L_0 Regularized Stationary-Time Estimation for Crowd Analysis

Shuai Yi, Xiaogang Wang, *Member, IEEE*, Cewu Lu, *Member, IEEE*, Jiaya Jia, *Senior Member, IEEE*, and Hongsheng Li

Abstract—In this paper, we tackle the problem of stationary crowd analysis which is as important as modeling mobile groups in crowd scenes and finds many important applications in crowd surveillance. Our key contribution is to propose a robust algorithm for estimating how long a foreground pixel becomes stationary. It is much more challenging than only subtracting background because failure at a single frame due to local movement of objects, lighting variation, and occlusion could lead to large errors on stationary-time estimation. To achieve robust and accurate estimation, sparse constraints along spatial and temporal dimensions are jointly added by mixed partials (which are second-order gradients) to shape a 3D stationary-time map. It is formulated as an L_0 optimization problem. Besides background subtraction, it distinguishes among different foreground objects, which are close or overlapped in the spatio-temporal space by using a locally shared foreground codebook. The proposed technologies are further demonstrated through three applications. 1) Based on the results of stationary-time estimation, 12 descriptors are proposed to detect four types of stationary crowd activities. 2) The averaged stationary-time map is estimated to analyze crowd scene structures. 3) The result of stationary-time estimation is also used to study the influence of stationary crowd groups to traffic patterns.

Index Terms—Stationary-time estimation, stationary crowd analysis, crowd video surveillance

1 Introduction

In large cities with high population densities, the assembly of large crowds in public areas, such as train stations and shopping malls, causes major concerns on public safety and transportation efficiency. Crowd analysis in video surveillance attracts considerable attention and has important applications in crowd management and traffic control [1], [2], [3], [4], [5], [6].

By estimating traffic flow and predicting crowd behaviors, crowd analysis can be used to detect abnormal crowd behaviors and control traffic to avoid congestion. It also provides valuable information for public space design in order to achieve maximal space usage and to increase robustness to crowd gathering.

Existing works focus on detecting motion patterns of crowds [1], [3], [4], [5] and analyzing the interactions between moving pedestrians [6], [7], [8], [9]. However, stationary crowds, which are able to provide surprisingly rich information for scene analysis and modeling, were not sufficiently studied in the literature.

1.1 Stationary Groups in Crowd Analysis

Stationary group is one of the basic elements in crowd scene modeling. People stay in a scene for a longer time for certain reasons (which are often of security interest), and usually have more influence on traffic patterns than those passing through the scene quickly. Therefore, detection and analysis of stationary groups provide useful information for crowd scene understanding and leads to interesting applications in crowd surveillance.

First of all, emergence, dispersal, stationary duration, and status of stationary groups may incur great security interest. From these detected activities, we can discover valuable information, such as relation of people and possible abnormality. Fig. 1 shows four activities to be detected in this paper. They are group gathering, group stopping-by, group relocating, and group deformation, respectively. For example, in group gathering, the members could have friendship or share the same goal.

Second, stationary groups change traffic flow and decrease traffic efficiency. Previous works mainly model the global motion patterns of pedestrians based on scene structures (e.g., entrances, exits, walls, and roads) and the interactions between individual moving pedestrians. However, studies [10], [11], [12] showed that stationary groups might have a greater impact on changing traffic patterns than moving pedestrians in some scenarios. When people move around, they adjust speed but not direction to avoid collisions with other moving pedestrians. Such self-organized behaviors keep traffic flow efficient. However, if stationary groups exist, other moving pedestrians might be forced to change walking directions to avoid them. As shown in Fig. 2, the emergence and dispersal of stationary groups might cause the dynamic variations of crowd traffic patterns. It is thus of great importance to incorporate stationary groups into dynamic scene modeling. Moreover, stationary groups decrease traffic efficiency as pedestrians need to walk longer way to

- *S. Yi, X. Wang, and H. Li are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: {syi, xgwang, hsli}@ee.cuhk.edu.hk.*
- *C. Lu and J. Jia are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: {cwlu, leojia}@cse.cuhk.edu.hk.*

Manuscript received 18 May 2015; revised 23 Oct. 2015; accepted 22 Apr. 2016. Date of publication 28 Apr. 2016; date of current version 10 Apr. 2017. Recommended for acceptance by G. Mori.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2560807

Fig. 1. Four major types of stationary group activities to be detection based on our proposed stationary-time estimation algorithm. (a) People join a group from different directions at different times. When all people arrive, the whole group moves to the same destination. (b) A group of people enters the view together, stay for a period of time, and leave together. (c) After staying at a place for a while, people move to another location and become stationary again. (d) People in a group have their own activities, taking photos for example.

bypass stationary group regions and special attention should therefore be paid to these regions.

Last, stationary groups help us better understand scene structures. It is informative to investigate where stationary groups are likely to emerge and how long they tend to stay. An average stationary-time map is shown in Fig. 3. It provides guidance for crowd management, as well as provision of facilities and support.

1.2 Stationary-Time Estimation

All the above mentioned applications rely on the estimation of *stationary-time*, i.e., the period of time since a pixel becomes stationary foreground for the same object. As shown in Fig. 4, our method produces a 3D stationary-time map in the spatio-temporal space for an input video sequence. This is different from the map calculated by background subtraction, where each pixel is either 0 or 1. We have experimented with simply detecting foreground at individual frames and computing how long a pixel has been in the foreground. The result is usually poor. We thus treat the estimation of stationary-time as a new challenge. As demonstrated by the applications in Section 6 and our recent work [13], it is an important step for further analysis on stationary crowds.

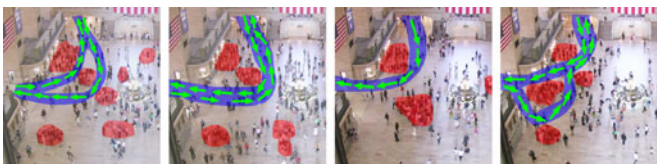


Fig. 2. The emergence and dispersal of stationary groups might cause dynamic variations of traffic patterns. Stationary groups and main traffic flows are marked in red and blue.

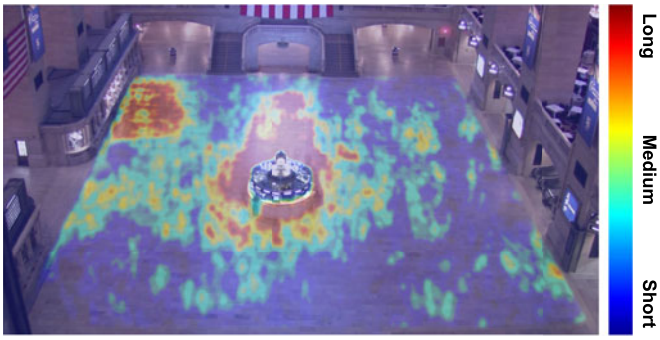


Fig. 3. Averaged stationary-time distribution over 4 hours of a train station scene. Stationary groups tend to emerge and stay long around the information booth and in front of the ticketing windows.

1.3 Challenges of Stationary-Time Estimation

Stationary-time estimation is able to provide more information than background subtraction and more difficulties arise in the meanwhile. Fig. 5 illustrates the inherent challenges. 1) Background subtraction does not distinguish between

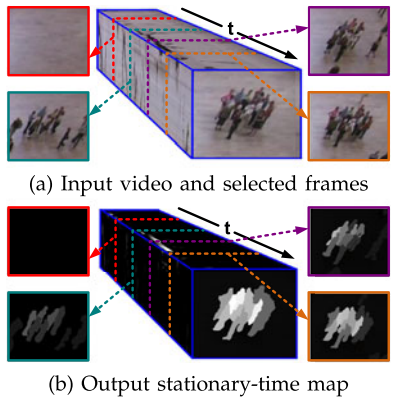


Fig. 4. Estimating a 3D stationary-time map for a video sequence. Results from a few frames are shown. The period of time since each pixel has been stationary up to each frame is represented by the intensity level. Brighter pixels correspond to longer stationary-times.

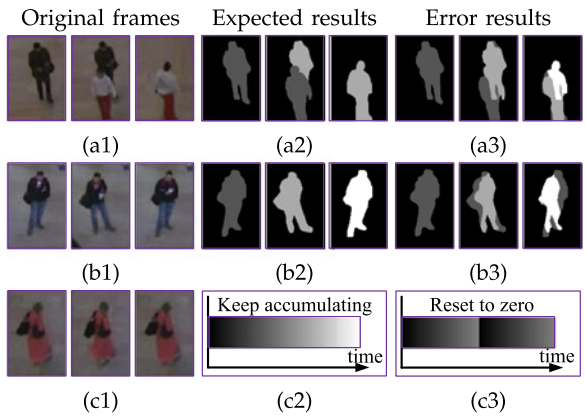


Fig. 5. Challenges of stationary-time estimation. Three example cases show that results from background subtraction are erroneous. (a) Two foreground objects with spatio-temporal overlap. (b) Local movement of objects also leads to estimation errors. (c) If a foreground pixel is misclassified as background in one frame, stationary-time resets to 0, which is wrong. In (c3), mis-classification happens in the middle, making time reset.

different foreground objects. If two objects overlap, the estimated stationary-time could be longer than what it should be in the overlapping region. This phenomenon happens frequently in crowd scenes and is illustrated in Fig. 5a. 2) People's local movements are common during the stationary period and the stationary-times of foreground pixels with local movements should be accumulated, instead of being frequently reset to 0. This challenge is illustrated in Fig. 5b. However, matching locally moving foreground objects especially in crowd scenes is not easy. 3) Most background subtraction methods do not take temporal consistency into account. If a foreground pixel is misclassified at one frame, stationary-time could be mistakenly reset to 0 which leads to large under-estimation of stationary-time. This challenge is illustrated in Fig. 5c. Given all these challenges coupled together, none of the existing approaches is ready to solve the problem of stationary-time estimation.

