L_0 Regularized Stationary Time Estimation for Crowd Group Analysis

Shuai Yi Xiaogang Wang Cewu Lu Jiaya Jia
The Chinese University of Hong Kong
{syi,xgwang}@ee.cuhk.edu.hk, {cwlu,leojia}@cse.cuhk.edu.hk

Abstract

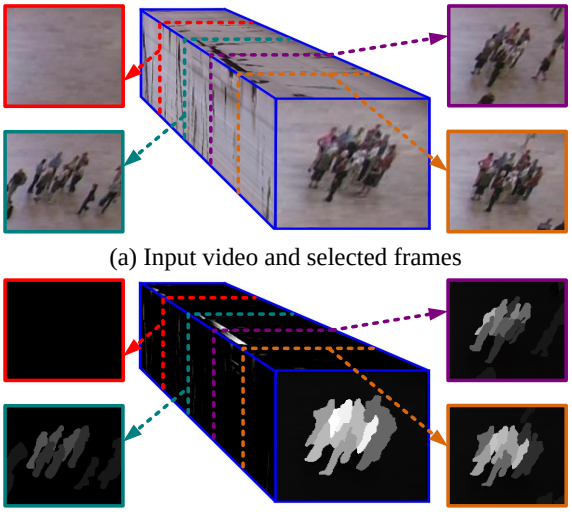
We tackle stationary crowd analysis in this paper, which is similarly important as modeling mobile groups in crowd scenes and finds many applications in surveillance. Our key contribution is to propose a robust algorithm of estimating how long a foreground pixel becomes stationary. It is much more challenging 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 accomplish decent results, sparse constraints along spatial and temporal dimensions are jointly added by mixed partials to shape a 3D stationary time map. It is formulated as a 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 used to detect four types of stationary group activities and analyze crowd scene structures. We provide the first public benchmark dataset[1] for stationary time estimation and stationary group analysis.

1. Introduction

Crowd analysis finds many important applications in video surveillance [25, 1, 4, 26, 15, 18, 32, 30, 6, 31, 20]. Crowd management and traffic control are common problems in public areas when the population density is high. Existing work focuses on detecting motion patterns of crowds [25, 1, 26, 18, 32, 13, 30, 31, 20] and analyzing interactions among pedestrians during movement [15, 17, 19, 28, 20]. On the other hand, stationary crowd group analysis has never been sufficiently studied although these groups can provide surprisingly rich information.

For example, study of [16] shows that stationary groups have a greater impact on changing traffic patterns than mobile groups in some cases. When pedestrians move around, they adjust speed but not direction to avoid collisions. Such self-organized behaviors keep traffic flow smooth. If pedestrians form stationary groups, they force others to change directions and much affect transportation efficiency. Emergence and dispersal of stationary groups cause dynamic variation of crowd traffic patterns. It is thus of great interest to incorporate stationary groups into dynamic modeling of crowd systems. It is also worth investigating where stationary groups are likely to emerge and how long they tend to stay. An average stationary time map is shown in Figure 3. It is informative for crowd management, as well as provision of facilities and support.



(a) Input video and selected frames

(b) Output stationary time map

Figure 1. Estimating a 3D stationary time map from a video sequence. Results from a few frames are shown. How long a pixel has been stationary up to each frame is encoded by the intensity level. Brighter pixels correspond to longer time.

Groups that stay for a period of time are often worth attention. Emergence, dispersal, stationary duration, and status of them may incur great security interest. From detected activities, we may discover valuable information, such as relation of people and possible abnormality. Figure 2 shows four types of stationary group activities that are to be detected in this paper.

Our method estimates *stationary time*, i.e., period that

[1] http://www.ee.cuhk.edu.hk/∼xgwang/CUHKStationaryCrowd.html

(a) group gathering (c) group relocating

(b) group stopping by (d) group deforming

Figure 2. Four major types of stationary group activities to be detected in our work, typical in the CVPR conference scene during the break between two oral sessions. (a) People join a group from different directions at different time. When all people arrive, the whole group moves along the same destination. (b) A group of people enter 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. Please view images in color.

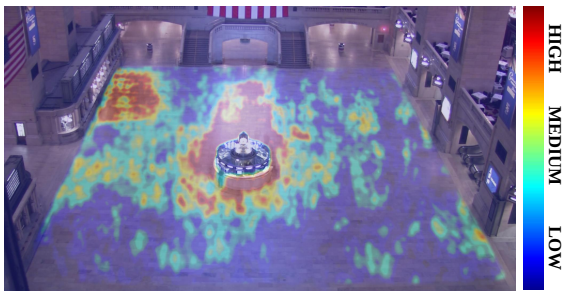


Figure 3. Averaged stationary time distribution over 4 hours. Stationary groups tend to emerge and stay long around the information booth and in front of the ticketing windows (better viewed in color).

