Autoencoder-based background reconstruction and foreground segmentation with background noise estimation

Bruno Sauvalle   Arnaud de La Fortelle
Centre de Robotique, Mines ParisTech PSL University
{bruno.sauvalle,arnaud.de_la_fortelle}@mines-paristech.fr

Abstract

Even after decades of research, dynamic scene background reconstruction and foreground object segmentation are still considered as open problems due to various challenges such as illumination changes, camera movements, or background noise caused by air turbulence or moving trees. We propose in this paper to model the background of a frame sequence as a low dimensional manifold using an autoencoder and compare the reconstructed background provided by this autoencoder with the original image to compute the foreground/background segmentation masks. The main novelty of the proposed model is that the autoencoder is also trained to predict the background noise, which allows to compute for each frame a pixel-dependent threshold to perform the foreground segmentation. Although the proposed model does not use any temporal or motion information, it exceeds the state of the art for unsupervised background subtraction on the CDnet 2014 and LASIESTA datasets, with a significant improvement on videos where the camera is moving. It is also able to perform background reconstruction on some non-video image datasets.

1. Introduction

We consider in this paper the tasks of dynamic background reconstruction and foreground/background segmentation, which can be described in the following way: The input is a sequence $\mathcal{X}$ of consecutive frames $X_1, .., X_N$ showing a scene cluttered by various moving objects, such as cars or pedestrians, and the expected output is a sequence $\hat{\mathcal{X}} = \hat{X}_1, .., \hat{X}_N$ of frames showing the backgrounds of each scene without those objects.

The foreground/background segmentation task similarly takes as input the same kind of frames sequence $X_1, .., X_N$, but the expected output is a sequence $\mathcal{M}$ of foreground masks $M_1, .., M_N$ whose values at the pixel p are equal to zero if this pixel shows the background in the considered frame, and equal to 1 if the background is masked by a foreground moving object at this pixel (Fig. 1). This task is often called background subtraction because the pointwise multiplication of the mask M_k and the input image X_k gives an image showing only the foreground moving objects present in X_k, the input image background being replaced by a black background.

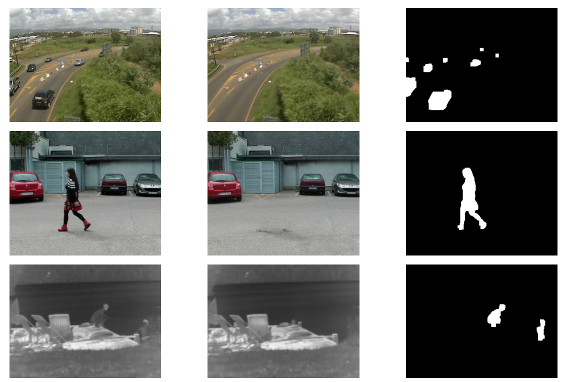

Figure 1. The proposed model takes as input a frame from the associated video (left column) and provides a reconstruction of the background (middle column) and a foreground mask (right column).

Background substraction is a fundamental tool in image analysis and has been studied for more than 30 years [67], but is still considered an open problem due to the various challenges appearing in real applications: illumination changes, high level of occlusion of the background, background motions caused by moving trees or water, challenging weather conditions, presence of shadows, etc. The applications of background subtraction are very diverse [19]: road, airport, store, maritime or military surveillance, observation of animals and insects, motion capture, human-computer interface, video matting, fire detection, etc.

The main application of background reconstruction is background subtraction, but other applications such as hole-filling in videos [39] have also been implemented. Efficient background reconstruction models are also necessary for unsupervised object detection and tracking [28, 23, 69].

The model presented in this paper starts from the classical assumption that the dynamic background of a scene can be modeled as a low dimensional manifold and uses an autoencoder to learn this manifold and perform dynamic background reconstruction. It then compares the input frame with the associated background predicted by the autoencoder to build the foreground segmentation mask. The main contributions of this paper are the following :

- We implement a more robust loss function to train the autoencoder, which gives a high weight to reconstruction errors associated to background pixels and a low weight to reconstruction errors associated to foreground pixels, and shows better performance than the L_1 loss usually considered for this task.
- We train the autoencoder to provide a background reconstruction, but also a background noise estimation, which gives a pixelwise estimate of the uncertainty of the background prediction. This noise estimation map is used to adjust the threshold necessary to compute the background/foreground segmentation mask.
- We reduce the risk of overfitting by developping a method for detecting significant background changes and implementing an early stopping criterion using this method if the video shows a fixed background.

The paper is structured as follows: We first review related work in section 2, then describe the proposed model in section 3. Experimental results are then provided in section 4.

2. Related work

Background subtraction methods can be split between supervised methods, which require labeled data, and unsupervised methods.

Supervised methods require labeled data as input, which are sets of pairs (X_k, M_k), where the image X_k is an image extracted from the sequence $X_1, .., X_N$ and the foreground mask M_k has to be provided by a human intervention. Supervised algorithms using linear methods such as as maximum margin criterion [34, 13] or graph signal reconstruction methods [20] have been proposed, but the current best performing supervised models use deep learning techniques with convolutional encoder-decoder structures [36, 35, 43], U-net structures [50, 44] or GANs [61, 73].

A spatio-temporal data augmentation strategy has been proposed [62] to improve generalization. One can also use as additional input to the deep learning model the output of an unsupervised background subtraction model [50, 49]. A background subtraction model can be substantially improved by combining its results with the output of a supervised semantic segmentation model [7, 71]. Although supervised models can reach very high accuracy results on a given video after labeling a significant number of frames of this video and training the model with these labeled data, their ability to generalize to new videos remain a major issue, and evaluations on unseen scenes lead to unfavorable results compared to unsupervised algorithms [43]. As a consequence, existing supervised models are not suited for real world applications where it is not possible to provide annotated data for each new input video.

One can classify **unsupervised methods** as statistical methods or reconstruction methods.

