Multimed Tools Appl (2017) 76:23777–23804
DOI 10.1007/s11042-016-4156-x

A robust multimedia surveillance system for people counting

Zeyad Q. H. Al-Zaydi[1,2] · **David L. Ndzi**[1] · **Munirah L. Kamarudin**[3] · **Ammar Zakaria**[4] · **Ali Y. M. Shakaff**[4]

Received: 12 May 2016 / Revised: 7 October 2016 / Accepted: 14 November 2016 /
Published online: 23 November 2016

Abstract Closed circuit television cameras (CCTV) are widely used in monitoring. This paper presents an intelligent CCTV crowd counting system based on two algorithms that estimate the density of each pixel in each frame and use it as a basis for counting people. One algorithm uses scale-invariant feature transform (SIFT) features and clustering to represent pixels of frames (SIFT algorithm) and the other uses features from accelerated segment test (FAST) corner points with SIFT features (SIFT-FAST algorithm). Each algorithm is designed using a novel combination of pixel-wise, motion-region, grid map, background segmentation using Gaussian mixture model (GMM) and edge detection. A fusion technique is proposed and used to validate the accuracy by

Highlights
•Two people counting algorithms based on CCTV cameras are proposed.
•Training error, set-up time and cost have been reduced by the proposed system.
•Motion edges, grid map, pixel-wise and fusion techniques are used in the proposed algorithms.
•Two indoor and outdoor datasets are used for evaluation.
•The accuracy of the proposed system is, at least, comparable with the state of the art methods.

✉ Zeyad Q. H. Al-Zaydi
zeyad.al-zaydi@port.ac.uk

David L. Ndzi
david.ndzi@port.ac.uk

Munirah L. Kamarudin
latifahmunirah@unimap.edu.my

Ammar Zakaria
ammarzakaria@unimap.edu.my

Ali Y. M. Shakaff
aliyeon@unimap.edu.my

[1] School of Engineering, University of Portsmouth, Portsmouth PO1 3DJ, UK

[2] Computer Centre, University of Technology, Baghdad, Iraq

[3] School of Computer and Communication Engineering, University Malaysia Perlis, Perlis, Malaysia

[4] School of Mechatronic Engineering, University Malaysia Perlis, Perlis, Malaysia

combining the result of the algorithms at frame level. The proposed system is more practical than the state of the art regression methods because it is trained with a small number of frames so it is relatively easy to deploy. In addition, it reduces the training error, set-up time, cost and open the door to develop more accurate people detection methods. The University of California (UCSD) and Mall datasets have been used to test the proposed algorithms. The mean deviation error, mean squared error and the mean absolute error of the proposed system are less than 0.1, 16.5 and 3.1, respectively, for the Mall dataset and less than 0.07, 5.5 and 1.9, respectively, for UCSD dataset.

Keywords Crowd counting systems · Monitoring · CCTV cameras · Background segmentation

1 Introduction

Closed circuit television cameras (CCTV) have already become ubiquitous and their use is growing exponentially. For instance, 4.2 million CCTV cameras were used in the United Kingdom [55] in 2004 and an estimated up to around 5.9 million in 2015 [79]. People counting systems are one of the most challenging systems in computer vision to implement [8, 35, 36, 50, 59, 63, 76]. People counting is a useful task for safety, security and operational purposes and can be important for improving awareness [50, 51, 59, 65, 70]. The number of people in a given space can be used to develop business intelligence, such as improving location of products within a shop and finding the number of visitors [51, 65, 70]. Crowd management [70], transport [39] and staff planning applications can be improved by using this kind of information. Heating, lighting and air conditioning can also be optimised using people counting and distribution information to enhance energy management [34, 74], or to improve emergency evacuation plan [74].

A significant amount of research has been carried out to find an accurate computer vision solution but there are still many challenges that need to be resolved. These include occlusions, varying lighting, long processing time and improving the accuracy in image processing [2, 36, 50, 63].

This work distinguishes itself with the following four main contributions. First, a new combination of SIFT and FAST features with pixel-wise technique is used to improve the accuracy. Second, motion edge pixels are used instead of foreground pixels to reduce the number of SIFT descriptors required. Third, a combination of grid map and pixel-wise technique is used to improve the cluster classification in frames which enables similar clusters in different cells to be assigned different densities depending on their location in the frame. Fourth, the algorithms are comprehensively tested and validated using two datasets, the University of California, San Diego (UCSD) and Mall datasets [13, 16].

2 Related work

Crowd counting can be classified into four categories; crowd counting based on detection, clustering, regression and optimisation.

2.1 People detection based algorithms

Detection based algorithms start by detecting people individually and then counting them [2]. The detection process depends either on the person's entire body or parts of the body such as face, head or head-shoulder [2]. The main advantages of these algorithms are that they count people and find their locations as well, therefore they are useful in people tracking [35]. The main disadvantages of these algorithms are that they are severely affected by varying lighting, occlusion and have long processing times [72]. They achieve good results in sparsely populated scenarios, whereas in crowded scenarios, the accuracy decreases significantly [35]. In addition, a high resolution camera is required to obtain a good accuracy [35]. Triggs and Dalal proposed histogram of oriented gradients (HOG) method thereby creating a basis for the development of a fast appearance-based detection algorithm [25]. Many improvements of the HOG technique have been proposed. One of the most promising variants is the fastest pedestrian detection in the west (FPDW) which has significantly increased the speed of detection [27].

Pedestrian detection is constrained to horizontal or vertical camera angles. In people counting, horizontal camera angles can be used but a vertical or downward facing angle is often preferred to minimise occlusions [67]. A majority of commercial people counting products are cameras that are placed on the ceiling pointing downwards to get the best view. However, this is not an optimal set-up if the detection area needs to be maximized.

People detection based algorithms can be classified into six categories: full body detection [25, 46, 73]; part body detection [28, 49, 77]; 3D camera detection [33]; shape matching detection, where ellipse and Bernoulli shapes are used to identify and count people in each blob [31, 48]; multi-camera detection, which is used to avoid occlusion [53]; and density-aware detection, which is used to reduce the false positive per image (FPPI) in low crowd density locations and decreases the miss rate in high crowd density locations in the frames [58].

2.2 Features trajectories clustering based algorithms

Clustering based algorithms track visual features over time and the feature trajectories are then clustered into unique tracks using temporal, spatial and other factors [9, 18, 56, 71]. The number of clusters is the estimated number of people [54]. Different approaches have been used to study the similarities between trajectories such as Dynamic Time Warping (DTW). DTW is a time series method that is widely used to measure similarities between two temporal sequences [4, 6]. Kanade–Lucas–Tomasi (KLT) feature-matching algorithm is sometimes used to find the trajectories of features [56]. The advantages of clustering based algorithms are that they can decrease the occlusion and the angle of the camera effects [75]. However, their accuracies significantly decrease in highly crowded environments with cluttered background and heavy occlusion. A complicated trajectory management technique is required to assess the similarities of trajectories with different lengths, which is another limitation of these algorithms [64]. In addition, errors in the number of people due to the cohesiveness of features that belong to different people also affect their accuracies [64]. These algorithms also require high video frame rate to work well because motion information can reliably be extracted [16]. Features trajectory clustering algorithms can be used to count people but it is difficult to use them in a real-time environments due to their long processing times.

2.3 Low-level features regression based algorithms

Regression based algorithms usually consist of three steps, starting with a background segmentation that is used on a frame by frame basis to detect the foreground information. Low-level features are then extracted from the foreground such as edge features [16, 19–21, 60], segment features [11–14, 19–21, 35, 81], texture features [15, 17, 19, 78] and keypoints [24]. A regression function is then trained using these features to find the relationship between the number of people and the extracted features which is then used to estimate the number of people [71]. Various types of regression functions have been used such as support vector machine tree [23, 78], linear [26, 52], neural networks [19–21, 60] and Gaussian process algorithms [11–14, 54]. A significant amount of research has been carried out to improve these algorithms by varying the number of features. Some other researchers have tried to improve them by using more than one regression function and then choose the best fitting features [29]. The main advantages of these algorithms are that the accuracy is higher than feature trajectory clustering and detection based algorithm in crowded scenarios, and the computational time is shorter [16, 29, 72]. Their main disadvantage is that different training datasets are required with different environments or camera set-ups [71]. Some new contributions have also been presented to improve their accuracies, handling occlusions and adapt to new environments. Recent technique in crowd counting has been tested using static pictures from crowded environments [37]. A deep-learning approach that uses convolutional neural networks to predict the number of people has been proposed in that technique. Occlussion is another problem that a new proposed technique tries to minimise [3]. Research in [3] takes occlusion into account by using two regression functions, one for the low occlusion frames and the second for the high occlusion frames. In addition, adaptive combination of features is used in each environment according to their nature. Statistical features have been used by Hafeezallah et al. [32] to train a neural network to develope a highly accurate crowd counting algorithm. The differences of the sequential frames with curvelet transform has been proposed by [32] to improve the accuracy. A random projection forest, as a regression function, has also been proposed by other researchres to increase the maximum number of features that is used for training [80]. A small number of features can be handled by traditional regression functions which can negatively affect the performances of crowd counting systems. Aravinda et al. [57] have proposed a combination of optical flow for motion cues and hierarchical clustering to estimate the crowd density. Hierarchical clustering have been in [57] used to isolate distinct pixels that correspond to different people in the frame. Multi-cameras knowledge transfer technique has been used by Nick et al. [69] to provide different views of the crowd which are used to minimise occlusion and improve performance. The main disadvantage of the technique is the long set-up time required and the high cost of the hardware. Finally, a quadratic programming technique is used with a regression function and network flow constraints to improve the accuracy of estimating the number of people [30]**.** They take into account the temporal domain of a series of frames to improve the accuracy. Regression based algorithms are classified into three categories; holistic, histograms (intermediate) and local algorithms [62].

Holistic algorithms use global image features and one regression function for the whole frame [11–14, 19–21]. The types of features that used by these algorithms include foreground, edge , keypoints and texture features. A limitation of these algorithms is that they apply one global regression function over the whole image thereby not taking into account the high variability of crowd distribution, behaviour and density in different regions of the image [62].

Histogram features are used by histograms algorithms such as, edge orientation histogram, blob size histogram and histogram of oriented gradients (HOG) [44, 45]. One global regression function is trained by these features to find the estimated number of people. These algorithms use histogram bin magnitude and edge direction to avoid noise and to distinguish people, respectively. Histograms algorithms also ignore high variations in crowd behaviour, distribution and density in different regions of the image [62].

