Multimed Tools Appl (2014) 72:453–487
DOI 10.1007/s11042-013-1367-2

Counting moving people in crowds using motion statistics of feature-points

Mahdi Hashemzadeh · Gang Pan · Min Yao

Published online: 9 February 2013

Abstract Reliable people counting is a crucial task in video surveillances. Among the available techniques, map-based approaches have shown a good performance in estimating the number of people in crowds. These approaches generally subtract the background, and then map the number of people to some features such as foreground area, texture features or edge count. However, in complex scenes, they suffer from inaccurate foreground/background segmentations, erroneous image features, and require large amount of training data to capture the wide variations in crowd distribution. This paper proposes a method using motion statistics of feature-points to estimate the number of moving people in a crowd. Simple feature-points are tracked within the scene. Then moving feature-points are partitioned into clusters corresponding to separate groups of people. For each group, three statistical features are calculated from related feature-points. The amount of moving feature-points is used to provide a rough estimate of group size. Furthermore, motion trajectories of feature-points are utilized to extract two other features related with the amount of occlusions present in groups. The extracted data are used to estimate the number of people in each group, so that the total crowd size is the sum of all group estimates. The experimental results show that the proposed method outperforms the state of the art approaches, e.g., with MSE of 2.357 and MAE of 1.093 for the benchmark video clip "Peds1". The proposed approach is good for estimating the number of

M. Hashemzadeh (✉) · G. Pan (✉) · M. Yao
College of Computer Science and Technology, Zhejiang University, Hangzhou, China
e-mail: gpan@zju.edu.cn

M. Hashemzadeh
e-mail: hashemzadeh@zju.edu.cn

M. Yao
e-mail: myao@zju.edu.cn

people in public places, such as pedestrian walkways and parks, where people are moving and partial occlusions present in the scene.

Keywords People counting · Crowd counting · Feature point · Feature tracking · Occlusion · Video surveillance

1 Introduction

Counting people in crowds is a crucial and challenging problem in video surveillance applications. An accurate estimation of the number of people in a crowd is a key indicator of the crowd security and safety, and it can be extremely useful information for economic purposes, resource management, scheduling public transportation, indexing multimedia archives, or advertising. Some other important applications have to do with estimating the number of people in over-populated places. For example, knowing the size and density of a crowd outside a school or public event can be helpful for the detection and early warning of unsafe situations. This capability can be useful in planning evacuation strategies [10, 13, 26], which is probably on the top of the list of public safety issues. Hence, a lot of work based on computer vision technology has been done to collect this data automatically.

Current approaches to this problem are generally classified into three categories: model-based methods, trajectory-clustering-based methods, and map-based methods (also called measurement-based or feature-regression-based). In the first two categories, people in the scene are first detected individually, and then the total number of people in the scene is counted. The model-based approaches attempt to segment and detect every single person in the scene using a model or appearance of human [14, 22, 24, 25, 28, 36, 41–43], and the trajectory-clustering-based approaches try to detect every independent motion by clustering interest-points on people tracked over time [4, 7, 34]. In contrast, the map-based approaches count the number of people without having to segment or detect each individual [1, 5, 6, 8, 9, 16, 18, 19, 30–32, 35, 37]. These approaches generally map the number of people to foreground pixels or some other features by training. The map-based method is considered to be more robust, since the correct segmentation and detection of people is a complex problem that cannot be solved accurately, especially when occlusions are present in the crowd.

Map-based approaches usually subtract the background and then utilize holistic features (i.e. features from the entire crowd) from the scene such as foreground area [6, 9, 16, 32, 37], texture features [6, 30, 35], edge count [6, 9, 37], or histograms of edge orientations [6, 19, 37] to estimate the crowd density by a regression function, e.g. linear [9, 32], Gaussian process [6], or neural networks [8, 16, 18, 19, 31, 37]. Almost all of these approaches have shown that the relationship between the foreground area and the number of people in the scene is approximately linear. However, this relation usually fails due to the occlusions and perspective problem.

In order to overcome the effects of perspective, a variety of techniques have been proposed in the literature. For example, a geometric factor was used in [32] to weight each pixel according to its location on the ground plane; or a perspective map was proposed in [6] to weight all extracted features from image. The problem of occlusions have been mitigated by using additional features, e.g. edge count in [9], histograms of edge orientations in [19], or by using a great quantity of features in [6]. The approach in [6], used a mixture of dynamic textures to segment the foreground motion in two directions. Then, a large number of features were extracted including foreground area, edge orientation histogram, perimeter pixel count and textural features. In total, 29 features were extracted to estimate the number of people walking in each direction. However, in complex scenes, these approaches suffer from some issues as follows:

- The foreground/background segmentation process which is needed in these approaches is by itself a difficult task that cannot be carried out accurately in crowded scenes.
- Edge-based features that are usually used in these methods can be extremely erroneous because the edges are completely messy when the background is complicated and the textures of human clothes are not smooth.
- Extracting a large amount of features, especially edge extraction, is very time-consuming.
- Because of the wide variability in crowd density and distribution, using holistic features from scene can give rise to extensively different features, and therefore a large amount of training data is required to capture the various crowd distributions.

To deal with the later problem and in order to reduce the required training data, some approaches proposed to use local features rather than holistic features. Local features are specific to one person or a group of people in the scene, while holistic features are calculated from the entire crowd. The important advantage of the local features is the availability of more training cases in one training image. Therefore, extracting local features from scene can help to capture various crowd distributions from a small amount of training data. For example, the approach proposed in [5] assumed a linear relationship between blob size and group size, and method in [17] used an elliptical cylinder model to estimate the number of people moving in separate groups. A supervised learning framework was proposed in [23], which estimates an image density whose integral over a region of interest yields the number of people. Approach in [37] used a foreground segmentation algorithm to obtain blobs in an image and then utilized several blob features to estimate group sizes, however obtained blobs are still prone to errors due to imperfect foreground segmentation in densely crowded scenes.

Recently, Albiol et al. [1] proposed a map-based approach and used corner-points as features. In this approach, firstly, corner-points are detected using Harris corner detector [15]. Then, moving corner-points (foreground corner-points) are distinguished by computing motion vectors between adjacent frames. Finally, the count of moving people is estimated from the number of moving corner-points in the scene. The estimation is done based on a direct proportionality relation with a constant factor determined using one frame of the video sequence. Although the underlying assumption in this method may appear rather simplistic, it was the winner of the PETS 2009 [33] contest on people counting task and has proven to be quite robust when compared to more sophisticated competitors [12]. Beside the good performance, this approach has other important advantages: no need to estimate the background, no need to deal with shadow problems, no need to segment foreground/background areas or individuals, and no need to extract a large amount of complicated features. However, it still meets problems in complex scenes and the attained accuracy is limited by the perspective effects and the presence of occlusions in various crowd distributions.

In this paper, we propose an approach using the motion statistics of low-level feature-points (FPs) to estimate the number of moving people in a crowd. Similar to Albiol's method, the amount of moving-feature-points (MFPs) in the scene is used as a clue to the foreground area, which can provide a coarse estimate of crowd size. Furthermore, our approach utilizes the motion trajectories of MFPs to capture other clues about crowd complexity. Our motivation is to extract some statistical features[1] from MFPs that are highly correlated with the level of occlusion present in a crowd. We introduce two occlusion-related features, namely the rate of boundary-feature-points and the mean duration of torso-feature-points, to produce more accurate estimates of the crowd size. In order to make our system generalizable to various crowd distributions, these features are extracted on a local level, i.e. local with respect to the groups of people moving together. To this end, MFPs are partitioned into clusters corresponding to separate groups of moving people in the crowd, and then statistical features are calculated for each group separately. Finally, the extracted features are used to estimate the number of people within each group, so that the total crowd size is the sum of all group estimates.

The proposed approach is evaluated on a large pedestrian dataset, containing very distinct camera views, locations, and pedestrian traffic. The proposed system is compared to Albiol's method and some other map-based approaches. The results show that our approach outperforms those methods. It is also shown that counting crowd in separate groups of people results in a quite robust and generalizable approach that is capable of extrapolating to count crowds not encountered during training and can be trained on a small amount of training data.

The proposed system is suitable to count the number of moving people in public places such as busy pedestrian walkways, shopping malls, parks, etc. where people are moving and partial occlusions present in the scene. Although the feature extraction based on motion information of FPs might be not accurate enough in highly dense crowds and mass gatherings such as music festivals, sports events or pilgrimage where it is not possible to generate meaningful trajectories of FPs, our experiments on very crowded videos show that the proposed method is able to provide a rough estimate of crowd size in such scenes.

The rest of this paper is organized as follows; Section 2 presents the proposed crowd counting system. Our experimental results are presented and analyzed in Section 3. Finally, in Section 4 we draw conclusions and discuss possible directions for future work.

[1] To avoid confusion, "feature" will be used when referring to the statistical features extracted from feature-points and "feature-point (FP)" will be used when referring to feature-points detected for tracking.

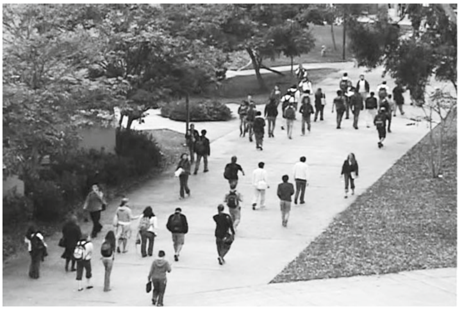


Fig. 1 An example frame of a crowd containing pedestrians moving away from, and towards the camera

2 Crowd counting using motion statistics of FPs

2.1 System overview

An example frame of a crowd containing pedestrians moving away from, and towards the camera is shown in Fig. 1. The goal is to estimate the number of moving people by using features that we extract from a set of FPs and their trajectories within each time-window of the video sequence. Firstly, a number of FPs are tracked within a specified time-window of the video. To detect and track the FPs, we use the KLT tracker [27, 39, 40], which has been widely used in people tracking [2, 34, 38]. Then, MFPs are separated from static FPs (background FPs). An example frame containing detected FPs inside the region of interest is shown in Fig. 2. Afterwards, in order to extract local features from MFPs, they are clustered into clusters corresponding to one person or a group of people in the scene. Then, three statistical features are extracted from each cluster in order to capture some clues about the number of people and occlusion level within each cluster. Finally, a classifier is trained to map the extracted features to the number of people within each group, and the total count for the scene is calculated by the sum of the group estimates.

In order to train this system, the ground truth annotation must specify a crowd count for separate groups of people in the scene. Therefore, ground truth annotation is performed after MFP clustering step, once the groups of people are detected. The number of people is manually counted for each group in an image; therefore each frame provides several instances of ground truth. This results in a system that is

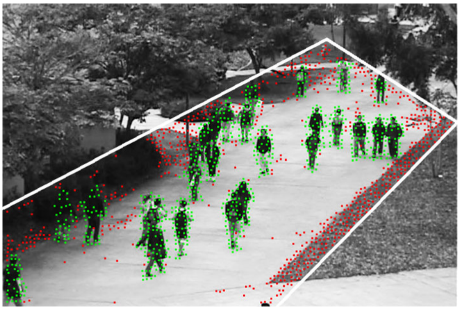


Fig. 2 An example frame including detected FPs. The MFPs are drawn in green and the static FPs in red

generalizable to crowd volumes not seen in the training set and can be trained on a small data, comparing with system that extract features from whole scene and each frame contains one instance of ground truth.

In the extraction of statistical features from FPs it is important to consider the effects of perspective, which cause that the farther the person is from the camera, the fewer are the detected FPs. Because objects far from the camera appear smaller than objects closer to the camera (see Fig. 2). One possibility is to weight each FP according to a perspective normalization map. In this work, we calculate the perspective map in the same manner as [6]. The calculated perspective map is used to weight FPs when we calculate any features from clusters of MFPs. In a more general setting, one can use the calibration methods presented in the literature for example [20].

In order to count people at a given frame, our algorithm processes the set of trajectories bounded by a finite time-window $\tau = \pm 2$ s, which spans equally forwards and backwards in time with respect to the current frame (i.e. at a frame rate of 10 fps, the time-window is 40 frames). The time-window is shifted frame by frame to process the entire video, which means that the counting is performed for all frames and for each one independently. In the next subsections we present the details of the proposed crowd counting approach.

