2 Motion saliency detection based on image warping

The first way to cancel the dominant motion is to warp the second image of the pair onto the first one. The method will then rely on the CNN to extract relevant features from the aligned frames to infer whether motion saliency is present or not, i.e., the frames are partly or fully aligned.

The first step of this method consists in warping the second frame onto the first one with the estimated affine motion model. Then, the frame classification is performed with the convolutional neural network introduced in Section 4.1. The two colour images, which are the first input image and the second warped image, are concatenated into a 6-channel input.

Depending on the dominant motion estimation algorithm used, this method will be called WS-Motion2D or WS-DeepDOM, with WS standing for Warping-

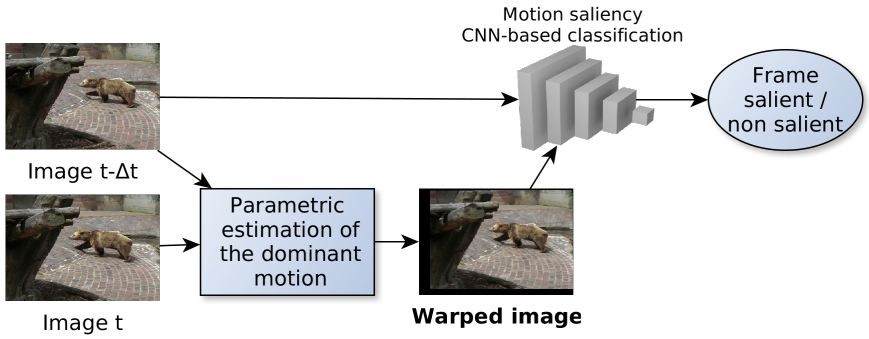


Figure 5: Motion saliency detection based on frame warping

based Saliency. The network is expected to compare the warped frames and to find motion saliency by implicitly searching for misaligned objects due to their independent motion.

4.3 Motion saliency detection based on the residual flow

Our second motion saliency detection method fully acknowledges the fact that motion saliency is first and foremost related to motion. Instead of using colour frames as input, we directly exploit information related to motion. This allows us to be explicitly agnostic on appearance. We compute the residual optical flow, by subtracting the affine flow $\omega_{\hat{\theta}}$, given by the estimated dominant motion model of parameters $\hat{\theta}$, to the computed optical flow field ω:

$$\forall p \in \Omega, \qquad \omega_{res}(p) = \omega(p) - \omega_{\hat{\theta}}(p) \tag{1}$$

where Ω is the image grid. The residual flow will serve as input of the CNN classifier of Fig. 4. This method is summarized in Fig. 6. The two components of the residual flow are the two channels of the input of the classifier. A residual flow close to a zero field over the whole image means a non salient frame.

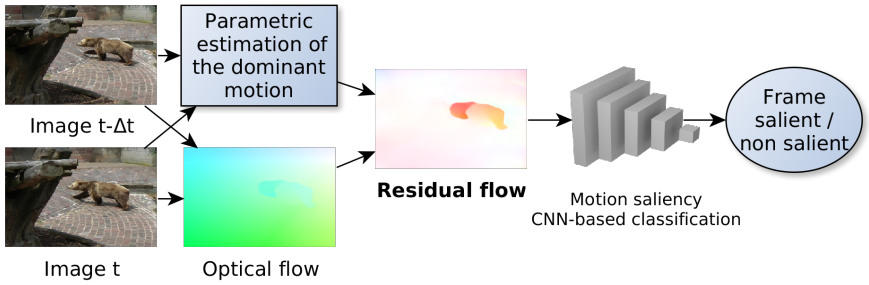


Figure 6: Motion saliency classification based on the residual flow

The optical flow is computed with FlowNet2.0 ([11]). FlowNet2.0 was chosen since it is real time, while delivering good performance. The speed of the optical flow algorithm is indeed critical, as optical flow is computed during training for each pair of every batch. Also, expected use cases of motion saliency detection methods are likely to require a real-time execution. Depending on the motion estimation algorithm used, this method will be denoted RFS-Motion2D or RFS-DeepDOM, with RFS standing for Residual Flow Saliency.

