A Sequential Monte Carlo Approach to Anomaly Detection in Tracking Visual Events [*]

Peng Cui Li-Feng Sun
Computer Science,
Tsinghua University,
Beijing, China
cuip05@mails.tsinghua.edu.cn

Zhi-Qiang Liu
School of Creative Media,
City University,
Hong Kong
ZQ.LIU@cityu.edu.hk

Shi-Qiang Yang
Computer Science,
Tsinghua University,
Beijing, China
yangshq@tsinghua.edu.cn

Abstract

In this paper we propose a technique to detect anomalies in both individual and interactive event sequences. Anomalies are categorized into two classes: abnormal event, and abnormal context. We model these two kinds of anomalies in the Sequential Monte Carlo framework which is extended by Markov Random Field for tracking interactive events. Firstly, we propose a novel pixel-wise event representation method to construct feature images, in which each blob corresponds to a visual event. Then we transform the original blob-level features into subspaces to model probabilistic appearance manifolds for each event-class. With the probability of an observation associated with each event-class (or state) derived from probabilistic manifolds and state transitional probability, the prior and posterior state distributions can be estimated. We demonstrate in experiments that the approach can reliably detect such anomalies with low false alarm rates.

1. Introduction

The Security problem has become more and more heightened in the world today. Millions of surveillance cameras are placed to discover abnormal objects or events, which require constant evaluation and scrutiny. It would make most sense to develop approaches to detecting anomalies automatically from visual events.

There are two main problems in detecting anomalies:

(1) How to define an anomaly? Anomalies manifest themselves mainly in two manners: First, their appearances are quantitatively or fundamentally different from normal events and are referred as abnormal events; Second, they appear to be normal individually, but happen in an abnormal order, e.g., in a shopping scenario, a *walkaway* event happens immediately after a *can-taken* event without a *pay* event, which is called abnormal contexts. The former anomalies can be detected by some pre-trained normal/abnormal event patterns and a binary or probabilistic matching criterion [8]. The latter anomalies have n-order Markovianity, that is, the degree of anomaly of an event depends on the former n events, which is previously modeled by HMM filter, Kalman filter, etc. The abnormal events can in fact be included by abnormal context, because an abnormal event would necessarily cause an abnormal context. So in this paper, we propose a context-relevant framework to unify the two kinds of anomaly detection.

The Sequential Monte Carlo (SMC) method provides a finite dimensional approximate solution to the posterior probability given the past observations. It is flexible, easy to implement, and applicable in very general settings [1]. Recently, Vaswani *et al.* [5] applied Particle Filtering, an instant of SMC, in anomaly detection. It tracks people's motions in the tangent shape space to detect abnormal motions.

In this paper, we propose an SMC method concentrating on detecting visual anomalies in finite discrete state space, with each state corresponding to a class of events, including individual events and interactive events. In traditional SMC methods, the hidden states are modeled as a Markov process of initial distribution and transition probability matrix. However, this makes SMC unable to model the spatial dependence of states which is required by interactive events. We introduce Markov Random Field (MRF) [6] in the SMC framework to extend its ability to track both individual and interactive events.

(2) How to model visual events? The states are hidden, and the observation is noisy, and the transformation between observation and state may be linear or non-linear,

[*] This work is supported by Hong Kong CityU SRG No. 7001794; National Natural Science Foundation of China, No. 60573167; National High-Tech Research and Development Plan of China, No. 2006AA01Z118; and National Basic Research Program of China, No. 2006CB303103.

so event models are necessary to estimate the probability of a noisy observation associated with a state. Generally, three steps are needed in event modeling: feature extraction, feature selection or transformation, and feature modeling.

Feature extraction: Xiang *et al.* [7] recently proposed a pixel-wised method to model autonomous visual events by extracting features from pixel changes, which avoids object segmentation and tracking. This method requires little prior knowledge and performs efficiently. The main differences between our method and [7] are: 1) In addition to using background substraction method to calculate the retainment of pixel changes, we propose an Adaptive Temporal Differencing method to estimate pixel changing process. 2) we construct a novel feature image by compressing all pixel change-relevant features during a period of time, with each blob in it corresponding to an event. Therefore each event-class can be conveniently represented by a set of blob appearances which are described by blob-level features.

Feature selection: Different classes of events often have their own representative features, and the number of features may vary from class to class. Principal component(PC) selection and variable selection are two solutions to this problem. In this paper, Principal Component Analysis(PCA) is used to extract low-dimensional subspace for each class.

Feature modeling: For high-dimensional space, Moghaddam [2] proposed a method to estimate the Gaussian density of an observation associated with a transformed subspace. Using this method we construct probabilistic manifolds for event-classes, by which the probability of an event associated with an event-class can be estimated.