Local algorithms count the number of people by partitioning the frame into several regions and one local regression function is trained for each region to count the total number of people in the whole frame. The regions can be cells having regular or irregular sizes [16] or the regions can be foreground blobs and the total number of people is counted by summing the numbers in all regions [10, 22, 40, 43, 60].

2.4 Pixel-wise optimisation based algorithms

Some researchers use pixel-wise techniques to estimate the number of people [47]. In this approach, the density of each pixel is found and then integrated over the whole frame to estimate the total number of people [47]. Optimisation is used instead of regression to train crowd counting systems. This approach can be used to improve people detection algorithms by combining it with full or part body detection based algorithms [58]. Full body, head and head-shoulder detection based algorithms can be improved and the accuracy can be increased by using the density of pixels [58]. The aim of this combination is to reduce the false positive per image (FPPI) in low crowd density locations in the frames which happens when it inaccurately detects the presence of people when there is actually nobody. In addition, this approach decreases the miss rate in high crowd density locations in the frames.

Pixel-wise optimisation based algorithms can be trained using a small number of frames in comparison to regression based algorithms [47]. As a consequence, the set-up time of the system can be reduced by more than 25% in comparison to regression based algorithms which lead also to low set-up cost. Using a large number of training frames can negatively affect the accuracy of the training because manually annotation is an error-prone task.

3 System design

In this paper, the proposed system depends on supervised learning to estimate the number of people. The training frames are annotated and Gaussian representation is used to represent people. Quadratic programming is used for learning and maximum excess over subarrays distance ($\mathrm{D_{MESA}}$) is used to measure the difference between the true and predicted count which represent the loss function as given by Eq. (2).

The proposed system assumes that each pixel (p) in a frame is represented by a SIFT or SIFT-FAST feature vector. The density function of each pixel is represented as a linear transformation of the pixel representation ($\mathrm{x_p}$) as given by Eq. (1);

$$F(p) = \mathrm{w^T x_p} \tag{1}$$

Where $\mathrm{w^T}$ is the weight of each pixel in the frame. At the learning stage, a training frames set with their ground truth (true count) are used to find the correct weight ($\mathrm{w^T}$) of each pixel. Then the densities of all pixels in the frame are summed to find the predicted count. $\mathrm{D_{MESA}}$ is used to compare between the predicted count and true count as a loss function. $\mathrm{D_{MESA}}$ is defined as [58];

$$D_{\mathrm{MESA}}(F1, F2) = max\left|\sum\nolimits_{p\in B} F1(p) - \sum\nolimits_{p\in B} F2(p)\right| \tag{2}$$

Where $F1(p)$ and $F2(p)$ are the predicted count and true count of people in a frame. $\mathrm{D_{MESA}}$ is chosen for the proposed system because it is not significantly affected by jitter and noise but it has a strong relationship with the number and positions of people [47]. The ultimate goal of the learning stage is to find the best weight for each pixel that minimises the sum of the errors between the true counts and the predicted counts (the loss function) [47];

$$w = \mathrm{argmin}_w\left(w^T w + \gamma \sum\nolimits_{i=1}^{N} D_{\mathrm{MESA}}\right) \tag{3}$$

Where γ is a scalar parameter to control the regularization strength, $argmin_w$ represents the best weight that minimises the D_{MESA}. Quadratic programming can be used to solve Eq. (3) by using;

$$\min_{w,\xi_1,\ldots\ldots,\xi_N}\left(w^T w+\gamma\sum\nolimits_{i=1}^{N}\xi_i\right) \tag{4}$$

Subject to;

$$\xi_i\geq\sum\nolimits_{p\in B}(F1(p)-F2(p)),\quad \xi_i\geq\sum\nolimits_{p\in B}(F2(p)-F1(p)) \tag{5}$$

Where ξ_i are the auxiliary variables of training frames. Quadratic programming uses iterations to optimise the results and find the best weight (w^T) of each pixel. The iterations terminate when the right side of equation (5) is within ($\xi_i + \beta$) factor. β is a small constant ($\beta << 1$). It uses to decrease the number of iterations and faster convergence. Choosing β equal to 0 solves the equations (4) and (5) exactly. However, the convergence will finish faster if β is chosen to a very small value and that will not affect the performance of training [47]. In the experiments of the proposed system, β has been chosen to be equal to 0.001. The flow diagram of the proposed system is illustrated in the Fig. 1. It consists of two counting algorithms, one video source and one fusion model.

3.1 Algorithm 1: SIFT features algorithm

This algorithm combines the following techniques to count the number of people; motion edges, SIFT descriptors, gird map and pixel-wise techniques. This combination that is used to find the density of each pixel, is novel. Edge pixels are used because their number is less than foreground pixels. As a consequence, the required time to find the SIFT descriptors and cluster them in a frame will be significantly reduced which makes the proposed system faster than other people counting techniques based on D_{MESA} optimisation. There is a high correlation between SIFT descriptors and the number of people. This is difficult for quadratic programming to be used to find the density for a large number of SIFT descriptors (equal to the number of edge motion pixels). To solve this problem, clustering is used to reduce the number of SIFT descriptors to 256 clusters. The main disadvantage of using clustering is that many SIFT descriptors can be grouped into one cluster to reduce the problem space but they represent different densities. Grid map is used to improve the cluster classification in the frames which enables similar clusters in different cells to be assigned different densities depending on their location in the frame. The proposed algorithm can better adapt to high variations in crowd behaviours, distributions and densities. As a result, the accuracy is improved. Figure 2 shows the flow diagram of this algorithm. The procedure of the algorithm is illustrated in the following steps;

1- Implement Gaussian mixture model (GMM) to find the foreground information of the frame.

$$F_{GMM}=GMM(i,j) \tag{6}$$

Where F_{GMM} is the foreground pixels of the frame and $GMM(i,j)$ is the Gaussian mixture model of each pixel of the frame.

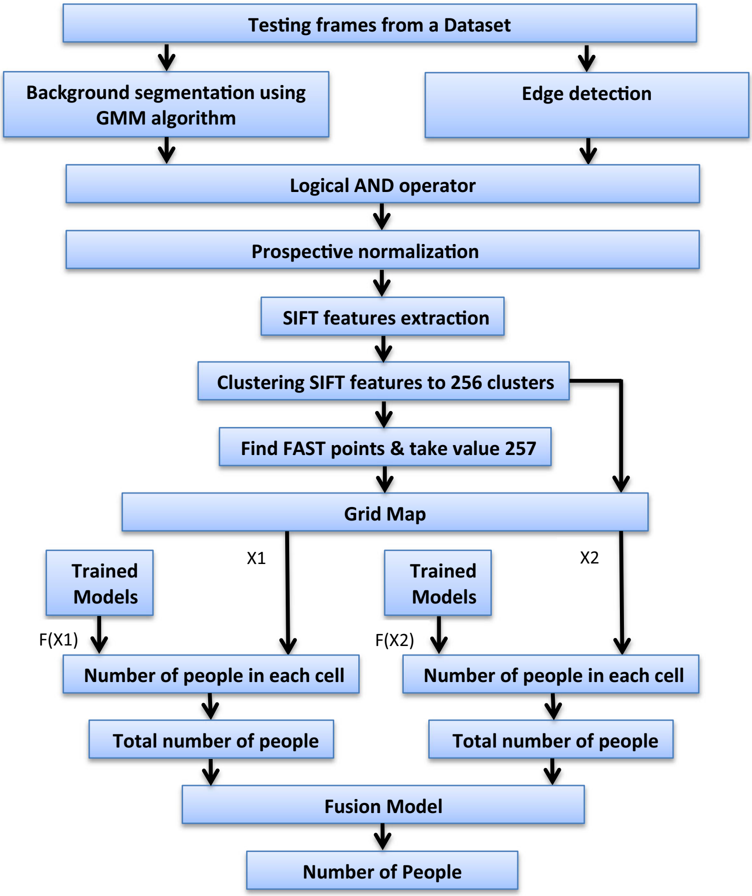

Testing frames from a Dataset
Background segmentation using GMM algorithm
Edge detection
Logical AND operator
Prospective normalization
SIFT features extraction
Clustering SIFT features to 256 clusters
Find FAST points & take value 257
Grid Map
Trained Models
X1
F(X1)
Trained Models
X2
F(X2)
Number of people in each cell
Number of people in each cell
Total number of people
Total number of people
Fusion Model
Number of People

Fig. 1 Flow diagram of the proposed system

2- Implement edge detection to find the edges of the frame.

$$F_{Edge} = E(i,j) \tag{7}$$

Where F_{Edge} is the edge of the frame and $E(i,j)$ is the detected edge of each pixel of the frame.

3- Perform logical (AND) operation between the foreground pixels of the frame and the detected edge to find the motion edge of the frame.

$$F_{motion\ edge} = F_{GMM}\ (i,j)\&\&F_{\mathrm{Edge}}\ (i,j) \tag{8}$$

Where $F_{motion\ edge}$ is the motion edge for the frame.

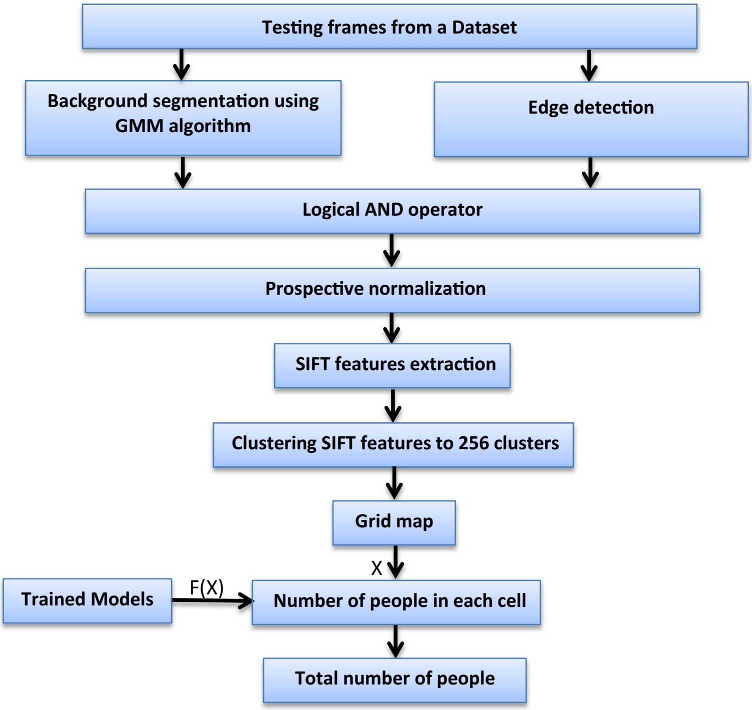

Testing frames from a Dataset
Background segmentation using GMM algorithm
Edge detection
Logical AND operator
Prospective normalization
SIFT features extraction
Clustering SIFT features to 256 clusters
Grid map
X
Trained Models
F(X)
Number of people in each cell
Total number of people

Fig. 2 Flow diagram of the SIFT Features algorithm

4- The pixels in each line of the frame are assigned different weight as a perspective normalization.

