Supplemental Material to Anomaly Detection and Localization in Crowded Scenes

APPENDIX A
MIXTURE OF DYNAMIC TEXTURES: LEARNING AND INFERENCE

A.1 The EM Algorithm for Learning Mixture of Dynamic Textures

Given a set of N independent and identically distributed (*i.i.d.*) samples $\mathcal{D}_i = \{\boldsymbol{x}^{(i)}\}_{i=1}^N$ (incomplete data), maximum likelihood estimates (MLE) of the parameters of an MDT $p(\boldsymbol{x}; \boldsymbol{\Theta})$ of K components

$$\begin{aligned}\boldsymbol{\Theta}^* &= \operatorname*{argmax}_{\boldsymbol{\Theta}} p(\mathcal{D}_i; \boldsymbol{\Theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{\Theta}} \sum_{i=1}^N \log p(\boldsymbol{x}^{(i)}; \boldsymbol{\Theta})\end{aligned} \tag{A.1}$$

are learned with the EM algorithm. There are two types of hidden variables in the MDT model: 1) the hidden state sequence $\boldsymbol{s}$ and 2) the assignment Z of each sequence to a mixture component. EM iterates between two steps

E-Step:

$$\mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k)}) = \mathbb{E}_{\mathcal{D}_h|\mathcal{D}_i;\boldsymbol{\Theta}^{(k)}}\Big[\log p(\mathcal{D}_c; \boldsymbol{\Theta})\Big], \tag{A.2}$$

M-Step:

$$\boldsymbol{\Theta}^{(k+1)} = \operatorname{argmax}_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k)}), \tag{A.3}$$

where the hidden data $\mathcal{D}_h$ consists of the hidden variables $\{\boldsymbol{s}^{(i)}\}_{i=1}^N$ and $\{z^{(i)}\}_{i=1}^N$, and the complete data $\mathcal{D}_c = \mathcal{D}_i \cup \mathcal{D}_h$. The assignment variable $z^{(i)}$ is represented by a vector $\mathbf{z}_i \in \{0,1\}^K$, such that $\mathbf{z}_{i,j} = 1$ if and only if $z^{(i)} = j$.

A.1.1 E-step

The log-likelihood of the complete data is (up to a constant)

$$\begin{aligned}\log p(\mathcal{D}_c; \boldsymbol{\Theta}) &= \sum_{i,j} \mathbf{z}_{i,j} \log \pi_j - \frac{1}{2}\sum_{i,j}\mathbf{z}_{i,j}\Bigg\{\log|S_j| \\ &+ \operatorname{Tr}\Big[S_j^{-1}\Big(P_{1,1}^{(i)} - 2\boldsymbol{s}_1^{(i)}\boldsymbol{\mu}_j^T + \boldsymbol{\mu}_j\boldsymbol{\mu}_j^T\Big)\Big]\Bigg\} \\ &- \frac{1}{2}\sum_{i,j}\mathbf{z}_{i,j}\Bigg\{(\tau-1)\log|Q_j| + \sum_{t=2}^{\tau}\operatorname{Tr}\Big[Q_j^{-1}\Big(P_{t,t}^{(i)} \\ &- 2P_{t,t-1}^{(i)}A_j^T + A_j P_{t-1,t-1}^{(i)} A_j^T\Big)\Big]\Bigg\} \\ &- \frac{1}{2}\sum_{i,j}\mathbf{z}_{i,j}\Bigg\{\tau\log|R_j| + \sum_{t=1}^{\tau}\operatorname{Tr}\Big[R_j^{-1}\Big(\boldsymbol{x}_t^{(i)}\boldsymbol{x}_t^{(i)T} \\ &- 2\boldsymbol{x}_t^{(i)}(\boldsymbol{s}_t^{(i)})^T C_j^T + C_j P_{t,t}^{(i)} C_j^T\Big)\Big]\Bigg\},\end{aligned} \tag{A.4}$$

where $P_{t,r}^{(i)} = \boldsymbol{s}_t^{(i)}(\boldsymbol{s}_r^{(i)})^T$. The $\mathcal{Q}$-function is then

$$\begin{aligned}&\mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k-1)}) = \\ &\quad \sum_j \hat{N}_j\Big[\log\pi_j - \frac{1}{2}\log|S_j| - \frac{\tau-1}{2}\log|Q_j| - \frac{\tau}{2}\log|R_j|\Big] \\ &\quad - \frac{1}{2}\sum_j \operatorname{Tr}\Big[S_j^{-1}\Big(\eta_j - \xi_j\boldsymbol{\mu}_j^T - \boldsymbol{\mu}_j\xi_j^T + \hat{N}_j\boldsymbol{\mu}_j\boldsymbol{\mu}_j^T\Big)\Big] \\ &\quad - \frac{1}{2}\sum_j \operatorname{Tr}\Big[Q_j^{-1}\Big(\varphi_j - \Psi_j A_j^T - A_j\Psi_j^T + A_j\phi_j A_j^T\Big)\Big] \\ &\quad - \frac{1}{2}\sum_j \operatorname{Tr}\Big[R_j^{-1}\Big(\Lambda_j - \Gamma_j C_j^T - C_j\Gamma_j^T + C_j\Phi_j C_j^T\Big)\Big],\end{aligned} \tag{A.5}$$