Statistical methods rely on a statistical modeling of the distribution of background pixel color values or other local features to predict whether a particular pixel is foreground or background. These statistical models can be parametric (univariate gaussian [67], mixture of gaussians [60], clusters [37], Student's t-distributions [46], Dirichlet process mixture models [6], Poisson mixture models [18], asymmetric generalized gaussian mixture models [15], etc.) or non parametric (pixel value histograms [72], kernel density estimation [14], codebooks [32], history of recently observed pixels [3, 24], etc.). The efficiency of these methods can be increased by using as input not only the pixel color values, but also features attached to superpixels [11] or local descriptors which are robust to illumination changes, such as SIFT [56], LBP or LBSP descriptors [58, 59]. If the camera is static, the segmentation of moving objects on a scene can also be performed by evaluating the motion associated to each pixel, using optical flow or flux tensor models. The blobs produced by these models are generally very fuzzy, but can be used as input to more complex models [8, 65].

Reconstruction methods use a background reconstruction model to predict the color (or other features) of the background at a particular pixel. The difference between the current image and the predicted background is then computed and followed by a thresholding to decide whether a pixel is background or foreground. Pixelwise reconstruction models try to predict the value of a background pixel at a particular frame from the sequence of values of the pixel of the last frames using a filter, which can be a Wiener filter [63], a Kalman filter [52] or a Chebychev filter [10]. A global prediction of the background can also be performed using the assumption that the background frames form a low dimensional manifold, which motivates the use of dimensionality reduction techniques such as principal component analysis (PCA) [48]. One can add to this approach a prior on the sparcity of the foreground objects by using a L_1 loss term applied to the foreground residuals, which leads to the development of models based on robust principal component analysis (RPCA) [68, 9]. More complex norms and additional regularizers have been proposed to improve the performance of this approach [42, 38, 70, 27, 26]. Non-linear dimensionality reduction using an autoencoder for background reconstruction has been proposed in [17, 51] and is further developed in the proposed model. Several unsupervised models can be also combined to form a more accurate model, such as the IUTIS-5 models, which is an ensemble model combining 5 different unsupervised models [5].

Background noise estimation Explicit background noise estimation for foreground segmentation has been introduced in [25]. Estimating the prediction uncertainty of a deep learning model is usually implemented using a negative log-likelihood loss function associated to a probabilistic model which includes a variance or concentration parameter [47, 31, 2, 45, 54].

3. Model description

The proposed model is a reconstruction model and has a general structure similar to the DeepPBM model [17]: We assume that the background frames form a low dimensional manifold and train an autoencoder to learn this manifold from the complete video. We however observe that the DeepPBM model described in [17] is not really unsupervised since it requires a significant engineering and optimization work for each new video, which is incompatible with any real-world application: The structure of the autoencoder and the number of latent variables have to be defined and fine-tuned on a scene by scene basis, which can be considered as a form of supervision. One also remarks that if the number of latent variables is too high, the autoencoder quiclky learns to reproduce the foreground objects, a phenomenon we call ovefitting, and fails to generate a proper background.

The model proposed in this paper is fully unsupervised: It uses a constant set of hyperparameter, and the structure of the autoencoder, which depends on the size of the image and on the complexity of the background, is defined automatically without human supervision.

3.1. Reconstruction loss using background bootstrapping

We implement a reconstruction loss using background bootstrapping, adapted from [53]. In the case of dynamic background reconstruction, this loss function allows to reduce the risk of overfitting to the foreground objects by giving a higher weight to background pixels than to foreground pixels during the optimization process. This loss is more robust to outliers than the L_1 loss which gives the same weight to small and large errors. The proposed reconstruction loss can be described by the following formulae [53]: We note $x_{n,c,i,j}$ the pixel color value of the image X_n for the channel c at the position (i,j) with $1 \leq c \leq 3$,$1 \leq i \leq h$ and $1 \leq j \leq w$, and $\hat{x}_{n,c,i,j}$ the pixel value of the reconstructed background $\hat{X}_n$ for the same channel and position. The local L_1 error associated to the pixel (i,j) is

$$l_{n,i,j} = \sum_{c=1}^{3} |\hat{x}_{n,c,i,j} - x_{n,c,i,j}|. \quad (1)$$

The soft foreground masks and spatially smoothed soft foreground masks are defined by the equations

$$m_{n,i,j} = \tanh\left(\frac{l_{n,i,j}}{\tau_1}\right) \quad (2)$$

and

$$\tilde{m}_{n,i,j}(\hat{X}_n, X_n) = \frac{1}{(2k+1)^2} \sum_{l=-k,p=-k}^{l=k,p=k} m_{n,i+l,j+p}, \quad (3)$$

where τ_1 and r are positive hyperparameters and $k = \lfloor w/r \rfloor$. The associated pixel-wise weight $w_{n,i,j}^{\text{bootstrap}}$ is then defined as

$$w_{n,i,j}^{\text{bootstrap}} = e^{-\beta \tilde{m}_{n,i,j}}, \quad (4)$$

where β is another positive hyperparameter. The reconstruction loss of the auto-encoder is then computed by weighting the pixelwise L_1 losses $l_{n,i,j}$ using these bootstrap weights:

$$\mathcal{L}_{\text{rec}}(\hat{\mathcal{X}}, \mathcal{X}) = \frac{1}{Nhw} \sum_{n=1,i=1,j=1}^{N,h,w} w_{n,i,j}^{\text{bootstrap}} l_{n,i,j} \quad (5)$$

The main differences between this loss function and the loss function defined in [53] is that it is a one-to-one loss, whereas the loss defined in [53] is one-to-many. It also does not use optical flow weights or abnormal image weights. Using optical flow weights would not allow to handle images taken from a moving camera, since it would give a low weight to all pixels associated to the moving background. We do not use abnormal image weights because we want the model to accurately reconstruct the background for each input image, which was not the case in [53], which is dedicated to fixed background reconstruction.

3.2. Optimized thresholding using background noise estimation

We remark that the bootstrap pixel weights $w_{n,i,j}^{\text{bootstrap}}$ can be used to get an estimate of the level of background noise of a frame sequence, considering that these weights are close to one when the associated pixel is a background pixel, and close to zero when this is not the case.

We therefore add a fourth output channel to the autoencoder, which is dedicated to give an estimate $\hat{l}_{n,i,j}$ of the value of the L_1 error $l_{n,i,j}$ for each pixel (i,j) for the frame X_n (Fig. 2).