4.4 Parametric estimation of the dominant motion

Most of the time, the image motion induced by the camera forms the dominant motion in the frame. This dominant motion corresponds to the apparent motion of static elements in the scene, which usually occupy the main part of the frame. In case of a shallow scene, i.e., depth and orientation variations in the static scene are small compared to the distance to the camera, a unique 2D parametric motion model, such as an affine model, correctly approximates the dominant motion. We adopt this simple but efficient approach. Results reported in Section 5 will show that it generalises well even for non shallow scenes in practice. Let us note that for a close-up on a moving object, the dominant motion becomes precisely the motion of this object. The temporal motion saliency detection is still valid, since we are interested in detecting local motion departing from the global one. For every pixel $p \in \Omega$ with $p = (x, y)$, the flow field given by the parametric motion model can be written as:

$$\omega_\theta(p) = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix} \tag{2}$$

where $\theta = (a_1, ..., a_6)$ is the vector of the model parameters.

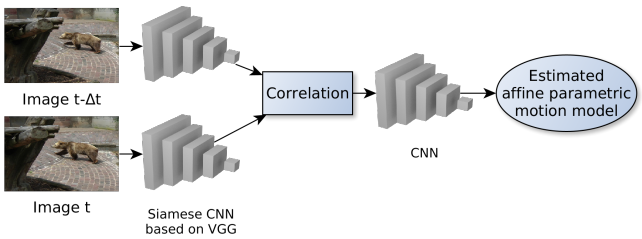


Figure 7: DeepDOM network leveraging the one from [29] for the parametric estimation of the dominant motion.

We used two methods to estimate the affine motion model. We first resorted to the classical robust multi-resolution algorithm Motion2D ([25]) to estimate the dominant affine motion model between two frames. It is real-time and it has proven its efficiency in many applications.

As an alternative, we have applied the network proposed in ([29]), initially designed to estimate geometric transformations between two images, which can

be acquired from very distant viewpoints. The objects to match may even be different instances of the same object class. In our case, the two input images are far closer and the displacements are smaller, but the frames may involve outliers, i.e., independently moving objects. The architecture is summarized in Fig. 7. It was trained separately from the saliency detection methods, with the synthetic dataset described in Section 3.1. The convolutional neural network after the correlation operation is composed of two successive convolutional layers with respectively 7x7 and 5x5 kernels, that are followed by a fully connected layer. This network is called DeepDOM, for Deep Dominant Motion estimation. The loss function used to train this network is similar to the one used in [29]. A grid $\mathcal{G}$ of nodes q is deformed by the estimated global motion of parameters $\hat{\theta}$ and by the ground truth θ_{GT}. The loss function $\epsilon(\hat{\theta})$ compares the two sets of grid displacements as follows:

$$\epsilon(\hat{\theta}) = \frac{1}{N} \sum_{q \in \mathcal{G}} ||\omega_{\theta_{GT}}(q) - \omega_{\hat{\theta}}(q)||_2^2. \tag{3}$$

4.5 Basic baseline

To assess the difficulty of the task, we defined a very simple baseline based on the residual flow. We compute the following criterion to decide whether the frame t is dynamically salient:

$$\Phi(t) \geq \lambda \ , \ \text{with} \quad \Phi(t) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \left\| \omega(p,t) - \omega_{\hat{\theta}(t)}(p) \right\|_2 \tag{4}$$

where ω is the optical flow, $\omega_{\hat{\theta}}$ is the flow field corresponding to the dominant motion model of estimated parameters $\hat{\theta}$, and Ω is the image domain. Motion2D is used to estimate the dominant motion, and FlowNet2.0 to compute the optical flow.