5- Find the SIFT descriptor for each motion edge pixel. Then, cluster the SIFT descriptors to 256 clusters. The centres of SIFT features are used as criteria for clustering them.

$$F_{SIFT} = SIFT(i,j) \qquad (i,j)\in \text{motion edge} \tag{9}$$

$$F_{Cluster} = Cluster(F_{SIFT}) \qquad (i,j)\in \text{motionedge} \tag{10}$$

Where F_{SIFT} is the SIFT descriptors of the frame and $F_{Cluster}$ is the SIFT descriptors clustering.

6- Divided the frames into cells (as a grid map) and count the number of people in each cell.

$$F_{Grid} = \sum\nolimits_{n} C_n \tag{11}$$

Where F_{Grid} is the grid map of each frame, C is a cell in the grid map and n is the number of cells in the grid map. Four cells configuration has been used in the proposed system which gives the best accuracies experimentally.

7- Use a quadratic programming (Interior-point-convex algorithm) to find the density of each cluster in each cell.
8- Integrate the densities of pixels over each cell to find the number of people in each cell.

$$N_{cell} = \sum_{(i,j)\in B_n} P_{density}(i,j) \tag{12}$$

Where N_{cell} is the number of people in each cell and $P_{density}(i,j)$ is the density of each pixel that belongs to this cell.

9- The summation of the number of people in all cells represents the total number of people in the frame.

$$N_{total} = \sum\nolimits_{n} N_{cell} \tag{13}$$

Where N_{total} is the total number of people in a frame and n is the number of cells.

3.2 Algorithm 2: SIFT-FAST features algorithm

This algorithm uses two features; FAST and SIFT. This algorithm combines the following techniques to count the number of people; motion edges, grid map, SIFT & FAST features and pixel-wise techniques. Edge pixels are used because their number is less than those of foreground pixels. The same approach as for SIFT feature algorithm described in Section 3.1 is used. However, FAST corner points are used to improve the accuracy due to the high correlation between the number of people and FAST corner points. The algorithm can also better adapt to high variations due to crowd behaviours, distribution and density. Figure 3 shows the flow diagram of the algorithm. Steps 1 to 5 are the same as for SIFT feature algorithm and descriptions from step 6 are as follows:

6- Find FAST points in each frame within the motion region.

$$F_{FAST} = F(i,j) \qquad\qquad (i,j)\in\text{motion regions} \tag{14}$$

Where F_{FAST} is the FAST corner points of a frame.

7- All pixels that are FAST corner points are assigned the value 257 so that quadratic programming can be used to find 257 density values instead of 256.
8- Divide the frame into cells (as a grid map) and the number of people in each cell is counted individually.

$$F_{Grid} = \sum\nolimits_{n} C_n \tag{15}$$

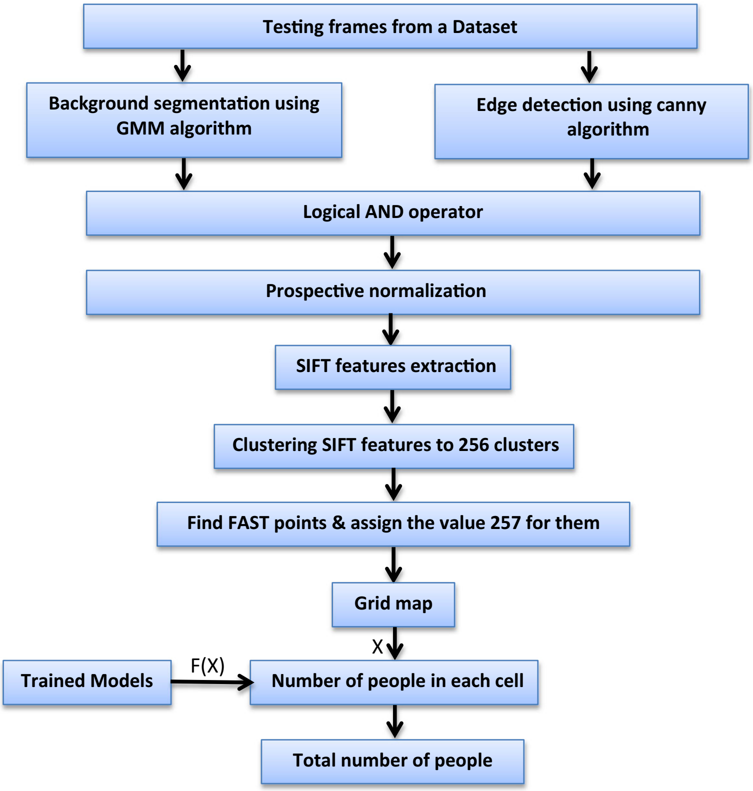

Testing frames from a Dataset
Background segmentation using GMM algorithm
Edge detection using canny algorithm
Logical AND operator
Prospective normalization
SIFT features extraction
Clustering SIFT features to 256 clusters
Find FAST points & assign the value 257 for them
Grid map
X
Trained Models
F(X)
Number of people in each cell
Total number of people

Fig. 3 Flow diagram of the SIFT-FAST features algorithm

Where F_{Grid} is the grid map of the frames, C is a cell in the grid map and n is the number of cells in the grid map.

9- Use a quadratic programming (Interior-point-convex algorithm) to find the density value of each cluster.

10- Integrate the densities of pixels over each cell to find the number of people in each cell.

$$N_{cell} = \sum_{(i,j)\in B_n} P_{density}(i,j) \tag{16}$$

Where N_{cell} is the number of people in each cell and $P_{density}(i,j)$ is the density of each pixel that belongs to the cell.

11- The summation of the number of people in all cells represents the total number of people in each frame.

$$N_{total} = \sum\nolimits_n N_{cell} \tag{17}$$

Where N_{total} is the total number of people in a frame, n is the number of cells.

3.3 Fusion technique

The fusion model is updated periodically using the results of all the algorithms. Each algorithm works independently to count the number of people and then they update the fusion model. Fusion is used to improve accuracy by determining the average error for each frame and to increase the confidence of the proposed system because the result of one algorithm is confirmed by that of another. This produces a cooperative paradigm and improves the confidence level in the results.

3.4 Geometric correction

At long distances, people appear smaller than those closer to the camera. Therefore, the extracted features of the same person at different locations in the scene are significantly different.

Re-scaling the pixels of the frames is implemented by assigning different weights to solve this problem. Fig. 4 shows the different sizes of the same pedestrian at different depths. Line (*ab*) is the reference line so the pixel's weight on that line is 1, the pixels of other lines are scaled and weighted using equation (18) [50];

$$weight_{line} = \frac{h_{ab} w_{ab}}{h_{\mathrm{line}} w_{\mathrm{line}}} \tag{18}$$

Where h_{line} and h_{ab} are the heights of a person at the line of interest and the height of the same person at the (*ab*) line, respectively. w_{line} and w_{ab} are the width of the rectangle at the line of interest and at (*ab*) line, respectively.

3.5 Background segmentation

Background segmentation is a process of extracting foreground information on a frame by frame basis. Background segmentation algorithms usually consist of three steps; background initialization, foreground detection and background maintenance [68]. In the background initialization, various techniques such as statistical, fuzzy and neuro-inspired techniques are used to build a background model. In foreground detection, a comparison is implemented between the current frame and the background model. Updating a background model according to changes in the environment is processed in the background maintenance step. Background segmentation

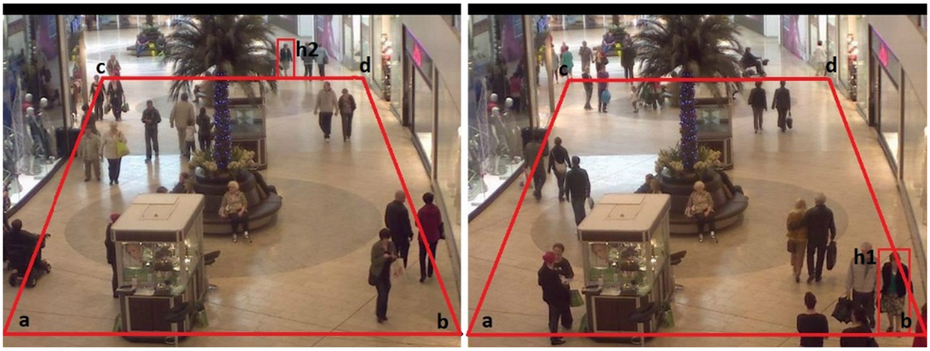


Fig. 4 The change of size of the same person at different locations

methods can be classified into recursive and non-recursive algorithms [2]. In non-recursive algorithms, the background model is considered to be static and does not update, whereas in recursive algorithm, it is a dynamic and changes depending on the change of environment [2]. Figure 5 shows the general block diagram of background segmentation algorithms.

GMM is one of the most widely used algorithms for background segmentation. This algorithm is a robust in light varying conditions and in environments with animated textures such as waves on the surface of water or trees being blown by wind [1]. Each pixel in a background model is formed using a mixture of Gaussian distributions (normally from three to five distributions) rather than one Gaussian distribution [1, 5].

$$p(X_t) = \sum_{i=1}^{K} w_{i,t} * f\left(x_t | \mu_{i,t}, \Sigma_{i,t}\right) \tag{19}$$

Where K is the number of Gaussian distributions and $w_{i\ ,\ t}$ is the weight of the i^{th} distribution at time t. Each Gaussian distribution can be found using the probability density function;

$$f(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{20}$$

Where μ is the mean and Σ is the covariance matrix. The background model is updated using an adaptive filter;

$$\mu_t = \alpha X_t + (1-\alpha)\mu_{t-1} \tag{21}$$

Where;

- μ_t denotes the spatial mean of the pixels at time t,
- μ_{t-1} denotes the previous spatial mean of the pixels at time t − 1,
- α is an empirical weight and.
- X_t is the current pixels values.