2.2 Clustering of the MFPs

In order to extract local features from MFPs, the algorithm needs to partition the MFPs into clusters corresponding to separate groups of people. In a crowded scene, people can appear in different positions and can be gathered in many different ways. Therefore, we cannot apply commonly used clustering methods (such as k-means) to cluster the MFPs, as we do not have any prior knowledge about the number and the shape of the clusters. For such a clustering problem, we use a scheme based on the graph theory. The set of MFPs in a frame are represented as a graph in which each MFP corresponds to a node. To build the edge set, an adjacency rule is defined based on the minimum distance between two FPs in the KLT tracker. We assume there exist an edge between two MFPs if distance between them is equal or less than $3q$, where q is the minimum distance between two FPs in the KLT tracker. Since FPs cannot be closer than q, we choose $3q$ as the radius of adjacency. Finally, the obtained graph is traversed (using standard algorithms such as breadth-first-search (BFS) or depth-first-search (DFS)) to find its connected components, corresponding to separate groups of people in the scene. Two sample frames containing the clustered MFPs are demonstrated in Fig. 3.

2.3 Extracting statistical features

After partitioning the MFPs into clusters, statistical features are extracted for each cluster in a given frame. Considering the good performance shown by Albiol's method in [1], it is expected that there should be a nearly linear relationship between the number of MFPs and the number of moving people in a crowd. Figure 4 plots the number of MFPs versus the crowd size on a part of "Peds1" video (see Subsection 3.1). While the overall trend is indeed linear, there exist local non-linearities that mostly arise from occlusion and the effects of perspective. Normalizing FPs by

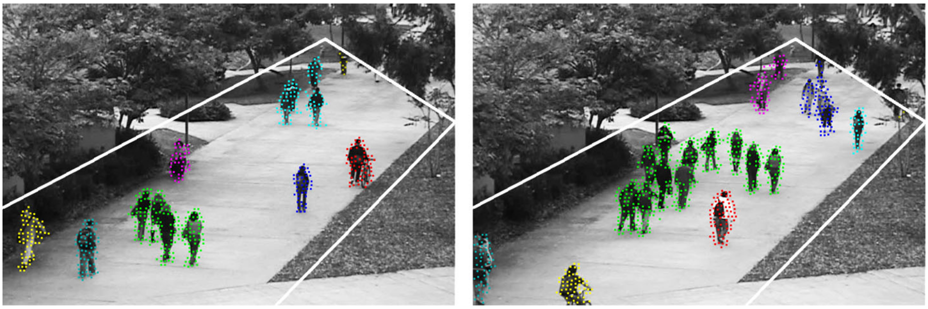


Fig. 3 Two sample frames containing the clustered MFPs. Clusters are depicted in distinct colors

using the calculated perspective map will compensate for the effects of perspective. Accordingly, for each cluster of MFPs, we calculate the *amount* of MFPs instead of the *number* of MFPs as follows:

$$|M_c^t| = \sum_{i \in M_c^t} w_i \tag{1}$$

where M_c^t is the set of MFPs in c-th cluster at frame t and w_i refers to weight of i-th FP in perspective map.

The remaining problem is occlusion. Indeed, the estimation function fails when occlusions happen and overlapping people cover some body-parts of each other. Consequently, only a subset of the corresponding FPs are detected. To model these non-linearities, we extract other features for each cluster. We hypothesize that there

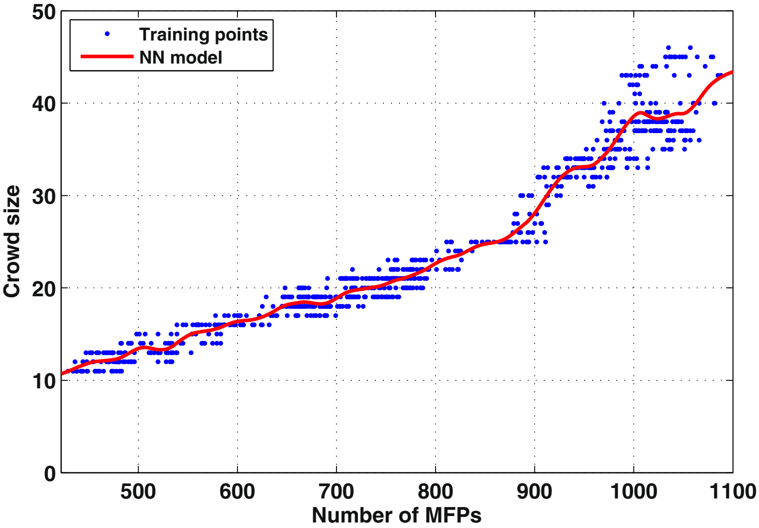


Fig. 4 Correspondence between number of MFPs and crowd size on a part of "Peds1" video

could be some relations between the level of occlusion (complexity of the crowd) and the motion statistics of MFPs. Therefore, in order to extract occlusion-related features from a cluster of MFPs, we utilize motion trajectories of MFPs within a specified time-window (see Subsection 2.1).

To extract the desired occlusion-related features, an initial pitfall is the detected FPs on the limbs of people, which are named as limb-feature-points (LFPs). These FPs usually move with varying speeds and in various directions, and their motion is not consistent with the other FPs on torso and head of the corresponding human, which are named as torso-feature-points (TFPs). Since our aim is to find some motion-based features that are highly correlated with the level of occlusion in a crowd, these irregular movements of LFPs prevent us from making an accurate statistical analysis of the MFPs' motion information. For example they are often lost very soon during the tracking, so they usually have very short trajectories; distance between them and other FPs varies continuously, etc. Therefore, before extracting occlusion-related features, we need to distinguish and separate the LFPs from the set of MFPs. In the following subsections, firstly, we present how LFPs are detected and then introduce the other features extracted from remaining MFPs.

2.3.1 Detecting LFPs

Due to the swinging motion of LFPs, they usually move with various speeds along the successive frame sequences, while other FPs on stable body-parts (i.e. TFPs) almost always have a continuously uniform motion with a constant speed. We make use of this fact to distinguish LFPs form TFPs. We assume that an FP is more likely on a limb if the variance of its motion speed is large.

Let T_i refer to the trajectory of i-th FP, $T_i^t = \left(x_i^t, y_i^t\right)$ is its coordinates at time t, $\boldsymbol{v}_i^t = \left(v_{x_i}^t, v_{y_i}^t\right) = \left(x_i^t - x_i^{t+1}, y_i^t - y_i^{t+1}\right)$ is its motion vector at time t, $s_i^t = \sqrt{\left(v_{x_i}^t\right)^2 + \left(v_{y_i}^t\right)^2}$ refers to its speed (magnitude of motion vector in pixel/frame) at time t, and Δ_i is for the whole lifetime of T_i (i.e. frames within the specified time-window that T_i has been tracked along them). In order to determine that whether i-th FP is an LFP or not, the variance of its motion speed along time (i.e. $\left\{s_i^t; \quad t \in \Delta_i\right\}$) is checked. If the variance value was greater than a specified threshold the FP is treated as an LFP. As an example, Fig. 5 shows the speeds of two sample FPs along the given time-window, where one FP is an LFP and the other is a TFP. In the figure, comparing speed variations of two FPs, one can observe that the speed of the LFP varies drastically within its lifetime while there is only a slight speed reduction along the time for the TFP. This speed reduction with time is caused by the perspective effect, i.e. in this case ("Peds1" video) the corresponding person is moving away from the camera. Although this variation is much smaller than the speed variance of the LFP (because perspective scale varies less during our considered time-window), we deem that these speed variations could be excessive in other cases and may affect the LFP detection process, therefore, we enrich our analysis as follows.

As shown in Fig. 5, in contrast to the LFPs, the speed of TFPs changes smoothly with time, and the speed difference between two successive frames is usually very slight. Therefore, to cope with the effects of perspective, firstly, we calculate the speed-difference between two successive frames during the lifetime of the motion trajectory of an FP, and then check the variance of the set of speed-differences. More precisely, instead of considering variance of $\left\{s_i^t; \quad t \in \Delta_i\right\}$, the variance

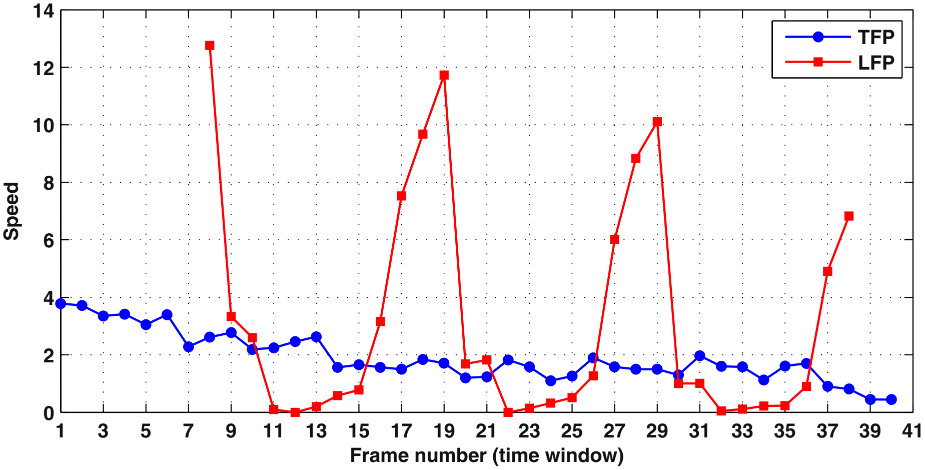


Fig. 5 The speed of two FPs over time. There is more variation in speed of LFP

of $\left\{\left(s_i^t - s_i^{(t+1)}\right); \quad t \in \Delta_i\right\}$ is taken into account. Figure 6 shows the speed-differences between successive frames for same FPs in Fig. 5. It can be seen that the variance of speed-differences for TFP is very small (almost equal to zero) while for LFP, it still has high variations. As a result, we can easily distinguish LFPs by applying a minimal threshold on the variance value. Therefore, *i*-th FP is an LFP if the following condition is satisfied:

$$Var\left(\left\{\left(s_i^t - s_i^{t+1}\right); \quad t \in \Delta_i\right\}\right) > \Phi \tag{2}$$

where Φ is a threshold value.

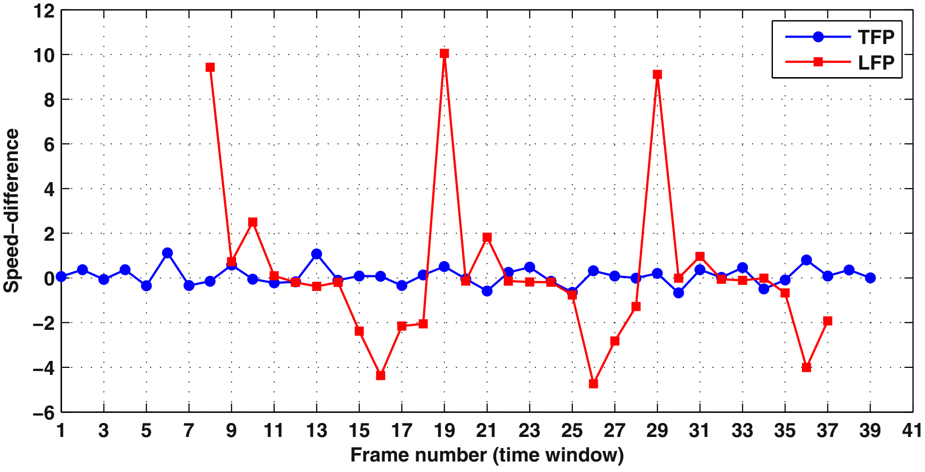


Fig. 6 The speed-differences between two successive frames for two FPs over time. The variance of speed-differences for TFP is very small (almost equal to zero)

After removing LFPs from the set, we can analyze the motion behavior of remaining TFPs to find the features related to the level of occlusion present in each group of people.