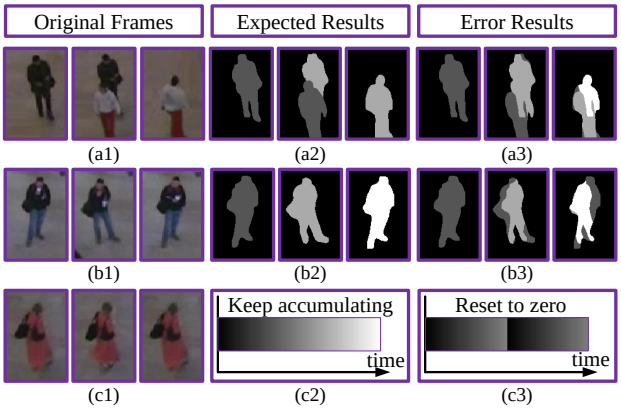


Figure 4. Challenges of stationary time estimation. Results from background subtraction are erroneous. (a) Two foreground objects with spatio-temporal overlap. (b) Local movement of objects also leads to estimation errors. (c) If a foreground pixel is misclassified as background in one frame, stationary time resets to 0, which is wrong. In (c3), misclassification happens in the middle, making time reset.

a foreground pixel exists in a local region allowing local movements. As shown in Figure 1, given a video sequence, our method produces a 3D *stationary time* map in the spatio-temporal space. This is different from only subtracting background. 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 it as a new research problem. It is an important step for further analysis on stationary crowds.

Figure 4 illustrates the inherent challenge. First, background subtraction does not distinguish among different foreground objects. If two objects overlap, the estimated stationary time could be longer than what it should be in the overlapping region, as shown in Figures 4(a) and 5. This happens frequently in crowded scenes.

Second, people's local movement is very common during the stationary period as shown in Figure 4(b). We should keep on accumulating stationary time even with these local movements, instead of frequently resetting time to 0. Matching locally moving foreground objects especially in crowded scenes is not easy. Third, most background subtraction methods do not consider temporal consistency. If a foreground pixel is misclassified at one frame, stationary time could be mistakenly reset to 0, as shown in Figure 4(c). Given all these challenges coupled, none of the existing approaches is ready to solve our problem.

Our contributions are as follows. (1) A robust stationary-time estimation algorithm is proposed. By using a locally shared foreground codebook, it separates foreground objects even if they are close or overlapping in the spatio-temporal space. It also allows for matching shifted foreground objects in local regions. (2) Sparse constraints in spatial and temporal dimensions are jointly added to construct a 3D stationary-time map. This is formulated as a L_0 optimization problem, much more powerful in regularization than commonly used local smoothness (e.g. MRF) added on image and temporal space separately. The op-

timization is performed on a batch of frames instead of individual ones. This process is robust to occasional local movement of stationary objects, occlusions, and misclassification. (3) A set of new descriptors are proposed to describe stationary group activities illustrated in Figure 2 and estimated stationary time is used to understand crowd scene structures as shown in Figure 3. (4) A dataset with annotated ground truth is provided to the public for stationary-time estimation and stationary group activity detection, which is the first in its kind.

2. Related Work

Background subtraction is well studied. The adaptive Gaussian mixture model proposed by Stauffer et al. [23] is one popular approach and its improved version was proposed by Zivkovic et al. [33]. It uses Gaussian mixture to adapt background change. Kim *et al.* [11] modeled complex background variations with a codebook. Challenges faced by these approaches have been discussed in Section 1. Robust PCA [3] separates foreground 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.

The background subtraction work more relevant to ours is the Bayesian model proposed in [21]. It employed joint features of color and location and performed nonparametric density estimation to handle local movement on background (e.g. waves). To enforce temporal persistence, the likelihood of a pixel being foreground increases by a constant if it is detected as foreground in the previous frame. Spatial smoothness was enforced within a single frame with MRF.

This method is quite different from ours. Its constraints on spatial and temporal consistency are separate, while ours jointly models them with sparsity on second-order gradients in the 3D space. Also, the Bayesian model performs optimization within a frame or between two frames, while ours jointly optimizes a batch of video frames. L_0 sparsity is a very powerful regularization form usable in many applications. Finally, it does not distinguish among foreground objects. Dense tracking [24] on foreground pixels for stationary time estimation could be problematic due to frequent occlusion in crowd and the difficulty of finding good feature points. Pedestrian detection and tracking do not work well too in crowd scenes due to heavy occlusion. Our experiments show that these approaches cannot produce similarly good results as ours.

Detecting social groups and analyzing their activities is another stream. Cristani et al. [7] studied the interaction of standing people in a sociological view. Other work along this line mainly considers moving groups. Pedestrians are grouped based on their relative distances and similarities of moving patterns [10, 9, 5]. Various features and models were proposed to recognize different mobile group behaviors [29, 14, 2, 22, 8, 12]. As discussed in Figure 2, stationary groups have their own properties and need special features to characterize them.

