Efficient People Counting with Limited Manual Interferences

Jingsong Xu[1], Qiang Wu[2], Jian Zhang[3], Boreak Silk[4], Gia Thuan Ngo[3], Zhenmin Tang[1]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]School of Computing and Communications, University of Technology, Sydney, Australia
[3]Advanced Analytics Institute and and School of Software, University of Technology, Sydney, Australia
[4]City Research, City of Melbourne, Australia
xjsxujingsong@gmail.com, tzm.cs@mail.njust.edu.cn, thuan_ngo@hotmail.com
{Qiang.Wu, Jian.Zhang}@uts.edu.au, Boreak.Silk@melbourne.vic.gov.au

***Abstract*—People counting is a topic with various practical applications. Over the last decade, two general approaches have been proposed to tackle this problem: a) counting based on individual human detection; b) counting by measuring regression relation between the crowd density and number of people. Because the regression based method can avoid explicit people detection which faces several well-known challenges, it has been considered as a robust method particularly on a complicated environments. An efficient regression based method is proposed in this paper, which can be well adopted into any existing video surveillance system. It adopts color based segmentation to extract foreground regions in images. Regression is established based on the foreground density and the number of people. This method is fast and can deal with lighting condition changes. Experiments on public datasets and one captured dataset have shown the effectiveness and robustness of the method.**

***Keywords*—people counting; pedestrian counting; image segmentation; regression**

I. Introduction

Given an image, the objective of people counting is to correctly estimate the number of people in the scene as in Fig. 1. This is a topic with a number of practical applications. The knowledge of the number of people at certain places at certain time can foster vital business decisions. In public area, the knowledge of population density can be used to reduce congestion. Another field where people counting can be applied is in automatic hazard management. Areas that require low density crowd such as emergency exists can be automatically monitored; warnings can be provided when the area is above the density limit.

There have been several attempts at people counting in literature. Previous work in this area can be classified into two main approaches: counting based on individual human detection and counting by measuring regression relation between the crowd density and number of people. Human detection methods attempt to identify humans directly from the scene, they thus are able to count the number of people as well as locating their individual positions. Regression based methods infer the number of people based on certain characteristics of the scene. Hou and Pang [1] have recently proposed a regression based method for people counting. Regression is performed on the number of foreground pixels by first separating the background and foreground using dynamic background subtraction. Such method has shown its effectiveness for people counting. However, it still has drawbacks. Firstly, it relies heavily on background model. The method assumes that a reliable background can be obtained by using an adaptive Gaussian Mixture Model (GMM) framework. This method shows difficulties when the scene is changing rapidly. The paper also noticed a performance decreases when static objects or humans are leaving the scene. Moreover, static humans are misclassified as the background, and thus leads to under-counting. For example in Fig. 3, people sit there for a long time in the park becoming part of background based on GMM. Secondly, it can only be applied into video based counting since the background model needs to be updated. Finally, the input feature is the whole number of foreground pixels (1-D), resulting in high variance which is not stable in many cases.

(a) (b)

Fig. 1: Images are from (a) UCSD dataset and (b) Mall dataset.

In this paper, we propose a regression based method to overcome these limitations. The major contributions of this paper are as follows. Instead of using GMM to model background and segment foreground region in the image, color and clustering based segmentation is applied to deal with images. The foreground pixels in different image blocks are taken as feature vector for regression. Such combination is more robust when background model is difficult to build. This efficient method achieves excellent performance in three datasets.

The remainder of this paper is structured as follows. Related work is given in section II; our regression based method is proposed in section III; the experimental results are presented in section IV and conclusion is in selection V.

II. Related Work

People counting is an indirect output for human detection task by simply adding all the detections. Sliding and scaling window which takes feature and classification for human detection is the most common technique. For each window, certain features are extracted and fed to a classifier. Over the decade, considerable progress has been made on visual feature extraction and classifiers for this technique. Histogram of oriented gradient (HOG) [2], covariance (COV) [3] and local binary patterns (LBP) [4] have been proposed. In the meantime, multiple features combination and feature mining algorithms [5], [6] are also proposed to enhance the performance of the single type features. In the field of classifier [7], Support Vector Machines and various boosted classifiers are still the two leading classifiers for their good performance and efficiency. On the other hand, part-based methods [8], [9] have archived excellent performances since they can deal with occlusion problem. Motion information can also be added when dealing with video streams [10]. Human detection methods encounter difficulties in scenes with high occlusions (e.g. Fig. 3) and they are time consuming. More importantly, this kind of method has the difficulty when dealing with the smaller size of people, for example the smallest requirement for human image is 64×128 pixels in HOG method.