2.3.2 Extracting occlusion-related features from TFPs

Two statistical features for capturing the various levels of occlusion are extracted from TFPs of each cluster as follows:

(1) Rate of boundary-feature-points

Our aim is to discover the level of occlusion existing in a crowd of people. One possibility is to determine how much overlapping people are present in the crowd. Since overlapping people share some overlapping boundaries, the amount of shared boundaries increases as the occlusions increase. Hence, we can acquire a clue to the level of occlusion by determining the amount of shared boundaries between people in a crowd. To capture this data, we try to determine the amount of FPs placed on the boundary regions between overlapping people. These FPs are named as boundary-feature-points (BFPs). In order to detect the BFPs, we use the temporal motion consistency of two adjacent FPs to discover whether these FPs belong to the same individual or they are placed on the overlapping boundary of two people which are treated as BFPs. We use the assumption that pairs of adjacent FPs corresponding to a region straddling two people are expected to have higher variance in their mutual distance when compared to pairs of adjacent FPs that are placed on one individual. A visualization of a sequence of FPs is provided in Fig. 7. As shown in the figure, the distance between two FPs on different persons is highly probable to vary with time, while the distance between FPs on the same person is almost constant.

To calculate the variance in distance of two adjacent i-th and j-th FPs with motion trajectories T_i and T_j, which are, respectively, extended in time over Δ_i and Δ_j, we

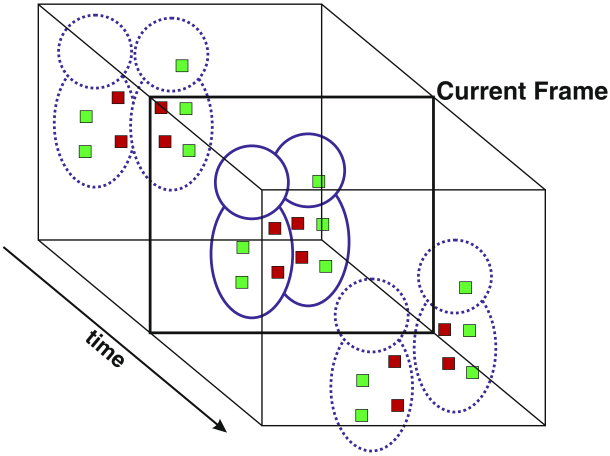


Fig. 7 A visualization of a sequence of FPs on two overlapping persons. BFPs are depicted in red and the others in green. Distance between two FPs on different persons is varying along the time, while distance between FPs on same person is almost constant

consider only the overlapping range of the time, $\Delta_i \cap \Delta_j$. We say two FPs are BFPs if the following condition is satisfied:

$$Var\left(Distance_{\text{Eucl}}\left(T_i, T_j\right)\right) > \theta \tag{3}$$

where,

$$Var\left(Distance_{\text{Eucl}}\left(T_i, T_j\right)\right) = Var\left(\left\{\sqrt{\left(x_i^t - x_j^t\right)^2 + \left(y_i^t - y_j^t\right)^2}\ ;\ t \in \Delta_i \cap \Delta_j\right\}\right) \tag{4}$$

and θ is a threshold value.

Our algorithm to detect the BFPs is as follows: For each FP in the set of TFPs in a cluster, we check its distance variance with remaining TFPs in its neighborhood using (3). If there is any FP that satisfies the condition, the under-analysis FP is treated as a BFP. To check the distance variance between each FP with its neighbors, we only consider those FPs that are within a box centered on the FP position at the time the trajectory begins. We choose $4q$ for the diameter of the box, where q is the minimum distance between two FPs in the KLT tracker. Because FPs cannot be closer than q, we choose $2q$ as the radius of adjacency, so $4q$ is the diameter. Algorithm 1 presents the detailed procedure of BFP detection process.

Algorithm 1 BFP Detection

```
Input: $\mathcal{F} = \{F_1, F_2, \ldots, F_n\}$ set of $n$ TFPs with trajectories $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$,
       threshold $\theta$
Output: $\mathcal{B}$: set of detected BFPs
Begin
Forall $T_i \in \mathcal{T}$ do
    $\mathcal{N} \leftarrow$ FindNeighbors($T_i$)
    if $\exists\ T_j \in \mathcal{N}$ subject to $\left(Var\left(Distance_{Eucl}\left(T_i, T_j\right)\right) > \theta\right)$ then
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{F_i\}$
end
End
```

FindNeighbors(T): Finds neighbor trajectories of T inside the defined neighborhood box.

After detecting BFPs in c-th cluster at frame t, we measure the amount of them, by taking into account their weights in the perspective map as follows:

$$|B_c^t| = \sum_{j \in B_c^t} w_j \tag{5}$$

where B_c^t is the set of detected BFPs in c-th cluster at frame t and w_j refers to weight of j-th FP in the perspective map. Finally, we measure the rate of BFPs as follows:

$$r_c^t = \frac{|B_c^t|}{|M_c^t - L_c^t|} \tag{6}$$

where r_c^t is the rate of BFPs in c-th cluster at frame t, L_c^t is the set of detected LFPs in c-th cluster at frame t, and $|M_c^t - L_c^t|$ is the amount of TFPs in c-th cluster at that frame. It is expected that the value of this rate increases as the amount of occlusions increase in crowds.

(2) Mean duration of TFPs

When moving people are well separated from each other in the scene, the TFPs are tracked excellently, and therefore, their motion trajectories usually have complete durations along the given time-window. However, in complex scenes, occlusions can result in some short trajectories due to the lost FPs in the tracking stage. Accordingly, we hypothesize that the duration of trajectories of TFPs (i.e. lifetime of TFPs) are highly correlated with the level of occlusion present within a group of people moving together. In other words, the number of short trajectories increases as the occlusions increase. The key idea is to compute the mean duration of trajectories of TFPs in each cluster. This parameter can tell us the occlusion level of the related group. We compute it as follows:

$$m_c^t = \frac{\sum_k |T_k| \cdot w_k}{\sum_k w_k} \quad , \quad k \in \left\{ M_c^t - L_c^t \right\} \tag{7}$$

where m_c^t is the mean duration of TFPs in c-th cluster at frame t, $|T_k|$ is the number of frames that the trajectory of the k-th FP is present in, w_k refers to its weight in the perspective map, and $\left\{ M_c^t - L_c^t \right\}$ is the set of TFPs in c-th cluster at frame t. It is expected that the value of this parameter decreases as the amount of occlusions increase in crowds.

2.4 Estimating crowd size

The extracted features from each cluster of MFPs serve as inputs to a classifier. The output of the classifier is the number of moving people within each cluster. Such a standard regression problem can be addressed by a multitude of machine learning tools. We use a single hidden layer feed-forward neural network to perform classification. Feed-forward neural networks are successful to model nonlinear systems [3], and have shown good performances in previous research [8, 16, 18, 19, 31, 37]. In our system, the input layer of neural network has three units, which correspond to features (i.e. the amount of MFPs, the rate of BFPs and the mean duration of TFPs) extracted from a cluster. There is only one output unit in the network, representing estimation of the people count for each cluster. Suppose the number of moving people in c-th cluster at frame t is denoted by X_c^t, and f expresses the relationship between number of moving people and the extracted features, as in (8).

$$X_c^t = f\left(|M_c^t|, r_c^t, m_c^t\right) \tag{8}$$

A training set is used for building the neural network to learn the relationship f, and the neural network is then used to estimate the number of people in clusters, and the total crowd estimate for frame t containing N_c clusters is calculated as:

$$E^t = \sum_{i=1}^{N_c} X_c^t. \tag{9}$$

3 Experiments

In this section, the proposed system is evaluated using different video sequences. Firstly, we examine the relationships between the extracted occlusion-related features and the level of occlusion present in the scene. Then, we report the counting results of the proposed approach on different video datasets, and compare them with results of other methods proposed in [1, 6, 9, 18, 19, 23, 37]. Afterwards, the results of our experiments are shown in order to demonstrate the effectiveness of the extracted features and perspective normalization scheme. Finally, the result of our experiment is presented in order to demonstrate the robustness and generalizability of our system against equivalent holistic system which calculate holistic rather than local features from scene.

3.1 Datasets

We use seven different videos with a large number of annotated frames for evaluation:

1. **"Peds1"** [6][2]: An oblique view of a walkway, containing a large number of pedestrians. The ground truth pedestrian counts inside a region of interest are available for 2000 frames of this video, featuring 11 to 46 people.
2. **"Peds2"** [29][3]: A side-view of a walkway, containing fewer people, compared with the "Peds1" dataset. In this video, the pedestrian movement is parallel to the camera plane. The ground truth pedestrian counts inside a region of interest have been provided for 2000 frames, consisting of 0 to 15 people.
3. **"USC"** [43]: A view of a walkway, consisting of 4 to 13 people captured from a camera above a building gate. We provide the ground truth counts for 300 frames of this video, manually.
4. **"Bridge"**: We captured this video on a bridge. The location is the entrance to a bridge stairway and people are walking very closely together and in various directions. The crowd density is variable, ranging from sparse to very crowded. We provide the ground truth counts for 1000 frames of this video, manually. It consists of crowds of size 6 to 30 people.
5. **"Gate"**: We captured this video with a camera mounted on top of a building gate. People are entering and exiting a building from various directions. The ground truth counts for 1000 frames of this video are provided, manually. It features crowds of size 1 to 22 people.
6. **"PETS2009"** [33]: This dataset is organized in four sections, but we use the section named S1 that was used to benchmark algorithms for the "estimation of the number of people in the field of view" PETS2009 contest. The ground truth count is obtained by annotating the number of moving people by hand, consisting of 0 to 43 people.

[2] Available at: http://www.svcl.ucsd.edu/projects/peoplecnt

[3] Available at: http://www.svcl.ucsd.edu/projects/anomaly

Table 1 Characteristics of five video sequences

	Peds1	Peds2	USC	Bridge	Gate
Length	1 hour	1 hour	30 sec	10 min	5 min
Frame rate[a]	10 fps	10 fps	10 fps	10 fps	10 fps
Frame size	720×480	720×480	360×240	640×480	640×480
Number of annotated frames	2000	2000	300	1000	1000
Number of training frames	160	80	60	66	66
Training frames	600:5:1400[b]	400:10:1200	1:5:300	1:15:1000	1:15:1000

[a] Refers to the number of annotated frames per second. The frame rate of all original videos is 30 fps.

[b] In Matlab notation. Within frames 600-1400, last frame of every five consecutive frames is selected.

7. **"Loveparade2010"** [21]: A video footage from crowd disaster at Loveparade 2010 in Duisburg, Germany. In this terrible stampede, 21 visitors died and more than 500 were injured. The festival area was monitored by seven cameras where three of them were static cameras. We use videos from "Camera 15" which is a static camera, and monitors huge number of visitors entering or exiting from the festival area.

Both of our own captured videos (i.e. "Bridge" and "Gate") are publicly available[4] to encourage future comparisons. Example frame of videos are shown in Figs. 14 and 19, and Table 1 summarizes some characteristics of each video. Due to existence of multiple scenarios in the experiments on "PETS2009" dataset and "Loveparade2010" video, the relevant information about these videos is reported in Subsections 3.4 and 3.8 respectively. As can be seen in Table 1, the resolutions of the videos used in our experiments are moderate. The KLT tracker can easily detect and track FPs on these videos. However, in the case of very poor image qualities, we may need to improve the efficiency of KLT tracker using techniques proposed in the literature such as [34].

A subset of annotated frames on each video is used to train the counting system, and the remaining frames are used for the testing purpose. The annotated frames and selected training frames for each video are reported in Table 1. In order to compare the performance of our system with the approaches in [6, 23, 37], the training and the testing frames on "Peds1" video are selected the same as in [6, 23, 37].

3.2 Parameter setting

3.2.1 Parameters in the KLT tracker

One of the parameters in the KLT tracker is the number of FPs. In our experiments, it is set to be large enough such that people show sufficient evidence of their existence. For all video sequences in our experiments, we extract the most significant 1200 KLT feature-points. Different numbers of FPs have been attempted in our evaluations. The results show that the counting system is not very sensitive to the number of FPs. However when the number of FPs is too small, there will be some people in the scene without any FPs detected on them, which can affect the counting results. Another parameter in the KLT tracker is the minimum distance between two FPs. For a

[4] Available at: http://www.cs.zju.edu.cn/~gpan/database/crowd.html

human being, FPs may be detected on the contours or clothing. Head-shoulder parts usually contain crucial KLT feature-points. Therefore, we set the minimum distance between two FPs such that the FPs from head-shoulder can be easily detected. The value of this parameter for all videos in our experiments is fixed to $q = 6$ pixels. However, in a more general setting, this parameter can be automatically adjusted in the training phase of the system and the optimal value is selected with respect to the best counting error rates. Finally, we use a fixed size of 7×7 pixels as the size of KLT feature window in all of the experiments.