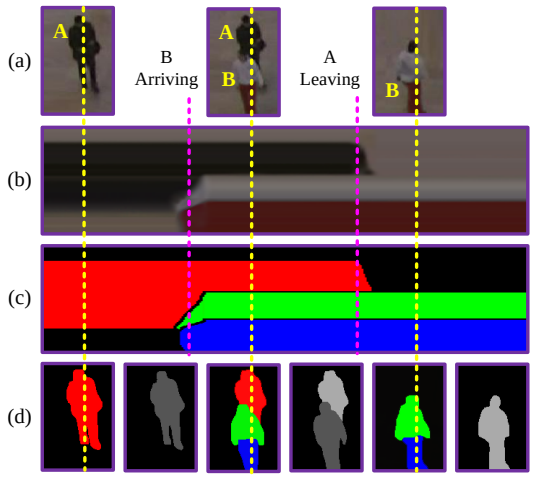


Figure 5. Illustration of using locally shared foreground codebook to separate foreground objects close or overlapped in the spatio-temporal space. (a) 3 frames from the same region. When person B arrives, A leaves the group. (b) Temporal slice image along the yellow line temporally. A and B overlap. (c) Foreground pixels assigned with three different codes. They are well separated. (d) Foreground codes (left) and estimated stationary time (right).

3. Stationary Time Estimation Model

Pixel-level stationary time is estimated from a color video sequence, which is uniformly divided into short clips with overlap, such that information of foreground codes and stationary time can be consistent across clips.

3.1. Guided Foreground Mode Encoding

In contrast to background subtraction that only indicates whether a pixel is foreground or not, we label pixels with multiple foreground *modes*, making pixels belonging to different people can be also differentiated.

Foreground pixels are clustered into M modes. Each pixel p is with a 5D feature vector, i.e., $I_p = [R_p, G_p, B_p, X_p, Y_p]^T$, where $[R_p, G_p, B_p]$ and $[X_p, Y_p]$ are RGB values and spatial coordinates of p. Spatial location makes it possible to share one mode in a local region, robust to small movement. Pedestrians with different RGB values are clustered into modes. Cluster centers are denoted as $\{\mathbf{d}_1, \ldots, \mathbf{d}_M\}$ ($\mathbf{d}_i \in \mathbb{R}^{5\times 1}, \forall i = 1, \ldots, M$), forming a $5 \times M$ matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_M]$. $\mathbf{D}$ is initialized by mean shift, and M varies for different video clips.

Our encoding process starts from the background subtraction result [11], where $u_p = 0$ indicates p is on the background and $u_p = 1$ denotes foreground. This result is noisy. Misclassification at a single frame could lead to large errors on stationary time estimation. Our goal is to

find a M dimension coding vector $\boldsymbol{\alpha}_p$ ($\boldsymbol{\alpha}_p = \{0,1\}^M$) for each pixel p. The entries of $\boldsymbol{\alpha}_p$ can only be 1 or 0, and at most one element can be 1. If all of the entries are 0, p is on the background and $\|\boldsymbol{\alpha}_p\|_1 = 0$. If the mth element is 1, p belongs to foreground mode $\mathbf{d_m}$ and $\|\boldsymbol{\alpha}_p\|_1 = 1$.

We use $\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha})$ to balance encoding errors on foreground pixels and deviation from the initial background subtraction result using two terms. It is expressed as

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

η is a parameter. $\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha})$ is minimized together with other sparse terms, to be detailed shortly. Both $\mathbf{D}$ and $\boldsymbol{\alpha}$ will be estimated.

3.2. Sparse Gradient Prior

The stationary time of a pixel p increases if it stays with the same foreground label $\boldsymbol{\alpha}_p$. Due to lighting variation, local movement, and occlusion, estimation of $\boldsymbol{\alpha}$ could be noisy. Our finding is that change of $\boldsymbol{\alpha}_p$ on ideal stationary objects without aforementioned problems should be very sparse. We accordingly impose a sparse prior $c(\boldsymbol{\alpha})$ during estimation of label $\boldsymbol{\alpha}$ to resist noise. It is written as

$$c(\boldsymbol{\alpha}) = \#\{p| \, ||\partial_{x,t}\boldsymbol{\alpha}_p||_2 + ||\partial_{y,t}\boldsymbol{\alpha}_p||_2 \neq 0\}, \tag{2}$$

where $\partial_{x,t}$ and $\partial_{y,t}$ are the second-order gradients with regard to $x-t$ and $y-t$ space derivatives. $\#$ counts the number of changes in the mixed partials.