Regression based methods infer the number of people based on certain characteristics of the scene. These methods have misclassification in presence of shadows, noise or cluttered background in the scene. The features vary from simple foreground segment feature to more sophisticated textures or edge features. The relationship between the features and the estimated number of people are learned using a regression framework. This kind of method comes from density estimation which classifies the population density based on texture analysis. The key idea is that fine texture can be considered as having high crowd density, whereas coarse texture images tend to have lower density [11]. Texture analysis is carried out by using various feature extractors such as gray level dependency matrix [12], Minkowski Fractal Dimension [13], Gray-level co-occurrence matrix (GLCM) [14], local binary pattern (LBP) and HOG. Note that texture analysis can only estimate the crowdedness without giving exact number of people. Besides, a number of features of foreground image can also be used for regression purpose [15]: area [1], perimeter, perimeter-area ratio and blob count. These features are evaluated in [15] which takes a video input for foreground regions extraction. [16] combined multiple sources (head detection, repetition of texture elements, frequency-domain analysis) to estimate the number of people in an extremely dense crowed image. [17] counted the objects by estimating density whose integral over an image region. This method was also extended to arbitrary objects and scenes [18]. [19] proposed a multi-output regression model for crowd counting which can learn the mapping between interdependent local features from different spatial locations as input and multi-dimensional structured outputs. Above methods either adopt complicated feature extraction methods or model training techniques which are not efficient for people counting.

Following regression based method, this paper proposes a new efficient and effective method for people counting in images. The number of foreground pixels in different image blocks is taken as feature. It reduces variances and provides more local information while [1] takes the number of the whole foreground pixels as input. The foreground image is obtained by color based image segmentation rather than GMM model which requires video as input. The extracted feature is then applied to an efficient support vector regression for counting.

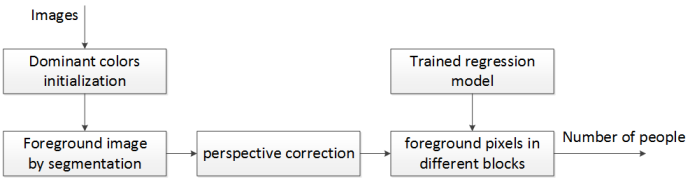


Fig. 2: Overview of the dominant colors segmentation method.

(a) (b)

Fig. 3: sample images with ROI.

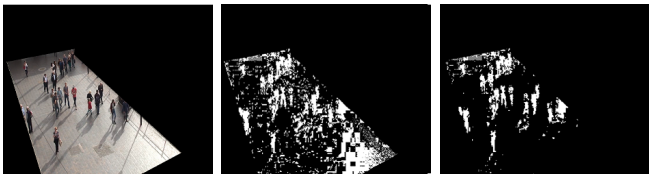
III. Proposed Method

Assuming that the background image consists of non-human objects and the foreground image consists of human, the foreground image can be used to infer the number of people in the scene. Foreground pixels methods estimate the number of people based on the number of foreground pixels. [1] uses GMM to build background model and obtains the foreground image by subtracting the scene from background image. A thresholding is then applied to the image to obtain the binary foreground image. Perspective correction is employed to bring objects to the same scale, ensuring that the number of pixels counted for each person is consistent regardless of their distance from the viewpoint. Finally, a neural network is trained on the relationship between the number of foreground pixels in the whole image and the number of people.

Following this method, in this paper we propose an improvement to this foreground pixels regression method. The flowchart is shown in Fig. 2. Instead of constructing GMM model, the proposed method separates foreground and background according to the scenes dominant colours. In this way, the proposed method is not constrained to either static image or video. We introduce the limited manual interferences which significantly improves the efficiency of proposed method. Clustering and morphological operations are performed to improve the results. The relationship between the foreground pixels and the number of people is learned by Support Vector Regression (SVR) [20]. Compared with [1], the key differences rely on color and clustering based segmentation which is adopted to deal with images rather than video frames. Besides, the whole image is divided into blocks and the feature in each block is concatenated for regression, providing more local information than the global regression in [1].

(a) (b) (c) (d)

Fig. 4: (a) Original image; (b) selected ROI; (c) selected dominant background colors; (d) Image after segmentation.



(a) (b) (c)

Fig. 5: (a) Frame 10000; (b) Frame 10000 after color segmentation; (c) Frame 10000 after clustering.