1.4 Method Overview and Main Contributions

A robust stationary-time estimation algorithm is proposed. Given a video clip, all pixels are encoded as one of the multiple foreground codewords or as the background. The foreground codebook and the encoding of foreground pixels are jointly optimized by minimizing the reconstruction cost of foreground regions. To distinguish foreground regions from the background during the encoding process, the rough result of background subtraction is used to guide the encoding process.

As the encoding result might be quite noisy, sparse constraints in both spatial and temporal dimensions are jointly added to encourage the spatio-temporal consistency of the encoding result. The sparse constraint is formulated as the L_0 norm of second order spatio-temporal gradients, which is much more powerful in regularization than the commonly used local smoothness prior applied to image and temporal spaces separately.

A joint optimization pipeline is adopted to alternatively optimize for the encoding cost and the sparse constraint. Optimization is performed on a batch of frames instead of individual ones. This process is robust to occasional local movements of stationary objects, occlusions, and mis-classification. Stationary-times of foreground pixels belonging to different codewords are accumulated separately to generate the final estimation result.

Our contributions are summarized into the following three aspects. 1) A robust stationary-time estimation algorithm is proposed, and it is a basic step for stationary crowd analysis. A novel guided foreground encoding term and an L_0 sparse prior term are proposed to solve the new challenges arising in stationary-time estimation. An optimization pipeline is introduced to solve the highly non-convex problem by alternatively solving a series of sub-problems. 2) Several novel applications based on stationary-time estimation are introduced. 2.a) Twelve new crowd descriptors are proposed to detect four stationary group activities as illustrated in Fig. 1. 2.b) The average stationary-time map can be used to help understand crowd scene structures as shown in Fig. 3. 2.c) The influence of stationary groups on traffic patterns can be studied based on detected stationary groups and the clustered traffic flows, as shown in Fig. 2. 3) A dataset with annotated ground truth is provided to the public for stationary-time estimation and stationary group activity analysis, which is the first in its kind.

2 Related Works

A straightforward solution to stationary-time estimation is to accumulate time of foreground pixels detected by background subtraction methods. The adaptive Gaussian mixture model [14] is one of the popular approaches and it was improved by Zivkovic [15]. Kim et al. [16] modeled complex background variations with a codebook. The Bayesian background subtraction method [17] employs joint features of color and location, and performs nonparametric density estimation to handle local movements on background. Challenges of using these approaches have been discussed in Section 1. Robust PCA [18] separates foreground objects and background as a sparse matrix and a low rank matrix. It is not suitable for this estimation task as foreground pixels with long stationary-time are very likely to be classified as background.

For other possible solutions, keypoint tracking [19], tracking-by-detection [20], optical flow estimation [21], [22], and pedestrian detection [23], [24] cannot generate satisfactory results because of their unreliable performance in crowded scenes, which is demonstrated by our experiments in Section 5.2.

There are significant amount of works on crowd motion analysis. Lagrangian coherent structures [1], Lie algebra representation [25] and topic models [3], [26], [27], [28], [29] have been widely used to model crowd motion patterns. Social force models [30], [31] can be used for pedestrian simulation [32], tracking [7], interaction analysis [8], and abnormal event detection [6] in crowd. All these works target on moving pedestrians. Stationary groups, although can also provide valuable information, are lack of attention in existing research works.

It is of interest to detect social groups and analyze their activities [12], [33], [34], [35], [36], [37]. Cristani et al. [38] studied the interactions of standing people in a sociological view. Other works along this line mainly considered moving groups. Pedestrians were grouped based on their relative distances and the similarities of moving patterns [33], [35], [39]. Various features and models were proposed to recognize different mobile group behaviors [36], [40], [41], [42], [43]. As discussed in Fig. 1, stationary groups have their own characteristics and special features are needed to characterize their activities and properties.

3 Stationary-Time Estimation

In this section, we introduce an optimization based algorithm to estimate the stationary-time of all pixels in color video, which is defined as the period of time since each pixel becomes stationary foreground for the same object. To achieve this goal, the problem is converted to encoding all pixels into either one of the multiple foreground codewords or the background. The objective function of the foreground encoding process consists of two terms, a guided foreground encoding term that jointly optimizes a foreground codebook and all pixels' encoding results (Section 3.1), and a sparse gradient prior term that effectively encourages the spatio-temporal consistency of the encoding results (Section 3.2).

The stationary-time can then be easily calculated for each foreground codeword separately (Section 3.4). A long video sequence is divided into short clips with overlap, such that information of codewords and stationary-time can be consistent across clips.

3.1 Guided Foreground Encoding

Given a video clip, encoding foreground pixels aims at simultaneously looking for an optimal foreground codebook $\mathbf{D}$ and determining which codeword in $\mathbf{D}$ each foreground pixel belongs to. It is achieved by minimizing reconstruction cost of replacing foreground pixels with assigned foreground codewords. In the remaining of this paper, "codeword" refers to "foreground codeword".

Each foreground pixel or codeword is associated with a 5D feature vector. Let p be a general pixel. The feature vector of p is written as $I_p = [R_p, G_p, B_p, X_p, Y_p]^T$, where $[R_p, G_p, B_p]$ and $[X_p, Y_p]$ are the RGB values and the spatial coordinates of p. Each feature channel is independently normalized to $[0, 1]$ to indicate the same importance. Let $\{\mathbf{d}_1, \ldots, \mathbf{d}_M\}$ ($\mathbf{d}_i \in \mathbb{R}^{5\times 1}$ for $i = 1, \ldots, M$) represent M codewords that form the codebook matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_M] \in \mathbb{R}^{5\times M}$. These codewords can be regarded as M cluster centers of the input foreground pixels. Pixels belonging to the same codewords denote that they belong to the same stationary part. The encoding result of p is represented by an M-dimensional binary vector $\boldsymbol{\alpha}_p \in \{0, 1\}^M$ and all pixels' encoding results are denoted as $\boldsymbol{\alpha}$. The entries of $\boldsymbol{\alpha}_p$ can only be 1 or 0, and at most one element can be 1. If all the entries are 0, p is labeled as background and $\|\boldsymbol{\alpha}_p\|_1 = 0$. If the ith element is 1, p is assigned with codeword $\mathbf{d}_i$ and $\|\boldsymbol{\alpha}_p\|_1 = 1$. As our goal is to distinguish foreground individuals with M codewords, the codebook size M should be no smaller than the number of pedestrians.

The encoding can be achieved by minimizing the cost of reconstructing foreground regions using codewords,

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \sum_{\{p|\|\boldsymbol{\alpha}_p\|_1=1\}} \|\mathbf{D}\boldsymbol{\alpha}_p - I_p\|_2^2. \quad (1)$$

The codebook $\mathbf{D}$ is initialized by k-means clustering, and is then jointly optimized with the encoding result $\boldsymbol{\alpha}_p$ for all foreground pixels. Note that I_p contains the spatial coordinates of p. Optimizing the reconstruction of spatial coordinates of foreground pixels implicitly encourages pixels in a local region to share the same codeword.

There exists a trivial solution to the foreground encoding term (1), which simply labels all the pixels as background. A guidance term is needed to encourage the resulting foreground regions to be similar to the result of background subtraction. A background subtraction result [16] is adopted to guide foreground encoding,

$$\min_{\boldsymbol{\alpha}} \sum_p (\|\boldsymbol{\alpha}_p\|_1 - u_p)^2, \quad (2)$$

where u_p denotes p's background subtraction result.

By combining (1) and (2), the guided foreground encoding term is denoted as $\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha})$, which balances the foreground encoding errors and the deviation from the rough background subtraction result,

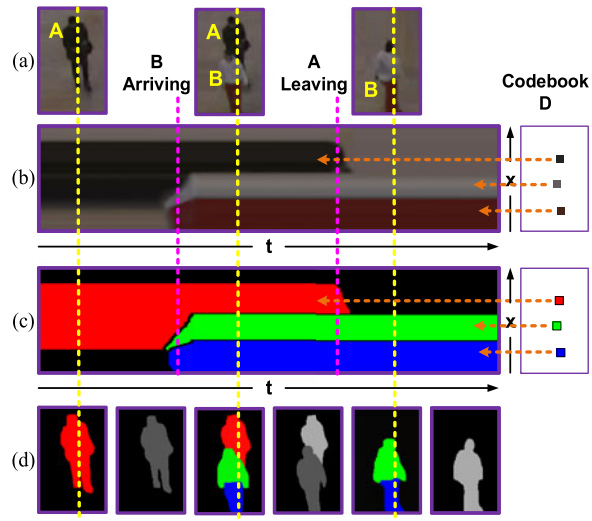


Fig. 6. Illustration of the foreground encoding that separates foreground objects close or overlapped in the spatio-temporal space. (a) 3 frames from the same region. After person B arrives, A leaves. (b) Temporal slice image along the yellow line, where A and B overlap. (c) Foreground pixels assigned with three different codewords. They are well separated. (d) Foreground codewords (colored) and estimated stationary-time (gray-scale) of input frames. The learned codebook $\mathbf{D}$ with $M = 3$ are shown on the right. Each codeword $\mathbf{d}_i$ is represented by one rectangle. The R, G, B color values are shown as the colors of the rectangles while the X, Y coordinates are illustrated as the locations of the rectangles.

$$\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha}) = \sum_{\{p|\|\boldsymbol{\alpha}_p\|_1=1\}} \|\mathbf{D}\boldsymbol{\alpha}_p - I_p\|_2^2 + \eta \sum_p (\|\boldsymbol{\alpha}_p\|_1 - u_p)^2, \quad (3)$$

where η is a parameter indicating the confidence on u_p. $\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha})$ is minimized w.r.t. the foreground codebook $\mathbf{D}$ and the encoding results of all pixels $\boldsymbol{\alpha}$.

Fig. 6 shows one example of our encoding result where multiple codewords are assigned to different foreground regions. With $M = 3$, the learned five-dimensional codeword vectors are visualized in Fig. 6. The corresponding foreground segmentation result is shown in Figs. 6b, 6c, and 6d.