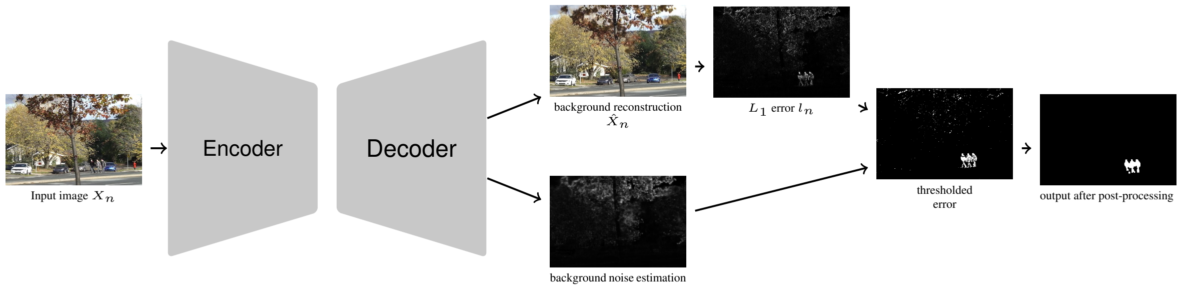

Figure 2. Schematic of the proposed model during inference (Error and noise images are normalized in the range [0,1])

The associated loss function is weighted using the bootstrap weights in order to limit its scope to background regions:

$$\mathcal{L}_{\text{noise}} = \frac{1}{3Nhw} \sum_{n=1 i=1, j=1}^{N,h,w} w_{n,i,j}^{\text{bootstrap}} |\hat{l}_{n,i,j} - l_{n,i,j}| \quad (6)$$

When the background is very noisy, the autoencoder is not able to predict accurately the value of a background pixel color. As a consequence, the expectation of $l_{n,i,j}$ is large, which leads to a high value of $\hat{l}_{n,i,j}$. One could consider that a more principled method would be to model the background noise as a gaussian distribution and estimate the variance of this distribution by learning the weighted average L_2 error instead of the L_1 error, but we have empirically found that such an approach is not robust to the presence of foreground objects.

The autoencoder is trained using the sum of the reconstruction loss and the loss associated to the background noise estimation. The complete loss function is then

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{noise}}. \quad (7)$$

The gradients of the weights $w_{n,i,j}^{\text{bootstrap}}$ are not computed during the optimization process [53]. We also do not use the gradient of $l_{n,i,j}$ in equation 6 because we do not want the quality of the background reconstruction be impacted by the background noise estimation optimization process.

In order to set the pixelwise threshold $\tau_{n,i,j}$ associated to the pixel (i, j) of the frame X_n and necessary to compute the background/foreground segmentation mask, we also take into account the average illumination $\hat{I}_n$ of the reconstructed background $\hat{X}_n$, as defined by the formula

$$\hat{I}_n = \frac{1}{3hw} \sum_{c=1, i=1, j=1}^{3,h,w} |\hat{x}_{n,c,i,j}|. \quad (8)$$

The threshold $\tau_{n,i,j}$ is then set according to the formula

$$\tau_{n,i,j} = \alpha_1 \hat{I}_n + \alpha_2 \hat{l}_{n,i,j}, \quad (9)$$

where α_1 and α_2 are two positive hyperparameters.The α_1 hyperparameter can then be interpreted as the threshold applicable to a scene showing a noiseless white background. The motivation of the second term is that if the background noise is high at some pixel, we have to increase the associated threshold for background/foreground segmentation in order to prevent the misclassification of background pixels as foreground caused by background noise.

For a given frame sequence $X_1, ..., X_n$ and a reconstructed background sequence $\hat{X}_1, ..., \hat{X}_n$, we then compute the foreground mask M_n before post-processing using the thresholding rule $M_{n,i,j} = 1$ if and only if $l_{n,i,j} > \tau_{n,i,j}$.

A post-processing is then applied in order to remove rain drops, snow flakes, and other spurious detections. It is composed of two morphological operations: a morphological closing using a 5×5 square structural element, followed by a morphological opening with a 7×7 square structural element.

3.3. Detecting significant background changes

The improved reconstruction loss function introduced in 3.1 reduces the risk of overfitting, but is not able to prevent it completely. We observe that the risk of overfitting increases when the number of optimization iterations and the number of parameters of the network increase. This is a significant issue because sequences showing background changes require a high number of training iterations and a model with a large number of parameters. In order to prevent overfitting, the number of training iterations and the complexity of the model are therefore adjusted to the complexity of the backgrounds sequence.

The main challenge here is to estimate without any human supervision whether the video shows substantial background changes or not. Such a task, which is very easy for a human, is far from trivial for a computer. For example, simply taking the variance of the various frames does not allow to estimate the complexity of the background changes because this variance will generally be dominated by foreground objects appearing in the video. More generally, it appears that in order to estimate the importance of the background changes, it is necessary to remove the foreground objects from the estimation process. We observe however that the proposed model can be used to perform this task. We then first train the model for a fixed small number N_{eval} of iterations, which is however sufficient to get a rough evaluation of the background changes. Using this trained model, we compute B_{eval} reconstructed backgrounds $\hat{X}_n$ using frames X_n sampled randomly from the sequence $\mathcal{X}$. Although these backgrounds estimates $\hat{X}_n$ are not accurate, we are confident that they do not show any foreground objects since a low number of iterations have been performed, so that the risk of overfitting is very low. We then compute the temporal median $\hat{X}$ of these backgrounds and compare this median background with the reconstructed backgrounds $\hat{X}_n$, computing soft masks $m_{n,i,j}$ following the same process as in formula 1 and 2. We then consider the average soft mask value over the B_{eval} reconstructed backgrounds

$$\bar{m} = \frac{1}{B_{\text{eval}} hw} \sum_{n,i,j}^{B_{\text{eval}},h,w} m_{n,i,j}. \tag{10}$$

If $\bar{m}$ is higher than a threshold τ_0, we consider that the background is a complex background. The partially trained model is discarded, a new autoencoder is created with more parameters and the number of training iterations is set to N_{complex} with a minimum of E_{complex} epochs for very long sequences. If this ratio is lower than τ_0, we consider that the background is a simple background, keep the partially trained model, and finish the training, with a total number of training iterations set to N_{simple}. The autoencoder structures for simple and complex backgrounds are described in the supplementary material.

4. Experimental results

4.1. Evaluation method

We consider the CDnet 2014, LASIESTA and BMC 2012 benchmark datasets for background subtraction. We use the public implementations of the algorithms PAWCS [59] and SuBSENSE [58] provided with the BGS library [57] to get baseline performance estimates for these methods when they are not available. We rely on published results for the other state of the art methods which do not provide public implementations.