Since the value of parameter q (minimum distance between two FPs) is also used to determine the radius of the specified neighborhood box in the BFP detection process, we test the sensitivity of this process to the value of q. We evaluate the accuracy of the BFP detection algorithm on different sets of FPs that are generated by using different values of q in the KLT tracker. To this end, after generating different sets of FPs on a video, we randomly select 100 FPs from each set and manually label the FPs on overlapping boundaries as BFPs. Then, we run the BFP detection algorithm on each set of the selected FPs. Figure 8 compares the false negative rate, false positive rate and total accuracy of the algorithm on different sets of FPs. As shown in this figure, the performance of the algorithm is slightly reduced as q decreases. The reason for this issue is that the small values of q results in a small radius for the defined neighborhood box. Consequently, there could be some BFPs that within the small search area around them, there will not exist any other BFPs, therefore, some BFPs are not detected by the algorithm. This fact is more obvious by considering the high false negative rates obtained by small values of q in the Fig. 8. Also, we can observe that by increasing the value of q, the performance of the algorithm is decreased remarkably. Because within a large search area (large radius of neighborhood box) around any FPs, it is highly probable to find some FPs that have inconsistent motions with under-analysis FP. As two FPs on one individual but far from each other do not always have a consistent motion. For example, as demonstrated in Fig. 8, almost all of the FPs are detected as BFPs when q is set

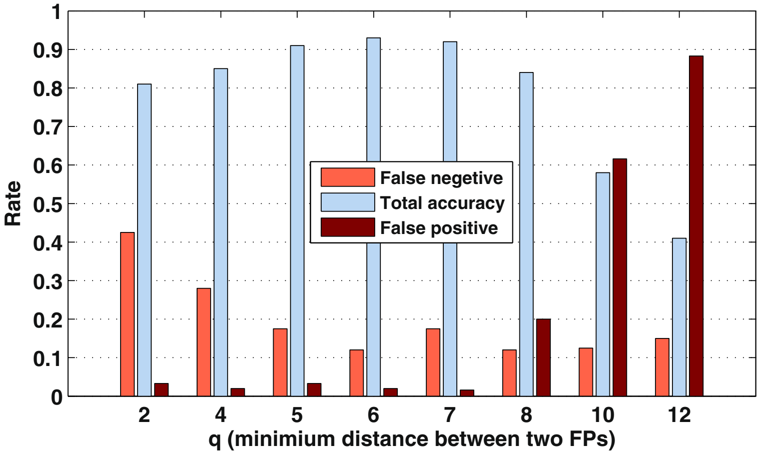


Fig. 8 Performance of the BFP detection algorithm on different sets of FPs generated using different values of q

to 12. In the case of using dense FPs (i.e. small values of q), in order to improve the robustness of BFP detection algorithm, we can consider a larger radius for the specified neighborhood box. However, for our application the fixed value of $q = 6$ pixels for all videos results in a satisfactory performance.

3.2.2 Adjacency radius in MFP clustering process

To partition the MFPs into clusters, an adjacency radius is defined based on the minimum distance between FPs in KLT tracker (see Subsection 2.2). It is obvious that two kinds of clustering error can occur if an unreasonable value is selected for this parameter; 1) some clusters will be split into several parts if very small radius is selected (for example people in one group are split into upper body and lower body parts), and 2) separate clusters of people will be joined together if the selected radius is very large. Accordingly, to test the sensitivity of the MFP clustering process regarding to the value of the adjacency radius, the accuracy of this process is evaluated using different values of this parameter. To this end, we randomly select some sample frames from each testing video containing more than 100 clusters of people, and manually label the desired clusters corresponding to the separate groups of people moving together. Then, we run the MFP clustering algorithm on the selected frames, using different radius values. Figure 9 compares the rates of error 1, error 2 and total accuracy obtained by the algorithm using different radius values. As shown in this figure, while very small and very large radiuses have influenced the clustering performance, there is not any remarkable change in the performance of the algorithm by using reasonable values for adjacency radius. For our application the fixed value of $3q$ for all videos almost avoids the occurrence of error 1. The error 2 happens in very rare cases which do not impact the overall performance of the system, because the joined clusters are considered as a large cluster of people similar to counting people on a holistic level.

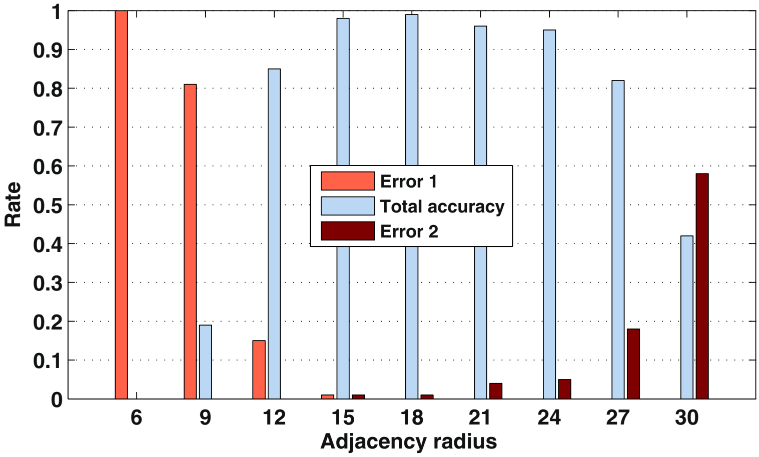


Fig. 9 Performance of the MFP clustering algorithm using different values of adjacency radius

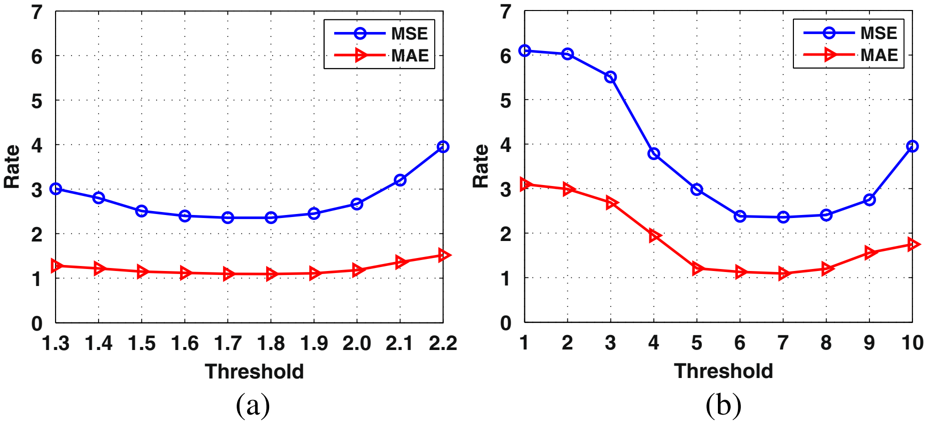


Fig. 10 Counting error rates obtained by different values of thresholds on "Peds1" video. **a** Threshold Φ, **b** Threshold θ

3.2.3 Parameters in the LFP and BFP detection processes

The values of thresholds Φ and θ in LFP and BFP detection processes are automatically adjusted during the training phase of the system. For each testing video, the counting system is trained using different combinations of Φ and θ values and optimal values of these thresholds are determined with respect to the best counting error rates. The optimal values found for each video are reported in Table 2. As an example, Fig. 10 shows the counting error rates (mean-squared-error and mean-absolute-error) obtained with different values of two thresholds on "Peds1" video. As shown in this figure, by using very large values of Φ or very small values of θ, the performance of the system is decreased remarkably. Because with large values of Φ LFPs are not detected accurately, and with very small values of θ most of the TFPs are detected as BFPs.

3.3 Experiment 1: Examining the relation between extracted features and level of occlusion

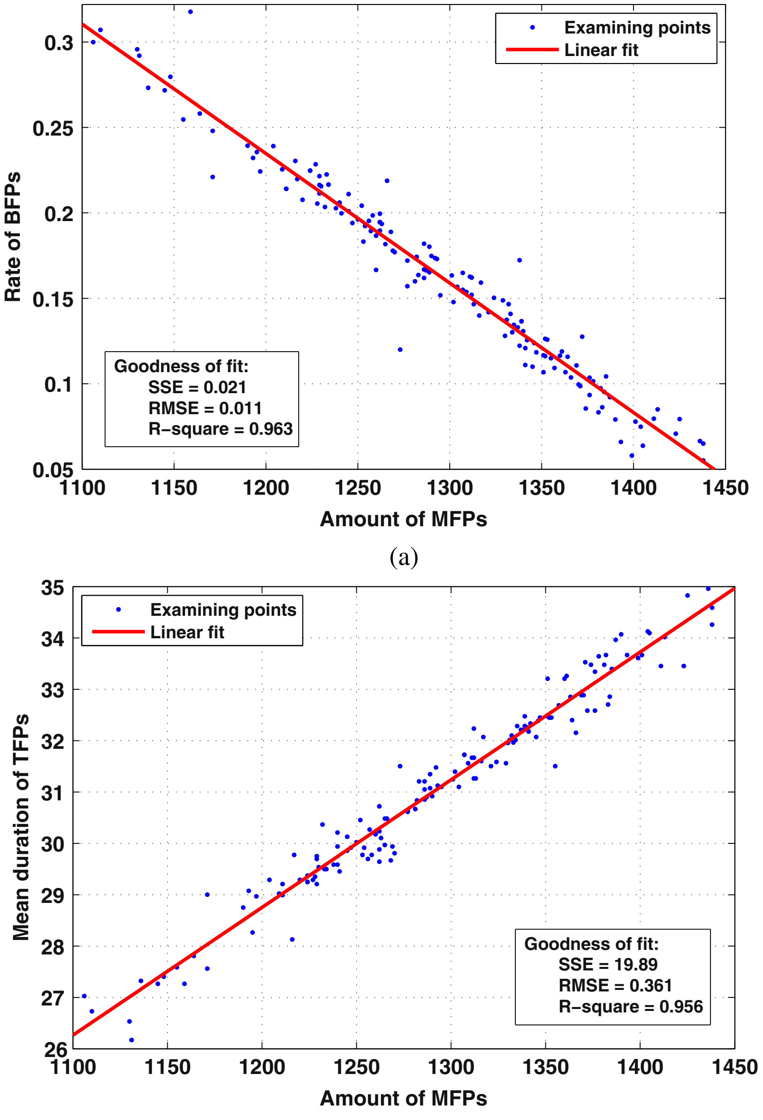
In order to demonstrate the relativeness of the extracted occlusion-related features, we examine the relationship between them and the level of occlusions present in the scene. To this end, we select all frames with 23 people from the "Peds1" video, which is the most common pedestrian count within the annotated range of this video (i.e. frames 600–1400). In total, there exist 154 frames with this count. The level of occlusion varies from frame to frame due to different amount of overlapping people in each frame. Thus, the amount of MFPs is different in different frames. In other words, although the number of people is the same in the selected frames, the

Table 2 Optimal values of thresholds used for each video

	Peds1	Peds2	USC	Bridge	Gate
Threshold Φ	1.8	2.1	1.5	1.7	1.8
Threshold θ	7	6	5	8	8

amount of MFPs is reduced as the occlusions increase. Therefore, we expect that the extracted occlusion-related features (the rate of BFPs and mean duration of TFPs) should be highly correlated with the amount of MFPs which varies with the amount of occlusions in the selected frames.

After calculating the amount of MFPs in selected frames, we measure the rate of BFPs and the mean duration of TFPs for each frame. Then, we examine the



(a)

(b)

Fig. 11 Correspondence between the amount of MFPs and **a** rate of BFPs, **b** mean duration of TFPs in frames with same crowd size but with different occlusion levels

expected relationships. Figure 11 plots the relationships between occlusion-related features and the amount of MFPs in frames with the same crowd size of 23 people but with different occlusion levels. It can be seen that the rate of BFPs increases as the amount of MFPs reduces. This linear relationship between these statistics confirms that the rate of BFPs can be an extremely useful clue to the level of occlusion in a scene. Also, we observe that the mean duration of TFPs is highly correlated with the amount of MFPs in frames with the same crowd size and various occlusion levels. This relationship indicates that this feature is also very helpful for discovering the various levels of occlusion in the scene.