$\Phi(t)$ characterizes the amount of residual motion in the frame t after cancelling the global motion. We threshold $\Phi(t)$ to classify the frame t as dynamically salient or not. To correctly set the threshold value λ, we compute the empirical distribution of $\Phi(t)$ on the frames of the validation set of the real video dataset. Then, an exponential law is fit to it. A p-value test with a probability of false alarm of 5% is applied to set the threshold λ.

4.6 Two-stream network as a baseline

Due to the lack of existing methods to compare with, we designed a second more challenging baseline, which is a direct application of the well-known two-stream network introduced in ([31]). The two-stream network involves two CNNs in parallel, one for the spatial stream, and one for the temporal stream. The two-stream network provided convincing results for the action recognition task. It was also exploited in other works, such as in [35] for dynamic texture synthesis

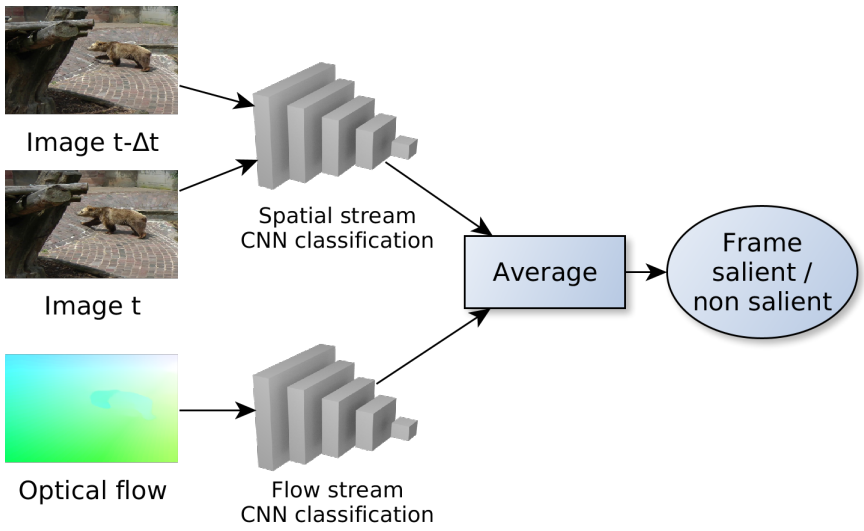


Figure 8: Two-stream network for motion saliency classification.

and for the prediction of eye-fixation maps in [1]. We can expect that it is a good candidate for temporal motion saliency detection as well. However, we need to adapt it for this new task. We have to produce a prediction for each frame of a video, and not one prediction for the whole video. Regarding the spatial stream, we concatenate the two considered frames of the video. Accordingly, as input of the temporal stream, we take the optical flow field computed between the two frames only. In this version of the two-stream network, we still use the classification CNN of Fig. 4 for both streams, and FlowNet2.0 to compute the optical flow. The resulting network is summarized in Fig. 8.

5 Experimental results

5.1 Experimental setting

We use the Caffe library ([12]) to implement the networks presented in Section 4. The optimization was achieved with the Adam method ([15]) with the parameters proposed by the authors. The runtime for the processing of a batch during the learning stage (prediction and back-propagation), with a GPU Tesla M40 and a 2.9 GHz processor is respectively of 1.4 sec, 1.8 sec, 0.7 sec and 1.2 sec for WS-DeepDOM, WS-Motion2D, RFS-DeepDOM and RFS-Motion2D. The batch size consists of 32, 32, 8 and 12 elements respectively. In the test stage, the prediction for one frame is performed in respectively 20.0, 15.2, 10.4 and 9.5 fps.

5.2 Comparison of the dominant motion estimation methods

The accuracy of the dominant motion estimation methods is likely to play an important role in the performance of the temporal motion saliency detection methods. We evaluated them on the synthetic dataset, which contains 2000 elements for which the ground truth is available by construction. With DeepDOM, we obtained a mean error for the estimation of the dominant motion of 0.20 pixels with a standard deviation of 0.08. With Motion2D, the mean error is of 0.03 pixels and the standard deviation is of 0.41. For the synthetic dataset, Motion2D provides an estimation of the motion closer to the ground truth. Yet, the accuracy obtained by DeepDOM remains reasonable.

