University Library



A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Rao, AS;Gubbi, J;Marusic, S;Palaniswami, M

Title:
Estimation of crowd density by clustering motion cues

Date:
2015-11-01

Citation:
Rao, A. S., Gubbi, J., Marusic, S. & Palaniswami, M. (2015). Estimation of crowd density by clustering motion cues. Visual Computer, 31 (11), pp.1533-1552. https://doi.org/10.1007/s00371-014-1032-4.

Persistent Link:
https://hdl.handle.net/11343/283076

**The Visual Computer manuscript No.**
(will be inserted by the editor)

# Estimation of Crowd Density by Clustering Motion Cues

**Aravinda S. Rao · Jayavardhana Gubbi · Slaven Marusic · Marimuthu Palaniswami**

Received: date / Accepted: date

**Abstract** Understanding crowd behavior using automated video analytics is a relevant research problem in recent times due to complex challenges in monitoring large gatherings. From an automated video surveillance perspective, estimation of crowd density in particular regions of the video scene is an indispensable tool in understanding crowd behavior. Crowd density estimation provides the measure of number of people in a given region at a specified time. While most of the existing computer vision methods use supervised training to arrive at density estimates, we propose an approach to estimate crowd density using motion cues and hierarchical clustering. The proposed method incorporates optical flow for motion estimation, contour analysis for crowd silhouette detection, and clustering to derive the crowd density. The proposed approach has been tested on a dataset collected at the Melbourne Cricket Ground (MCG) and two publicly available crowd datasets—Performance Evaluation of Tracking and Surveillance (PETS) 2009 and University of California, San Diego (UCSD) Pedestrian Traffic Database—with different crowd densities (medium to high density crowds) and in varied environmental conditions (in the presence of partial occlusions).

Aravinda S. Rao · Jayavardhana Gubbi · Slaven Marusic · Marimuthu Palaniswami
ISSNIP, Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, VIC - 3010, Australia.

Aravinda S. Rao
Tel.: +61-3-9035 4203
Fax: +61-3-8344 6678
E-mail: aravinda@student.unimelb.edu.au

We show that the proposed approach results in accurate estimates of crowd density. While the maximum mean error of 3.62 was received for MCG and PETS datasets, it was 2.66 for UCSD dataset. The proposed approach delivered superior performance in 50% of the cases on PETS 2009 dataset when compared with existing methods.

**Keywords** Video surveillance · crowd · density estimation · people counting · crowd monitoring · optical flow · clustering.

## 1 Introduction

Monitoring crowd and understanding its behavior has been one of the foremost research frontiers currently. One of the objectives of crowd monitoring is to ensure safety and security of the individuals in public gatherings (music concerts), public shared spaces (shopping malls), roadways (pedestrian crossing, urban streets), transport facilities (in airports, public transport terminals), sporting events (such as in stadia, olympics), and importantly, in the case of emergency evacuations (in the event of fire, terror threat, natural disaster, building collapse etc.). In the event of emergencies such as fire, stampede, eruption of violence etc., we require an estimate of crowd density in different spaces. Manual counting of the people from several live streams of videos at a centralized location becomes tedious, which is the current and widely adopted approach. Often, personnel would experience fatigue with continuous monitoring of these feeds and it becomes impossible to monitor several streams in case of emergencies.

To help alleviate the problem of manual counting, several people detection, counting and tracking systems have been proposed with some commercial outcomes. A comprehensive survey of human activity recognition and behavior can be found in [80]. Often these systems fail in occluded scenarios due to either unavailability of information about the occluded object during detection or the inability to establish the correspondence upon reappearance of the same object during tracking. People counting based crowd monitoring systems detect and locate the individuals to arrive at the number of people in a scene, whereas density estimation based approaches, without locating individuals, provide a mapping of the group of people to the crowd density in the scene (in a specific area) using certain features. For monitoring the crowd in public spaces, counting people by detecting the individuals and then tracking is infeasible because of the occlusions and matching the reappearances.

One of the primary objectives of monitoring the crowd in public spaces is to provide a reasonable estimate of people for emergency evacuations and path planning. Zhan *et al.* [92] and Jacques Jr. *et al.* [41] have provided the importance of crowd analysis and their applications (public space design, creation of virtual environments, visual surveillance and intelligent environments) in managing large crowds. Crowd monitoring systems face the difficulties in detecting motion from the scene due to varying environmental conditions. Nonuniform illumination, cast shadows, nonrigid human movements and occlusions are some of the major challenges [90]. The challenges associated with people detection and tracking are mainly ascribed to the nonrigid human body motions. Humans as nonrigid objects, in general, fall under the category of elastic and fluid movements [2, 45]. The challenges in object detection and tracking include [90] : (a) loss of information (from 3D to 2D), (b) presence of video noise, (c) nonrigid (articulated) object motions, (d) inter-object occlusions, self-occlusions, partial occlusions and full occlusions, (e) random object motions, (e) nonuniform scene illumination, and (f) near real-time processing requirements.

Most of the existing visual monitoring systems are applied to detect and track vehicles (particularly cars) by the police to patrol highways. Many crowd monitoring systems do exist, but they face challenges and problems during the high-density crowded scenarios. Detecting and tracking a single object is relatively less challenging when compared to the multi-object and high-density circumstances. Additionally, a further degree of difficulty is added for the human detection and tracking because of the nonrigid body movements. Nonrigid body movements imply that the movements of an object is incoherent and different parts of the body move differently (varying directions and velocities); whereas, rigid body movement (such as vehicle) follows coherent motion (holds the parts of an object together). Consequently, the techniques developed for vehicle detection and tracking would be inapplicable without modifications to the pedestrian tracking in occluded scenarios. Hence, it is of utmost importance to study, analyze and develop algorithms that are specifically targeted at monitoring the crowd.

In this work, we collected data from the Melbourne Cricket Ground (MCG) as a means to estimate crowd density at entry and exit points of the concourse. Six locations were chosen for monitoring crowd movement. Fig. 1 shows the schematic map of the MCG camera locations and their view. The goal of this research is to estimate the crowd density without using supervised video subvolume features or classification to estimate the density. In this work online refers to the way the processing of frames are carried out. The proposed ap-

proach does not require all the frames; instead, the density is estimated framewise. Offline processing refers to batch processing where the processing is commenced after the availability of the complete video. The key differentiating factor from the existing works is that most of the methods are offline: use features that are extracted from the the entire video sequence, trained and classified to arrive at the estimate.

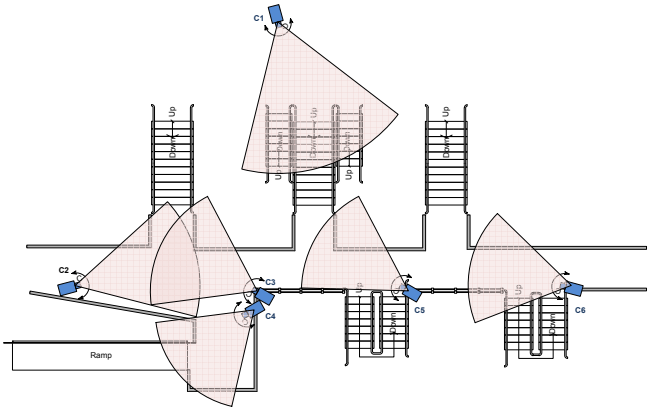



**Fig. 1** Crowd monitoring at MCG: a small section of the concourse was used as experimental testbed. This figure shows the schematic view of six camera locations deployed at the MCG.

In this regard, we present an optical flow based approach for crowd density estimation. The optical flow motion estimation is used as a primary basis for deducing all the outcomes. Different signal processing blocks, filters and mathematical tools are analytically and efficiently combined to provide the estimate of the density in the scene. We believe that this work provides the necessary foundation for motion tracking and analysis in the crowded scenarios, and also a basis for future research in crowd behavior analysis. The proposed approach has been tested on MCG dataset—with low light, severe occlusions, perspective distortion and low-quality video. To validate the proposed approach, the method was tested on two papular public crowd datasets—PETS 2009 and UCSD Pedestrian Database—with medium to high density crowds in the presence of shadows and occlusions. These datasets contain only humans and hence it is assumed that detection of objects refer to humans only. For identification of vehicles, humans, animals and other objects, the proposed approach requires addition of detection and classification steps to distinguish these objects, which is not pursued in this work. The advantages of the proposed approach are: (a) use of motion features to calculate the crowd density as opposed to background modeling, model-based

approaches and texture methods and (b) estimation of density is online as opposed to offline.

Section 2 provides a review of relevant background modeling, motion estimation, human modeling, head detection and crowd density estimation approaches. Section 3 provides the flow of the proposed approach. Section 4 provides the object detection using motion estimation and contour analysis. Section 5 describes the crowd density estimation problem and the proposed approach. Section 6 provides the complete dataset and implementation details followed by results and discussion, and with Section 7 concluding this work.

## 2 Related Work

The primary requirement in object detection is to separate the moving objects from the background. Intuitively, one of the ideas is to model the background of the scene and then classify the objects that do not belong to the background model as foreground (objects). A review of such background modeling schemes is provided in Section 2.1. The other approach to extract the foreground is the motion estimation scheme, which estimates the inter-frame motion information based on the pixel displacements. Section 2.2 provides the review of motion estimation schemes. The second step in the object detection is to ascertain that the detected objects are indeed humans. A review of model based human detection is provided in Section 2.3. Others prefer head detection to ascertain the humans. Section 2.4 provides a review of head detection based approaches and finally Section 2.5 provides a review of relevant density estimation approaches.

### 2.1 Background Modeling

In the process of motion detection and estimation for object tracking, the separation of the object is the most critical of all. Frame differencing, a basic approach to detect the changes, can be obtained by performing difference of two frames or by taking difference between a reference frame and the current frame [25]. The resulting difference can be binarized based on global thresholding [46] or multiple thresholds [62]. Additionally, the accumulative difference image is an another way to segment the background from the foreground [42]. Pfinder [85] used a single-Gaussian multi-class statistical pixel-based model that maintains the pixel values using the Gaussian model and thus tolerates the spontaneous pixel noise.

The notion of adaptive background modeling was introduced using the Mixture of Gaussian (MoG) [26, 74]. Each pixel value was modeled as a MoG of recently observed pixel values (most probable background pixels will have more weight in the mixture). A new Gaussian was created by replacing the least probable Gaussian from an open-ended list of Gaussians to incorporate the new pixel value. Depending on the mean and variance of the pixel values, the pixel that did not fit into any of the Gaussians was labeled as foreground. This is one of the most popularly adopted approach in background modeling.

