Efficient Online Spatio-Temporal Filtering for Video Event Detection

Xinchen Yan[1,2], Junsong Yuan[2], and Hui Liang[2]

[1]Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China, 200240
[2]School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore, 639798
skywalkeryxc@gmail.com, jsyuan@ntu.edu.sg, hliang1@e.ntu.edu.sg

Abstract. We propose a novel spatio-temporal filtering technique to improve the per-pixel prediction map, by leveraging the spatio-temporal smoothness of the video signal. Different from previous techniques that perform spatio-temporal filtering in an offline/batch mode, e.g., through graphical model, our filtering can be implemented online and in real-time, with provable lowest computational complexity. Moreover, it is compatible to any image analysis module that can produce per-pixel map of detection scores or multi-class prediction distributions. For each pixel, our filtering finds the optimal spatio-temporal trajectory in the past frames that has the maximum accumulated detection score. Pixels with small accumulated detection score will be treated as false alarm thus suppressed. To demonstrate the effectiveness of our online spatio-temporal filtering, we perform three video event tasks: salient action discovery, walking pedestrian detection, and sports event detection, all in an online/causal way. The experimental results on the three datasets demonstrate the excellent performances of our filtering scheme when compared with the state-of-the-art methods.

1 Introduction

Despite the success of object/event detection in images, it remains a challenging task to extend state-of-the-art image analysis techniques to streaming videos. It is not uncommon that the generated per-pixel prediction map becomes more noisy and unreliable due to the low quality of video data, e.g., illumination variations, motion blur, low resolution, not to mention the challenges caused by moving camera.

Instead of performing image analysis on each video frame independently, enforcing consistent labels among pixels over space and time has shown great improvement of the prediction map and avoids the "flickering" prediction in streaming videos [22, 20, 26]. Many existing spatio-temporal filtering methods [20, 26, 10, 23, 4, 35], only works in an offline/batch mode where the whole video is required to perform the spatio-temporal smoothing of prediction maps. In streaming video applications, however, we need to refine the prediction map by only using the previous detections without accessing future frames. As a result, offline filtering methods are not directly applicable. Therefore, we still lack efficient online spatio-temporal filtering schemes for video event detection.

Motivated by classic linear causal filtering, we propose a novel online spatio-temporal filtering method to improve the per-pixel prediction map of streaming videos. Suppose

the image analysis module can produce a per-pixel discriminative detection map for each video frame, e.g. positive value for positive class while negative value for negative class. For each pixel at the detection map, we search for its optimal spatio-temporal trajectory in the previous frames with maximum accumulated detection score, as illustrated in Figure 1. Pixel with small accumulated detection score will be treated as false alarm thus filtered. Pixel that miss detected can be retrieved if one can find a historical trajectory of high accumulated scores to support itself. Compared to classic linear causal filters, our proposed filtering method can adaptively choose the temporal window size to perform temporal filtering. Our spatio-temporal filtering can be easily extended to handle per-pixel prediction map where each pixel is associated with a multi-class probability distribution rather than a single discriminative score. It can also easily incorporate appearance modeling to improve detection results. The proposed filtering method is general enough for video event/object detection applications.

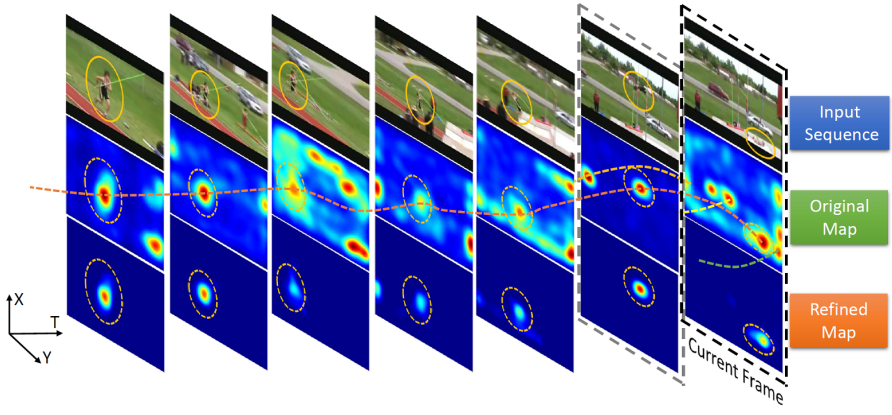

Fig. 1: Overview of our proposed spatio-temporal filtering, which refines the current prediction map by accumulating prediction score at previous frames.

In order to search for optimal spatio-temporal trajectories of each pixel with high efficiency, we further design an online dynamic programming algorithm that can achieve the goal with lowest time complexity and small memory cost. Instead of searching for the optimal trajectory per pixel, we propagate the accumulated score from one frame to another, with optimality guarantee. In practice, our online spatio-temporal filtering algorithm can run 67 frames per second for video of size 320×240, given that per-pixel detection map is available and parameters have been set.

To evaluate the effectiveness of our online spatio-temporal filtering, we perform three video event tasks: salient action discovery, walking pedestrian detection, and sports event detection. The excellent performances compared with the state-of-the-art methods validate the effectiveness and efficiency of the proposed spatio-temporal filtering method.

2 Related Work

Omniscient Spatio-Temporal Filtering. Omniscient approaches [26, 23, 4, 35, 1, 32, 7, 38, 30, 33, 39, 2, 11] take both past and future data into consideration. Andriluka *et al.* [1] introduced a hierarchical Gaussian process latent variable model to improve people-detection by people-tracklet detection in frames. Berclaz *et al.* [4] formulated multi-target tracking as a k-shortest node-disjoint paths problem and utilized disjoint path algorithm in calculation. Pirsiavash *et al.* [26] formulated multi-target tracking as a "spatio-temporal grouping" problem and proposed a near-optimal algorithm based on dynamic programming. Lan *et al.* [18] built a figure-centered model for joint action localization and categorization based on statistical scene context and structural representation of individual person. Tran *et al.* [35] formulated video event detection as a spatio-temporal path discovery problem. They proposed an OPD algorithm that searches the global optimal path that connects a sequence of regions with efficiency. Tran *et al.* [34] and Nataliya *et al.* [23] implemented structured output learning combined with spatio-temporal smoothing over entire video sequences.

