Spatio-Temporal Self-Organizing Map Deep Network for Dynamic Object Detection from Videos

Yang Du[1], Chunfeng Yuan[1]*, Bing Li[1], Weiming Hu[1] and Stephen Maybank[2]

[1]CAS Center for Excellence in Brain Science and Intelligence Technology,
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences;
University of Chinese Academy of Sciences, Beijing, China
duyang2014@ia.ac.cn, cfyuan@nlpr.ia.ac.cn, bli@nlpr.ia.ac.cn, wmhu@nlpr.ia.ac.cn
[2]Birkbeck College, London
sjmaybank@dcs.bbk.ac.uk

Abstract

In dynamic object detection, it is challenging to construct an effective model to sufficiently characterize the spatial-temporal properties of the background. This paper proposes a new Spatio-Temporal Self-Organizing Map (STSOM) deep network to detect dynamic objects in complex scenarios. The proposed approach has several contributions: First, a novel STSOM shared by all pixels in a video frame is presented to efficiently model complex background. We exploit the fact that the motions of complex background have the global variation in the space and the local variation in the time, to train STSOM using the whole frames and the sequence of a pixel over time to tackle the variance of complex background. Second, a Bayesian parameter estimation based method is presented to learn thresholds automatically for all pixels to filter out the background. Last, in order to model the complex background more accurately, we extend the single-layer STSOM to the deep network. Then the background is filtered out layer by layer. Experimental results on CDnet 2014 dataset demonstrate that the proposed STSOM deep network outperforms numerous recently proposed methods in the overall performance and in most categories of scenarios.

1. Introduction

Dynamic object detection is a critical task for video processing in computer vision and it is fundamental for many applications such as target tracking, recognition and behavioral analysis [20]. Modern detection algorithms [9][15][4][21] are generally attained by background modeling. The difficulty of background modeling is to tackle background motions. We conclude that the motions of complex background mainly have two properties:

- Variation of the global background in the space. It is mainly caused by the zoom, translation, jitter, etc, of the camera. We refer to it as the spatial property of background motion.
- Variation of the local background in the time. It mainly indicates the dynamic elements in background and at different frames, such as river, fountain and bad weather. We refer to it as the temporal property of background motion.

In human visual system, visual cortex (V1) simultaneously perceives the stimuli of background and dynamic objects. Input-driven *Self-Organizing Map* (SOM) [19] is constructed by imitating the structures of superficial areas in the V1 to restore how the neurons in the essential visual cortex respond to input stimuli. SOM has been successfully used in dynamic object detection [16]. Generally, SOM based methods learn the information of background by updating the weights of neural nodes. Subsequently via setting a threshold for it, each pixel is filtered as the foreground or background. Weight updating and threshold setting are two key problems in the SOM based dynamic object detection.

Previous SOM based methods cannot sufficiently characterize the spatial-temporal properties of the background and are not well suitable for complex scenarios. In terms of the two properties of background motions, we propose a new STSOM and train it in two aspects, using the whole frames from spatial perspective and using the sequence of a pixel over time from temporal perspective. Then we propose a new method based on Bayesian parameter estimation

*Corresponding Author

[17] to automatically learn the spatio-temporal threshold of background filtering.

In order to further accurately model the complex background, we stack multiple STSOMs to form a deep network with a STSOM as a layer. The different parts of complex background are accurately modeled by different layers. The dynamic objects are detected by filtering out the background layer by layer and the segments are increasingly accurate with the deeper layer. Experiments prove that the proposed STSOM is able to effectively learn the spatio-temporal property of background in complex scenarios.

The architecture of this paper is arranged as follows. Some related works are briefly reviewed in **Section 2**. Then we briefly introduce the general Self-Organizing Map in **Section 3**. Subsequently, the dynamic object detection based on the STSOM deep network is detailed in **Section 4**. The comparison experiments are shown in **Section 5**. Finally, conclusions are given in **Section 6**.

2. Related Works

Non parametric methods [7] [8] directly rely on the observed data to statistically model the background. Although these methods can deal with fast changes in the background, they are time consuming and have an high memory requirement. Improvements have been proposed in [32][26][29] to overcome these problems. Some methods based on SOM [13] are proven to outperform traditional methods, such as KDE [8] and GMM [24][25][12][2]. But they cannot adapt to complex scenarios well.

SOM is used in background modeling by [16] for the first time and this method is called SOBS. Each pixel is modeled by a SOM and it is trained by updating weights of the winner node which has the smallest distance with the weight vectors, and its neighbouring nodes. This method utilizes the temporal property by modeling each pixel and utilizes the part of spatial property by the adjacent arrangement of SOMs of all pixels. An advanced method proposed by [5] reduces the numbers of neural nodes connecting to each pixel to one (called one-to-one SOM) and achieves good experimental results. Another relative method proposed by [6] automatically sets the threshold in one-to-one SOM [5] by fuzzy methods instead of manual setting. In [18], a neural network based on *Retinoscopic Self-Organzing Map* (RESOM) is proposed to accomplish background modeling. The background features are extracted by averaging weight matrixes of RESOM which has been trained sufficiently by input images. In [31], a *Stacked Multilayer Self-Organizing Map* (SMSOM) is proposed based on SOBS.

