Spatio-Temporal Attention Model for Foreground Detection in Cross-Scene Surveillance Videos

Dong Liang¹,*, Jiaxing Pan¹, Han Sun¹ and Huiyu Zhou²

¹ College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
² Department of Informatics, University of Leicester, LE1 7RH, United Kingdom; hz143@leicester.ac.uk
* Correspondence: liangdong@nuaa.edu.cn

Abstract: Foreground detection is an important theme in video surveillance. Conventional background modelling approaches build sophisticated statistical model based on low-level features to cover the diversity of dynamic background and increasingly update the background model. While modern semantic or instance segmentation approaches could provide high-level semantic annotation for each frame, but ignore the temporal relevance. In this paper, we propose a Spatio-Temporal Attention Model (STAM) for cross-scene foreground detection. Appearance and motion features, low-level and high-level features are synthesised to weaken the semantic gap by attention modules via guiding the feature learning procedure. First, the static frame and its optical flow feed two independent encoders. Then high-level features guide attention modules to re-weight low-level features to reconstruct the foreground in pixel-level. For training, we employed randomly 5% video frames and their ground truth from CDnet 2014 [53]. Experimental results on three challenging benchmarks (CDnet 2014, Wallflower and PETS) validate it outperforms many state of the art methods under various situations with respect to the average ranking over different evaluation metrics, especially, under the cases of bad weather, intermittent object motion, shadow, thermal, turbulence and light switch. STAM containing attention mechanism and temporal information exceeds models excluded these parts, with F-measure 9% and 6% increasing respectively. It improves the overall Precision 9% and F-measure 4% than Cascade CNN. The processing speed is 10.8 fps for the frame size 256 by 256.

Keywords: foreground detection; attention model; optical flow; background modeling

0. Introduction

Detecting foreground plays a very important role in an intelligent surveillance system. It is often integrated with various tasks, such as tracking objects, recognizing their behaviors and alerting when abnormal events occur. However, object detection suffers from non-stationary scenes in surveillance videos, especially in two potentially serious cases: (1) illumination variation, such as outdoor sunlight changes and indoor lights turning on/off; (2) physical motion, such as ripples on the water surface, atmospheric disturbance, trees swaying and the motion of indoor artificial objects, which include fans, escalators and auto-doors. If the actual background contains a combination of factors mentioned above, it becomes even more difficult to perform detection.

In order to be free from illumination changes and dynamic backgrounds, early studies focused on statistical distributions to build the background model, from the earliest single Gaussian model for each pixel, to the multi-modal background representation Gaussian mixture model (GMM), and the non-parametric models [1–4]. To cover the variation of illumination change, the background model occupies a large range of intensity, so that the detection would be insensitive. Local features can represent the spatial characters [5–10] but cannot adapt to many non-ideal cases, such as texture-less background. In addition, conventional algorithms handle gradual illumination changes by updating the statistical background models progressively as time goes by. In practice, this kind of model update is usually relatively slow to avoid mistakenly integrating foreground elements into the background model, making it difficult to adapt to sudden illumination changes and burst motion. Modern
semantic or instance segmentation approaches could provide high-level semantic annotation for each frame, but ignore the temporal relevance. On the other hand, the obstacle to introduce more sophisticated learning technique is that foreground detection is a scene-dependent and pixel-wise processing procedure [11] which requires a relatively lightweight training and detection model to reduce the resources occupancy.

Essentially, foreground detection in video is a empirical semantic segmentation problem closely related to appearance and motion, but does not have clear definition, which cannot be well solved by both low-level background modeling or high-level frame-based semantic segmentation. Semantic segmentation of appearance focuses on the screening of common foreground regions such as pedestrians and vehicles in the scene but lacks motion cue. Significant motion region are likely to be foreground targets, but regions of repetitive motions in time and space are often not foreground targets.

To bridge the semantic gap, in this paper, we propose a Spatio-Temporal Attention Model (STAM) to combine the spatial and temporal information for cross-scene foreground detection, in which the feature engineering and parameter tuning become unnecessary when handling various video scenes. The static frame and its optical flow feed two independent encoders, then high-level features guide the attention modules to re-weight low-level features to reconstruct the foreground in pixel-level.

The highlights of this work are:

1. The proposed attention modules utilize the high-level features to guide the selection of low-level features in order to preserve the objects’ boundaries and local details. While the conventional Encoder-decoder connects the low-level and high-level features blindly without any distinction. Thus our model can be seen as a "attention-guided weight-able connection encoder-decoder", to preserve the effective connections and suppress the invalid connection.

2. Cross-Scene foreground segmentation. We use all the scenes in CDnet2014 [53] to train only one model with only 5% ground-truth samples for training, and all the testing results on different scenes are given by this model, which doesn’t need to retrain for other datasets.