3.6 Edge detection

They refer to the process of localising pixel intensity transitions [61]. There is a strong relationship between the complexity of crowds and the number of people because crowded

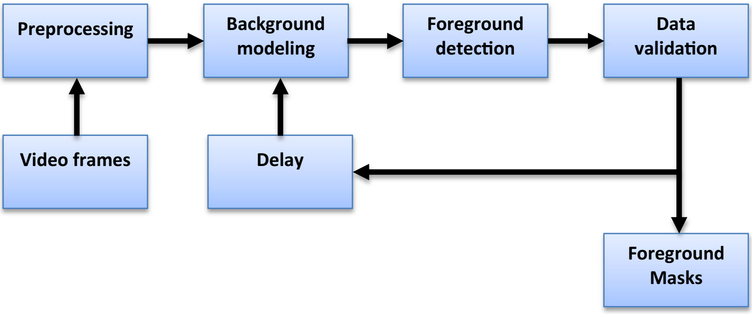


Fig. 5 General block diagram of background segmentation algorithms

environments tend to produce complex edges, while sparse environments tend to produce coarse edges [50]. Edges can be extracted using different algorithms such as Sobel, Canny, Prewitt, Roberts and Fuzzy logic algorithms [41, 42]. Canny edge detection is used in the proposed system. The following steps explain the procedure of canny edge algorithm [66]:

1- Smooth the image using a Gaussian filter to minimise noise.

$$S(i,j) = G(i,j,\Sigma)*I(i,j) \tag{22}$$

Where $G(i,j,\Sigma)$ is a Gaussian filter and $I(i,j)$ is a pixel.

2- Use derivative approximation by finite differences to find gradient magnitude and orientation. Firstly, partial derivatives $X(i,j)$ and $Y(i,j)$ is found by using the smoothed array $S(i,j)$:

$$X(i,j) \approx (S(i,j+1) - S(i,j) + S(i+1,j+1) - S(i+1,j))/2 \tag{23}$$

$$Y(i,j) \approx (S(i,j) - S(i+1,j) + S(i,j+1) - S(i+1,j+1))/2 \tag{24}$$

The partial derivatives $X(i,j)$ and $Y(i,j)$ are then used to find the magnitude and orientation of the gradient:

$$M(i,j) = \sqrt{X(i,j)^2 + Y(i,j)^2} \tag{25}$$

$$\theta(i,j) = \arctan(X(i,j), Y(i,j)) \tag{26}$$

3- Non-Maximal Suppression algorithm (NMS) is performed to thin out the edges. The edges are then detected using the double thresholding algorithm.

3.7 Clustering

Clustering is used in the proposed system to reduce the number of different descriptors (hundreds of thousands for 640 × 480 frame size) into a reasonable number of clusters (256 clusters in the SIFT features algorithm and 257 clusters in the SIFT-FAST features algorithm) that can be used with quadratic programming. K-means clustering is a method of vector quantisation and aims to partition n observations into $k \le n$ clusters such that each observation belongs to the cluster with the nearest mean [7]. In other words, it aims to find:

$$argmin_S \sum_{i=1}^{K} \sum_{X \in S_i} \|X - \mu_i\|^2 \tag{27}$$

Where X is the observation, S_i is the i^{th} cluster and μ_i is the mean of cluster S_i. In the proposed system, the k-means algorithm is used to cluster the SIFT descriptors of the datasets frames and produce a codebook of 256 entries. The codebook is constructed using only the descriptors of the training frames and then the descriptors of the testing frames are clustered by the K-means algorithm and the codebook. The SIFT descriptor of each pixel is represented by one value between 1 and 256. A vector of length 256 is used to convert each pixel and quantise it by comparing them with the centroids in the codebook.

4 Results and discussion

4.1 Benchmark datasets

The pedestrian dataset from University of California, San Diego (UCSD) and the Mall datasets have been used to evaluate the proposed system [13, 16]. UCSD dataset has been widely used for testing and validating people counting methods [82]. Mall dataset is a newer and more comprehensive dataset to use in that it covers a different range of crowd densities, different activity patterns (static and moving crowds), collected under a large range of illumination conditions at different times of the day with more severe perspective distortion. Thus individual objects may exhibit larger variations in size and appearance at different depths of the scene [50]. The Mall dataset was introduced by Chen [16]. It has been collected inside a cluttered indoor and includes 2000 annotated frames. The two datasets have the same length (2000 frames) but they have different features in terms of the frame rate (fps), resolution, colour, location, shadows, reflections, crowd size and frame type [62, 63]. Table 1 shows the features of each dataset.

For the Mall and UCSD datasets, the datasets are partitioned into a training set, for learning the proposed system, and a test set, for validation. 100 frames from different locations of each dataset are allocated indivdually for training and 1900 frames for testing.

4.2 Evaluation metrics

Three metrics have been used as performance indicators for crowd counting; Mean deviation Error (MDE), mean absolute error (MAE) and mean squared error (MSE) [50]. The MDE is defined as;

$$MDE = \frac{1}{N}\sum_{n=1}^{N}\frac{\left|y_n - \hat{y}_n\right|}{y_n} \tag{28}$$

Table 1 The features of the benchmark datasets

	Mall dataset	UCSD dataset
Year	2012	2008
Length (frames)	2000	2000
Frame rate (fps)	<2	10
Resolution	640 × 480	238 × 158
Colour	RGB	Grey
Location	Indoor	Outdoor
Shadows	Yes	No
Reflections	Yes	No
Loitering	Yes	No
Crowd size	11–45	13–53
Frame type	.jpeg	.png

The MAE is defined as;

$$MAE = \frac{1}{N}\sum_{n=1}^{N}\left|y_n - \hat{y}_n\right| \tag{29}$$

The MSE is given as;

$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - \hat{y}_n\right)^2 \tag{30}$$

Where N is the total number of the test frames, y_n is the actual count, and y_n is the estimated count of n_{th} frames. MAE and MSE are indicative quantities of the error of the estimated crowd count but they contain no information about how crowded the environment is [50]. MDE takes into account the crowdedness and gives an indication of how good a measurement is relative to the actual count [32].

4.3 Background segmentation, edge detection and motion edge extraction

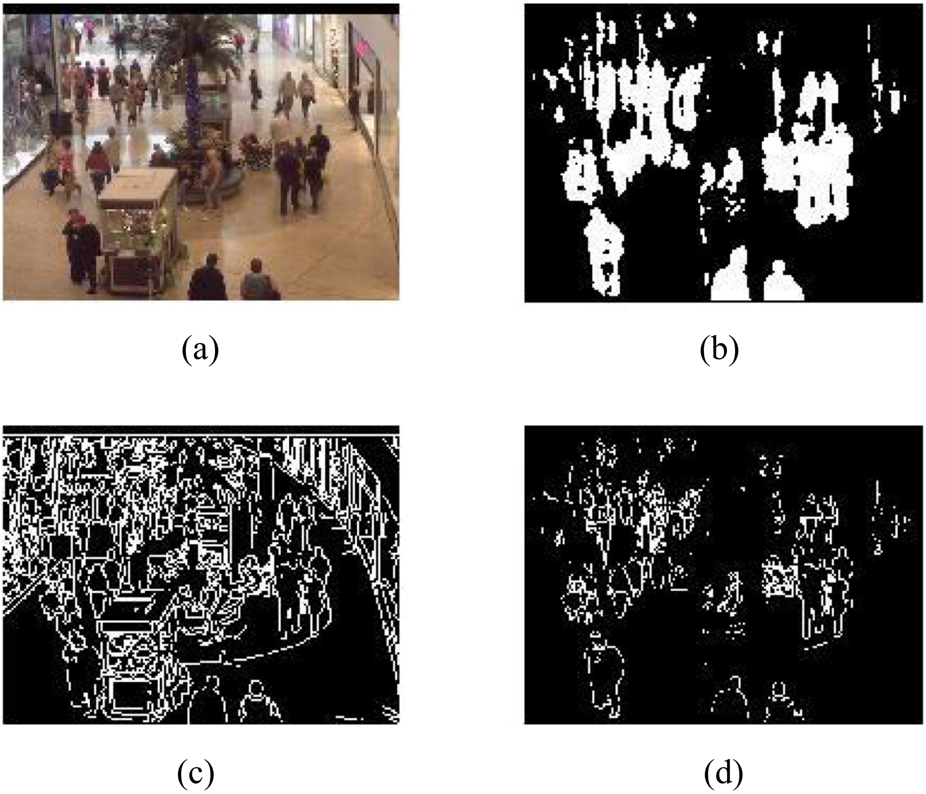
The GMM is used for background segmentation and the Canny edge algorithm is performed to extract the edges of the frames. The logical 'AND' is used to extract motion edge. Figs. 6 and 7 show the results of the background segmentation, edge detection and motion edge extraction of two sample frames, one from the Mall dataset and the second from the UCSD dataset.

4.4 Performance evaluation of the proposed system using the mall dataset

As shown in Table 2, the mean deviation error (MDE) of the SIFT features algorithm is 0.099 and 0.094 for SIFT-FAST features algorithm. The results are compared with results presented by other researchers for the same dataset as a measure of accuracy of the proposed system. From the results, we can see that the accuracy of the SIFT-FAST features algorithm is slightly better than that of SIFT features algorithm. It shows that there is a reasonable improvement in the accuracies of the implemented algorithms when compared to those published by other researchers. Figure 8 shows the percentage of frames within the MDE distribution of the algorithms. Figure 9 shows the true count (TC) of people from sample frames of the Mall dataset, which is annotated by red dots. EC1 and EC2 represent the estimated number of people using SIFT and SIFT-FAST features algorithms, respectively.

The performance of crowd systems is measured using the accuracy (MAE, MSE and MDE) and practicality. Practicality is measured by the percentage of the training frames minimisation [60]. Crowd counting systems are practical if they are easy to deploy. In the real world, crowd systems are deployed in different environments which means they are individually trained for the location. Therefore, it is very important to reduce the number of the training frames required. The ground truth (the actual number of people) for each training frame is required when training crowd counting systems. Each environment needs several hundreds of frames (usually 400–800 training frames) for the training [15, 23, 24, 83], so the training process becomes time-consuming.