The original contributions of the proposed method are threefold: firstly, we use SMC to track event sequence in discrete state space for anomaly detection, and propose an MRF-based method to extend SMC for both individual and interactive events; secondly, we propose an Adaptive Temporal Differencing method to describe pixel changes, and an effective and efficient event representation approach; thirdly, we combine SMC and subspace method to realize event tracking in probabilistic manifolds.

The rest of the present paper is organized as follows: Section 2 introduces the SMC framework for event tracking and anomaly detection; then we provide the event modeling method for constructing probabilistic appearance manifolds in Section 3; the experiment results are given in Section 4, followed by conclusions in Section 5.

2. SMC for Anomaly Detection

2.1. Particle Filters

SMC is normally implemented in terms of Particle Filters (PF) which is a simulation-based method to estimate at time instant t, the posterior distributions in the state space ϕ. With x_t representing a configuration of ϕ, and y_t representing a configuration of observation space ψ, the tracking problem is to estimate the posterior probability $p(x_t|y_{1:t})$ given all past observations up to time instant t. Given a cloud of N particles, we can estimate the posterior probability as follows:

$$\hat{P}_N(dx_t|y_{1:t}) = \frac{1}{N}\sum_{i=1}^{N}\tilde{\omega}_t^{(i)}\delta_{x_t^i}(dx_t), \qquad (1)$$

where $\tilde{\omega}_t^{(i)}$ is the weight of the i^{th} sample, and $\delta_{x_t^i}(dx_{(t)})$ denotes the delta-Dirac mass located in x_t^i. In most cases, the weight is approximated by the prior probability which will be specified in the next section:

$$\tilde{\omega}_t^{(i)} = p(y_t|x_t^i). \qquad (2)$$

Now assuming that the distribution of x_{t-1} given observations up to time $t-1$ has been approximated as $\hat{P}_N(dx_{t-1}|y_{1:t-1})$ by the N particles, then at time instant t, for $i = 1, ..., N$, sample $\tilde{x}_t^i$ according to $p(x_t|x_{t-1})$. Then evaluating each new particle's importance weight by $\tilde{\omega}_t^{(i)}$, and normalize the weights. Finally, resample N particles from $\tilde{x}_t^{(i)}$ conform to the importance weights. Interested readers are referred to [1] for more details.

In our case, the Markov hidden state space ϕ is discrete, with each state corresponding to a class of event. In the PF framework, two prior probability distributions are required: $p(x_t|x_{t-1})$ as the sampling criterion, and $p(y_t|x_t^i)$ as the weight factor. The two terms are specified respectively in Section 2.2 and Section 3.

2.2. Event Tracking

In most cases, there are multiple events caused by different persons or objects in a certain scene. We use Event-Sequence (ES) $x_i = (x_{i,1}, x_{i,2}, ..., x_{i,t})$ to indicate events caused by the same person(s) or object(s) at different time instants (Note that x_i can also stand for the persons and objects that cause the event sequence x_i). Tracking is restricted within ES along temporal axis. The acquirement of ES is an iterative process. Given $x_{i,t-1}$, $x_{i,t}$ is the event happening at a spatially connected location with $x_{i,t-1}$. More specifically, $x_{i,t-1}$ ($x_{i,t}$) is the ascendent(descendent) event of $x_{i,t}$ ($x_{i,t-1}$).

We classify events into individual events and interactive events by the number of persons or objects that cause the events. In individual event case, all events in a scene can be regarded to be independent with each other, so it is reasonable to estimate the prior probability in one-order Markov Chain framework as $p(x_{i,t}|x_{i,t-1})$. However, when interactive event happens at time t, $x_{i,t}$ would have more than one ascendent events. For example, before a two-man-handshaking event happens, two man-walking events have happened. These ascendent events all influence $x_{i,t}$'s distribution. (Apparently, the probability of handshake event happening after two man-fast-run events is much smaller than that after two man-walk events.) So the central issue in estimating the posterior probability distribution of $x_{i,t}$ is to judge whether it is an interactive event and determine its ascendent events.