3. The introduction of motion cue to combine the spatial and temporal information for foreground detection. The static frame and its optical flow feed two independent encoders, then high-level features guide the attention modules to re-weight low-level features to reconstruct the foreground in pixel-level. Compared to the state-of-the-art model without motion cue, our model gets significant improvements.

1. Related work

Conventional background subtraction

Since observations of the background in image sequences can be considered as stochastic events, many statistical approaches have been employed to model effective backgrounds. The former background modeling approaches can be classified into two categories: (1) Independent pixel-wise modeling, which employs the statistical processing of time-domain observations to each pixel. (2) Spatial-dependence modeling, which employs principles to exploit spatial-dependence among pixels to build a local or global model.

Most of the earlier background modeling approaches tend to fall into the first category. Wren [12] modeled the observations (YUV) of each pixel as a single Gaussian probability density function. To cope with periodic moving background patterns, the Gaussian mixture model (GMM) [13,14] was proposed. Elgammal [2] employed kernel density estimation (KDE) as a data-driven modeling method. Since KDE is a non-parametric model, it is closer to the real probability distribution than GMM. Hidden Markov models (HMMs) [15,16] have also been applied to model the background; topology free HMMs were described and several state splitting criteria were compared in the context of background modeling in [15], and a non-adaptive three-state HMM was used to model the background in [16]. [3] presented a real-time algorithm, which sampled background pixel values and quantized them into compressed codebooks (CBs). To improve the processing efficiency of the codebooks, Guo [17] presented a hierarchical scheme. All the above methods use a learning rate function for updating the background model online. They have a well-known trade-off problem: with a low learning rate, they can not adapt to sudden changes of illumination, e.g., turning on/off a light, while with a high learning rate, slowly moving objects or temporarily stopped objects will be detected as background.
The second category uses spatial information to exploit the spatial dependencies of pixels in the background. Matsuyama [18] proposed a regional block matching method against varying illumination, and Seki [19] proposed a co-occurrence based block correlation method. The above two methods can only yield coarse region-level detection. Toyama et al. [20] proposed a three-layer algorithm in which Weiner filters were employed. It used region and frame-level information to verify the pixel-wise background model. Oliver [21] employed eigenspace decomposition in which the background was modelled by the eigenvectors corresponding to the largest eigenvalues. Sheikh [8] used the joint representation of image pixels in a local spatial distribution (proximal pixels) and colour information to build both background and foreground KDE models competitively in a decision framework. Monnet [22] and Zhong [23] built an auto-regressive moving average (ARMA) model in dynamic scenes, which is used to incrementally learn (using PCA) and then predict motion patterns in the scene. Heikkilä and Pietikäinen [5] used a local binary pattern (LBP) to subtract the background and detect moving objects in real time. This method models each pixel as a group of adaptive LBP histograms that were calculated over a predefined circular region around the pixel. Similarly, the statistical reach feature (SRF) [24] builds a local texture model for each target pixel to be brightness-invariant. [25] modeled appearance changes by incrementally learning a tensor subspace representation by adaptively updating the sample mean and an eigenbasis for each unfolding matrix. In our previous research, we pay our attention on Co-occurrence pixel-pair background models [9,26–29]. The models employed an alignment of supporting pixels for the target pixel which held a stable intensity subtraction in training frames without any restriction of locations. The intensity subtraction of the pixel pairs allowed the background model to tolerate noise and be illumination-invariant.

### CNN based foreground detection

Modern general instance segmentation approaches based on deep convolutional networks have great potential in this task. A surveillance video can be split into frames and then segmented as foreground and background frame by frame. This kind of approaches could be roughly divided into two families. One family relies on the R-CNN proposals, which is a bottom-up pipeline that the segmentation results are based on the proposals and then labelled by a classifier [30–35]. The other family relies on semantic segmentation results [36–39] where instance segmentation following semantic segmentation by classifying pixels into different instances. A state-of-the-art method Mask-RCNN [40], built upon object detectors [41,42], also depends on the proposals but features are shared by classes, box predictors and mask generators, then all results are collected in parallel. All of those algorithms contain complex multiple-stage cascading and hungry on huge training data.

