Machine Vision and Applications
DOI 10.1007/s00138-012-0426-4

ORIGINAL PAPER

Panoramic Gaussian Mixture Model and large-scale range background substraction method for PTZ camera-based surveillance systems

Kang Xue · Yue Liu · Gbolabo Ogunmakin · Jing Chen · Jiangen Zhang

Received: 12 May 2011 / Revised: 2 March 2012 / Accepted: 21 March 2012

Abstract In this paper, we present a novel approach for constructing a large-scale range panoramic background model that provides fast registration of the observed frame and localizes the foreground targets with arbitrary camera direction and scale in a Pan–tilt–zoom (PTZ) camera-based surveillance system. Our method consists of three stages. (1) In the first stage, a panoramic Gaussian mixture model (PGMM) of the PTZ camera's field of view is generated off-line for later use in on-line foreground detection. (2) In the second stage, a multi-layered correspondence ensemble is generated off-line from frames captured at different scales which is used by the correspondence propagation method to register observed frames online to the PGMM. (3) In the third stage, foreground is detected and the PGMM is updated. The proposed method has the capacity to deal with the PTZ camera's ability to cover a wide field of view (FOV) and large-scale range. We demonstrate the advantages of the proposed PGMM background subtraction method by incorporating it with a tracking system for surveillance applications.

Keywords PTZ camera · Panoramic Gaussian mixture background · Multi-layered propagation · Foreground detection · Object tracking

1 Introduction

Due to their ability to cover a wider FOV and a large-scale range, PTZ cameras are becoming more popular than stationary cameras in surveillance systems [1,2]. Numerous background modeling methods have been proposed within existing object detection and tracking algorithms [3–5], however, most of them require stationary cameras to generate a background model, so they are not suitable for PTZ cameras. The objective of this paper is to propose an effective background subtraction method for a PTZ camera-based surveillance system with a wide FOV and large-scale range. This paper presents an approach for detecting foreground objects with arbitrary camera direction and scale by performing fast registration of observed frames to the PGMM. This method is divided into three stages.

Stage 1 generates a PGMM which provides global information for the PTZ camera's FOV. Stage 2 performs multi-layered, feature-based propagation to calculate the observed frame's camera position, which serves to enable large-scale registration with the PGMM to estimate the correspondence background. Stage 3 performs foreground detection using the correspondence background, and updates the PGMM. Using the panoramic background, we visualize the target's trajectory within the camera's FOV.

The remainder of this paper is organized as follows: Sect. 2 reviews related work in background modeling and PTZ camera-based surveillance systems. Section 3 describes the generation of the PGMM. Section 4 presents the feature-based, multi-layered correspondence propagation method. Section 5

A short version has been accepted by ICIP2011.
Scale in our paper is defined as the zoom ratio of the PTZ camera which could be found in Eq. 22, Sect. 6.2.

K. Xue (✉) · Y. Liu · J. Chen · J. Zhang
Beijing Institute of Technology, Beijing, China
e-mail: xuek622@gmail.com

K. Xue · G. Ogunmakin
Georgia Institute of Technology, Atlanta, GA, USA

Published online: 20 April 2012

describes the foreground detection and background update method. Section 6 illustrates the experimental results and Sect. 7 concludes the paper.

2 Related work

Background subtraction is often the first task in automatic surveillance systems. The performance of background subtraction is dependent mainly on the background modeling algorithm. Background modeling algorithms can be classified into two categories [5]: pixel-based and block-based models. In *pixel-based models*, probability distributions, such as Gaussian, Mixture of Gaussians (MOG), or non-parametric [4,6], are used to model the pixel. MOG, introduced by Friedman and Russell [7], extended the single Gaussian model by using more than one Gaussian distribution per pixel to improve background subtraction for a traffic surveillance system. MOG is modified by Stauffer and Grimson [8], who use an on-line K-means approximation to update the parameters of the MOG model. *Block-based models* mainly use the features of independent or slightly overlapped blocks. Heikkila and Pietikainen [3] proposed a method based on local binary pattern (LBP) operators to create a background model. LBP operators have the ability to tolerate illumination changes and have excellent performance in many applications. Due to its high computational demands, Helmut and Horst [9] combine Haar-like features and HOG into an on-line feature selection framework to improve the LBP approach.

Many researchers have attempted to extend stationary background modeling methods for PTZ cameras. In the literature of PTZ camera-based tracking systems, there are three categories: frame-to-frame methods, frame-to-background methods and frame-to-global methods. *Frame-to-frame methods* use the information of the overlap region between an observed frame and its previous frames. The advantage of these methods is that their time cost for training is low. Kang and Paik [10] present an adaptive background generation algorithm which uses a geometric transform-based mosaic method. The system updates its background, which is generated using a Gaussian Model, as the PTZ camera is rotating. However, a limitation of this system is that it cannot zoom in to get high-quality view of the target. Wu and Zhao [11] use a frame-to-frame method which obtains the camera's parameters relative to the previous frame. The limitation of this method is that it accumulates error over a long sequence. Christian and Gian L. [12] introduce *Frame-to-background methods* in their paper. Both previous frame and current frame, after compensation, are processed by a change detection method to detect moving objects. However, their method cannot be used when the scale changes. *Frame-to-global methods* use a panorama or an added camera to provide global spatial information for foreground detection and tracking [13,14]. Sinha and Pollefys [15] use a number of high-resolution frames at different scales to stitch a panorama for each scale and calibrate all the panoramas together in order to get detailed information on the camera's FOV, but their method does not provide a robust background for foreground detection since the panorama at each scale may include moving objects. Chen and Yao [16,17] use an added camera to extend the FOV in their surveillance system, but a small object is difficult to detect in the low-resolution images when the added camera is capturing the entire surrounding, especially in a large area. Although these methods provide global information to meet the PTZ's rotation change, they do not work when the PTZ's scale changes rapidly.

In this paper, we use a pixel-based background model with a frame-to-global method. Compared with frame-to-frame methods, our model provides global information even when the PTZ camera's direction and scale changes rapidly, meaning our method is more robust for the PTZ camera's application. Unlike Chen and Yao, we use a single PTZ camera in our system because our background model provides enough information for foreground detection. The contributions of our paper are threefold. First, we create a panoramic background mixture model for a PTZ camera that is effective for foreground detection. Second, we introduce a multi-layered propagation method that provides fast registration, even when observed frames are captured at different scales. Third, this method allows visualization of a target's trajectory on a panoramic background, no matter in which scale the observed frame is captured, which can be used to determine if a target's behavior is abnormal.

3 Panoramic Gaussian mixture background model

Adaptive background models have recently been proven as an effective method for foreground detection. One such model is the Gaussian Mixture Model (GMM), which is traditionally used for stationary cameras. We extend the model to a PTZ camera by using *Panoramic Frames*. Each panoramic frame is a panorama that covers the camera's FOV. By using the panoramic frames and GMMs, an adaptive panoramic gaussian mixture background model can be generated. This section describes the panoramic frame generation and background model initialization.

3.1 Generating panoramic frames

To generate a panoramic frame, we use several frames captured by the PTZ camera. These frames are called key frames. A set of key frames have two characteristics. First, the set covers the surveillance area. Second, each key frame must overlap with at least one other key frame. The overlap region is

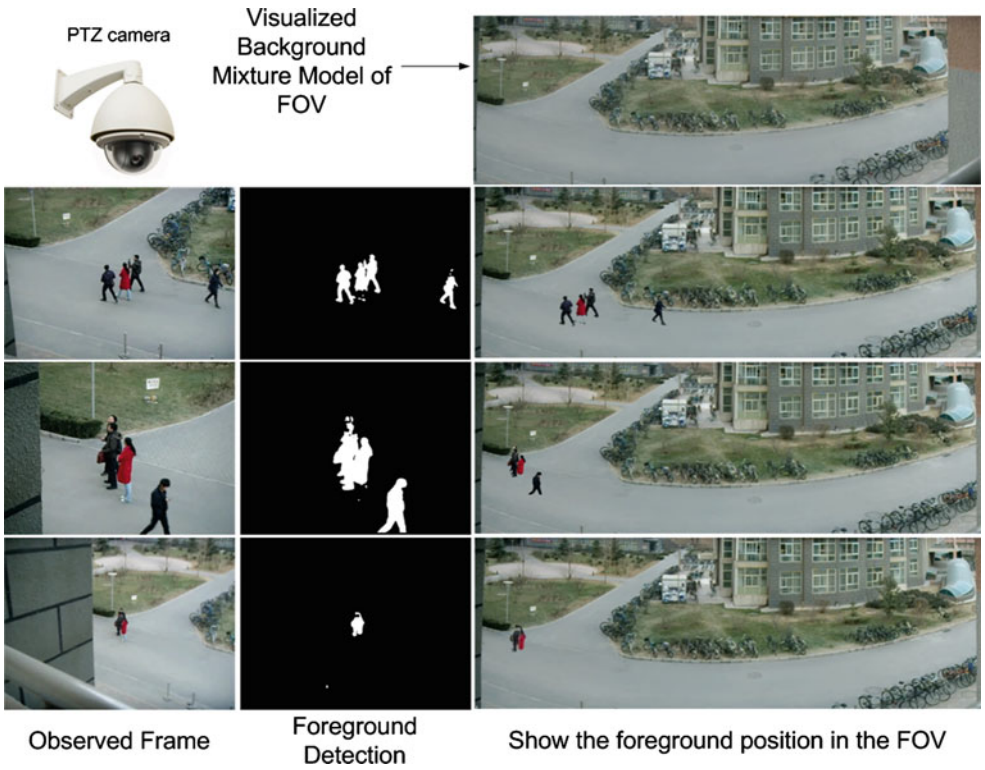


Fig. 1 Example of foreground detection and localization by using our background model