with

$$\begin{aligned}&\hat{N}_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}, && \Phi_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\sum_{t=1}^{\tau}\hat{P}_{t,t|j}^{(i)}, \\ &\xi_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\hat{\mathbf{s}}_{1|j}^{(i)}, && \varphi_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\sum_{t=2}^{\tau}\hat{P}_{t,t|j}^{(i)}, \\ &\eta_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\hat{P}_{1,1|j}^{(i)}, && \phi_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\sum_{t=2}^{\tau}\hat{P}_{t-1,t-1|j}^{(i)}, \\ &\Psi_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\sum_{t=2}^{\tau}\hat{P}_{t,t-1|j}^{(i)}, \\ &\Lambda_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\sum_{t=1}^{\tau}\boldsymbol{x}_t^{(i)}(\boldsymbol{x}_t^{(i)})^T, \\ &\Gamma_j = \textstyle\sum_i \hat{\mathbf{z}}_{i,j}\sum_{t=1}^{\tau}\boldsymbol{x}_t^{(i)}(\hat{\mathbf{s}}_{t|j}^{(i)})^T,\end{aligned} \tag{A.6}$$

where

$$\begin{aligned}\hat{\mathbf{z}}_{i,j} &= p(z^{(i)} = j|\boldsymbol{x}^{(i)}; \boldsymbol{\Theta}^{(k-1)}) \\ &= \frac{\pi_j p(\boldsymbol{x}^{(i)}|z^{(i)} = j; \boldsymbol{\Theta}^{(k-1)})}{\sum_{k=1}^{K} \pi_k p(\boldsymbol{x}^{(i)}|z^{(i)} = k; \boldsymbol{\Theta}^{(k-1)})}\end{aligned} \quad \text{(A.7)}$$

are the posterior assignment probabilities, which can be calculated as discussed in Appendix A.3; and the statistics are aggregates of the expectations

$$\hat{\boldsymbol{s}}_{t|j}^{(i)} = \mathbb{E}_{\boldsymbol{s}^{(i)}|\boldsymbol{x}^{(i)}, z^{(i)}=j; \boldsymbol{\Theta}^{(k-1)}} \left(\boldsymbol{s}_t^{(i)}\right), \quad \text{(A.8)}$$

$$\hat{P}_{t,r|j}^{(i)} = \mathbb{E}_{\boldsymbol{s}^{(i)}|\boldsymbol{x}^{(i)}, z^{(i)}=j; \boldsymbol{\Theta}^{(k-1)}} \left(P_{t,r}^{(i)}\right). \quad \text{(A.9)}$$

The conditional expectation of (A.8) and (A.9) can be efficiently computed via the Kalman smoothing filter [**?**] (see Appendix A.2), *e.g.*, $\hat{P}_{t,r|j}^{(i)} = \hat{V}_{t,r|j}^{(i)} + \hat{\boldsymbol{s}}_{t|j}^{(i)}(\hat{\boldsymbol{s}}_{r|j}^{(i)})^T$, where

$$\hat{V}_{t,r|j}^{(i)} = \mathrm{cov}_{\boldsymbol{s}^{(i)}|\boldsymbol{x}^{(i)}, z^{(i)}=j; \boldsymbol{\Theta}^{(k-1)}} (\boldsymbol{s}_t^{(i)}, \boldsymbol{s}_r^{(i)}) \text{(A.10)}$$

for $r = t, t-1$.

A.1.2 M-step

In the M-step, the new parameter estimate $\boldsymbol{\Theta}^{(k)}$ is computed by maximizing the $\mathcal{Q}$-function: $\boldsymbol{\Theta}^{(k)} = \mathrm{argmax}_{\boldsymbol{\Theta}}\, \mathcal{Q}(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(k-1)})$. This leads to the updates

$$\begin{aligned}&A_j^{(k)} = \Psi_j(\phi_j)^{-1}, Q_j^{(k)} = \tfrac{1}{(\tau-1)\hat{N}_j}\left(\varphi_j - A_j^{(k)}\Psi_j^T\right), \\ &C_j^{(k)} = \Gamma_j(\Phi_j)^{-1}, R_j^{(k)} = \tfrac{1}{\tau \hat{N}_j}\left(\Lambda_j - C_j^{(k)}\Gamma_j\right), \\ &\boldsymbol{\mu}_j^{(k)} = \tfrac{1}{\hat{N}_j}\xi_j, \qquad S_j^{(k)} = \tfrac{1}{\hat{N}_j}\eta_j - \boldsymbol{\mu}_j^{(k)}(\boldsymbol{\mu}_j^{(k)})^T, \\ &\pi_j^{(k)} = \tfrac{\hat{N}_j}{N}.\end{aligned} \quad \text{(A.11)}$$