Here we propose a method based on Markov Random Field (MRF) to predict at time $t-1$ whether there would be an interactive event at t. We construct an undirected graph $G = (V_t, E_t)$ to describe the spatial relationship of events, where $V_t(x_{1,t}, x_{2,t}, ..., x_{m,t})$ represents the appearing events at time instant t. m is variable, because of cases such as new events coming into the scene, and events disappearing from the scene. E_t is the pairwise distance matrix which is commonly used in MRF methods to describe the spatial distance of events in V_t. After introducing the pairwise MRF, we constrain ourselves on considering the interactive events between two people or objects. Then the state transition probability becomes:

$$p(x_{i,t}|x_{i,t-1}, x_{j,t-1}). \tag{3}$$

If $\min_j(E_{t-1}(i,j)) > \varepsilon$, which means $x_{i,t-1}$ is in a single clique, then $x_{i,t-1}$ and $x_{j,t-1}$ can be regarded to be independent. Hence,

$$p(x_{i,t}|x_{i,t-1}, x_{j,t-1}) = p(x_{i,t}|x_{i,t-1}). \tag{4}$$

An interactive event can be predicted at time $t-1$ by the following two conditions:

$$\min_j(E_{t-1}(i,j)) < \varepsilon, \tag{5}$$

$$E_{t-1}(i,j) < E_{t-2}(i,j). \tag{6}$$

The first condition indicates that $x_{i,t-1}$ and $x_{j,t-1}$ are very near; and the second indicates that x_i and x_j are approaching. When the two conditions are met, $x_{i,t-1}$ and $x_{j,t-1}$ are considered as ascendent candidates of the upcoming interactive event. Then the prediction is validated at time t by

$$(E_t(i,j)) < \varepsilon. \tag{7}$$

If validated, an interactive event happens and the $x_{i,t-1}$ and $x_{j,t-1}$ are regarded as its ascendent events.

Actually, each interactive event can be considered as either two dependent individual events or one event as a whole. For the ease of understanding and representation, we use one variable $x_{r,t}$ to represent the interactive event. Note that $r \neq i \bigwedge r \neq j$ because the objects or persons causing $x_{r,t}$ are different from $x_{i,t}$ and $x_{j,t}$ but a sum of them. As a result, an interactive event is the end of its ascendent individual event sequences and the start of a new interactive event sequence.

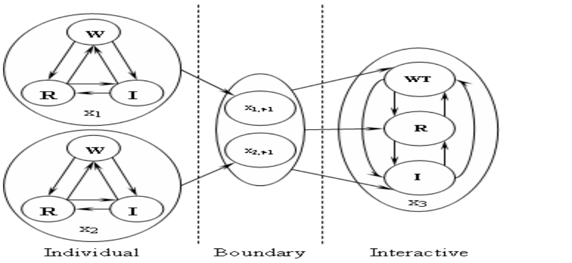

Figure 1. Interactive event tracking. x_1 and x_2 are two independent individual event sequences including Walk, Run and Inactive events (people stay at a place with tiny actions), and x_3 is an interactive event sequence including Walk-together, Run-together and Inactive events. $x_{1,t-1}$ and $x_{2,t-1}$ represent the state of x_1 and x_2 immediately before the interactive event happens.

An illustrated chart of the state transition process is shown in Figure (1). Before $t-1$, the two individual event sequences x_1 and x_2 progress independently. They approach each other and at time $t-1$, their pairwise distance become smaller than predefined threshold, and step into the boundary condition. Then the predicted interactive event is validated at t and an interactive event sequence x_3 has generated with x_1 and x_2 ended at the same time.

Using this method, the prior state transitional probabilities $p(x_t|x_{t-1})$ for both individual and interactive events are unified in the same framework. The other prior probability $p(y|x)$ required by PF will be specified in Section 3.

2.3. Anomaly Detection

Based on the past observations of $y_0 \sim y_{t-1}$, we can estimate the prior probability $p(x_t|y_{1:t-1})$. When y_t is available, the posterior probability $\hat{P}_N(dx_t|y_{1:t})$ can also be estimated. The difference between the prior and posterior probabilities is an important criterion to reveal the degree of anomaly at t. Here, we use the ELL proposed in [5] to measure the difference between the two probability distribution:

$$ELL_{i,t} = E_{\hat{P}_N(dx_{i,t}|y_{1:t-1})}[-\log \hat{P}_N(dx_{i,t}|y_{1:t})]. \tag{8}$$

In our case, if $ELL_{i,t} > \xi$, the event at time instant t in x_i is regarded as an anomaly.

3. Event Modeling

In order to assign weights to particles, the weight factor $p(y_t|x_t)$ is required in PF framework. In this section, the observation y_t is represented by a set of novel pixel-wise features (in Section 3.1) and the relations between observations and hidden states are modeled by probabilistic manifolds (in Section 3.2).

3.1. Pixel-wise Event Representation

As a visual event can be regarded as a group of dynamic pixel changing in a certain rule within a period of time, we represent different events by different pixel change-relevant features. In fixed camera scenarios, there are two main methods to detect pixel changes: Background Substraction and Temporal Differencing [8]. Xiang *et al.* [7] built a dynamic background model, and used Background Substraction method to calculate each foreground pixels' changing history which is used to model events.