Causal Spatio-Temporal Filtering. Causal approaches [22, 17, 25, 24, 42, 12, 6, 13, 14, 19, 29, 8] perform online spatio-temporal filtering relying on past data only. Kalman filtering had been used to aggregate data over time by Kim and Woods [17], and Patti *et al.* [25]. Paris *et al.* [24] derived the equivalent tool of mean-shift image segmentation for video streams based on ubiquitous use of the Gaussian kernel. Leibe *et al.* [19] introduced a non-Markovian hypothesis selection framework that searches globally optimal set of spatio-temporal trajectories for object detection. Grundmann *et al.* [12] proposed a hierarchical graph-based algorithm to segment long video sequences. Chen and Corso [6] introduced a Video Graph-Shifts approach for efficiently incorporating temporal consistency into MRF energy minimization framework. Hernández *et al.* [13] proposed a generic framework for object segmentation using depth maps based on Random Forests and Graph-cuts theory. Miksik *et al.* [22] designed a filtering algorithm to improve scene analysis by learning visual similarities across frames.

3 Online Spatio-Temporal Filtering

3.1 Problem Formulation

For each frame I_t of a video sequence S, we denote its per-pixel prediction map as M_t. For each pixel $\mathbf{p} = (x, y)$, let $M_t(\mathbf{p})$ denote the corresponding detection score. Without loss of generality, we assume the prediction map is discriminative, where $M_t(\mathbf{p}) \in [-1, 1]$. A positive score of $M_t(\mathbf{p})$ indicates a strong response at pixel $\mathbf{p}$ of the $t-$th frame, while a negative score stands for weak response.

As the prediction map M_t is generated independently per image, temporal consistency of pixel labels among video frames is not considered. Thus, it is necessary to avoid the "flickering" prediction issue by enforcing consistent labels among pixels over time. To denoise the prediction map, we believe the online spatio-temporal filtering should be able to achieve the following three goals.

1. it can suppress false alarm prediction of M_t based on previous prediction maps;
2. it can recover missing prediction of M_t based on previous prediction maps;
3. it can be performed online and implemented in real-time.

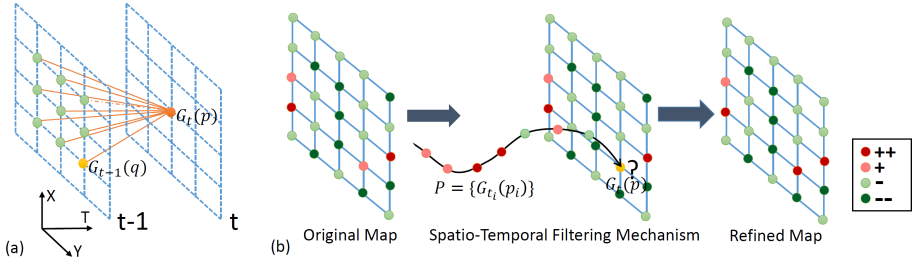

Fig. 2: Online Spatio-Temporal Filtering by refining prediction score of each pixel through finding a spaito-temporal path that has the maximum accumulated score. (a) spatio-temporal trellis. (b) our spatio-temporal filtering mechanism.

For the above three requirements, classic linear causal filters such as temporal mean filter may not provide satisfactory result as the prediction noise can be abrupt and non-Gaussian. To provide a better filtering, we design a novel non-linear filtering criterion. To help better explain our idea, we introduce a 3-dimensional trellis $W \times H \times T$ denoted by $\mathcal{G}$, where $W \times H$ is the frame size and T is the length of sequence. We denote each pixel $\mathbf{p}$ of the $t-$th frame as a vertex $\mathcal{G}_t(\mathbf{p})$. As shown in Figure 2, in the trellis, each vertex is connected to the spatial neighbors, e.g., 9-neighborhood in the previous frame. In general, we measure the distance between two pixels $\mathbf{p}$ and $\mathbf{q}$ by $\|\mathbf{p}-\mathbf{q}\|_\infty$, where $\|\mathbf{x}\|_\infty = \max\{|x_1|, |x_2|, \cdots, |x_d|\}$. For simplicity, $\|\mathbf{p}-\mathbf{q}\|_\infty$ is denoted as $\|\mathbf{p}-\mathbf{q}\|$. We define a spatio-temporal trajectory as a path $\mathcal{P}$, which connects a sequence of vertices $\mathcal{P} = \{\mathcal{G}_{t_i}(\mathbf{p}_i)\}$ in consecutive frames.

For each vertex $\mathcal{G}_t(\mathbf{p})$, we search for the maximum path in the past prediction maps that can generate the overall accumulated score. For each pixel, its maximum path score is denoted by $U_t(\mathbf{p})$. As the length of our path is adaptive (e.g. $1 \le s \le t$), we actually perform adaptive temporal filtering rather than performing temporal filtering with a fixed length window. For each pixel, its refined prediction map is the multiplication of the original prediction score and its maximum path score. Formally, the following definition explains how to obtain the refined prediction map from the original map.

$$
\begin{aligned}
U_t(\mathbf{p}) &= \max_{s\,:\,1\le s\le t} \left\{ \max_{\mathbf{p}_i:\|\mathbf{p}_i-\mathbf{p}_{i+1}\|\le R} \sum_{t_i=s}^{t-1} M_{t_i}(\mathbf{p}_i) \right\} + M_t(\mathbf{p}) \\
M'_t(\mathbf{p}) &= M_t(\mathbf{p}) \times U_t(\mathbf{p})
\end{aligned}
\tag{1}
$$

From Eq. 1, we can see that the accumulated detection score $U_t(\cdot)$ serves as the confidence weight that strengths or weakens original prediction score $M_t(\cdot)$. For example, the prediction $M_t(\mathbf{p})$ will be enhanced if there exists a spatio-temporal trajectory

$\mathcal{P} = \{\mathcal{G}_{t_i}(\mathbf{p}_i)\}$ ending at $\mathcal{G}_t(\mathbf{p})$ with large enough accumulated score. On the other hand, even if the pixel has a strong positive detection at the current location, if it is an isolated one and cannot find a historical path to support itself, the detection at that location will be treated as false alarm thus suppressed.