A.2 Kalman Smoothing Filter

The mean and covariance of the state sequence $\{\boldsymbol{s}_t\}_{t=1}^{\tau}$, conditioned on the entire observed sequence $\{\boldsymbol{x}_t\}_{t=1}^{\tau}$, can be estimated by the Kalman smoothing filter [**?**], [**?**]. Defining the expectations conditioned on the observed sequence from time $t = 1$ to $t = r$ as

$$\hat{\boldsymbol{s}}_t^r = \mathbb{E}_{\boldsymbol{s}_t|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r}\left[\boldsymbol{s}_t\right], \quad \text{(A.12)}$$

$$\hat{V}_{t,k}^r = \mathbb{E}_{\boldsymbol{s}_t|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_r}\left[(\boldsymbol{s}_t - \hat{\boldsymbol{s}}_t^r)(\boldsymbol{s}_k - \hat{\boldsymbol{s}}_k^r)^T\right], \text{(A.13)}$$

the estimates are calculated with the following recursion: for $t = 1, \ldots, \tau$, compute

$$\hat{V}_{t,t}^{t-1} = A\hat{V}_{t-1,t-1}^{t-1}A^T + Q, \quad \text{(A.14)}$$

$$K_t = \hat{V}_t^{t-1}C^T(C\hat{V}_{t,t}^{t-1}C^T + R)^{-1} \quad \text{(A.15)}$$

$$\hat{V}_{t,t}^t = \hat{V}_{t,t}^{t-1} - K_tC\hat{V}_{t,t}^{t-1} \quad \text{(A.16)}$$

$$\hat{\boldsymbol{s}}_t^{t-1} = A\hat{\boldsymbol{s}}_{t-1}^{t-1} \quad \text{(A.17)}$$

$$\hat{\boldsymbol{s}}_t^t = \hat{\boldsymbol{s}}_t^{t-1} + K_t(\boldsymbol{x}_t - C\hat{\boldsymbol{s}}_t^{t-1}), \quad \text{(A.18)}$$

where the initial conditions are $\hat{\boldsymbol{s}}_1^0 = \boldsymbol{\mu}$ and $\hat{V}_1^0 = S$. The estimates $\hat{\boldsymbol{x}}_t^{\tau}$, $\hat{V}_{t,t}^{\tau}$ and $\hat{V}_{t,t-1}^{\tau}$ are then computed with the following backward recursion: for $t = \tau, \ldots, 1$,

$$J_{t-1} = \hat{V}_{t-1,t-1}^{t-1}A^T(\hat{V}_{t,t}^{t-1})^{-1} \quad \text{(A.19)}$$

$$\hat{\boldsymbol{s}}_{t-1}^{\tau} = \hat{\boldsymbol{s}}_{t-1}^{t-1} + J_{t-1}(\hat{\boldsymbol{s}}_t^{\tau} - A\hat{\boldsymbol{s}}_{t-1}^{t-1}) \quad \text{(A.20)}$$

$$\hat{V}_{t-1,t-1}^{\tau} = \hat{V}_{t-1,t-1}^{t-1} + J_{t-1}(\hat{V}_{t,t}^{\tau} - \hat{V}_{t,t}^{t-1})J_{t-1}^T, \text{(A.21)}$$

and for $t = \tau, \ldots, 2$,

$$\begin{aligned}\hat{V}_{t-1,t-2}^{\tau} = \hat{V}_{t-1,t-1}^{t-1}&J_{t-2}^T \\ &+ J_{t-1}(\hat{V}_{t,t-1}^{\tau} - A\hat{V}_{t-1,t-1}^{t-1})J_{t-2}^T\end{aligned} \quad \text{(A.22)}$$

with initial condition $\hat{V}_{\tau,\tau-1}^{\tau} = (I - K_{\tau}C)A\hat{V}_{\tau-1,\tau-1}^{\tau-1}$.

A.3 Probabilistic Models for Dynamic Textures

The conditional distribution $p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1})$ of the DT of (2) is

$$\begin{aligned}p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) &= \mathcal{N}(\boldsymbol{s}_t; A\boldsymbol{s}_{t-1}, Q) \\ &= \frac{1}{\sqrt{(2\pi)^n|Q|}}\exp\left\{-\frac{1}{2}\left\|\boldsymbol{s}_t - A\boldsymbol{s}_{t-1}\right\|_Q^2\right\}.\end{aligned} \quad \text{(A.23)}$$