In this paper, we exploit two important change-relevant features: 1) Pixel Change Frequency (PCF): the changing times of a pixel given a certain period of time; 2) Pixel Change Retainment (PCR): the duration of a pixel retaining a value different from background. As pixel-wise methods are sensitive to noise, we firstly represent each frame in a pyramid structure, in which a block of 8*8 pixels are downsampled into a super-pixel, and assign the average grey-scale value of these pixels to the super-pixel which is referred to as pixel in the remaining part of the paper for simplicity. Then we use Temporal Differencing method for PCF and Backgound Substraction method for PCR. Finally, given a time duration Δt, all pixels' PCF and PCR are reflected on a feature image, with each blob in it corresponding to an event happening within Δt.

3.1.1 Pixel Change Frequency (PCF)

PCF is the number of changes a pixel has undergone within in Δt, which can be computed using temporal differencing methods. In the literature, temporal differencing is often used to detect pixel-wise valid changes (pvc) between two consecutive frames:

$$pvc_{t,t-1}(i,j) = \begin{cases} 1 & if(v_t(i,j) - v_{t-1}(i,j)) > \vartheta_t(i,j) \\ 0 & otherwise, \end{cases} \tag{9}$$

where $v_t(i,j)$ represents the pixel value of pixel (i,j) in frame t, and $\vartheta_t(i,j)$ is the threshold. After that, the PCF can be calculated as:

$$PCF_{t,t+\Delta t}(i,j) = \sum_{m=t}^{t+\Delta t} pvc_{m+1,m}(i,j). \tag{10}$$

In the above method, there are two important issues: 1) How to set the threshold $\vartheta_t(i,j)$? 2) Is it reasonable to compare consecutive frames?

(1) Threshold for change detection

Due to the spectrality of lights and the noise of CCD sensor, all acquired measures are noisy. Therefore a proper threshold is needed to detect changes caused by object motions but not noises. Psychophysical studies have revealed that the visual threshold (also known as just-noticeable difference) depends on the illumination of background [4]:

$$\frac{\Delta I_b}{I_b} \approx \alpha, \tag{11}$$

where I_b is the intensity of background and ΔI_b is the minimum difference from I_b required for human's perception.

We set local threshold $\Delta I_b = \alpha I_b$ for each pixel (i,j) according to its background value. More specifically, we assign 0.1 to α, which is adequate in our experiment.

(2) Adaptive temporal differencing

When very slow motion happens, pixel's difference between any two consecutive frames cannot surpass the predefined threshold because of its slow and gradual change, which results in the loss of change-relevant information. To remedy this, we propose the adaptive temporal differencing method to adaptively adjust the frame sampling period.

We use f_t to represent the frame number sampled at time instant t, and sp_t to represent sampling period at time t. In our method, sp_t also plays a role of backward window; that is, f_t can compare with only frames during $[f_t - sp_t, f_t - 1]$. The initial value of sp_t is set to 1, and its upper limit is set to sp_{max}.

Note that f_t and sp_t are different from pixel to pixel, because the motions on different pixels are not the same. So f_t and sp_t are replaced by $f_t(i,j)$ and $sp_t(i,j)$ to add the constraint of pixel's position. For a certain pixel (m,n), $sp_t(m,n)$ is adjusted in following cases:

If $\sum_{i=1}^{sp_t(m,n)} pvc_{t,t-i}(m,n) > 0$, that is, in the frame sequence $[f_t(m,n) - sp_t(m,n), f_t(m,n)]$ (which sizes $sp_t(m,n)$) there happens a change on pixel (m,n), then

$$sp_{t+1}(m,n) = \arg\min_{\sigma}(pvc_{t,t-\sigma}(m,n) = 1); \tag{12}$$

$$f_{t+1} = f_t + sp_{t+1}. \tag{13}$$

If $\sum_{i=1}^{sp_t(m,n)} pvc_{t,t-i}(m,n) = 0$, that is, there is no change happening on pixel (m,n) in frame sequence $[f_t(m,n) - sp_t(m,n), f_t(m,n)]$, then

$$sp_{t+1}(m,n) = \min(sp_t(m,n) + 1, sp_{max}); \tag{14}$$

$$f_{t+1} = f_t + 1. \tag{15}$$

An illustration of these cases are shown in Figure (2). The second case happens at frame 3, where sp increases; the first case happens at frame 2,4,6,8,9, where at frame 8 sp decreases, and at other frames sp remains to be the last sp.