SOBS and some methods based on SOBS such as SMSOM have a large-scale arrangement of SOMs. It leads to a complex calculation. Specifically, every pixel is modeled by an individual SOM and is influenced by the adjacent SOM of other pixels. Assume the sizes of every SOM and a frame

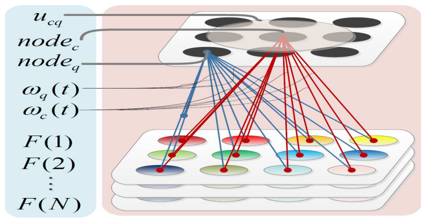


Figure 1. Structure of a self-organizing map.

are 5x5 and WxH respectively, the size of whole SOM is 5x5xWxH in SMSOM and SOBS. In our model, all pixels in a frame share a SOM of 5x5. We reduce the parameter numbers significantly.

Moreover, the above methods mainly consider either of spatial or temporal property, therefore it is hard to learn a sufficient background model in complex scenarios. Not considering the temporal property of background, RESOM is not very appropriate to the application where there are much dynamic elements in background. SOBS and SMSOM mainly consider the temporal property of background and the method to learn threshold is not precise enough. So they do not have a very good performance in complex scenarios. We train our network from spatio-temporal perspectives and propose a method of Bayesian parameter estimation to calculate the threshold. It makes SOM much adaptive in more scenarios in which previous SOM based methods nearly cannot work.

3. Self-Organizing Map

In this section, we briefly introduce the SOM [14]. The general SOM consists of a set of neural nodes which learn feature patterns of the input stimulus through the self-organizing of the weights of the neural nodes. The structure of SOM is shown in **Figure 1**. Assume that the input stimulus is a time sequence $\{F(t)\}_{t=1}^{N}$, where $F(t)$ is the *t-th* input of a P-dimensional real vector and N is the length of the input. There are Q neural nodes which are denoted as $node_q$ $(q = 1, 2, .., Q)$ in the SOM. The elements of input are fully connected to all nodes, and the connection is represented by a weight vector, denoted as $w_q(t)$. Specifically, the winner node $node_c$ is defined as the one with the weight vector $w_c(t)$ that has the smallest Euclidean distance from $F(t)$. The index c of $node_c$ is formulated as:

$$c = \underset{q}{argmin}\{||F(t) - w_q(t)||\}. \tag{1}$$

The learning rule of SOM is that, finding the winner node then updating the weights of the winner node and its neighbouring nodes. This updating rule is the criteria of SOM proposed by Kohonen [14]. The weights of SOM are updated by,

$$w_q(t+1) = w_q(t) + u_{cq} * \alpha(t) * [F(t) - w_q(t)], \tag{2}$$

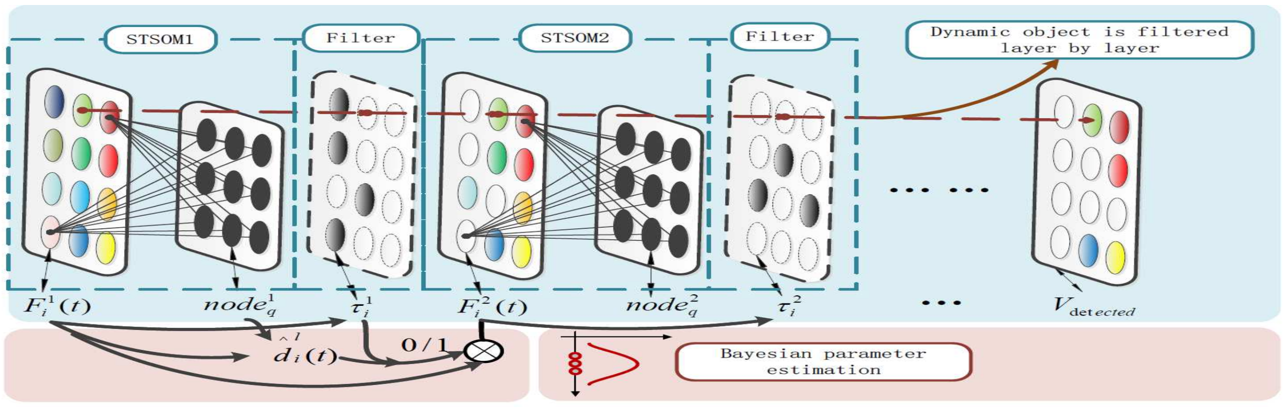


Figure 2. The architecture of the STSOM deep network.

where $\alpha(t)$ is the learning rate and u_{cq} is the neighborhood function. Generally, $u_{cq} = \exp\left(-\frac{||q-c||}{2\sigma^2}\right)$. This function resembles the kernel that is applied in usual smoothing processes. SOM uses this neighborhood function to preserve the topological properties of the input space by updating their weights of neighboring nodes of the winner node.

4. STSOM Deep Network for Dynamic Object Detection

In this section, we firstly give an overview of the proposed STSOM deep network and then present a pre-training method of the network. Subsequently, we describe the fine-tuning of this network. Finally, we describe how the final dynamic objects are detected from the STSOM deep network.

4.1. Overview

The basic structure of the STSOM is the same with SOM as illustrated in **Figure 1**. All input pixels are fully connected to neural nodes which learn feature patterns of the input stimulus through the self-organizing of the weights between neural nodes and input pixels. We construct a STSOM deep network containing multiple stacked STSOM layers to model all pixels of each frame from a video, as shown in **Figure 2**. We assume that $F_i^l(t)$ represents a pixel in a frame, where *i* $(1 \leq i \leq P)$ is the index of the pixel, *P* is the number of all pixels in a frame, *l* $(1 \leq l \leq L)$ is the index of the layer, *L* is the number of STSOM layers, and *t* is the temporal index of the frame. For each layer, we use a STSOM to model all pixels in a frame. The neural node of the STSOM in the l-th layer is denoted as $node_q^l$, where q represents the index of the node, $q = 1, 2, ..., Q$ (Q is the number of neural nodes). The node number of STSOM can be customized and the weight of the connection between $F_i^l(t)$ and $node_q^l$ is denoted by $w_{iq}^l(t)$. For example, we assume the node number of the first layer is 9 (3×3), so every pixel is associated with 9 neural nodes in this layer. In order to obtain a highly accurate background model, we stack multiple STSOM layers with the same structures to construct a deep network, in which the next STSOM layer models the left background filtered by the previous STSOM layer. In theory, the deeper layer obtains the more accurate modeling of the background. The input of the 1-th STSOM layer is the input video sequence. Subsequently, the input video is filtered by the threshold τ_i^1 and the filtered results are propagated forward to the next layer as the input. The output of the last layer is the dynamic object. Next we will introduce how to pre-train this network, namely learning the weights of every neural node and the spatial-temporal threshold of background filtering.