Using the Markov property, for Gaussian initial conditions $\boldsymbol{s}_1 \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$, the distribution of state sequence $\boldsymbol{s}_1^{\tau} = \begin{bmatrix} \boldsymbol{s}_1^T, & \cdots, & \boldsymbol{s}_{\tau}^T \end{bmatrix}^T$ is Gaussian

$$p(\boldsymbol{s}_1^{\tau}) = p(\boldsymbol{s}_1)\prod_{t=2}^{\tau} p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}) = \mathcal{N}(\boldsymbol{s}_1^{\tau}; \boldsymbol{\mu}_1^{\tau}, \Sigma). \quad \text{(A.24)}$$

Since

$$\boldsymbol{s}_t = A^{t-1}\boldsymbol{s}_1 + \sum_{i=1}^{t-1} A^{t-1-i}\boldsymbol{n}_i, \quad \text{(A.25)}$$

it follows that

$$\boldsymbol{\mu}_t = \mathbb{E}\left[\boldsymbol{s}_t\right] = A^{t-1}\boldsymbol{\mu}_1, \quad \text{(A.26)}$$

$$\begin{aligned}\Sigma_{n,m} &= \mathbb{E}\left[(\boldsymbol{s}_n - \boldsymbol{\mu}_n)(\boldsymbol{s}_m - \boldsymbol{\mu}_m)^T\right] \\ &= A^{n-1}S(A^{m-1})^T + \sum_{i=1}^{k-1} A^{n-1-i}Q(A^{m-1-i})^T,\end{aligned} \quad \text{(A.27)}$$

where $k = \min(n, m)$. The covariance can be computed recursively, given $\Sigma_{1,1} = S$,

$$\Sigma_{n,m} = \begin{cases} (A^{m-n}\Sigma_{n,n})^T, & \text{if } m > n, \\ A(\Sigma_{n-1,n-1})A^T + Q, & \text{if } m = n, \\ A^{n-m}\Sigma_{m,m}, & \text{if } m < n. \end{cases} \quad \text{(A.28)}$$

Hence, the sufficient statistics of $p(\boldsymbol{s}_1^{\tau})$ are

$$\boldsymbol{\mu}_1^{\tau} = \begin{bmatrix} \boldsymbol{\mu}_1^T, & \cdots, & \boldsymbol{\mu}_{\tau}^T \end{bmatrix}^T \quad \text{(A.29)}$$

and

$$\Sigma = \begin{bmatrix} \Sigma_{1,1} & (A\Sigma_{1,1})^T & \cdots & (A^{\tau-1}\Sigma_{1,1})^T \\ A\Sigma_{1,1} & \Sigma_{2,2} & \cdots & (A^{\tau-2}\Sigma_{2,2})^T \\ \vdots & \vdots & \ddots & \vdots \\ A^{\tau-1}\Sigma_{1,1} & A^{\tau-2}\Sigma_{2,2} & \cdots & \Sigma_{\tau,\tau} \end{bmatrix}. \text{(A.30)}$$

Since the image sequence $\boldsymbol{x}_1^{\tau}$ is a linear transformation of the state sequence $\boldsymbol{s}_1^{\tau}$, it has distribution

$$p(\boldsymbol{x}_1^{\tau}) = \mathcal{N}(\boldsymbol{x}_1^{\tau}; \tilde{\gamma}, \tilde{\Phi}), \quad \text{(A.31)}$$

where $\tilde{\gamma} = \tilde{C}\boldsymbol{\mu}_1^{\tau}$ and $\tilde{\Phi} = \tilde{C}\Sigma\tilde{C}^T + \tilde{R}$, and $\tilde{C}$ and $\tilde{R}$ are block diagonal matrices where each diagonal block is equal to C and R respectively.

APPENDIX B
ALGORITHMS

In this appendix, we summarize the algorithms used to compute the hierarchical spatial anomaly maps (Algorithm 1) of Section 4.4 in the main manuscript, the inference procedure of the proposed CRF filter (Algorithm 2 and Algorithm 3) discussed in Section 5.2.2 and the procedure used to predict anomalies by combining hierarchical anomaly maps and the CRF filter (Algorithm 4), discussed in the same section.

Algorithm 1: spatial_anomaly

Input : a video $\boldsymbol{x}$, a set of frames $\{t_i\}_{i=1}^{n_f}$, observation sites S, multi-scale spatial supports $\{\mathcal{R}^k\}_{k=1}^{L}$
Output: spatial anomaly maps $\{\mathcal{S}^k\}_{k=1}^{L}$ with multi-scale spatial surrounds

foreach *frame* t_i **do**
  extract spatio-temporal patches around t_i:
    $\{pat_i\} =$ *extract_patch*$(\boldsymbol{x}(t_i))$;
  learn a MDT: $\{DT_i\} =$ *clustering*$(\{pat_i\})$ with (A.6) and (A.11);
  compute intra-component KLs:
    $\mathrm{KL}(i,j) = \mathrm{KL}(DT_i, DT_j),\ \forall i \neq j$, with (9);
  foreach *observation site* $\mathsf{S}_j^{t_i}$ *in* t_i **do**
    foreach *spatial support with the size of* $\mathcal{R}_{i,j}^k$ **do**
      compute spatial anomaly $\mathcal{S}^k(i,j)$ using $\mathrm{KL}(i,j)$ and the segmentation maps of (4) (8) and (10);
    end
  end
end

Algorithm 2: sample_label_field

Input : previous prediction $\{\boldsymbol{y}^{(t)}\}_{t=1}^{\tau-1}$; initialization $\boldsymbol{y}_0$; number of iterations T.
Output: predicted label field $\boldsymbol{y}'$.