We use the F-measure as main evaluation criteria. To compute the F-measure associated to a sequence of foreground masks predictions $M_1, .., M_n$, we first compute the sums TP, TN, FP, FN of the true positives, true negatives, false positives and false negatives associated to the sequence of masks $M_1, .., M_n$, and then compute the F-measure associated to this sequence as the harmonic mean of precision and recall.

We provide in Figure 3 some samples of background reconstruction, with the associated predicted foreground mask, and a comparison with foreground masks obtained using PAWCS and SuBSENSE. Other samples are provided in the supplementary material.

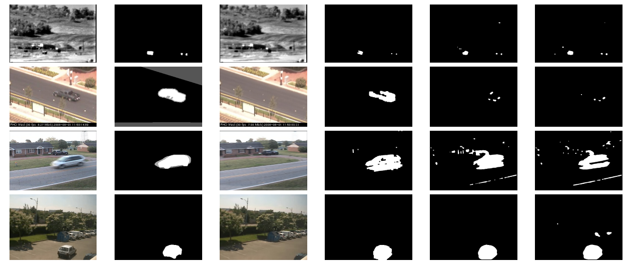

Figure 3. Examples of background reconstruction and foreground segmentation produced using the proposed model and comparison with PAWCS and SuBSENSE

4.2. CDnet 2014 dataset

The CDnet 2014 dataset [66] is composed of 53 videos, for a total of 153 278 frames, selected to cover the various challenges which have to be addressed for background subtraction: dynamic background (scenes with water or trees), camera jitter, intermittent object motion, presence of shadows, images captured by infrared cameras, challenging weather (snow, fog), images captured with a low frame rate, night images, images filmed by a pan-tilt-zoom camera, air turbulence. Ground truth foreground segmentation masks are provided for all frames of the dataset, with specific labels for shadow pixels which are not considered in the F-measure computation. We provide in Table 1 the F-measure results per category of the proposed model for each category of the CDnet 2014 dataset, with a comparison with the results obtained by other unsupervised models.

The proposed model gets a higher average F-measure on the CDnet 2014 dataset than all published unsupervised models, including ensemble models such as IUTIS-5, with an average F-measure of 0.784. One can observe a significant improvement in accuracy with the proposed model in the "pan-tilt-zoom" (PTZ) category with an average F-measure of 0.800 on this category. To our best knowledge, the proposed model is the first able to correctly handle videos taken from a moving camera.

4.3. LASIESTA dataset

The LASIESTA dataset [12] is composed of 48 videos grouped in 14 categories, for a total of 18 425 video frames. All frames are provided with ground truth pixel labels, with a specific label for pixels associated to stopped moving objects which are excluded from the F-measure computation. These videos are very short (The average number of frames per video is 383), which is challenging for the proposed deep-learning based model. We provide in Table 2 the average F-measure results of the proposed model for all 14 categories. Out of the 48 videos of the dataset, 4 videos are taken with a moving camera (categories IMC and OMC), and 24 videos include simulated camera motion (categories ISM and OSM). These 28 videos which include real or sim-

Table 1. Comparison of top unsupervised BGS algorithms according to the per-category F-measures on CDnet-2014

Method	Bad weather	Base-line	Camera jitter	Dynamic backgr.	Int. obj. motion	Low framerate	Night	PTZ	Shadow	Thermal	Turbu-lence	Overall
AE-NE (ours)	0.8337	0.8959	0.9230	0.6225	0.8231	0.6771	0.5172	0.8000	0.8947	0.7999	0.8382	**0.7841**
IUTIS-5 [5]	0.8248	0.9567	0.8332	0.8902	0.7296	0.7743	0.5290	0.4282	0.9084	0.8303	0.7836	0.7717
WisenetMD [33]	0.8616	0.9487	0.8228	0.8376	0.7264	0.6404	0.5701	0.3367	0.8984	0.8152	0.8304	0.7535
SuBSENSE [58]	0.8619	0.9503	0.8152	0.8177	0.6569	0.6445	0.5599	0.3476	0.8986	0.8171	0.7792	0.7408
PAWCS [59]	0.8152	0.9397	0.8137	0.8938	0.7764	0.6588	0.4152	0.4615	0.8913	0.8324	0.6450	0.7403
C-EFIC [1]	0.7867	0.9309	0.8248	0.5627	0.6229	0.6806	0.6677	0.6207	0.8778	0.8349	0.6275	0.7307
MSCL [27]	0.83	0.87	0.83	0.85	0.80	n/a	n/a	n/a	0.82	0.80	0.80	n/a
B-SSSR [26]	0.92	0.97	0.93	0.95	0.74	n/a	n/a	n/a	0.93	0.86	0.87	n/a

ulated camera motion are very difficult for existing background subtraction models and to our best knowledge, no paper has ever published category-wise evaluation results for these videos. In order to allow a comparison with these published results, we therefore also provide the average F-measure over the 10 categories showing only videos taken from a fixed camera. We observe that the proposed model performs better than available unsupervised algorithms on static scenes, and with a significant improvement on scenes where the camera is moving.

4.4. BMC 2012 dataset

The BMC dataset [64] contains 9 videos showing real scenes taken from static cameras and including the following challenges: shadows, snow, rain, presence of trees or big objects. Three of these sequences are very long (32 965, 117 149 and 107 815 frames). For fair comparison with other published results for this dataset, we provide the F-measure results for our model obtained using the the usual F-measure definition described in 4.1, but also the results obtained using the executable evaluation tool provided with the dataset which does not use the same definition of the F-measure [64]. We compute SuBSENSE and PAWCS results on this dataset and provide published evaluation results for other models in Table 3.

We observe that the proposed model gets again a better average F-measure than PAWCS and SuBSENSE on this dataset using the standard definition of the F-measure.

4.5. Non-video image datasets : Clevrtex, ObjectsRoom, ShapeStacks

The proposed model, which does not use any temporal information, can be adapted to perform background reconstruction and foreground segmentation on some image datasets which are not extracted from video sequences. We have tested this approach on three synthetic image datasets: Clevrtex [30], ShapeStacks, [21] and ObjectsRoom[29]. We use on ShapeStacks and ObjectsRoom the same preprocessing as in [16]. Although each image of these datasets shows a different background, the model is able to recognize that all the backgrounds appearing in a given dataset lie in a low dimensional manifold, which is the case because they have been generated using the same method. These datasets are provided with segmentation annotations for each object appearing in the scenes, which we converted to binary foreground segmentation masks in order to compute the F-measure of the predicted foreground masks.