This is exactly an L_0 gradient minimization problem. Xu *et al.* [27] discussed a similar problem and showed that L_0 norm, unlike L_1 and L_2 sparsity, has many excellent properties in solving a large variety of image problems. They achieved decent results by globally controlling the number of non-zero gradients in order to constrain the produced sparse structure. We employ a solver similar to the one publicly available in the project website [27]. We make modifications on processing spatio-temporal information with second-order gradients, detailed below.

Why Second-Order Gradients? To enforce smoothness during segmentation and foreground region labeling, a simple prior incorporating first-order gradients along each dimension may be used, expressed as

$$c'(\boldsymbol{\alpha}) = \#\{p| \, ||\partial_x\boldsymbol{\alpha}_p||_2 + ||\partial_y\boldsymbol{\alpha}_p||_2 + ||\partial_t\boldsymbol{\alpha}_p||_2 \neq 0\}. \tag{3}$$

We compare this form with that in Eq. (2) to show second-order gradients are more effective for stationary-time estimation. In Eq. (3), any nonzero values in x, y, or t gradients would result in nonzero c'. When calculating c', a stationary person produces the result shown in Figure 6(c), where all body boundaries inevitably produce many nonzero values.

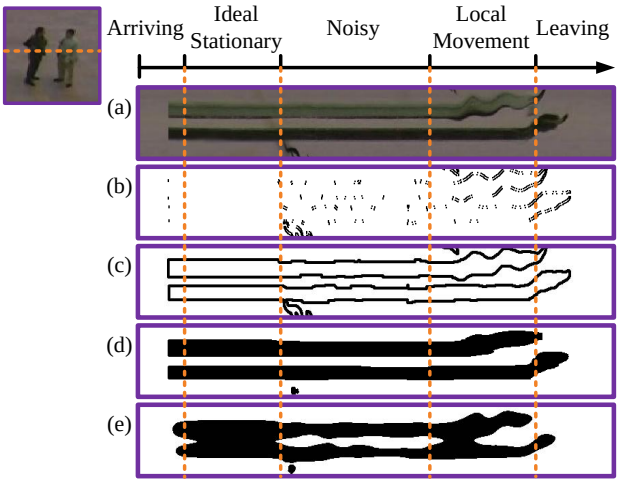


Figure 6. By adding the sparse prior, we estimate $\boldsymbol{\alpha}$ better from noisy and/or locally moving objects. (a) Stages of two pedestrians arriving, staying, locally moving, and leaving (x-axis: time; y-axis: scanline pixels highlighted in orange). (b) Non-zero $\partial_{x,t}$ values are shown in black. (c) Nonzero gradients ∂_x+∂_t shown in black. (d) Our labeling result with $\partial_{x,t}$. (e) Erroneous labeling result with ∂_x+∂_t.

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 Eq. (2), as spatial-boundary-caused nonzero gradients would be eliminated if the object is stationary when calculating $\partial_{x,t}$ for those pixels. As shown in Figure 6(b), only a few *moving boundary* pixels yield nonzero c. Thus only penalizing these pixels would result in very sparse labeling structures, robust to noise and outliers. We compare the final results of our system by using these two priors respectively in Figure 6(d) and (e). The second order gradient is effective to produce reasonable labels for different foreground regions.

3.3. Joint Optimization

Eqs. (1) and (2) are fed into unified optimization as

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \{\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha}) + \lambda c(\boldsymbol{\alpha})\}, \quad s.t. \;\; \boldsymbol{\alpha}_p = \{0,1\}^M, \|\boldsymbol{\alpha}_p\|_1 \leq 1. \tag{4}$$

The data term $\mathcal{Q}(\mathbf{D}, \boldsymbol{\alpha})$ generates M mid-level semantic modes from hundreds of intensity levels, which significantly simplifies locally matching foreground pixels. The prior $c(\boldsymbol{\alpha})$ captures the structural sparsity for each mode of a stationary object in the spatio-temporal space.

3.4. Pixel-Wise Stationary Time Estimation

It is easy to estimate stationary time based on the change of foreground modes (or labels). If foreground mode of a pixel is $\mathbf{d_i}$ starting from frame t_1, and it is changed to a different foreground mode $\mathbf{d_j}$ or background at frame t_2, the

stationary time of this pixel is $t_2 - t_1$. If a pixel is changed from background to a foregound mode, it locally searches for a pixel with the same mode in its previous frame. If such a matched pixel is found, its stationary time will be inherited by the current pixel, instead of counting from zero. This avoid underestimation caused by local movement.

If a frame is close to the boundary of a video clip, estimation may not be similarly reliable. Our system uses overlapping video clips with shared buffer frames. Only the estimated stationary time in frames outside the buffer is kept for reliability's sake. If objects stay longer than one clip duration, their foreground modes can be matched across clips so that the stationary time can continue in accumulation.