To verify the effectiveness of our filtering scheme, we simulate a video sequence of prediction maps $\{M_t\}$. The noise ϵ is generated first, in which the noise follows a Gaussian distribution $\mathcal{N}(\mu_\epsilon, {\sigma_\epsilon}^2)$. On top of the noise map, the object is generated as a sequence of bounding boxes centered at pixel $\mathbf{p}_t$ in the $t-$th frame. To incorporate the slow motion constraint, we require $\|\mathbf{p}_i - \mathbf{p}_{i+1}\| \leq R$ (R is an upper bound on spatial-temporal consistency). For each frame, pixels in the object bounding box are treated as the target and the values also follow Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Finally, by following Eq 1, we visualize the simulated prediction map, filtered map (mean) and our refined map in Figure 3. The mean filter refers to averaged score over fixed number of frames (10) for each pixel at the same location. It is worth noting that mean filter addresses short-term temporal consistency well but fails to take long-term temporal consistency into consideration (shown in Figure 3). Detailed configuration of the simulation has been included in supplementary materials.

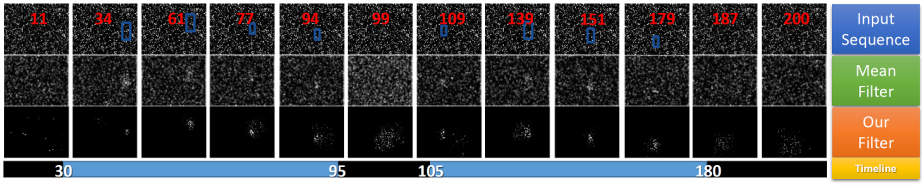

Fig. 3: Visualized snapshots of a simulated video sequence. The top row shows the groundtruth and frame number, in which the target exists from 30th frame to 180th frame and is temporally overwhelmed by noise from 95th frame to 105th frame. The second row and the third row compares results of mean filter and our filter. It is clear that our filter achieves better refined prediction map. More importantly, mean filter fails to recover the signal from 95th frame to 105th frame when signal is overwhelmed by noise.

From the simulation result, it shows that our filtering can suppress false alarm predictions. Since a false alarm prediction $M_t(\mathbf{p})$ is usually not supported by a high accumulated detection score $U_t(\mathbf{p})$ and hence will be suppressed. Similarly, the filtering criterion can address miss predictions. A missing prediction at pixel $\mathbf{p}$ is often featured by $0 < M_t(\mathbf{p}) < M_t(\mathbf{q})$, where $\mathbf{q}$ is another pixel with much lower accumulated detection score $U_t(\mathbf{q})$ compared to $U_t(\mathbf{p})$. Since refined prediction map is generated by multiplying M_t and U_t, it is likely $0 < M_t(\mathbf{q}) \times U_t(\mathbf{q}) < M_t(\mathbf{p}) \times U_t(\mathbf{p})$. Thus, the missing prediction in M_t will be recovered in the refined prediction map M'_t with high chance. In our formulation, we do not recover missing prediction when $M_t(\mathbf{p}) \leq 0$, since it may lead to false alarm prediction again.

3.2 Online Filtering Algorithm

Despite the effectiveness of our online spatio-temporal filtering, its implementation is a non-trivial issue. The search space of the spatio-temporal path for each pixels is $O(c^T)$, where c denotes the number of spatial neighbors. Following the idea of using dynamic programming, [35] and [26], we can reduce the cost of finding one path to $O(T)$, thus the cost of a whole frame is $O(W \times H \times T)$ as there are $W \times H$ pixels. Considering that we have in total T frames, the computational cost will be $O(W \times H \times T^2)$.

In order to provide real-time implementation, we propose an efficient online filtering algorithm that further reduces the complexity to obtain the accumulated map U_t. Instead of finding spatio-temporal path for each pixel separately, dynamic programming helps to avoid redundant calculation of subproblems (e.g. $U_{t-1}(\mathbf{p})$ and $U_{t-1}(\mathbf{q})$). The idea of our algorithm can be explained in the following Lemma.

Lemma 1. *$U_t(\mathbf{p})$ resulted from Eq. 2 is the maximum accumulated detection score tracing back from $\mathcal{G}_t(\mathbf{p})$.*

$$U_t(\mathbf{p}) = \max\left\{\max_{\|\mathbf{p}-\mathbf{q}\|\leq R} U_{t-1}(\mathbf{q}), 0\right\} + M_t(\mathbf{p}) \tag{2}$$

The correctness of Lemma 1 is proved in supplementary material. Intuitively, for $U_{t-1}(\mathbf{q}) < 0$, it cannot bring higher accumulated detection score in the future and hence will be neglected. On the contrary, for $U_{t-1}(\mathbf{q}) > 0$, we can obtain a higher accumulation score by adding it to the next frame. Our online filtering algorithm achieves the lowest time complexity $O(W \times H)$ for one step, and so it has an overall complexity $O(W \times H \times T)$ for the whole video. Based on the observation that calculation of $U_t(\mathbf{p})$ in Eq. 2 only relies on input $M_t(\mathbf{p})$ and accumulated detection score in previous frame $U_{t-1}(\mathbf{q})$, we then implement an iterative filtering algorithm with the memory cost $O(W \times H)$.