5.3 Choice of the time step

An important aspect is the choice of the time step Δt between the two frames at test time. A time step of 1 means that we consider two successive frames of the video, a time step of 2, frames at $t-2$ and t, etc. We can expect that more distant frames will make the highlight of independently moving objects with small motion magnitude easier. On the other hand, the parametric estimation of the dominant motion is supposed to be more precise on temporally closer frames. Table 1 illustrates this behaviour for the RFS-Motion2D method. To find the best trade-off, time step was set to 1 during training, and the validation set with real videos is used to select the best time step for every method. Each method has thus the most appropriate time step for testing. It will also make the comparison fairer.

5.4 Experimental evaluation and comparison

5.4.1 Evaluation on the synthetic dataset

All the motion saliency detection methods were trained first on the synthetic database introduced in Section 3.1. The evaluation on the synthetic validation set showed that for all the methods, the accuracy was higher than 98%. Such a good performance can be explained by the fact that the synthetic dataset is ideal, as it displays strictly affine motions.

5.4.2 Evaluation on the real video dataset

Table 2 collects the comparative results. First of all, the simple baseline relying on criterion (4) based on the residual flow yields only a 54.4% accuracy, which shows that the problem is not trivial. The two-stream network has an accuracy of 80.9%, which is better, but leaves room for improvement. Let us note that the spatial and temporal streams, when used separately to make the prediction, do not provide equivalent results. The temporal stream performs better than the spatial stream by a large margin. This confirms that for the temporal motion saliency detection task, explicit motion information is the key input. In

contrast to the action classification task investigated in ([31]), no improvement was obtained by combining the two streams.

Table 2 shows that the best method is RFS-Motion2D, which is based on the residual flow. It reaches an overall accuracy of 87.5%. Globally, we draw the same conclusion than for the two-stream baseline, that is, for a given dominant motion estimation algorithm, methods which take flow as input perform better that methods that take images as input.

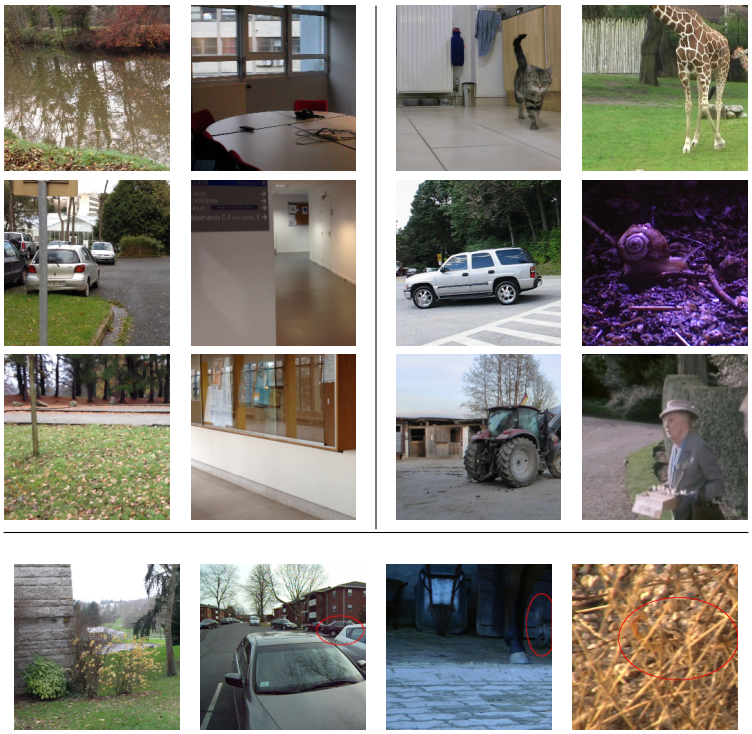


Figure 9: Classification examples for RFS-Motion2D. Top left rows and top right rows show six frames properly classified as respectively non dynamically salient and dynamically salient. The bottom row displays four failure cases. The first frame is wrongly classified as dynamically salient due to wind in the bushes; the red circle surrounds the salient moving object in the three next frames wrongly classified as non salient.