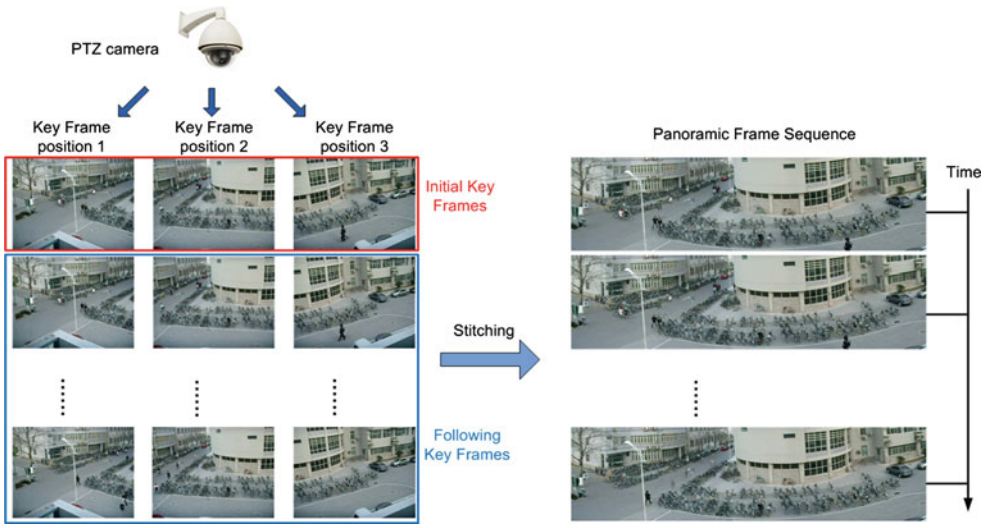


Fig. 2 The PTZ camera first captures a video sequence at the $1st$ key frame position. Then we rotate the PTZ camera to the $2nd$ and the $3rd$ key frame position to capture video sequences. These frames are stitched to get the panoramic frames sequence

determined by the background's texture and each key frame's position is chosen manually during system initialization. These positions are saved so we do not need to keep reselecting the key frames' positions after initialization. In our experiments, the texture of the surveillance area is rich, so we set the overlap region to be about 20 % of one key frame's size. However, if the PTZ camera is capturing an area where the texture is not rich, the overlap region needs to be expanded to extract enough feature points.

In order to decrease the size of panorama, these key frames are collected at the lowest scale of PTZ camera. To decrease the time it takes to generate a panoramic frame, our method calculates the camera's parameters at each initial key frame's position, as shown in Fig. 2. Then, based on the camera's

parameters, we repeat image stitching on the following key frames to generate panoramic frames. Since we do not need to recalculate camera parameters for the following key frames when stitching, this process is fast.

The key problem in generating a panoramic frame is the calculation of the camera's parameters for each key frame's position. Each key frame's camera parameters is described by a vector $P = [\alpha, \beta, f]$, where α is the pin angle, β is the tilt angle and f is the focal length. Since the PTZ camera rotates about its optical center, the group of transformations the key frames undergo is a special group of homographies. This gives pairwise homographies between the ith key frame and the jth key frame:

$$H_{ij} = K_i R_i R_j^T K_j^{-1} \tag{1}$$

where

$$K_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

and

$$R_i = \begin{bmatrix} \cos\alpha_i & 0 & -\sin\alpha_i \\ \sin\alpha_i \sin\beta_i & \cos\beta_i & \cos\alpha_i \sin\beta_i \\ \sin\alpha_i \cos\beta_i & -\sin\beta_i & \cos\alpha_i \cos\beta_i \end{bmatrix} \tag{3}$$

In our method, we use SIFT features [18] for image alignment due to its robustness and distinctiveness. First, we extract SIFT features from all key frames. Then, we use the Fast Approximate Nearest Neighbors (FANN) [19] Search method to accelerate SIFT matching. To refine the solution, once pairwise matches have been established between two key frames, Random Sample Consensus (**RANSAC**) [20] is used to remove the outliers which are not compatible with the homography between these two frames.

After getting a set of geometrically consistent matches between the frames, we use bundle adjustment [21] to solve the key frame position's camera parameters simultaneously by minimizing the sum over all key frames of the residual errors. The bundle adjustment is an essential step because it preserves the multiple constraints between images and it eliminates the errors accumulated from the concatenation of pairwise homographies. The residual errors can be represented as:

$$err = \sum_{i=1}^{n} \sum_{j \in I(i)} \sum_{k \in F(i,j)} ||u_i^k - \widetilde{p}_{ij}^k||^2 \tag{4}$$

$$\widetilde{p}_{ij}^k = H_{ij} u_j^k. \tag{5}$$

Given a correspondence $u_i^k \longleftrightarrow u_j^l$ (u_i^k denotes the position of the kth feature in frame i), where n is the number of frames, $I(i)$ is the set of frames matching frame i, $F(i, j)$ is the set of feature matches between frames i and j. $\widetilde{p}_{ij}^k$ is the projection from frame j to frame i of the point corresponding to u_i^k.

After computing the camera's parameters ($[\alpha, \beta, f]$) of each key frame position, all key frames are aligned according to a series of homographies. The optimized parameter vector of each key frame is saved for future registration. The panorama is generated after rendering by using multi-band blending as described in [18]. Repeating this step, we generate a sequence of panoramic frames. Figure 2 shows the panoramic frame generation sequence.

3.2 Initialization of the PGMM background model

For a stationary camera-based surveillance system, Friedman and Russell [7] proposed to model each background pixel using a mixture of K Gaussians corresponding to road, vehicle and shadow. This model is initialized using an EM algorithm [22]. We follow their approach but modify it for our PTZ camera-based surveillance application. First, we use the panoramic frames to make a Gaussian mixture background model for the surveillance area of the PTZ camera. Second, unlike [7], after we generate enough components for each pixel, we save the remaining pixel values that do not belong to the distribution for future updates.

In our method, the number of Gaussian components of the mth pixel in the background is denoted as K_m. For example, the probability that the mth pixel has a value of $X_{t,m}$ at time t can be written as

$$P(X_{t,m}) = \sum_{i=1}^{K_m} \theta_{i,t,m} \mathcal{N}(X_{t,m}, \mu_{i,t,m}, \Sigma_{i,t,m}) \tag{6}$$

where $\theta_{i,t,m}$ is the weight parameter of the mth pixel's ith Gaussian component. Here $\mathcal{N}$ is the Gaussian probability density function, μ is the mean color vector, Σ is the covariance matrix.