Algorithm 1 Online Filtering Algorithm

Calculate $M_1(\mathbf{p})$ based on input I_1;
$U_1(\mathbf{p}) \leftarrow M_1(\mathbf{p}), \forall \mathbf{p}$;
for $t \leftarrow 2$ to n **do**
  Calculate $M_t(\mathbf{p})$ based on input I_t;
  for all $\mathbf{p} \in [1, W] \times [1, H]$ **do**
    $\mathbf{q}' \leftarrow \text{argmax}_{\|\mathbf{p}-\mathbf{q}\|\leq R} U_{t-1}(\mathbf{q})$;
    $U_t(\mathbf{p}) \leftarrow \max\{U_{t-1}(\mathbf{q}'), 0\} + M_t(\mathbf{p})$;
  end for
  Release the space of $U_{t-1}(\cdot)$ and $M_{t-1}(\cdot)$;
  Calculate $\widetilde{U}_t(\cdot)$ based on $U_i(\cdot)$;
  for all $\mathbf{p} \in [1, W] \times [1, H]$ **do**
    $M'_t(\mathbf{p}) \leftarrow M_t(\mathbf{p}) \times \widetilde{U}_t(\mathbf{p})$;
  end for
end for

To ensure that $M'_t \in [-1, 1]$, accumulated map U_t will be normalized to $\widetilde{U}_t$ that satisfying $\widetilde{U}_t \in [0, 1]$. The technical details will be discussed in experiments. The radius R can be chosen by users to satisfy different requirements. When slow motion constraint can be satisfied, a smaller R is preferred. Otherwise, larger R is utilized to handle the effect of fast camera motion.

Multi-class label prediction. Our proposed method can be easily applied to multi-class pixel labeling. In such a case, each pixel has a prediction of multi-class distribution. Instead of working on a single prediction map, we can separate K different prediction maps $M_t(\cdot, k)$ for different classes, where K is the number of classes. Similarly, we accumulate detection scores separately via K different accumulated maps $U_t(\cdot, k)$.

3.3 Spatio-Temporal Filtering by Appearance Tracking

One limitation of our spatio-temporal filtering in Sec. 3.2 is that it only takes the prediction map into consideration while does not leverage extra information such as the appearance of the target. To mitigate this problem, we can incorporate tracking into our framework by adding the weight of the edge in the trellis. By measuring the visual similarity between two neighboring vertices in the trellis, we define $\mathcal{W}_{t_i}(\mathbf{p}_i, \mathbf{p}_{i+1})$ as the weight of the edge connecting two vertices.

Instead of only summing up vertex scores in a path, the accumulated score now sums all the edge scores and vertex scores. For each spatio-temporal trajectory $\mathcal{P} = \{\mathcal{G}_{t_i}(\mathbf{p}_i)\}$, we introduce an energy function $\mathcal{E}(\mathcal{P})$ that accumulates not only the detection scores, but also the tracking scores.

$$\mathcal{E}(\mathcal{P}) = -U(\mathcal{P}) - \lambda \cdot \mathcal{E}_a(\mathcal{P}) \tag{3}$$

The first term $U(\mathcal{P}) = \sum_{t_i=1}^{T} M_{t_i}(\mathbf{p}_i)$ represents the accumulated detection score while the second term $\mathcal{E}_a(P) = \sum_{t_i=1}^{T-1} \mathcal{W}_{t_i}(\mathbf{p}_i, \mathbf{p}_{i+1})$ stands for the accumulated visual similarity measure thus is the tracking score. Intuitively, the energy $\mathcal{E}(P)$ is minimized with presence of higher accumulated detection and appearance score, where λ is referred as *appearance weight* which will be further discussed in the experiments.

Lemma 2. *$\mathcal{E}_t(\mathbf{p})$ resulted from Eq. 4 is the minimum accumulated energy tracing back from $\mathcal{G}_t(\mathbf{p})$.*

$$\mathcal{E}_t(\mathbf{p}) = \begin{cases} -M_t(\mathbf{p}) & t = 1 \\ \min\left\{\min_{\|\mathbf{p}-\mathbf{q}\| \le R}\{\mathcal{E}_{t-1}(\mathbf{q}) - \lambda \cdot \mathcal{W}_t(\mathbf{q}, \mathbf{p})\}, 0\right\} - M_t(\mathbf{p}) & t > 1 \end{cases} \tag{4}$$

In order to calculate the minimum energy *tracing back from* $\mathcal{G}_t(\mathbf{p})$ (denoted by $\mathcal{E}_t(\mathbf{p})$) with high efficiency, we design a similar algorithm, as shown in Algorithm 2.

The correctness of Lemma 2 and Algorithm 2 have been proved in supplementary materials. In our implementation, we set $\mathcal{W}_{t_i}(\mathbf{p}_i, \mathbf{p}_{i+1}) = K(h_{t_i}(\mathbf{p}_i), h_{t_i}(\mathbf{p}_{i+1}))$, where $h_t(\mathbf{p})$ is intensity histogram of local patch centered at $\mathbf{p}$ at the $t-$th frame I_t. The similarity is measured by $K(h, h_*) = b - \|h - h_*\|$. Here, the bias b is to ensure discriminative score of appearance similarity, where histograms h and h_* should be normalized before calculation.

Algorithm 2 Energy Minimization

Calculate $M_1(\mathbf{p})$ and $h_1(\mathbf{p})$ based on input I_1;
$\mathcal{E}_1(\mathbf{p}) \leftarrow -M_1(\mathbf{p}), \forall \mathbf{p}$;
for $t \leftarrow 2$ to n **do**
  Calculate $M_t(\mathbf{p})$ and $h_t(\mathbf{p})$ based on input I_t;
  for all $\mathbf{p} \in [1, W] \times [1, H]$ **do**
    $\mathbf{q}' \leftarrow \operatorname{argmin}_{\|\mathbf{q}-\mathbf{p}\| \leq R} \{\mathcal{E}_{t-1}(\mathbf{q}) - \lambda \cdot K(h_{t-1}(\mathbf{q}), h_t(\mathbf{p}))\}$;
    $\mathcal{E}_t(\mathbf{p}) \leftarrow \mathcal{E}_{t-1}(\mathbf{q}') - \lambda \cdot K(h_{t-1}(\mathbf{q}'), h_t(\mathbf{p})) - M_t(\mathbf{p})$;
  end for
  Release the space of $\mathcal{E}_{t-1}(\cdot)$, $h_{t-1}(\cdot)$, and $M_{t-1}(\cdot)$;
end for

4 Experiments

The evaluation of our online filtering algorithm is composed of three different experiments. In Sec. 4.1, we show temporal consistency of video saliency can be utilized to discover video event in an unsupervised way. This helps to detect salient object in videos, which achieves higher accuracy in UCF101 Dataset [1] [31]. In Sec. 4.2, we illustrate that our appearance tracking version achieves superior performance when detecting actions (walking pedestrians) at UIUC-NTU Youtube Walking Dataset [2] [35]. In Sec. 4.3, we demonstrate that when combined with Exemplar-SVMs, our spatio-temporal filter can achieve excellent performances when localizing complex actions with large intra-class variations.