Considering that on these datasets the risk of overfitting is very low and the background complexity is very high, we substantially increased the number of iterations, which is set to 500 000. We do not use morphological post-processing on the ShapeStacks and ObjectsRoom datasets, because these images have a very low resolution (64×64). We provide in Table 4 the average F-measure obtained on the test sets of these datasets after training on the associated training sets, and in Figure 4 some image samples. To our best knowledge, no other model is able to perform background reconstruction on these datasets.

4.6. Robustness to domain shift and fine-tuning

The proposed model is a batch model. In order to see whether it could be adapted for real-time applications, we studied whether a trained model could perform background reconstruction on new unseen images of the scene which do not belong exactly to the same distribution as the images used for training due to various possible domain shifts such as unseen illumination changes. We then have performed the following experiment: We have split each of the 53 videos provided in the CDnet dataset in two videos of equal lengths. The first half of each video is used to train the autoencoder, and the second half is used as a test dataset. The results of this experiment are provided in Table 5 and show stable results on three categories (baseline, bad weather, camera jitter) which do not show noticeable domain shifts, but a significant worsening on the other categories.

We then adopt the pretrain/fine-tune paradigm, consider the models trained on the first half of the videos as pretrained models, and study how many fine-tuning iterations using images randomly sampled from the second half of the videos are necessary to get competitive test results. We observe that the number of required iterations is very low compared to the number of iterations necessary for a full training, and conclude that a trained model is not robust to domain shifts, but can be quickly updated with a small number of fine-tuning iterations.

Table 2. Average per category of video F-measures on LASIESTA (sources : [12],[4], authors experiments for PAWCS and SuBSENSE)

	static camera										moving camera or simulated motion					
Method	ISI	ICA	IOC	IIL	IMB	IBS	OCL	ORA	OSN	OSU	IMC	ISM	OMC	OSM	Average. 10 categ.	Average. 14 categ.
AE-NE (ours)	0.91	0.88	0.91	0.81	0.92	0.79	0.94	0.80	0.82	0.91	0.83	0.79	0.86	0.89	**0.87**	**0.86**
PAWCS [59]	0.90	0.88	0.90	0.79	0.81	0.79	0.96	0.93	0.69	0.82	0.48	0.77	0.43	0.75	0.85	0.78
SuBSENSE [58]	0.90	0.89	0.95	0.65	0.77	0.73	0.92	0.90	0.81	0.79	0.33	0.70	0.31	0.65	0.83	0.73
Cuevas [4]	0.88	0.84	0.78	0.65	0.93	0.66	0.93	0.87	0.78	0.72	n/a	n/a	n/a	n/a	0.81	n/a
Haines [22]	0.89	0.89	0.92	0.85	0.84	0.68	0.83	0.89	0.17	0.86	n/a	n/a	n/a	n/a	0.78	n/a
Maddalena [41]	0.95	0.86	0.95	0.21	0.91	0.40	0.97	0.90	0.81	0.88	n/a	n/a	n/a	n/a	0.78	n/a
Maddalena [40]	0.87	0.85	0.91	0.61	0.76	0.42	0.88	0.84	0.58	0.80	n/a	n/a	n/a	n/a	0.75	n/a

Table 3. Comparison of top unsupervised BGS algorithms according to the video F-measure on BMC 2012

Method	Video 001	Video 002	Video 003	Video 004	Video 005	Video 006	Video 007	Video 008	Video 009	Average 9 videos
F-measure (standard definition)										
AE-NE (ours)	0.81	0.72	0.78	0.78	0.60	0.73	0.32	0.84	0.77	**0.71**
PAWCS [59]	0.70	0.58	0.85	0.72	0.27	0.79	0.58	0.74	0.80	0.67
SuBSENSE [58]	0.70	0.62	0.83	0.69	0.21	0.76	0.53	0.68	0.83	0.65
F-measure (using BMC evaluation tool)										
AE-NE (ours)	0.90	0.86	0.89	0.89	0.80	0.87	0.51	0.92	0.89	0.84
PAWCS [59]	0.86	0.77	0.93	0.86	0.66	0.89	0.79	0.87	0.90	0.84
SubSENSE [58]	0.85	0.80	0.92	0.85	0.68	0.87	0.75	0.84	0.91	0.83
DeepPBM [17]	0.73	0.86	0.94	0.90	0.71	0.81	0.70	0.76	0.69	0.78
G-LBM [51]	0.73	0.85	0.93	0.91	0.71	0.85	0.70	0.76	0.63	0.79
MSCL-FL [27]	0.84	0.84	0.88	0.90	0.83	0.80	0.78	0.85	0.94	0.86
B-SSSR [26]	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	**0.88**

Table 4. F-Measure on the Clevrtex, ShapeStacks and ObjectsRoom datasets

dataset	image size	number of frames training set	number of frames test set	average F-measure on test set
Clevrtex	128×128	40000	5000	0.78
ObjectsRoom	64×64	980000	20000	0.84
ShapeStacks	64×64	217888	46656	0.83

4.7. Implementation details

The proposed model is implemented using Python and the Pytorch framework. The associated code is available on the Github platform. Optimization is performed using the Adam optimizer with a learning rate of 5.10^{-4} and batch size equal to 32. The learning rate is divided by 10 when the number of optimization or fine-tuning iterations reaches 80% of the total number of iterations. The most important hyperparameters β, r and τ_1, which are associated to the loss function, are set to the values recommended in [53] i.e. $\beta = 6$, $r = 75$, $\tau_1 = 0.25$. The other hyperparameter values, which are related to the segmentation threshold and the detection and management of complex background changes, were found empirically using manual hyperparameter tuning. We then set $\alpha_1 = 96/255$, $\alpha_2 = 7$, $N_{\text{eval}} = 2000$, $B_{\text{eval}} = 480$, $\tau_0 = 0.24$, $N_{\text{simple}} = 2500$, $N_{\text{complex}} = 24000$, $E_{\text{complex}} = 20$.

For non-video dataset experiments, which take small images (64×64 and 128×128) as inputs, the batch size and learning rate are increased to 128 and 2.10^{-3} and the number of iterations N_{complex} is set to 500 000. The other hyperparameters remain the same. The autoencoder architecture is described in the supplementary material.