$\boldsymbol{y}' \leftarrow \boldsymbol{y}_0$;
for $i \leftarrow 1$ **to** T **do**
  foreach $j \in \mathsf{S}^{(\tau)}$ **do**
    draw y'_j from
      $y'_j \backsim p(y_j | \{\boldsymbol{x}^{(t)}\}_{t=1}^{\tau}, \{\boldsymbol{y}^{(t)}\}_{t=1}^{\tau-1}, \boldsymbol{y}'_{-j}; \boldsymbol{\Theta})$,
    using (23)-(25) and (12)-(15), where
    $\boldsymbol{y}'_{-j} = \{y'_1, \cdots, y'_{j-1}, y'_{j+1}, \cdots, y'_{|\mathsf{S}|}\}$;
  end
end

Algorithm 3: CRF_inference

Input : previous prediction $\{\boldsymbol{y}^{(t)}\}_{t=1}^{\tau-1}$; initialization $\boldsymbol{y}_0 = \{y_{0,1}, \cdots, y_{0,|\mathsf{S}|}\}$; observation up to current frame τ $\{\boldsymbol{x}^{(t)}\}_{t=1}^{\tau}$, cooling time T_c, sampling period T_s, number of samples N_s, threshold γ.
Output: predicted anomaly labels for current frame $\boldsymbol{y}^{(\tau)}$.

$\boldsymbol{y} \leftarrow \boldsymbol{0}$, $\boldsymbol{y}' \leftarrow$ *sample_label_field*$(\{\boldsymbol{y}^{(t)}\}_{t=1}^{\tau-1}, \boldsymbol{y}_0, T_c)$;
for $n \leftarrow 1$ **to** N_s **do**
  $\boldsymbol{y}' \leftarrow$ *sample_label_field*$(\{\boldsymbol{y}^{(t)}\}_{t=1}^{\tau-1}, \boldsymbol{y}', T_s)$;
  $\boldsymbol{y} \leftarrow \boldsymbol{y} + \frac{1}{N_s}\boldsymbol{y}'$;
end
$\boldsymbol{y}^{(\tau)} = I(\boldsymbol{y} \geqslant \gamma \boldsymbol{1})$, where $I(\cdot)$ is the element-wise indicator function.

Algorithm 4: anomaly_detector

Input : a query video clip $\boldsymbol{x}$, a set of frames $\{t_i\}_{i=1}^{n_f}$, observation sites S, CRF filter $\boldsymbol{\Theta}$, multi-scale spatial supports $\{\{\mathcal{R}_i^k\}_{i=1}^{n_k}\}_{k=1}^{L}$ and associated temporal MDTs $\{\{\boldsymbol{M}_i^{(k)}\}_{i=1}^{n_k}\}_{k=1}^{L}$.
Parameter: cooling time T_c, sampling period T_s, number of samples N_s, threshold γ.
Output : predicted anomaly labels for each site of each frame $\{\boldsymbol{y}^{(t_i)}\}_{i=1}^{n_{kf}}$.

foreach *frame* t_i **do**
  compute spatial anomaly maps:
  $\{\mathcal{S}^k\}_{k=1}^{L} \leftarrow$ *spatial_anomaly(*$\boldsymbol{x}$, $\{t_i\}$, S, $\{\mathcal{R}\}$*)*;
  foreach *observation site* S_j **do**
    foreach *spatial scale* k **do**
      compute hidden state sequence $\boldsymbol{s}_1^{\tau}$ for each MDT component using (A.12)-(A.22);
      compute temporal anomaly maps $\mathcal{T}_j^k$ with (3) and (A.24);
    end
  end
  draw the initial label field by logistic regression, using (22): $\boldsymbol{y}_{(0)}^{(t_i)} \sim p(\boldsymbol{y}|\boldsymbol{x}^{(t_i)}; \boldsymbol{w})$;
  infer labels using $\boldsymbol{y}_{(0)}^{(t_i)}$ as the starting point:
  $\boldsymbol{y}^{(t_i)} \leftarrow$ *CRF_inference(*$\{\mathcal{S}^k\}$, $\{\mathcal{T}^k\}$, $\{\boldsymbol{y}^{(t_j)}\}_{(j<i)}$, $\boldsymbol{y}_{(0)}^{(t_i)}$, T_c, T_s, N_s, γ*)*;
end

APPENDIX C
EXPERIMENT

C.1 Descriptor Comparison

In this appendix, we present ROC curves corresponding to the comparisons of Table 2 and Table 3. These provide a more detailed picture of the results presented in the tables and may be useful for performance comparison with future methods. Fig. C.2 presents ROC curves for the various descriptors of Table 2. Fig. C.3 presents ROC curves for the filters of Table 3. In general, the ROC figures confirm the conclusions derived from the tables.