By sharing codewords in local regions, the under-estimation errors caused by local movements of foreground objects (shown in Fig. 5b) can be effectively eliminated. By clustering foreground pixels into different codewords, different pedestrians or body parts can be well separated even they occlude each other in the spatio-temporal space as shown in Fig. 6c. After the encoding result is generated by our algorithm, stationary time of different codewords can be accumulated separately (detailed in Section 3.4). Whenever a new codeword is generated, its stationary time is accumulated from zero. The over-estimation in the overlapping regions can be avoided as shown in Figs. 5a and 6d.

3.2 Sparse Gradient Prior

The stationary-time of a foreground pixel p increases if it stays with the same encoding result $\boldsymbol{\alpha}_p$. Due to lighting variation, local movement, and occlusion, the estimation of $\boldsymbol{\alpha}_p$ could be quite noisy if using $\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha})$ alone. The estimated stationary-times might be constantly reset to 0, as shown in Fig. 5c. We observed that the change of $\boldsymbol{\alpha}_p$ on ideal stationary objects should be very sparse. We accordingly impose a sparse gradient prior $c(\boldsymbol{\alpha})$ to eliminate noise and maintain spatio-temporal consistency,

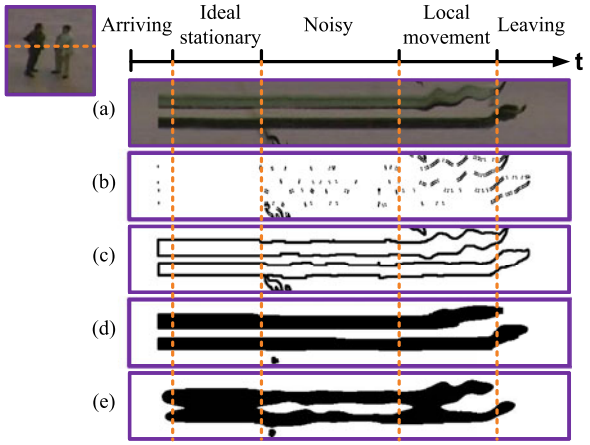


Fig. 7. By utilizing the sparse prior, we estimate $\boldsymbol{\alpha}$ better from noisy and/or locally moving objects. (a) Stages of two pedestrians arriving, staying, locally moving, and leaving (horizontal-axis: time; vertical-axis: scanline pixels highlighted by the orange dashed line). (b) Pixels with non-zero $\partial_{x,t}$ values. (c) Pixels with non-zero ∂_x+∂_t values. (d) Our foreground encoding result with $\partial_{x,t}$ in (b). (e) Erroneous encoding result with ∂_x+∂_t in (c).

$$c(\boldsymbol{\alpha}) = \#\{p|\ ||\partial_{x,t}\boldsymbol{\alpha}_p||_2 + ||\partial_{y,t}\boldsymbol{\alpha}_p||_2 \neq 0\}, \tag{4}$$

where $\partial_{x,t}$ and $\partial_{y,t}$ are the second-order gradients w.r.t. $x-t$ and $y-t$ space derivatives. If the current pixel p is indexed by the spatial and temporal coordinates (x_p, y_p, t_p), its second order gradient can be calculated numerically as $\partial_{x,t}\boldsymbol{\alpha}_p = [\boldsymbol{\alpha}_{x_p,y_p,t_p} - \boldsymbol{\alpha}_{x_p,y_p,t_p-1}] - [\boldsymbol{\alpha}_{x_p-1,y_p,t_p} - \boldsymbol{\alpha}_{x_p-1,y_p,t_p-1}]$, and $\partial_{y,t}\boldsymbol{\alpha}_p = [\boldsymbol{\alpha}_{x_p,y_p,t_p} - \boldsymbol{\alpha}_{x_p,y_p,t_p-1}] - [\boldsymbol{\alpha}_{x_p,y_p-1,t_p} - \boldsymbol{\alpha}_{x_p,y_p-1,t_p-1}]$. $\#$ counts the number of nonzero values in the mixed partials.

3.2.1 L_0 Norm Sparse Constraint

The sparse constraint is formulated as an L_0 norm term, which has unique properties in gradient domain compared with L_1 norm.

Non-zero L_0 gradients of $\boldsymbol{\alpha}_p$ denote changes of encoding results along spatial and temporal dimensions while zero L_0 gradients of $\boldsymbol{\alpha}_p$ denote invariance of encoding results. For the stationary time estimation problem, we need to minimize the number of encoding changes along spatial and temporal dimensions to regularize the solution, which is quite suitable to be modeled by the L_0 norm sparse constraints.

Moreover, L_0 norm globally regularizes the number of non-zero gradients and all non-zero gradients of $\boldsymbol{\alpha}_p$ share the same importance. However, the cost of L_1 norm loss function increases if the gradient magnitude is large. Mathematically, L_1 norm satisfies positive scalability constraint $L_1(ax) = |a| \cdot L_1(x)$, which indicates L_1 norm penalizes more on larger gradient magnitudes. Some noises with small gradients cannot be removed which may lead to frequent changes of encoding results and large stationary time estimation errors. Contrarily, L_0 norm satisfies $L_0(x) = L_0(ax)$ with any non-zero a. It is preferable for the stationary-time estimation problem as L_0 norm penalizes changes of $\boldsymbol{\alpha}_p$ equally regardless of the magnitudes of the changes.

In addition, various solvers [44] are proposed to solve different computer vision problems with L_0 norm constraints and decent results were achieved.

3.2.2 Second-Order Gradients

To enforce spatio-temporal consistency of the foreground encoding result, a simple prior incorporating first-order gradients along each dimension may be used,

$$c'(\boldsymbol{\alpha}) = \#\{p|\ ||\partial_x\boldsymbol{\alpha}_p||_2 + ||\partial_y\boldsymbol{\alpha}_p||_2 + ||\partial_t\boldsymbol{\alpha}_p||_2 \neq 0\}. \tag{5}$$

We compare this prior with that in (4) to show second-order gradients are more effective. In (5), any nonzero values in x, y, or t gradients result in nonzero c'. When calculating c', a stationary person produces the result shown in Fig. 7c, where all body boundaries inevitably produce many nonzero values. When using c' as a prior for regularization, all these boundary pixels will be regularized, which is not our intention.

There is no such problem in (4). Nonzero c' caused by spatial boundaries would be eliminated if the object is stationary when calculating second-order gradients $\partial_{x,t}$ and $\partial_{y,t}$ for those pixels. As shown in Fig. 7b, only a few *moving boundary* pixels yield nonzero c. Thus only penalizing these pixels would result in a very sparse encoding result, robust to noise and outliers. We compare the final results of our system by using these two priors respectively in Figs. 7d and 7e, and observe that the second order gradient is effective to produce reasonable encoding results for foreground pixels.

3.3 Joint Objective Function

Eqs. (3) and (4) are integrated to a joint objective function,

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \{\mathcal{Q}(\mathbf{D},\boldsymbol{\alpha}) + \lambda c(\boldsymbol{\alpha})\}, \quad s.t.\ \ \boldsymbol{\alpha}_p = \{0,1\}^M, \|\boldsymbol{\alpha}_p\|_1 \leq 1. \tag{6}$$

The data term $\mathcal{Q}(\mathbf{D},\boldsymbol{\alpha})$ produces M mid-level semantic codewords from hundreds of intensity levels, which lead to robust stationary-time estimation against local movements. The prior $c(\boldsymbol{\alpha})$ captures the structural sparsity for each codeword of stationary objects in the spatio-temporal space. It guarantees the stability of $\boldsymbol{\alpha}_p$ and avoid frequent change of $\boldsymbol{\alpha}_p$ even for a large M.

3.4 Pixel-Wise Stationary-Time Estimation

Stationary-time can be estimated based on the change of $\boldsymbol{\alpha}$. If the foreground codeword of a pixel is $\mathbf{d}_i$ starting from frame t_1, and it is changed to a different codeword $\mathbf{d}_j$ or background at frame t_2, its stationary-time is $t_2 - t_1$. If a pixel is changed from background to a foreground codeword, it locally searches for a pixel with the same codeword in previous frames. If such a matched pixel is found, its stationary-time will be inherited by the current pixel, instead of counting from zero. This avoids under-estimation caused by foreground local movements, including waving hands, looking around, turning around, and some other body movements, which are quite common and frequent during the stationary period.

If a frame is close to the boundary of a video clip, estimation is not reliable. We use overlapping video clips with shared buffer frames. Only estimated stationary-time in frames outside the buffer is kept for reliability's sake. If an object stays longer than the duration of a clip, the foreground codewords are matched across clips using the overlapping

part so that the stationary-times can continue in accumulation. Codebook $\mathbf{D}$ is dynamically updated to track the change of background.

4 Optimization

$\mathbf{D}$ and $\boldsymbol{\alpha}$ in (6) are coupled and optimization is highly non-convex. A set of axillary vectors $\boldsymbol{\alpha}_p^0 \in \mathbb{R}^M$ are introduced to relax the original problem as

$$\begin{aligned} &\min_{\mathbf{D},\boldsymbol{\alpha},\boldsymbol{\alpha}^0} \left\{ \mathcal{Q}(\mathbf{D},\boldsymbol{\alpha}^0) + \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 + \lambda c(\boldsymbol{\alpha}) \right\}, \\ &s.t.\ \boldsymbol{\alpha}_p = \{0,1\}^M, \|\boldsymbol{\alpha}_p\|_1 \le 1, \boldsymbol{\alpha}_p^0 = \{0,1\}^M, \|\boldsymbol{\alpha}_p^0\|_1 \le 1. \end{aligned} \tag{7}$$

When β_1 is large, $\boldsymbol{\alpha}_p^0$ approaches $\boldsymbol{\alpha}_p$. It makes the challenging problem boil down to two sub-ones. Satisfactory results are achieved by solving the two sub-problems iteratively (Sections 4.1 and 4.2) and increasing β_1 after each iteration. This strategy was used in [44] and proved effective to solve L_0 gradient minimization problems.