4.1 Unsupervised Video Event Discovery via Saliency Map Filtering

In this experiment, we show that video saliency maps can be improved by proposed filtering algorithm. In addition, the refined saliency map can be utilized to discover video event in an unsupervised way. Discovering events based on accumulated saliency parallels the theory that selective visual attention results from competition among multiple responses in visual cortex [36, 5].

Method. We leverage a Phase Discrepancy method [43] to generate motion saliency map. This method works well with slow moving background, but will output noisy saliency map when applied to fast moving background, changes in lighting condition, and camera zooming. Since values of saliency map ranges from 0 to 1, we therefore introduce a further step to generate discriminative prediction scores M_t. That is, M_t is normalized by $M_t \sim \mathcal{N}(0, 1)$. In this experiment, we fix the radius to $R = 3$ pixels.

We use UCF101 Dataset [31] for test, which provides fully annotated bounding boxes for 25 action categories. For this experiment, more than 2000 video sequences from 15 categories have been tested.

Results. We provide qualitative results of our filtering method first, as shown in Figure 4. The video sequences are quite challenging: saliency maps generated by baseline

[1] http://crcv.ucf.edu/data/UCF101.php
[2] http://www.cs.dartmouth.edu/~dutran/projects/event/

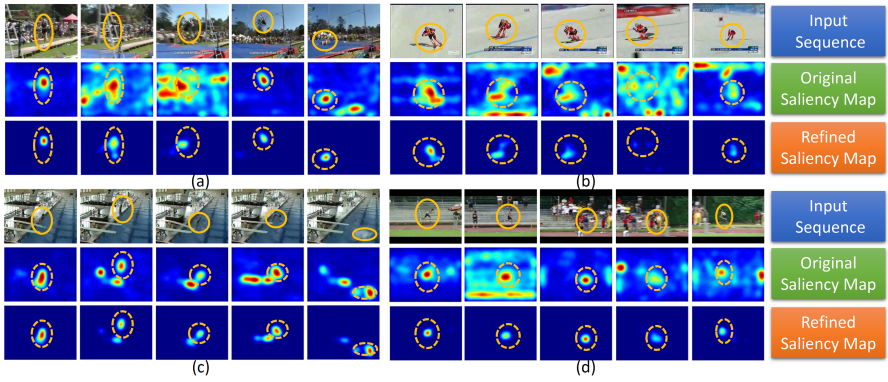

Fig. 4: Qualitative results on UCF101 Dataset. Four sequences are (a) "pole vault", (b) "skiing", (c) "diving", and (d) "long jump". In general, the four sequences represent four challenges in saliency map generation: (a) fast movement, (b) noisy saliency map, (b) several saliency regions, and (d) fast camera motion. For each video sequence, we visualize original saliency map as well as refined map obtained from our spatio-temporal filter. Note our filter is able to refine saliency map via long-term temporal consistency.

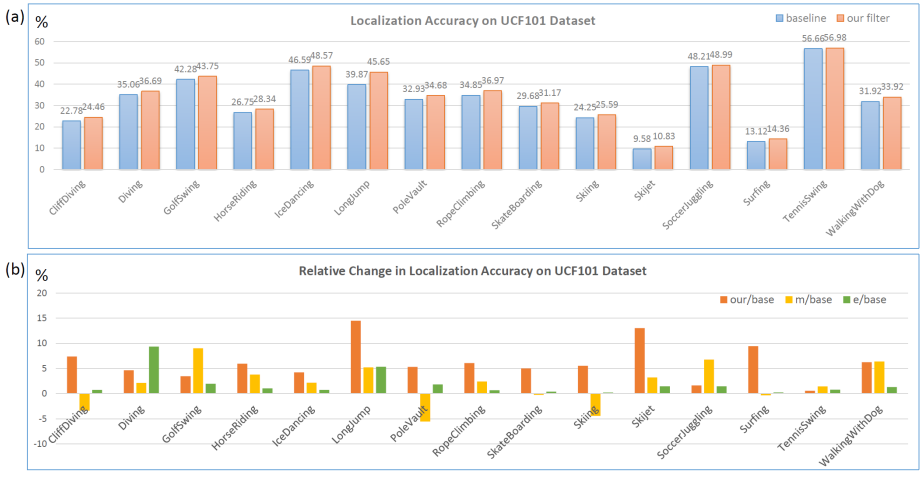

Fig. 5: Quantitative comparison of localization accuracy on UCF101 Dataset. (a) Comparison between our filter and baseline method. We use Phase Discrepancy method [43] as our baseline. On average, our filter improves about 2% of absolute accuracy compared to the baseline method. (b) Comparison in localization accuracy (%) among our filter, (temporal) mean filter, and (temporal) exponential filter. To highlight the difference, we utilize relative change as our measurement ("our/base" stands for "our filter vs. baseline method", "m/base" stands for "mean filter vs. baseline method", and "e/base" stands for "exponential filter vs. baseline method"). In this test, we fix the temporal window of size 10 for both mean filter and exponential filter.

method [43] are frequently overwhelmed by noise. In such situations, existing causal linear filters cannot work well, since their performance are largely determined by few previous frames. However, our filter deals with this difficulty by treating all previous frames equally and selectively accumulating scores. The refined saliency maps, are shown to have higher quality.

To further illustrate the strength of our saliency filter, we provide a qualitative comparison. Since this dataset does not provide groundtruth mask for motion saliency, we take a further step to localize salient object in videos. That is, we evaluate the saliency of a region Ω by summing up the discriminative scores at every pixel: $\sum_{\mathbf{p}\in\Omega} M_t(\mathbf{p})$.