The results of the proposed system have been compared with recent results from other researchers for the evaluation. The comparison with other methods based on the accuracy metrics (MAE, MSE and MDE) is not enough to measure the performance for many reasons:



(a)

(b)

(c)

(d)

Fig. 6 **a** An example of the Mall dataset form; **b** foreground, using GMM algorithm; **c** edge using Canny detector; **d** the motion edge, using logical 'AND'

firstly, pixel-wise optimisation based algorithms can be trained using a small number of frames in comparison to regression based algorithms [47]. The proposed system uses 100 frames for the training whilst the other state of the art methods use between 400 and 800 frames [15, 23, 24, 83]. In conclusion, the proposed system is more practical because the set-up time is faster by a factor of at least four (uses 4 times less training frames) compared to regression based algorithms which lead also to low set-up cost. Secondly, the lower number of training frames required in the training stage reduces the potential for error being introduced because manually annotation is an error-prone task. The accuracy of crowd counting systems are significantly affected by errors in the training stage. Thirdly, the proposed system is a multipurpose system because it can be used for crowd counting and also in people detection [58].

The comparison is only used to show that although the proposed system reduces the training error, speed, cost and can be used to develop more accurate people detection methods, its accuracy is, at least, comparable with the state of the art methods. None of the published results presented in Table 2 performs better than SIFT-FAST features algorithm based on the metrics used in this paper. In terms of MSE, only the algorithms presented in [16, 80] and [38] produced slightly better results but not in terms of MAE and MDE metrics. Finally, the MDE of the proposed system is less than the acceptable error (0.2) which is meeting the minimum accuracy requirements of system operators [62].

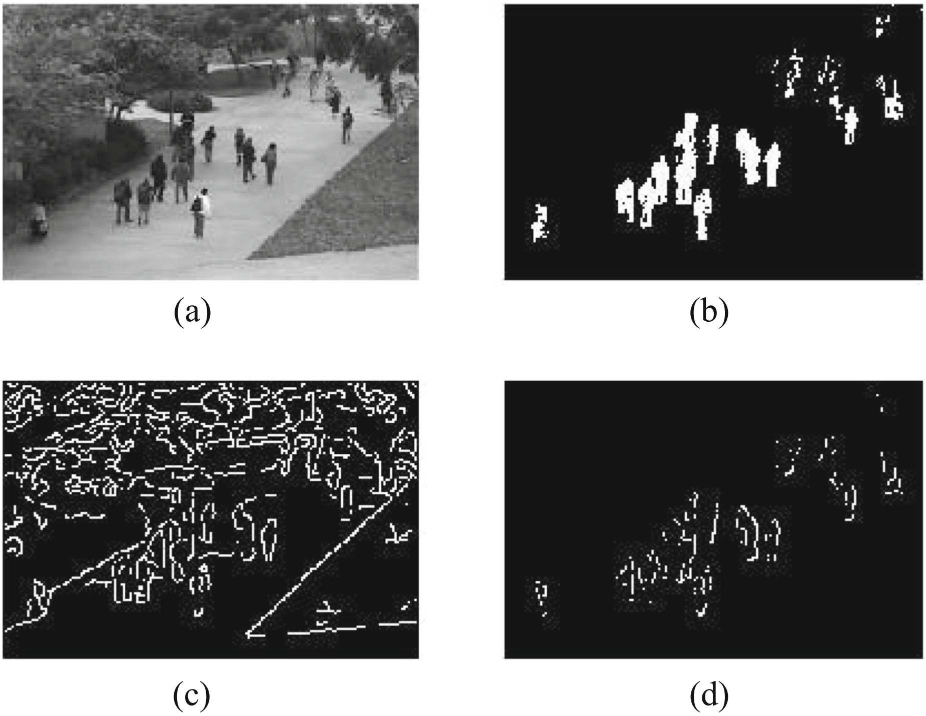


Fig. 7 **a** An example of the UCSD dataset form; **b** foreground, using GMM algorithm; **c** edge using Canny detector; **d** the motion edge, using logical 'AND'

Table 2 Comparison for the Mall dataset results between the proposed system and the state of the art algorithms

Algorithm	Mall dataset		
	MAE	MSE	MDE
Algorithm 1: SIFT Features Algorithm	3.08	16.31	0.099
Algorithm 2: SIFT-FAST Features Algorithm	2.94	14.64	0.094
Cumulative attribute based model (CA-RR) [17]	3.43	17.70	0.105
Squares Support Vector Machine Regression (LSSVR) [17]	3.51	18.20	0.108
Kernel Ridge Regression (KRR) [17]	3.51	18.10	0.108
Random Forest Regression (RFR) [17]	3.91	21.50	0.121
Gaussian Process Regression (GPR) [16, 17]	3.72	20.1	0.115
Ridge regression (RR) [16, 17]	3.59	19.00	0.110
Multi Output Ridge Regression (MORR) [16]	3.15	15.70	0.099
Multiple Localised Regression (MLR) [16]	3.90	23.90	0.119
Weighted Ridge Regression (WRR) [15]	3.44	18.00	0.105
Random Projection Forest (RPF) [80]	3.22	15.50	-
Cost-sensitive Sparse Linear Regression (CS-SLR) [38]	3.23	15.77	0.104

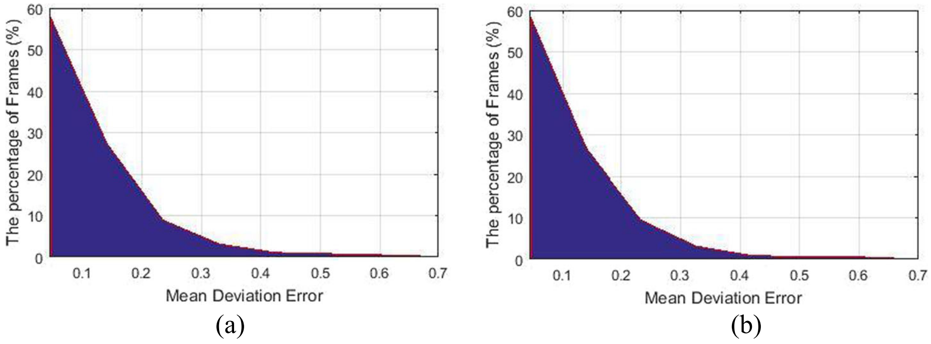
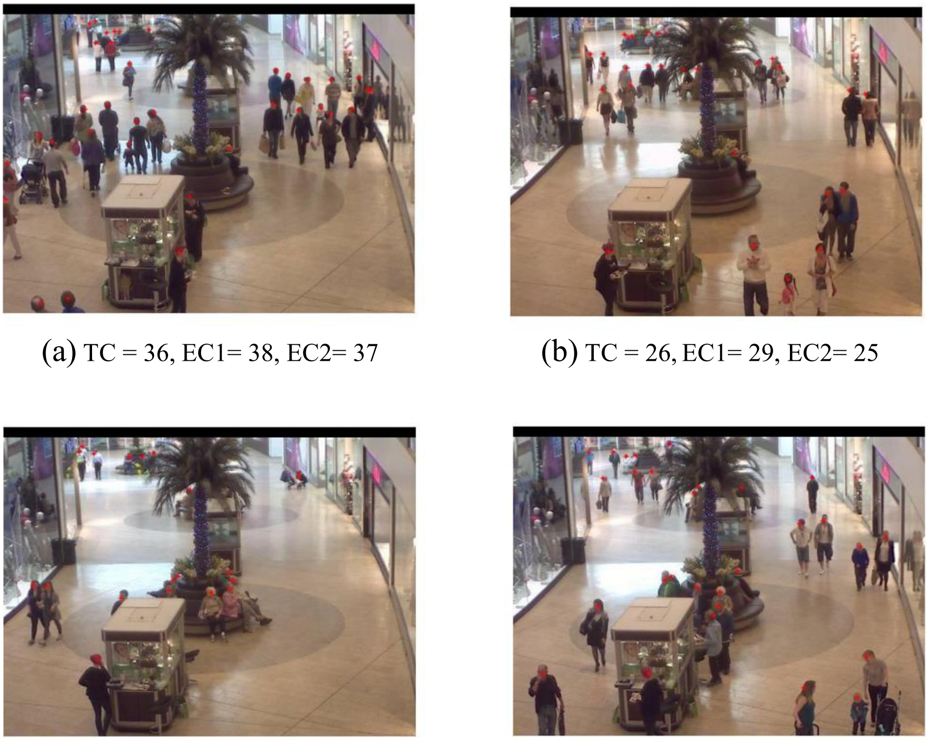


Fig. 8 **a** The MDE of SIFT algorithm **b** the MDE of SIFT-FAST algorithm

4.5 Performance evaluation of the proposed system using the UCSD dataset

The UCSD dataset represents people moving in two directions along a walkway. As shown in Table 3, the MDE of SIFT features algorithm is 0.066 and 0.064 for SIFT-FAST features algorithm. From the results, it can be seen that the accuracy of SIFT-FAST features algorithm is better than that of SIFT features algorithm. Figure 10 shows the percentage of frames within



(a) TC = 36, EC1= 38, EC2= 37

(b) TC = 26, EC1= 29, EC2= 25

(c) TC = 19, EC1= 20, EC2= 20

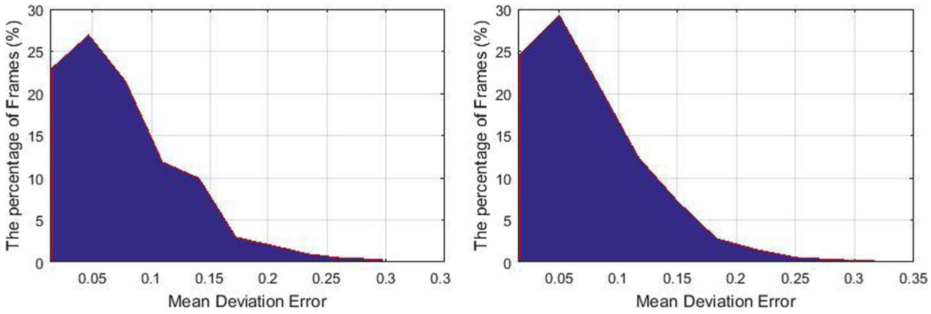
(d) TC = 29, EC1= 32, EC2= 28

Fig. 9 Examples of the true count (TC) & the estimated count of people using SIFT (EC1) and SIFT-FAST (EC2) algorithms