The EM algorithm [23] is used to initialize the weight of $X_{t,m}$, the mean μ, and the covariance matrix Σ. When inputting a new panoramic frame, the EM algorithm updates the weight, mean, and covariance using:

$$\hat{\theta}_{k_m}^{t+1} = \hat{\theta}_{k_m}^{t} + \frac{1}{t+1}(\hat{p}(\omega_{k_m}|X_{t+1,m}) - \hat{\theta}_{k_m}^{t}) \tag{7}$$

$$\hat{\mu}_{k_m}^{t+1} = \hat{\mu}_{k_m}^{t} + \frac{\hat{p}(\omega_{k_m}|X_{t+1,m})}{\sum_{i=1}^{t+1} \hat{p}(\omega_{k_m}|X_{i,m})}(X_{t+1,m} - \hat{\mu}_{k_m}^{t}) \tag{8}$$

$$\hat{\Sigma}_{k_m}^{t+1} = \hat{\Sigma}_{k_m}^{t} + \frac{\hat{p}(\omega_{k_m}|X_{t+1,m})}{\sum_{i=1}^{t+1} \hat{p}(\omega_{k_m}|X_{i,m})} \left((X_{t+1,m} - \hat{\mu}_{k_m}^{t})^T (X_{t+1,m} - \hat{\mu}_{k_m}^{t}) - \hat{\Sigma}_{k_m}^{t} \right) \tag{9}$$

If ω_{k_m} is the kth Gaussian component of the mth pixel in the background, we set $\hat{p}(\omega_{k_m}|X_{t+1,m}) = 1$, otherwise we set $\hat{p}(\omega_{k_m}|X_{t+1,m}) = 0$. We use 600 panoramic frames to initialize the background model. The three components that have the highest weights are designated as the background

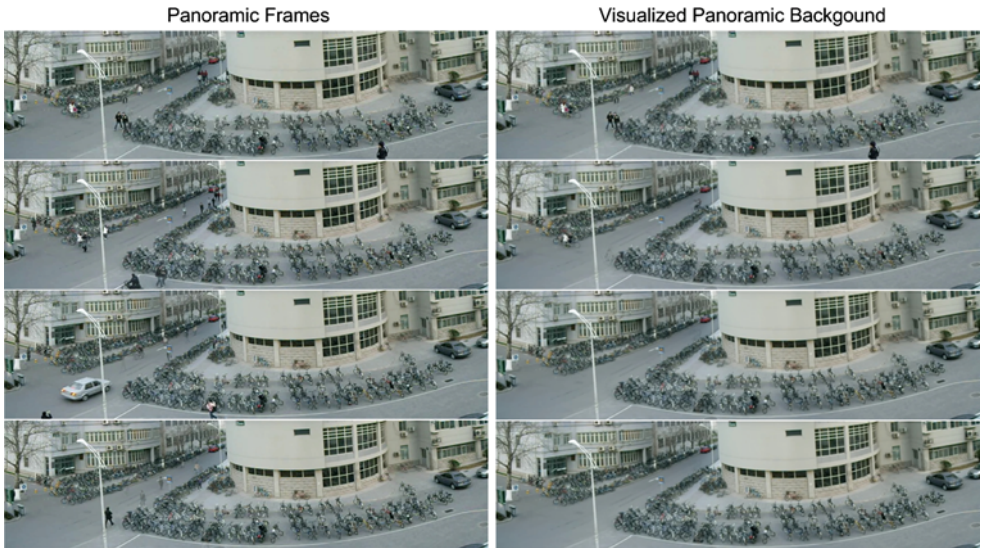


Fig. 3 The *left column* displays the 1st, 150th, 450th and 600th panoramic frame respectively. The *right column* shows the visualized panoramic background after training using 1, 150, 450 and 600 panoramic frames

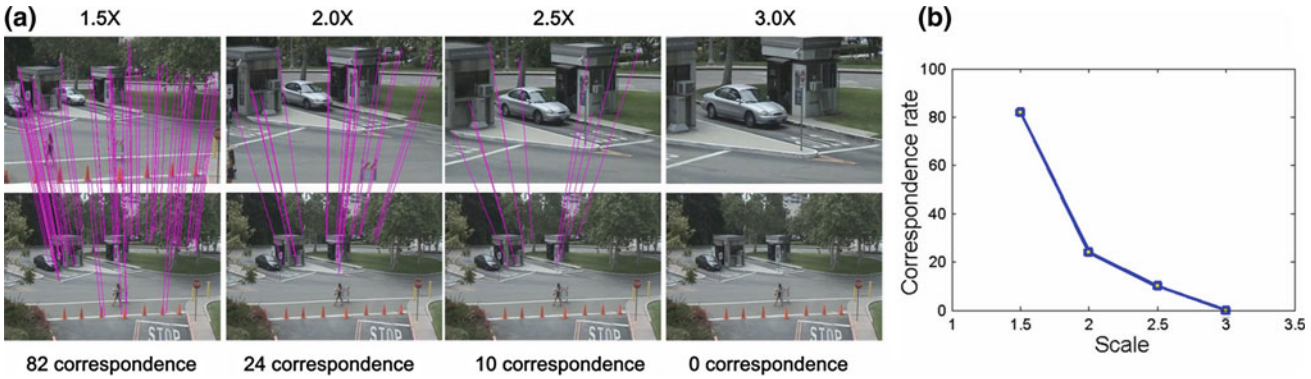


Fig. 4 An example which shows that SIFT features do not work when the scale change is high. **a** shows that as the PTZ camera zooms in, the number of correspondence decreases. **b** shows that the correspondence rate decreases as the scale increases (color figure online)

distribution. An intuitive approach to obtaining a visual representation of the background mixture model, is computing the expected value of the background distribution (assuming $K_m > 3$):

$$X_e = \sum_{i=1}^{3} \frac{\theta_{i,m}\mu_{i,m}}{\sum_{i=1}^{3}\theta_{i,m}} \tag{10}$$

Figure 3 shows a visual representation of the gaussian mixture background model after initialization.

4 Feature-based multi-layered correspondence propagation

After initializing the PGMM, a key problem is the registration of the observed frame to the panoramic background. Estimating camera parameters using a single PTZ camera becomes challenging when the observed frames are captured at different scales. Feature-based detectors like the Difference of Gaussian (DoG) or the Harris–Laplace function [24] are widely used for scale invariance. However, the performance of these scale-invariant detectors depends on the frame size; the larger the frame size, the better the scale invariance. For our application, the frame size is [352,240] so the performance of these traditional detectors in providing large-scale invariance is limited. As shown in Fig. 4, the number of matching feature pairs between the key frame and the test frame decreases as the scale of the test frame increases, so there is not enough correspondence pairs for registration.

Thus, in our paper, we propose a new multi-layered propagation method that works over a large-scale range. Figure 5 shows the structure of Section 4, the propagation method contains two parts: an *off-line step* for generating a layered

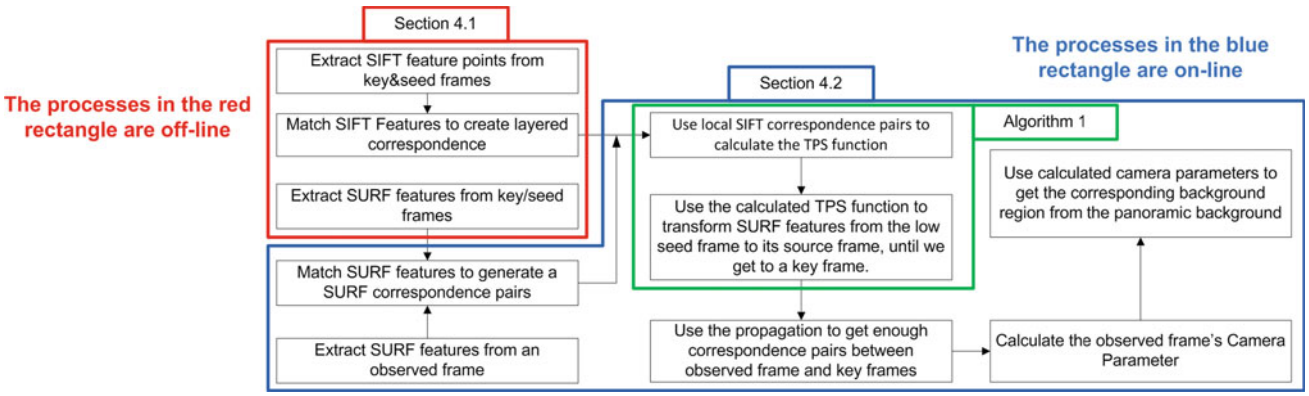


Fig. 5 The structure of *Section 4.1* in the *red rectangle* is off-line. *Section 4.2* in the *blue rectangle* is on-line. Algorithm 1 in the *green rectangle* is our propagation method which is described in 4.2

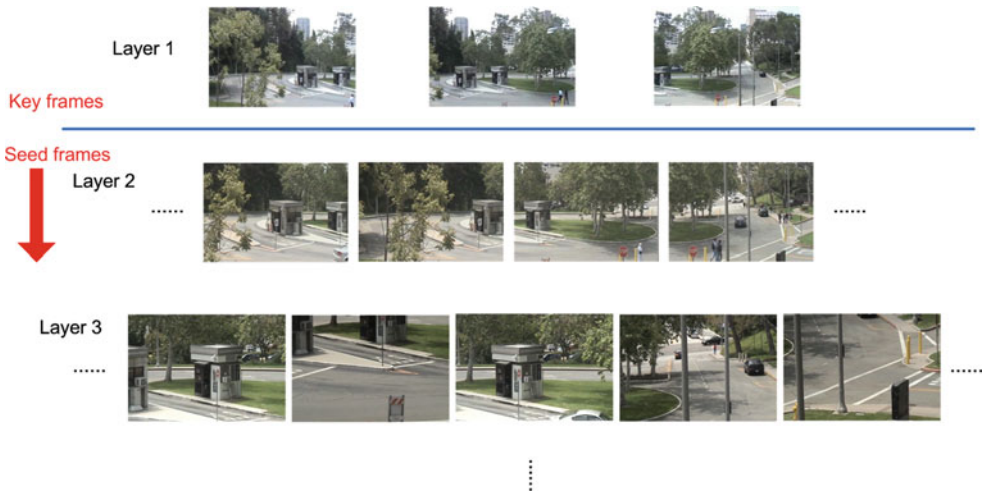


Fig. 6 This figure shows how to capture seed frames for generating multi-layered correspondence. The frames above the blue line are key frames, and the ones under the *blue line* are seed frames (color figure online)

correspondence ensemble (presented in Sect. 4.1), and an *on-line step* for correspondence propagation to register the observed frame with the panoramic background (presented in Sect. 4.2).