4.2. Pre-training

We propose a new spatio-temporal updating method to pre-train this deep network. Specifically, a video with N frames $\{F_i^1(t), t = 1, 2, ..., N\}$ is used to train the first STSOM layer one by one. Since HSV color model is similar to human perception so we use HSV color information to represent each pixel. Namely, $F_i^1(t) = \{h_i^1(t), s_i^1(t), v_i^1(t)\}$. The input of the other layers $F_i^l(t)$ $(l = 2, ..., L)$ has the same form with $F_i^1(t)$. As SOM uses the weights of neural nodes to learn the patterns of input, the weight of the connection between $F_i^l(t)$ and $node_q^l$ has the same form. Namely, $w_{iq}^l(t) = \{w_{h_{iq}}^l(t), w_{s_{iq}}^l(t), w_{v_{iq}}^l(t)\}$. Because HSV color model lies in Hexcone Space, we calculate the distance $d_{iq}^l(t)$ between $F_i^l(t)$ and $node_q^l$ in Catesian space [31] as follows,

$$\begin{aligned} d_{iq}^l(t) = &||(v_i^l(t)s_i^l(t)\cos(h_i^l(t)), v_i^l(t)s_i^l(t)\sin(h_i^l(t)) \\ &, v_i^l(t)) - (w_{v_{iq}}^l(t)w_{s_{iq}}^l(t)\cos(w_{h_{iq}}^l(t)), \\ &w_{v_{iq}}^l(t)w_{s_{iq}}^l(t)\sin(w_{h_{iq}}^l(t)), w_{v_{iq}}^l(t))||_2^2. \end{aligned} \tag{3}$$

As a result, we obtain a distance matrix $D^l(t)$, which consists of $d_{iq}^l(t)$ between the input pixel and the neural node. This matrix is the fundamental tool of spatio-temporal weight updating. This updating process is divided into two steps, which includes spatial weight updating and temporal weight updating.

4.2.1 Spatial Weight Updating

The deep model based on STSOM must has a representative ability to tolerate the global variance of the background. So the first step is to let the deep network learn the spatial characteristics of the background using the whole frames to train this network. The distance between the t-th frame and the q-th node $node_q^l$ in the l-th STSOM layer is defined as follows,

$$D_{spatial.q}^l(t) = \sum_{i=1}^{P} d_{iq}^l(t). \tag{4}$$

The node with the minimum distance among the distances between the t-th frame and all nodes, is selected as the winner node for this frame. The position of the winner node is defined as $q_{spatial}^*(t)$, formulated as follows,

$$q_{spatial}^*(t) = \underset{q}{argmin}\{D_{spatial.q}^l(t), q = 1, 2, ..., Q\}. \tag{5}$$

After the position of the winner node $q_{spatial}^*(t)$ is calculated, the weights between the winner node and all input pixels as well as the weights between each of the neighboring nodes of the winner node and all input pixels are updated. The weight is updated by the following formula,

$$w_{iq}^l(t+1) = u_q * (w_{iq}^l(t) + \alpha_{train} * (F_i^l(t) - w_{iq}^l(t)), \tag{6}$$

where α_{train} is the learning rate of the STSOM deep network and it is set manually and u_q represents the updating degree of the weight. In this paper, u_q is set as a Gaussian kernel function $u_q = \exp\left(-\frac{||q-q_{spatial}^*(t)||}{2\sigma^2}\right)$, where σ^2 is the variance of Gaussian kernel function. When the winner node is updated, $u_q = 1$. Its value changes smaller when the physical distance of the neighbouring node towards the winner node is further.

4.2.2 Temporal Weight Updating

The training process should consider the variation of each pixel at different frames to adapt to the variation of local background. So the second step is to let the deep network learn the temporal characteristics of the background by train it using the sequence of values taken by a pixel over time. For the pixel i in the l-th layer, its winner node is the node with the minimum distance from this pixel among all nodes, which are connected to this pixel. The position of the i-th pixel's winner node is

$$q_{i.temporal}^*(t) = \underset{q}{argmin}\{d_{iq}^l(t), q = 1, 2, ..., Q\}. \tag{7}$$

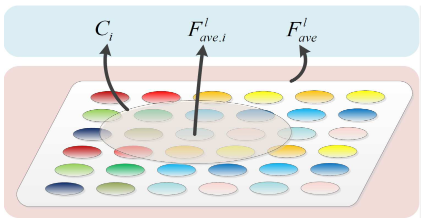


Figure 3. Estimation of γ_i and δ_i^2 with MLE.

It is similar to spatial weight updating. After the winner node $q_{i.temporal}^*(t)$ is obtained, the weights between the pixel i and its winner node as well as its neighbours are updated by **Eq. 6**. Combination of both the spatial and temporal weight updating makes STSOM have superiorly representative ability. So it is able to tolerate both the global variation of the whole frame and the partial variation of single pixel in the background.