MoG was generalized using the Kernel Density Estimation (KDE) [21], where the recent values of the pixels were modeled as kernel. Up to this point only the intensity values were used to model the background. Color Gaussian distribution of each pixels was modeled to exploit the color information by [95] inspired by [85]. It is worth noting that color information may change significantly when cast shadows and nonuniform illumination are present, particularly in monitoring crowd in the unconstrained environment. Others have used the eigenvalue decomposition for background subtraction [60]. Eigenvalues represent the variance in the decreasing order. Hence, eigenvalues of a region of pixels identify those pixels with the higher fluctuations. Some have adopted to the running Gaussian average [98], spatial correlation of pixels [69], or the bimodal distribution of pixel values (either foreground or background) [27–29] for modeling the background. It should be noted that all of these approaches use recent pixel observations. In contrast, in [7], for each pixel (in a given color space), a set of sampled values were taken from the past pixel values or from the neighborhood and the Euclidean distance between the new pixel value and the set of samples were computed and stored. The downside of modeling background is that the systems need to learn the model of the scene. In general, learning the scene model depends on the learning rate and the sample frames (with no objects). Furthermore, when the background has the wide dynamic ranges due to the presence of large number of individual objects and occluded objects, modeling the background would require a comparatively longer time than when only a few objects are present and are not occluded.

### 2.2 Motion Estimation

In contrast to the background modeling schemes, motion estimation comprises of estimating the 2D pixel velocities between frames. Motion estimation deduces movements in the scene based on the apparent motion of the objects in the scene. Optical flow is one such

approach to estimate the motion (measuring velocity) components of objects in the scene. Based on estimating the 2D flow vectors, motion estimation techniques can be grouped into four categories [8]: (a) gradient-based techniques, (b) region-based techniques, (c) energy-based techniques and (d) phase-based techniques. Gradient-based techniques [34, 50, 59, 79] derive pixel velocities using the spatio-temporal derivatives. Region-based techniques aim to match the pixel intensities in a regional window by maximizing the similarity scores [5, 70]. Energy-based techniques use the spatio-temporal windows and texture features in the Fourier (frequency) domain to estimate the power spectra of the window and calculated the optical flow velocities [33, 32]. Phase-based techniques compute the velocity vectors based on tracking contour phases [82, 23]. Dense optical flow [34] considers the global view of the scene and estimates the motion, where estimation of motion vectors are smooth because of regularization. Sparse optical flow [50] averages the pixel values in a window and then estimates the motion. The drawbacks of the motion estimation approaches is that the techniques developed are applicable only when there is a motion. In cases of object positioned at a particular location for a long period of time, the object goes undetected.

### 2.3 Human Modeling

To overcome the difficulty of incoherent motions of the humans [2], shape-based analysis of the detected regions are widely used to the address problem of human detection and classification. In [93] and [49], the problem of segmentation (human crowd with occlusion) was handled using the three-dimensional models and shapes to interpret the foreground. In [94] and [96] both shape and appearance-based (color features) models were maintained. Templates were used to represent the human shape information [11, 24, 20, 97] and wavelet-based templates were also been employed [61]. Part-based matching methods include the prior object shape information such as head, hands, torso, legs etc., to be matched with the detected shapes [18, 91, 78, 16]. Shape models may also include the points, silhouettes, articulated shapes [10], and skeleton models. Active Appearance Models (AAM) use both the shape (spatial) and local (texture, color, edge) features [43, 9, 88, 67]. AAMs are suitable for face recognition, eye tracking and medical image segmentation. Nevertheless, matching the human models require definite models, object structures, predetermined color and reproducible features.

### 2.4 Head Detection

In most of the surveillance and tracking systems, it is assumed that the humans are walking upright. One of the prominent features to track humans is to detect the head. Since most of the body parts follow articulated motion, head maintains a stable shape for each of the individuals. In this regard, people tracking and counting systems have employed head detection approaches [29, 40, 36, 57, 58, 55, 39]. Head detection based methods are also used in estimating the crowd density. Head detection assists in localizing the individuals in crowded scenarios. Global head detection and 2D Gaussian kernel based regression was employed in [66] to estimate the density in crowded scenarios. Interest points based on the orientation of the gradients were used to form a binary image. In [75], LUV channels, intensity of the gradient channels and six orientation of the gradient channels were used as features for head detection. However, the visibility of the heads depend on the camera angle and the severity of occlusion. Complete top-view would always provide better results using the head detection, whereas tilted cameras suffer from the occlusion. Further, other density estimation approaches based on head detection will be dependent on the accuracy of detected heads and the detection rate is currently low.

### 2.5 Density Estimation

Most of the crowd density estimation schemes use either the texture features on the local and global levels or extract the foreground pixels using the motion information. The extracted features of the foreground pixels are then mapped to the crowd density in a given region. Texture features such as Gray Level Dependence Matrix (GLDM), Minkowski Fractal Dimension (MFD), Translation Invariant Orthonormal Chebyshev Moments (TIOCM) were used to classify the crowd density into five classes (very low, low, moderate, high and very high) using Self Organizing Maps (SOM) [64]. GLDM was used at local and global levels as features and Support Vector Machine (SVM) as classifier for the abnormal crowd density detection [86]. Kanade-Lucas-Tomasi (KLT) [50] tracking features were clustered to estimate the crowd density in public venues [3]. Statistical texture measures such as contrast, homogeneity, energy, entropy, Gradient Orientation Co-occurrence Matrix (GOCM) were used in conjunction with the $\nu$-Support Vector Regression ($\nu$-SVR) for estimating density [53,

52]. Gaussian regression process to density estimation using geometrical, edge and texture features can be found in [12].

Background modeling based approaches can also be found in the literature. Features such as blob area, Harris corner, KLT feature points, contour number, contour perimeter, ratio of contour perimeter to area, Canny edge and fractal dimension were treated as inputs to multi-variable linear regression model [54]. Combining neural networks with foreground pixels and morphological operations, the crowd density was estimated in [37]. Density was also estimated by accumulating foreground pixels over time followed by the use of texture features such as the four orientations: $0^\circ$, $45^\circ$, $90^\circ$ and $135^\circ$ to create the Gray Level Co-occurrence matrix (GLCM) and statistical features (energy, entropy, homogeneity, and contrast) [73].

In contrast, crowd modeling was also performed using motion estimation followed by Markov Random Field (MRF) to locate the objects' positions and application of the least-squares method to reduce the neighborhood search space [37]. Furthermore, motion frequency using the Discrete Cosine Transform (DCT) coefficients and SVM to estimate the density was proposed [38]. Using moving Speed Up Robust Feature (SURF) points and their clusters as input features for the $\varepsilon$-SVR, density estimation was performed in [17]. Most of the aforementioned approaches to density estimation require training. Consequently, a change in the scene or camera or orientation requires retraining of the system, which is undesirable as an end user.

In this paper, we propose the crowd density estimation using motion estimation followed by the hierarchical clustering. Willick and Yang [83] indicated that Horn and Schunk's motion estimation scheme [34] performed better compared to Nagel's work [59]. Additionally, since the selection of window size in Lucas-Kanade [50] affects the flow vector determination, nonrigid human motion may generate noisy vectors compared to dense optical flow. Therefore, in this work, we use the dense optical flow for motion estimation by Horn and Schunck [35]. Fig. 2 presents an overview of the developed crowd monitoring system. The system is divided into five major blocks (see Fig. 2): (a) camera inputs, (b) prepossessing filters, (c) people tracking, (d) motion tracking, and (e) crowd events. In this work, the path chosen to estimate the crowd density is: input from camera:1 $\rightarrow$ preprocessing filters:2 $\rightarrow$ multiple objects:3 $\rightarrow$ density estimation:4 $\rightarrow$ people count:5. We assume that the video data from cameras are directly accessible, the cameras are calibrated and only human objects are

present. Sample frames from the three datasets (MCG, PETS 2009, and UCSD)are shown in Figs. 3, 4, and 5.

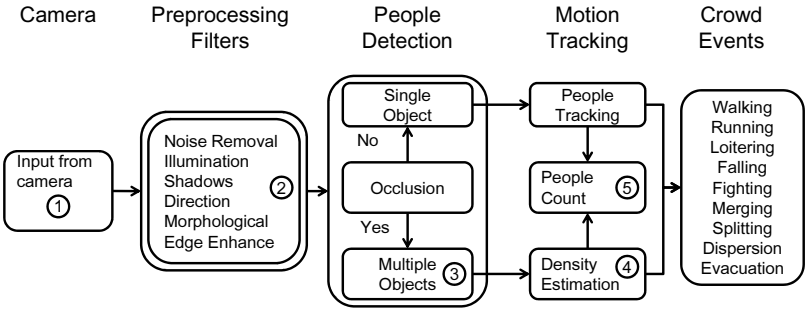



**Fig. 2** Overview of crowd monitoring system

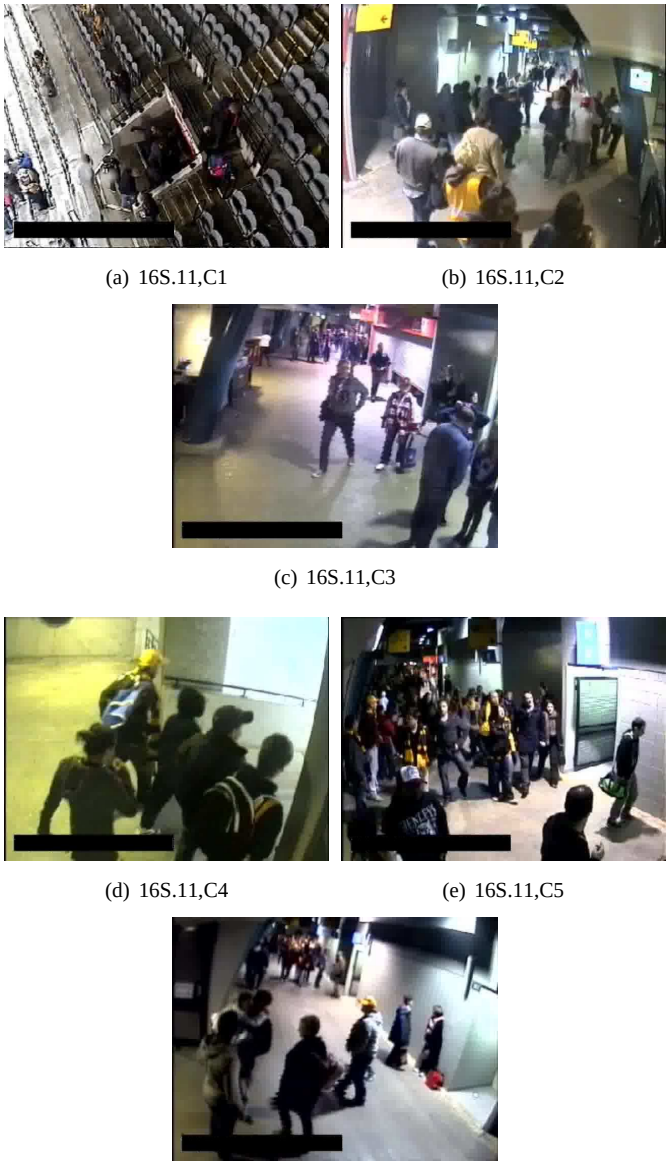
## 3 Flow of the Proposed Approach

Let $\mathbf{I}_{p\times q}$ represent an area in the frame $\mathbf{I}(x,y)$, where $p,q \in \mathbb{Z}_+$ represent the number of pixels. The problem of crowd density estimation then translates to

$$f : \mathbf{I}_p \times \mathbf{I}_q \in \mathbb{R}^2 \rightarrow f(\mathbf{I}_{p\times q}) \in \mathbb{N}, \tag{1}$$

where $f(\mathbf{I}_{p\times q})$ is the crowd density in $\mathbf{I}_{p\times q}$ of $\mathbf{I}(x,y)$. Thus, the goal of the crowd density estimation now turns to estimating number of people in a given region. In other words, the function maps the number of moving objects to the region in the foreground.