4. Solver

$\mathbf{D}$ and $\boldsymbol{\alpha}$ in Eq. (4) are coupled and the problem is highly non-convex. We introduce a set of axillary vectors $\boldsymbol{\alpha}_p^0 \in \mathbb{R}^M$ to relax the original problem, expressed as

$$\min_{\mathbf{D},\boldsymbol{\alpha},\boldsymbol{\alpha}^0} \left\{ \mathcal{Q}(\mathbf{D},\boldsymbol{\alpha}^0) + \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 + \lambda c\left(\boldsymbol{\alpha}\right) \right\},$$
$$s.t.\ \boldsymbol{\alpha}_p = \{0,1\}^M, \|\boldsymbol{\alpha}_p\|_1 \le 1, \boldsymbol{\alpha}_p^0 = \{0,1\}^M, \|\boldsymbol{\alpha}_p^0\|_1 \le 1. \quad (5)$$

When β_1 is large enough, $\boldsymbol{\alpha}_p^0$ perfectly approaches $\boldsymbol{\alpha}_p$. It makes the original challenging problem boil down to two sub-ones. Satisfactory result can be achieved by solving the two optimization problems iteratively and increasing β_1 after each iteration. This strategy was used in [27]. It is rather effective to solve L_0 gradient minimization problems.

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

With $\boldsymbol{\alpha}_p$ fixed, the sparse prior term is a constant and can therefore be omitted. We rewrite Eq. (5) as

$$\min_{\mathbf{D},\boldsymbol{\alpha}^0} \left\{ \mathcal{Q}(\mathbf{D},\boldsymbol{\alpha}^0) + \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 \right\},$$
$$s.t.\ \boldsymbol{\alpha}_p^0 = \{0,1\}^M, \|\boldsymbol{\alpha}_p^0\|_1 \le 1. \quad (6)$$

$\mathbf{D}$ and $\boldsymbol{\alpha}_p^0$ are estimated iteratively. Given $\boldsymbol{\alpha}_p^0$, $\mathbf{D}$ is obtained by solving a least square problem. Given $\mathbf{D}$, $\boldsymbol{\alpha}_p^0$ is achieved by naively searching $(M+1)$ possibilities of foreground modes and background.

4.2. Solve for $\boldsymbol{\alpha}_p$

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

$$\min_{\boldsymbol{\alpha}} \left\{ \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 + \lambda c\left(\boldsymbol{\alpha}\right) \right\}. \quad (7)$$

The constraint that $\boldsymbol{\alpha}_p = \{0,1\}^M$ is first omitted and is then added back with threshold when $\boldsymbol{\alpha}_p$ converges. Eq. (7) is non-convex. We further employ axillary vectors $\mathbf{h}$ and $\mathbf{v}$ to approximate $\partial_{x,t}\boldsymbol{\alpha}$ and $\partial_{y,t}\boldsymbol{\alpha}$ similar to Eq. (5). It yields

$$\min_{\boldsymbol{\alpha},\mathbf{h},\mathbf{v}} \left\{ \beta_1 \sum_p \|\boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0\|_2^2 + \lambda c\left(\mathbf{h},\mathbf{v}\right) + \beta_2 \sum_p \left(\|\partial_{x,t}\boldsymbol{\alpha}_p - \mathbf{h}_p\|_2^2 + \|\partial_{y,t}\boldsymbol{\alpha}_p - \mathbf{v}_p\|_2^2 \right) \right\}. \quad (8)$$

$c(\mathbf{h},\mathbf{v}) = \#\{p | ||\mathbf{h}_p||_2^2 + ||\mathbf{v}_p||_2^2 \neq 0\}$. We solve Eq. (8) again with two sub-optimization problems iteratively, similar to solving Eq. (5).

Estimate $(\mathbf{h},\mathbf{v})$ Eq. (8) is simplified to

$$(\widehat{\mathbf{h}},\widehat{\mathbf{v}}) = \arg\min_{\mathbf{h},\mathbf{v}} \left\{ \lambda c\left(\mathbf{h},\mathbf{v}\right) + \beta_2 \sum_p ||\partial_{x,t}\boldsymbol{\alpha}_p - \mathbf{h}_p||_2^2 + \beta_2 \sum_p ||\partial_{y,t}\boldsymbol{\alpha}_p - \mathbf{v}_p||_2^2 \right\}. \quad (9)$$

As the problem becomes independent of p, pixel-wise solution is yielded as

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

A larger λ means the structure term is more important, and more non-zero gradients are set to zeros. β_2 increases in iterations.

Estimate $\boldsymbol{\alpha}$ Eq. (8) is updated to

$$\widehat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left\{ \beta_1 \sum_p \| \boldsymbol{\alpha}_p - \boldsymbol{\alpha}_p^0 \|_2^2 + \beta_2 \sum_p \left(||\partial_{x,t}\boldsymbol{\alpha}_p - \mathbf{h}_p||_2^2 + ||\partial_{y,t}\boldsymbol{\alpha}_p - \mathbf{v}_p||_2^2 \right) \right\}, \quad (10)$$

This is a quadratic optimization problem with a closed-form solution. Its solution and relevant proof of the solver are included in the supplementary material downloadable from the project website.