4.2.3 Forward Propagation

The STSOM deep network consists of cascaded STSOM layers and the pre-training process proceeds layer by layer. The extracted foreground is transferred from the current layer to the next layer till the last STSOM layer. Assume that the first layer has been trained successfully, then the input data of the next layer is filtered by a threshold for each pixel. The variation of whole frame and the variation of single pixel at different frames will simultaneously contribute to the threshold of each pixel, so the threshold of each pixel is obtained by fusion of spatial and temporal thresholds.

First, we obtain a rough background model from the training frames by follows, $F_{ave.i}^l = \frac{1}{N}\sum_{t=1}^{N} F_i^l(t)$. Next, we use the method of Bayesian parameter estimation to achieve the final background model. Let B_i^l (i is the index of pixel) denote the final background model. Assume that $B_i^l \sim \mathbf{N}(\mu_i, \sigma_i^2)$, $\mu_i \sim \mathbf{N}(\gamma_i, \delta_i^2)$, and $F_i^l(t)$ is sampled from the B_i^l, its probability density function is denoted as $\mathbf{N}(F_i^l(t)|\mu_i, \sigma_i^2)$, where $\mathbf{N}$ denotes the Gaussian distribution and the distribution of μ_i is the prior distribution of background in this video. We estimate γ_i and δ_i^2 of pixel i by its surrounding pixels using Maximum Likelihood Estimation (MLE), formulated as follows,

$$\hat{\gamma_i} = 1/|C_i| \sum_{i \in C_i} F_{ave.i}^l,$$
$$\hat{\delta_i^2} = 1/|C_i| \sum_{i \in C_i} (F_{ave.i}^l - \hat{\gamma_i})(F_{ave.i}^l - \hat{\gamma_i})^T, \tag{8}$$

where C_i denotes the surrounding pixels of the pixel i as shown in **Figure 3**. Experiments demonstrate that the radius of C_i within 7 to 10 pixels is effective. Smaller radius has slightly improvement on the result and bigger radius drops results from good performance. By derivations, we can estimate μ_i as follows,

$$\hat{\mu_i} = \frac{N/\sigma_i^2}{N/\sigma_i^2 + 1/\hat{\delta}_i^2} F_{ave.i}^l + \frac{1/\hat{\delta}_i^2}{N/\sigma_i^2 + 1/\hat{\delta}_i^2} \hat{\gamma}_i. \quad (9)$$

We derive that σ_i^2 can be expressed as a function of $\hat{\mu_i}$. Then, the logarithm likelihood function $L(\hat{\mu_i}, \sigma_i^2)$ of $\hat{\mu_i}$ and σ_i^2 is formulated as follows,

$$L(\hat{\mu_i}, \sigma_i^2) = \sum_{t=1}^{N} \ln \mathbf{N}(F_i^l(t) | \hat{\mu_i}, \sigma_i^{\mathbf{2}}). \quad (10)$$

Next, we use Maximum Likelihood Estimation to maximize $L(\hat{\mu_i}, \sigma_i^2)$,

$$\hat{\mu_i}|_{\frac{\partial L}{\partial \hat{\mu_i}}=0} = \underset{\hat{\mu_i}}{argmax}\{L(\hat{\mu_i}, \sigma_i^2)\}. \quad (11)$$

Finally we use $\hat{\mu_i}$ to estimate B_i^l. Using the obtained background model B_i^l, we calculate the distance matrix $D^l = (d_{iq}^l)_{i=1,2,...,P;q=1,2,...,Q}$ where d_{iq}^l is the distance between B_i^l and $node_q^l$. According to the matrix D^l, we define two thresholds to filter out the background. Using **Eq. 4**, the spatial threshold $\tau_{spatial.i}^l$ of each pixel is calculated as follows,

$$\tau_{spatial.i}^l = max\{D_{spatial.q}^l, q = 1, 2, ..., Q\}/P. \quad (12)$$

Next we calculate the temporal threshold $\tau_{temporal.i}^l$ of each pixel by the following formula,

$$\tau_{temporal.i}^l = max\{d_{iq}^l, q = 1, 2, ..., Q\}. \quad (13)$$

The final threshold τ_i^l is calculated by the average of them,

$$\tau_i^l = (\tau_{temporal.i}^l + \tau_{spatial.i}^l)/2. \quad (14)$$

After the pre-training of the l-$layer$ STSOM of our method, the threshold of every pixel is learnt automatically by the formula above. We recompute the distance between all pixels in the training frames and the newest learnt nodes of the l-$layer$, and obtain the distance between $F_i^l(t)$ and its winner node, denoted as $\hat{d}_i^l(t)$, namely,

$$\hat{d}_i^l(t) = \min\{\hat{d}_{i1}^l(t), \hat{d}_{i2}^l(t), ..., \hat{d}_{iQ}^l(t)\}. \quad (15)$$

If $\hat{d}_i^l(t) > \tau_i^l$, $F_i^l(t)$ is considered as the dynamic object and is used as the input of the next layer, namely $F_i^{l+1}(t) = F_i^l(t)$. Otherwise, this pixel is considered as the background and $F_i^{l+1}(t) = 0$. So far the training data of the next STSOM layer has been determined in the form of $\{F_i^{l+1}(1), F_i^{l+1}(2), ..., F_i^{l+1}(N)\}$. By repeating the process of spatial and temporal weight updating, the next STSOM layer will be trained so that this deep network will be pre-trained layer by layer and meanwhile spatio-temporal thresholds of every layer will be obtained in order to generate the input of the next layer.