3.4 Experiment 2: Crowd counting results and comparisons with other methods

The proposed system is trained and tested using training and testing sets of each video reported in Table 1. Two measures, Mean-Squared-Error (MSE) and the Mean-Absolute-Error (MAE), are used to analyze the accuracy of counting quantitatively:

$$MSE = \frac{1}{N} \cdot \sum_{t=1}^{N} \left(G^t - E^t\right)^2, \tag{10}$$

$$MAE = \frac{1}{N} \cdot \sum_{t=1}^{N} |G^t - E^t|, \tag{11}$$

where G^t and E^t are the ground truth and estimated count for frame t, and N is the number of testing frames. Since the results of the neural networks differ slightly from test to test (because initial values for neural network are selected, randomly), the system is retested five times for each video and the average results are recorded. The calculated average value is rounded to the nearest integer to produce the final crowd count. In the following subsections, the obtained results on different videos are presented.

3.4.1 Results on "Peds1" video

Since the majority of recent work [6, 23, 37] have performed experiments on "Peds1" video and quantitative results have been provided in the related papers, firstly, we report the counting results obtained by our approach on this data and compare them with results of other methods. Table 3 shows the MSE and MAE

Table 3 Comparison of different approaches on "Peds1" video

	MSE	MAE
Ours	**2.357**	**1.093**
Ryan et al. [37]	3.850	1.558
Chan et al. [6]	4.181	1.621
Lempitsky et al. [23]	N/A	1.70
Chan et al. [6] using features of [19]	5.438	1.808
Chan et al. [6] using features of [9]	6.953	1.995
Kong et al. [18]	6.492	2.043
Ours using only the amount of MFPs	7.022	2.167
Albiol et al. [1]	9.351	2.821

rates of different approaches on "Peds1" video. The reported results of methods in [6, 23, 37] are quoted directly from the related papers. Since the approach in [6] estimates pedestrian counts in either direction and does not provide a total count, we report the error rates of this approach on counting pedestrians walking away from the camera, which contains the majority of the crowd in "Peds1" video. The performance of approach in [18] has been measured by authors in [37] with their own implementation. For the sake of comparison, the error rates of approach in Albiol et al. [1], which we have provided our own implementation, are also reported in Table 3.

From the results in Table 3, it is evident that the proposed method outperforms the other approaches with respect to both MSE and MAE performance indices. Our approach, even by using only the amount of MFPs, performs better than the method proposed by Albiol et al. [1]. This is due to the effects of perspective and also simple proportionality relation assumed between crowd size and the number of moving corner points in [1]. The result of our approach, utilizing three features from FPs, is better than approaches in [6] and [37], while approach in [6] uses a larger quantity of complicated features (29 features) and approach in [37] utilizes similar features used in [6] on a local rather than holistic level. The performance of our system is also better than approach in [23] that uses foreground and gradient information. Most of the utilized features in approaches [6, 9, 18, 19, 37] are based on the edge features. Edge-based features can be extremely erroneous, as the edges are completely messy when the background is complicated and the textures of human clothes are not smooth. In contrast, our detected BFPs are related to the overlapping boundaries (edges) between people not to the all of the edges present in the scene.

3.4.2 Results on "PETS2009" dataset

Before discussing the results on "PETS2009" dataset, it is important to indicate the frame rate instability problem that we faced in our experiments on this data. During the LFP detection process on "PETS2009" dataset, we found that almost all of the MFPs are detected as LFPs (i.e. FPs with large variations in speed). We observed that these large speed variations of all MFPs are caused by changes in the frame rate (acquisition problem); a problem which had been further confirmed by the PETS2009 committee [1]. The PETS2009 metadata states that the frame rate is approximately 7 fps, but we found that it is not constant along the frame sequences and the time-interval between two successive frames changes at some moments. Therefore, the displacement vectors of MFPs change depending on that time interval. However, in order to make this dataset usable for our experiments, we track the FPs by block-matching technique and use a time filter in LFP detection process in the same manner as [1]. Hence, in our LFP detection process on "PETS2009" dataset, we do not take into account the speed of FPs in frames where the average speed of FPs is very large (i.e. there is a large time-interval between two consecutive frames). For more details about the time filtering technique used on this data, see [1].

For our experimentations, we use View1 from the S1.L1.13-57, S1.L1.13-59, S1.L2.14-06, and S1.L3.14-17 videos of this dataset that were used in the people counting contest held in PETS2009 [33]. An example frame of a video is shown in Fig. 12, along with the three defined regions of interest (R0, R1, and R2). For each test videos and regions, we train our system by using the training sets listed in Table 4, and test it on the remaining frames of each video. The selected testing regions are

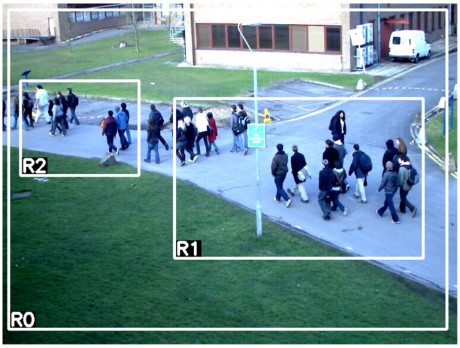


Fig. 12 An example of View1 from "PETS2009" dataset, along with regions of interest

as same as those in Albiol et al. [1] that were reported in [12]. Table 5 shows the obtained results and compares them with results of approach in Albiol et al. [1] which are quoted directly from [12]. From Table 5 it is clear that the proposed approach again outperforms Albiol's method with respect to the counting error rate on most of the testing videos. The poorer performance of our approach on S1.L3.14-17 video is due to the very small amount of training data existed for this video.

3.4.3 Results on "Peds2", "USC", "Bridge", and "Gate" videos

Table 6 summarizes the performance of our system on four testing videos together with results of approach in [1], obtained by our own implementation, on the same videos. As shown in the table, the counting results of our approach on different videos are promising. Again, its performances, even by using only the amount of MFPs, are better than the results obtained by Albiol's method [1]. Figure 13 compares the counting results of our approach and the ground truth on four testing videos. As shown in this figure, the crowd estimations by our system track the ground truth well in most of the testing frames. Figure 14 shows some result frames produced by our system on six videos.

We also compare the performance of our approach against a model-based human detection approach in [43]. Although, existing model-based approaches are not able to deal with high level of occlusions in crowds, they have shown good performances in the situations where crowds are small such as "USC" video. We calculate the counting error rate of approach in [43] on "USC" video by using reported scores (i.e. correct detections, false alarms, and valid humans count) in the corresponding paper. This rate is become equal to 7 %. Also, we measure this rate from counting results obtained by our approach on this video, considering total ground truth count of people and the total estimation for all testing frames. For our approach, this rate

Table 4 Test videos and training sets in "PETS2009" dataset

	S1.L1.13-57	S1.L1.13-59	S1.L2.14-06	S1.L3.14-17
Length	221 frame	241 frame	201 frame	91 frame
Testing region	R0	R0	R1	R1
Number of training frames	44	48	40	22
Training frames	1:5:221	1:5:241	1:5:201	1:4:91

Table 5 MAE rate of each approach on different test videos in “PETS2009” dataset

	S1.L1.13-57.R0	S1.L1.13-59.R0	S1.L2.14-06-R1	S1.L3.14-17.R1
Ours	1.031	1.146	1.687	1.463
Albiol et al. [1]	1.4	1.8	1.9	1.4

is equal to 5 %. This comparable result of the proposed approach implies that the performance of the system does not drop in sparse scenes.

3.5 Experiment 3: Effects of occlusion-related features

To show the influences and advantages of extracted occlusion-related features, Fig. 15 compares the counting error rates of the system using different feature sets on five testing videos. As shown in this figure, by using the occlusion-related features, considerable improvements in counting are achieved on all testing videos. Our approach, using only the amount of MFPs performs the worst, and performance improves steadily as the other features are added. This shows the informativeness of the extracted features: the amount of MFPs provides a coarse linear estimate, which is refined by the rate of BFPs and mean duration of TFPs accounting for various non-linearities.

3.6 Experiment 4: Effect of perspective normalization

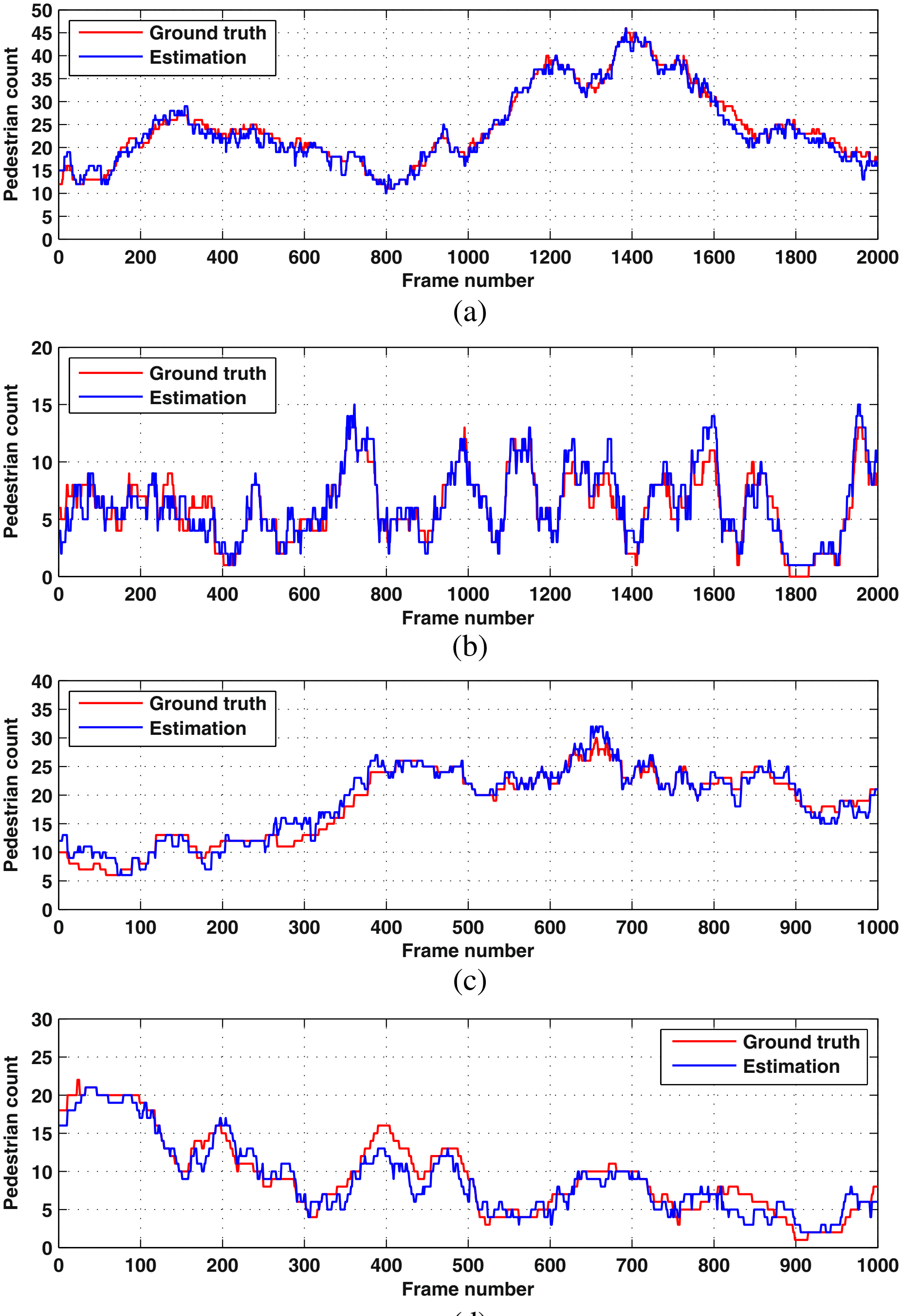
To cope with the effects of perspective, we propose to normalize the features, using a perspective map. To examine the effectiveness of this normalization on the final results, we evaluate our approach with and without perspective normalization. Figure 16 compares the error rates of our approach in both cases on five testing videos. This figure obviously shows the effectiveness of the perspective normalization on all videos. This improvement is especially remarkable on “Peds1” video as it contains a wide camera-view of the scene and due to very low camera tilt angle, the size of people varies greatly in different locations of the scene.