We evaluate the localization accuracy in each frame by PASCAL metric (shown in Figure 5(a)). On average, our filter achieves 1.76% improvements in accuracy (6.20% of relative improvements). To show the difference between our filter and classic linear causal filters (e.g. mean filter and exponential filter), we also compare the relative change of localization accuracy (shown in Figure 5(b)). Although the quantitative evaluation of saliency filtering is conducted indirectly (via localization accuracy), improvements in localization accuracy still illustrate effectiveness of our filtering method to some extent. Note that performance of salient object detector varies between different action categories. For actions like "long jump" and "diving", in which actor moves horizontally or vertically, baseline detector is able to output valid saliency map. But actor's tiny scale in "cliff diving" and actor's fast movement in "pole vault" greatly affect the quality of saliency map. Even in such scenarios, our proposed filtering algorithm still contributes to accuracy improvements.

Finally, our online filtering algorithm can run at 67 fps with binary classification setting when input size is 320×240 pixels, radius is 3 using C++ Implementation, where the experiments are conducted on a computer with Intel(R) Core(TM) i5-4570 CPU and 8GB RAM. Note that we do not take into account I/O delays and the time for generating baseline saliency map.

4.2 Walking Pedestrian Detection

In this experiment, we show that our online spatio-temporal filtering can benefit object/action localization in video sequences by appearance modeling and tracking.

Method. Following [35], a walking pedestrian detector was previously trained by SVM with TUD-Motion Pairs [40] Dataset, while features of HOG [9], IMHd2, and Self-Similarity [37] are combined in the training step. We use UIUC-NTU Youtube Walking Dataset for testing, which consists of 27 video sequences (25 short videos of 100-150 frames and 2 long videos of 800-900 frames) of catwalk models. This data-set is challenging since catwalk models are frequently exposed to occlusions, changes in lighting conditions and cluttered background.

We first generate dense detection maps using our SVM classifier on 15 different scales. Then, we take detection maps as input for our spatio-temporal filtering. To localize walking pedestrian, we report our filtering result by the region $\Omega(\mathbf{p})$ centered at pixel $\mathbf{p}$ with maximum refined prediction score in each frame. To take scale changes into consideration, we modify the distance metric a little: $\|\mathbf{p}-\mathbf{q}\| + \|\delta_{\mathbf{p}} - \delta_{\mathbf{q}}\| \leq R$, where $\delta_{\mathbf{p}}$ corresponds to the scale order of region $\Omega(\mathbf{p})$. For our appearance tracking version, a simple color histogram h is used to calculate appearance score and we fix

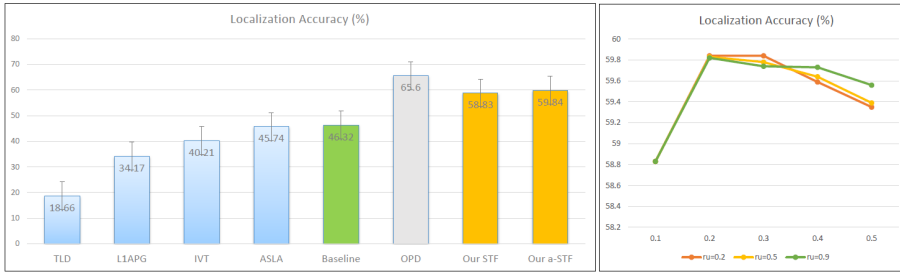

Fig. 6: Quantitative comparison on NTU-UIUC Youtube Walking Dataset. (a) Comparison of four online tracking algorithms, detection baseline, OPD algorithm and our spatio-temporal filtering method ("STF" for original version and "a-STF" for appearance tracking version). Our proposed filtering method improves 12-14% of accuracy compared to baseline detector and achieves superior performance over online tracking algorithms. (b) Averaged localization accuracy (%) of appearance tracking version over different *appearance weights* λ. The y-coordinate is localization accuracy, while the x-coordinate represents *appearance weight* λ. Three different *appearance adaptation ratios* r_u are used in the test.

radius R to 2. In addition, we introduce a ratio r_u in Eq. 5 to set appearance score dynamically for *appearance adaptation* and better tracking.

$$\begin{aligned} \mathcal{W}_{t_i}(\mathbf{p}_i, \mathbf{p}_{i+1}) &= K(\overline{H}_{t_i}(\mathbf{p}_i), h_{t_i}(\mathbf{p}_{i+1})) \\ \overline{H}_{t_{i+1}}(\mathbf{p}_{i+1}) &= r_u \cdot \overline{H}_{t_i}(\mathbf{p}_i) + (1 - r_u) \cdot h_{t_{i+1}}(\mathbf{p}_{i+1}) \end{aligned} \tag{5}$$

Such a strategy brings certain improvements in our test. However, achieving minimum energy $\min_{\mathbf{p}} \mathcal{E}_t(\mathbf{p})$ under Eq. 5 turns out to be intractable in polynomial time. We attempt to minimize the energy by updating dynamic appearance $\overline{H}_t$.

For comparison, detection baseline, OPD algorithm [35] as well as four online tracking algorithms (L1APG [3], ASLA [15], TLD [16], and IVT [28]) are utilized. We report the region with maximum prediction score in each frame as baseline detection result. Note that OPD algorithm utilizes both past and future prediction maps to localize walking pedestrian, which is an omniscient (non-causal) method. For each of four trackers, we report tracking accuracy based on maximum score under *SRE* defined in [41], so the result is comparatively independent of initialized bounding box.

Results. The evaluation is based on averaged localization accuracy of each frame under PASCAL metric. Figure 6(a) compares localization accuracy of four online tracking algorithms, baseline detector, OPD algorithm [35], and two versions of our spatio-temporal filtering method. In general, baseline detector fails to localize walking pedestrian due to challenging testing sequences with cluttered background. By utilizing temporal consistency, our filter can improve localization to about 12-14% compared to baseline result. For roughly one-third of video sequences, tracking algorithms fail to capture the target even in the first 30 frames due to catwalk models' fast walking. It is worth noting that OPD [35] achieves higher accuracy by taking both past and future data, while our filter is a causal method and can be implemented in real-time.