The first approach for background subtraction using CNN was proposed by Brahamand Droogenbroeck [43]. They used a fixed background model, which was generated from a temporal median operation over \( N \) video frames. Afterwards, a scene-specific CNN is trained with corresponding image patches from the background image, video frames and ground truth pixels, or alternatively, segmentation results coming from other background subtraction algorithms. After extracting a patch around a pixel, feeding the patch through the network and comparing it with the score threshold, the pixel is assigned with either a background or a foreground label. However, due to the over-fitting caused by using highly redundant data for training, the network is scene-specific, i.e. can only process a certain scenery, and needs to be retrained for other video scenes (with scene-specific data). Another approach is DeepBS [44], which utilizes a trained CNN and a spatial-median filter to realize foreground detection in various video scenes. This approach is fast, but as the foreground is detected based on independent frame, the temporal relevance of the neighbouring frames is ignored. Cascade CNN [45] proposed a semi-automatic method who releases pressure on a mount of training data. CNN branches processing images in different size are cascaded together that helps the cascade CNN to detect foreground objects in multi-scale. Similarly, temporal information dose not been took into consideration in this model.

### Attention model

Evidence from human perception process [46] shows the importance of attention mechanism, which uses top information to guide bottom-up feed-forward process. The attention mechanism of the human brain is, at a particular moment, human attention is always focused on a part of the scene, while ignoring the other parts. The attention mechanism of human brain could be equivalent to a resource allocation model. Tentative efforts have been made towards applying attention into deep neural network. Deep Boltzmann Machine (DBM) [47] contains top-down attention by its reconstruction process in the training stage. Attention mechanism has also been
Figure 1. The framework of the proposed foreground detection model, STAM

2. The proposed approach

Optical flow indicates the moving information of the scene and local objects where the moving areas have high probability to carry foreground objects in many real-world applications. So it provides a prior knowledge on where should be focused on in a static image. High-level features have larger reception field, contain global context and they are good at scene classification but weak in predicting labels for every pixel in input resolution [50]. While low-level features carry much fine grained information which can help high-level features to reconstruct objects’ details during up-sampling process. U-net is an efficient structure to combine these features [51,52], it propagates information from the down-sampling layers to all corresponding symmetrical up-sampling layers. However, U-net concatenates the encoder and decoder features without any selecting, so it cannot determine whether the features chosen is necessary for foreground segmenting or not. The design of the proposed attention structure in our STAM network is inspired by recent development of a semantic segmentation model [50] who employs the high-level features to re-weight the fine-grained features in channel-wise. The proposed model merges the decoder and encoder features through a serious attention processes during decoder phase. In detail, high-level features provide global information to guide attention modules to select (weight) proper low-level features who make contribution to binary prediction in input image that encoder features are re-weighted by the decoder layers in pixel-level and concatenated with the later.

2.1. Model structure

As illustrated in figure 1, the model combines the spatial and temporal information, and attention module is employed to mix encoder features together with decoder ones. The blocks in green represent the encoder layers and “IConv” and “OConv” are two encoders fed with static image and optical flow, respectively. The blocks in pink and orange represent the decoder layers and attention modules. The plus sign in green means the addition in pixel-level while the plus sign in red represents the concatenate operation. For example, there are
two feature maps with dimension \( m \times m \times n \), and there is an \( m \times m \times (2 \times n) \) tensor got through addition and an \( m \times m \times (2 \times n) \) tensor outputted by the later operation.

Table 1 shows details of each layer in STAM. It is fed with a \( 256 \times 256 \times 3 \) static image and a \( 256 \times 256 \times 1 \) optical flow then outputs a \( 256 \times 256 \times 1 \) foreground mask. “IConv” and “OConv” are two encoders with the same structure and 8 convolution layers. Also, the decoder has 8 layers and up-sampling processed in each layer and 7 attention modules are applied to make features mixtures. The stride for every convolution is 2 in both encoder and decoder but 1 in attention module. Dropout is utilized to avoid over-fitting in the first three layers of decoder and nodes in these layers with 50% probability to be dropped in training phase.

![Table 1. Filter size and output size of each layer in encoders, decoders and attention modules.](image)

### 2.2. The proposed attention module

The design of the proposed attention structure is inspired by a semantic segmentation model [50], it employs the high-level features to re-weight the fine-grained features in channel-wise. Different from [50], our model merges the decoder and encoder features through a serious attention processes during decoder phase. In detail, high-level features provide global information to guide attention modules to weight proper low-level features who make contribution to binary prediction in input image that encoder features are re-weighted by the decoder layers in pixel-level and concatenated with the later. As it is shown in Figure 2, our attention modules merge the high-level and low-level features guided by the former ones. Y1 and Y2 are features from image encoder and optical flow encoder, and X is the decoder feature respectively. Where \( H, W \) and C are the height, width and channel numbers of a feature map. We apply a single convolution operation \( conv() \) onto \( X \) followed by a sigmoid activation function \( \sigma \) that makes the weights belong to 0 to 1. Where \( b \) is the bias value of a convolution operator.
Then we use those weights $f_{\text{weights}}$ to reweight the sum of encoder features. Finally, the decoder feature $X$ and the reweighted features are concatenated $f_{\text{output}}$ as the input of next convolutional layer.

\begin{align}
    f_{\text{weights}} &= \sigma(\text{conv}(X) + b) \\
    f_{\text{output}} &= \text{concat}(f_{\text{weights}} \otimes (Y_1 \oplus Y_2), X)
\end{align}

Where $\otimes$ and $\oplus$ denote the pixel-wise multiplication and sum operation, and $\text{concat}(\cdot)$ is a concatenate process on two features.