4.3. Fine-tuning

The pre-training process provides the STSOM deep network with an effective initialization. In order to make it more adaptive for the change of complex scenarios, the weights of each layer are online updated by the fine-tuning process but the thresholds are not updated, when a new frame enters into this deep network. Specifically, both the spatial weights and the temporal weights are updated. This process is similar to pre-training as follows,

$$w_{iq}^l(t+1) = w_{iq}^l(t) + u_q * \alpha_{update} * (F_i^l(t) - w_{iq}^l(t)). \quad (16)$$

In pre-training, we use background frames to train STSOM and set α_{train} to a bigger value to converge STSOM efficiently. In fine-tuning, we adjust α_{update} to tackle the change of background. The thresholds are learnt in pre-training and do not change in fine-tuning. α_{update} is set according to the change rate of background. Generally it is set as a small value so that the deep network will not change too fast and be stable.

4.4. Dynamic Object Detection

When the deep network is completely trained, this model can be used for dynamic object detection. A new frame of a video enters into this network and then its dynamic objects will be extracted layer by layer till the last output layer. More specifically, the deeper layer represents the more accurate modeling of the background and the result of dynamic object detection will be more effective with the deeper layers in theory. At the last layer, a binary matrix of the same size as the original frame is constructed as follows,

$$V_{detected.i} = \begin{cases} 1, & \hat{d}_i^L(t) > \tau_i^L, \\ 0, & otherwise. \end{cases} \quad (17)$$

5. Experiments

In this section, we evaluate the proposed STSOM in four aspects and the parameters are set as follows, $\alpha_{train} = 0.8$, $\alpha_{update} = 0.005$, $Q = 25$, $L = 3$, $C_i = 8$.

5.1. Evaluation on the CDnet 2014 Dataset

We evaluate our method on the CDnet 2014 dataset which totally includes 53 scenarios, categorized into 11 classes, baseline (BL), dynamic background (DB), camera jitter (CJ), shadow (SH), intermittent object motion (IOM), thermal (TH), bad weather (BW), low framerate (LF), night videos (NV), PTZ (PTZ) and turbulence (TU). The state-of-the-art experimental results on CDnet 2014 can be downloaded from CDnet website: http://changedetection.net/. The official metrics [28][10] used to evaluate methods are Recall (Re), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classifications (PWC), Precision (Pr) and

Category	$Recall$	$Specificity$	FPR	FNR	PWC	$Precision$	$F\text{-}Measure$
baseline	0.9644	0.9985	0.0015	0.0356	0.3960	0.9546	0.9576
camera jitter	0.8677	0.9908	0.0092	0.1323	1.8882	0.9094	0.8881
dynamic background	0.8970	0.9991	0.0009	0.1030	0.4397	0.9557	0.9235
intermittent object motion	0.8698	0.9936	0.0064	0.1302	1.1467	0.8041	0.8357
shadow	0.9203	0.9920	0.0080	0.0897	1.3050	0.8812	0.9003
thermal	0.8258	0.9912	0.0088	0.1742	1.9796	0.8732	0.8488
bad weather	0.9125	0.9975	0.0025	0.0875	0.4080	0.8736	0.8926
low framerate	0.8056	0.9923	0.0077	0.1944	1.5475	0.8197	0.8125
night videos	0.5974	0.9654	0.0346	0.4026	3.4946	0.5325	0.5631
ptz	0.7909	0.9731	0.0269	0.2091	3.1760	0.4462	0.5759
turbulence	0.8398	0.9989	0.0011	0.1602	0.1780	0.7662	0.8009
Overall	0.8447	0.9902	0.0098	0.1563	1.4508	0.7988	0.8164

Table 1. Complete results of the proposed STSOM deep network on the CDnet2014 dataset.

F-Measure (FM). We primarily use FM to compare the performance as it is closely correlated with the ranks used on the CDnet website, and is generally accepted as a good indicator of overall performance. We compare our results with the methods mentioned in **Related Works** and the best methods reported on the official website above, including 12 methods in total, IUTIS-5 [3], SharedModel [30], SuBSENSE [22], PAWCS [23], C-EFIC [1], MBS [11], FTSG [27], S-Subsense, SMSOM, SOBS, KDE and GMM.

Table 5.1 shows complete results of STSOM deep network on the CDnet2014 dataset and **Table** 5.1 shows the overall and per-category FM of our method and the state-of-the-art methods. The following points are concluded according to different scenario classes.

- $dynamic\ background$ includes river, dynamic trees , fountains and etc. The videos in $bad\ weather$ consists of outdoor surveillance footage taken under snowy conditions and $turbulence$ shows long distance thermal-infrared video surveillance with important air turbulence due to a high temperature environment. Both of these three classes have dynamic elements of partial background over time. And there are global jitter of background in the $camera\ jitter$. Our method promotes 5.5% performance in $camera\ jitter$ scenarios and promotes 3.3% average performance in these four classes compared with the best one among state-of-the-art methods. It demonstrates our model can tolerate both the change of the overall region in the space and local pixels of the background over time.

- $thermal$ are composed of gray scale images so similar gray scale results in detection difficulty for foreground. The $shadow$ contains videos with prevalent hard and soft shadow which is challenging to be differentiated from dynamic objects. There are similar color elements in both foreground and background in these two classes. Experimental results show that our method is able to accurately model the background which is difficult to be differentiated from foreground by color.

- The $intermitten\ object\ motion$ and $low\ framerate$ classes, which are challenging for the adaptability of methods, contain background objects moving away, discontinuous frames, abandoned objects and objects stopping for a short while and then moving away. Our network achieves the best results on them since we can effectively update our model in real time.