β_1 and β_2 are initialized as 1. They automatically increase in iterations, as described in [27]. Our optimization quickly converges after 3-5 iterations. An example is shown in Figure 7. Experiments in [27] and for this method show convergence is not sensitive to initial β_1 and β_2 values.

5. Experiments and Results

Two datasets are used for evaluation: one is the Grand Central Train Station dataset [32] 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 more than 8 million pixels) uniformly sampled from the two

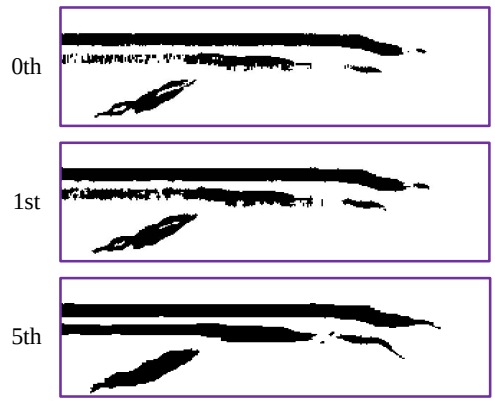
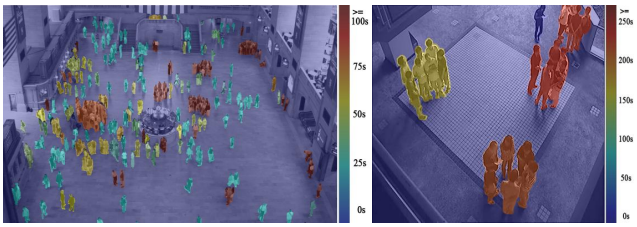


Figure 7. Illustration of convergence of α in the $x-t$ plane. Initial estimate and update in different iterations are shown. Noise is gradually removed.



(a) (b)

Figure 8. Annotated stationary time maps on the Grand Central dataset [32] (a) and the dataset collected by us (b). **Images are better viewed in color**.

datasets are annotated at the pixel level. Examples of annotated stationary time maps are shown in Figure 8. Details of these datasets are included in Table 1.

Each video is segmented into one-minute clips. There is 25% overlap between neighboring clips. Our current unoptimized MATLAB implementation on Intel CPU @3.3GHz takes 12 minutes to process a one-minute video clip.

η in Eq. (1) and λ in Eq. (4) are the parameters to be set. Their choice depends on the quality of initial background subtraction results and confidence on the sparse prior. We empirically set $\eta = 1.5$ and $\lambda = 20$.

Different types of evaluation are conducted. The average estimation **e**rror on stationary **t**ime (ET) for all foreground pixels is obtained. In addition, we compute the *ratio* between the estimation error and ground truth for each foreground pixel. Then all the ratios on foreground pixels are averaged. This measure is denoted as average estimation **e**rror **r**atio on stationary **t**ime (ERT). If a pixel has become stationary longer than 10 seconds up to the current frame, it is regarded as stationary. Several detection measures are used including (1) false alarm rate (FAR); (2) missed detection rate (MDR); and (3) total error rate (TER).

We compare our results with those of several background subtraction methods including improved adaptive Gaussian mixture model [33], codebook model [11], and adaptive Bayesian model [21]. Stationary time is accumulated if a pixel is detected as foreground. We also test dense tracking [24] on detected foreground pixels [11]. Stationary time is estimated as the length of the trajectory since a pixel becomes foreground. We also report the result of replacing the second-order gradients (Eq. (2)) with first-order ones (Eq. (3)) in our approach. This simplified version is denoted as "Ours (FOrder)".

Table 1. Details of Datasets

	Dataset I [32]	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

Table 2. Results of stationary time estimation on Dataset I. ET is measured in seconds.

Methods	FAR	MDR	TER	ET	ERT
Ours	0.29%	**3.49%**	**0.39%**	**10.04**	**12.21%**
Ours (FOrder)	0.51%	5.90%	0.69%	16.12	26.77%
GMM [33]	0.27%	24.51%	1.11%	29.46	43.98%
Codebook [11]	**0.26%**	21.03%	0.93%	29.51	40.14%
Bayesian [21]	0.33%	20.18%	1.01%	26.70	39.16%
Tracking [24]	0.30%	24.26%	1.09%	40.78	56.49%

Table 3. Results of stationary time estimation on Dataset II. ET is measured in seconds.