### 2.3. Loss function

STAM is fed with static image $x_{\text{img}}$ and its optical flow image $x_{\text{of}}$, and then a foreground mask $G(x_{\text{img}}, x_{\text{of}})$ is generated. Manhattan distance is measured between generated mask and ground truth one $y$. So the loss function of STAM is,

$$L_{\text{STAM}} = ||G(x_{\text{img}}, x_{\text{of}}) - y||_1$$

STAM is trained by minimizing $L_{\text{STAM}}$. It detects foreground in each video frame by feeding spatio-temporal information without any post-processing like median-filtering.

In this work, we also apply adversarial process to STAM for comparison. We take STAM as a G, and it outputs fake mask $G(x_{\text{img}}, x_{\text{of}})$. We follow the D used in [51], and the objective function is:

$$\text{Goal}(G, D) = \arg \min_G \max_D \mathbb{E}_{(x_{\text{img}}, x_{\text{of}}, y)} \log(D(G(x_{\text{img}}, x_{\text{of}}, y))) + \mathbb{E}_{(1 - D(G(x_{\text{img}}, x_{\text{of}}, y))}) + \lambda L_{\text{STAM}}$$

$(x_{\text{img}}, x_{\text{of}}, y)$ is real pair while $(G(x_{\text{img}}, x_{\text{of}}), y)$ is fake pair. $E$ is an expectation function and $\lambda$ is a constant. $D(x_{\text{img}}, x_{\text{of}}, y)$ and $D(G(x_{\text{img}}, x_{\text{of}}), y)$ are the probabilities of real and fake pair calculated by D, respectively.

### 3. Experiments

#### 3.1. Preparing and setting

In order to train and evaluate our model, we employ the Change Detection 2014 dataset (CDnet 2014) [53], which contains various set of camera-captured videos including PTZ, bad weather (BDW), baseline (BSL), camera jitter (CJT), dynamic background (DBG), intermittent object motion (IOM), low frame rate (LFR), night videos (NVD), shadow (SHD), thermal (THM) and turbulence (TBL).

Following the experiment setting in [44], for the training data, we randomly select 5% samples with their ground truths of each subset from CDnet 2014. The samples left are used to test the efficiency of our model. The optical flow [54] of every video is extracted at 50% down-sampling ratio in advance. Each frame, ground truth and optical flow image are resized to $256 \times 256$. We avoid training scene-specific models for every scene but a single one to train all of the scenes in CDnet 2014. Segmented foreground is obtained without any post-processing.

Wallflower [20] and PETS [55] are two other challenging benchmarks for foreground detection. In order to explore the ability to detecting cross various scenes, we apply STAM trained on CDnet 2014 to test these two datasets without any additional training phase. We remove the attention module from STAM to get STAM$_{\text{NoAtt}}$ and combine the encoder feature and decoder feature through a concatenate operation straightly, also we remove the encoder layers associated with optical flow information from STAM that STAM$_{\text{NoOF}}$ outputs the foreground mask relying on static image only. STAM$_{\text{GAN}}$ employs an adversarial process. Their effects will be discussed then.

#### 3.2. Results and evaluation in CDnet 2014

We compute 7 different evaluation metrics for each algorithm compared in CDnet 2014, shown in Table 2. The STAM based method surpasses the state-of-the-art algorithms in most of the metrics. Especially, the Precision of standard STAM is 0.9851, while the rank second and third of Precision Cascade CNN 0.8997 and DeepBS 0.8332. STAM improves the Precision 9-15%. For Recall and FNR, CascadeCNN surpasses STAM, but...
Figure 3. Comparison on the output samples of STAM and the model without attention model. Each column has 5 images and there are static image, ground truth, optical flow, segmented results of STAM NoAtt and STAM, from left to right. Abbreviation: bad weather (BDW), camera jitter (CJT), dynamic background (DBG).