4.1 Layered correspondence ensemble

To generate the layered correspondence ensemble, first we capture frames at different scales. As seen in Fig. 6, the frames above the blue line are key frames, and the ones below the blue line are seed frames. All the frames within one layer are captured at the same scale and they have the same characteristics (Sect. 3.1) as key frames. For example in Fig. 6, the frames of Layer 2 are captured at the same scale and the frames of Layer 3 are captured at a different scale which is larger than Layer 2. Each *seed* frame has a **source** frame in a lower scale (e.g. frames in layer 2 are seeds of frames in layer 1, and frames in layer 1 are sources of frames in layer 2, frames in layer 3 are seeds of frames in layer 2, and frames in layer 2 are sources of frames in layer 3).

To build up a layered correspondence ensemble, we use SIFT and SURF features. We extract SIFT features and use them to generate layered correspondence that will be used for calculating the Thin Plate Spline (TPS) transformation [25] on-line because they are more robust and distinctive than SURF so they provide more accurate propagation (described in Sect. 4.2). Since SURF features are faster to extract than SIFT features [26], we use them instead of SIFT features in the on-line registration method.

Unlike [15], the seed frames do not need to be calibrated in order to build up multi-resolution pyramids for image registration since we only need them to generate a layered correspondence ensemble. SIFT features are extracted and used to generate the pairwise correspondence between the seeds and their sources. The right column of Fig. 7 shows some examples of pairwise correspondence. After pairwise correspondence has been generated, we choose sparse features

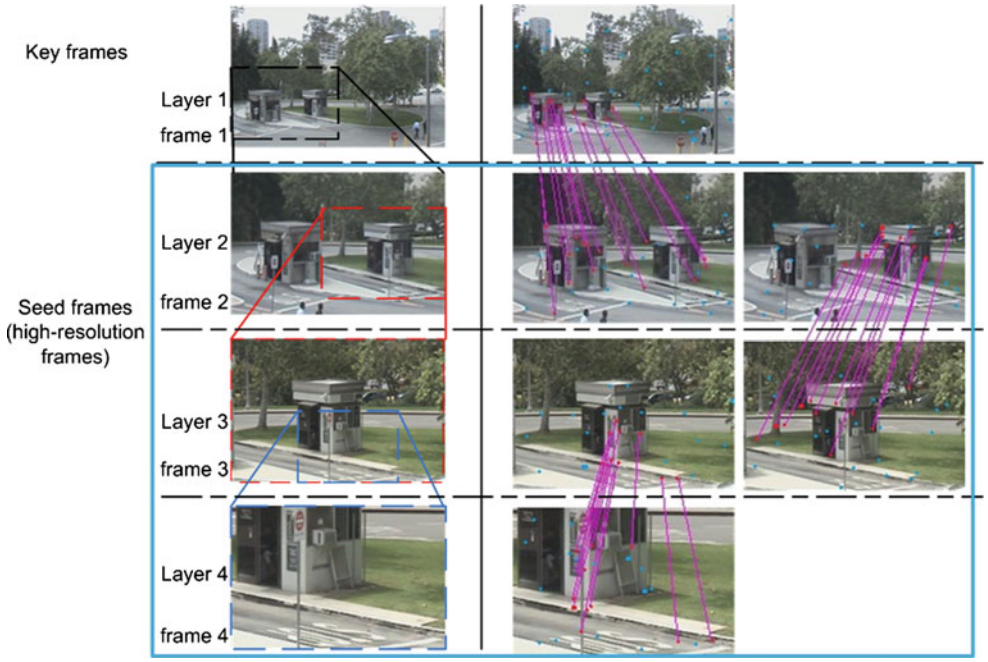


Fig. 7 An example of creating a layered correspondence ensemble (SIFT features (*red points*) and SURF features (*blue points*)). The frames in *blue box* are the seed frames which are captured at different scales by the PTZ camera. The *left column* is the relation between source and seed. The *right column* shows some SIFT feature correspondences (*red line*) between one seed frame and its source frame. The *blue points* in key/seed frame are the SURF feature points (color figure online)

that are evenly distributed to speed up the on-line registration process.

The pairwise correspondence is defined as:

$$\mathbf{C}_{(s,i)} = \bigcup_{\Re(j)} C_{(s,i),(s-1,j)} \tag{11}$$

where $\Re(j)$ is the set of sources corresponding to seed/key frame $I(s,i)$ (i.e. the ith frame in scale s) and $C_{(s,i),(s-1,j)} = \{(x_k^{(s,i)}, y_k^{(s,i)}, x_k^{(s-1,j)}, y_k^{(s-1,j)}), I(s-1,j)\}$ is the corresponding point pair match between $I(s,j)$ and $I(s-1,j)$.

After generating the SIFT-layered correspondence, we extract SURF features from key frames and seed frames to create SURF feature index tree: $index_{(s,j)}$ is the index of frame $I(s,j)$. We save the SIFT-layered correspondence and SURF feature index as the layered correspondence ensemble for each key/seed frame. The ensemble of $I(s,j)$ is defined as:

$$\text{ensemble}_{(s,j)} = \{I(s,i), \mathbf{C}_{(s,i)}, \text{index}_{(s,j)}\} \tag{12}$$

4.2 Correspondence propagation

The aim of correspondence propagation is to find enough correspondence pairs between observed frame and key frames for calculating camera parameters of observed frame. The first step is to extract SURF feature points from the observed frame. As mentioned in the previous Sect. 4.1, the SURF feature points are matched with the saved SURF features of key frames and seeds frames $\prod index_{(s,j)}$ to find their correspondence. These correspondences are used in our propagation algorithm (Algorithm 1) to register the observed frame.

Considering the nonlinear lens distortion, especially when the PTZ camera is working in a large-scale range, we propose a correspondence propagation method based on the TPS transformation. Chui and Rangarajan [27] propose a transition from the standard TPS function to an energy function, that can be minimized. Given two point sets P, and Q in homogeneous coordinates, (i.e., $p_i = (p_{ix}, p_{iy}, 1)$ and $q_i = (q_{ix}, q_{iy}, 1)$) with L (landmark) corresponding points, we find a function f that minimizes the TPS energy function:

$$E_{\text{TPS}} = \sum_{i=1}^{L} \|p_i - f(q_i)\|^2 + \lambda \int\int \left(\left(\frac{\partial f^2}{\partial^2 x}\right)^2 + \left(\frac{\partial f^2}{\partial x \partial y}\right)^2 + \left(\frac{\partial f^2}{\partial^2 y}\right)^2 \right) \mathrm{d}x\mathrm{d}y \tag{13}$$

where $f(q_i)$ is the TPS transformation (or warp) of the points in set Q. f can be decomposed into an affine part and a non-affine part. For any point s in $\Re$, the TPS mapping of s can be written as:

$$f(s) = D \cdot s + \Omega \cdot \phi(s) \tag{14}$$

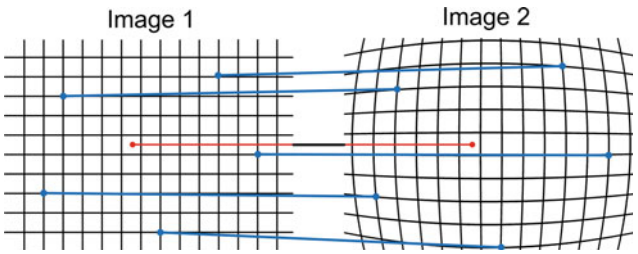


Fig. 8 The *blue points* are known correspondence point pairs between Image 1 (point set P) and Image 2 (point set Q). The *red point* has an unknown correspondence position in Image 2. The TPS function f is used to calculate the *red point*'s correspondence position (color figure online)

where D is a 3×3 matrix of affine transformation parameters and Ω is a $L \times 3$ matrix of non-affine warping coefficients. $\phi(s)$ is a $1 \times L$ vector for any point s,

$$\phi(s) = \begin{bmatrix} \phi_1(s) \\ \phi_2(s) \\ \vdots \\ \phi_L(s) \end{bmatrix} \tag{15}$$

where $\phi_i(s) = c\|s - q_i\|^2 \log \|s - q_i\|, i \in \{1, 2, \ldots, L\}$.