The proposed method has been proved to be effective in our experiments with $sp_{max} = 3$. An example is given in Figure (3). As shown, the average of sp_t in slow-motion

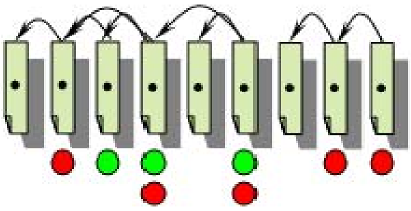

Figure 2. Adaptive temporal differencing. This is an Adaptive Temporal Differencing process of a certain pixel (the black point) on nine frames. Red circle indicates that the pixel change is detected on the frame, and green circle indicates no change. Two circles under one frame means that the current frame is respectively compared with its former two frames.

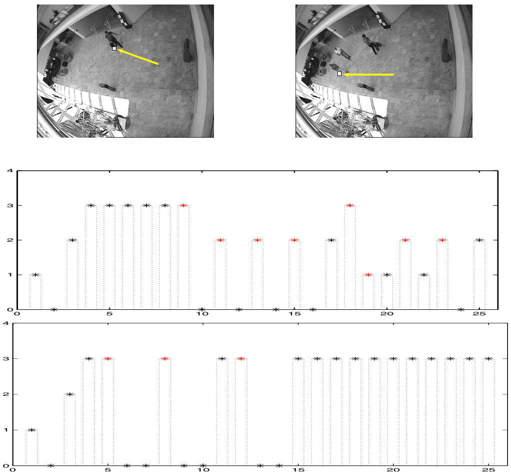

Figure 3. Experiment results for change detection in slow motion and normal motion events. The top two illustrated frames correspond respectively a normal motion scenario and a slow motion scenario. Observation pixels are indicated by the arrow on the two frames, and the change detection results are respectively represented in two graphs in which the x-axis represents the frames; y-axis represents the sampling period for the indicated pixel in a frame; the red stars indicate that a change of the pixel is detected in current frame; and black stars indicate no change detected.

case is higher than that in normal-motion case, and consequently the changes detected in slow-motion is less than that in normal-motion, which conforms to our expectation. We have also tested these sequences in traditional temporal differencing methods which only compare consecutive frames. All changes in slow-motion case can't be detected, and among eight changes in normal-motion case only one can be detected.

With the threshold derived based on the human visual properties and the proposed adaptive temporal differencing method, we can calculate PCF concisely. In the next part, we specify another change-relevant feature PCR.

3.1.2 Pixel Change Retainment (PCR)

PCR is dedicated to describing the change retaining duration, which is another important aspect of pixel changes. Eventually, PCF and PCR are not completely independent. Given a time period Δt, PCR of a pixel's change vary from 1 to Δt. If $PCF = 0$ in Δt, then the maximum error of estimating PCR using PCF is Δt; if $PCF = 1$, then the maximum estimating error reduces to $\Delta t/2$; if PCF is very high, the PCR of each change would approximate $\Delta t/PCF$. In practice, to calculate PCR for each pixel change requires too heavy computation and storage space, and would cause the features hard to model. In order to describe PCR economically, we only consider the case of a change retaining for the whole Δt.

The change retaining for the whole Δt should meet three conditions: 1) at least one change happens on this pixel during $[t - \Delta t, t]$ to turn the pixel value from v_b (background value) to v_f (foreground value); 2) no change happens during $[t, t + \Delta t]$; 3) the pixel value during $[t, t + \Delta t]$ is not equal to v_b. So PCR can be modeled as:

$$PCR_{t,t+\Delta t}(i,j) = \begin{cases} \Delta t & if PCF_{t-\Delta t,t}(i,j) \neq 0 \\ & PCF_{t,t+\Delta t}(i,j) = 0 \\ & and\ v_t(i,j) \neq v_b(i,j) \\ 0 & otherwise \end{cases}, \tag{16}$$

where $v_b(i,j)$ represents the background value of pixel (i,j) which is learned by a dynamic background model proposed in [3].

We have proposed PCF and PCR to express pixel change's frequency property and retainment property, which are jointly used to describe pixel changes. In the following section, we combine relevant pixels together to represent a visual event.

3.1.3 Feature Image

Given a time duration Δt, the dynamic pixels would pose different changing characteristics which are described by PCF and PCR. From (10) and (16), we can see that PCF and PCR are incompatible in the sense that $PCR \times PCF = 0$. Besides, PCR is a binary value of 0 or Δt, whereas PCF is any value in the range of $[0, \Delta t)$ (it is almost impossible for a pixel to change in every frame). In practice, PCR characterizes the static events such as static-object-intrusion, man-lying-down, etc. and PCF characterizes dynamic events such as man-walking, man-running events, etc.