1. At first, nonlinear distortions introduced such as barrel or pincushion distortions by low-cost optics are rectified by applying lens correction profile to each of the frames. Additionally, the *perspectivity* introduced in imaging from 3D to 2D to central projection are fixed by perspective correction.
2. Next, the video frames are pre-processed to remove any video noise such as chroma noise, blurred scene and speckle noise or so by applying suitable filters.
3. Later, motion vectors of the scene are computed from the optical flow to narrow the search space (of foreground pixels) for estimating the crowd density. In other words, we do further computation on motion vectors and their locations of the video scene in the current scene instead of the entire video scene.



(a) 16S.11,C1 (b) 16S.11,C2

(c) 16S.11,C3

(d) 16S.11,C4 (e) 16S.11,C5

(f) 16S.11,C6

**Fig. 3** Sample frames from MCG dataset for different cameras

4. Motion vectors corresponding to the zero angular orientation are eliminated. We consider only those motion vectors that have both horizontal and vertical motion components. This step provides a means of filtering the speckle noise equivalent generated by the motion vectors.



(a) S1-L1, Timestamp: 13-57, View-001, Frame #130



(b) S1-L3, Timestamp: 14-17, View-001, Frame #090



(c) S1-L2, Timestamp: 14-06, View-001, Frame #090

**Fig. 4** Sample frames from PETS 2009 dataset [22]



(a) vidf1_33_000, Frame #001



(b) vidf1_33_007, Frame #001



(c) vidf1_33_008, Frame #001

**Fig. 5** Sample frames from UCSD Pedestrian Database dataset [13].

5. Contours are drawn for each of the connected regions. This step demarcates the crowd region and provides as a input to the next level of processing. To demarcate these regions truthfully to crowd regions only, step 2 is essential.

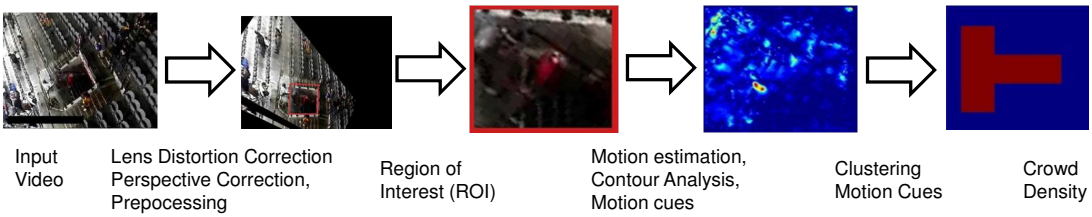



**Fig. 6** Flow of crowd density estimation by clustering motion cues. In the Region of Interest (ROI), the actual number of people were 2. One can notice that this is a low-light region. The proposed solution estimated the density to be 3.

6. Refine the motion cues by spatially filtering the motion vectors using the median filter to reduce rapid changes in motion information. Motion vectors in a space of the video region can posses high variance in case of crowd due to articulated movements of people. This variance is reduced by means of median filter and assigning the filtered value to all the elements of the block.
7. Finally, foreground pixels are mapped to the crowd density by clustering the motion cues hierarchically. The association of the crowd density in a region is a function of block size and the motion information obtained from spatially-filtered motion vectors. Clustering the motion vectors is tantamount to localizing the movements corresponding to grouped objects and thus providing an estimate of the crowd density.

The complete flow of the proposed crowd density estimation is provided in Fig. 6.

## 4 Object Detection

In this section, the building blocks of detecting the moving objects for density estimation are described. The preprocessing block, estimates of the the motion in the scene from two consecutive frames using optical flow, and finding the contours and their analyses are presented in the following subsections.

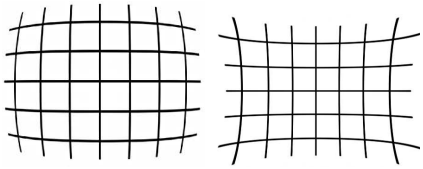
### 4.1 Lens Distortion Correction

Most of the optical devices introduce nonlinear distortions. In case of image formation in a camera, two main types of distortion are introduced: radial distortion and tangential distortion. Radial distortion introduces barrel and pincushion effects, whereas the less-severe

tangential distortion arises when the image plane and the lens plane are not parallel. Radial distortion proves to extremely distort the images compared to tangential. Tangential distortion is attributed to the manufacturing defects. Consequently, in this work, only the radial distortion is considered. The radial distortion is corrected by determining the coefficients ($K = \{k_1, k_2, k_3\}$) regulated by the 6$^{\text{th}}$-order polynomial equation [51, 30]:

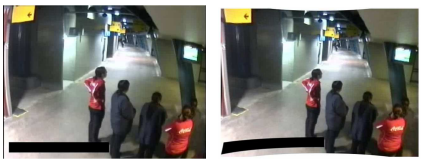
$$f(r) = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \tag{2}$$

where $f(r)$ is the functional with $r = x^2 + y^2$ and $(x, y)$ forming the image coordinates. When $f(r) < 1$, the barrel distortion arises and pincushion effect surfaces when $f(r) > 1$. Fig. 7 shows the barrel and pincushion distortions and Fig. 8 shows an example of "barrel" distortion present in one of the frames of MCG dataset and its correction.



(a) Barrel distortion (b) Pincushion distortion

**Fig. 7** (a) Barrel distortion and (b) pincushion distortion due to nonlinear optical effects of a camera lens [63].



(a) 16S.11,C2 (b) 16S.11,C2

**Fig. 8** (a) Presence of the barrel distortion in C2 is more clearly visible towards the left edge of the frame where the vertical structure appears to be bent outwards and (b) distortion corrected frame.

## 4.2 Perspective Correction

During image formation, the light from the 3D projective space ($\mathbb{P}^3$) is projected onto a 2D space $\mathbb{P}^2$ through the fixed centre of projection. If we denote $(X, Y, Z, 1)$ as a point in $\mathbb{P}^3$ and

$(x,y,1)$ in $\mathbb{P}^2$ in homogeneous coordinates, then the perspective projection (camera matrix) can be written as:

$$\underbrace{\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}}_{\text{image plane}} = \underbrace{\begin{bmatrix} f_x & sk & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic parameters}} \underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} & t_X \\ R_{21} & R_{22} & R_{23} & t_Y \\ R_{31} & R_{32} & R_{33} & t_X \end{bmatrix}}_{\text{extrinsic parameters}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\text{world plane}} \tag{3}$$

$$= \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

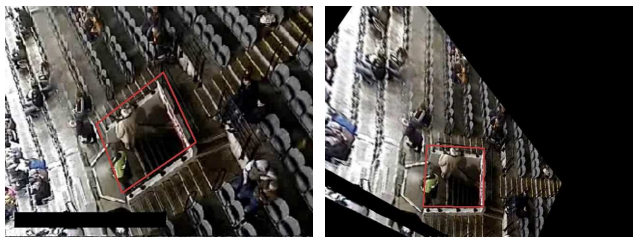
where $(f_x, f_y)$ are the focal lengths of in the $(x,y)$ directions of the image in terms of pixel units, $(x_0,y_0)$ is the principal point, *sk* is the skew parameter are the 5 intrinsic parameters. The extrinsic parameters indicate the rotation (**R**) and translation (**t**) between the camera coordinates and the external world.

However, projective transformation introduces perspective distortion through central projection mapping. This distortion projects objects closer to camera to appear bigger and objects far away from the camera to be smaller. The distortion is corrected by estimating the *projectivity* or *collineation* or *homography* from points to points, from lines to lines, from $\mathbb{P}^2$ to $\mathbb{P}^2$ [31]. The homography matrix (**H**) is a non-singular matrix and hence invertible. The homography is estimated by using Direct Linear Transform (DLT) given by:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{4}$$

$$= \mathbf{H} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

where *s* is a nonzero scalar. The matrix **H** has 8 degrees of freedom and hence requires 4 point correspondences between images. An example of perspective corrected frame from MCG dataset is shown in Fig. 9.



(a) 16S.11,C1 (b) 16S.11,C1

**Fig. 9** (a) View of C1 and (b) its perspective corrected view for the MCG dataset.

### 4.3 Preprocessing

Let $(x,y)$ denote a pixel in a frame $\mathbf{I}(x,y)$. In this work, $x$ axis indicates the horizontal axis of the frame and the $y$ axis represents the vertical axis of the frame. At this stage, every frame is first bilaterally filtered to preserve edges and reduce variations in color pixels with $\sigma_d = 5$ and $\sigma_r = 10$. Bilateral filtering is applied to boost the sharpness of the frames while maintaining perceptually strong colors and edges. Next, these frames are converted to grayscale. The grayscale frames are then low-pass filtered to eliminate high frequency noise. This is accomplished using two-dimensional Gaussian filter with $\mu = 0$ and $\sigma_x = \sigma_y = 0.1$. The filtered frames are then used for motion estimation using the optical flow [34].

### 4.4 Motion Estimation

For a given sequence of frames, the motion is computed using the optical flow approach. The brightness at point $\mathbf{I}(x,y)$ is denoted by $\mathbf{E}(x,y,t)$, where $t$ represent the time. Assuming that a moving object retains the constant brightness, estimation of motion between two frames of a video sequence is given by optical flow [34]:

$$\frac{\partial \mathbf{E}}{\partial t}\frac{dx}{dt} + \frac{\partial \mathbf{E}}{\partial t}\frac{dy}{dt} + \frac{\partial \mathbf{E}}{\partial t} = 0 \tag{5}$$

Further,

$$\mathbf{E}_x u + \mathbf{E}_y v + \mathbf{E}_t = 0 \tag{6}$$

where $\mathbf{u} = \frac{dx}{dt}$ and $\mathbf{v} = \frac{dy}{dt}$, represent the velocities of matching pixels in two frames. Spatially, for each pixel locations $(x,y)$, the resultant velocity vector of two-dimensional motion ( $\mathbf{u}(x,y) = \frac{dx}{dt}$ and $\mathbf{v}(x,y) = \frac{dy}{dt}$ ) with time is calculated. The matrix of resultant vectors'

magnitude and directions are given by:

$$\mathbf{I}_R(x,y) = \sqrt{(\mathbf{u}(x,y)^2 + \mathbf{v}(x,y)^2)} \tag{7}$$

and

$$\mathbf{I}_D(x,y) = \arctan\left(\frac{\mathbf{v}(x,y)}{\mathbf{u}(x,y)}\right) \tag{8}$$

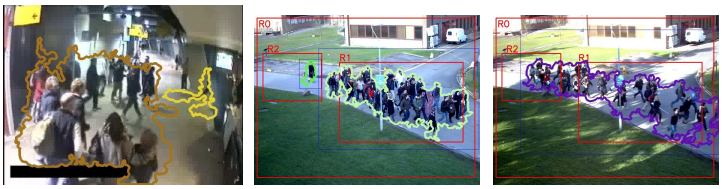
where $\mathbf{I}_R(x,y)$ and $\mathbf{I}_D(x,y)$ together indicate the presence of moving objects. Matrices $\mathbf{I}_R$ and $\mathbf{I}_D$ are used to detect the object boundaries in contour analysis. Algorithm 1 shows a pseudocode of preprocessing and motion estimation.