Following the TPS energy function formulation in [28] we minimize:

$$E_{\text{TPS}}(d, w) = \|P - DQ - \Omega\Phi\|^2 + \lambda_1 \text{trace}(\Omega^2 \Phi \Omega) + \lambda_2 \text{trace}[D - I]^2[D - I], \tag{16}$$

λ_1 and λ_2 are regularization parameters for the non-affine and the affine terms respectively. Φ is a $L \times L$ matrix formed from the ϕ's.

Figure 8 illustrates the TPS-based transformation of one point considering nonlinear lens distortion. The blue points are pairwise correspondences between image 1 and image 2. Based on the correspondence set, we calculate the TPS function f from image 1 to image 2, which is used to calculate the red point's unknown position in image 1.

The TPS Algorithm shown in [27] uses all the corresponding point pairs between two images, however we choose only the corresponding point pairs around the point to be transmitted because it's faster. We generate a TPS function for every SURF feature detected in the observed frame. To generate the TPS function, we use SIFT correspondence pairs from the layered correspondence ensemble as the P and Q point sets. To make the TPS function local, we use the closest SIFT features to the SURF feature's location. Using the calculated TPS function, we transform SURF features from a lower seed to its source frame, until we get to highest level, which is a key frame. Algorithm 1 shows the pseudocode for the modified TPS propagation algorithm.

Input: A SURF correspondence point pair between observed frame and a key/seed frame $I(s, i)$, which are denoted as (x, y) and $(x_{(s,i)}, y_{(s,i)})$ respectively
Output: (x, y)'s correspondence point on key frame, (x', y')
for *each SURF feature* **do**
  Search SIFT correspondences $C_{(s,i),(s-1,j)}$ to find the $(x_{(s,i)}, y_{(s,i)})$'s nearest neighbors $\Im(C_{(s,i),(s-1,j)})$;
  $(x_{(s-1,k)}, y_{(s-1,k)}) = f_{TPS}(\Im(C_{(s,i),(s-1,j)}))$;
  if *scale* $s-1$ *is not the key frame* **then**
    $s = s - 1$;
  else
    break;
  end
end

Algorithm 1: The modified TPS-based propagation algorithm.

Using this propagation, we get enough correspondence pairs between the observed frame and the key frames, which allows us to calculate the observed frame's camera parameters using the same method described in Sect. 3.1. After registering the observed frame and projecting it on to the background, we get the corresponding background region from the adaptive panoramic background (Fig. 9).

5 Foreground detection and PGMM update

Foreground detection using pixel-to-pixel correspondence is not very robust because there might be some registration errors due to the propagation method. Instead we use the minimum distance between the pixel and a block centered on the correspondence background pixel to generate the probability of being foreground. As shown in Fig. 10, this provides more accurate results than pixel-to-pixel correspondence. The Mahalanobis distance is incorporated into our distance function to determine if a pixel falls into a distribution in the background model. Our distance function Dis is:

$$Dis(X_{t+1}||W(X)) = \arg\min|| \sqrt{(X_{t+1} - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_{t+1} - \mu_{i,t})} - m\sigma_{i,t}|| \tag{17}$$

where $W(X)$ is the block centered on X_{t+1}'s correspondence background pixel after propagation, $i \in \{1, 2, \ldots, 9\}$ is the pixel's number in the block. We use a 3×3 block, and m is a constant threshold equal to 2.5. The foreground probability is:

$$P(X_{t+1}|W(X)) = \exp(Dis_M(X_{t+1}||W(X))) \tag{18}$$

If $P(X_{t+1}|W(X))$ is above a threshold, it is declared as a foreground pixel. If the pixel's declared as background, its corresponding pixel's distribution in the background is

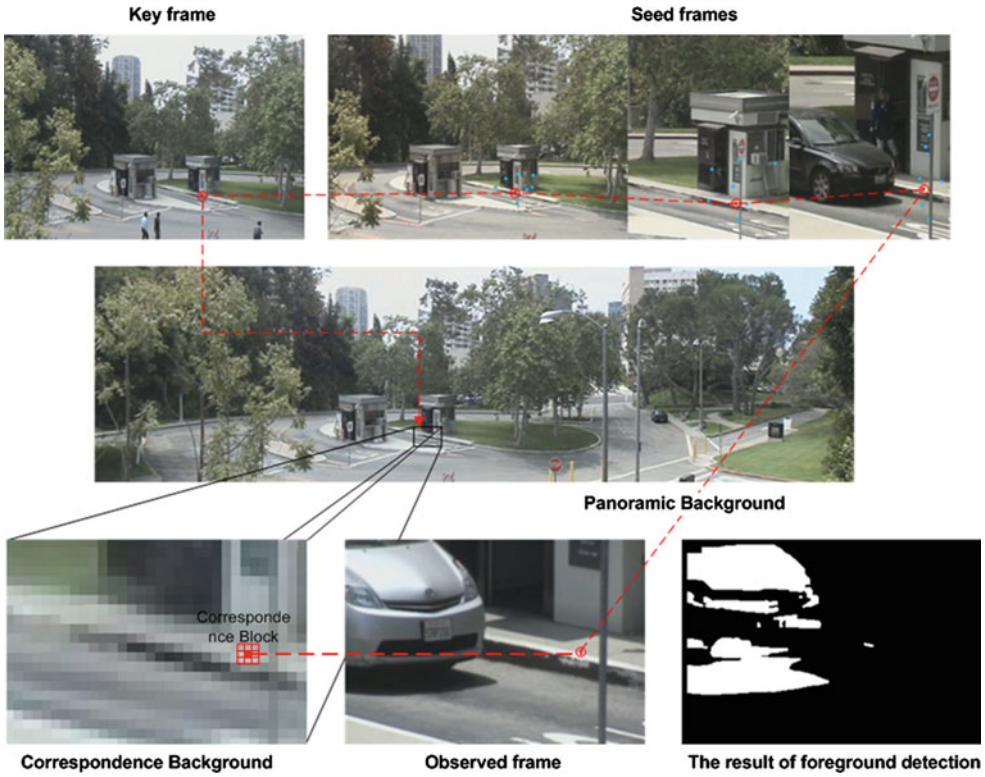


Fig. 9 The *red point* is a SURF feature point of observed frame, the *blue points* around it are the supporters for propagation. After finding enough correspondence between current observed frame and key frame, the correspondence background can be extracted from the panoramic background. The red 3 × 3 block in the correspondence background is the SURF feature's correspondence block which is used for foreground detection (color figure online)

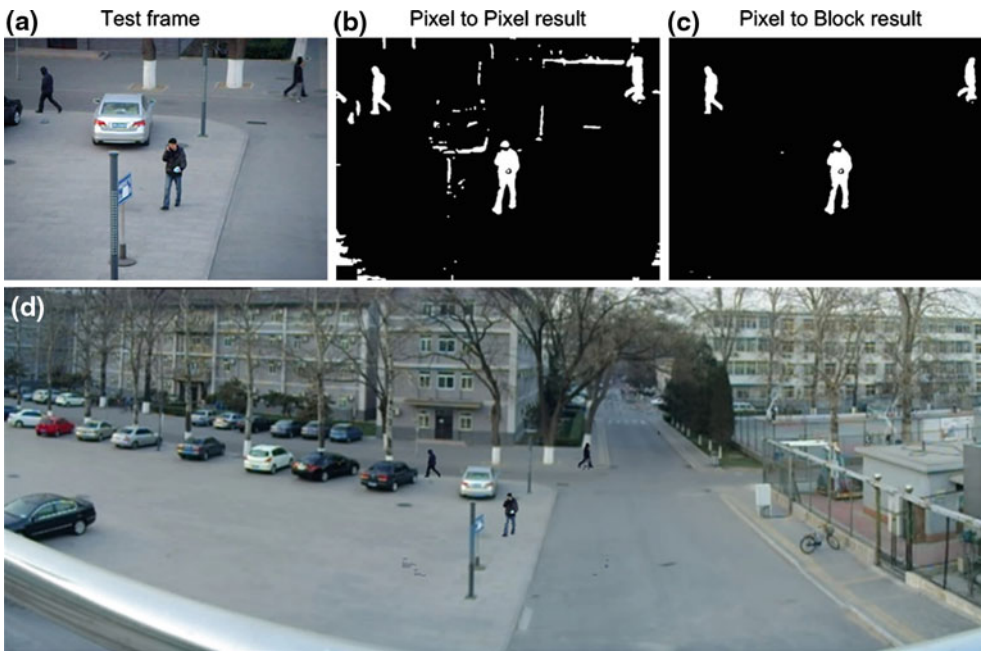


Fig. 10 **a** is a test frame. **b** is the result of foreground detection using the pixel-to-pixel method. **c** is the result which uses block information centered on the correspondence pixel. **d** is the current panoramic frame we built which shows the foreground on the visualized panoramic background to show the foreground objects' position in the FOV