For ease of understanding, we represent these features in a grey-scale feature image, in which the pixels' grey-scale values are proportional to the pixel level feature in (17). Some examples are given in Figure (4).

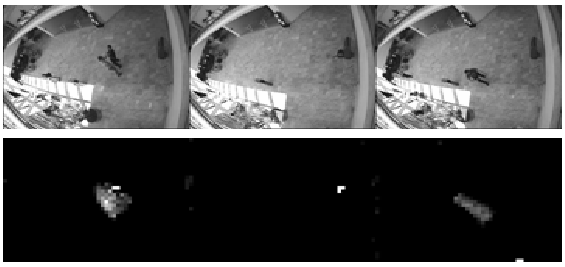

Figure 4. Event image examples. The events from left to right are respectively Fight, Left-bag and Walk.

$$\mathscr{F}_p(i,j) = PCF_{t,t+\Delta t}(i,j) + PCR_{t,t+\Delta t}(i,j). \quad (17)$$

As each event is caused by motions of person(s) or object(s), each action would cause the changes of a group of neighbouring pixels, and pixel changes caused by the same person or object are spatially connected. We combine the non-zero pixels in the feature image into blobs using connected component method. Then each blob corresponds to a visual event which is characterized by the blob appearance.

On the feature image, each blob can be described by some blob-level features, such as size, shape, elongatedness, luminance histogram, among which the luminance histogram is rotation, displacement invariant. We adopt luminance histogram as the blob-level feature. In order to make it scaling invariant, we construct a percentile luminance histogram with ℓ equally spaced bins for each blob to describe event-level feature:

$$\mathscr{F}_e =< P_1, P_2, ..., P_\ell >, \quad (18)$$

where P_i represents the percentage of pixels falling in the i^{th} bin.

The features of some typical visual events are shown in Figure (5), which demonstrates that the proposed features are discriminative to differentiate these events. Walk events only has distribution on $1 \sim 7$ dimensions; only Fight events has distribution on $8 \sim 12$ dimensions; only Interactive events have both distributions on $1 \sim 5$ dimensions and 17^{th} dimension; and Fall-down events only have distribution on 17^{th} dimension.

From Figure (5), we can see that different class of events characterize themselves in different feature dimensions. In the next Section, we transform the original features into different subspaces to assign each event-class with the most representative features, and model the appearance variation of each event-class in probabilistic manifolds.

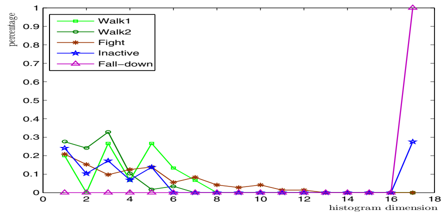

Figure 5. Feature distributions of typical events. Features of two Walking, one Fighting, one Inactive, and one Man-Fall-Down events are plotted.

3.2. Probabilistic Appearance Manifolds

Different event-classes have different representative features, and the number of features required to describe an event-class may not equal. In order to better discriminate different event-classes, we transform the original feature space into different subspaces for different event-classes using PCA. Each event-class has multiple different appearances because of different views and acting objects. The samples of different appearances for each event-class constitute a manifold, which is approximated by a PCA plane.

Given an observation y which is represented by $\mathscr{F}_e$, we aim to calculate the probability $p(y|x)$ which is required by PF framework as a weight factor. As x_i (a state or an event-class) has been approximated by an affine subspace ψ_i, the probability could be estimated by

$$p(y|x_i) = p(y|\psi_i). \quad (19)$$

We use the method proposed in [2] to estimate the Gaussian densities in subspaces:

$$p(y|\psi_i) = \left[\frac{exp(-\frac{1}{2}\sum_{m=1}^{k} \frac{\ (m)^2}{\lambda_m})}{(2\pi)^{k/2} \prod_{m=1}^{k} \lambda_m^{1/2}} \right] \left[\frac{exp(-\frac{\sum_{m=k+1}^{p} \ (m)^2}{2\rho})}{(2\pi\rho)^{(p-k)/2}} \right] \quad (20)$$

where p and k are respectively the total component number and the selected PC number; λ is eigenvalue; $y = (y(1), ..., y(p))$, $\ = (\ (1), ..., \ (p))$, and $\ = \varphi(y - \overline{y_i})$, where φ is the eigenvector matrix, and $\overline{y_i}$ is the center of y in i^{th} event-class; and

$$\rho = \frac{1}{p-k} \sum_{m=k+1}^{p} \lambda_m. \quad (21)$$

Using this method, the Gaussian density is divided into two terms. The first term is the true marginal density in eigenspace and the second term is the estimated marginal density in the orthogonal complement space. In high dimensional space, observations only occupy a very tiny part of the hyperspace, so the densities are in great disparity.