- The difficulty of PTZ lies in that the background is moving all the way and the neon glow in $night\ videos$ makes it hard to distinguish the foreground covered by the light. Our method has unsatisfactory results on $night\ videos$ and PTZ but they are still comparable to other top-ranked methods.

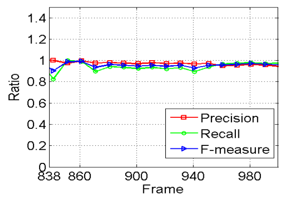
- In general, we can find that our method has nearly 82% overall performance on CDnet 2014 and outranks the state-of-the-art methods in overall performance and in most categories. Moreover, our method exceeds the currently best method IUTIS-5 5% and another SOM based method SOBS 20%.

5.2. Evaluation of Pre-training and Fine-tuning

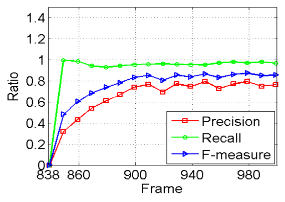
We respectively test our method in the pre-training way and the way without pre-training on the video $canoe$ from the 838th frame. $canoe$ is a video with dynamic background and a boat appears in the river from the 838th frame. The first 837 frames are all the background. Nearly 70% of the background is the dynamic river so that it is challenging to exclude the disturbance of dynamic background. For the pre-training method, weights are initialized with the first frame of background, namely, $w_{iq}^l(1) = F_i^l(1)$, $q = 1, 2, .., Q$. This network is pre-trained and thresholds are learnt by using the first 837 frames. The results are shown in **Figure** 4(a). It can be seen that STSOM shows a strong

Method	$FM_{overall}$	FM_{BL}	FM_{CJ}	FM_{DB}	FM_{IOM}	FM_{SH}	FM_{TH}	FM_{BW}	FM_{LF}	FM_{NV}	FM_{PTZ}	FM_{TU}
STSOM	**0.816**	**0.957**	**0.888**	**0.923**	**0.835**	**0.910**	**0.848**	**0.892**	**0.812**	*0.563*	*0.575*	**0.800**
IUTIS-5	*0.771*	*0.956*	*0.833*	0.890	0.729	*0.908*	0.830	0.824	*0.774*	0.529	0.428	*0.783*
SharedModel	0.747	0.952	0.814	0.822	0.672	0.845	0.831	0.798	0.728	0.541	0.386	0.733
SuBSENSE	0.741	0.950	0.815	0.817	0.656	0.864	0.817	*0.861*	0.644	0.559	0.347	0.779
PAWCS	0.740	0.939	0.813	*0.893*	0.776	0.871	0.832	0.815	0.658	0.415	0.461	0.645
C-EFIC	0.730	0.930	0.824	0.562	0.622	0.845	*0.834*	0.786	0.680	**0.667**	**0.620**	0.627
MBS	0.728	0.928	0.836	0.791	0.756	0.826	0.819	0.798	0.635	0.515	0.552	0.585
FTSG	0.728	0.933	0.751	0.879	*0.789*	0.853	0.776	0.822	0.625	0.513	0.324	0.712
S-Subsense	0.717	0.948	0.807	0.815	0.601	0.865	0.685	0.859	0.651	0.534	0.339	0.751
SMSOM	-	0.944	0.732	0.675	-	-	0.793	-	-	-	-	-
SOBS	0.596	0.933	0.705	0.643	0.562	0.721	0.683	0.662	0.546	0.450	0.040	0.488
KDE	0.568	0.909	0.572	0.596	0.408	0.766	0.742	0.757	0.547	0.436	0.036	0.447
GMM	0.556	0.838	0.596	0.633	0.520	0.715	0.662	0.738	0.537	0.409	0.152	0.466

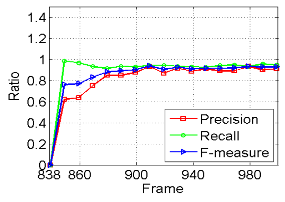
Table 2. Overall and per-category FM of our method and the state-of-the-art methods tested on the CDnet 2014 recently. Bold entries indicate the best results and blue/italics the second best.



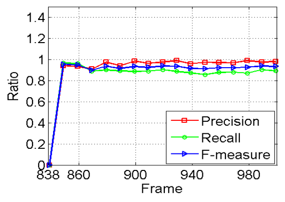
(a) pre-training & fine-tuning



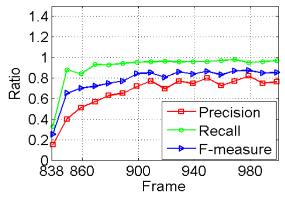
(b) τ_i^l = 0.01, without pre-training



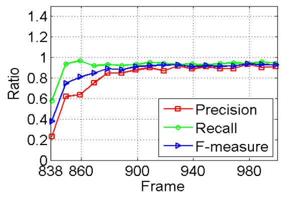
(c) τ_i^l = 0.02, without pre-training



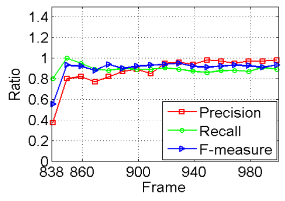
(d) τ_i^l = 0.03, without pre-training



(e) τ_i^l = 0.01, with pre-training



(f) τ_i^l = 0.02, with pre-training



(g) τ_i^l = 0.03, with pre-training

Figure 4. Curves of *Precision*, *Recall* and *F-Measure* with respects to the time on *canoe*.