3.7 Experiment 5: Scaling the approach on long-range videos

In order to examine the robustness of the proposed approach against the wide variability in crowd density and distribution, we run our system on long-range videos (i.e. on one hour of “Peds1”, one hour of “Peds2”, 10 minutes of “Bridge”, and 5 minutes of “Gate” dataset). It is obvious that these long testing videos contain much various crowd distributions, compared with the small amount of frames used to train and test the system. In this experiment, same training frames (reported in Table 1)

Table 6 Results on “Peds2”, “USC”, “Bridge”, and “Gate” videos

	Peds2		USC		Bridge		Gate	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Ours	**1.647**	**0.893**	**2.118**	**1.034**	**2.694**	**1.168**	**2.766**	**1.289**
Ours; only amount of MFPs	5.842	2.028	5.559	2.271	8.026	2.107	7.437	2.119
Albiol et al. [1]	8.714	2.521	7.036	2.451	9.148	2.754	8.891	2.415



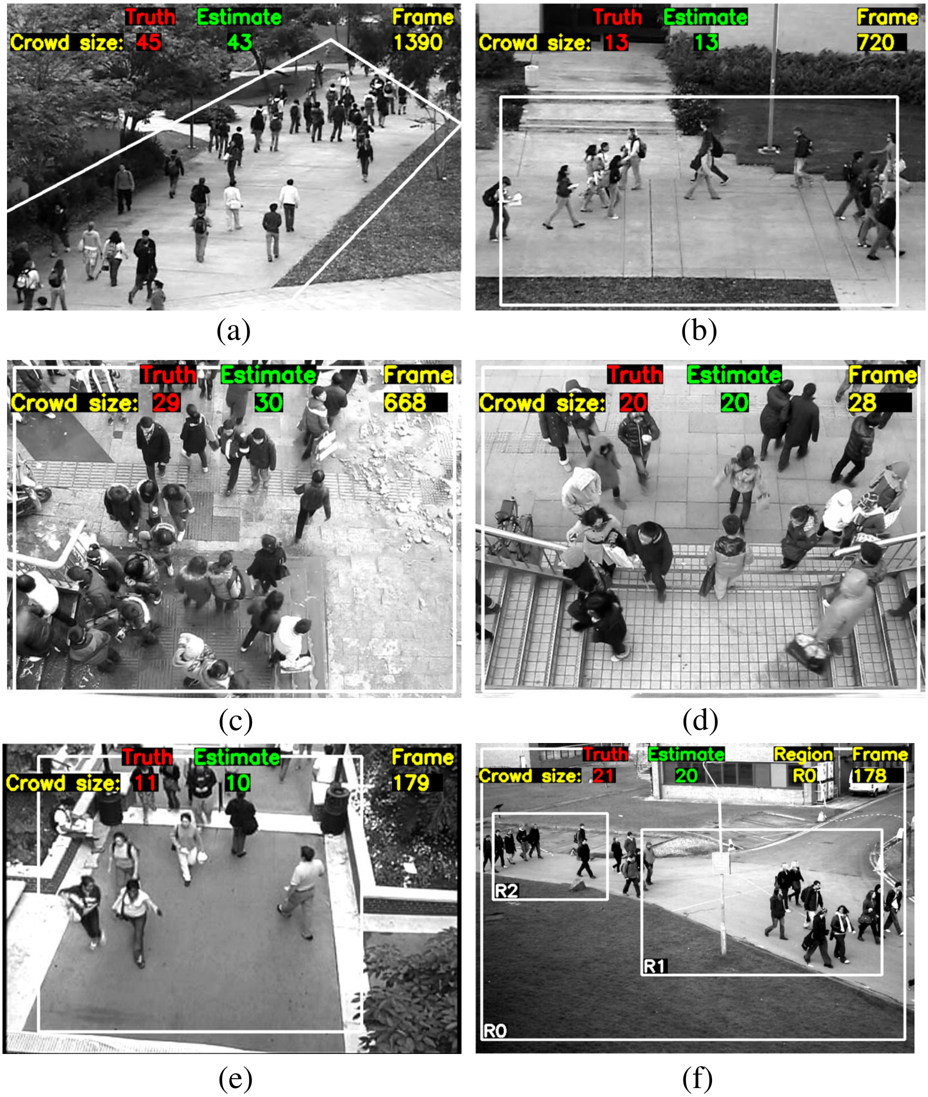
(a)

(b)

(c)

(d)

Fig. 13 Counting results of our method and the ground truth on **a** “Peds1”, **b** “Peds2”, **c** “Bridge”, and **d** “Gate” videos



(a) (b)

(c) (d)

(e) (f)

Fig. 14 Examples of crowd counting on **a** "Peds1", **b** "Peds2", **c** "Bridge", **d** "Gate", **e** "USC", and **f** "PETS2009" datasets

are used for each video, and testing is performed on full video of each dataset. The counting results are evaluated manually, using 100 frames of each video, chosen by a random number generator. Figure 17 compares the MSE and MAE rates of these tests with corresponding rates for each video in Table 3 and Table 6. As shown in the figure, the counting results on all long-range videos are comparable with the obtained results for the finite number of testing frames of each video. It is evident that the proposed method, using local features from scene, is able to capture wide variations in crowd distribution from a small amount of training data.

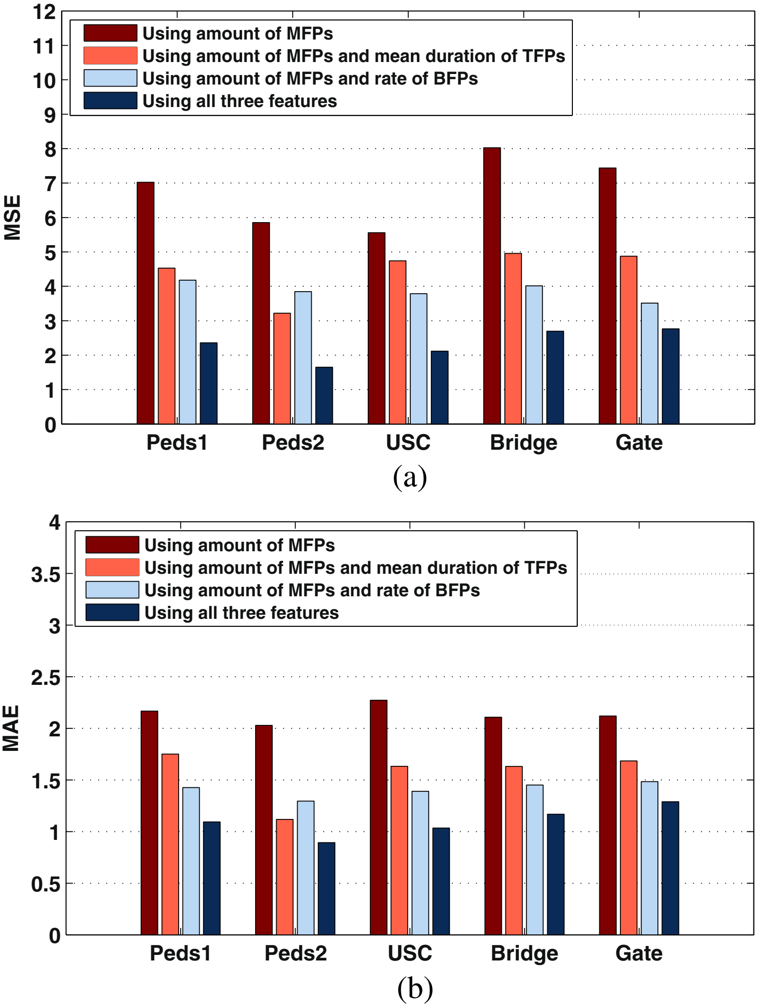


Fig. 15 Comparisons of counting error rates of the system using different feature sets on five testing videos. **a** MSE, and **b** MAE

Another important advantage of our approach, which can be deduced from this experiment, is the robustness of extracted features against small environmental changes (e.g. illumination, shadows) over pretty long time-periods. In fact, feature-points are not very sensitive to small environmental changes, while the methods based on foreground segmentation are very sensitive to these variations.

3.8 Experiment 6: Evaluating the approach on highly dense crowds

In order to assess the performance of our system on very dense crowds, we run it on "Loveparade2010" video. The main goal of this experiment is to examine whether the proposed approach is able to extract features from FPs tracked in a highly dense

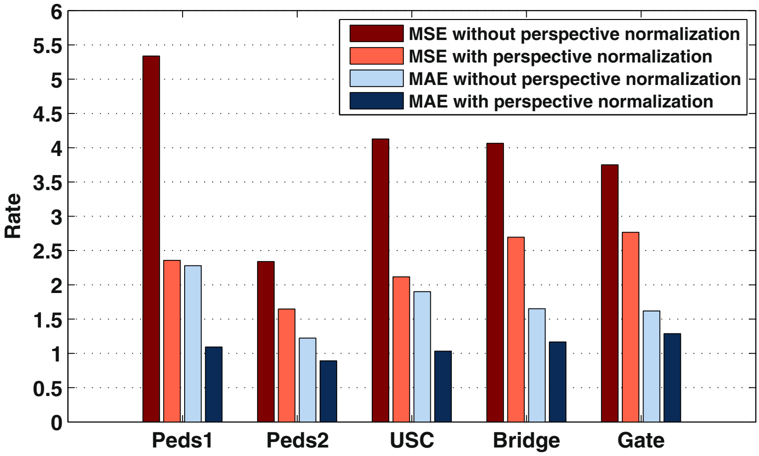


Fig. 16 Counting results produced by our system with and without perspective normalization on five testing videos

crowd or not. In “Loveparade2010” video, dense crowds of visitors are passing in a tunnel to enter or exit from Loveparade festival area. Video recordings are available for the time between 13:30 h and 16:40 h. In this video, the number of people is increasing with time. For example, between 13:30 h and 15:00 h, crowds of people are moving in the scene with normal walking speed and occasionally some gaps occur between crowds, while 30 minutes later, the crowd density is almost doubled and people are moving very close together. Finally, an extremely dense crowd is formed around 16:20 h where a large number of people are present in the scene that can hardly move. For more details about this video, see [21].