Samples of results are displayed in Fig. 9 for visual assessment. In the processed videos, the static scene is not always shallow, and moving objects may be not easily visible. However, the classification results remain convincing as illustrated in the top rows of Fig. 9. Difficult cases involving static objects in the foreground as trees or walls, dynamic textures as a flowing river, or camouflaged moving objects as the snail, are correctly classified. The bottom

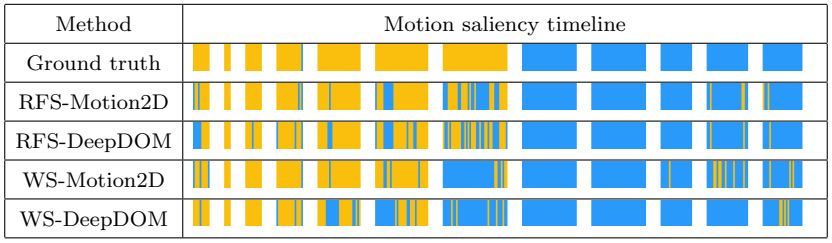


Figure 10: Comparative timelines of the frame classification on 12 real videos of different length (orange stands for dynamically salient, blue for non-salient).

row of Fig. 9 contains failure cases for frames involving wind in the bushes, or small moving objects (respectively car, dog, scorpion) partially hidden.

5.5 Impact of the quality of the dominant motion estimation

Using DeepDOM did not improve the results compared to the classical method Motion2D. This suggests that Motion2D still outperforms DeepDOM for real images. Methods WS and RFS are not equally affected by a lower accuracy of the dominant motion estimation method on real videos. Table 2 shows that the performance decrease is 2.9% for RFS and 8.7% for WS, when the training is performed on the synthetic dataset only. This suggests that RFS is more robust to a less precise motion estimation method. Intuitively, WS really needs a correct cancellation of the camera motion when comparing registered pixels at the same location to detect motion saliency. In contrast, optical flow contains information directly exploitable, and the camera motion compensation acts as a "denoising" step.

5.5.1 Choice of the parametric motion model

All the reported experiments involve an affine motion model. To evaluate how the chosen motion model impacts the classification performance, we ran WS-Motion2D with two different motion models: the affine one and the 8-parameter quadratic one. The latter consists of a polynomial of degree 2 for the two components of the velocity vector for a total of 8 free parameters. It accounts for the 2D projected motion of a 3D rigid motion of a planar scene, similarily to the homography for geometric transformation. The WS-Motion2D method has been chosen for this comparative evaluation, since it is more affected by the quality of the global motion estimation as mentioned above. The classification CNN is not trained again. Results reported at the bottom of Table 2 (referred as WS-Motion2D affine and quadratic) show that modifying the motion model at test time has almost no impact on the performance.

5.6 Impact of fine-tuning

By comparing results collected in Table 2 and Table 3, which reports the performance of the methods trained only on the synthetic dataset, we notice that further training on real videos always improves the performance. WS-Motion2D has already good performance with training on synthetic data only, but the other methods really benefit from the fine-tuning on real videos.

5.7 Temporal evaluation

The temporal behaviour of the four variants is illustrated in Fig. 10 with timeline plots. The 12 real video clips respectively depict moving camel, cars (twice), cat, tractor, giraffe and scorpion for the seven dynamically salient clips, and field, river, countryside, campus and indoors for the five non salient ones. Nine of them are represented in Fig. 9. All methods supply more stable results on non salient videos than on dynamically salient ones. The seventh clip in the row is a very difficult example (the scorpion one) of the Camouflaged Animals dataset. RFS methods are able to partly detect motion saliency in this video, whereas WS methods fail, demonstrating that motion information is the key clue and appearance may be useless.