4.8. Computation time

We provide in Table 6 some computation time measurements, obtained using an AMD EPYC 7402 2,8 GHz CPU and a Nvidia RTX 3090 GPU. The inference and training times of the proposed model depend on the size of the image and the complexity of the background. The inference speed is between 50 frames per second and 240 frames per second. The time necessary to perform 100 training iterations is between 3,5 and 27 seconds.

4.9. Limitations

This model is not suited for night videos, considering the low score obtained on this category on the CDnet dataset. One also notes that although the model is able to handle correctly small objects staying still for a long time, as shown by the good results obtained the intermittent object category of the CDnet dataset, it suffers from overfitting when large foreground objects stay still (or appear to stay still) for a long time in a frame sequence. Out of the 110 tested videos contained in the datasets CDnet, LASIESTA and BMC, we observed this problem on 4 videos: ”office”, ”library” and ”canoe” in the CDnet dataset, and ”video007” in the BMC dataset (Fig. 5). The proposed model should then not be used when the video is expected to show large objects staying still for a long time. This model is a batch model and adapting it to real-time applications requires further work in order to reduce the latency caused by the fine-tuning iterations described in section 4.6.

4.10. Ablation study

In order to assess the impact of the various model features described in this paper, we have implemented sev-

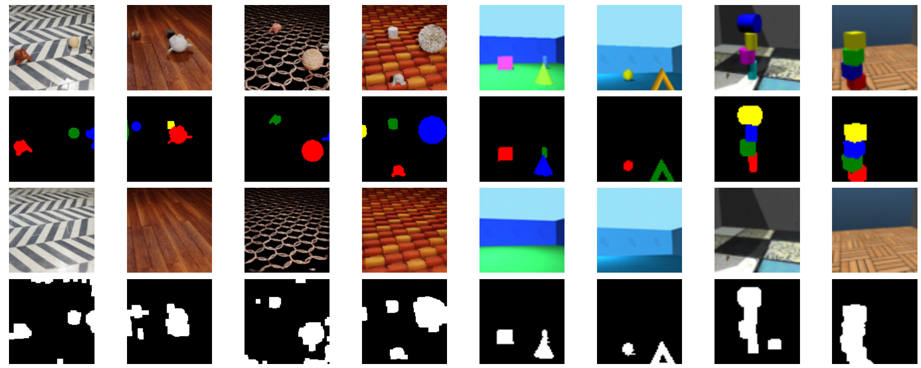

Figure 4. Examples of background reconstruction and foreground segmentations on the datasets Clevrtex (columns 1-4), ObjectsRoom (columns 5-6) and ShapeStacks (columns 7-8)

Table 5. F-measure results obtained on the CDnet dataset with a model pretrained using the first half of each video as training set, and fine-tuned on the last half using various numbers of fine-tuning iterations. Test results are for the last half of each video.

	Bad weather	Base-line	Camera jitter	Dynamic backgr.	Int. obj. motion	Low framerate	Night	PTZ	Shadow	Thermal	Turbu-lence	Overall	no pretraining
no fine-tuning	0.8114	0.8660	0.8768	0.3845	0.4199	0.5732	0.3998	0.2426	0.7371	0.5872	0.6447	0.5948	
100 iterations	0.8137	0.9063	0.9520	0.5846	0.5956	0.5891	0.4789	0.4723	0.9276	0.7639	0.6639	0.7044	0.4918
200 iterations	0.8078	0.9105	0.9543	0.6111	0.6536	0.5859	0.4977	0.4969	0.9316	0.7849	0.7523	0.7261	0.5658
400 iterations	0.8080	0.9125	0.9560	0.6309	0.7298	0.5842	0.5137	0.5465	0.9326	0.7880	0.8560	0.7507	0.6218
800 iterations	0.8104	0.8965	0.9577	0.6348	0.8212	0.5946	0.5420	0.6403	0.9293	0.7828	0.8763	0.7714	0.6934

Table 6. Computation time of the proposed model, PAWCS and SubSENSE for some sequences of the CDnet and BMC datasets

sequence name	highway	Video 009	blizzard	zoomin zooomout	continuous pan
image size	240x320	288x352	480x720	240x320	480x704
number of frames	1700	107817	7000	1130	1700
background complexity	simple	simple	simple	complex	complex
computation times (seconds)					
AE-NE (proposed model)					
- training	92	114	394	1443	7175
- backgrounds and masks generation	7	560	139	5	33
SuBSENSE	92	7161	1586	65	471
PAWCS	158	11290	2311	164	980

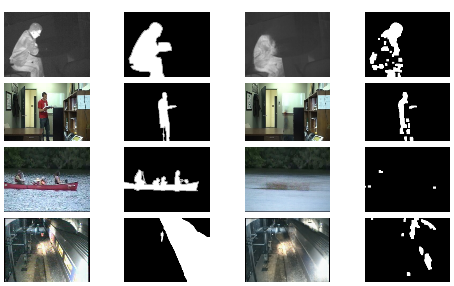

Figure 5. Failure cases due to overfitting on the datasets CDnet 2014 and BMC 2012: sequences "library", "office", "canoe" and "video007"

eral modifications of the proposed model and measured the average F-measure (FM) of these models on the CDnet2014 dataset. The results of these experiments are provided in Table 7. They show that the design of the loss function and the use of the background noise estimation layer have a substantial positive impact on the accuracy of the model. The improvement associated to post-processing is also significant, as already observed for other unsupervised

Table 7. Evaluation of various ablations of the proposed model

model description	average F-measure on the CDnet dataset	evolution vs reference model
proposed model (reference)	0.7841	
modified models :		
- no bootstrap weights ($w^{\text{bootstrap}}_{n,i,j}$ set to 1)	0.2771	-64,6 %
- inference without using the background noise estimation (α_2 set to 0)	0.6220	-20.7 %
- $w^{\text{bootstrap}}_{n,i,j}$ set to 1 and α_2 set to 0	0.4557	-41,9%
- training with L_2 reconstruction loss, α_2 set to 0	0,3384	-56,8 %
- inference without morphological post-processing	0.7170	-8.5%
- all backgrounds are considered as simple (τ_0 set to 1)	0,7397	-5,6 %
- using optical flow weights as in [53]	0,7701	-1,8%
- using abnormal image weights as in [53]	0,7690	-1,9%

background subtraction methods [55]. The model remains competitive on CDnet if the background complexity of all frames sequence is set to simple, an option which may be considered if training computation time is an issue.