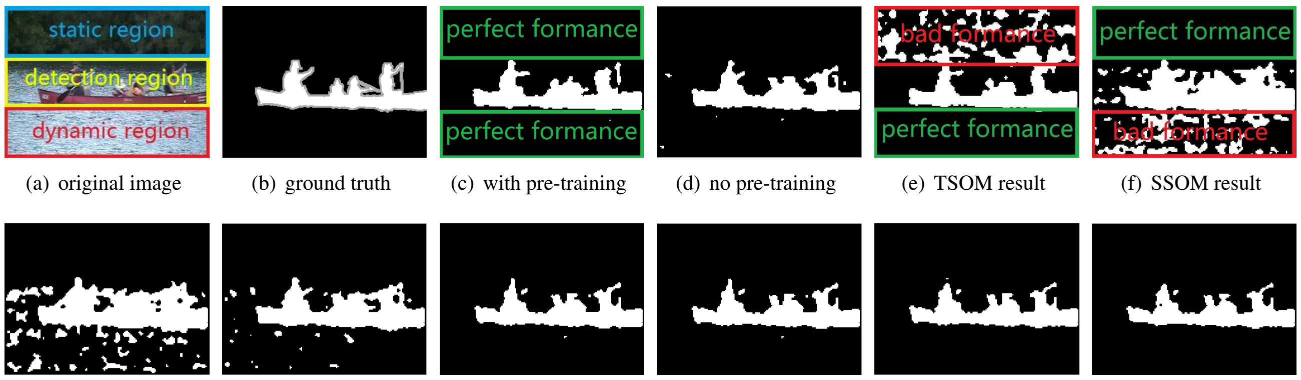
representative ability from the 838th frame. If there is only fine-tuning without pre-training, weights are initialized with zeros, namely, $w_{iq}^l(1) = 0$. In **Figure 4(b)** - **4(d)**, these experiments are implemented without pre-training process (only fine-tuning process) and τ_i^l is artificially set to a fixed value. We can find that the performance of STSOM deep network is poor at first but these curves rise gradually and converge to the state with good and steady detection results later. The detection starts from the 838th frame in both two ways. All of these results demonstrate that pre-training provides a nice initialization including thresholds and weights of STSOM, and fine-tuning makes STSOM much adaptive.

In order to evaluate the effectiveness of the learnt thresholds based on Bayesian parameter estimation, we respectively test our STSOM methods with the fixed thresholds and the learnt thresholds after pre-training. The results of our method with the learnt thresholds after pre-training are shown in **Figure 4(a)**. The results of our method with the fixed thresholds after pre-training are shown in **Figure 4(e)** - **4(g)**. We can find that the method with the learnt thresholds largely outperforms that with the fixed thresholds at the beginning. Later, results of the latter converge to a satisfactory state by fine-tuning but are still worse than the former. It demonstrates that the learnt thresholds are more effective.

The actual detection results of the 977th frame corresponding to **Figure 4(a)** and **Figure 4(c)** on *canoe* are shown in **Figure 5(c)** and **Figure 5(d)**.

5.3. Evaluation of SSOM, TSOM and STSOM

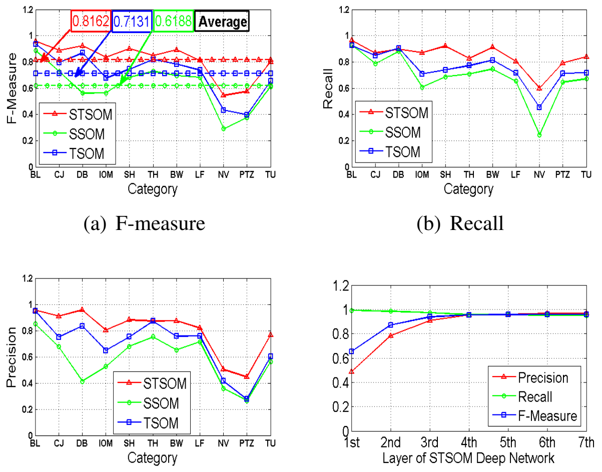
We compare our STSOM with another two methods, one is *Spatial* SOM (SSOM), which only has spatial weight updating and spatial thresholds. The other one is *Temporal* SOM (TSOM), which only has temporal weight updating and temporal thresholds. Quantitative results of SSOM, TSOM and STSOM on CDnet2014 are shown in **Figure 6(a)**



(a) original image (b) ground truth (c) with pre-training (d) no pre-training (e) TSOM result (f) SSOM result

(g) result of 1st layer (h) result of 2nd layer (i) result of 4th layer (j) result of 5th layer (k) result of 6th layer (l) result of 7th layer

Figure 5. Detection results of frame 977 on *canoe*.



(a) F-measure (b) Recall

(c) Precision (d) Results about layers

Figure 6. Performance curves of STSOM, SSOM ,TSOM and multiple layers.

- 6(c). STSOM promotes nearly 20 % and 10 % on overall performance compared with SSOM and TSOM respectively. Results show that STSOM has a better overall performance and is more universal. STSOM is slightly slower than SOM and TSOM in pre-training, while the speed of them are nearly same in detection process with the same size of video. In **Figure** 5, we show the detection results of the 977th frame of *canoe* by several methods. **Figure** 5(a) is the original frame, which mainly includes three parts: static region composed of trees, detection region and dynamic region composed of changeable river. From **Figure** 5(f) and **Figure** 5(e), It can be seen that SSOM is more effective than TSOM in static region and the result is opposite in dynamic region. The result of STSOM in **Figure** 5(c) shows that our model for combining spatial and temporal properties of background is more effective in both static and dynamic backgrounds.