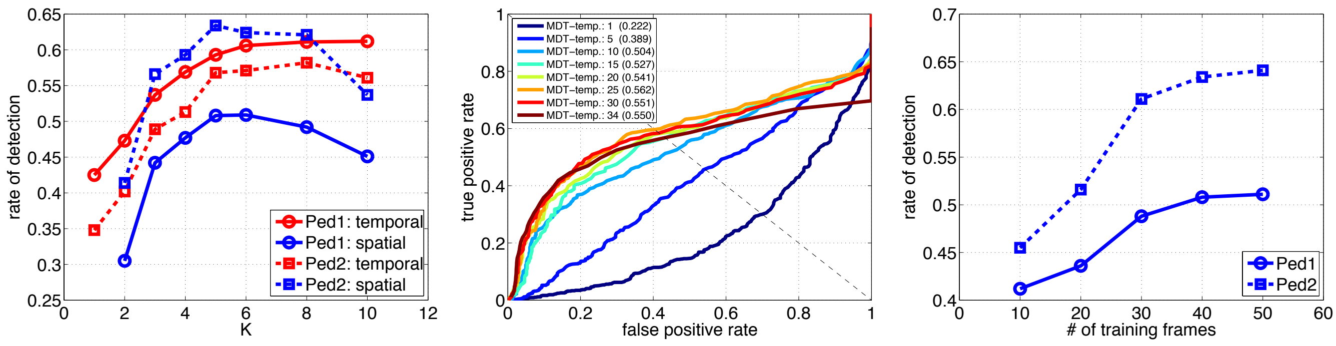

Fig. C.1. Anomaly detection performance (pixel-level criterion) on Ped1. Left: RD *v.s.* number of MDT components. Middle: ROC curves of temporal anomaly detection *v.s.* number of training clips. Right: RD of spatial anomaly detection *v.s.* number of frames used for segmentation.

C.2 Parameter Sensitivity

The performance of the proposed anomaly detector depends on a few parameters. These include descriptor settings, such as the number of MDT components, and the size of the training data. In particular, temporal anomaly maps degrade when MDTs are learned from small training samples. Spatial anomaly detection is more flexible, as it has no memory. This is illustrated in the second row of Fig. 6, for an anomalous cart at the bottom of the frame. Since no events have been previously observed in this region, there is no training data for the temporal MDTs. Some time is thus required to learn these models, and the temporal map does not capture this anomaly. This is unlike the spatial map, where the cart is robustly detected. On the other hand, the segmentation required for spatial anomaly detection can be computationally more intensive than the detection of temporal anomalies, depending on how many video frames it requires.

Several experiments were performed to evaluate the impact of these parameters on anomaly detection accuracy. Fig. C.1 characterizes the performance of one-layer temporal/spatial anomaly detection under different parameter settings. The figure on the left shows that both temporal and spatial anomaly detection improve with the number of DT components, with best performance for $K \in \{5, 6, 8\}$. Note, in particular, the significant improvement over the DT ($K = 1$). Above $K = 8$ there is some potential for overfitting and performance can degrade. Since more components imply more computation, we use $K = 5$ in all our experiments. The center figure presents ROC curves for temporal anomalies on Ped1, as a function of the number of MDT training clips. Performance increases quickly from 1 to 15 clips (200 to 3000 frames), saturating after 25. The right figure characterizes the trade-off between the efficiency and accuracy of spatial anomaly detection, as more frames are considered in the segmentation process. Increasing the number of frames from 10 to 40 improves RD by more than 10%. Beyond that, performance saturates.

C.3 Error Analysis

In this appendix, we briefly discuss the errors made by the different detector components. Some of these turned out to be mislabeled instances on the two datasets. For example, the first column of Fig. C.4 depicts a pedestrian that suddenly redirects her route, unexpectedly moving across the walkway. Similarly, the third column of the figure depicts a pedestrian who takes a very unorthodox route, so as to clear the way for an incoming cart. It is, in our opinion, positive that the detector flags these events, demonstrating ability to detect subtle anomalies that would not even be necessarily detected by a human without close inspection. This also confirms the well known fact that anomalies are, by definition, difficult to define a priori.

A second type of false-positives, which are technically incorrect detections, arise due to normal events that are either unusual or occur in unusual scenes. For example, in the second column of Fig. C.4, a person walking leftwards at the bottom of the scene is identified as an anomaly by the temporal detector. This is because the overwhelming majority of the training events in this region are of vertically moving pedestrians (the south-north walkway leads to a much busier area of the campus than the east-west one). A more careful training set collection, using standard bootstrap procedures [**?**], would eliminate these false positives. With regards to spatial anomalies, unusually sparse scenes can be a source of concern. For example, in the fourth column of Fig. C.4, a pedestrian entering a very sparsely populated walkway is denoted a spatial anomaly. These errors are not very serious, since spatial anomaly detection could simply be disabled for sparse scenes, or the anomaly detector could be complemented by vision techniques that perform well in these scenes (*e.g.*, pedestrian detection).