5. Conclusion

We have proposed in this paper a new fully unsupervised dynamic background reconstruction and foreground segmentation model which does not use any temporal or motion information and is on average more accurate than available unsupervised models for background subtraction. The main strength of the proposed model is that it is able to perform background reconstruction on videos taken from a moving camera. Future works includes adapting the model for real-time applications, and using it to perform unsupervised object detection on real world scenes with complex backgrounds.

Acknowledgment We thank Sascha Hornauer for useful comments on the first draft of this paper.

References

[1] Gianni Allebosch, David Van Hamme, Francis Deboeverie, Peter Veelaert, and Wilfried Philips. C-EFIC: Color and Edge Based Foreground Background Segmentation with Interior Classification. pages 433–454, 2016.

[2] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and Exploiting the Aleatoric Uncertainty in Surface Normal Estimation. *Proceedings of the IEEE International Conference on Computer Vision*, pages 13117–13126, 2021.

[3] Olivier Barnich and Marc Van Droogenbroeck. ViBE: A powerful random technique to estimate the background in video sequences. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 945–948, 2009.

[4] Daniel Berjón, Carlos Cuevas, Francisco Morán, and Narciso García. Real-time nonparametric background subtraction with tracking-based foreground update. *Pattern Recognition*, 74:156–170, 2018.

[5] Simone Bianco, Gianluigi Ciocca, and Raimondo Schettini. How far can you get by combining change detection algorithms? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10484 LNCS:96–107, 2017.

[6] Himani K Borse and Bharati Patil. Background Subtraction with Dirichlet process Gaussian Mixture Model (DP-GMM) for Motion Detection. 3(7):70–75, 2015.

[7] M Braham, S Pierard, and M Van Droogenbroeck. Semantic background subtraction. *Ieee*, pages 4552–4556, 2017.

[8] Filiz Bunyak, Kannappan Palaniappan, Sumit Kumar Nath, and Gunasekaran Seetharaman. Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking. *Journal of Multimedia*, 2(4):20–33, 2007.

[9] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

[10] Remy Chang, Tarak Gandhi, and Mohan M. Trivedi. Vision modules for a multi-sensory bridge monitoring approach. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 971–976, 2004.

[11] Andrew Tzer Yeu Chen, Morteza Biglari-Abhari, and Kevin I.Kai Wang. SuperBE: computationally light background estimation with superpixels. *Journal of Real-Time Image Processing*, 16(6):2319–2335, 2019.

[12] Carlos Cuevas, Eva María Yáñez, and Narciso García. Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. *Computer Vision and Image Understanding*, 152:103–117, 2016.

[13] Farcas Diana and Thierry Bouwmans. Background modeling via a supervised subspace learning To cite this version : University of La Rochelle. 2010.

[14] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1843:751–767, 2000.

[15] Tarek Elguebaly and Nizar Bouguila. Finite asymmetric generalized Gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 117(12):1659–1671, 2013.

[16] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement. 2021.

[17] Amirreza Farnoosh, Behnaz Rezaei, and Sarah Ostadabbas. Deeppbm: Deep probabilistic background model estimation from video sequences. *arXiv*, 2019.

[18] Alberto Faro, Daniela Giordano, and Concetto Spampinato. Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1398–1412, 2011.

[19] Belmar Garcia-Garcia, Thierry Bouwmans, and Alberto Jorge Rosales Silva. Background subtraction in real applications: Challenges, current models and future directions. *Computer Science Review*, 35:100204, 2020.

[20] Jhony H. Giraldo and Thierry Bouwmans. GraphBGS: Background subtraction via recovery of graph signals. *Proceedings - International Conference on Pattern Recognition*, pages 6881–6888, 2020.

[21] Oliver Groth, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi. ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11205 LNCS:724–739, 2018.

[22] Tom S.F. Haines and Tao Xiang. Background subtraction with dirichletprocess mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):670–683, 2014.

[23] Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposi-

tion in 3D. *Advances in Neural Information Processing Systems*, 2020-Decem(Section 3), 2020.

[24] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–43, 2012.

[25] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–43, 2012.

[26] Sajid Javed, Arif Mahmood, Somaya Al-Maadeed, Thierry Bouwmans, and Soon Ki Jung. Moving Object Detection in Complex Scene Using Spatiotemporal Structured-Sparse RPCA. *IEEE Transactions on Image Processing*, 28(2):1007–1022, 2019.

[27] Sajid Javed, Arif Mahmood, Thierry Bouwmans, and Soon Ki Jung. Background-Foreground Modeling Based on Spatiotemporal Sparse Subspace Clustering. *IEEE Transactions on Image Processing*, 26(12):5840–5854, 2017.

[28] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. SCALOR: Generative World Models with Scalable Object Representations. 2019.

[29] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-Object Datasets. https://github.com/deepmind/multi-object-datasets/, 2019.

[30] Laurynas Karazija, Iro Laina, and Christian Rupprecht. ClevrTex: A Texture-Rich Benchmark for Unsupervised Multi-Object Segmentation. (NeurIPS), 2021.

[31] Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5575–5585, 2017.

[32] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. *Proceedings - International Conference on Image Processing, ICIP*, 2:3061–3064, 2004.

[33] Sang Ha Lee, Gyu Cheol Lee, Jisang Yoo, and Soonchul Kwon. WisenetMD: Motion detection using dynamic background region analysis. *Symmetry*, 11(5), 2019.

[34] Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. *Advances in Neural Information Processing Systems*, 17(1):157–165, 2004.

[35] Long Ang Lim and Hacer Yalim Keles. Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications*, 23(3):1369–1380, 2020.

[36] Long Ang Lim and Hacer Yalim Keles. Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters*, 112:256–262, 2018.

[37] Citable Link and Darren E Butler. Real-Time Adaptive Foreground / Background Segmentation. 2005(14), 2018.

[38] Xin Liu, Guoying Zhao, Jiawen Yao, and Chun Qi. Background subtraction based on low-rank and structured sparse decomposition. *IEEE Transactions on Image Processing*, 24(8):2502–2514, 2015.