Results obtained with RFS-Motion2D for a subset of the test set, and concatenated in a single video, are provided in the supplementary material. The

Time step Δt	1	2	3	4	5	6	7	8	9
Accuracy	84.9	89	88.3	88.9	**89.6**	88.9	88.9	89.2	88.5

Table 1: Rates of correct classification in percentage for RFS-Motion2D on the real validation set for several time steps Δt.

Method	Time step	Overall	Salient frames	Non salient frames
Baseline Φ	4	54.4	64.6	42.8
Two-stream network	1	80.9	64.3	99.7
Temporal stream only	2	83.2	71.4	96.7
Spatial stream only	3	72.6	50.4	98.0
WS-DeepDOM	2	76.5	59.4	96.2
RFS-Motion2D	5	**87.5**	79.7	96.4
RFS-DeepDOM	3	84.6	73.0	98.0
WS-Motion2D affine	6	<u>85.2</u>	78.4	93.0
WS-Motion2D quadratic	6	<u>85.2</u>	77.9	93.5

Table 2: Rates of correct classification on the test set with real videos. Best performance in bold, second best underlined.

Method	Overall	Salient frames	Non salient frames
WS-Motion2D	**80.9**	76.8	85.7
WS-DeepDOM	67.3	62.3	73.1
RFS-Motion2D	76.0	62.2	91.8
RFS-DeepDOM	69.7	44.1	99.0

Table 3: Rates of correct classification on the test set with real videos, where methods are trained only on the *synthetic* dataset.

colour of the frame border, resp. orange or blue, designates the prediction, resp. dynamic saliency or non saliency. The green (resp. red) square at the bottom right indicates that the prediction is correct (resp. wrong). The video shows that non saliency is correctly predicted even when the shallow scene assumption is not valid, with for instance walls, trees or pillars in the foreground. The video includes the camouflaged snail case which is successfully handled.

5.8 Additional experiments

We applied RFS-Motion2D to the DAVIS2016 dataset ([26]), which includes 50 videos with 3455 annotated frames. It was initially built for the video object segmentation task. The objects of interest are foreground moving objects. All the frames should be labeled as dynamically salient. We run RFS-Motion2D without any fine-tuning on the DAVIS2016 dataset. We obtained an overall correct classification rate of 93.3%, which is significantly better than the one obtained for our real dataset (79.7% for salient frames, see Table 2). Moving objects in our dataset usually exhibit a smaller motion magnitude than those of the DAVIS2016 dataset. Our dataset also includes the challenging Camouflaged Animals samples.

In addition, we conducted an experiment regarding the claim made at the end of Section 2. We applied the method of [38] on the non-salient videos, using the code made available by the authors. As expected, the method supplied non-empty motion saliency maps in every frame, sometimes involving large non-salient areas. To be fair enough, we added a decision step for an effective classification, and we got a correct classification rate of non-salient frames of 58%. In contrast, our method reached a rate of 98% (Table 2). This shows that existing methods are not able to handle the problem we solve.

6 Conclusion

We have formulated the problem of temporal motion saliency detection in videos. We proposed two methods involving camera motion compensation and CNN-based classification. They were favourably compared to a baseline exploiting the two-stream network. A synthetic dataset has been constructed and an

existing real dataset has been extended. The best method (RFS-Motion2D) reaches an overall accuracy of 87.5% on our real dataset, and even 93.3% on the DAVIS2016 dataset. It takes advantage of an explicit motion information given by the residual flow. Future work will try to improve the dominant motion estimation for complex scenes, and to investigate the introduction of temporal regularisation in the frame-based classification.

Acknowledgements

This work was supported in part by the DGA and the Région Bretagne through co-funding of Léo Maczyta's PhD thesis.

References

[1] C. Bak, A. Kocak, E. Erdem, and A. Erdem. Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, 20(7):1688–1698, July 2018.

[2] Pia Bideau and Erik G. Learned-Miller. A detailed rubric for motion segmentation. *CoRR*, abs/1610.10033, 2016.

[3] Pia Bideau and Erik G. Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *European Conference on Computer Vision (ECCV)*, 2016.