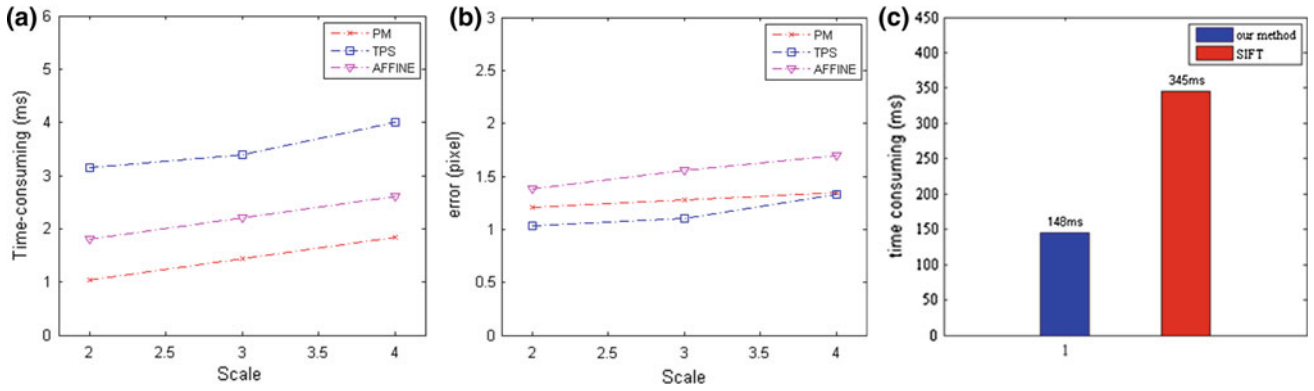


Fig. 11 **a** and **b** the comparison in time-consuming and point-error, respectively; *PM* is short for our propagation method, *TPS* is short for traditional TPS transformation and AFFINE denotes affine transformation. **c** the average time-consuming of our method and SIFT to register same observed frames to the panoramic background

updated as follows:

$$\theta_{k,t+1} = (1-\alpha)\theta_{k,t} + \alpha \tag{19}$$

$$\mu_{k,t+1} = (1-\rho)\mu_{k,t} + \rho X_{t+1} \tag{20}$$

$$\sigma^2_{k,t+1} = (1-\rho)\sigma^2_{k,t} + \rho(X_{t+1} - \mu_{k,t+1})(X_{t+1} - \mu_{k,t+1})^T \tag{21}$$

where α is a constant learning rate, we set $\alpha = 0.0016$ in our method (manually set based on experimental results), $\rho = \alpha\mathcal{N}(X_{t+1}, \mu_k, \Sigma_k)$.

If the pixel is declared as foreground, we check to see if it falls into other components of the mixture model which are not background components. If it falls into a component, then that component is updated as defined above. Otherwise, the pixel with the smallest weight is discarded, and the last component of its distribution is initialized to this pixel's value. Once the foreground objects have been detected, any tracking algorithm, such as mean-shift, can be used to track a target.

6 Experiments

We demonstrate the speed of our correspondence propagation method and evaluate its accuracy compared to the TPS and affine transformation. We also demonstrate its speed compared to SIFT to justify using SURF features. Then, we test the efficiency of our adaptive panoramic multi-layered background modeling method for foreground detection. For our experiments, we use a Core2 2.66 GHz PC with 2G RAM and a PTZ camera whose focal length is in the 32 mm to 360 mm range, which allows a 12× zoom. Our test sequences contain frames with spatial resolution 352 × 240.

6.1 Evaluation of the correspondence propagation method

The evaluation of our correspondence propagation method, which uses a modified TPS transformation, includes two parts. First, we evaluate the correspondence propagation algorithm's speed and accuracy, compared with the original TPS transformation and affine transformation. Second, we compare the registration speed of our propagation method with the registration speed using SIFT without layered correspondence. To evaluate the propagation methods, we generate ground truth correspondence pairs from a key frame to a seed frame. We extract SIFT features and generate layered correspondence.

We use this layered correspondence to propagate the ground truth point on the seed frame to its calculated correspondence pixel in the key frame using our modified TPS, original TPS, and affine transformation. Then we compare the speed and accuracy of these methods. We take 10 correspondence pairs from 5 different key frames at 3 different scales. We take an average of the time it takes to register each set of 10 pixels to each key frame at each scale. We calculate the registration error of the different methods by comparing the calculated correspondence pixels to the ground truth using the Euclidean distance. As seen in Fig. 11, our modified TPS function is faster than the other two methods, and the error is slightly greater than the original TPS but it is better than affine.

To show that the multi-layered correspondence propagation method, which uses SURF features, provides faster registration to the key frame than SIFT, we measure the average time it takes to extract SURF features from the observed frame and register to the key frame using our multi-layered correspondence propagation. We compare it to the average time it takes to extract SIFT feature points from the observed frame and register it to the key frame without using multi-layered correspondence propagation. We use 50 frames to

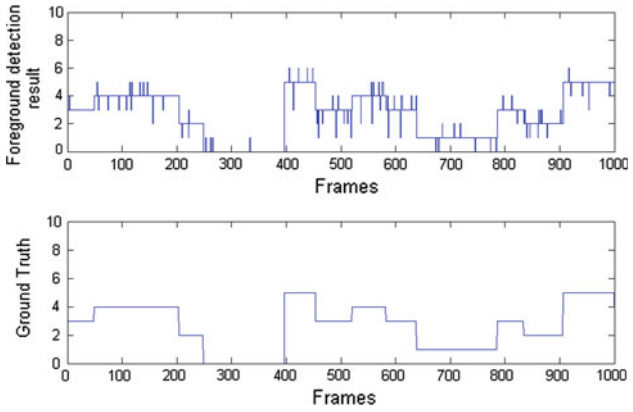


Fig. 12 Foreground detection results, the *Y* axis means the number of foreground in one frame. Several people in one crowd are considered as one foreground object

calculate the average time. As seen in Fig. 11c, the average time of our method is 148 ms per frame and SIFT is 345 ms per frame. Our method takes less than half the time it takes using SIFT.

6.2 The performance of background model

To test the accuracy of our background model, we perform foreground detection, as described in Sect. 5, and compare the results to ground truth data which is generated manually. We use four different scenes for evaluation. Figure 12 shows the accuracy of our foreground detection results. Figure 13 shows several results of foreground detection using the adaptive panoramic and large-scale range background model.