Table 3 Comparison for the UCSD dataset results between the proposed system and the state of the art algorithms

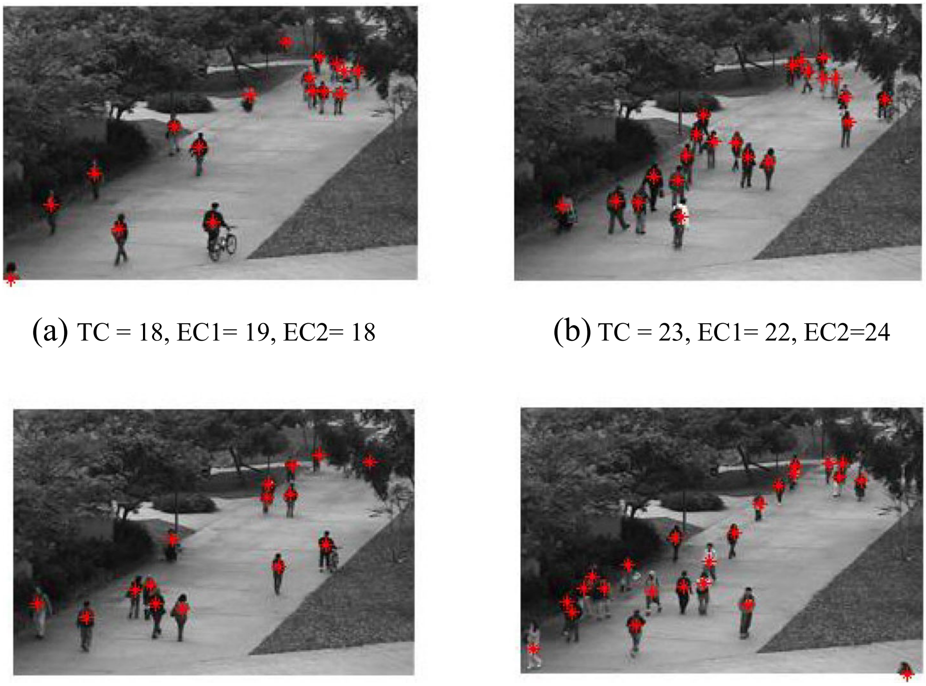
Algorithm	UCSD dataset		
	MAE	MSE	MDE
Algorithm 1: SIFT Features Algorithm	1.82	5.24	0.066
Algorithm 2: SIFT-FAST Features Algorithm	1.76	4.93	0.064
Improved Iterative Scaling -Label Distribution Learning (IIS-LDL) [83]	2.08	7.25	0.098
Kernel Ridge Regression (KRR) [83]	2.16	7.45	0.107
Random Forest Regression (RFR) [83]	2.42	8.47	0.116
Gaussian Process Regression (GPR) [15, 83]	2.24	7.97	0.112
Ridge Regression (RR) [15, 83]	2.25	7.82	0.110
Multi Output Ridge Regression (MORR) [83]	2.29	8.08	0.109
Cumulative attribute based model (CA-RR) [17, 83]	2.07	6.86	0.102
Weighted Ridge Regression (WRR) [15]	2.05	6.75	0.102
Linear regression (LR), Partial Least Squares Regression (PLSR), KRR, LSSVR, GPR and RFR [50]	>2.02	>6.67	>0.100
Random Projection Forest (RPF) [80]	1.90	6.01	-
Cost-sensitive Sparse Linear Regression (CS-SLR) [38]	1.83	5.04	0.079
Moving SIFT algorithm [24]	3.26	-	0.180

the MDE distribution of the algorithms. Figure 11 shows the true count (TC) of people from sample frames of the UCSD dataset, which is annotated by red dots. In general, the accuracies of the proposed system with the UCSD dataset are better than the results from the Mall dataset. The potential justification is that the Mall dataset is more complicated in terms of shadows, reflections and crowd size [62, 63]. In addition, the Mall dataset is collected with more severe perspective distortion than the UCSD dataset. As is the case with the MDE from the Mall dataset, the MDE of this dataset is significantly lower than the acceptable error (0.2) which is meeting the minimum accuracy requirements of system operators [62]. Results of both datasets show that their average accuracies for each dataset are almost similar but their accuracies at frame level are different. The difference of estimation between the SIFT and SIFT-FAST algorithms for each frame is usually between 0 and 4. EC1 and EC2 at each frame are



(a) (b)

Fig. 10 **a** MDE of SIFT algorithm **b** MDE of SIFT-FAST algorithm



(a) TC = 18, EC1= 19, EC2= 18

(b) TC = 23, EC1= 22, EC2=24

(c) TC = 15, EC1= 15, EC2= 17

(d) TC = 23, EC1= 21, EC2= 22

Fig. 11 Examples of the true count (TC) & the estimated count of people using SIFT (EC1) and SIFT-FAST (EC2) algorithms

correlative because both algorithms use almost the same approach. However, FAST corner points are used with SIFT-FAST features algorithm to improve the accuracy due to the high correlation between the number of people and FAST corner points. SIFT-FAST features algorithm gives the best results compared to all published results presented in Table 3. Only results presented in [38] gives a comparable results to the SIFT features algorithgm.

4.6 Performance evaluation of the proposed system in sparse and crowded scenarios

To evaluate the proposed system with sparse and crowded scenarios, the test set of the Mall dataset is split the same as in [50] into a sparse set which includes all the frames with ground truth (number of people), less than or equal to 30, and crowded set which includes all the frames with ground truth

Table 4 System performance with sparse and crowded scenarios (Mall dataset)

Algorithm	Sparse scenario			Crowded scenario		
	MAE	MSE	MDE	MAE	MSE	MDE
SIFT features Algorithm	3.20	18.39	0.126	2.96	14.27	0.081
SIFT-FAST features Algorithm	3.15	17.21	0.124	2.73	12.11	0.075

Table 5 System performance with sparse and crowded scenarios (UCSD dataset)

Algorithm	Sparse scenario			Crowded scenario		
	MAE	MSE	MDE	MAE	MSE	MDE
SIFT features Algorithm	1.67	4.41	0.093	1.93	5.84	0.055
SIFT-FAST features Algorithm	1.65	4.29	0.084	1.84	5.41	0.056

values greater than 30. The test set of the UCSD dataset is also split the same as in [50] into a sparse set which includes all the frames that their ground truth is less than or equal to 23, and crowded set which includes all the frames that their ground truth is greater than 23.

To ensure that the proposed system is practical and robust, the training set was not been split because the technical definition of the boundary that separates the sparse and crowded frames is not clear [3]. In addition, partitioning the training set into two sets would required two training stages. The test sets are processed by the proposed system jointly and then the results are analysed by splitting them into sparse and crowded sets. In conclusion, the split between sparse and crowded scenarios have mainly been carried out by identifying which frames could be classified into each of the categories. No differential training of the system has been carried out. Tables 4 and 5 show the results of both algorithms with sparse and crowded scenarios. The MDE of both algorithms in the sparse scenarios is higher than the MDE crowded scenarios. The proposed system is more applicable for high density crowds and this can be seen from the achieved good results in crowded

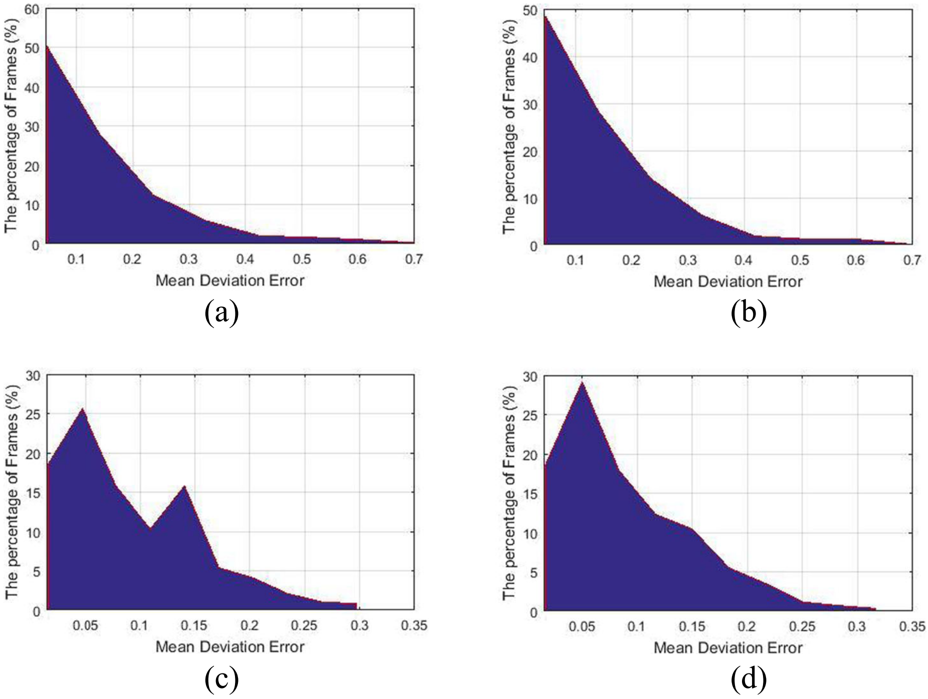


Fig. 12 System performance with sparse scenarios. (**a**) and (**b**) are the MDE of the of SIFT algorithm and SIFT-FAST algorithm on Mall dataset, respectively; (**c**) and (**d**) are the MDE of the of SIFT algorithm and SIFT-FAST algorithm on UCSD dataset, respectively