Table 2. Average performance comparison of different methods over the 11 categories in CDnet 2014. Abbreviation: STAM NoOF without optical flow, STAM NoAtt without attention, STAM GAN with adversarial net.

| Method         | Recall | Specificity | FPR  | FNR  | PWC  | F-measure | Precision |
|----------------|--------|-------------|------|------|------|-----------|-----------|
| STAM NoOF      | 0.9294 | 0.9955      | 0.0045 | 0.0706 | 0.6682 | 0.9030    | 0.8781    |
| STAM NoAtt     | 0.8364 | 0.9977      | 0.0023 | 0.1636 | 0.7698 | 0.8791    | 0.9265    |
| STAM GAN       | 0.9475 | 0.9995      | 0.0005 | 0.0525 | **0.2267** | **0.9655** | 0.9842    |
| STAM           | 0.9458 | **0.9995** | 0.0005 | 0.0542 | 0.2293 | 0.9651    | **0.9851** |
| Cascade CNN[45]  | 0.9506 | 0.9968      | 0.0032 | **0.0494** | 0.4052 | 0.9209    | 0.8997    |
| SuBSENSE[56]   | 0.8124 | 0.9904      | 0.0096 | 0.1876 | 1.6780 | 0.7448    | 0.7509    |
| GMM[13]        | 0.6846 | 0.9750      | 0.0250 | 0.3154 | 3.7667 | 0.5707    | 0.6025    |
| DeepBS[44]     | 0.7545 | 0.9905      | 0.0095 | 0.2455 | 1.9920 | 0.7548    | 0.8332    |
| IUTS-5[57]     | 0.7849 | 0.9948      | 0.0052 | 0.2151 | 1.1986 | 0.7717    | 0.8087    |
| PAWCS[58]      | 0.7718 | 0.9949      | 0.0051 | 0.2282 | 1.1992 | 0.7403    | 0.7857    |

Table 3. F-measures comparison of different methods on each category in CDnet 2014. Abbreviation: bad weather (BDW), baseline (BSL), camera jitter (CJT), dynamic background (DBG), intermittent object motion (IOM), low frame rate (LFR), night videos (NVD), shadow (SHD), thermal (THM) and turbulence (TBL).

| Method         | PTZ  | BDW   | BSL  | CJT  | DBG  | IOM  | LFR  | NVD  | SHD  | THM  | TBL  |
|----------------|------|-------|------|------|------|------|------|------|------|------|------|
| STAM           | 0.8648 | **0.9703** | **0.9885** | 0.8989 | 0.9483 | **0.9155** | 0.6683 | 0.7102 | 0.9663 | **0.9907** | 0.9328 |
| Cascade CNN[45]  | **0.9168** | 0.9431 | 0.9786 | **0.9758** | **0.9658** | 0.8505 | **0.8370** | **0.8965** | 0.9414 | 0.8958 | 0.9108 |
| SuBSENSE[56]   | 0.3476 | 0.8619 | 0.9503 | 0.8152 | 0.8177 | 0.6569 | 0.6445 | 0.5599 | 0.8464 | 0.8171 | 0.7792 |
| GMM[13]        | 0.1522 | 0.7380 | 0.8245 | 0.5969 | 0.6330 | 0.5207 | 0.5373 | 0.4097 | 0.7156 | 0.6621 | 0.4663 |
| DeepBS[44]     | 0.3133 | 0.8301 | 0.9580 | 0.8990 | 0.8761 | 0.6098 | 0.6002 | 0.5835 | 0.9092 | 0.7583 | 0.8455 |
| IUTS-5[57]     | 0.4282 | 0.8248 | 0.9567 | 0.8332 | 0.8902 | 0.7296 | 0.7743 | 0.5290 | 0.8766 | 0.8303 | 0.7836 |
| PAWCS[58]      | 0.4615 | 0.8152 | 0.9397 | 0.8137 | 0.8938 | 0.7764 | 0.6588 | 0.4152 | 0.8710 | 0.8324 | 0.6450 |
Figure 4. Comparison on the output samples of STAM and the model without temporal information. Each column has 5 images and there are static image, ground truth, optical flow, segmented results of STAM\textsuperscript{NoOF} and STAM, from left to right. Abbreviation: bad weather (BDW), night videos (NVD), shadow (SHD).