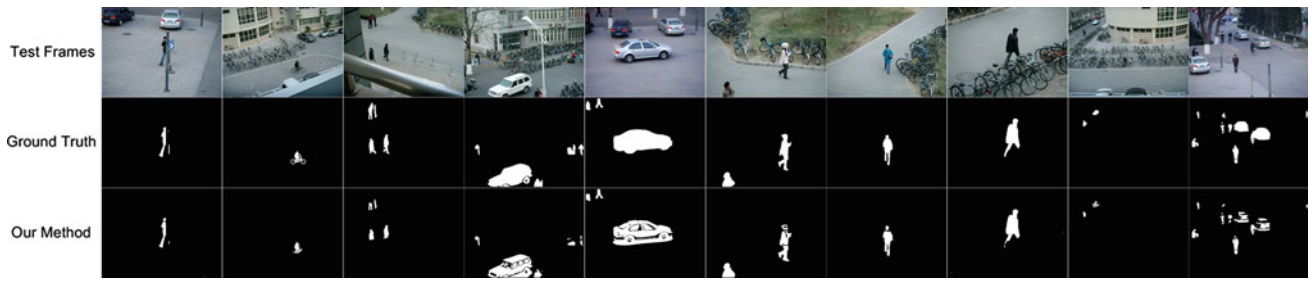


Fig. 13 Example of foreground detection results

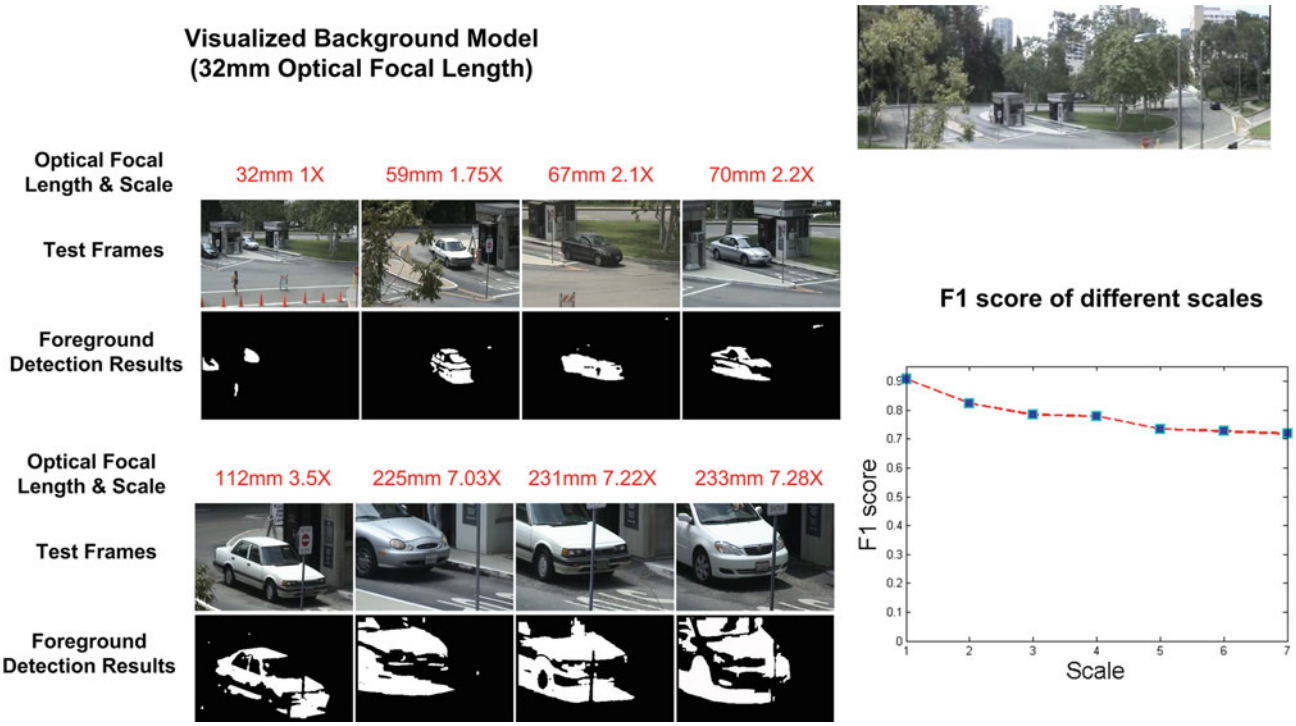


Fig. 14 Example of our method's capacity to work in a large-scale range. The scale of the test current frame increases as the focal length increases. The F1 scores of our method's results at different scales are also showed. Scale means the optical focal length of PTZ camera (color figure online)

Fig. 15 Compare with the method in [13], the *second row* is our method's result and the *third row* is the [13]'s results. Because of the foregrounds existing on the background (1,2,3 *red rectangle*), there are more false negative and false positive in the detection results of [13]'s method. The *red rectangles* in the binary images shows the effect caused by the foreground on the backgrounds panorama (color figure online)

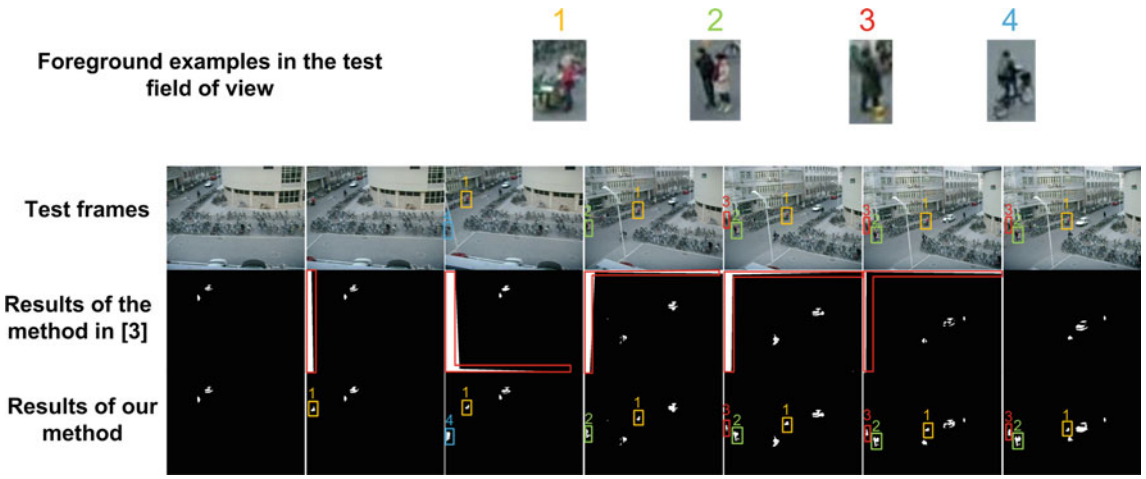


Fig. 16 Compare with the method in [10], the *second row* is the result of the method in[10] and the *third row* is our results. When the PTZ camera rotates, there will be a new monitor region (inside of the *red line* in the result of [10]) cannot be detected using method of [10]. Meanwhile, the foreground not moving during this period cannot be detected either (for instance target 1, 2 and 3). Our model has the capacity to deal the problem of method in [10]

Figure 14 shows that our background model works well over a large-scale range. In an optical system, such as a PTZ camera, the scale changes when the focal length changes, which can be presented by:

$$\text{Scale} = \frac{f_{\text{current}}}{f_0} \tag{22}$$

where f_{current} is the current frame's focal length and f_0 is the minimum focal length. In our experiment, the system's $f_0 = 32\,\text{mm}$ which is used to capture key frames for generating the panoramic background.

Figure 17 shows another feature of our background model: the ability to combine the foreground detection results with the panoramic background to localize the foregrounds' position in the monitored area.

After detecting the foreground position from a observed frame, a tracking algorithm can be used to track targets. For our experiments, we use a mean-shift tracker. The tracker only uses the region obtained from foreground detection. Figure 18 shows some tracking results, in different scales, on outdoor sequences with a large field of view. Figure 19 shows the trajectory of the target on the panoramic background, using tracking results acquired from the mean shift tracker.

6.3 Comparison with existing methods

We compare our method to a frame-to-global method from [13] and a frame-to-frame method from [10] by testing their performance for foreground detection. The method proposed in [13] randomly selects frames from the PTZ camera's FOV

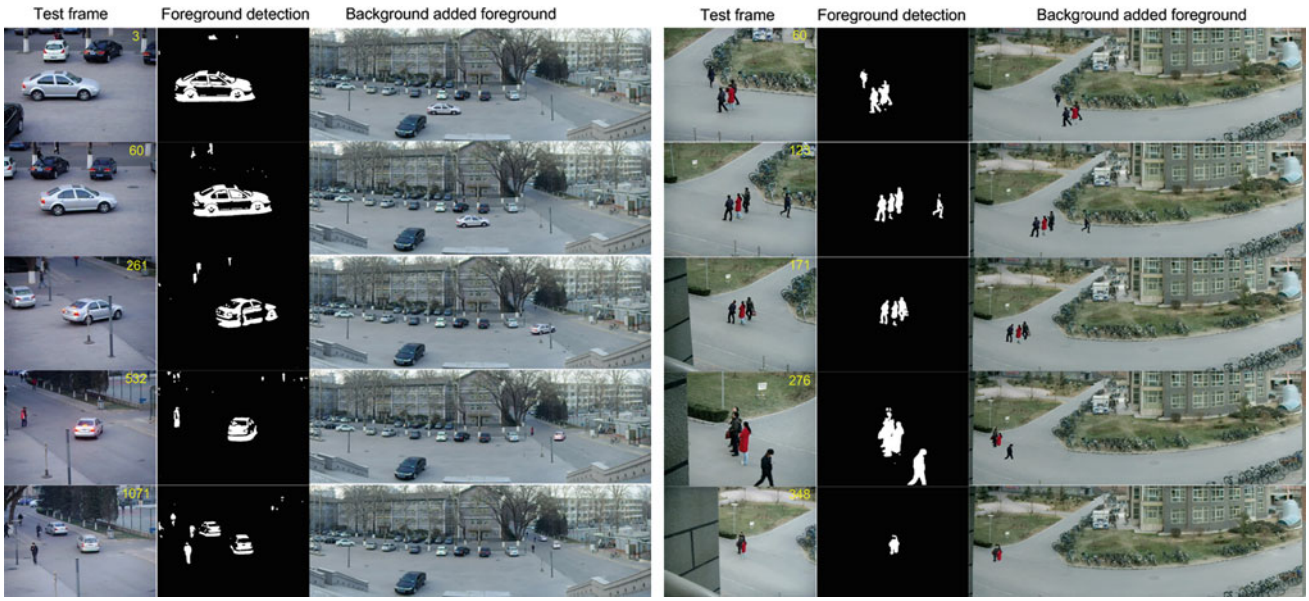


Fig. 17 The figure shows some results of generating background added with foreground to show the global position of them

Fig. 18 The tracking results of our system

Fig. 19 The trajectory of the targets

to generate a panorama. These frames sometime include foreground objects which cause false negatives or false positives when performing foreground detection on an observed frame, as shown in Fig. 15. Since we use GMMs to generate our panoramic background model, our method has the capacity to eliminate foreground objects within the key frames, thereby allowing for more robust foreground detection in observed frames compared to the method in [13].