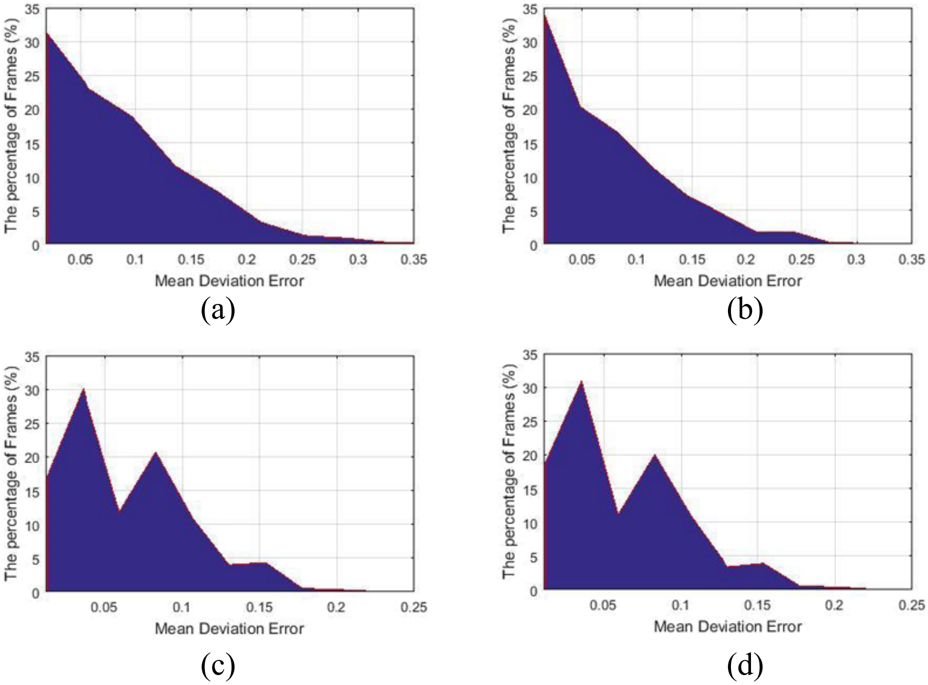


Fig. 13 System performance with crowded scenarios. (**a**) and (**b**) are the MDE of the of SIFT algorithm and SIFT-FAST algorithm on Mall dataset, respectively; (**c**) and (**d**) are the MDE of the of SIFT algorithm and SIFT-FAST algorithm on UCSD dataset, respectively

scenarios. This opens the door for using the proposed system in a high crowded environments. Figures 12 and 13 show the percentages of frames within the MDE distribution for the sparse and crowded scenarios based on the Mall and UCSD datasets, respectively.

5 Conclusions

CCTV cameras are already widely used, the objective of the research presented in this paper was to develop a system that can be incorporated with existing CCTV cameras to provide the number of people in a given space. Two algorithms have been proposed and implemented using a novel combination of four techniques; motion edges, grid map, SIFT & FAST features and pixel-wise techniques. The use of edge pixels for which their number is small compared to foreground pixels significantly reduces the run time of the algorithms. SIFT and FAST features have been chosen due to their high correlation with the number of people. In addition, a grid map approach has been proposed and used to allow similar clusters in different cells to be assigned different densities depending on their location in the frame. This is used to improve the adaption of the proposed algorithms to high variations in crowd behaviours, distributions, lighting and densities.

The UCSD and Mall datasets have been used to evaluate the proposed system. The results have shown that the proposed algorithms achieve good results in heavily occluded environment with perspective distortions. Comparisons with the low-level features regression based methods published in literature show that the proposed algorithms improve the accuracies based on MDE, MSE and MAE metrics (less than 0.1, 16.5 and 3.1, respectively, for the Mall dataset and less than 0.07, 5.5 and 1.9, respectively, for UCSD dataset). The proposed system is more practical than low-level features regression based methods because it can be trained with a lower number of frames so it is relatively easy to deploy. In addition, it reduces the training error, speed, cost and, opens the door to developing more accurate people detection methods. The proposed algorithms can also be used to estimate crowd densities at specific locations in a scene. This shows significant promise as it can be used to detect localised abnormalities in applications such crowd control, evacuation planning and product displays. Comparison of the proposed system in sparse and crowded scenarios shows that it performs better in crowded environments.

References

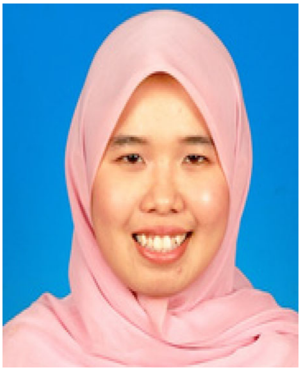
1. Adegboye AO (2013) Single pixel robust approach for background subtraction for fast people-counting and direction estimation. University of Pretoria, Dissertation
2. Adegboye A, Hancke G, Jr GH (2012) Single-pixel approach for fast people counting and direction estimation. South. Africa Telecommun, Networks Appl
3. Al-Zaydi ZQH, Ndzi DL, Yang Y, Kamarudin ML (2016) An adaptive people counting system with dynamic features selection and occlusion handling. J Vis Commun Image Represent 39:218–225. doi:10.1016/j.jvcir.2016.05.018
4. Antonini G, Thiran JP (2004) Trajectories clustering in ICA space an application to automatic counting of pedestrians in video sequences. Adv. Concepts Intell. Vis, Syst
5. Benezeth Y, Jodoin P-M, Emile B et al (2010) Comparative study of background subtraction algorithms. J Electron Imaging 19:33003. doi:10.1117/1.3456695
6. Berndt D, Clifford J (1994) Using dynamic time warping to find patterns in time series. Report, AAAI
7. Bottesch T, Markus K, Kaechele M, Ulm U (2016) Speeding up k -means by approximating Euclidean distances via block vectors. Int. Conf. Mach. Learn, In, pp. 2578–2586
8. Bouwmans T, El Baf F, Vachon B (2008) Background modeling using mixture of Gaussians for foreground detection - a survey. Recent Patents Comput Sci 1:219–237. doi:10.2174/2213275910801030219
9. Brostow GJ, Cipolla R (2006) Unsupervised Bayesian detection of independent motion in crowds. IEEE Conf Comput Vis Pattern Recognit. doi:10.1109/CVPR.2006.320
10. Çelik H, Hanjalić A, Hendriks EA (2006) Towards a robust solution to people counting. Int. Conf. Image Process, In, pp. 2401–2404
11. Chan AB, Vasconcelos N (2009) Bayesian poisson regression for crowd counting. In: IEEE Int. Conf, Comput. Vis. IEEE, pp. 545–551
12. Chan AB, Vasconcelos N (2012) Counting people with low-level features and bayesian regression. IEEE Trans Image Process 21:2160–2177. doi:10.1109/TIP.2011.2172800
13. Chan AB, Liang ZSJ, Vasconcelos N (2008) Privacy preserving crowd monitoring: counting people without people models or tracking. IEEE Conf Comput Vis Pattern Recognit. doi:10.1109/CVPR.2008.4587569
14. Chan A, Morrow M, Vasconcelos N (2009) Analysis of crowded scenes using holistic properties. In: Perform. Eval, Track. Surveill. Work. IEEE, pp. 101–108
15. Chen K, Kamarainen J-K (2014) Learning to count with back-propagated information. Int. Conf. Pattern Recognit. IEEE, In, pp. 4672–4677
16. Chen K, Loy CC, Gong S, Xiang T (2012) Feature Mining for Localised Crowd Counting. Br Mach Vis Conf. doi:10.5244/C.26.21
17. Chen K, Gong S, Xiang T, Loy CC (2013) Cumulative attribute space for age and crowd density estimation. In: IEEE Conf. Comput, Vis. Pattern Recognit, pp. 2467–2474
18. Cheriyadat AM, Bhaduri BL, Radke RJ (2008) Detecting multiple moving objects in crowded environments with coherent motion regions. IEEE Conf Comput Vis Pattern Recognit Work. doi:10.1109/CVPRW.2008.4562983
19. Cho SY, Chow TW (1999) A fast neural learning vision system for crowd estimation at underground stations platform. Neural Process Lett 10(2):111–120
20. Cho S-Y, Chow T, Leung C (1999) A neural-based crowd estimation by hybrid global learning algorithm. IEEE Trans Syst Man, Cybern Part B Cybern 29:535–541. doi:10.1109/3477.775269
21. Chow TWS, Yam JYF, Cho SY (1999) Fast training algorithm for feedforward neural networks: application to crowd estimation at underground stations. Artif Intell Eng 13:301–307. doi:10.1016/S0954-1810(99)00016-3
22. Conte D, Foggia P, Percannella G et al (1743–1746) (2010) counting moving people in videos by salient points detection. Int. Conf. Pattern Recognit. pp, In