4.1 Solve for $\mathbf{D}$ and $\boldsymbol{\alpha}_p^0$

With $\boldsymbol{\alpha}_p$ fixed, the sparse prior term is a constant and can therefore be omitted. The first sub-optimization problem of (7) can be written as

$$\begin{aligned} &\min_{\mathbf{D},\boldsymbol{\alpha}^0} \left\{ \mathcal{Q}(\mathbf{D},\boldsymbol{\alpha}^0) + \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 \right\}, \\ &s.t.\ \ \boldsymbol{\alpha}_p^0 = \{0,1\}^M, \|\boldsymbol{\alpha}_p^0\|_1 \le 1. \end{aligned} \tag{8}$$

Similar to k-means, $\mathbf{D}$ and $\boldsymbol{\alpha}^0$ are estimated iteratively. Given $\boldsymbol{\alpha}^0$, $\mathbf{D}$ is obtained by solving a least square problem. Given $\mathbf{D}$, $\boldsymbol{\alpha}_p^0$ can be obtained by naively pixel-wise searching $(M+1)$ possibilities of foreground codewords and background.

4.2 Solve for α_p

Given $\mathbf{D}$ and $\boldsymbol{\alpha}_p^0$ fixed, the second sub-problem is

$$\min_{\boldsymbol{\alpha}} \left\{ \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 + \lambda c(\boldsymbol{\alpha}) \right\}. \tag{9}$$

The constraint that $\boldsymbol{\alpha}_p = \{0,1\}^M$ is first omitted and then added back using thresholding after $\boldsymbol{\alpha}_p$ converges. (9) is non-convex. We further employ axillary vectors $\mathbf{h}$ and $\mathbf{v}$ to approximate $\partial_{x,t}\boldsymbol{\alpha}$ and $\partial_{y,t}\boldsymbol{\alpha}$ in a similar way as (7), which yields

$$\begin{aligned} \min_{\boldsymbol{\alpha},\mathbf{h},\mathbf{v}} \Bigg\{ & \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 + \lambda c(\mathbf{h},\mathbf{v}) \\ & + \beta_2 \sum_p \left(\|\partial_{x,t}\boldsymbol{\alpha}_p - \mathbf{h}_p\|_2^2 + \|\partial_{y,t}\boldsymbol{\alpha}_p - \mathbf{v}_p\|_2^2 \right) \Bigg\}. \end{aligned} \tag{10}$$

$c(\mathbf{h},\mathbf{v}) = \#\{p | \|\mathbf{h}_p\|_2^2 + \|\mathbf{v}_p\|_2^2 \neq 0\}$. We solve (10) again with two sub-optimization problems (Sections 4.2.1 and 4.2.2) iteratively in the same way as solving (7).

4.2.1 Solve for $(\mathbf{h},\mathbf{v})$

Given $\boldsymbol{\alpha}$, (10) is equivalent to

$$\begin{aligned} (\widehat{\mathbf{h}},\widehat{\mathbf{v}}) = \arg\min_{\mathbf{h},\mathbf{v}} \Bigg\{ & \lambda c(\mathbf{h},\mathbf{v}) + \beta_2 \sum_p \|\partial_{x,t}\boldsymbol{\alpha}_p - \mathbf{h}_p\|_2^2 \\ & + \beta_2 \sum_p \|\partial_{y,t}\boldsymbol{\alpha}_p - \mathbf{v}_p\|_2^2 \Bigg\}. \end{aligned} \tag{11}$$

(11) is independent on p, thus can be pixel-wisely solved,

$$(\widehat{\mathbf{h}}_p,\widehat{\mathbf{v}}_p) = \begin{cases} (\mathbf{0},\mathbf{0}) & \text{if} \quad \lambda/\beta_2 \ge \|\partial_{x,t}\boldsymbol{\alpha}_p\|_2^2 + \|\partial_{y,t}\boldsymbol{\alpha}_p\|_2^2 \\ (\partial_{x,t}\boldsymbol{\alpha}_p, \partial_{y,t}\boldsymbol{\alpha}_p) & \text{elsewhere} \end{cases}.$$

Here $(\mathbf{h}_p,\mathbf{v}_p)$ is used to approximate $(\partial_{x,t}\boldsymbol{\alpha}_p, \partial_{y,t}\boldsymbol{\alpha}_p)$. In this step, we need to decide whether to preserve the change of $\boldsymbol{\alpha}_p$. If $\lambda/\beta_2 \ge \|\partial_{x,t}\boldsymbol{\alpha}_p\|_2^2 + \|\partial_{y,t}\boldsymbol{\alpha}_p\|_2^2$, the change of $\boldsymbol{\alpha}_p$ would be removed and $(\widehat{\mathbf{h}}_p,\widehat{\mathbf{v}}_p)$ would be set as zero. Otherwise, the change would be preserved and $(\widehat{\mathbf{h}}_p,\widehat{\mathbf{v}}_p)$ would be set as the original second order gradient $(\partial_{x,t}\boldsymbol{\alpha}_p, \partial_{y,t}\boldsymbol{\alpha}_p)$.

With a larger λ, more non-zero gradients are set to zeros. It is because the optimization has a higher weight on the sparsity constraint and the changes of $\boldsymbol{\alpha}_p$ along temporal and spatial dimensions are more likely to be removed. With a larger $\|\partial_{x,t}\boldsymbol{\alpha}_p\|_2^2 + \|\partial_{y,t}\boldsymbol{\alpha}_p\|_2^2$, the gradient of $\boldsymbol{\alpha}_p$ is less likely to be set to zero. It is because the ground truth change of $\boldsymbol{\alpha}_p$ is significant and removing the change may lead to large error. β_2 also influences the final encoding results. Initially, β_2 is small, the changes at more locations are allowed to be removed. After several iterations, β_2 is large, the removal of the changes of $\boldsymbol{\alpha}_p$ becomes more difficult. In this way, we can keep the optimization process more stable and the optimization would eventually converge with some large β_2 value.

4.2.2 Solve for α

Given $(\mathbf{h},\mathbf{v})$, (10) is equivalent to the following quadratic optimization problem with a closed-form solution:

$$\begin{aligned} \widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \Bigg\{ & \beta_1 \sum_p \| \boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0 \|_2^2 \\ & + \beta_2 \sum_p \left(\|\partial_{x,t}\boldsymbol{\alpha}_p - \mathbf{h}_p\|_2^2 + \|\partial_{y,t}\boldsymbol{\alpha}_p - \mathbf{v}_p\|_2^2 \right) \Bigg\}. \end{aligned} \tag{12}$$

4.3 Discussions on Convergence

We propose to solve (9) with alternative relaxing and rounding, which is common for many binary optimization problems [45]. In the experiments, we observe that the numerical optimization results of $\boldsymbol{\alpha}_p$ are quite close to the allowable states (1 or 0) even without the binary constraints. Thus we can adopt this alternative optimization strategy in our framework. After several iterations, a decent solution is yielded. $\boldsymbol{\alpha}_p$ is recovered, and binary constraints are satisfied in the meanwhile.

The weights β_1, β_2, and β_3 indicate the relaxation degree and are initialized as 1. Initially, a loose relaxation (small β) is used to avoid local minimum. Afterwards, we gradually reduce the relaxation degree (increase β) to approximate

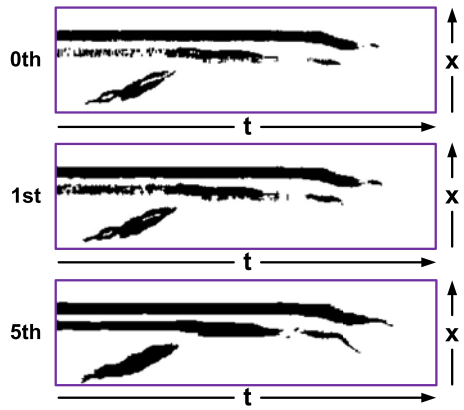


Fig. 8. Illustration of convergence of $\boldsymbol{\alpha}$ in one $x-t$ plane. Initial estimation and following updates in different iterations are shown. Noise is gradually removed.

our objective function (non-convex). Our optimization converges after 3-5 iterations.

Energy non-increasing can be theoretically guaranteed in our framework, since all sub-problems have optimal solutions. Similar to all other L_0 norm problems (NP-hard), measuring the distance between a solution to the optimal one is still an ongoing problem for the theoretic community [46]. Our framework can provide a decent solution for stationary crowd analysis in practice. The relaxation steps and smoothing results of one example are shown in Fig. 8.

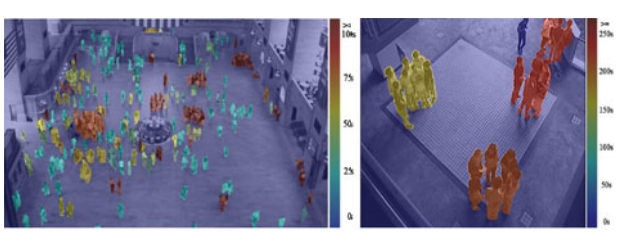
5 Experiments

Extensive experiments on real and synthetic data are conducted to evaluate the proposed stationary-time estimation method and its major components.

5.1 Datasets and Experiment Setup

Two datasets are used for evaluation, one is the Train Station dataset [47] and the other is collected by us. For each foreground pixel, its stationary-time up to the current frame is manually annotated. 17 frames (with over 8 million pixels) uniformly sampled from the two datasets are annotated at pixel level. They cover 70 percent frames in dataset I and 100 percent frames in dataset II. If the estimated stationary time T of the current frame is correct, all the previous T frames should be correct. Examples of annotated stationary-time maps are shown in Fig. 9. Details of the datasets are recorded in Table 1.

A one-minute video is spatially fragmented into 6×4 small video clips of size 160×135 and its frame rate is downsampled to 2.4 fps. For such a short clip, it takes around 30 seconds to optimize with an Intel CPU @3.3 GHz



(a) (b)

Fig. 9. Annotated stationary-time maps on (a) the Grand Central dataset [47] and (b) the dataset collected by us.