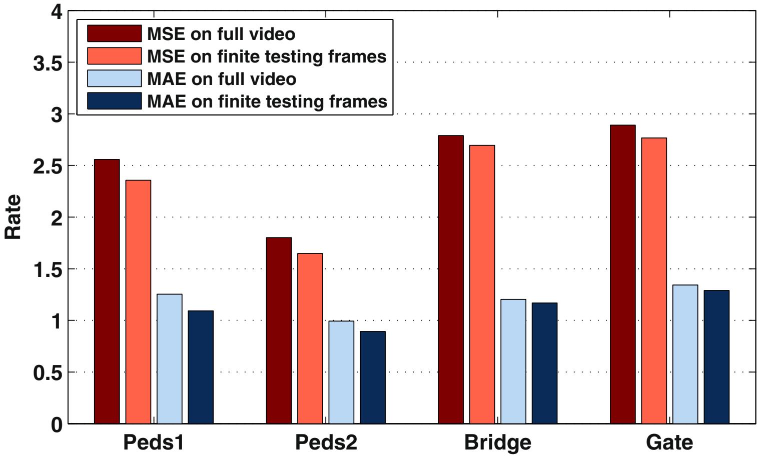


Fig. 17 Counting results produced by our system on finite number of testing frames and long-range full video of four datasets

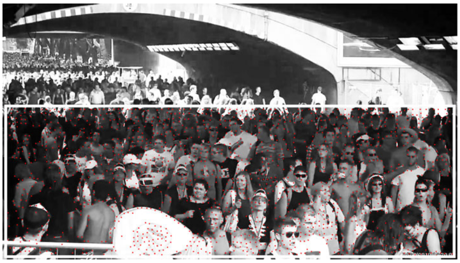
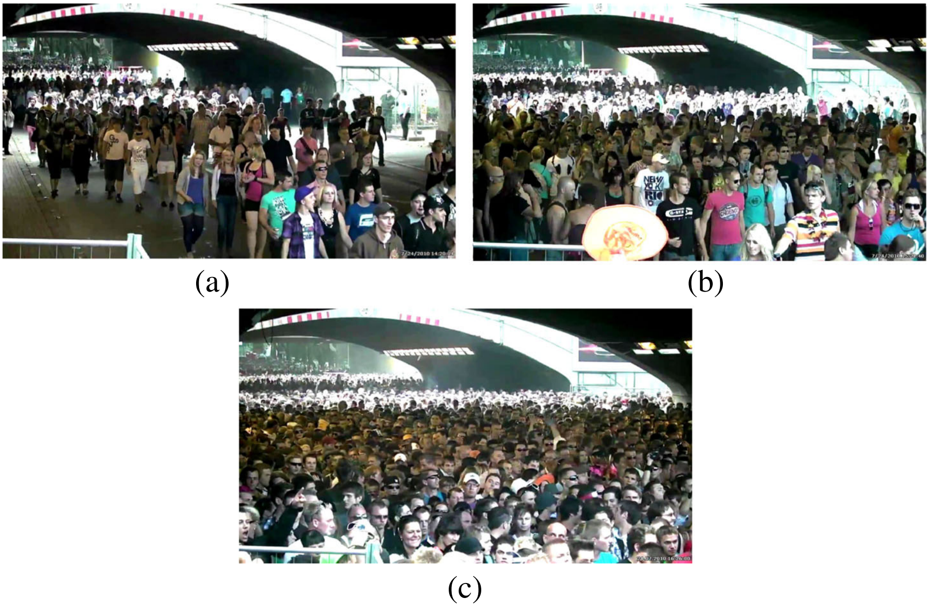


Fig. 18 A sample frame from "Loveparade2010" video including selected region of interest and detected FPs within that

In order to evaluate our approach on different levels of crowd in this video, we provide ground truth count for 3000 frames from three separate parts of the video. A region of interest is selected on the scene, as shown in Fig. 18, and ground truth count is annotated every 5 seconds manually. Assuming that the number of people does not change substantially in consecutive frames, crowd counts in the remaining frames are estimated with linear interpolations. Each annotated part contains 1000 frames (100 seconds) of the video as follows:

- "Part1": Between 14:20:00 h and 14:21:40 h, consisting of 60 to 80 people inside the selected region of interest.
- "Part2": Between 15:20:00 h and 15:21:40 h, featuring denser crowds, compared with the "Part1", and contains crowds of size 85 to 120 people inside the region of interest. In this part, people are moving very close together.



(a)

(b)

(c)

Fig. 19 Example frames from annotated parts in "Loveparade2010" video. **a** "Part1", **b** "Part2", and **c** "Part3"

– “Part3”: Between 16:25:00 h and 16:26:40 h, featuring a huge crowd of visitors about 250 people congested in the scene and stepping from one foot to the other in order to keep their balance.

Example frame of each part is shown in Fig. 19. Due to the very large number of people in “Part3”, the ground truth counts for this part are provided based on Jacobs method. This method involves dividing the area occupied by a crowd into sections, determining an average number of people in each section, and multiplying by the number of sections occupied.

The counting system is trained using frames 251–750 (500 frames) from each part, and testing is performed on the remaining frames (frames 1–250 and 751–1000) of related part. In addition, the performance of the system is also evaluated on all three parts together, using training and testing frames from all parts (1500 frames for training and 1500 frames for testing purpose). In order to analyze the effectiveness of each statistical feature, different subsets of features are used in these experiments. Table 7 shows the error rates for different sets of data, under different feature representations.

As shown in Table 7, the performance of the system on “Part1” data, where people are moving fluidly and torsos of them are not entirely occluded, is comparable with results obtained for other testing videos (see Subsection 3.4). It is obvious that the amount of MFPs has provided a rough estimate of crowd size which has been improved steadily by adding other features. The larger error values obtained here are not unexpected because the densities of crowds in the videos used in our previous experiments are not as high as this video. However, as can be seen in Table 7, the performance of the system is dropped remarkably when it is run on “Part2” and on three parts together (“Part1”, “Part2”, and “Part3”). Since the number of people in other parts is much larger than “Part1”, larger error values obtained by using only amount of MFPs are reasonable, but the important issue is that the occlusion-related features (i.e. rate of BFPs and mean duration of TFPs) have not significantly decreased these error rates. The MAE of 15.741 obtained by amount of MFPs on “Part2” data which contains crowds of size 85 to 120 people, is a promising result. However, adding occlusion-related features only reduces this rate about one people, which is achieved by adding mean duration of TFPs, and the rate of BFPs has not any contribution. This shortcoming of occlusion-related features in highly dense crowds is due to the impossibility of generating meaningful and sufficient motion trajectories for FPs in such scenes. Our analysis on trajectories obtained by KLT tracker on “Part2” and “Part3” show that due to the heavy occlusions present in crowds, most of the FPs are lost quickly during the tracking. Therefore, majority of the motion trajectories become very short and do not contain enough information to be compared with other trajectories. This problem results in weakness of occlusion-related features which are calculated based on only motion trajectories of FPs.

Table 7 MAE rates on four sets of data, using different combinations of features

	Part1	Part2	Part3	Three parts
Amount of MFPs	8.618	15.741	7.634	28.337
Amount of MFPs and rate of BFPs	6.324	15.426	7.126	27.663
Amount of MFPs and mean duration of TFPs	5.671	**14.632**	6.873	21.754
All three features	**4.416**	15.329	**6.711**	**21.207**

However, as shown in Table 7, an improvement is obtained by utilizing mean duration of TFPs for counting crowds on all three parts together. This improvement is due to the very different levels of crowds present in different parts, i.e. the mean duration of TFPs calculated for frames in “Part1” are much larger than for frames in “Part2” and “Part3”. Therefore, it is helpful to capture the occlusion level in different parts. Also, it should be noted that, good estimation of crowd size obtained on “Part3” is due to this fact that all training and testing frames of this part contain an almost equal number of people congested in the scene.

Although the feature extraction based on motion information of FPs might be not accurate enough in highly dense crowds, the proposed approach is capable to provide a rough estimate of crowd size in such scenes. Even in highly crowded scenes, MFPs can be separated from static FPs due to small motions of people in the scene (e.g. stepping from one foot to the other in order to keep their balance). So we can calculate amount of MFPs as a clue to the foreground area and crowd size, while methods based on foreground/background segmentation are absolutely prone to errors in this kind of scenes.

3.9 Experiment 7: Examining the robustness and generalizability of the local features

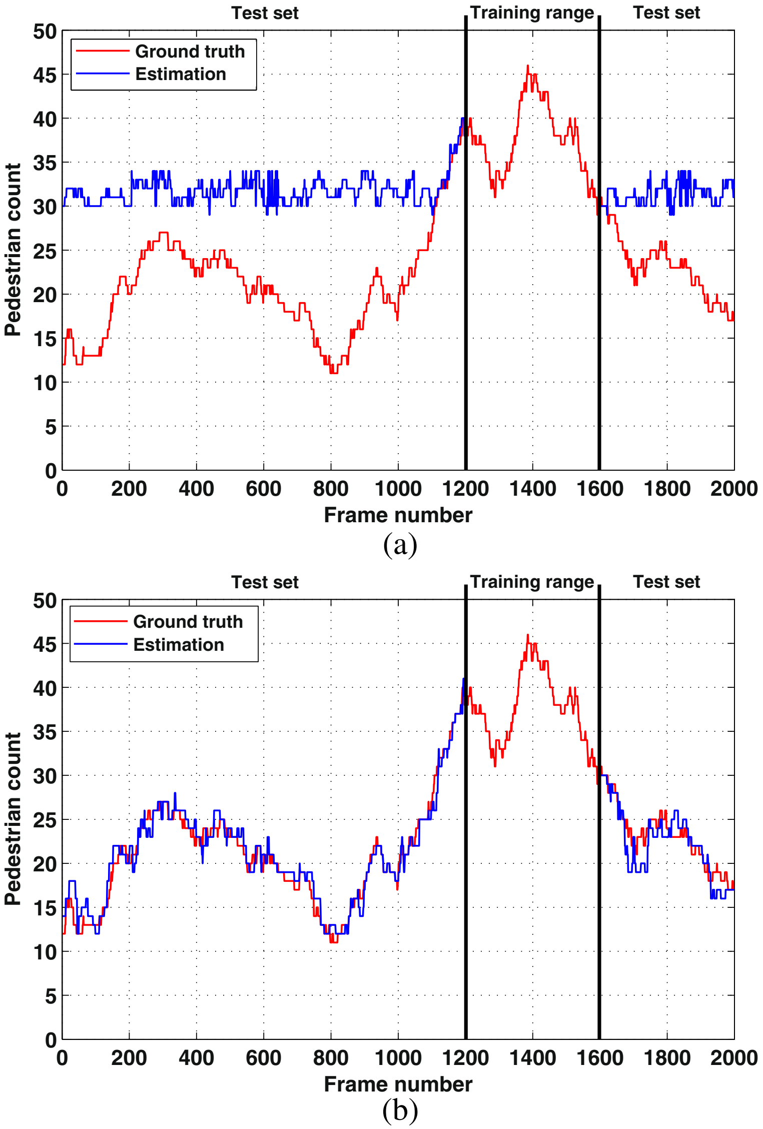
To gain an understanding of the advantages of local features, the performance of the proposed system is compared against equivalent holistic system. The holistic system uses the same features as the proposed system, taken on a holistic level. In this system, MFPs are not clustered into clusters, i.e. statistical features are calculated from all MFPs in the scene. Ground truth is also provided on a holistic level. To compare the accuracy of two systems, the holistic system is trained and tested on “Peds1” video by same training and testing sets used for evaluate the proposed system (reported in Table 1). Table 8 presents counting error rates of two systems on “Peds1” video. In addition, cumulative error rates (CEs) are reported in order to compare the uncertainty rates of two methods in estimating the crowd size. The CE(x) is defined as the percentage of frames for which the counting error is less than or equal to x number of people. For example, the count is within 3 people of the ground truth 98 % of the time for the proposed system. As shown in Table 8, by all measures of accuracy and uncertainty, the proposed system outperforms the equivalent holistic system. The good performance of the proposed system is due to the availability of more training cases existed in one training frame, while in holistic system each frame contains only one training case.

To compare the generalizability of two systems, we reduce the training set of “Peds1” video from 160 frames to 80 frames. The training set is formed by taking the last frame out of every five consecutive frames within frames 1201–1600, which contain crowds of size 30–46. These frames include a mixture of small and large groups of people. The testing is done on frames 1–1200 and 1601–2000, featuring crowds of size 11–40. The counting results provided by two systems are shown in

Table 8 Comparison of two methods on “Peds1” video

	MSE	MAE	CE(1)	CE(2)	CE(3)
Proposed system	**2.357**	**1.093**	**65 %**	**87 %**	**98 %**
Holistic system	2.612	1.259	61 %	87 %	97 %

Fig. 20. As this figure demonstrates, the holistic system fails and it is unable to provide any meaningful results for crowd volumes not seen in the training set. In contrast, the proposed system, trained on clusters of various sizes, is able to count smaller crowds, too. This problem may happen for any other approaches that utilize holistic features from scene unless they use a large amount of training data to capture the wide variations in crowd distribution. However, it is not practical to provide hundreds of frames of ground truth to setup a counting system in a real world application, especially in facilities that contain numerous cameras.



(a)

(b)

Fig. 20 Crowd counting result. **a** By holistic system. **b** By proposed system

Table 9 Performance of the neural network using different numbers of hidden neurons

	Number of hidden neurons			
	10	30	50	70
Training time	6.6 s	6.4 s	8.2 s	9.4 s
MSE	2.764	2.873	2.357	2.419

Although counting people on a local level increases the generalizability of system very well, in a real-life application where the statistics may changes due to the temporal changes in environment (e.g. illumination, shadows) and crowd density, a simple trained classifier is not always adequate to achieve a good performance. Instead, it would be desired to have a mechanism, which would provide the system with the capability to automatically test its performance and be automatically retrained when its performance is not acceptable. In our case, we can utilize a retrainable neural network structure proposed in [11].