### 4.5 Contour Analysis

Contour analysis includes demarcating the boundaries of the possible object detected from the previous steps. Both the magnitude ($\mathbf{I}_R$) and direction ($\mathbf{I}_D$) matrices are used for this analysis. For each iteration of the optical flow between two frames, the direction matrix $\mathbf{I}_D$ comprises of vector directions for each of the pixel movements. The angle of these vectors are confined to $[0,360)$ degrees. To segment the motion of objects, $\mathbf{I}_D$ is binarized for only

**Algorithm 1:** Initialization

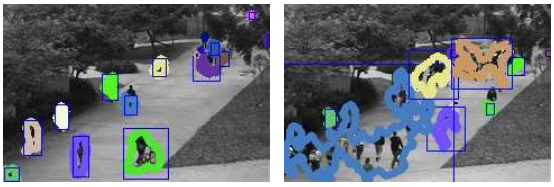
```
Data: initialization(video, perspective point correspondences, lens profile)
Result: opticalFlow (I_O)
/* Correct radial lens distortion                                   */
lensProfileCorrection(video);
/* Initialize video frames                                          */
totalNumberOfFrames = numberOfFrames(video);
processingFrame = 1;
while processingFrame < totalNumberOfFrames do
    /* Perform perspective correction                               */
    currentFrame = perspectiveCorrection(currentFrame);
    nextFrame = perspectiveCorrection(nextFrame);
    /* Remove noise                                                 */
    currentFrame = preprocessing(currentFrame);
    nextFrame = preprocessing(nextFrame);
    /* Compute optical flow                                         */
    I_O = computeOpticalFlow(currentFrame, nextFrame);
end
```



(a) 16S.11, C2, MCG Dataset

(b) S1-L2, Timestamp: 14-06, View-001, Frame #136, PETS 2009 dataset

(c) S1-L3, Timestamp: 14-17, View-001, Frame #090, PETS 2009 dataset



(d) vidf1_33_000, Frame #003, UCSD dataset

(e) vidf1_33_007, Frame #003, UCSD dataset

**Fig. 10** Examples of contour analysis: (a) Contours detected for MCG dataset, (b)-(c) PETS 2009 dataset, and (d)-(e) UCSD dataset. (Note: perspective correction is not shown here for image clarity)

those pixels with $|\mathbf{I}_D(x,y)| > 0$. The speckle noise in the foreground pixels that may originate either due to presence of **u** or **v** component is eliminated. The binarized $\mathbf{I}_D$ is further smoothened using $(7 \times 7)$ median filter to reduce any speckle noise. Morphological closing is performed to enhance the motion segmentation. Next, the contours $\mathbf{C} = \bigcup_i^n c_i$ are determined for each of the nonzero foreground components. For simplicity, the subscript $i$ is dropped in future contexts. Fig. 10 shows the examples of detected contours following preprocessing step (without perspective correction) including motion estimation and filtering of the flow vectors for MCG dataset, PETS 2009 and UCSD dataset. Similar results would be obtained with the inclusion of perspective correction. However, the size of the objects would be normalized in the latter case. Algorithm 2 shows the extraction of contours.

## 5 Crowd Density Estimation

### 5.1 Foreground Mapping

Each object occupies certain area (region of foreground pixels belonging to an object) in the foreground due to its motion. In case of single object, the foreground region corresponding to the object will be less when compared to multiple objects moving together at same

**Algorithm 2:** Contour analysis

**Data**: contourAnalysis($\mathbf{I}_O$)
**Result**: contours $\mathbf{I}_C$
`/* Extract contours */`
**if** $\mathbf{I}_O$ **then**
  $\mathbf{I}_R(x,y) = \sqrt{(\mathbf{u}(x,y)^2 + \mathbf{v}(x,y)^2)}$ ;
  $\mathbf{I}_D(x,y) = \arctan\left(\frac{\mathbf{v}(x,y)}{\mathbf{u}(x,y)}\right)$;
  **while** *not end of frame* **do**
    `/* Apply region of interest mask */`
    $\mathbf{I}_R(x,y) = \mathbf{I}_R(x,y)$ **AND** ROI;
    $\mathbf{I}_D(x,y) = \mathbf{I}_D(x,y)$ **AND** ROI;
    **if** $\mathbf{I}_D(x,y) == 0$ **then**
      $\mathbf{I}_R(x,y) = 0$ ;
    **else**
      $\mathbf{I}_R(x,y) = 1$ ;
    **end**
  **end**
  `/* Extract contours from boundaries */`
  $\mathbf{I}_C$ = findContours($\mathbf{I}_R(x,y)$);
**end**

scale. The contour of the multiple unoccluded objects form gaps (valleys between objects—Zhao *et al.* [94] provided silhouette projections of multiple objects where one can observe valleys) in the foreground mapping, whereas, the contour of a partially occluded group fills these gaps to a major extent (this was the phenomenon that was found in case of partially occluded objects). Now, the problem is to learn the crowd density corresponding to the foreground regions. Motion cues has been used for mapping the movements of the objects to the foreground pixels, median filter of the motion vectors to create the blocks in the foreground regions and the hierarchical clustering to map the foreground regions into the crowd density.

## 5.2 Selection of Motion Cues

Most of the methods describing density estimation in Section 2 require training to deduce a function that provides crowd density in the scene. Briefly, once the contours are extracted, the region of interest (ROI) is binarized to generate a contour mask $c_{\text{mask}}$. The optical flow magnitude $\mathbf{I}_R$ corresponding to the $c_{mask}$ are processed for density estimation. Bounding

box information is derived for each of the contours. The bounding box region of the $\mathbf{I}_R$ (corresponding to $c_{mask}$) is divided into $b \times b$ block. In order to alleviate the flow noise, for every block, the pixel values in each of the blocks are replaced by median filter ($b \times b$) output. The resulting matrix is termed as *motion matrix* $\mathbf{M}$, is a matrix with the blocks of size $b \times b$ for every contour and each block corresponding to the median values of original $\mathbf{I}_R$. In this approach, since perspective correction is used, the block size is fixed to $32 \times 32$.

Block size for uncorrected perspective scene can be determined by the height and width of the objects obtained during camera installation and calibration process. For each of the contours in $C$, the contour width $c_{\text{width}}$, contour height $c_{\text{height}}$, contour orientation $c_{\text{angle}}$, contour starting point $c_x$ and $c_y$ are obtained. Average single object size $O_s$ and object area $O_a$ are determined using contour dimensions. The block size $b$ is determined by using the single object contour size in the scene. The block size $b$ is given by:

$$b = \frac{1}{N_{\text{SO}}} \times \sqrt{c_{\text{SO}_{\text{width}}} \times c_{\text{SO}_{\text{height}}}} \tag{9}$$

where $c_{\text{SO}_{\text{width}}}$ and $c_{\text{SO}_{\text{height}}}$ are height and width of a single object contour, and $N_{SO} = 3$ is chosen such that at least $N_{SO}$ blocks cover the width of a single object.

**Algorithm 3:** Motion cues

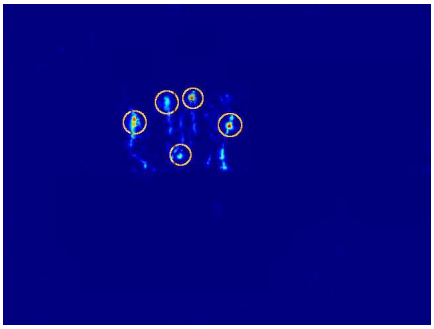
**Data**: motionCues($\mathbf{I}_C$)
**Result**: Motion cues ($\mathbf{I}_M$)
**while** $\mathbf{I}_C$ **do**
  `/* Compute motion matrix */`
  $\mathbf{M}$ =createBlocks($\mathbf{I}_O$);
  **for** *each block in* $\mathbf{M}$ **do**
    $\mathbf{M}_{\mathbf{b}\times\mathbf{b}}$ = mean($\mathbf{M}_{\mathbf{b}\times\mathbf{b}}$)
  **end**
  `/* Derive motion cues */`
  $\mathbf{I}_M = \frac{(\mathbf{M}-\mathbf{M}_{min})}{(\mathbf{M}_{max}-\mathbf{M}_{min})}$
**end**

## 5.3 Clustering Motion Cues

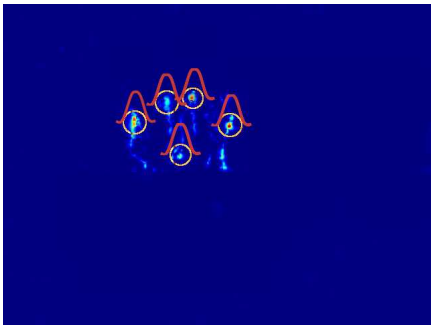
Let $\mathbf{M}_{min}$ and $\mathbf{M}_{max}$ denote the minimum and maximum of $\mathbf{M}$. The *motion cue matrix* or *motion cues* $\mathbf{I}_M$ is obtained by relatively scaling the motion information as $\mathbf{I}_M =$



(a) Current frame



(b) Motion vectors' signature for 5 people in the scene



(c) Gaussian shaped distinct points for 5 people

**Fig. 11** Refining and clustering motion cues to estimate density. (Note: perspective correction is not shown here for image clarity)

$\frac{(\mathbf{M}-\mathbf{M}_{min})}{(\mathbf{M}_{max}-\mathbf{M}_{min})}$. Fig. 11 shows an example of the motion information matrix (without blocks). We see that relatively high magnitude motion information form a 2D Gaussian-shaped points. The distinct points (highlighted points) correspond to distinct number of people in the scene. As a result, one can generalize that the motion cue matrix $\mathbf{I}_M$ highlights the regions in the scene that approximately correspond to identifying individual objects i.e. if there are $N$ people in a scene, then there would approximately be $N$ corresponding distinct points. Thus, we can deduce the density of the people in the scene from the motion cues.

In order to deduce density, isolation of distinct points is necessary. A natural way to achieve this is by clustering the motion cues. At first, dissimilarity matrix of the motion cues is obtained as $\mathbf{D} = {\mathbf{I}_M}^T \mathbf{I}_M$. Spectral clustering methods such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD) require input about top $d$ eigenvalues, which limits our ability to determine the number of clusters present. $K-$means clustering requires the desired number of clusters, which is unattractive for our purpose. However, on

the other hand, hierarchical clustering provides us a means to isolate distinct points that correspond to number of people in the scene. This forms the basis for the use of hierarchical clustering of motion cues.