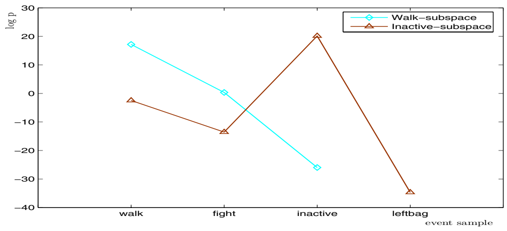

Figure 6. Gaussian densities in subspaces. We sampled 20 event appearances with 5 from each event-class, and project them respectively into the subspaces of Walk and Inactive event-class. The average probability densities of each subset of test samples are plotted. Note that the density of leftbag test samples in Walk subspace is zero, so its log value is omitted in the figure.

This would cause all particles in SCM framework gathering onto one state. To remedy this, we use $\lg(p(y|x))$ instead to measure the weight of a particle (note that $p(y|x)$ is a probability density), and if $\lg(p(y|x)) < 0$, then the weight of that particle is set to 0.

We have performed cross validation experiments in our dataset and typical results are given in Figure (6). It shows that in Walk event-class's subspace, the probability density of walk test sample is much higher than that of the other samples. In Inactive event-class subspace, the density of the inactive test sample is highest. The results indicate that the probabilistic appearance manifolds constructed by the proposed method are both discriminative for out-class samples (outliers) and representative for in-class samples.

We have so far derived the prior probability of $p(y|x)$ by the process: (1) representing a visual event with a blob in the feature image; (2) extracting blob-level features, and transforming these features into subspaces; (3) in subspace, constructing probabilistic appearance manifolds under Gaussian assumption. Then the prior probability $p(y_t|x_t)$ is used in PF framework for event tracking and anomaly detection.

4. Experiment

4.1. Experiment Data

We have conducted experiments on PETS2004 dataset with resolution of 384*288 pixels. The data set consists of 28 video sequences with in total 26419 frames to describe 5 scenarios, that is, Walking, Browsing, Collapse, Leaving objects, Meeting and Fighting appended with ground truth. These frames are divided into clips with each clip constituted of 25 frames (about 1 second). Each event inside a clip is considered as an event sample. We extract from these samples five event classes for training and testing: individual_walking, two_man_fighting, left_bag, inactive, man_fall_down among which half of walking, inactive event samples are used for training, and the rest samples are used for testing. The illustrated frames of two event sequences containing anomalies are given in Figure(7).

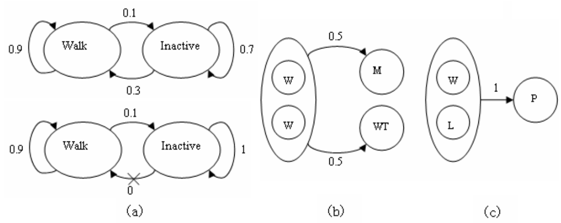

Figure 8. State transition.

4.2. Individual Events

In this subsection, we testify the proposed method on individual event sequences. From the individual training dataset, we construct event models in form of probabilistic appearance manifolds, and derive the transition probabilities of states (event-class) as shown in the top graph of Figure (8(a)).

In the surveillance scenario, we regard the event sequences only consisting of walking and inactive events as the normal individual events sequence; a sequence of walk-inactive-falldown event sequence as the abnormal event sequence in which the falldown event should be detected as anomaly. The result is shown in Figure (9). The curve of normal1, normal2 and normal3 sequences are smooth, and an outlier in the abnormal event sequence is detected which corresponds rightly to the abnormal falldown event.

In order to test the abnormal context cases, we assume another scenario in which only the process of walk-inactive-inactive is allowed, that is, the walk event is normal before inactive event happens, but abnormal after that. The state transition probability is expressed in the bottom graph of Figure (8(a)). (Note that the transition probability from Inactive to Walk would approximate 0 in this case.) Then, as shown on the abnormal context curve, in the sequence of walk-inactive-walk, anomaly is detected on the walk event after inactive.

We use a threshold of $ELL = 40$ to detect anomalies in 30 event sequences including 20 normal sequences, 2 abnormal and 8 context abnormal sequences. Among all the test sequences, the detecting rate is 100% and false alarm rate is 10%. Only one walk event is mistaken as an inactive event because a man wearing a black suit move slowly in vertical direction, which cause blob-level features similar with inactive events.