Methods	FAR	MDR	TER	ET	ERT
Ours	**0.91%**	**0.54%**	**0.86%**	**15.88**	**8.67%**
Ours (FOrder)	1.37%	0.98%	1.32%	16.90	10.68%
GMM [33]	0.92%	16.24%	3.06%	57.41	39.76%
Codebook [11]	1.03%	13.37%	2.75%	58.28	40.67%
Bayesian [21]	1.05%	12.26%	2.60%	45.20	32.19%
Tracking [24]	0.92%	5.75%	1.60%	54.14	38.86%

5.1. Result Analysis

The results on the two datasets are reported in Tables 2 and 3. Overall, our approach outperforms all the other alternatives on both the indoor and outdoor datasets. Its false alarm rate is slightly higher than a few methods because of the smoothing effect yielded by the high sparsity prior. Its misdetection rate is 6-15 times lower. The stationary time estimation error is also at least 2.5 times lower than other methods. If some shadow cannot be perfectly removed by initial background subtraction, false alarms may be caused.

With large misdetection rates and estimation errors, background subtraction is not that suitable for stationary time estimation. In general, the adaptive Bayesian model

[21] works better than other approaches, because it adds smoothness constraints in the spatial domain and between two successive frames. But it is still not similarly good as ours because of various reasons discussed in Sections 1 and 2. The smoothness prior causes more false alarms than ours, which manifest the necessity to employ the second-order gradients sparse priors.

6. Applications

We present a few applications of stationary groups and their formation duration estimation.

6.1. Stationary Group Activity Detection

We apply our method to detecting stationary group activities. The ground central train station images [32] contain various stationary group activities. We detect the four types of activities illustrated in Figure 2 because they are common in crowd surveillance. Ground truth is manually annotated on this dataset. 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 easily in surveillance applications. A detector is trained for each type of activities separately, searching the entire video sequence. A true positive is counted if the overlap between a detected group and the ground truth is larger than 50% in the spatio-temporal space. The numbers of training and test samples are summarized in Table 4.

A good detector has three key features. (1) It should accurately estimate the stationary period and identify emergence and dispersal activity of a stationary group. (2) It should cluster detected stationary pixels into groups. Our method employs mean shift for clustering given our fairly reasonable pixel-level time estimation. (3) Motion descriptors to characterize emergence, dispersal, and group deformation should be resulted in. We propose 12 stationary group descriptors ($\{\mathcal{D}_1, \ldots, \mathcal{D}_{12}\}$) based on keypoint trajectories extracted with a KLT tracker, detailed below.

$\mathcal{D}_1$-$\mathcal{D}_4$ 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. The histograms of incoming trajectories over time ($\mathcal{E}_T$) and directions ($\mathcal{E}_A$) are computed. The dominant modes ($\mathcal{M}_T$ and $\mathcal{M}_A$) of the two histograms are obtained by mean shift. We set $\mathcal{D}_1 = |\mathcal{M}_T|/|\mathcal{E}_T|$ and $\mathcal{D}_3 = |\mathcal{M}_A|/|\mathcal{E}_A|$, where $|\cdot|$ denotes the size of a mode or histogram. $\mathcal{D}_2$ and $\mathcal{D}_4$ are the variance of $\mathcal{E}_T$ and $\mathcal{E}_A$.

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

Table 4. Activity Detection Result (False Alarm / Misdetection)

Activities	Gather	Stop by	Relocate	Deform
Training samples	30	30	30	30
Test samples	45	58	27	50
Ours	3 / 6	5 / 6	4 / 1	6 / 4
GMM [33]	4 / 23	6 / 25	4 / 9	7 / 19
Codebook [11]	3 / 22	4 / 23	4 / 8	7 / 18
Bayesian [21]	2 / 23	4 / 24	3 / 8	6 / 17
Tracking [24]	4 / 25	5 / 28	5 / 12	6 / 20

$\mathcal{D}_{10}$-$\mathcal{D}_{12}$ denote whether a stationary group keeps its internal structure stable or not. They compute the topological variation of feature points inside the stationary group. For a point i in the group, its K-nearest neighbors $\mathcal{N}_t(i)$ and topology of neighbors tend to be stable. We use $\mu_t(i) = 1 - |\mathcal{N}_t(i) \cap \mathcal{N}_{t-\Delta}(i)|/K$ to measure the portion of varying neighbors from time $t-\Delta$ to t. The K' invariant neighbors are ranked according to their distances to i. $\mathcal{R}_t(i) = [\sigma_t^1(i), \ldots, \sigma_t^{K'}(i)]$ and $\mathcal{R}_{t-\Delta}(i) = [\sigma_{t-\Delta}^1(i), \ldots, \sigma_{t-\Delta}^{K'}(i)]$ are the rankings of neighbors at time t and $t-\Delta$. $\varsigma_t(i)$ is the distance between $\mathcal{R}_t(i)$ and $\mathcal{R}_{t-\Delta}(i)$. Similarly, $\kappa_t(i)$ is computed from rankings based on angles. Finally, $\mathcal{D}_{10}$-$\mathcal{D}_{12}$ are computed as the average over all the feature points during the whole stationary period based on $\mu_t(i)$, $\varsigma_t(i)$, and $\kappa_t(i)$, respectively.