TABLE 1
Details of Datasets

	Dataset I [47]	Dataset II
Scene type	Indoor	Outdoor
Video length	3,500 seconds	800 seconds
Frame rate	24 fps	24 fps
Resolution	960×540	768×576
Number of annotated frames	8	9
Number of stationary pixels on the annotated frames	147,930	553,505
Total number of pixels on the annotated frames	4,147,200	3,981,312

in MATLAB. The optimization process for a one-minute video takes 12 minutes.

We empirically set η as 1.5, λ as 20, and the increasing ratio of β as 2 for both datasets. Their influences on the final encoding results are investigated in Section 5.5.

Several measures are used. The average estimation error on stationary-time (ET) for all foreground pixels is obtained. We compute the *ratio* between the estimation error and the ground truth for each foreground pixel. Then all the ratios on foreground pixels are averaged. This measure is denoted as average estimation error ratio on stationary-time (ERT). If a pixel has become stationary longer than 10 seconds up to the current frame, it is regarded as a stationary pixel. Several detection measures used include 1) false alarm rate (FAR), 2) missed detection rate (MDR), and 3) total error rate (TER).

Various baselines are evaluated and results are reported in Tables 2 and 3. We tested replacing the proposed second-order gradients (4) with first-order ones (5) (denoted as "Ours (FOrder)"), and replacing the proposed L_0 norm prior with L_1 norm one (denoted as "Ours (L_1)"). We also evaluate the results of excluding the two important components of the proposed method. First, foreground encoding procedure is omitted (all the foreground pixels would be assigned with the same codeword) and all other parts remain the same. This comparison is denoted as "Ours (NoCode)". Second, the codeword sharing step is omitted and the stationary-time cannot transfer among pixels. It is denoted as "Ours (NoShare)". Sharing codeword is explained the first paragraph of Section 3.4. Moreover, we also evaluate the performance of conducting 3D Markov

TABLE 2
Results of Stationary-Time Estimation on Dataset I

Methods	FAR	MDR	TER	ET(s)	ERT
Ours	0.29%	**3.49%**	**0.39%**	**10.04**	**12.21%**
Ours (FOrder)	0.51%	5.90%	0.69%	16.12	26.77%
Ours (L_1)	0.28%	4.91%	0.43%	14.29	19.53%
Ours (NoCode)	0.33%	3.50%	0.43%	16.94	21.02%
Ours (NoShare)	0.27%	13.74%	0.69%	19.24	24.33%
Encoding + MRF [48]	0.30%	8.91%	0.57%	15.16	19.81%
GMM [15]	0.27%	24.51%	1.11%	29.46	43.98%
Codebook [16]	**0.26%**	21.03%	0.93%	29.51	40.14%
Bayesian [17]	0.33%	20.18%	1.01%	26.70	39.16%
Keypoint tracking [19]	0.30%	24.26%	1.09%	40.78	56.49%
Person tracking [20]	0.29%	30.52%	1.23%	52.32	59.91%

ET is measured in seconds.

TABLE 3
Results of Stationary-Time Estimation on Dataset II

Methods	FAR	MDR	TER	ET(s)	ERT
Ours	0.91%	**0.54**%	**0.86**%	**15.88**	**8.67**%
Ours (FOrder)	1.37%	0.98%	1.32%	16.90	10.68%
Ours (L_1)	1.01%	0.76%	0.98%	17.04	12.44%
Ours (NoCode)	1.04%	0.55%	0.97%	27.76	15.30%
Ours (NoShare)	**0.89**%	4.15%	1.35%	32.46	18.11%
GMM [15]	0.92%	16.24%	3.06%	57.41	39.76%
Encoding + MRF [48]	0.90%	0.91%	1.89%	19.52	11.44%
Codebook [16]	1.03%	13.37%	2.75%	58.28	40.67%
Bayesian [17]	1.05%	12.26%	2.60%	45.20	32.19%
Keypoint tracking [19]	0.92%	5.75%	1.60%	54.14	38.86%
Person tracking [20]	1.01%	7.90%	1.89%	58.62	44.61%

ET is measured in seconds.

Random Field [48] smoothing after the proposed foreground encoding process (denoted as "Encoding+MRF").

We also compare our results with several background subtraction methods including the improved adaptive Gaussian mixture model [15], the codebook based model [16], and the adaptive Bayesian model [17]. Stationary-time is accumulated if a pixel is detected as foreground. Two tracking algorithms, including dense tracking [19] on detected foreground pixels [16] and multi-person tracking method [20] are also tested. Stationary-time is estimated as the length of the trajectory since a pixel becomes foreground.

5.2 Result Analysis

Our approach outperforms all the alternatives on both the indoor and outdoor datasets. Any component change or removal results in larger ET and ERT. The first order gradient prior is not powerful enough to constrain the stationary structure, thus "Ours (FOrder)" obtains a worse result than the proposed second order gradient prior "Ours". The L_1 norm term is less effective at constraining the sparse structure of encoding result $\boldsymbol{\alpha}_p$ than the proposed L_0 norm term. More experiments and discussions on the sparse gradient prior are in Section 5.3. Without the encoding process, different persons cannot be distinguished and there would be only one foreground codeword. Locally sharing of foreground codeword results in over-estimation, so the false positive rate of "Ours (NoCode)" is slightly higher than the proposed method. If foreground codewords cannot be shared, a lot of stationary-time information is lost and stationary-times restart from zero frequently, which leads to the large mis-detection rate of "Ours (NoShare)". The result of 3D MRF demonstrates that our alternative optimization scheme generates more accurate results than applying MRF after the first subproblem. The parameters in two subproblems (encoding and smoothing) affect each other, thus should be optimized alternatively.

With large mis-detection rates and large errors of estimated time, background subtraction and tracking based methods are not suitable for stationary-time estimation. The false positive rate of the proposed method is slightly higher than a few comparisons because of the smoothing effect yielded by the sparsity prior. However, the mis-detection rate of our method is much lower. The stationary-time estimation error (ET) is also at least 2.5 times lower than [15], [16], [17], [19], [20]. If some shadow cannot be perfectly removed by the initial background subtraction, false positives may arise.

In general, the adaptive Bayesian model [17] works better than other approaches, because it adds smoothness constraints in the spatial domain and between two successive frames. However, it is still not similarly good as ours because of the reasons discussed in Sections 1 and 2. This smoothness prior causes more false positives than ours, which manifest the necessity to employ the second-order gradient sparse prior. Both tracking algorithms cannot achieve satisfactory results because of their unreliable performance for crowded scenes. In addition, for the tracking-by-detection method [20], it only provides bounding-box results which roughly annotate pedestrians' locations and sizes, while our problem requires pixel-level stationary time maps. For the keypoint tracking method [19], it tends to generate fragmented tracklets for the same keypoint and might frequently reset stationary time back to zero.

5.3 Evaluation of the Proposed Sparse Gradient Prior

In this section, experimental evaluation is conducted to prove the effectiveness of the proposed sparse constraint, i.e., L_0 norm of second order gradient. Its advantages have been discussed in Section 3.2.

Different levels $(0 - 0.5)$ of noise are added to a synthetic video clip to test the robustness of the proposed sparse constraint. Noise is added by randomly switching between background and foreground pixels. The switching probability is denoted as noise level. Zero noise level means no noise. 0.5 noise level removes all the information of the original video. The synthetic video clip is manually designed. It contains multiple stationary foreground pedestrians of different sizes, which are simplified as cylinders. For synthetic data, it is easy to quantitatively add noise and calculate error rates for evaluating the regularization power of different methods. In Fig. 10, only one $x - t$ plane is shown and these pedestrians appears as rectangles in the images.

Fig. 10 shows the spatio-temporal planes of the input noisy videos and the reconstruction results of different methods. The first row shows six $x - t$ planes of the input video clips with different noise levels from 0 to 0.5. As the noise level increases, the pedestrian rectangles are blurred and no information remains when it reaches 0.5.

The average filtering, L_0 norm of first order gradient prior, and L_1 norm of second order gradient prior are used for comparison to show the effectiveness of the proposed L_0 norm of second order gradient prior. Reconstruction results of the three comparisons at different noise levels are shown in the following rows in Fig. 10. Results of the proposed prior are shown in the last row. The error rate of each case is measured as the percentage of erroneous pixels in total pixels. Error rate curves of different methods are shown in Fig. 12a.

The proposed method outperforms all the other comparisons from the results shown in Figs. 10 and 12a. When no noise is added, both the L_1 norm second order gradient prior and the L_0 norm first order gradient prior can achieve the same zero error rate as the proposed L_0 norm second order gradient prior. However, the average filtering results in a non-zero error rate because the rectangle corners are

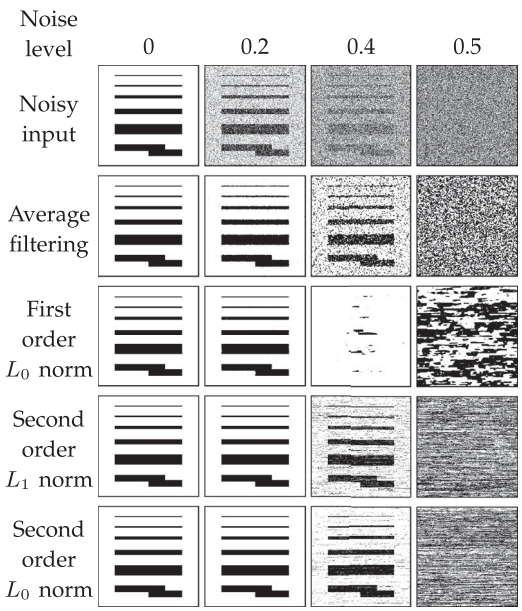


Fig. 10. Reconstruction results when adding noise to the whole synthetic video. Only one $x-t$ plane of the input/output video is shown. Input frames with different noise levels are in the first row. Corresponding reconstruction results using average filtering, L_0 norm constraint of first order gradient, L_1 norm constraint of second order gradient, and L_0 norm constraint of second order gradient, are in the following four rows. Noise level ranges from 0 to 0.5, shown in different columns. Black rectangles simulate foreground pedestrians of different sizes.