Intuitively, the idea is to form the clusters of motion vectors representing the foreground objects as patterns of crowd. The $\mathbf{I}_M$ provides a natural 2D Gaussian model for forming clusters with peak values indicating the highest velocities at the center of the objects and decreasing values as moved away from the center. In case of crowded scenes, the inter-cluster variance, a property of clustering algorithm, is used to cluster different objects in the scene. A *K*-means tree representing the clusters of motion vectors would be an ideal approach; however, we do not know the density of the crowd and furthermore, the density changes with time and hence we cannot fix the number of clusters. Instead, we use hierarchical clustering where we use agglomerative clustering to build the clusters (bottom-up). The dissimilarity matrix consists of features from spatially-filtered median motion vectors. In the process of clustering, the most similar blocks are merged and the process is continued across the scene. In agglomerative clustering process, single-linkage clustering is used (nearest-neighbor clustering) as the motion vectors belonging to an object will be similar to its centre and also the dissimilarity increase with increased distance and peaks at the interface between two objects. This process identifies the distinct motion regions in the scene. Minimum spanning tree is constructed using single-linkage clustering and the computational complexity of the clustering amounts to $\mathcal{O}(N^2)$. Average-linkage and complete-linkage computational complexities are of the order $\mathcal{O}(N^3)$. Since we are using 2D plane for the scene analysis, Euclidean distance is used as a metric to perform cluster analysis ,where inter-cluster and intra-cluster variance are used simultaneously to determine the formation of the clusters. Algorithm 3 provides the pseudocode for obtaining motion cues from contours and Algorithm 4 describes the procedure of clustering to extract distinct points. The nearest-neighbor or the single-linkage would create a minimum spanning tree to create clusters where the blocks are distinct since the clusters are formed by linking the motion signatures through edges that are distinct. Algorithm 5 provides a brief overview of process to estimate crowd density.

**Algorithm 4:** Clustering motion cues

**Data**: motionCues

**Result**: clusters

`/* Initialize cluster */`

$m = 0$;

`/* Initialize sequence number for clusters */`

$s = 0$;

`/* Initialize hierarchical level of cluster */`

level(m) = 0;

**while** *clusters* **do**

`/* Find minimum of all dissimilarity blocks */`

$d(i,j) = \arg\min \left\|\mathbf{D}_i - \mathbf{D}_j\right\|_2 = \arg\min \left\|\mathbf{I}_{\mathbf{M_i}} - \mathbf{I}_{\mathbf{M}_j}\right\|_2$;

`/* Increment the sequence number */`

$s = s + 1$;

`/* Merge the clusters to form a new cluster and assign level */`

level($m$) = $d(i,j)$;

`/* Renew the` $I_M$ `with new cluster */`

`/* Repeat until all elements are clustered */`

**end**

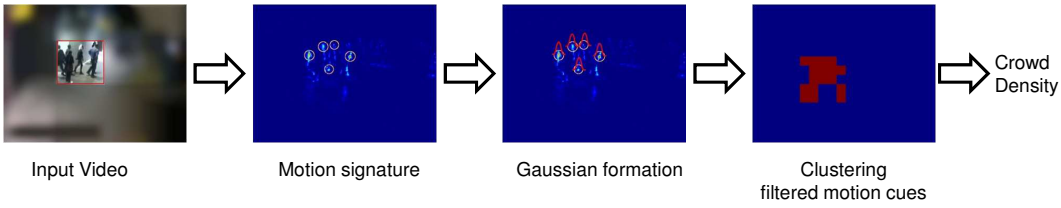



**Fig. 12** Overview of clustering motion cues. (Note: perspective correction is not shown here for image clarity).

## 6 Performance Evaluation

The implementation of the proposed crowd density estimation approach was carried out in OpenCV 2.3 on a Virtual Box Linux machine (32-bit Ubuntu 12.04 LTS) equipped with 1.5GB RAM and Intel® i7 – 2600 CPU running at 3.4GHz. The proposed approach was tested on the MCG dataset and two popular crowd datasets: PETS 2009 [22] and UCSD Pedestrian Database [13].

**Algorithm 5:** Crowd density estimation

```
Data: crowdDensity=(video, perspective point correspondences, lens profile)
Result: crowdDensity
totalNumberOfFrames = numberOfFrames(video);
processingFrame = 1;
while processingFrame < totalNumberOfFrames do
    if (currentFrame AND nextFrame) ≠ NULL then
        /* Correct lens distortion                                   */
        lensProfileCorrection();
        /* Correct perspective distortion                            */
        currentFrame = perspectiveCorrection(currentFrame);
        nextFrame = perspectiveCorrection(nextFrame);
        currentFrame = preprocessing(currentFrame);
        nextFrame = preprocessing(nextFrame);
        /* Compute optical flow                                      */
        I_O = computeOPticalFlow(currentFrame, nextFrame);
        /* Extract contoours                                         */
        I_C = contourAnalysis(I_O);
        /* Select motion cues                                        */
        I_M = motionCues(I_C);
        /* Identify objects                                          */
        crowdDensity= clusteringMotionCues(dissimilarity(I_{M_{b×b}}));
    else
        End of video reached or error in reading frames;
    end
end
```

## 6.1 Dataset

Table 1 provides detailed information about the three datasets. MCG dataset was originally collected in Advanced Systems Format (asf) from six cameras on four different days of Australian Football League (AFL) matches held at MCG. The data was collected during the start and end of the games. We collected a total of nearly 6 hours of data per camera from each camera at 30 fps (equivalent to 36 hours of data at 30fps in total). For the purpose of crowd density estimation, four cameras were chosen (C1, C2, C4, & C6). In-line with standard public datasets, one minute of video from each camera was extracted during the peak crowd

movements and converted to JPG format for analysis. The frame size is $640 \times 480$ and in RGB format. The ground truth was generated manually using customized software.

PETS 2009 dataset images are in JPG format with a size of $768 \times 576$. View 001 is considered for the density estimation for sequences S1.L1, S1.L2 and S1.L3. Different environmental conditions such as shadows, overcast, medium and high crowd density were present with both walking and running events. The crowd density is estimated in regions R0, R1 and R2 as shown in 10 for sequences S1.L1 and S1.L2. For S1.L3, density is estimated in region R1. The ground truth information for PETS 2009 dataset is available in [56]. In the PETS 2009 dataset, the regions R0, R1 and R2 had been marked by the dataset providers. The dataset has three roads (surrounded by green lawns at an educational campus) intersect at a point. From View-001, R1 oversees one of the roads and the intersection point. The road from R1 leads to another road where R2 overlooks that road. Region R0 encompasses not only regions R1 and R2, but also other regions including the third road and walkable area where there is a possibility of peoples movement. The size of the objects in R1 is comparatively larger than in R2. Furthermore, R1 is more occluded compared to R2 because of the camera view and angle.

UCSD dataset contains walking events and comparatively smaller sized objects. The images are with a resolution of $238 \times 158$ in grayscle and PNG format. UCSD dataset is challenging in terms of smaller object size (less motion information and reduced texture features) and techniques dependent on color information would perform poorer because of lack of color information. Initially, using empirical knowledge the block size was set to 6. The ground truth information for UCSD Pedestrian Database dataset is available at [13].

## 6.2 Results and Discussion

Performance of the density estimation is measured using Mean Absolute Error (MAE), Mean Relative Error (MRE) and Root Mean Squared Error (RMSE). MAE, MRE and RMSE can be calculated using (10), (11) and (12):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|A(i) - P(i)| \tag{10}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|A(i) - P(i)|}{A(i)} \tag{11}$$

**Table 1** MCG 2011, PETS 2009 and UCSD Pedestrian Database datasets used in this work. Details of each dataset and video sequence with their conditions are provided. MCG dataset sequences are in (RGB) JPG format ($640 \times 480$). PETS 2009 sequences are in (RGB) JPG format and have the frame resolution of $768 \times 576$. UCSD Pedestrian Database dataset sequences are in (grayscale) PNG format and have the frame resolution of $238 \times 158$

| Dataset | Sequence | View | Environment | Event | Frames | Frame rate | No. of people | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | (fps) | Min | Max |
| MCG 2011 | 16S.11 | C1 | Low density crowd<br>Low light | Walking | 302 | 6 | 1 | 6 |
| | 16S.11 | C2 | High density crowd<br>Occlusions | Walking | 313 | 6 | 4 | 15 |
| | 16S.11 | C4 | Medium density crowd<br>Low light | Walking | 303 | 6 | 0 | 11 |
| | 16S.11 | C5 | High density crowd<br>Low light, Occlusions | Walking | 296 | 6 | 5 | 13 |
| PETS 2009 | S1.L1,13-57 | 001 | Medium density crowd<br>Overcast | Walking | 221 | 7 | 0 | 19 |
| | S1.L1,13-59 | 001 | Medium density crowd<br>Overcast | Walking | 241 | 7 | 0 | 20 |
| | S1.L2,14-06 | 001 | High density crowd<br>Overcast | Walking | 201 | 7 | 0 | 21 |
| | S1.L3, 14-17 | 001 | Medium density crowd<br>Bright sunshine<br>Shadows | Running | 91 | 7 | 0 | 22 |
| UCSD Pedestrian Database 2008 | vidf1_33_000.y | - | Medium density crowd | Walking | 200 | 10 | 12 | 22 |
| | vidf1_33_001.y | - | Medium density crowd | Walking | 200 | 10 | 20 | 27 |
| | vidf1_33_002.y | - | Medium density crowd | Walking | 200 | 10 | 19 | 25 |
| | vidf1_33_003.y | - | Medium density crowd | Walking | 200 | 10 | 11 | 21 |
| | vidf1_33_004.y | - | Medium density crowd | Walking | 200 | 10 | 11 | 23 |
| | vidf1_33_005.y | - | High density crowd | Walking | 200 | 10 | 20 | 40 |
| | vidf1_33_006.y | - | High density crowd | Walking | 200 | 10 | 32 | 45 |
| | vidf1_33_007.y | - | High density crowd | Walking | 200 | 10 | 29 | 45 |
| | vidf1_33_008.y | - | Medium density crowd | Walking | 200 | 10 | 21 | 31 |
| | vidf1_33_009.y | - | Medium density crowd | Walking | 200 | 10 | 17 | 24 |

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|A(i)-P(i)|^2} \tag{12}$$

where $i$ indicates the $i^{\text{th}}$ frame, $A(\cdot)$ is the ground truth crowd density and $P(\cdot)$ is the predicted density. MAE provides a measure of how well the performance is in terms of absolute error i.e. the absolute difference between the actual measured values (ground truth) and the predicted values for each frame and averaged for the given sequence (all the frames). MRE provides a percentage relative error i.e. the ratio of absolute error for each of the frames to the actual value expressed in percentage. This gives us a measure of how good the prediction is relative to the size of the actual value. RMSE provides a measure about variations in predictions and will always be greater than or equal to MAE and $\text{MAE} \leq \text{RMSE} \leq \sqrt{N} \cdot \text{MAE}$ [84]. In other words, closer the RMSE values to MAE, better the predictions results are. Although, these three measures have been used in literature commonly, comparison among different approaches are mainly evaluated based on MAE. This is because of the nature of the application (crowd density) that is more meaningful in terms of MAE for a given sequence and the way the density is estimated is different for different approaches.