Given $\mathcal{D}_1$-$\mathcal{D}_{12}$ learned, linear SVM is used as the classifier. More details are in our supplementary material. Table 4 reports the numbers of false alarms and missed detection by different approaches. All methods use the same clustering method and group descriptors. They differ by the way to estimate the stationary time as described in the previous section. Because stationary time estimation is essential in this application, our approach achieves the best results due to its robustness to suppress noise.

6.2. Scene Understanding

Stationary time estimation can help scene understanding and provide valuable statistics computed over time. For example, an averaged stationary-time map computed over all the groups in four hours on grand central train station videos is shown in Figure 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 often appear at an entrance to a building, alarm can be triggered for taking further actions to improve traffic there.

7. Conclusion and Future Work

We have explored a new research topic of stationary crowd group analysis, which has many important applications. A fundamental step towards getting useful informa-

tion is to estimate the stationary time of foreground pixels, which cannot be solved with existing background subtraction techniques. We propose a robust algorithm that adopts a locally shared foreground codebook and uses second-order gradients to shape the 3D stationary time map. It is formulated as a L_0 minimization problem and is solved by a practically effective scheme. The proposed method, as well as the research topic, can be applied to detecting stationary group activities and crowd scene understanding.

We believe it will find many more interesting and valuable applications in future. For example, it may be incorporated into existing systems to model the influence of stationary groups on changing moving traffic and predicting social relationship among pedestrians. The potential to study this problem and deploy our solution is boundless.

Acknowledgments

This work is supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. 417110, 417011, 429412, and 413113).

References

[1] S. Ali and M. Shah. A lagrangian particle dynamic approach for crowd flow segmentation and stability analysis. In *CVPR*, 2007.

[2] M. R. Amer and S. Todorovic. A chains model for localizing participants of group activities in videos. In *ICCV*, 2011.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

[4] A. B. Chan, Z. S. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008.

[5] M. Chang, N. Krahnstoever, and W. Ge. Probabilistic group-level motion analysis and scenario recognition. In *ICCV*, 2011.

[6] K. Chen, S. Gong, T. Xiang, and C. C. Loy. Cumulative attribute space for age and crowd density estimation. In *CVPR*, 2013.

[7] 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 *BMVC*, 2011.

[8] Y. Fu, M. Hopedales, T. Xiang, and S. Gong. Attribute learning for understanding unstructured social activity. In *ECCV*, 2012.

[9] W. Ge, R. T. Collins, and R. B. Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE Trans. on PAMI*, 34:1003–1016, 2012.

[10] I. Haritaoglu and M. Flickner. Detection and tracking of shopping groups in stores. In *CVPR*, 2001.

[11] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging*, 11:172–185, 2005.

[12] T. Lan, Y. Wang, W. Wang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *IEEE Trans. on PAMI*, 34:1549–1562, 2012.

[13] C. C. Loy, T. Xiang, and S. Gong. Incremental activity modelling in multiple disjoint cameras. *IEEE Trans. on PAMI*, 34:1799–1813, 2012.

[14] V. Mahadevan, X. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

[15] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *CVPR*, 2009.

[16] M. Moussaid, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PLoS ONE*, 5, 2010.

[17] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.

[18] M. Rodriguez, J. Sivic, I. Laptev, and J. Audibert. Data-driven crowd analysis in videos. In *CVPR*, 2011.

[19] P. Scovanner and M. F. Tappen. Learning pedestrian dynamics from the real world. In *ICCV*, 2009.

[20] J. Shao, C. C. Loy, and X. Wang. Scene-independent group profiling in crowd. In *CVPR*, 2014.

[21] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11):1778–1792, 2005.

[22] B. Solmaz, B. Moore, and M. Shah. Identifying behaviors in crowd scenes using stability analysis for dynamical systems. *IEEE Trans. on PAMI*, 34:2064–2070, 2012.

[23] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[24] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[25] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. In *CVPR*, 2007.

[26] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Trans. on PAMI*, 31:539–555, 2009.

[27] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. *ACM Trans. Graphics*, 30, 2011.

[28] K. Yamaguchi, A. C. Berg, T. Berg, and L. Ortiz. Who are you with and where are you going. In *ICCV*, 2011.

[29] Z. Zhong, W. Ye, S. Wang, M. Yang, and Y. Xu. Crowd energy and feature analysis. In *ICIT*, 2007.

[30] B. Zhou, X. Tang, and X. Wang. Coherent filtering: detecting coherent motions from crowd clutters. In *ECCV*, pages 857–871. 2012.

[31] B. Zhou, X. Tang, and X. Wang. Measuring crowd collectiveness. In *CVPR*, 2013.

[32] B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *CVPR*, 2012.

[33] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR*, 2004.