[39] Guibo Luo, Yuesheng Zhu, Zhaotian Li, and Liming Zhang. A Hole Filling Approach Based on Background Reconstruction for View Synthesis in 3D Video. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:1781–1789, 2016.

[40] Lucia Maddalena and Alfredo Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[41] Lucia Maddalena and Alfredo Petrosino. The SOBS algorithm: What are the limits? *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–26, 2012.

[42] Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Network flow algorithms for structured sparsity. *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, pages 1–9, 2010.

[43] Murari Mandal and Santosh Kumar Vipparthi. Scene Independency Matters: An Empirical Study of Scene Dependent and Scene Independent Evaluation for CNN-Based Change Detection. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14, 2020.

[44] V. Mondéjar-Guerra, J. Rouco, J. Novo, and M. Ortega. An end-to-end deep learning approach for simultaneous background modeling and subtraction. *30th British Machine Vision Conference 2019, BMVC 2019*, pages 1–12, 2020.

[45] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud De La Fortelle. CoordiNet: Uncertainty-aware pose regressor for reliable vehicle localization. *Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, pages 1848–1857, 2022.

[46] Dibyendu Mukherjee and Q. M.Jonathan Wu. Real-time video segmentation using student's t mixture model. *Procedia Computer Science*, 10:153–160, 2012.

[47] David A. Nix and Andreas S. Weigend. Estimating the mean and variance of the target probability distribution. *IEEE International Conference on Neural Networks - Conference Proceedings*, 1:55–60, 1994.

[48] Nuria M Oliver, Barbara Rosario, Alex P Pentland, and Senior Member. A Bayesian Computer Vision System for Modeling Human Interactions. 22(8):831–843, 2000.

[49] Montse Pardàs and Gemma Canet. Refinement network for unsupervised on the scene foreground segmentation. *European Signal Processing Conference*, 2021-Janua:705–709, 2021.

[50] Gani Rahmon, Filiz Bunyak, Guna Seetharaman, and Kannappan Palaniappan. Motion U-Net: Multi-cue encoder-decoder network for motion segmentation. *Proceedings - International Conference on Pattern Recognition*, pages 8125–8132, 2020.

[51] Behnaz Rezaei, Amirreza Farnoosh, and Sarah Ostadabbas. G-LBM: Generative Low-Dimensional Background Model Estimation from Video Sequences. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12357 LNCS:293–310, 2020.

[52] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive Background Estimation and Foreground Detection using Kalman Filtering. *Proceedings of the International Conference on Recent Advances in Mechatronics (ICRAM 1995)*, pages 193–199, 1995.

[53] Bruno Sauvalle and Arnaud de La Fortelle. Fast and Accurate Background Reconstruction Using Background Bootstrapping. *Journal of Imaging*, 8(1), 2022.

[54] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. pages 1–24, 2022.

[55] Ajmal Shahbaz, Joko Hariyono, and Kang Hyun Jo. Evaluation of background subtraction algorithms for video surveillance. *2015 Frontiers of Computer Vision, FCV 2015*, 2015.

[56] Arnold W M Smeulders. Efficient projection pursuit density estimation for background subtraction 3 . Projection pursuit density estima- tion. 0(3).

[57] Andrews Sobral. BGSLibrary: An OpenCV C++ Background Subtraction Library. *IX Workshop de Visao Computacional (WVC'2013)*, (JUNE 2013):1–3, 2013.

[58] Pierre Luc St-Charles, Guillaume Alexandre Bilodeau, and Robert Bergevin. SuBSENSE: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1):359–373, 2015.

[59] Pierre Luc St-Charles, Guillaume Alexandre Bilodeau, and Robert Bergevin. Universal Background Subtraction Using Word Consensus Models. *IEEE Transactions on Image Processing*, 25(10):4768–4781, 2016.

[60] Chris Stauffer and W. E.L. Grimson. Adaptive background mixture models for real-time tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:246–252, 1999.

[61] Maryam Sultana, Arif Mahmood, Sajid Javed, and Soon Ki Jung. Unsupervised deep context prediction for background estimation and foreground segmentation. *Machine Vision and Applications*, 30(3):375–395, 2019.

[62] M. Ozan Tezcan, Prakash Ishwar, and Janusz Konrad. BSUV-Net 2.0: Spatio-Temporal Data Augmentations for Video-Agnostic Supervised Background Subtraction. *IEEE Access*, 9:53849–53860, 2021.

[63] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. *Proceedings of the IEEE International Conference on Computer Vision*, 1:255–261, 1999.

[64] Antoine Vacavant, Thierry Chateau, Alexis Wilhelm, and Laurent Lequièvre. A benchmark dataset for outdoor foreground/background extraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7728 LNCS(PART 1):291–300, 2013.

[65] Rui Wang, Filiz Bunyak, Guna Seetharaman, and Kannappan Palaniappan. Static and moving object detection using flux tensor with split gaussian models - Wang et al. - 2014 - IEEE Computer Society Conference.pdf. *IEEE Change Detection Workshop, CVPR*, pages 414–418, 2014.

[66] Yi Wang, Pierre Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. CDnet 2014: An expanded change detection benchmark dataset. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 393–400, 2014.

[67] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785, 1997.

[68] John Wright, Yigang Peng, Yi Ma, Arvind Ganesh, and Shankar Rao. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, pages 2080–2088, 2009.

[69] Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner. APEX: Unsupervised, Object-Centric Scene Segmentation and Tracking for Robot Manipulation. 2021.

[70] Bo Xin, Yuan Tian, Yizhou Wang, and Wen Gao. Background Subtraction via generalized fused lasso foreground modeling. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:4676–4684, 2015.

[71] Dongdong Zeng, Xiang Chen, Ming Zhu, Michael Goesele, and Arjan Kuijper. Background Subtraction with Real-Time Semantic Segmentation. *IEEE Access*, 7:153869–153884, 2019.

[72] Shengping Zhang, Hongxun Yao, Shaohui Liu, Shengping Zhang, Hongxun Yao, Shaohui Liu, Dynamic Background, Subtraction Based, Shengping Zhang, Hongxun Yao, and Shaohui Liu. Dynamic Background Subtraction Based on Local Dependency Histogram To cite this version. 2008.

[73] Wenbo Zheng, Kunfeng Wang, and Fei Yue Wang. A novel background subtraction algorithm based on parallel vision and Bayesian GANs. *Neurocomputing*, 394:178–200, 2020.