Figure 5. Samples segmented by different foreground detection methods among several scenes in CDnet-2014. Abbreviation: bad weather (BDW), baseline (BSL), camera jitter (CJT), shadow (SHD), thermal (THM).
Table 4. F-measures comparison of different methods in Wallflower.

| Category          | STAM    | DeepBS[44] | SuBSENSE[56] | PBAS[59] | GMM[13] |
|-------------------|---------|------------|--------------|----------|---------|
| Bootstrap         | 0.7414  | **0.7479** | 0.4192       | 0.2857   | 0.5306  |
| Camouflage        | 0.7369  | **0.9857** | 0.9535       | 0.8922   | 0.8307  |
| ForegroundAperture| **0.8292** | 0.6583 | 0.6635       | 0.6459   | 0.5778  |
| LightSwitch       | **0.9090** | 0.6114 | 0.3201       | 0.2212   | 0.2296  |
| TimeOfDay         | 0.3429  | 0.5494     | 0.7107       | 0.4875   | **0.7203** |
| WavingTrees       | 0.5325  | 0.9546     | 0.9597       | 0.8421   | **0.9767** |
| Overall           | 0.7138  | **0.7512** | 0.6711       | 0.5624   | 0.6443  |

Note that, STAM gives all the testing results on different scenes are given by this single model. While another model CascadeCNN was trained with a scene-specific style following its original experiment setting. For example, training a model on subset PTZ in CDnet2014 and tested on PTZ, while for another subset, Baseline, retrain the models and tested on Baseline. So their models could over-fit specific scene. But STAM is a single model to solve all the sub-scenes in CDnet2014 that doesn’t need to retrain for every scene in CDnet2014. So it is reasonable that the results of our model is worse than some scene-specific models. On the other hand, our model still gets improvement in scenes like Bad Weather, Shadow, Thermal and overall performance. What’s more, compared to the state-of-the-art model DeepBS with the same training and testing way [44], the proposed model gets significant improvements.

Figure 3 illustrates the samples segmented by the proposed model STAM and the model without attention processing. The STAM provides much clearer boundaries and accurate segmentation.

As illustrated in Figure 4, the STAM takes temporal information into consideration that helps to find objects hidden in static images and suppresses false alarm.

Table 3 represents the F-measures computed through STAM and the state-of-the-art approaches in different sub-sets. STAM gets the highest F-measure scores among the other algorithms in 6 out of 11 categories and the average performance, under the cases of bad weather, intermittent object motion, shadow, thermal, turbulence and light switch. The visualized results provided in Figure 5.

3.3. Results and evaluation in Wallflower and PETS

We directly apply the STAM trained on CDnet 2014 to Wallflower and PETS without any tuning to test its ability to segmenting cross scenes. There are 7 different scenes in Wallflower, and only one hand-segmented ground truth is provide for each scene. Because there is no foreground illustrated in the ground truth of “Moved Object”, we exclude this scene in the experiments. Table 4 illustrates the quantitative results on Wallflower, and STAM presents better performance on 2 subsets. Quantitative comparisons on another dataset PETS are exhibited in Table 5 and Figure 6 illustrates some samples segmented by the proposed model in Wallflower and PETS. Although we test on these two datasets without any retraining, STAM still provides acceptable results on both
Figure 6. Foreground detection in Wallflower and PETS. The first three rows are from Wallflower while the other three rows are from PETS.

quantitative and visualized results that proves its generalization. The test speed of STAM is about 10.8 fps for the frame size 256 by 256 on a single GTX1080TI with 32GB RAM and Ubuntu 16.04 LTS operating system.

4. Conclusion

We propose a Spatio-Temporal Attention Model for cross-scene foreground detection. The benefit of the proposed model is that, appearance and motion features, low-level and high-level features are synthesised to weaken the semantic gap by attention modules via guiding the feature learning procedure. The experiments on the forms of ablation also validate these two points: the model with optical flow will have 6% better performance than without it, the model with attention will have 9% better performance than without attention. Moreover, experiments also show that the adversarial module could be optional to improve some evaluation metrics. Quantitative and visualized performance on Wallflower and PETS benchmarks show the generalization ability of the model without any additional training. The performance of the proposed model surpasses state-of-the-art methods under the cases of bad weather, intermittent object motion, shadow, thermal, turbulence and light switch. It improves the overall Precision 9% and F-measure 4% than Cascade CNN. Also it is promising to process surveillance videos in real time.