A. Foreground Estimation

1) Initialization: Without building a background model for the scene, the Region of Interest (ROI) is constructed using a manually supplied image mask shown as Fig. 4b and 4c. Non-black pixels indicate the region of interest while black pixels are ignored. The purpose of ROI is due to the difficulty in distinguishing small and occluded humans at distance, even to human eyes. ROI can also improve the performance by ignoring areas without people as shown in Fig. 3. The region to be counted is reasonably dense and sensible for processing. From the ROI, user manually selects multiple dominant background colors (background regions) in the scene. This step presents the dominant colors to the system, which uses this information in latter stages to extract the background of the image. The number of dominant colors should be related to the specific scene and be low (3 in this paper) to avoid misclassification of the foreground.

2) Segmentation: The goal of this step is to separate foreground from the image. For each dominant color selected, the system performs the following three separate steps, then the segmentation result from each dominant color is joined into the final image.

Color based foreground estimation: The mean and standard deviation of each dominant colour is extracted from the initialization step. For color images only the hue channel is used for this extraction process. For grey images the value channel is used instead. Assuming that the color follows a Gaussian normal distribution with mean μ and standard deviation σ then the main information is between $[-3\sigma, +3\sigma]$. Any pixel with its color within this range is considered as a background pixel. Otherwise the pixel is considered as foreground (see Fig. 4d for an example).

Clustering re-segmentation: Last step has roughly estimated the foreground regions. A clustering step then is used to refine the results and ensure robustness against lighting condition chances. The dominant color step is adopted as a clue to assist with segmentation [21]. It can minimize the total variance from both foreground and background classes. Given the last time segmentation result, the mean and deviation of each class are re-calculated and adopted to segment the whole image again. This re-segmentation step stops when the total variance of new iteration exceeds the previous one. Fig. 5 shows an example that in 10000-*th* frame, the segmentation is really bad considering lighting condition changes a lot from the beginning. After applying cluttering, the segmentation result is improved greatly.

Post processing: Noises in color based foreground estimation process and clustering process can lead to over-counting of foreground pixels and thus lower the overall accuracy. To reduce white noises, some post processing methods like morphological operations are employed.

B. Regression based people counting

1) Perspective correction: Due to perspective distortion, objects located in the further positions appears smaller (i.e. with less pixels) than the objects at the position closer to camera. The number of foreground pixels must be corrected by giving correct weights to pixels at the different positions. Such weights compensate the variance caused by the distortion.

From a sequence of images (video), a non-occluded and visually distinguishable human is tracked at the point of entry (closer position) and exit (further position) of the region of interest. Denote the height at the point of entry as h_1 and exit as h_2 with their corresponding y coordinates (image rows) y_1 and y_2. Assuming that perspective distortion only occurs in the vertical direction and that this distortion is linearly proportional to the y-coordinate, then a linear regression can be constructed to approximate the height of the person at any y-coordinate:

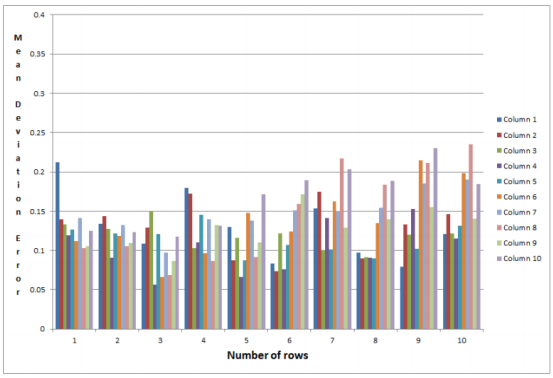
$$h = k_1 y + k_2 \tag{1}$$

where $k_1 = \frac{h_2 - h_1}{y_2 - y_1}$ and $k_2 = h_1 - k_1 y_1$. For a single image, two objects at a position further to camera and a position closer to the camera can be selected respectively. Such relation can be obtained by domain knowledge. Note that more complicated methods, such as camera calibration [22], [23] can also be applied.

Using this equation, the expected height of a person at any location in the image can be estimated. To improve the accuracy, multiple positions along the distance to the camera are recorded. The averages of k_1 and k_2 are obtained. Then to remove perspective distortion, the weight of each pixel can be obtained from

$$w(y) = \frac{1}{k_1 y + k_2} \tag{2}$$

2) Support vector regression: Using the total number of foreground pixels (i.e. 1-D feature vector) in the whole image would lead to high variance in the regression process. Noise also would severely affect this feature vector. To reduce variance, the whole image is divided into several blocks. The number of foreground pixels in each block is counted then concatenated into one single feature vector. The number of



(a)

Fig. 6: mean deviation error under increasing number of rows.

pixel calculation in each block B_i is a kind of weight summary:

$$\sum_{(x,y)\in B_i} w(y)\delta(x,y),\ \delta(x,y) = \begin{cases} 1, & (x,y) \in foregound \\ 0, & (x,y) \in background \end{cases} \tag{3}$$

$w(y)$ is the wight according to its y coordinates. This feature vector is then used for regression process to build the relationship to the number of people.