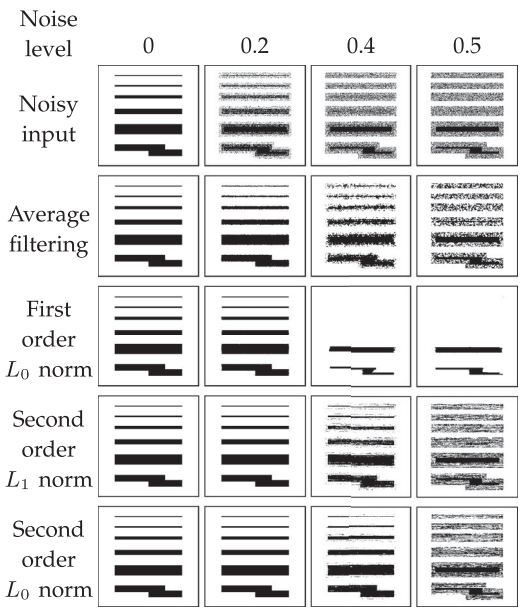


Fig. 11. Reconstruction results when adding noise around foreground boundaries of the synthetic video. Other settings are the same as those in Fig. 10.

smoothed. When noise level increases, results of the average filtering show lots of error patches in both foreground and background regions.

The second order gradient priors can maintain the rectangle shape while a lot of information is lost when using the first order gradient prior. Error rate curves of first and second order gradient priors start to increase at the noise level of 0.15 and 0.30, respectively. Even when the noise level reaches 0.5, reconstruction result of second order gradient priors are mostly lines parallel to the temporal dimension, which is close to the pattern of stationary pixels. However, first order gradient prior results in meaningless noisy patches. Our investigation shows that the target shape structure is important for the selection of sparse priors. Although the first order gradient prior achieves good results in many other image processing tasks, it is much less effective for our problem. The stronger regularization power of our proposed L_0 norm second order gradient prior can also be demonstrated through the comparison with L_1 norm second order gradient prior.

The most challenging regions for stationary-time estimation are foreground boundaries, because noise caused by occlusion, interaction, and local movements mostly happens in these regions. Another experiment is conducted and noise is only added on foreground boundaries. Noisy inputs are shown in the first row of Fig. 11. Other settings are the same as the previous experiment. Results are shown in Figs. 11 and 12b. L_0 norm second order gradient prior also performs best.

5.4 Analysis of the Background Subtraction Errors

In our algorithm, rough background subtraction result u_p is used to guide the foreground encoding process. Our optimization pipeline penalizes the differences between the encoded foreground pixels and the rough guidance u_p. The sparse constraint helps correct inaccurate background subtraction result to a certain degree.

In order to test the robustness of the proposed pipeline against errors of background subtraction results, one experiment is designed on a synthetic video by adding random noise on the ground truth background subtraction result. Different noise levels $(0-0.5)$ and noise sizes (1×1 and 3×3) are tested. The noisy background subtraction results and the final encoding results are shown in Fig. 13. The curves of encoding error rates with varying noise percentages are shown in Fig. 14.

From the results, we can observe that our method is able to correct a certain degree of errors of background subtraction through the joint optimization with the sparse gradient prior. From Fig. 13, we can see that our proposed optimization framework can generate satisfactory results even with quite poor background subtraction results (noise level $=0.35$ for noise size 1×1, and noise level $=0.30$ for noise size 3×3), which demonstrates that our method is very robust to the errors in background subtraction result.

5.5 Analysis of Parameter Settings

Our optimization framework has three main parameters, η, λ, and the increase step of β (denoted as β ratio). As shown in Fig. 15 and Table 4, six different parameter settings with

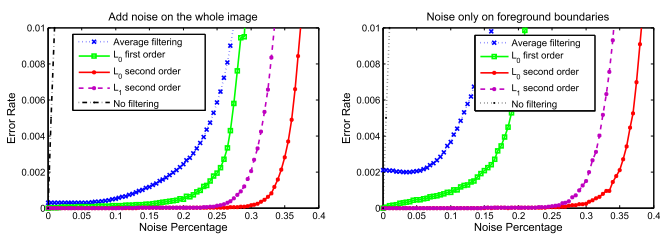


Fig. 12. Reconstruction error rate curves of different methods when adding noise (a) to the whole synthetic video clip, and (b) around foreground boundaries of the synthetic video.

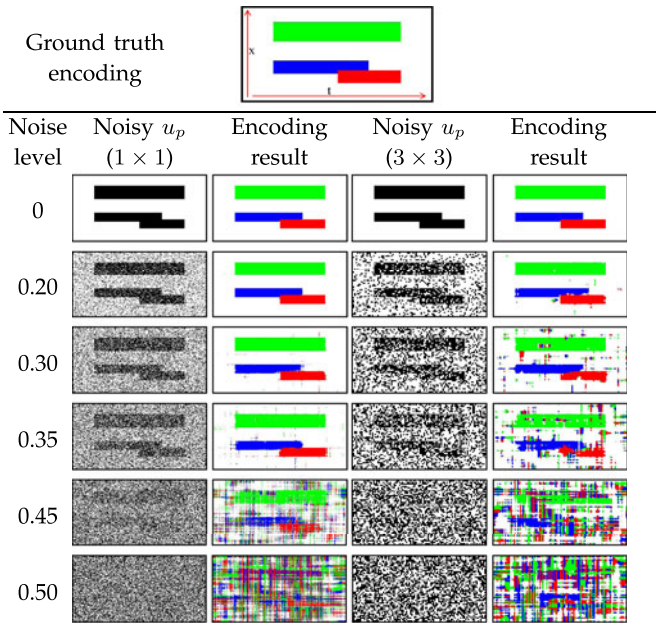


Fig. 13. Encoding results when adding noise to the background subtraction results u_p. Ground truth encoding is shown in the first row. Different colors represent different foreground objects. The background subtraction results u_p with different noise levels and the corresponding encoding results are shown in the following rows. The noise level ranges from 0 to 0.5. In the u_p map, black regions represent foreground pixels while white regions represent backgrounds. Encoding results of different codewords are shown in different colors.

respect to our default values are tested on a synthetic clip and also on datasets I and II, including (1) increasing β ratio to 20, (2) increasing β ratio to 200, (3) decreasing η to 0.15, (4) increasing η to 15, (5) decreasing λ to 2, and (6) increasing λ to 200. The results show that our method is robust to small changes of the parameters, and our current parameter setting achieves the best performance on both datasets I and II.

β ratio controls the convergence speed of our joint optimization scheme. Increasing β ratio to 20 still generate satisfactory results (Fig. 15(1)). However, when β ratio is too large, large encoding errors appear (Fig. 15(2)). Ideally, smaller β ratio leads to better performance, but more iterations are required for convergence. η balances the reconstruction of foreground with codewords and the deviation from the rough background subtraction result. With a smaller η, more background subtraction noise can be removed (Fig. 15(3)) With a larger η, the final encoding results mainly depend on the initial background subtraction result (Fig. 15(4)). λ balances the data term and the regularization term. If we decrease λ, the

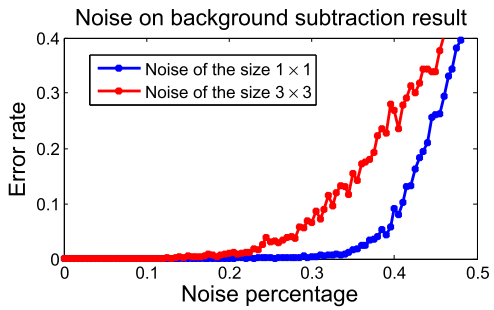


Fig. 14. Encoding error rate curves when adding noise to the synthetic background subtraction result.

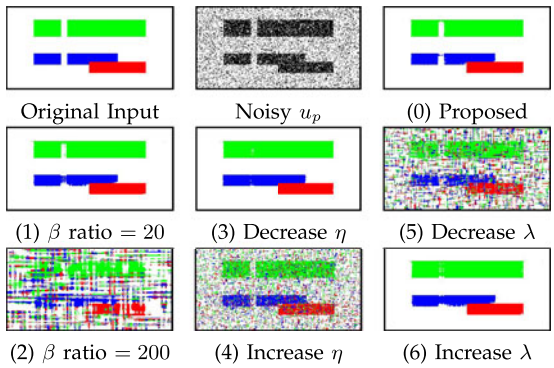


Fig. 15. Final foreground encoding results with different parameter settings.

data term influences more on the final results. The noise cannot be successfully removed without the sparse constraint term (Fig. 15(5)). Larger λ leads to smoother encoding result, since the sparse regularization is quite strong (Fig. 15(6)).

6 Applications

Several new applications are proposed in this section based on our stationary-time estimation algorithm.

6.1 Stationary Group Activity Detection

We apply our proposed method to detect stationary group activities and test it on the Ground Central Train Station dataset [47]. This dataset has numerous stationary group activities. We select four types of them as illustrated in Fig. 1 for the detection purpose, because these activities are of great interest in crowd surveillance and have enough samples in this dataset.

The stationary group activity detection task contains three main components, stationary group detection (Section 6.1.1), relevant trajectory selection (Section 6.1.2), and group activity description (Section 6.1.3).