References

1. Barnich, O.; Van Droogenbroeck, M. ViBe: A Universal Background Subtraction Algorithm for Video Sequences. *IEEE TIP* 2011, 20, 1709–1724.
2. Elgammal, A.; Duraiswami, R.; Harwood, D.; Davis, L.S. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE 2002*, 90, 1151–1163.
3. Kim, K.; Chalidabhongse, T.H.; Harwood, D.; Davis, L. Real-time foreground–background segmentation using codebook model. *Real-time imaging* 2005, 11, 172–185.
4. Wang, H.; Suter, D. Background subtraction based on a robust consensus method. *ICPR*. IEEE, 2006, Vol. 1, pp. 223–226.
5. Heikkila, M.; Pietikainen, M. A texture-based method for modeling the background and detecting moving objects. *IEEE TPAMI* 2006, 28, 657–662.
6. Learned-Miller, E.G.; Narayana, M.; Hanson, A. Background modeling using adaptive pixelwise kernel variances in a hybrid feature space. *CVPR*, 2012.
7. Liao, S.; Zhao, G.; Kellokumpu, V.; Pietikainen, M.; Li, S.Z. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. *CVPR*. IEEE, 2010, pp. 1301–1306.
8. Sheikh, Y.; Shah, M. Bayesian modeling of dynamic scenes for object detection. *IEEE TPAMI* 2005, 27, 1778–1792.
9. Zhao, X.; Satoh, Y.; Takauiji, H.; Kaneko, S.; Iwata, K.; Ozaki, R. Object detection based on a robust and accurate statistical multi-point-pair model. *Pattern Recognition* 2011, 44, 1296–1311.
10. Huynh-The, T.; Banos, O.; Lee, S.; Kang, B.H.; Kim, E.S.; Le-Tien, T. NIC: A Robust Background Extraction Algorithm for Foreground Detection in Dynamic Scenes. *IEEE TCSVT* 2016.

11. Chen, M.; Yang, Q.; Li, Q.; Wang, G.; Yang, M.H. Spatiotemporal Background Subtraction Using Minimum Spanning Tree and Optical Flow. *ECCV*, 2014, pp. 521–534.

12. Wren, C.R.; Azarbayejani, A.; Darrell, T.; Pentland, A.P. Pfinder: Real-time tracking of the human body. *IEEE TPAMI* 1997, 19, 780–785.

13. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. *CVPR*, 1999, Vol. 2.

14. Stauffer, C.; Grimson, W.E.L. Learning patterns of activity using real-time tracking. *IEEE TPAMI* 2000, 22, 747–757.

15. Rittscher, J.; Kato, J.; Joga, S.; Blake, A. A probabilistic background model for tracking. *ECCV* 2000, pp. 336–350.

16. Stenger, B.; Ramesh, V.; Paragios, N.; Coetzez, F.; Buhmann, J.M. Topology free hidden markov models: Application to background modeling. *ICCV*, 2001, Vol. 1, pp. 294–301.

17. Guo, J.M.; Liu, Y.F.; Hsia, C.H.; Shih, M.H.; Hsu, C.S. Hierarchical method for foreground detection using codebook model. *IEEE TCSVT* 2011, 22, 831–843.

18. Matsuuyama, T.; Ohya, T.; Habe, H. Background subtraction for non-stationary scenes. *ACCV*, 2000, pp. 662–667.

19. Seki, M.; Wada, T.; Fujiwara, H.; Sumi, K. Background subtraction based on cooccurrence of image variations. *CVPR*, 2003, Vol. 2, pp. II–65.

20. Toyama, K.; Krumm, J.; Brumitt, B.; Meyers, B. Wallflower: Principles and practice of background maintenance. *ICCV*, 1999, Vol. 1, pp. 255–261.

21. Oliver, N.M.; Rosario, B.; Pentland, A.P. A Bayesian computer vision system for modeling human interactions. *IEEE TPAMI* 2000, 22, 831–843.

22. Monnet, A.; Mittal, A.; Paragios, N.; Ramesh, V. Background modeling and subtraction of dynamic scenes. *ICCV*, 2003, pp. 1305–1312.

23. Zhong, J.; Sclaroff, S. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. *ICCV*, 2003, pp. 44–50.

24. Iwata, K.; Sato, Y.; Ozaki, R.; Sakaue, K. Robust background subtraction based on statistical reach feature method. *IEICE Transactions on Information and Systems* 2009, 92, 1251–1259.

25. Hu, W.; Li, X.; Zhang, X.; Shi, X.; Maybank, S.; Zhang, Z. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *IJCV* 2011, 91, 303–327.

26. Zhao, X.; Satoh, Y.; Takauji, H.; Kaneko, S.; Iwata, K.; Ozaki, R. Robust adapted object detection under complex environment. *AVSS*, 2011, pp. 261–266.

27. Liang, D.; Kaneko, S.; Sun, H.; Kang, B. Adaptive local spatial modeling for online change detection under abrupt dynamic background. *ICIP*, 2018.

28. Liang, D.; Kaneko, S.; Hashimoto, M.; Iwata, K.; Zhao, X. Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes. *Pattern Recognition* 2015, 48, 1374–1390.

29. Zhou, W.; Shun’ichi, K.; Manabu, H.; Yutaka, S.; Liang, D. Foreground detection based on co-occurrence background model with hypothesis on degradation modification in dynamic scenes. *Signal Processing* 2019, 60, 66–79.

30. Dai, J.; He, K.; Sun, J. Instance-Aware Semantic Segmentation via Multi-task Network Cascades. *CVPR* 2016, pp. 3150–3158.