More problematic are errors due to pedestrians that move "against the flow" of the surrounding crow. This is the case of the fifth column of Fig. C.4, where a left moving pedestrian enters the walkway when all other pedestrians are moving right. This behavior could be considered anomalous in some cases but not in others, depending on the scene context. For example, if the crowd was fleeing from a dangerous occurrence on the left of the scene (*e.g.*, fire)

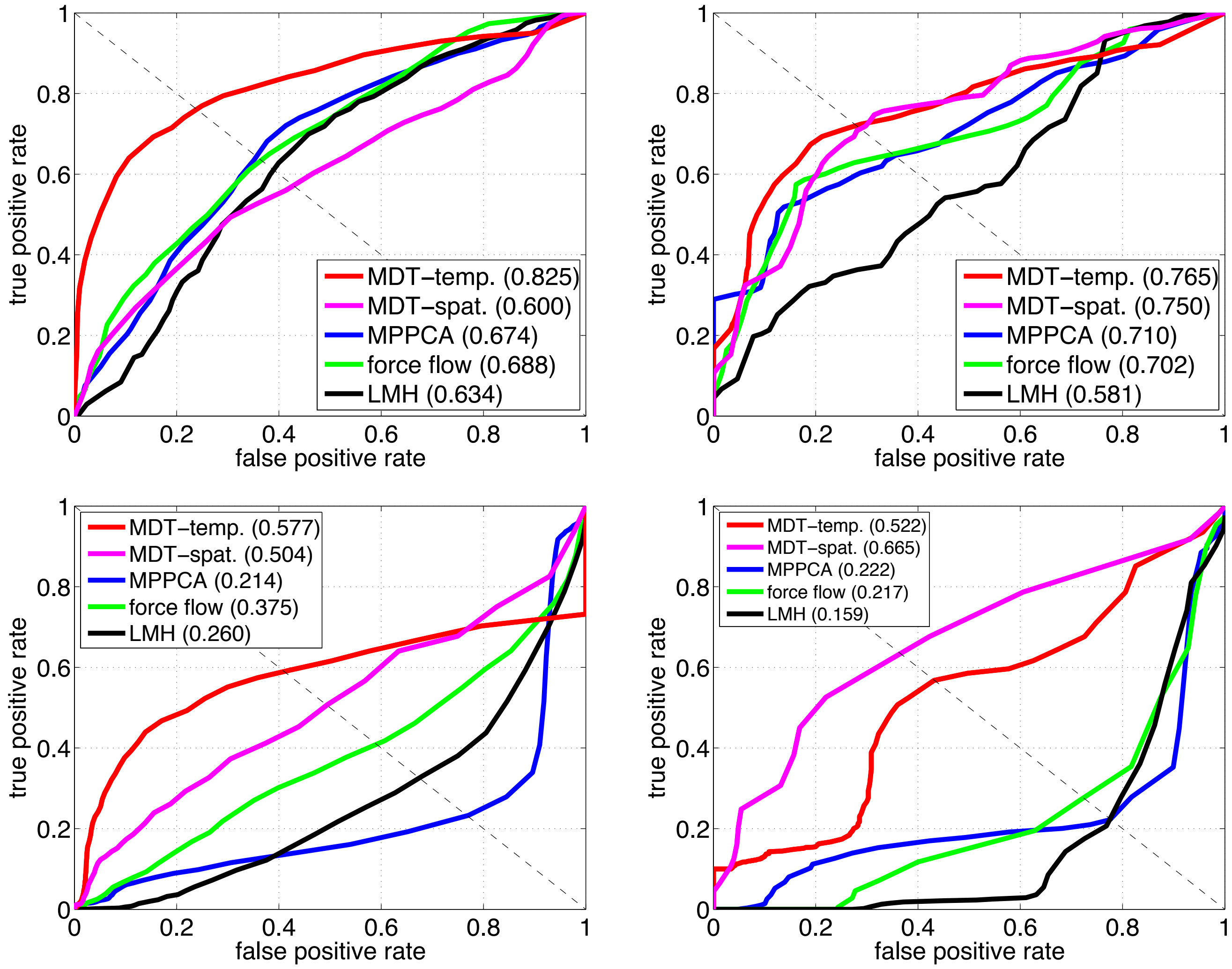

Fig. C.2. Descriptor ROC curves on UCSD anomaly dataset. Plots relative to frame-level criterion are shown on the top row, pixel-level criterion on bottom row. Left: Ped1. Right: Ped2. Shown in brackets are the areas under the curve (AUC). For frame-level, chance performance is the diagonal from $(0, 0)$ to $(1, 1)$. For pixel-level, it is close to a line at 0.

the pedestrian should be stopped. Otherwise, there is no anomaly. Again, we believe that these errors are acceptable in principle, although further studies would be required to verify that they do not overwhelm the operator of the surveillance system when there are no anomalies.