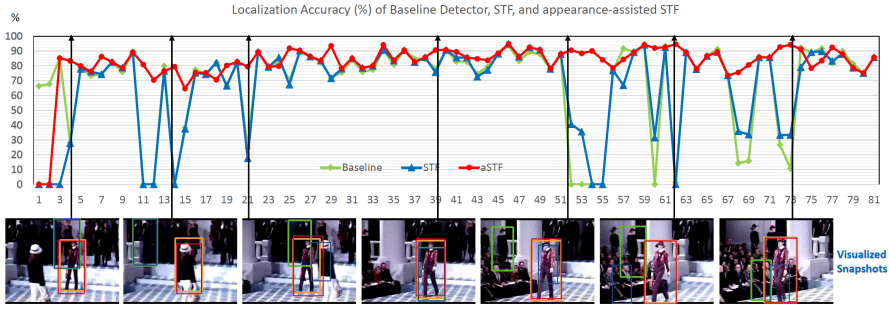

Fig. 7: Plots of localization accuracy of baseline detector, our spatio-temporal filtering method, and appearance tracking version with visualized snapshots. In the visualized snapshots, localization results are represented by bounding boxes of different colors ("orange" for groundtruth, "green" for baseline, "blue" for our filter, and "red" for our appearance tracking version).

Figure 6(b) illustrates the performance of proposed filtering method with different λ, the parameter to balance between detection score and visual similarity. When baseline detector outputs reasonable detection map, small λ is enough to refine detection map. But when baseline detector fails frequently to provide valid detection results, larger weight λ is preferred as appearance tracking can help throughout filtering process. For all 27 video sequences, our appearance tracking achieves better performance compared to our original version that does not utilize appearance tracking (shown in Figure 7).

4.3 UCF Sports Action Detection

In this experiment, we show that our filtering with Exemplar-SVMs [21] can beat state-of-the-art action detection approaches on more complex actions (with large intra-class variations).

Method. We evaluate our action detection method on the UCF Sports dataset, which consists of 150 video clips from broadcast television channels with 10 different action categories (e.g., "diving", "golf swinging", "kicking"). Two action categories, "diving" and "horse riding", are evaluated, since they are representative. For action "diving", athlete is moving fast vertically with various shapes and poses. But for all athletes, appearances are quite similar. For action "horse riding", athlete is moving horizontally with various appearances and poses.

Following the training-testing splits proposed by [18], we train our Exemplar-SVMs for each category using two-thirds of video frames and test on remaining one-third frames. TUD MotionPairs [40] is chosen as negative set, which consists of 196 image pairs taken from city district. To better adapt this negative set to our action detection task, we add 40 examples from Google search engine with similar background configuration as our positive samples. Intuitively, these additional negative samples enhance the generalization of our trained model. Half of training frames are used to train individual exemplar while remaining frames are utilized for calibration. In total, 270 exemplars are trained for action "diving" and 240 for action "horse riding" with HOG feature.

Action detection is based on the localization scheme. For each frame, the maximum score corresponded region is selected as our action detection results. For our filtering framework, we generate a collection of prediction maps with different aspect ratios and scales with interpolated discriminative scores based on sparse outputs of Exemplar-SVMs.

We compare our method with three state-of-the-art approaches. [18] implemented figure-centered model trained by latent SVM with HOG feature. [34] and [23] implemented structured output learning with both static feature HOG and motion feature HOF and HOMB [27] and both results are smoothed in a spatio-temporal scheme over entire video.

Results. As shown in Table 1, our method achieves more than 5% accuracy over Lan's and Tran's methods on two categories of UCF Sports Dataset and also approximates Nataliya's method when detecting "diving" action.

Note that [18] built a figure-centered model for each action, while [34] and [23] utilized motion feature like HOF and implemented spatio-temporal smoothing over entire video. However, our method filters the detection map with only HOG feature in use. We attribute the improved localization results to the unique training mechanism and our filtering method. More specifically, the Exemplar-SVMs help to address large intra-class variations in sports actions. When Exemplar-SVMs fail to work well, online spatio-temporal filtering can still improve the localization results a little bit (see Table 1).

Method / Category	Lan[18]	Tran[34]	Nataliya[23]	Exemplar	Exemplar+OSTF
Horse Riding	21.75	68.06	20.30	**73.59**	**73.59**
Diving	42.67	36.54	**52.37**	48.33	**50.19**

Table 1: Quantitative comparison of action localization accuracy on UCF Sports Dataset. The averaged localization accuracy is evaluated based on PASCAL metric (intersection divided by the union of detection and ground truth). The five columns are methods of [18], [34], [23], our Exemplar-based action detection, and our Exemplar-based action detection with online spatio-temporal refinement. In both categories, our final version (corresponding to the rightmost column) shows excellent performances when performin action localization.

As illustrated in Figure 8, our detection method performs better localization results compared to current localization methods. Note that [18] tends to miss detect target action and leads to false alarm detections due to background clutter. Also, the temporal consistency is not well addressed by its figure-centered model. Compared to Lan's method, [34] produces more stable detection results temporally. However, the bounding boxes produced by Tran's method are generally much smaller than the groundtruth bounding boxes. This can be contributed to its smoothing mechanism via maximum path algorithm. When smoothing in the spatio-temporal domain, only very confident regions are selected. Our method, however, is more robust to background clutter and intra-class variations due to our unique training mechanism. In the "horse riding" se-

quence, the target is never lost while in the "diving" sequence, the target is only temporally miss detected.

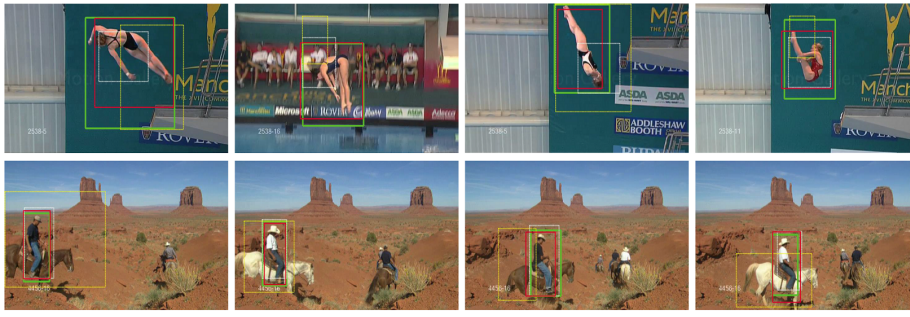

Fig. 8: Visualization of action localization of our Exemplar-based method, methods of [18] and [34]. In the visualized snapshots, localization results are represented by bounding boxes of different colors ("green" for groundtruth, "yellow" for Lan's method, "white" for Tran's method, and "red" for our Exemplar-based version (without online spatio-temporal filtering).