6.1.1 Stationary Group Detection

The first step is to automatically detect all stationary groups in the entire video. Stationary-times of foreground pixels are estimated, and stationary foreground pixels are then selected by thresholding the estimated stationary-times. For each frame, stationary foreground pixels are clustered into groups with mean-shift [49]. Temporal overlaps of stationary foreground clusters are then used to match stationary

TABLE 4
Results of Stationary-Time Estimation by Modifying Different Parameters

	Parameters			Dataset I		Dataset II	
	λ	η	β ratio	TER	ET(s)	TER	ET(s)
(0) Proposed	20	1.5	2	0.39%	10.04	0.86%	15.88
(1) β ratio = 20	20	1.5	20	0.42%	12.79	0.87%	22.63
(2) β ratio = 200	20	1.5	200	0.50%	17.56	0.99%	29.76
(3) Decrease η	20	0.15	2	0.53%	15.28	0.92%	19.47
(4) Increase η	20	15	2	0.51%	16.17	1.04%	25.38
(5) Decrease λ	2	1.5	2	0.54%	13.83	0.96%	27.11
(6) Increase λ	200	1.5	2	0.49%	11.24	0.98%	23.42

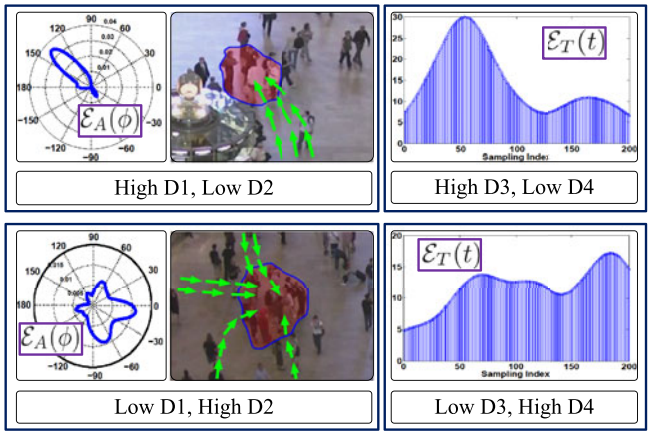


Fig. 16. Four examples of calculating $\mathcal{D}_1$ to $\mathcal{D}_4$ representing different formation types. The left two figures show two histograms of incoming trajectories over directions ($\mathcal{E}_A(\phi)$) which result in different values of $\mathcal{D}_1$ - $\mathcal{D}_2$ (people joining the group from the same direction vs different directions). The right two figures show two histograms of incoming trajectories over time ($\mathcal{E}_T(t)$) which result in different values of $\mathcal{D}_3$-$\mathcal{D}_4$ (people joining the group around the same time versus from different time).

group regions over time. In this way, stationary groups can be detected.

It is important to accurately estimate stationary periods. Let T_s and T_e be the time points when a stationary group emerges and disperses. They help to identify the emergence and dispersal periods of the group. Our motion descriptors are designed for these specific periods.

6.1.2 Relevant Trajectory Selection

The second step is to select relevant trajectories for each detected stationary group. Feature points are detected and tracked with the KLT tracker [50]. Tracking is not reliable in crowded environment. To avoid wrong data association, we adopt a conservative tracking strategy where trajectories with dramatic change of velocities are fragmented. Relevant trajectories are selected according to the spatial and temporal overlap with stationary groups. Trajectories relevant to the group are classified into three categories: incoming trajectories (**I**), outgoing trajectories (**O**), and trajectories inside a group (**P**).

6.1.3 Group Activity Descriptors

Pedestrians may join the group from the same direction within a short period, or from multiple directions over an extended period. All the group members may leave together towards the same direction or disperse in many directions at different time. The emerging and dispersal processes are used to characterize group activities such as gathering and stopping-by, while group topological states and change of group centers are used to detecting group deforming and relocating.

Twelve descriptors $\{\mathcal{D}_1, \ldots, \mathcal{D}_{12}\}$ are introduced to reflect the relationship and goals of group members. These descriptors are proposed based on the results of stationary group detection and selected relevant trajectories to distinguish different stationary group activities.

$\mathcal{D}_1$-$\mathcal{D}_4$ characterize the emergence process, i.e., whether members join a group from the same direction within a short period, or from multiple directions over an extended period. As shown in Fig. 16, $\mathcal{E}_A(\phi)$ and $\mathcal{E}_T(t)$ are computed as the histograms of incoming trajectories (**I**) over direction and time, where ϕ refers to direction angle and t refers to time. Both $\mathcal{E}_A(\phi)$ and $\mathcal{E}_T(t)$ are clustered with mean-shift and their dominant modes are denoted as $\mathcal{M}_A$ and $\mathcal{M}_T$. $\mathcal{D}_1$ to $\mathcal{D}_4$ are computed as:

$$\mathcal{D}_1 = \frac{\sum_{\phi \in \mathcal{M}_A} \mathcal{E}_A(\phi)}{\sum_{0 \le \phi < 2\pi} \mathcal{E}_A(\phi)}, \; D_2 = \sum_{\phi \notin \mathcal{M}_A} \frac{d(\phi - \hat{\phi})\mathcal{E}_A(\phi)}{2\pi \mathcal{E}_A(\hat{\phi})},$$

$$\mathcal{D}_3 = \frac{\sum_{t \in \mathcal{M}_T} \mathcal{E}_T(t)}{\sum_{T_s \le t \le T_e} \mathcal{E}_T(t)}, \; D_4 = \sum_{t \notin \mathcal{M}_T} \frac{|t - \hat{T}|\mathcal{E}_T(t)}{(T_e - T_s)\mathcal{E}_T(\hat{t})},$$

where $\hat{\phi} = \arg\max_\phi \mathcal{E}_A(\phi)$, $\hat{t} = \arg\max_t \mathcal{E}_T(t)$ represent the most probable incoming direction and arrival time, $d(\phi - \hat{\phi})$ is the angular distance, and 2π and $(T_e - T_s)$ are normalization terms. $\mathcal{D}_1$ and $\mathcal{D}_3$ characterize the aggregation degrees of the dominant modes over direction and time distributions, while $\mathcal{D}_2$ and $\mathcal{D}_4$ characterize the scatter degrees of other modes.

Similarly, $\mathcal{D}_5$-$\mathcal{D}_8$ characterize the dispersal process based on outgoing trajectories (**O**), i.e., whether members leave a group towards the same direction around the same time, or in many directions at different times. $\mathcal{D}_9$ is the spatial variance of a group center and can be used to detect group relocating.

$\mathcal{D}_{10}$-$\mathcal{D}_{12}$ characterize whether a stationary group keeps its internal structure stable or not. They are computed based on the topological variations of feature points inside the stationary group. In order to be robust to projective distortion and cross-scene variation, $\mathcal{D}_{10}$-$\mathcal{D}_{12}$ are based on topological distance instead of geometric distance and only feature points inside the stationary group are considered. These feature points are collected from trajectory set (**P**). If a feature point i stays inside a stable group, its k-nearest neighbor set $\mathcal{N}_t(i)$ and topology of neighbors tend to remain unchanged over time. $\mu_t(i)$ is introduced to measure the portion of changed neighbors of feature point i from $t - \Delta$ to t, $\mu_t(i) = 1 - |\mathcal{N}_t(i) \cap \mathcal{N}_{t-\Delta}(i)|/K$. The K' invariant neighbors from $t - \Delta$ to t are ranked according to their distances to point i. $\mathcal{R}_t(i)$ and $\mathcal{R}_{t-\Delta}(i)$ are defined as the rankings of K' invariant neighbors at time t and $t - \Delta$. $\mathcal{R}_t(i) = [\sigma_t^1(i), \ldots, \sigma_t^{K'}(i)]$, and $\mathcal{R}_{t-\Delta}(i) = [\sigma_{t-\Delta}^1(i), \ldots, \sigma_{t-\Delta}^{K'}(i)]$. $\varsigma_t(i)$ is calculated as the Kendall tau distance between the two rankings $\mathcal{R}_t(i)$ and $\mathcal{R}_{t-\Delta}(i)$. Similarly, $\kappa_t(i)$ is computed based on rankings of angles.

$\mathcal{D}_{10}$, $\mathcal{D}_{11}$, and $\mathcal{D}_{12}$ are computed as the average of all the feature points during the whole stationary period based on $\mu_t(i)$, $\varsigma_t(i)$, and $\kappa_t(i)$, respectively. Examples of these descriptors are shown in Fig. 17.

6.1.4 Experimental Evaluation

All stationary groups containing these activities are manually annotated as ground truth. We only consider groups whose stationary-time is longer than 30 seconds and sizes are larger than 2,500 pixels, since large groups with long stationary-time draw attention in surveillance.

For each activity, 30 groups are randomly selected as training samples. Linear SVM is trained for each activity separately. $\mathcal{D}_1$-$\mathcal{D}_8$ are used for the detectors of group gathering and stopping-by. $\mathcal{D}_9$ is used for the detector of group-relocating. $\mathcal{D}_{10}$-$\mathcal{D}_{12}$ are used for the detector of

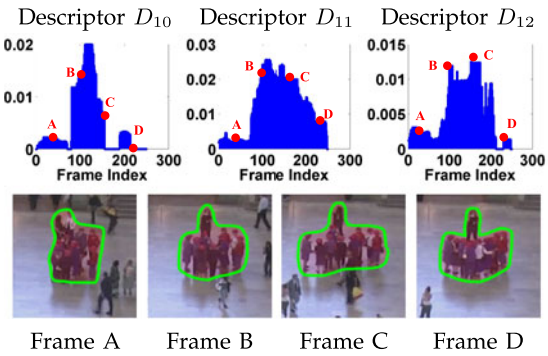


Fig. 17. Examples of descriptors $\mathcal{D}_{10}$ to $\mathcal{D}_{12}$ which characterize the stability of the internal structures of stationary groups. The dynamic variations of their values are shown. The topological structure of the group has large variations from frame B to C, when its members start to line up to take photos. The structure is stable at A and D, when the group members have discussion together at A and when the members are already lined up at D.

group-deforming. The trained detectors search through the entire video. A true positive is counted if the overlap between a detected group and the ground truth is larger than 50 percent in the spatio-temporal space.

Table 5 reports the numbers of false positives and missed detections of different approaches. All these approaches use the same tracking result and group descriptors, but different ways of estimating stationary-time. It is shown that estimating stationary-time has large influence on the activity detection results. To study the effectiveness of the proposed method across datasets, the detectors trained on dataset I are tested on both datasets, and the results are reported in Table 5. Although there is no group gathering activity in dataset II, we also report the number of false positives.

6.2 Scene Understanding

Stationary-time estimation can help scene understanding and provide valuable statistics over time. For example, an averaged stationary-time map computed over all the groups in the four-hour Grand Central Train Station video is shown in Fig. 3. It indicates where stationary groups tend to emerge, and how long they generally stay. Such information is important for crowd management, public facility design, event monitoring, and traffic control. A simple scenario is that if stationary groups appear at an entrance to a building, alarm can be triggered for taking further action to improve traffic there.