MCG dataset was collected indoors, whereas PETS and UCSD were collected outdoors. As a means to validate the proposed approach, it was validated on PETS and UCSD datasets. Further, we provide the comparison with respect to the existing methods on PETS 2009 dataset. Table 2 provides the results for MCG dataset. For three out of four sequences, the MAE error was less than 2 and less than 4 for the entire dataset. Table 3 shows the results for the PETS 2009 datset. For region R0, the MAE was less than 3; MAE was less than 4 in region R1 and less than 1 for region R2. Table 4 shows the results for UCSD dataset. The MAE for four sequences were less than 1, three sequences less than 2 and three sequences (the MAE and MSE for pedestrian movements using supervised schemes for the UCSD dataset can be found in [14]). The results demonstrate that the proposed method provides a reasonably good estimate of the crowd density.

The prediction error plots for the three datasets have been included in Online Resource. The prediction error plots include ground truth, predicted results and the error. Furthermore, for each sequence, mean and standard deviation per person is provided. Most of the methods have provided the mean error for the entire sequence (for all the frames). However, we believe that such a measure would not provide adequate information about the density since the error during increased or decreased crowd density will be averaged over the en-

**Table 2** Results for MCG Dataset. MAE, MRE and RMSE measures have been tabulated.

| Dataset | Sequence | View | Prediction Error | | |
|---|---|---|---|---|---|
| | | | MAE | MRE | RMSE |
| MCG 2011 | 16S.11 | C1 | 01.22 | 29.34% | 3.30 |
| | 16S.11 | C2 | 03.62 | 32.72% | 5.74 |
| | 16S.11 | C4 | 01.95 | 62.31% | 3.93 |
| | 16S.11 | C5 | 01.82 | 20.78% | 4.67 |

**Table 3** Results for PETS 2009 dataset (View-001). MAE, MRE and RMSE measures have been tabulated.

| | Sequence | Timestamp | Prediction Error | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | R0 | | | R1 | | | R2 | | |
| PETS 2009 | | | MAE | MRE | RMSE | MAE | MRE | RMSE | MAE | MRE | RMSE |
| | S1.L1 | 13-57 | 02.78 | 21.89% | 03.93 | 02.97 | 38.29% | 02.86 | 00.75 | 52.66% | 02.63 |
| | S1.L1 | 13-59 | 02.14 | 31.65% | 03.78 | 02.15 | 16.97% | 04.19 | 00.58 | 57.63% | 02.78 |
| | S1.L2 | 14-06 | 02.95 | 40.72% | 04.68 | 03.62 | 34.13% | 03.35 | 00.83 | 48.50% | 02.10 |
| | S1.L3 | 14-17 | - | - | - | 01.40 | 19.40% | 01.35 | - | - | - |

**Table 4** Results for UCSD Pedestrian Database dataset. MAE, MRE and RMSE measures have been tabulated.

| | Sequence | Prediction Error | | |
|---|---|---|---|---|
| | | MAE | MRE | RMSE |
| UCSD Pedestrian Database 2008 | vidf1_33_000.y | 01.34 | 42.26% | 05.38 |
| | vidf1_33_001.y | 00.63 | 13.32% | 04.54 |
| | vidf1_33_002.y | 00.54 | 13.28% | 04.41 |
| | vidf1_33_003.y | 01.15 | 22.50% | 04.01 |
| | vidf1_33_004.y | 01.76 | 34.76% | 05.02 |
| | vidf1_33_005.y | 02.66 | 14.63% | 03.63 |
| | vidf1_33_006.y | 02.29 | 23.64% | 04.14 |
| | vidf1_33_007.y | 02.97 | 12.25% | 03.74 |
| | vidf1_33_008.y | 00.47 | 07.25% | 02.87 |
| | vidf1_33_009.y | 00.87 | 24.11% | 04.95 |

tire sequence. Instead, we have provided the prediction error in terms of density that clearly highlights the trend of the system error. MCG dataset results have been presented in Online

**Table 5** Results for PETS 2009 dataset. In this table, results of the previous works presented in [75] has been tabulated for comparison in the regions R0, R1 and R2. For each sequence in the regions R0, R1 and R2, the best MAE results have been highlighted.

| Approach | Methods | Sequence | Timestamp | Prediction Error | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | R0 | | R1 | | R2 | |
| | | | | MAE | MRE | MAE | MRE | MAE | MRE |
| Background Modeling + Head Detection | Yao and Odobez [44] + Subburaman *et al.* [75] | S1.L1 | 13-57 | 06.17 | 35.00% | 02.27 | 21.00% | 02.95 | 36.00% |
| | | S1.L1 | 13-59 | **02.08** | 14.00% | **01.64** | 18.00% | 01.02 | 16.00% |
| | | S1.L2 | 14-06 | 03.01 | 13.00% | 02.35 | 14.00% | 01.42 | 10.00% |
| | | S1.L3 | 14-17 | - | - | 02.21 | 15.00% | - | - |
| Background Modeling + Head Detection | Barnich and Droogenbroeck [7] + Subburaman *et al.* [75] | S1.L1 | 13-57 | 05.95 | 30.00% | **01.90** | 29.00% | 02.50 | 32.00% |
| | | S1.L1 | 13-59 | **02.08** | 11.00% | 01.86 | 18.00% | 00.86 | 11.00% |
| | | S1.L2 | 14-06 | **02.40** | 12.00% | **01.40** | 10.00% | 01.70 | 01.70% |
| | | S1.L3 | 14-17 | - | - | 01.89 | 10.00% | - | - |
| Head Detection | Subburaman *et al.* [75] | S1.L1 | 13-57 | 04.80 | 25.00% | 02.40 | 32.00% | 03.40 | 42.00% |
| | | S1.L1 | 13-59 | 04.70 | 35.00% | 02.60 | 47.00% | 01.10 | 21.00% |
| | | S1.L2 | 14-06 | 05.00 | 60.00% | 03.10 | 74.00% | 03.30 | 23.70% |
| | | S1.L3 | 14-17 | - | - | 02.88 | 56.00% | - | - |
| Motion Estimation + Density Estimation | Proposed | S1.L1 | 13-57 | **02.78** | 21.89% | 02.97 | 38.29% | **00.75** | 52.66% |
| | | S1.L1 | 13-59 | 02.14 | 31.65% | 02.15 | 16.97% | **00.58** | 57.63% |
| | | S1.L2 | 14-06 | 02.95 | 40.72% | 03.62 | 34.13% | **00.83** | 48.50% |
| | | S1.L3 | 14-17 | - | - | **01.40** | 19.40% | - | - |

Resource Fig. 1 and Fig. 2. PETS 2009 results are depicted in Online Resource Figs. 3-6. UCSD dataset results are presented in Online Resource Fig. 7-10. The mean error increases as the density crosses 10 for MCG dataset, 15 for PETS 2009 dataset and 10 for UCSD dataset. The error is attributed to the heavy inter-object occlusions.

Various methods have been proposed by research communities on PETS 2009 dataset. The proposed method is compared with the results provided in [47,1,75], tested on PETS 2009 dataset, and [12] tested on UCSD dataset. Table 5 provides a comparison of previous work [75] and the proposed work. The table highlights the prediction error in terms of MAE and MRE for the regions R0, R1 and R2. Table 5 provides similar measures (MAE and MRE) compared to the others in regions R0 and R1. Since the region R2 consists of

**Table 6** Comparison of results for PETS 2009 dataset.In this table, results of previous works presented in [4, 17, 75] has been tabulated for comparison of individual sequences. For each sequence, the best MAE results have been highlighted.

| | Prediction Error | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Method | Albiol *et al.* [4] | | Conte *et al.* [17] | | Subburaman *et al.* [75] | | Proposed | |
| Learning Type | Supervised Approach | | Supervised Approach | | Supervised Approach | | Semi-supervised Approach | |
| Detection Type | People Detection | | People Detection | | Head Detection | | People Detection | |
| Sequence | MAE | MRE | MAE | MRE | MAE | MRE | MAE | MRE |
| S1.L1, 13-57 | 2.80 | 12.6% | **1.92** | 08.70% | 5.95 | 30% | 2.17 | 37.61% |
| S1.L1,13-59 | 3.86 | 24.9% | 2.24 | 17.30% | 2.08 | 11% | **1.62** | 35.41% |
| S1.L2, 14-06 | 5.14 | 26.1% | 4.66 | 20.50% | **2.40** | 12% | 2.47 | 41.11% |
| S1.L3,14-17 | 2.64 | 14.0% | 1.75 | 09.20% | 2.20 | 10% | **1.40** | 19.40% |

single-object cases for larger extents of time, the MRE measure is skewed. Hence, the MRE measure in R2 shows higher error rate. In [75], Subburaman *et al.*used gradient features and training for arriving at the results. The proposed approach use motion features for determination of density.

Table 6 provides a comparison of previous work [75] and the proposed work for individual sequences rather than the regions R0, R1 and R2. Results for [4] was obtained from [17] and [75]. For the purpose of comparison, we have averaged our obtained measures (MAE and MRE) in the regions R0, R1 and R2 to obtain the MAE and MRE for each sequence. Considering the supervised approaches [4, 17, 75], the proposed approach provides similar results and is the best in case of S1.L3 (Timestamp: 14-17) and S1.L1 (Timestamp: 13-59) with the MAE measures of 1.40 and 1.62. The proposed method performed better in 2 out of 4 sequences. In sequence S1.L1 (Timestamp: 13-57) people move from right to left, where people are partially occluded and boundary between groups are not clearly demarcated. In S1.L2 (Timestamp: 14-06) people move from right to left as a single group with occlusions. In contrast, in S1.L1 (Timestamp: 13-59), people move in groups from right to left such that groups as well as individuals are well separated and occlusions are very limited. In sequence S1.L3 (Timestamp: 14-17), people move across the scene from left to right while running.

During the running event, people are individually separated. The better performance of our purposed approach in 2 out of 4 cases compared to others is attributed to the fact that during limited occlusions, the proposed method performs better. Occlusion handling will be addressed in future for enhancing the system.

Noise induced at the low-level due to the nature of digital image formation can at many times introduce erroneous results in the high-level processing. The noise would generally be small variations in the pixel values in a given neighborhood. Though these variations are undetected by human perceptions (because our visual system is filtering these variations), these small variations will result in high-amplitude noise at higher-level processing in the digital image domain. As a result, filtering or reducing the noise at the low-level of video processing forms one of the key building blocks of the video analysis. Filtering in this context implies that reducing variations among perceptually similar colors and maintaining edges to determine the characteristic features. Bilateral filter is one such state-of-the-art filter that was first proposed by Aurich and Weule [6] as “non-Guassian” filter in 1995 and later reintroduced by Smith and Bardy [71] in 1997 and by Tomasi and Manduchi [76] in 1998. Tomasi and Manduchi [76] coined it as “bilateral” filter and remains the most popular edge-preserving even today. The biggest advantage of the the bilateral filtering is that it maintains colors and edges that are perceptually meaningful to humans, which mainly imitates human vision system. In other words, the filter maintains geometric and photometric closeness of neighborhood pixels. In the proposed, this has been applied on the input color frames. Application of bilateral filter to optical flow estimation can be found in [87, 68]. While preservation of edges are essential, presence of multiple object boundaries invite large number of edges that degrade the high-level processing in the case of crowd movements. On the other hand, Gaussian filter is a low-pass filter that eliminates high-frequency signals (edges) by averaging the pixels in the local neighborhood according 2D Gaussian function [76]. In the proposed approach, Gaussian filter has been added on top of bilateral filter to ensure that high-amplitude edges are suppressed. Thus the combination of filters yield smooth object surfaces and simultaneously, reduced high-amplitude edges for smooth motion estimation.