[4] Souad Chaabouni, Jenny Benois-Pineau, and Ofer Hadar. Prediction of visual saliency in video with deep CNNs. In *SPIE Optical Engineering+ Applications*, volume 9971 of *Applications of Digital Image Processing XXXIX*, August 2016.

[5] T. Crivelli, B. Cernuschi-Frias, P. Bouthemy, and J. Yao. Motion textures: Modeling, classification, and segmentation using mixed-state markov random fields. *SIAM Journal on Imaging Sciences*, 6(4):2484–2520, 2013.

[6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, Dec 2015.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[8] Y. Fang, Z. Wang, W. Lin, and Z. Fang. Video saliency incorporating spatiotemporal cues and uncertainty weighting. *IEEE Transactions on Image Processing*, 23(9):3910–3921, Sept 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[10] C. R. Huang, Y. J. Chang, Z. X. Yang, and Y. Y. Lin. Video saliency map detection by dominant camera motion removal. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(8):1336–1349, Aug 2014.

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, July 2017.

[12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pages 675–678, 2014.

[13] A. H. Karimi, M. J. Shafiee, C. Scharfenberger, I. BenDaya, S. Haider, N. Talukdar, D. A. Clausi, and A. Wong. Spatio-temporal saliency detection using abstracted fully-connected graphical models. In *IEEE International Conference on Image Processing (ICIP)*, pages 694–698, Sept 2016.

[14] W. Kim and C. Kim. Spatiotemporal saliency detection using textural contrast and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(4):646–659, April 2014.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 12 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th Int. Conf. on Neural Information Processing Systems (NIPS)*, NIPS'12, pages 1097–1105, 2012.

[17] T. Le and A. Sugimoto. Video salient object detection using spatiotemporal deep features. *IEEE Transactions on Image Processing*, 27(10):5002–5015, Oct 2018.

[18] Olivier Le Meur, Patrick Le Callet, and Dominique Barba. Predicting visual fixations on video based on low-level visual features. *Vision research*, 47(19):2483–2498, 2007.

[19] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G.M. Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41 – 50, 2018.

[20] Z. Liu, X. Zhang, S. Luo, and O. Le Meur. Superpixel-based spatiotemporal saliency detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(9):1522–1540, Sept 2014.

[21] D. Mahapatra, S. O. Gilani, and M. K. Saini. Coherency based spatio-temporal saliency detection for video object segmentation. *IEEE Journal of Selected Topics in Signal Processing*, 8(3):454–462, June 2014.

[22] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *Int. Journal of Computer Vision*, 126(9):942–960, Sep 2018.

[23] M. Narayana, A. Hanson, and E. Learned-Miller. Coherent motion segmentation in moving camera videos using optical flow orientations. In *2013 IEEE International Conference on Computer Vision*, pages 1577–1584, Dec 2013.

[24] 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.

[25] Jean-Marc Odobez and Patrick Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348 – 365, 1995.

[26] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

[27] 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.

[28] Wenliang Qiu, Xinbo Gao, and Bing Han. Eye fixation assisted video saliency detection via total variation-based pairwise interaction. *IEEE Transactions on Image Processing*, PP:1–1, 06 2018.

[29] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, July 2017.

[30] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. In *2016 International Conference on Learning Representations (ICLR)*, 2016.

[31] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576. 2014.

[32] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, Oct 2017.

[33] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[34] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[35] Matthew Tesfaldet, Marcus A. Brubaker, and Konstantinos G. Derpanis. Two-stream convolutional networks for dynamic texture synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 531–539, July 2017.

[37] W. Wang, J. Shen, and L. Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Transactions on Image Processing*, 24(11):4185–4196, Nov 2015.

[38] W. Wang, J. Shen, and L. Shao. Video salient object detection via fully convolutional networks. *IEEE Trans. on Image Processing*, 27(1):38–49, Jan 2018.

[39] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117 – 127, 2017.

[40] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4507–4515, Dec 2015.