TABLE 5
Activity Detection Results (False Positive / Mis-Detection)

Activities	Gather	Stop by	Relocate	Deform
Training samples (dataset I)	30	30	30	30
Test samples (dataset I)	45	58	27	50
Ours	3/6	5/6	4/1	6 / 4
GMM [15]	4/23	6/25	4/9	7/19
Codebook [16]	3/22	4/23	4/8	7/18
Bayesian [17]	2/23	4/24	3/8	6/17
Tracking [19]	4/25	5/28	5/12	6/20
Test samples (dataset II)	0	9	2	4
Ours	1/0	0/2	1/0	1/2

Fig. 18. Examples of the dynamic changes of the stationary blocking regions at two different times.

Moreover, the average stationary-time maps for small temporal periods of ten minutes are shown in Fig. 18. The dynamic changes of the stationary blocking regions of the scene can be observed. Some travelers stay in front of the ticket window to buy tickets (left), and in front of the entrance to board trains (right). From the dynamic variations of stationary blocking regions, crowd behaviors can be observed and their relations to scene structures can be better understood.

6.3 Influence on Traffic Patterns

Stationary groups have great influence on traffic flow yet to be discovered. To analyze the influence, we first cluster pedestrian trajectories using the random field topic model [51], and the most probable path is generated by averaging clustered trajectories. Then the correlation between the dynamic variations of stationary groups and the most probable traffic paths can be discovered.

Some examples of influence of stationary crowd groups on traffic patterns are shown in Fig. 19. In (a1) and (b1), the influences of stationary groups on traffic flows are not significant as the blocking stationary groups are not large. In (a2) and (b2), the influences are significant due to the blocking of large stationary groups. In (a3) and (b3), as the stationary groups are sparse and walking pedestrians choose to go through the stationary regions to their destinations. Traffic flow changes a lot due to the dynamic changes of stationary groups.

7 Conclusion

We have explored stationary crowd group analysis, which has many important applications but was less studied in the literature. A fundamental step is to estimate the stationary-time of foreground pixels. We propose a robust algorithm that optimizes a locally shared foreground codebook and uses second-order gradient prior to constrain the 3D stationary-time map. It is formulated as an L_0 minimization problem and is solved by a practically effective scheme. The

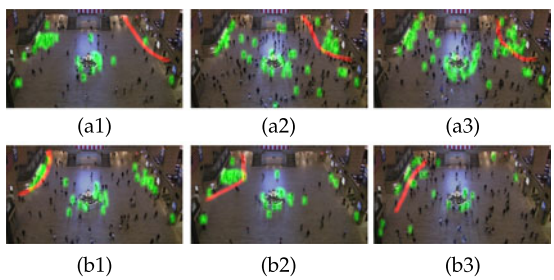


Fig. 19. Examples of the influences of stationary crowds on pedestrian traffic patterns. Stationary groups are detected by the proposed method and are marked in green. The average pedestrian walking paths are marked in red.

effectiveness of the proposed method is demonstrated through several applications such as detecting stationary group activities, crowd scene understanding, and studying the influence of stationary groups on traffic patterns.

ACKNOWLEDGMENTS

This work is partially supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (Nos. CUHK14206114, CUHK14205615, CUHK419412, CUHK14203015, CUHK413113), the Hong Kong Innovation and Technology Support Programme (No. ITS/221/13FP), National Natural Science Foundation of China (Nos. 61371192, 61133009, 61301269), PhD programs foundation of China (No. 20130185120039), and Sichuan Hi-tech R&D Program (No. 2014GZX0009).

REFERENCES

[1] S. Ali and M. Shah, "A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–6.

[2] A. B. Chan, Z.-S. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–7.

[3] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[4] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert, "Data-driven crowd analysis in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1235–1242.

[5] B. Zhou, X. Tang, H. Zhang, and X. Wang, "Measuring crowd collectiveness," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1586–1599, Aug. 2014.

[6] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 935–942.

[7] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 261–268.

[8] P. Scovanner and M. F. Tappen, "Learning pedestrian dynamics from the real world," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 381–388.

[9] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1345–1352.

[10] G. Le Bon, *The Crowd: A Study Popular Mind*. New York, NY, USA: Macmillan, 1897.

[11] D. Forsyth, *Group Dynamics*. Boston, MA, USA: Cengage Learning, 2009.

[12] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS One*, vol. 5, no. 4, p. e10047, 2010.

[13] S. Yi, H. Li, and X. Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3488–3496.

[14] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1999, pp. 246–252.

[15] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recog.*, 2004, pp. 28–31.

[16] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, 2005.

[17] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.

[19] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3169–3176.

[20] J. Zhang, L. L. Presti, and S. Sclaroff, "Online multi-person tracking by tracker hierarchy," in *Proc. IEEE 9th Int. Conf. Adv. Video Signal-Based Surveillance*, 2012, pp. 379–385.

[21] S. Denman, V. Chandran, and S. Sridharan,, "An adaptive optical flow technique for person tracking systems," *Pattern Recog. Lett.*, vol. 28, no. 10, pp. 1232–1239, 2007.

[22] L. Sevilla-Lara, D. Sun, E. G. Learned-Miller, and M. J. Black, "Optical flow estimation with channel constancy," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 423–438.

[23] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *Int. J. Comput. Vis.*, vol. 73, no. 1, pp. 41–59, 2007.

[24] D. Vazquez, A. M. Lopez, J. Marin, D. Ponsa, and D. Geroimo, "Virtual and real world adaptation for pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 797–809, Apr. 2014.

[25] D. Lin, E. Grimson, and J. Fisher, "Learning visual flows: A Lie algebraic approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 747–754.

[26] D. Kuettel, M. D. Breitenstein, L. Van Gool, and V. Ferrari, "What's going on? discovering spatio-temporal dependencies in dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010.

[27] T. Hospedales, S. Gong, and T. Xiang, "A Markov clustering topic model for mining behaviour in video," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 1165–1172.

[28] T. M. Hospedales, J. Li, S. Gong, and T. Xiang, "Identifying rare and subtle behaviors: A weakly supervised joint topic model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2451–2464, Dec. 2011.

[29] R. Emonet, J. Varadarajan, and J.-M. Odobez, "Extracting and locating temporal motifs in video scenes using a hierarchical non parametric Bayesian model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3233–3240.

[30] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proc. Nat. Acad. Sci.*, vol. 99, no. Suppl 3, pp. 7280–7287, 2002.

[31] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[32] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[33] W. Ge, R. T. Collins, and R. B. Ruback, "Vision-based analysis of small groups in pedestrian crowds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1003–1016, May 2012.

[34] T. Lan, Y. Wang, W. Yang, and G. Mori, "Beyond actions: Discriminative models for contextual group activities," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1216–1224.

[35] M.-C. Chang, N. Krahnstoever, and W. Ge, "Probabilistic group-level motion analysis and scenario recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 747–754.

[36] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, "Discriminative latent models for recognizing contextual group activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1549–1562, Aug. 2012.

[37] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1354–1361.

[38] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, and V. Murino, "Social interaction discovery by statistical analysis of f-formations," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–23.

[39] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2001, pp. 431–438.

[40] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 1975–1981.

[41] M. R. Amer and S. Todorovic, "A chains model for localizing participants of group activities in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 786–793.

[42] B. Solmaz, B. E. Moore, and M. Shah, "Identifying behaviors in crowd scenes using stability analysis for dynamical systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2064–2070, Oct. 2012.

[43] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Attribute learning for understanding unstructured social activity," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 530–543.
[44] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, 2011, Art. no. 174.
[45] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
[46] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.
[47] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 2871–2878.
[48] R. Kindermann, J. L. Snell, et al., *Markov Random Fields Their Application*. Providence, RI, USA: American Mathematical Society Providence, 1980, vol. 1.
[49] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[50] B. D. Lucas, T. Kanade, et al.,"An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, vol. 81, pp. 674–679.
[51] B. Zhou, X. Wang, and X. Tang, "Random field topic model for semantic region analysis in crowded scenes from tracklets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3441–3448.

Shuai Yi received the BEng degree in electronic engineering from Tsinghua University in 2012. He is currently working toward the PhD degree at the Department of Electronic Engineering, Chinese University of Hong Kong. His research interests include computer vision and machine learning, specifically for crowd analysis and video surveillance.

Xiaogang Wang received the BS degree from the University of Science and Technology of China in 2001, the MS degree from the Chinese University of Hong Kong in 2003, and the PhD degree from the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology in 2009. He is currently an associate professor in the Department of Electronic Engineering, The Chinese University of Hong Kong. His research interests include computer vision and machine learning. He is a member of the IEEE.

Cewu Lu received the BS and MS degrees from the Chongqing University of Posts and Telecommunications and Graduate University of Chinese Academy of Sciences in 2006 and 2009, respectively, and the PhD degree in computer science and engineering from the Chinese University of Hong Kong in 2013. He is currently a research fellow at the Hong Kong University of Science and Technology and a visiting scholar in Stanford university. His research interests include activity recognition, object detection, and image/video processing. He is a member of the IEEE.

Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2004 and is currently an associate professor in the Department of Computer Science and Engineering at the Chinese University of Hong Kong (CUHK). He was a visiting scholar at Microsoft Research Asia from March 2004 to August 2005 and conducted collaborative research at Adobe Systems in 2007. He heads the research group in CUHK, focusing specifically on computational photography, 3D reconstruction, practical optimization, and motion estimation. He is a senior member of the IEEE.

Hongsheng Li received the bachelor's degree in automation from the East China University of Science and Technology, and the master's and doctorate degrees in computer science from Lehigh University, Pennsylvania, in 2006, 2010, and 2012, respectively. He is currently with the Department of Electronic Engineering at the Chinese University of Hong Kong. His research interests include computer vision, medical image analysis, and machine learning.