References

[1] R. Shumway and D. Stoffer, "An approach to time series smoothing and forecasting using the em algorithm," *J. Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[2] S. Roweis and Z. Ghahramani, "A unifying review of linear gaussian models," *Neural computation*, vol. 11, no. 2, pp. 305–345, 1999.

[3] T. P. Kah-Kay Sung, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

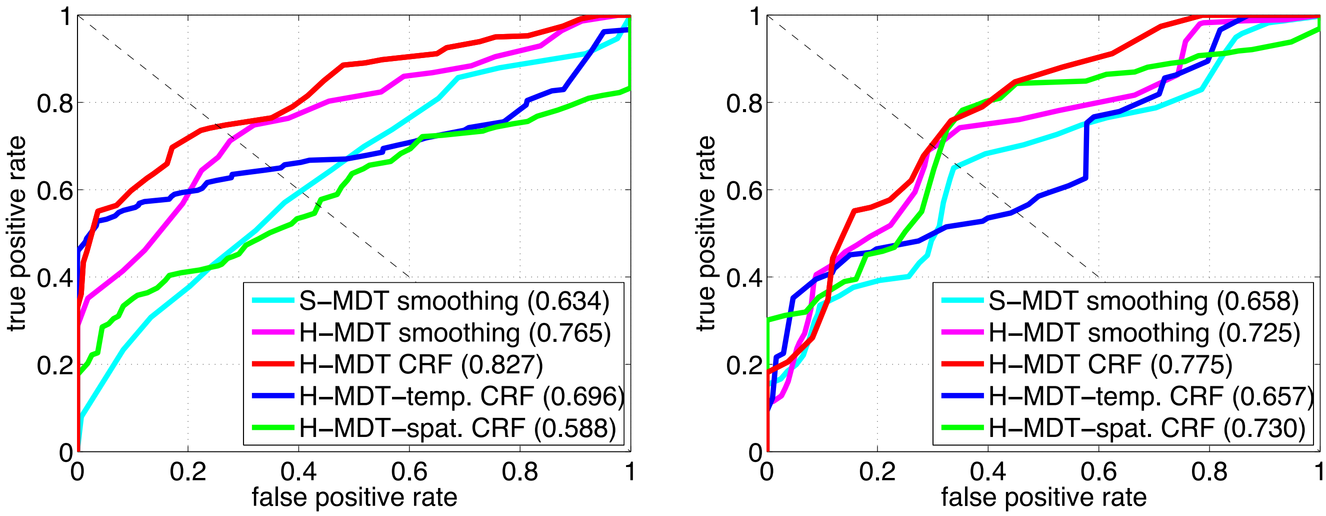

Fig. C.3. Filter ROC curves (pixel-level) on UCSD anomaly dataset. Left: Ped1. Right: Ped2.

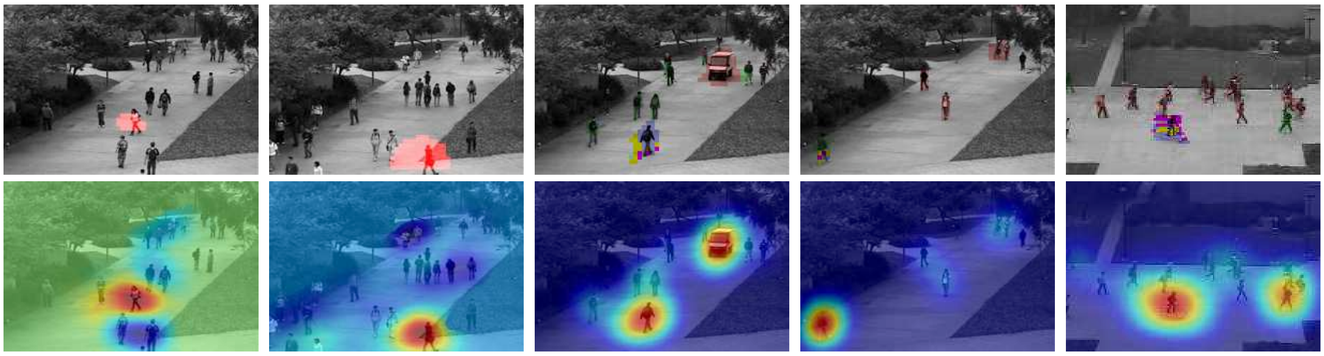

Fig. C.4. False positive anomalies. Left two columns: temporal anomaly detector. Top row shows anomaly predictions in red, the bottom the temporal anomaly maps in "jet" colormap. Right three columns: spatial anomaly detector. Top rows shows the crowd segmentations, bottom row shows the spatial anomaly maps.