The proposed approach is suitable for sparse camera (single-view) networks where such networks form the essential components of surveillance systems [72]. Counting people using tracking may be accomplished in sparse camera networks or dense camera networks. In case of dense camera networks, multi-view invariant features assist to help resolve oc-

clusions with the combination of trackers. The challenges in sparse camera networks are that only single-view features are available and that the features must be invariant. Gradient-based optical flow, used in this work, is one of such invariant features. Furthermore, perspective normalization in dense camera networks provide accurate object sizes. In sparse camera networks, only camera calibration parameters are available. It must be noted that most of the existing methods apply weight to the features to compensate for perspective distortion. In these cases there is an assumption that features are available for all the detected objects. Additionally, it is assumed that features are consistent for both objects nearer to the camera and farther from the camera. Most of the Optical Character Recognition (OCR) systems and License Plate Recognition (LPR) systems employ perspective correction [77, 89]. However, the features available for the farthest object decreases with increase in perspective distance away from the camera [77]. Indeed, research indicates that perspective corrected images yield more accurate results. For instance, a study by Clark *et al.* [15] regarding visual image registration based using Scale-Invariant Feature Transform (SIFT), SURF and optical flow indicated that the two natural features, SIFT and SURF delivered poor performance. With the inclusion of optical flow and perspective correction, invariance properties were achieved delivering superior results. Therefore, detecting prominent features without correcting the perspective distortion will not yield accurate results, which has been the major trend in the existing approaches. In this work, we have explicit 2D-2D geometric mapping that compensates for *perspectivity*.

In literature, many methods employ the background modeling approach to segment the foreground objects and map the foreground pixels to the number of objects. Others extract the texture, color or intensity features and then use the training and classification using classifiers while assuming that the objects are available. Compared to these methods, in general, the proposed method uses motion estimation and contour analysis to extract the foreground objects without training. For instance, Chan *et al.* [12] use temporal subvolume of the video to extract features. A similar video-temporal subvolume-based approach was proposed using motion only features by Rao *et al.* [65], in which several frames were considered. An improvement in this work is that the density estimation is based on individual frames rather from several frames (except for motion features where two frames are required), as this was our primary goal of the proposed approach.

An analogy can be drawn between the crowd monitoring system shown in Fig. 2 and human visual system for density estimation. In human vision system, rods and cones act as

sensors in capturing the visual images. Rods are sensitive to low-light vision, provide coarse information and fast, whereas cones require minimum luminance, provide fine details and are slow in response. Ganglion cells aggregate information signals from rods (many-to-one mapping) and cones (one-to-one) for sending it to Lateral Geniculate Nucleus (LGN), which acts a carrier of signals for higher level processing. The signals from LGN are projected into visual cortex for object recognition, registration, and tracking. Edge, direction, spatial, temporal and chromaticity information are clubbed (logical AND/OR) and a decision is made at higher dimensional to accurately identify the objects invariance of scale, space and identity [19,81,48]. In the proposed approach, camera networks is tantamount to rods and cones; preprocessing filters to ganglion cells and LGN; people detection, motion tracking and density estimation are equivalent to functions of visual cortex.

Although the overall results are impressive, slow movements and shadows limits the effectiveness of the proposed algorithm. If the objects are moving slowly such that the optical flow is unable to determine the apparent motion, then the object detection, contour extraction and density estimation may not be accurate. One such scenario is when the people start to gather at a point and the people at the center do not move. In this scenario, only the exterior of the crowd has the movements while the interior of the crowd has zero movements. In those cases, use of area information from motion cues would be valuable to deduce the crowd activity. We used background modeling based on Mixture of Gaussians (MoG) [26, 74] to handle such scenarios. Furthermore, the dense optical flow was used to obtain smooth vectors globally. Sparse optical flow methods are limited by small movements and in case of articulated movements of people, dense flow handles articulated movements smoothly. The results of density estimation can be greatly improved with the incorporation of explicit shadow handling method. However, it should be noted that in occluded scenarios, formation of shadows are less likely and also the presence of shadows would impact less on counting because of occlusion. There are several instances where occlusion causes the number of foreground detected regions that do not match with the actual crowd density of the scene. In such cases tracking of the people coupled with density estimation correction would be an ideal solution. Again, either the full body detection or the head detection, and tracking depends on the number of distinguishable features and tracker efficiency. Most of the trackers assume that the heads of the people are visible. In case of UCSD dataset, even head detection may also become inefficient because of lack of features (small-sized objects) and the frames provided are purely grayscale.

The future work includes handling explicit static crowd movement, resolving ambiguity during occlusions without tracking and shadows during crowded scenarios. One possibility to handle crowd occlusions would be to adopt the concept of particle video as described by Sand and Teller [68]. However, the dynamics of crowd movements must be taken into account, which necessitates further study in this direction. Addition of texture information will add another dimension to addressing problems with static crowd movements. Whether the texture information provides any improvement in handling occlusions and shadows in crowded scenes need to be investigated. Additionally, further study is required whether texture information is necessary or to the degree to which it is redundant with respect to motion and vice versa, and how well can the combined features be applied to crowd monitoring.

## 7 Conclusion

Understanding the crowd behavior using an automated video analytics is a relevant research problem from a video surveillance perspective. Crowd density estimation in a video scene is necessary in understanding the crowd behavior. Most of the existing density estimation methods use the supervised training schemes to arrive at the density estimates. In this work, we proposed crowd density estimation based on clustering motion cues. The proposed approach results in accurate estimates of the crowd density. It delivered superior performance in 50% of cases on PETS 2009 dataset when compared with existing methods. While the maximum mean error of 3.62 was received for MCG and PETS datasets, it was 2.66 for UCSD dataset. The maximum mean error was found to be nominal in estimating crowd density.

## Acknowledgements

This work is partially supported by the Australian Research Council (ARC) linkage project LP100200430, partnering the University of Melbourne, Melbourne Cricket Club and ARUP. Authors would like to thank representatives and staff of ARUP and MCG.

## References

1. Acampora, G., Loia, V., Percannella, G., Vento, M.: Trainable estimators for indirect people counting: A comparative study. In: 2011 IEEE International Conference on Fuzzy Systems (FUZZ), pp. 139–145.

IEEE (2011)
2. Aggarwal, J.K., Cai, Q., Liao, W., Sabata, B.: Articulated and elastic non-rigid motion: a review. In: Proceedings of the 1994 IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 2–14. IEEE (1994)
3. Aijun, S., Mao, L., Jianfeng, L.: Real-time crowd massing risk supervision system based on massing crowd counting in public venue. In: 2009 International Symposium on Computer Network and Multimedia Technology (CNMT 2009), pp. 1–7. IEEE (2009)
4. Albiol, A., Silla, M.J., Albiol, A., Mossi, J.M.: Video analysis using corner motion statistics. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 31–38. IEEE (2009)
5. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision **2**(3), 283–310 (1989). DOI 10.1007/BF00158167
6. Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: Mustererkennung 1995, pp. 538–545. Springer (1995)
7. Barnich, O., Van Droogenbroeck, M.: Vibe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image Processing **20**(6), 1709–1724 (2011)
8. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. International Journal of Computer Vision **12**(1), 43–77 (1994)
9. Batur, A.U., Hayes, M.H.: Adaptive active appearance models. IEEE Transactions on Image Processing **14**(11), 1707–1721 (2005)
10. Bhaskar, H., Mihaylova, L., Maskell, S.: Articulated human body parts detection based on cluster background subtraction and foreground matching. Neurocomputing **100**(0), 58–73 (2013)
11. Bo, W., Nevatia, R.: Tracking of multiple, partially occluded humans based on static body part detection. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 951–958. IEEE (2006)
12. Chan, A., Liang, Z.S., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1–7. IEEE (2008). DOI 10.1109/CVPR.2008.4587569
13. Chan, A., Vasconcelos, N.: Pedestrian traffic database. http://www.svcl.ucsd.edu/projects/peoplecnt/ (2008). [Online; accessed 22-July-2013]
14. Chan, A.B., Vasconcelos, N.: Counting people with low-level features and bayesian regression. IEEE Transactions on Image Processing **21**(4), 2160–2177 (2012). DOI 10.1109/tip.2011.2172800
15. Clark, A., Green, R., Grant, R.: Perspective correction for improved visual registration using natural features. In: 23rd International Conference Image and Vision Computing New Zealand (IVCNZ 2008), pp. 1–6. IEEE (2008)
16. Conde, C., Moctezuma, D., Martn De Diego, I., Cabello, E.: Hogg: Gabor and hog-based human detection for surveillance in non-controlled environments. Neurocomputing **100**(0), 19–30 (2013)
17. Conte, D., Foggia, P., Percannella, G., Tufano, F., Vento, M.: A method for counting people in crowded scenes. In: 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 225–232. IEEE (2010)
18. Da Xu, R.Y., Kemp, M.: Multiple curvature based approach to human upper body parts detection with connected ellipse model fine-tuning. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 2577–2580. IEEE (2009)

19. DiCarlo, J.J., Zoccolan, D., Rust, N.C.: How does the brain solve visual object recognition? Neuron **73**(3), 415–434 (2012)
20. Duc Thanh, N., Wanqing, L., Ogunbona, P.: A part-based template matching method for multi-view human detection. In: 24th International Conference Image and Vision Computing New Zealand (IVCNZ '09), pp. 357–362. IEEE (2009)
21. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: D. Vernon (ed.) Computer Vision ECCV 2000, *Lecture Notes in Computer Science*, vol. 1843, pp. 751–767. Springer Berlin / Heidelberg (2000)
22. Ferryman, J.: PETS 2009 Benchmark Data (2009). URL http://www.cvg.rdg.ac.uk/PETS2009/a.html
23. Fleet, D., Jepson, A.: Computation of component image velocity from local phase information. International Journal of Computer Vision **5**(1), 77–104 (1990)
24. Gavrila, D.M.: A bayesian, exemplar-based approach to hierarchical shape matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(8), 1408–1421 (2007)
25. Gonzalez, R., Woods, R.: Digital Image Processing (3rd Edition). Prentice Hall (2007)
26. Grimson, W.E.L., Stauffer, C., Romano, R., Lee, L.: Using adaptive tracking to classify and monitor activities in a site. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 22–29. IEEE (1998)
27. Haritaoglu, I., Harwood, D., Davis, L.S.: $W^4$: Who? when? where? what? a real time system for detecting and tracking people. In: Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 222–227. IEEE (1998)
28. Haritaoglu, I., Harwood, D., Davis, L.S.: A fast background scene modeling and maintenance for outdoor surveillance. In: Proceedings of the 15th International Conference on Pattern Recognition, vol. 4, pp. 179–183. IEEE (2000)
29. Haritaoglu, I., Harwood, D., Davis, L.S.: $W^4$: real-time surveillance of people and their activities. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 809–830 (2000)
30. Hartley, R., Kang, S.B.: Parameter-free radial distortion correction with center of distortion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(8), 1309–1321 (2007)
31. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)
32. Heeger, D.: Optical flow using spatiotemporal filters. International Journal of Computer Vision **1**(4), 279–302 (1988)
33. Heeger, D.J.: Model for the extraction of image flow. Journal of Optical Society of America A **4**(8), 1455–1471 (1987)
34. Horn, B.K., Schunck, B.G.: Determining optical flow. Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA (1980)
35. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17**, 185–203 (1981)
36. Hosub, Y., Dohyung, K., Suyoung, C., Youngjo, C.: A robust human head detection method for human tracking. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4558–4563. IEEE (2006)
37. li Hou, Y., Pang, G.K.H.: Automated people counting at a mass site. In: IEEE International Conference on Automation and Logistics (ICAL 2008), pp. 464–469. IEEE (2008)