Figure 16 shows the results of our method compared to the frame-to-frame method presented in [10]. As the PTZ camera pans or tilts, newly appeared areas in the observed frame need to be trained in method presented in [10]. The method fails to detect foreground objects within these new areas for several frames until training is complete. If there are stationary foreground objects during the training period, the method will determine such objects to be background and does not detect them correctly, creating false negatives; when the foreground objects move, it creates false positives. Although, compared to [10], our initial training period is longer, because we generate a PGMM for the PTZ camera's complete FOV, the PGMM, however, allows us to solve the frame-to-frame problem when the camera changes its view rapidly, since it provides global information for the background. Meaning, our method can detect foreground object

Table 1 Comparison between PTZ camera-based background substraction methods in three scenes

Scenario	Our method	Method in [13]	Method in [10]
First	0.8802	0.6240	0.7644
Second	0.9243	0.6022	0.7814
Third	0.8925	0.7451	0.7250

The first, second and third column represent the F1 score of the three methods' results respectively. Each row is related to a scenario

accurately in the new observed frame without any training time. We detect foreground objects faster than [10] because we do not need to train on the new view.

Table 1 shows the F1 score comparison of the foreground detection results in three different scenarios between our method, [13], and [10]'s. Figure 14 shows our method's performance as the scale changes. Since the existing methods do not have the capacity to work within a large-scale range (PTZ camera's focal length changes rapidly) like our method, we cannot make a fair scale change comparison.

7 Conclusion

In this paper, we proposed a novel background subtraction method for PTZ camera-based surveillance systems. The method is divided into three parts: *Generation of the Panoramic Gaussian Mixture Background Model (PGMM)*, *Feature-based Multi-layered Correspondence Propagation*, and *Foreground Detection and PGMM update*. The PGMM provides global information for the camera's FOV. The multi-layered correspondence propagation method registers observed frames to the background and obtains the correspondence background even when the scale of the PTZ camera changes rapidly. Using the correspondence background, foreground objects are detected and the PGMM is updated using the observed frames.

Acknowledgments The authors would like to thank Professor Songchun Zhu from UCLA, Professor Patricio A. Vela from Georgia Tech and Dr. Liang Lin from Sun Yat-Sen University for their help. This work is supported by the National Natural Science Foundation of China (60827003, 60903070), National Science and Technology Major Project (2012ZX03002004) and the Innovation Team Development Program of the Chinese Ministry of Education (IRT0606).

References

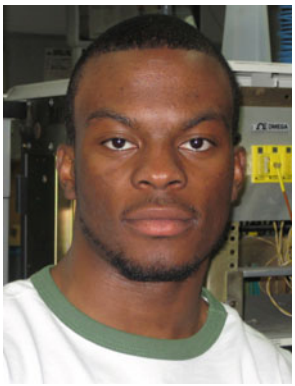
1. Darvish, P., Varcheie, Z., Bilodeau, G.-A.: People tracking using a network-based PTZ camera. In: Machine Vision and Applications (2010) (Sept. 2010)
2. Xu, Y., Song, D.: Systems and algorithms for autonomous and scalable crowd surveillance using robotic PTZ cameras assisted by a wide-angle camera. In: Auton Robot (2010) vol. 29, pp. 53–66. (Apr. 2010)
3. Heikkila, M., Pietikanen, M.: A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Mach. Intell. **28**(4), 657–662 (2006)
4. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE **90**(7), pp. 1151–1163 (2002)
5. Hu, W., Gong, H., Zhu, S.-C., Wang, Y.: An integrated background model for video surveillance based on primal sketch and 3D scene geometry. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008 (2008)
6. Kim, K., Chalidabhongse, T., Harwood, D., Davis, L.: Background modeling and subtraction by codebook construction, ICIP 2004 (2004)
7. Friedman, N., Russell, S.: Image segmentation in video sequences: A probabilistic approach. UAI, pp. 175–181 (1997)
8. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. CVPR, pp. 246–252 (1999)
9. Helmut, G., Horst, B.: On-line boosting and vision. In: Computer Vision and Pattern Recognition (CVPR) (2006)
10. Kang, S.P., Joonki, K.A., Abidi, B., Mongi, A.A.: Real-time video tracking using PTZ cameras. In: Proc. of SPIE 6th International Conference on Quality Control by Artificial Vision, Gatlinburg, TN, vol.5132, pp. 103–111, May (2003)
11. Wu, S., Zhao, T., Broaddus, C., Yang, C., Aggarwal, M.: Robust Pan, Tilt and Zoom Estimation for PTZ Camera by Using Meta Data and/or Frame-to-Frame Correspondences. In: Control, Automation, Robotics and Vision (2006)
12. Micheloni, C., Foresti Gian, L.: Real-time image processing for active monitoring of wide areas. J. Vis. Commun. Image Represent. **17**(3), 589–604 (2006)
13. Azzari, P., Di Stefano, L., Bevilacqua, A.: An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. In: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 511–516 (Sept. 2005)
14. Bevilacqua, A., Azzari, P.: High-quality real time motion detection using ptz cameras. In: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1923 (Nov. 2006)
15. Sinha, S.N., Pollefeys, M.: Pan–tilt–zoom camera calibration and high-resolution mosaic generation. Comput. Vis. Image Understand. **103**(3), 170–183 (2006)
16. Chen, C.-H., Yao, Y., Page, D., Abidi, B.R., Koschan, A., Abidi, M.: Heterogeneous fusion of omnidirectional and PTZ cameras for multiple object tracking. IEEE Trans Circuits Syst Video Technol **18**(8), 1052–1063 (2008)
17. Chen, C.-C., Yao, Y., Drira, A., Koschan, A., Abidi, M.: Cooperative mapping of multiple PTZ cameras in automated surveillance systems. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR (2009)
18. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **20**, 91–110 (2003)
19. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal (Feb 2009)
20. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography Tech report 213, AI Center, SRI International (1980)
21. McLauchlan, P., Jaenicke, A.: Image mosaicing using sequential bundle adjustment. Image Vis. Comput. **20**(9–10), 751–759 (2002)
22. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (Methodological) **39**(1), 1–38 (1977)

23. KaewTraKulPong, P., Bowden, R.: An improved adaptive background mixture model for real-time tracking with shadow detection. In: Proc. European Workshop Advanced Video Based Surveillance Systems (2001)
24. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
25. Bookstein, F.L.: Principal warps: thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal. Mach. Intell. **11**(6), 567–585 (1989)
26. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features European Conference on Computer Vision, pp. 7–13. Graz, Austria May 2006
27. Chui, H., Rangarajan, A.: A new algorithm for non-rigid point matching. In: IEEE Conference on ComputerVision and Pattern Recognition (CVPR), vol. 2, pp. 44–51 (2000)
28. Capel, D., Zisserman, A.: Automated mosaicing with super-resolution zoom. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 885–891, Santa Barbara (1998)

Author Biographies

Kang Xue received his B.S. degree in Optical Electronics from Beijing Institute of Technology in 2006. He is now a Ph.D. candidate in Optical Engineering at School of Optics and Electronics, Beijing Institute of Technology and co-advised at School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include computer vision, augmented reality and machine learning.

Yue Liu is a Professor of Optical Engineering at the School of Optics and Electronics, Beijing Institute of technology. He received his Ph.D. in Telecommunication and Information System from Jilin University, Jilin Province, China in 2000 and his M.S. degree in Telecommunication and Electronic system from Jilin University of Technology, Jilin Province, China in 1996. His current research interests include human computer interaction, virtual and augmented reality, accurate tracking of the pose of camera, 3D display system and camera calibration etc.



Gbolabo Ogunmakin received his B.S. degree in Electrical and Computer Engineering from Morgan State University in 2008. He is currently pursing a Ph.D. degree in Electrical and Computer Engineering at Georgia Institute of Technology. His research interests include computer vision, event detection, action recognition, video surveillance, target tracking and re-identification.

Jing Chen was born in 1974. She received her Ph.D. degrees of engineering from Beijing Institute of Technology in 2002. She is now an associate professor of the School of Optoelectronics, Beijing Institute of Technology. Her main research interests are in computer vision, augmented reality and image detection and tracking.

Jiangen Zhang received his B.S. degree in Information and Computing Sciences from the Minzu University of China, in 2006 and his M.S. degree in Operational Research and Cybernetics from Beijing Jiaotong University, China, in 2008. He is currently pursuing a Ph.D. degree in Computer Science at Beijing Institute of Technology. His main research interests are computer vision and machine learning, with a current specific focus on event recognition and inferring social roles of agents.