23. Conte D, Foggia P, Percannella G et al (2010) A method for counting people in crowded scenes. In: IEEE Int. Conf, Adv. Video Signal Based Surveill, pp. 225–232
24. Conte D, Foggia P, Percannella G, Vento M (2013) Counting moving persons in crowded scenes. Mach Vis Appl 24:1029–1042. doi:10.1007/s00138-013-0491-3
25. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE Conf. Comput, Vis. Pattern Recognit, pp. 886–893
26. Davies A, Yin JH, Velastin S (1995) Crowd monitoring using image processing. Electron Commun Eng J. doi:10.1049/ecej:19950106
27. Dollar P, Belongie S, Perona P (2010) The fastest pedestrian detector in the west. Br Mach Vis Conf. doi:10.5244/C.24.68
28. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminative trained part based models. IEEE Trans Pattern Anal Mach Intell 32:1627–1645
29. Fradi H, Dugelay JL (2012) Low level crowd analysis using frame-wise normalized feature for people counting. Int. Work. Inf. Forensics Secur, In, pp. 246–251
30. Gao L, Wang Y, Ye X, Wang J (2016) Crowd Pedestrian Counting Considering Network Flow Constraints in Videos. arXiv Prepr
31. Ge W, Collins RT (2009) Marked point processes for crowd counting. In: Comput. Vis, Pattern Recognit. Work. IEEE, pp. 2913–2920
32. Hafeezallah A, Abu-Bakar S (2016) Crowd counting using statistical features based on curvelet frame change detection. Multimed Tools Appl. doi:10.1007/s11042-016-3869-1
33. Harville M (2002) Stereo person tracking with adaptive plan-view statistical templates. Proc. ECCV Work. Stat. Methods Video Process, In, pp. 67–72
34. Hashimoto K, Morinaka K, Yoshiike N et al (1997) People count system using multi-sensing application. Int. Solid State Sensors Actuators Conf, In, pp. 1291–1294
35. Hou YL, Pang GKH (2011) People counting and human detection in a challenging situation. IEEE Trans Syst Man, Cybern Part ASystems Humans 41:24–33. doi:10.1109/TSMCA.2010.2064299
36. Hu X, Zheng H, Chen Y, Chen L (2015) Dense crowd counting based on perspective weight model using a fisheye camera. Int J Light Electron Opt 126:123–130. doi:10.1016/j.ijleo.2014.08.132
37. Hu Y, Chang H, Nian F et al (2016) Dense crowd counting from still images with convolutional neural networks. J Vis Commun Image Represent 38:530–539. doi:10.1016/j.jvcir.2016.03.021
38. Huang X, Zou Y, Wang Y (2016) Cost-sensitive sparse linear regression for crowd counting with imbalanced training data. IEEE Int. Conf. Multimed, Expo
39. Intelcom DILAX (2015) Public Transport https://www.dilax.com/. Accessed 1 Oct 2016
40. Jeong CY, Choi S, Han SW (2013) A method for counting moving and stationary people by interest point classification. In: IEEE Int. Conf, Image Process. IEEE, pp. 4545–4548
41. Joshi NS, Choubey NS (2014) Comparison of traditional approach for edge detection with soft computing approach. Int J Comput Appl 96:17–23
42. Kaur G, Virk IS (2014) Edge detection through fuzzy system using type I format. Int J Comput Appl 102:24–27
43. Kilambi P, Ribnick E, Joshi AJ et al (2008) Estimating pedestrian counts in groups. Comput Vis Image Underst 110:43–59. doi:10.1016/j.cviu.2007.02.003
44. Kong D, Gray D, Tao H (2005) Counting pedestrians in crowds using viewpoint invariant training. Procedings Br Mach Vis Conf. doi:10.5244/C.19.63
45. Kong D, Gray D, Tao H (2006) A viewpoint invariant approach for crowd counting. Int. Conf. Pattern Recognit, In, pp. 1187–1190
46. Leibe B, Seemann E, Schiele B (2005) Pedestrian detection in crowded scenes. In: IEEE Conf. Comput, Vis. Pattern Recognit, pp. 878–885
47. Lempitsky V, Zisserman A (2010) Learning to count objects in images. Adv. Neural Inf. Process. Syst, In, pp. 1324–1332
48. Li J, Huang L, Liu C (2011) Robust people counting in video surveillance: dataset and system. Int. Conf. Adv. Video Signal Based Surveill, In, pp. 54–59
49. Lin S, Chen J, Chao H (2001) Estimation of number of people in crowded scenes using perspective transformation. IEEE Trans Syst Man Cybern 31:645–654
50. Loy C, Chen K, Gong S, Xiang T (2013) Crowd counting and profiling: methodology and evaluation. Model. Simul. Vis. Anal. Crowds. Springer New York, In, pp. 347–382
51. Ltd B (2013) Use CCTV to Count People http://www.videoturnstile.com/. Accessed 1 Oct 2016
52. Ma R, Li L, Huang W, Tian Q (2004) On pixel count based crowd density estimation for visual surveillance. In: IEEE Conf. Cybern, Intell. Syst. IEEE, pp. 1–3
53. Ma H, Zeng C, Ling CX (2012) A reliable people counting system via multiple cameras. ACM Trans Intell Syst Technol 3:1–22. doi:10.1145/2089094.2089107

54. Merad D, Aziz KE, Thome N (2010) Fast people counting using head detection from skeleton graph. Adv. Video Signal Based Surveill. IEEE, In, pp. 233–240
55. Norris C, Mccahill M, Wood D (2004) Editorial. The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space. Surveill Soc 2:110–135
56. Rabaud V, Belongie S (2006) Counting crowded moving objects. In: IEEE Conf. Comput, Vis. Pattern Recognit, pp. 705–711
57. Rao AS, Gubbi J, Marusic S, Palaniswami M (2015) Estimation of crowd density by clustering motion cues. Vis Comput 31:1533–1552. doi:10.1007/s00371-014-1032-4
58. Rodriguez M, Superieure EN, Laptev I et al (2011) Density-aware person detection and tracking in crowds. Int. Conf. Comput. Vis. IEEE, In, pp. 2423–2430
59. Ryan DA (2013) Crowd monitoring using computer vision. Queensland University of Technology, Dissertation
60. Ryan D, Denman S, Fookes C, Sridharan S (2009) Crowd counting using multiple local features. Digit. Image Comput. Tech. Appl. IEEE, In, pp. 81–88
61. Ryan D, Denman S, Fookes C, Sridharan S (2014) Scene invariant multi camera crowd counting. Pattern Recogn Lett 44:98–112. doi:10.1016/j.patrec.2013.10.002
62. Ryan D, Denman S, Sridharan S, Fookes C (2015) An evaluation of crowd counting methods, features and regression models. Comput Vis Image Underst 130:1–17. doi:10.1016/j.cviu.2014.07.008
63. Saleh SAM, Suandi SA, Ibrahim H (2015) Recent survey on crowd density estimation and counting for visual surveillance. Eng Appl Artif Intell 41:103–114. doi:10.1016/j.engappai.2015.01.007
64. Shbib R, Zhou S, Ndzi D, Al-kadhimi K (2013) Distributed monitoring system based on weighted data fusing model. Am J Soc Issues Humanit 3:53–62
65. ShopperTrak (2013) ShopperTrak Solutions http://www.shoppertrak.com/. Accessed 1 Oct 2016
66. Shrivakshan GT, Chandrasekar C (2012) A Comparison of various Edge Detection Techniques used in Image Processing Int J Comput Sci Issues:9
67. Sidla O, Lypetskyy Y, Brändle N, Seer S (2006) Pedestrian detection and tracking for counting applications in crowded situations. IEEE Int Conf Video Signal Based Surveill. doi:10.1109/AVSS.2006.91
68. Sobral A, Vacavant A (2014) A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Comput Vis Image Underst 122:4–21. doi:10.1016/j.cviu.2013.12.005
69. Tang NC, Lin Y-Y, Weng M, Liao HM (2015) Cross-camera knowledge transfer for Multiview people counting. IEEE Trans Image Process 24:80–93. doi:10.1109/TIP.2014.2363445
70. Technology A (2013) Our customers http://www.peoplecounting.co.uk/our-customers. Accessed 1 Oct 2016
71. Topkaya IS, Erdogan H, Porikli F (2014) Counting people by clustering person detector outputs. In: IEEE Int. Conf, Adv. Video Signal Based Surveill. IEEE, pp. 313–318
72. Tu J, Zhang C, Hao P (2013) Robust real-time attention-based head-shoulder detection for video surveillance. In: IEEE Int. Conf, Image Process. IEEE, pp. 3340–3344
73. Tuzel O, Porikli F, Meer P (2008) Pedestrian detection via classification on Riemannian manifolds. IEEE Trans Pattern Anal Mach Intell. doi:10.1109/TPAMI.2008.75
74. Wang M (2014) Data assimilation for agent-based simulation of smart environment. Georgia State University, Dissertation
75. Wang M, Wang X (2011) Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In: IEEE Conf. Comput, Vis. Pattern Recognit, pp. 3401–3408
76. Wang J, Fu W, Liu J et al (2014) Spatiotemporal group context for pedestrian counting. IEEE Trans Circuits Syst Video Technol 24:1620–1630
77. Wu B, Nevatia R (2007) Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. Int J Comput Vis 75:247–266. doi:10.1007/s11263-006-0027-7
78. Xiaohua L, Lansun S, Huanqin L (2006) Estimation of crowd density based on wavelet and support vector machine. Trans Inst Meas Control 28:299–308. doi:10.1191/0142331206tim178oa
79. Xing X, Wang K, Lv Z (2015) Fusion of gait and facial features using coupled projections for people identification at a distance. Signal Process Lett 22:2349–2353
80. Xu B, Qiu G (2016) Crowd density estimation based on rich features and random projection Forest. IEEE Winter Appl. Comput. Vis, In, pp. 1–8
81. Zhang J, Tan B, Sha F, He L (2011) Predicting pedestrian counts in crowded scenes with rich and high-dimensional features. IEEE Trans Intell Transp Syst 12:1037–1046. doi:10.1109/TITS.2011.2132759
82. Zhang C, Li H, Wang X (2015a) Cross-scene crowd counting via deep convolutional neural networks. Proc IEEE Conf Comput Vis Pattern Recognit. doi:10.1109/CVPR.2015.7298684
83. Zhang Z, Wang M, Geng X (2015b) Crowd counting in public video surveillance by label distribution learning. Neurocomputing 166:151–163. doi:10.1016/j.neucom.2015.03.083

Zeyad Q. H. Al-Zaydi received his B.S. in Computer Engineering from University of Technology, Iraq in 2003. He received his M.S. in Computer Engineering from University of Baghdad, Iraq in 2012. He is currently a PhD student in the school of Engineering, University of Portsmouth, UK. His research interests are in the areas of multimedia surveillance systems, image processing, computer vision, image representation, pattern recognition, artificial intelligence, people detection and crowd counting.

Dr. David L. Ndzi graduated with BSc (Joint Honours) in Electronics and Mathematics from Keele University in 1994, and a PhD in Telecommunications from the University of Portsmouth in 1998. He has been a lecturer since 1999 and International Coordinator of the Faculty of Technology since 2007. His research focuses on video and image processing, wireless sensor networks and mesh networks for applications in precision agriculture, environmental monitoring, behavioural economics, security, building control and energy management, etc.



Dr. Munirah L. Kamarudin graduated with Ph. D in Computer Engineering, Universiti Malaysia Perlis, UniMAP (2012). M. Sc Communication Network Management and Planning, University of Portsmouth, UK (2008). B. Eng (Hons) Computer Science and Media Engineering, University of Yamanashi,Japan (2006). Her research interests include network architecture, wireless sensor network, mobile communications and information and communication technology.

Dr. Ammar Zakaria graduated with B. Eng. (Hons.) in Electronic and Computer Engineering from the University of Portsmouth, UK. in 2008. He received his a PhD in Mechatronic Engineering from Universitiy Malaysia Perlis, 2013. His research interest include sensor technology, robotics, wireless sensor and actuator network and Image processing.

Prof. Ali Y.M. Shakaff is currently a Professor at the School of Mechatronic Engineering, Universiti Malaysia Perlis. His academic career has spanned over a period of 23 years, throughout which he has been thoroughly involved in teaching and research in various areas of electronic engineering. He has also spent a considerable time in academic management, notably in the development of new engineering schools for Universiti Malaysia Perlis (UniMAP) and before that, Universiti Sains Malaysia (USM). Prior to joining UniMAP in 2002 (as part of the founding team), he was the Dean for the School of Electrical & Electronic Engineering, USM (1996-2002). He was formerly UniMAP's Deputy Vice-Chancellor for Academic and International Affairs (2002-2009).