38. Hsu, W.L., Lin, K.F., Tsai, C.L.: Crowd density estimation based on frequency analysis. In: 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 348–351. IEEE (2011)
39. Huazhong, X., Pei, L., Lei, M.: A people counting system based on head-shoulder detection and tracking in surveillance video. In: 2010 International Conference on Computer Design and Applications (ICCDA), vol. 1, pp. V1–394–V1–398. IEEE (2010)
40. Ishii, Y., Hongo, H., Yamamoto, K., Niwa, Y.: Real-time face and head detection using four directional features. In: In Proc. Sixth IEEE International Conference onAutomatic Face and Gesture Recognition, pp. 403–408. IEEE (2004)
41. Jacques Junior, J.C.S., Musse, S.R., Jung, C.R.: Crowd analysis using computer vision techniques. IEEE Signal Processing Magazine **27**(5), 66–77 (2010)
42. Jain, R., Nagel, H.H.: On the analysis of accumulative difference pictures from image sequences of real world scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-1**(2), 206–214 (1979)
43. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(10), 1296–1311 (2003)
44. Jian, Y., Odobez, J.M.: Multi-layer background subtraction based on color and texture. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07), pp. 1–8. IEEE (2007)
45. Kambhamettu, C., Goldgof, D.B., Terzopoulos, D., Huang, T.S.: Nonrigid motion analysis. In: T.Y. Young (ed.) Handbook of pattern recognition and image processing (vol. 2), pp. 405–430. Academic Press, Inc., Orlando, FL, USA (1994)
46. Kilger, M.: A shadow handler in a video-based real-time traffic monitoring system. In: Proceedings of IEEE Workshop on Applications of Computer Vision, pp. 11–18. IEEE (1992)
47. Krahnstoever, N., Yu, T., Patwardhan, K.A., Gao, D.: Multi-camera person tracking in crowded environments. In: 2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-Winter), pp. 1–7. IEEE (2009)
48. Leibo, J.Z., Mutch, J., Poggio, T.: Why the brain separates face recognition from object recognition. In: Advances in Neural Information Processing Systems, pp. 711–719 (2011)
49. Lu, W., Yung, N.H.C.: Three-dimensional model-based human detection in crowded scenes. IEEE Transactions on Intelligent Transportation Systems **13**(2), 691–703 (2012)
50. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th international joint conference on Artificial intelligence, pp. 674–679. Morgan Kaufmann Publishers Inc. (1981)
51. Ma, L., Chen, Y., Moore, K.L.: Rational radial distortion models of camera lenses with analytical solution for distortion correction. International Journal of Information Acquisition **1**(02), 135–147 (2004)
52. Ma, W., Huang, L., Liu, C.: Crowd density analysis using co-occurrence texture features. In: 2010 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), pp. 170–175. IEEE (2010)
53. Ma, Y., Bai, G.: Short term prediction of crowd density using v-svr. In: 2010 IEEE Youth Conference on Information Computing and Telecommunications (YC-ICT), pp. 234–237. IEEE (2010)

54. Mao, Y., Tong, J., Xiang, W.: Estimation of crowd density using multi-local features and regression. In: 2010 8th World Congress on Intelligent Control and Automation (WCICA), pp. 6295–6300. IEEE (2010)
55. Merad, D., Aziz, K.E., Thome, N.: Fast people counting using head detection from skeleton graph. In: 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 151–156. IEEE (2010)
56. Milan, A.: PETS 2009 - ground truth (2011). http://www.gris.informatik.tu-darmstadt.de/ aandriye/data.html
57. Min, L., Zhaoxiang, Z., Kaiqi, H., Tieniu, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 19th International Conference on Pattern Recognition, pp. 1–4. IEEE (2008)
58. Min, L., Zhaoxiang, Z., Kaiqi, H., Tieniu, T.: Rapid and robust human detection and tracking based on omega-shape features. In: 16th IEEE International Conference on Image Processing (ICIP), pp. 2545–2548. IEEE (2009)
59. Nagel, H.H.: On the estimation of optical flow: Relations between different approaches and some new results. Artificial intelligence **33**(3), 299–324 (1987)
60. Oliver, N.M., Rosario, B., Pentland, A.P.: A bayesian computer vision system for modeling human interactions. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 831–843 (2000)
61. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian detection using wavelet templates. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 193–199. IEEE (1997)
62. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics **9**(1), 62–66 (1979)
63. Park, J., Byun, S.C., Lee, B.U.: Lens distortion correction using ideal image coordinates. IEEE Transactions on Consumer Electronics **55**(3), 987–991 (2009)
64. Rahmalan, H., Nixon, M.S., Carter, J.N.: On crowd density estimation for surveillance. In: The Institution of Engineering and Technology Conference on Crime and Security, pp. 540–545. IEEE (2006)
65. Rao, A.S., Gubbi, J., Marusic, S., Stanley, P., Palaniswami, M.: Crowd density estimation based on optical flow and hierarchical clustering. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI 2013), pp. 494–499. IEEE (2013)
66. Rodriguez, M., Laptev, I., Sivic, J., Audibert, J.Y.: Density-aware person detection and tracking in crowds. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2423–2430. IEEE (2011)
67. Salti, S., Cavallaro, A., Di Stefano, L.: Adaptive appearance modeling for video tracking: Survey and evaluation. IEEE Transactions on Image Processing **21**(10), 4334–4348 (2012)
68. Sand, P., Teller, S.: Particle video: Long-range motion estimation using point trajectories. International Journal of Computer Vision **80**(1), 72–91 (2008)
69. Seki, M., Wada, T., Fujiwara, H., Sumi, K.: Background subtraction based on cooccurrence of image variations. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II–65–II–72. IEEE (2003)

70. Singh, A., Allen, P.: Image-flow computation: An estimation-theoretic framework and a unified perspective. CVGIP: Image understanding **56**(2), 152–177 (1992)
71. Smith, S.M., Brady, J.M.: Susana new approach to low level image processing. International journal of computer vision **23**(1), 45–78 (1997)
72. Song, M., Tao, D., Maybank, S.J.: Sparse camera network for visual surveillance – a comprehensive survey. Computing Research Repository **abs/1302.0446** (2013)
73. Srivastava, S., Ng, K.K., Delp, E.J.: Crowd flow estimation using multiple visual features for scenes with changing crowd densities. In: 2011 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 60–65. IEEE (2011)
74. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252. IEEE (1999)
75. Subburaman, V., Descamps, A., Carincotte, C.: Counting people in the crowd using a generic head detector. In: 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 470–475. IEEE (2012). DOI 10.1109/AVSS.2012.87
76. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, pp. 839–846. IEEE (1998)
77. Tommy, R., Mohan, S.: An approach for fully automating perspective images based on symmetry and line intersection. In: 2011 International Conference on Image Information Processing (ICIIP), pp. 1–5. IEEE (2011)
78. Tosato, D., Farenzena, M., Cristani, M., Murino, V.: Part-based human detection on riemannian manifolds. In: 2010 17th IEEE International Conference on Image Processing (ICIP), pp. 3469–3472. IEEE (2010)
79. Uras, S., Girosi, F., Verri, A., Torre, V.: A computational approach to motion perception. Biological Cybernetics **60**(2), 79–87 (1988). DOI 10.1007/BF00202895
80. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. The Visual Computer **29**(10), 983–1009 (2013). DOI 10.1007/s00371-012-0752-6. URL http://dx.doi.org/10.1007/s00371-012-0752-6
81. Wang, R.: The primary visual cortex (straite cortex, v1). http://fourier.eng.hmc.edu/e180/lectures/v1/node1.html (2013). [Online; accessed 09-August-2013]
82. Waxman, A., Wu, J., Bergholm, F.: Convected activation profiles and the measurement of visual motion. In: Computer Society Conference on Computer Vision and Pattern Recognition, pp. 717–723. IEEE (1988)
83. Willick, D., Yang, Y.H.: Experimental evaluation of motion constraint equations. CVGIP: Image Understanding **54**(2), 206–214 (1991)
84. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate Research **30**(1), 79–82 (2005). DOI 10.3354/cr030079
85. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: real-time tracking of the human body. In: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 51–56 (1996)

86. Wu, X., Liang, G., Lee, K.K., Xu, Y.: Crowd density estimation using texture analysis and learning. In: IEEE International Conference on Robotics and Biomimetics (ROBIO '06), pp. 214–219. IEEE (2006)
87. Xiao, J., Cheng, H., Sawhney, H., Rao, C., Isnardi, M.: Bilateral filtering-based optical flow estimation with occlusion detection. In: Computer Vision–ECCV 2006, pp. 211–224. Springer (2006)
88. Xinbo, G., Ya, S., Xuelong, L., Dacheng, T.: A review of active appearance models. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **40**(2), 145–158 (2010)
89. Yang, S.J., Ho, C.C., Chen, J.Y., Chang, C.Y.: Practical homography-based perspective correction method for license plate recognition. In: 2012 International Conference on Information Security and Intelligence Control (ISIC), pp. 198–201. IEEE (2012)
90. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys **38**(4), 1–45 (2006)
91. Yu-Ting, C., Chu-Song, C., Yi-Ping, H., Kuang-Yu, C.: Multi-class multi-instance boosting for part-based human detection. In: 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1177–1184. IEEE (2009)
92. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: Crowd analysis: a survey. Machine Vision and Applications **19**(5-6), 345–357 (2008)
93. Zhao, T., Nevatia, R.: Bayesian human segmentation in crowded situations. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II–459–66. IEEE (2003)
94. Zhao, T., Nevatia, R.: Tracking multiple humans in complex situations. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(9), 1208–1221 (2004). DOI 10.1109/TPAMI.2004.73
95. Zhao, T., Nevatia, R., Lv, F.: Segmentation and tracking of multiple humans in complex situations. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II–194–II–201. IEEE (2001)
96. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(7), 1198–1211 (2008)
97. Zhe, L., Davis, L.S.: Shape-based human detection and segmentation via hierarchical part-template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(4), 604–618 (2010)
98. Zhou, J., Hoang, J.: Real time robust human detection and tracking system. In: Proceedings of the IEEE Computer Society Conference onComputer Vision and Pattern Recognition - Workshops, pp. 149–149. IEEE (2005)