5.4. Evaluation of Deep Network

In order to demonstrate that our deep network is more effective than single layer, we list 6 results of other layers on frame 977 as shown in **Figure** 5(g) - 5(l). In order to exclude the influence of pre-training, this experiment was implemented without pre-training. The result of the third layer is shown in **Figure** 5(d). In the first layer and the second layer the detection results are bad, but from the 3rd layer we begin to obtain satisfactory results. So other experiments in this paper are implemented with the three-layers architecture taking both the speed and effect into consideration. Quantitative results on frame 977 is shown in **Figure** 6(d). This result demonstrates that more superior detection performance can be obtained by constructing more deeper network of STSOM.

6. Conclusion

In this paper, we have proposed a *Spatio-Temporal Self-Organizing Map* (STSOM) and a new training method which is composed of both spatial and temporal weight updating. Then, all pixels in the same frame have been efficiently modeled by a shared STSOM. Based on Bayesian parameter estimation, the threshold for object detection has been learnt by exploiting spatio-temporal properties of the background. Moreover, we have extended the single STSOM to a deep network with a more superior performance than other existing methods in numerous comparison experiments.

7. Acknowledgement

This work is partly supported by the 973 basic research program of China (Grant No. 2014CB349303), the Natural Science Foundation of China (Grant No. U1636218, 61472420, 61472063, 61370185, 61472421), and the CAS External cooperation key project.

References

[1] G. Allebosch, D. V. Hamme, F. Deboeverie, P. Veelaert, and W. Philips. C-EFIC: Color and edge based foreground background segmentation with interior classification. *Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2015. 6

[2] M. S. Allili, N. Bouguila, and D. Ziou. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Journal of Electronic Imaging*, 17(1):1778–1792, 2005. 2

[3] S. Bianco, G. Ciocca, and R. Schettini. How far can you get by combining change detection algorithms? *IEEE Transactions on Image Processing, arXiv*, 2015. 6

[4] T. Bouwmans. Recent advanced statistical background modeling for foreground detection - a systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011. 1

[5] M. Chacon, G. Sergio, and V. Javier. Simplified som-neural model for video segmentation of moving objects. *International Joint Conference on Neural Networks*, pages 474–480, 2009. 2

[6] M. Chacon-Murguia and S. Gonzalez-Duarte. An adaptive neuralfuzzy approach for object detection in dynamic backgrounds for surveillance systems. *IEEE Transactions on Industrial Electronics*, 59(99):1–1, 2012. 2

[7] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Computer Vision and Pattern Recognition Workshops*, 90(7):1151–1163, 2002. 2

[8] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *European Conference on Computer Vision*, 1843:751–767, 2000. 2

[9] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed. Moving object detection in spatial domain using background removal techniques-stateof-art. *Recent patents on computer science*, 1(1):32–54, 2008. 1

[10] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. *IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 1–8, 2012. 5

[11] S. Hasan and S.C.Sen-Ching. Universal multimode background subtraction. *IEEE Transactions on Image Processing*, 2015. 6

[12] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure. Robust foreground extraction technique using background subtraction with multiple thresholds. *Optical Engineering*, 46(9):097004–097004, 2007. 2

[13] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biol. Cybern*, 43(1):59–69, 1982. 2

[14] T. Kohonen. Essentials of the self-organizing map. *Neural Networks*, 37(1):52–65, 2013. 2

[15] D. B. M. Cristani, M. Farenzena and V. Murino. Background subtraction for automated multisensor surveillance: a comprehensive review. *EURASIP Journal on Advances in signal Processing*, 2010:43, 2010. 1

[16] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(2):1168–1177, 2008. 1, 2

[17] M. Radford. Bayesian methods for machine learning. *Conference and Workshop on Neural Information Processing*, 2004. 2

[18] J. Ramirez-Quintana and M. Chacon-Murguia. Self-organizing retinotopic maps applied to background modeling for dynamic object segmentation in video sequences. *International Joint Conference on Neural Networks*, pages 1–8, 2013. 2

[19] M. Risto, A. James, and C. Yoonsuck. Computational maps in the visual cortex. 2005. 1

[20] M. Sedky, M. Moniri, and C. Chibelushi. Classification of smart video surveillance systems for commercial applications. *Advanced Video and Signal Based Surveillance. AVSS 2005. IEEE Conference on*, pages 638–643, 2005. 1

[21] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122(0):4–21, 2014. 1

[22] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin. Subsense : A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 2014. 6

[23] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin. A self-adjusting approach to change detection based on background word consensus. *IEEE Winter Conference on Applications of Computer Vision*, pages 6–9, 2015. 6

[24] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1999. 2

[25] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition*, 2:246–252, 1999. 2

[26] D. A. T. Tanaka, A. Shimada and R. i. Taniguchi. A fast algorithm for adaptive background model construction using parzen density estimation. *Advanced Video and Signal Based Surveillance*, pages 528–533, 2007. 2

[27] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan. Static and moving object detection using flux tensor with split gaussian models. *Computer Vision and Pattern Recognition Workshops*, pages 420–424, 2014. 6

[28] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. CDnet 2014: An expanded change detection benchmark dataset. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 387–394, 2014. 5

[29] H. N. Y. Nonaka, A. Shimada and R. i. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. *Computer Vision and Pattern Recognition Workshops*, 90(7):9–14, 2012. 2

[30] C. Yingying, W. Jinqiao, and L. Hanqing. Learning sharable models for robust background subtraction. *International Conference on Multimedia and Expo*, 2015. 6

[31] Z. Zhenjie and Z. Xuebo. Stacked multilayer self-organizing map for background modeling. *IEEE Transactions on Image Processing*, 24(9):2841–2850, 2015. 2, 3

[32] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006. 2