5 Conclusions

In this paper, we have introduced a novel spatio-temporal filtering method to improve per-pixel prediction map by accumulating detection score along spatio-temporal trajectories. To search maximum accumulated detection score, we have proposed an online filtering algorithm with the lowest time complexity and small memory cost. We have extended the online filtering algorithm to enable multi-channel processing and incorporate appearance information. With refined prediction map, we have shown that our method can benefit streaming video analysis tasks like human body recognition, saliency detection, and specific action detection.

Our online filtering algorithm only assumes the temporal dependence between video frames, with which many real-world video sequences share. As our filtering can perform in real-time, the refined prediction map can benefit other tasks like multiple object detection and tracking, which will be our future work.

6 Acknowledgment

The authors would like to thank Prof. Liqing Zhang for valuable discussions and thoughtful comments. This work was supported by Nanyang Assistant Professorship (SUG M4080134) and the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.

References

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR (2008)
2. Badrinarayanan, V., Budvytis, I., Cipolla, R.: Semi-supervised video segmentation using tree structured graphical models. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35(11), 2751–2764 (2013)
3. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR (2012)
4. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. PAMI 33(9), 1806–1819 (2011)
5. Borji, A., Itti, L.: State-of-the-art in visual attention modeling. PAMI 35(1), 185–207 (2013)
6. Chen, A.Y., Corso, J.J.: Temporally consistent multi-class video-object segmentation with the video graph-shifts algorithm. In: WACV (2011)
7. Choi, W., Pantofaru, C., Savarese, S.: A general framework for tracking multiple people from a moving camera. PAMI 35(7), 1577–1591 (2013)
8. Couprie, C., Farabet, C., LeCun, Y.: Causal graph-based video segmentation (2013)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
10. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV (2003)
11. Floros, G., Leibe, B.: Joint 2d-3d temporally consistent semantic segmentation of street scenes. In: CVPR (2012)
12. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR (2010)
13. Hernández-Vela, A., Zlateva, N., Marinov, A., Reyes, M., Radeva, P., Dimov, D., Escalera, S.: Graph cuts optimization for multi-limb human segmentation in depth maps. In: CVPR (2012)
14. Hoai, M., De la Torre, F.: Max-margin early event detectors. In: CVPR (2012)
15. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: CVPR (2012)
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. PAMI 34(7), 1409–1422 (2012)
17. Kim, J., Woods, J.W.: Spatio-temporal adaptive 3-d kalman filter for video. Image Processing, IEEE Trans. on 6(3), 414–424 (1997)
18. Lan, T., Wang, Y., Mori, G.: Discriminative figure-centric models for joint action localization and recognition. In: ICCV (2011)
19. Leibe, B., Schindler, K., Cornelis, N., Van Gool, L.: Coupled object detection and tracking from static cameras and moving vehicles. PAMI 30(10), 1683–1698 (2008)
20. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: CVPR (2011
21. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV (2011)
22. Miksik, O., Munoz, D., Bagnell, J.A., Hebert, M.: Efficient temporal consistency for streaming video scene analysis. In: ICRA (2013)
23. Nataliya, S., Michalis, R., Leonid, S., Greg, M.: Action is in the eye of the beholder: Eye-gaze driven model for spatio-temporal action localization. In: NIPS (2013)
24. Paris, S.: Edge-preserving smoothing and mean-shift segmentation of video streams. In: ECCV (2008)
25. Patti, A.J., Tekalp, A.M., Sezan, M.I.: A new motion-compensated reduced-order model kalman filter for space-varying restoration of progressive and interlaced video. Image Processing, IEEE Trans. on 7(4), 543–554 (1998)

26. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR (2011)
27. Raptis, M., Kokkinos, I., Soatto, S.: Discovering discriminative action parts from mid-level video representations. In: CVPR (2012)
28. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. IJCV 77(1-3), 125–141 (2008)
29. S. Hussain, R., Matthias, G., Irfan, E.: Geometric context from video
30. Sharma, P., Huang, C., Nevatia, R.: Unsupervised incremental learning for improved object detection in a video. In: CVPR (2012)
31. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild (2012)
32. Supancic III, J.S., Ramanan, D.: Self-paced learning for long-term tracking. In: CVPR (2013)
33. Tang, K., Ramanathan, V., Fei-Fei, L., Koller, D.: Shifting weights: Adapting object detectors from image to video. In: NIPS (2012)
34. Tran, D., Yuan, J.: Max-margin structured output regression for spatio-temporal action localization. In: NIPS (2012)
35. Tran, D., Yuan, J., Forsyth, D.: Video event detection: From subvolume localization to spatio-temporal path search. PAMI (2013)
36. Ungerleider, S.K., G, L.: Mechanisms of visual attention in the human cortex. Annual review of neuroscience 23(1), 315–341 (2000)
37. Walk, S., Majer, N., Schindler, K., Schiele, B.: New features and insights for pedestrian detection. In: CVPR (2010)
38. Wang, X., Hua, G., Han, T.X.: Detection by detections: Non-parametric detector adaptation for a video. In: CVPR (2012)
39. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: ECCV (2008)
40. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: CVPR (2009)
41. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR (2013)
42. Zhang, L., Tong, M.H., Cottrell, G.W.: Sunday: Saliency using natural statistics for dynamic analysis of scenes. In: Proceedings of the 31st Annual Cognitive Science Conference (2009)
43. Zhou, B., Hou, X., Zhang, L.: A phase discrepancy analysis of object motion. In: ACCV (2010)