3.10 Implementation

We implemented our algorithm in C++ for the KLT and feature extraction parts, and Matlab for the neural network side. The neural network fitting tool in Matlab is used in our experiments. We use a single hidden layer feed-forward neural network with sigmoid hidden neurons, and train the network using Levenberg-Marquardt backpropagation algorithm. In order to determine the number of neurons in hidden layer, different numbers are tried in our experiments. Table 9 compares the performance of the network using different numbers of hidden neurons on the same training and testing data. All the results are an average of five trials. As can be seen, the differences between the training times are not considerable. However, the number of hidden neurons is chosen to be 50 which results in a more accurate performance. Since the training procedure is an offline procedure, the computational cost of neural network would not be a big concern for the proposed method. However, due to small structure of the neural network in our application, the training phase is done very fast.

In our experiments, the average execution time to estimate the crowd count in one frame across all the videos is about 0.35 sec (i.e., ≈ 3 fps). Currently, our method performs the counting process for all frames in the sequence. In order to make the approach suited for counting people in an online video, the given time-window (see Subsection 2.1) can be shifted by more than one frame, assuming that the number of people does not change substantially in consecutive frames.

4 Conclusions and future work

This paper proposed a crowd counting approach based on only motion information of low-level feature-points. The proposed approach detected some feature-points in the scene and tracked them along the time. Moving-feature-points were detected and partitioned into clusters, corresponding to separate groups of moving people, in order to extract some local features for each cluster. Then, feature-points of each cluster were carefully classified into categories of: limb-feature-points, torso-feature-points, and boundary-feature-points. Three statistical features were extracted for each cluster: first feature, the amount of moving-feature-points, was used as a clue to the foreground area, and the other two features, namely the rate of boundary-feature-points and the mean duration of torso-feature-points, were extracted to capture the various levels of occlusion present in the scene. To cope with the effects of perspective distortion, a perspective map was used to weight the FPs. A neural network was trained using extracted features to estimate the number of people in crowds. Promising counting results were obtained on different video sequences. Comparisons with other methods showed that the proposed approach has a superior performance. The results of our extensive experiments demonstrated that the extracted features are highly informative: the amount of moving-feature-points provides a coarse linear estimate of crowd size, which is refined by the rate of boundary-feature-points, and mean duration of torso-feature-points accounting for various non-linearities caused by occlusions. With the method proposed in this paper, we might not expect a fully accurate estimate of crowd size in highly crowded scenes such as music festivals, sports events or pilgrimage, as it is not possible to generate meaningful motion trajectories of feature-points in such situations. However, our experiments on highly crowded videos showed that the proposed approach is able to provide a rough estimate of crowd size in such scenes.

The limitations of our motion-only method are not unexpected. Counting errors can occur when non-human objects such as cars, bicycles, etc. appear in the scene, which result in an overestimation of the crowd size. Also, the system proposed here is only able to estimate the number of moving people and is not able to consider stationary people in the estimation. These cases might be mitigated by using some robust object detector techniques. However, these flaws exist in other map-based methods including work that were compared with the proposed approach, as they mainly use motion information of people to segment the foreground area. Finally, similar to other map based approaches, the crowd size is estimated independently for each frame, meaning that the proposed system either cannot count the total number of people passed through a field-of-view.

As a future work, we plan to use further motion information from feature-points to improve the proposed counting system. We are interested in exploiting motion direction of trajectories to segment the crowds into sub-parts moving in different directions. This will enable us to provide a count for the number of people moving in each direction. Also, other motion features of trajectories like velocity, speed, etc. can be used in order to distinguish different kinds of object which improve the system to provide class-specific counts (e.g. cars vs. pedestrians). Another interesting extension of this approach is to improve the system to work with moving cameras which would be truly beneficial in densely crowded scenes. Finally, the proposed approach will also be tested on low-resolution videos, and the use of other type of corner points will be investigated in such videos.

Acknowledgements This work was partly supported by the 973 Program (2013CB329504), NSF of China (No. 61070067), and Qianjiang Talent Program of Zhejiang (2011R10078).

References

1. Albiol A, Silla MJ, Albiol A, Mossi JM (2009) Video analysis using corner motion statistics. In: Proc. of the IEEE Int. workshop on performance evaluation of tracking and surveillance (PETS), pp 31–38

2. Benfold B, Reid I (2011) Stable multi-target tracking in real-time surveillance video. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR11), pp 3457–3464
3. Bishop CM. (1995) Neural networks for pattern recognition. New York: Oxford University Press
4. Brostow GJ, Cipolla R (2006) Unsupervised bayesian detection of independent motion in crowds. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR06), pp 594–601
5. Celik H, Hanjalic A, Hendriks E (2006) Towards a robust solution to people counting. In: Proc. of the IEEE int. conf. on image processing, pp 2401–2404
6. Chan AB, Liang Z, Vasconcelos N (2008) Privacy preserving crowd monitoring: counting people without people models or tracking. In: Proc. of Int. conf. on computer vision and pattern recognition (CVPR08)
7. Cheriyadat AM, Bhaduri BL, Radke RJ (2008) Detecting multiple moving objects in crowded environments with coherent motion regions. In: Proc. of sixth IEEE Workshop on Perceptual Organization in Computer Vision (POCV08), in conjunction with IEEE CVPR08
8. Cho SY, Chow TWS, Leung CT (1999) A neural-based crowd estimation by hybrid global learning algorithm. IEEE Trans Syst Man Cybern B 29(4):535–541
9. Davies AC, Yin JH, Velastin SA (1995) Crowd monitoring using image processing. Electron Comm Eng J 7:37–47
10. Doulamis N (2009) Evacuation planning through cognitive crowd tracking systems. In: Proc. of the 16th int. conf. on signals and image processing, pp 1–4
11. Doulamis A, Doulamis N, Kollias S (2000) On line retrainable neural networks: improving the performance of neural networks in image analysis problems. IEEE Trans Neural Netw 11(1):137–155
12. Ellis A, Shahrokni A, Ferryman J (2009) PETS 2009 and Winter-PETS 2009 Results: A Combined Evaluation. In: Proc. of 12th IEEE int. workshop on performance evaluation of tracking and surveillance (PETS)
13. Haibo W, Hong F (2010) The research of emergency evacuation model based on digital city management platform. In: Proc. of Int. Conf. on Multimedia Technology, pp 1–4
14. Haritaoglu I, Harwood D, Davis LS (1999) Hydra: multiple people detection and tracking using silhouettes. In: Proc. of second IEEE workshop on visual surveillance, pp 280–285
15. Harris C, Stephens M (1988) A combined corner and edge detector. In: Proc. of the 4th Alvey vision conference, pp 147–151
16. Hou YL, Pang GKH (2011) People counting and human detection in a challenging situation. IEEE Trans Syst Man Cybern A, 41(1):24–33
17. Kilambi P, Ribnick E, Joshi AJ, Masoud O, Papanikolopoulos N (2008) Estimating pedestrian counts in groups. Comput Vis Image Underst 110(1):43–59
18. Kong D, Gray D, Tao H (2006) A viewpoint invariant approach for crowd counting. In: Proc. of the 18th int. conf. on pattern recognition, vol 3, pp 1187–1190
19. Kong D, Gray D, Tao H (2005) Counting pedestrians in crowds using viewpoint invariant training. In: Proc. of British Machine Vision Conf
20. Krahnstoever N, Mendona PRS (2005) Bayesian autocalibration for surveillance. In: Proc. of int. conf. on computer vision (ICCV05), vol 2, pp 1858–1865
21. Krausz B, Bauckhage C (2011) Loveparade 2010: automatic video analysis of a crowd disaster. Comput Vis Image Underst 116(3):307–319
22. Leibe B, Seemann E, Schiele B (2005) Pedestrian detection in crowded scenes. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR05), vol 1, pp 878–885
23. Lempitsky V, Zisserman A (2010) Learning to count objects in images. In: Advances in neural information processing systems (NIPS), pp 1324–1332
24. Lim J, Kim W (2012) Detecting and tracking of multiple pedestrians using motion, color information and the AdaBoost algorithm. Multimedia Tools Appl J. doi:10.1007/s11042-012-1156-3. Springer
25. Lin Z, Davis L (2010) Shape-based human detection and segmentation via hierarchical part-template matching. IEEE Trans Pattern Anal Mach Intel 32(4):604–618
26. Lin Z, Liu L, Yan Z, Li Z (2011) Multi-agent modeling of city emergency evacuation. In: Proc. of int. conf. on multimedia technology, pp 3570–3574
27. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. Proc. of 7th int. joint conf. on artificial intelligence (IJCAI81), pp 674–679
28. Ma H, Zeng C, Ling CX (2012) A reliable people counting system via multiple cameras. ACM Trans Intel Syst Technol 3(2):1–22
29. Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR10), pp 1975–1981

30. Marana AN, da Fontoura Costa L, Lotufo RA, Velastin SA (1999) Estimating crowd density with Minkoski fractal dimension. In: Proc. of int. conf. acoust, speech, signal processing, pp 3521–3524
31. Marana AN, Velastin SA, Costa LF, Lotufo RA (1997) Estimation of crowd density using image processing. In: IEE colloquium on image processing for security applications, vol 11, pp 1–8
32. Paragios N, Ramesh V (2001) A mrf-based approach for real-time subway monitoring. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR01), vol 1, pp 1034–1040
33. PETS: performance evaluation of tracking and surveillance workshop at CVPR 2009. Miami, Florida (2009) http://www.cvg.rdg.ac.uk/PETS2009/
34. Rabaud V, Belongie SJ (2006) Counting crowded moving objects. In: Proc. of Int. conf. on computer vision and pattern recognition (CVPR06), pp 705–711
35. Rahmalan H, Nixon MS, Carter JN (2006) On crowd density estimation for surveillance. In: The institution of engineering and technology conference on crime and security, pp 540–545
36. Rittscher J, Tu PH, Krahnstoever N (2005) Simultaneous estimation of segmentation and shape. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR05), vol 2, pp 486–493
37. Ryan D, Denman S, Fookes C, Sridharan S (2009) Crowd counting using multiple local features. In: Proc. of conf. on digital image computing: techniques and applications, pp 81–88
38. Sugimura D, Kitani K, Okabe T, Sato Y, Sugimoto A (2009) Using individuality to track individuals: clustering individual trajectories in crowds using local appearance and frequency trait. In: Proc. of Int. conf. on computer vision (ICCV09), pp 1467–1474
39. Tomasi C, Kanade T (1991) Detection and tracking of point features. Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132
40. Tomasi C, Shi J (1994) G ood features to track. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR94), pp 593–600
41. Wu B, Nevatia R (2007) Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. Int J Comput Vis 75:247–266
42. Zeng C, Ma H, Ming A (2010) Fast human detection using mi-SVM and a cascade of HOG-LBP features. In: Proc. of IEEE int. conf. on image processing, pp 3845–3848
43. Zhao T, Nevatia R (2003) Bayesian human segmentation in crowded situations. In: Proc. of int. conf. on computer vision and pattern recognition (CVPR03), pp 459–466

Mahdi Hashemzadeh received the B.S. and M.S. degrees in software engineering from Azad University, Iran, in 2002 and 2006, respectively. He is currently working toward the Ph.D. degree at the CCNT Biometrics Lab in the college of Computer Science and Technology, Zhejiang University, China. His research interests include visual surveillance, computer vision and machine learning.

Gang Pan received the B.Sc. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 1998 and 2004, respectively. He is currently a professor with the Department of Computer Science, Zhejiang University. He has published more than 100 refereed papers. He visited the University of California, Los Angeles (UCLA), during 2007–2008. His research interests include pervasive computing, computer vision, and pattern recognition. Dr. Pan has served as a Program Committee Member for more than ten prestigious international conferences, such as IEEE ICCV and IEEE CVPR, and as a reviewer for various leading journals, such as IEEE T-PAMI and IEEE T-IP.

Min Yao received the B.E. degree in radio technique from Hefei University, China, in 1982, the M.E. degree in computer science from Hefei University of Technology, China, in 1986, and the Ph.D. degree in biomedical engineering and instrumentation from Zhejiang University, China, in 1995. He is currently a professor in the College of Computer Science and Technology at Zhejiang University, Hangzhou, China. His research interests include pattern recognition, computational intelligence and data mining.