31. Pinheiro, P.H.O.; Lin, T.; Collobert, R.; Dollar, P. Learning to Refine Object Segments. *ECCV* 2016, 9905, 75–91.

32. Dai, J.; He, K.; Li, Y.; Ren, S.; Sun, J. Instance-Sensitive Fully Convolutional Networks. *ECCV* 2016, pp. 534–549.

33. Uijlings, J.R.R.; De Sande, K.E.A.V.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. *IJCV* 2011, 104, 151–171.

34. Arbeláez, P.; Pont-Tuset, J.; Barron, J.; Marques, F.; Malik, J. Multiscale Combinatorial Grouping. *CVPR*, 2014, pp. 328–335.

35. Pinheiro, P.H.O.; Collobert, R.; Dollar, P. Learning to segment object candidates. *NIPS* 2015, pp. 1990–1998.

36. Kirillov, A.; Levinek, E.; Andres, B.; Savchynskyy, B.; Rother, C. InstanceCut: From Edges to Instances with MultiCut. *CVPR* 2017, pp. 7322–7331.

37. Bai, M.; Urtasun, R. Deep Watershed Transform for Instance Segmentation. *CVPR* 2017, pp. 2858–2866.

38. Arnah, A.; Torr, P.H.S. Pixelwise Instance Segmentation with a Dynamically Instantiated Network. *CVPR* 2017, pp. 879–888.

39. Liu, S.; Jia, J.; Fidler, S.; Urtasun, R. SGN: Sequential Grouping Networks for Instance Segmentation. *ICCV* 2017, pp. 3516–3524.

40. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R.B. Mask R-CNN. *ICCV* 2017, pp. 2980–2988.
41. Girshick, R. Fast R-CNN. *ICCV 2015*, pp. 1440–1448.

42. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE TPAMI 2017*, 39, 1137–1149.

43. Braham, M.; Droogenbroeck, M.V. Deep background subtraction with scene-specific convolutional neural networks. *IJWSSIP*, 2016.

44. Babaee, M.; Dinh, D.T.; Rigoll, G. A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition 2018*, 76, 635–649.

45. Wang, Y.; Luo, Z.; Jodoin, P. Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters 2017*, 96, 66–75.

46. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent Models of Visual Attention. *NIPS 2014*, pp. 2204–2212.

47. Larochelle, H.; Hinton, G.E. Learning to combine foveal glimpses with a third-order Boltzmann machine. *NIPS 2010*, pp. 1243–1251.

48. Kim, J.; Lee, S.; Kwak, D.; Heo, M.; Kim, J.; Ha, J.; Zhang, B. Multimodal Residual Learning for Visual QA. *NIPS 2016*, pp. 361–369.

49. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.C.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *ICML 2015*, pp. 2048–2057.

50. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid Attention Network for Semantic Segmentation. *British Machine Vision Conference 2018*, p. 285.

51. Isola, P.; Zhu, J.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR 2017*, pp. 5967–5976.

52. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *medical image computing and computer assisted intervention 2015*, pp. 234–241.

53. Goyette, N.; Jodoin, P.M.; Porikli, F.; Konrad, J.; Ishwar, P. Changedetection.net: A new change detection benchmark dataset. *CVPR CDW*, IEEE, 2012, pp. 1–8.

54. Liu, C. Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. *Doctoral Thesis, Massachusetts Institute of Technology 2009*.

55. Performance evaluation of PETS surveillance dataset 2001. [http://limu.ai.t.kyushu.ac.jp/en/dataset/](http://limu.ai.t.kyushu.ac.jp/en/dataset/).

56. Pierre-Luc, S.C.; Guillaume-Alexandre, B.; Robert, B. SuBSENSE: a universal change detection method with local adaptive sensitivity. *IEEE TIP 2014*, 24, 359–373.

57. Bianco, S.; Ciocca, G.; Schettini, R. Combination of Video Change Detection Algorithms by Genetic Programming. *IEEE Transactions on Evolutionary Computation 2017*, 21, 914–928.

58. St-Charles, P.L.; Bilodeau, G.A.; Bergevin, R. A Self-Adjusting Approach to Change Detection Based on Background Word Consensus. IEEE Winter Conference on Applications of Computer Vision, 2015.

59. Hofmann, M.; Tiefenbacher, P.; Rigoll, G. Background segmentation with feedback: The Pixel-Based Adaptive Segmentor. *CVPR Workshops*, 2012.

© 2019 by the authors. Submitted to *Sensors* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license ([http://creativecommons.org/licenses/by/4.0/](http://creativecommons.org/licenses/by/4.0/)).