4.3. Interactive Event

In the interactive event test case, we assume three normal scenarios: 1) walkA-walkB-meet, which means two individual men approach each other to meet (here meet is

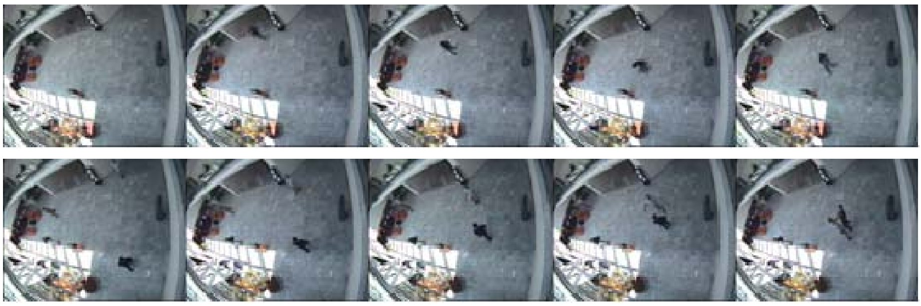

Figure 7. Illustrated frames of abnormal event sequences. The top row is the abnormal individual sequence of fall_down, and the bottom is the abnormal individual sequence of fighting.

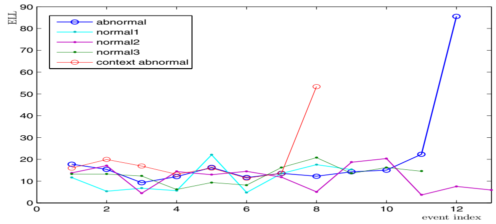

Figure 9. ELL of individual event sequences.

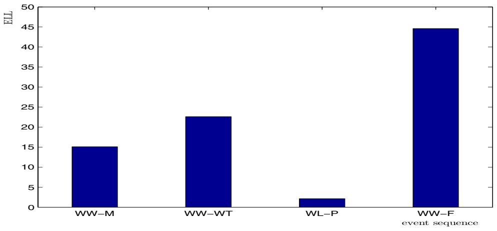

Figure 10. ELL of interactive event. W, WT, L, P, F respectively represents Walk, WalkTogether, Left_bag, Pickup, Fight events. For ease of understanding, this figure only plot the ELL at the time instant when interactive events happen.

a kind of Inactive events); 2) walkA-walkB-walktogether, which means two individual men approach each other to walk together; 3) walk-leftbag-pickup, which means an individual man approach a left-bag to pick it up (a kind of inactive event). Then we test on a walk-walk-fight sequence, in which two men approach each other and fight.

The state transition matrix derived from training set are illustrated by Figure (8(b)(c)). Walk-together events have the similar features with individual walk events in our event modeling method; meet and bag-pickup events are described as inactive events. With these priors, the anomaly detection results for interactive events are shown in Figure (10).

From the experiment results, we can see that the abnormal interactive event is detected by its too high ELL value. The results also demonstrate that the interpretation of an interactive event depends on its ascendent individual events; for example, the inactive event happens after two_man_walk is Meet event, and that happens after Walk-Leftbag events is Leftbag_pickup event.

5. Conclusion

In this paper, we have proposed an anomaly detection framework by combining pixel-wise event representation, probabilistic manifold construction, and Sequential Monte Carlo methods. The experiment results show that our implementation of the framework is able to reliably detect both abnormal events (including both individual events and interactive events) and abnormal contexts.

References

[1] A. Doucet, N. Freitas, and N. Gordon. Sequential Monte Carlo Methods in Practice. Springer, 2001. 1, 2

[2] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, No. 6, June, 2002. 2, 6

[3] C. Stauffer, WEL. Grimson. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol 22, pp. 747-757, 2000. 5

[4] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec. High Dynamic Range Imaging. Elsevier, 2006. 4

[5] N. Vaswani, A. RoyChowdhury, R. Chellappa. "Shape Activity": A Continuous State HMM for Moving/Deforming Shapes with Application to Abnormal Activity Detection. IEEE Trans. Image Processing, Vol. 14, No. 10, October 2005. 1, 3

[6] S. Li. Markov Random Field Modeling in Computer Vision. Springer Verlag, 1995. 1

[7] T. Xiang and S. Gong. Model selection for unsupervised learning of visual context. International Journal of Computer Vision, Vol. 69, No. 2, pp. 181-201, 2006. 2, 4

[8] W. Hu, T. Tan, L. Wang, S. Maybank. A survey on visual surveillance of object motion and behaviors. IEEE Trans. Systems, Man, and Cybernetics, Vol. 34, No. 3, 2004. 1, 4