Given a training set $\mathbf{X} = \{(\mathbf{x}_1, y_i), (\mathbf{x}_2, y_i), ..., (\mathbf{x}_N, y_i)\}$ in which each sample is a $d-$dimensional variables $\mathbf{x}_i$ (feature vector, d is the number of image blocks) and its corresponding target value y_i (the number of people). The objective is to find a linear regression function $f(\mathbf{x})$ that relates $\mathbf{x}$ and y. In support vector regression (SVR) [20], the input sample $\mathbf{x}$ is mapped to a high-dimensional feature space using $\phi(\mathbf{x})$, then a linear model is built to estimate a regression function:

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w} \cdot \phi(\mathbf{x}) + b \tag{4}$$

This problem can be written as a convex optimization problem:

$$\begin{aligned} &\arg\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &\text{s.t.} \begin{cases} y_i - \mathbf{w}\cdot\mathbf{x}_i - b & \leq \epsilon + \xi_i \\ \mathbf{w}\cdot\mathbf{x}_i + b - y_i & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq 0 \end{cases} \end{aligned} \tag{5}$$

where ϵ is precision which assigns zero error if the absolute difference between the prediction $f(x, \mathbf{w}, b)$ and the target y is less than ϵ (> 0). ξ_i and ξ_i^* are slack variables. C is a constant which determines the trade off between overfitting and the deviations higher than ϵ that are tolerated.

IV. Experiment

A. Dataset

Three datasets are used in the evaluation process. The first dataset is a short video recorded in Sydney as shown in Fig. 4a. The ground truth count is done manually. This dataset contains people density from 10 to 30 with varied lighting condition changes. The first public dataset is UCSD [24] as shown in Fig. 1a with the ground truth count obtained from [25]. Perspective distortion is not severe in this dataset and the lighting condition is stable. The image is at very low resolution and the background is distinct from the humans. The second public dataset is Mall dataset [19], [26], [15] as shown in Fig. 1b. This is a harder dataset since there are various lighting condition changes at specific parts in the background. Moreover the background consists of multiple colors which renders dominant color segmentation difficult.

Dataset	Mean Abs. Error	Mean Sq. Error	Mean Dev. Error	10%	15%
Sydney	1.59	8.04%	5.56	68.60%	82.60%
USCD	1.14	5.63%	2.71	76.75%	90.25%
Mall	4.19	12.36%	33.56	46.00%	64.00%

TABLE I: Performance of the proposed method in the three datasets.

	Proposed	Segment	Edge	GLCM	LBP	S+E+GLCM	S+E+LBP
USCD	0.0937	0.0961	0.1177	0.1076	0.1192	0.0984	0.0938
Mall	0.1156	0.1564	0.1572	0.1612	0.1629	0.1532	0.1483

TABLE II: Mean deviation error (lower is better) of different features in the sparse scenarios.

B. Metrics

Standardised metrics are used in the evaluation process to allow comparison against other methods [15]. A common first choice for metrics is mean absolute error. This is the average difference between the predicted results $\hat{y}_n$ and the ground truth y_n in *n-th* frame.

$$\epsilon_{abs} = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n| \tag{6}$$

This metric however does not provide any information about density of the scene. While an error of 5 might be large in a scene with 10 people, it is insignificant in a scene with more than 1000. Therefore the mean deviation error is proposed to correct this bias [15]. It is the mean absolute error, normalized by the density of the scene.

$$\epsilon_{dev} = \frac{1}{N}\sum_{n=1}^{N}\frac{|y_n - \hat{y}_n|}{y_n} \tag{7}$$

Another popular metric is mean squared error. This metric squares the error in each test sample, essentially to punish large errors.

$$\epsilon_{sqr} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 \tag{8}$$

Finally two metrics are used for the usability of the system: Percentage of samples with error less than 10% and 15% [1]. These are calculated by counting the number of test samples with mean deviation error less than 10% and 15% respectively. A practical people counting system generally does not require a perfect count. In most cases an estimate with errors within 15% of the ground truth is reasonable. These two metrics determine if the system can be used in practice. They are not for benchmarking against other methods.

C. Results

We first explore the parameter setting for the number of blocks that a image should be divided into and then fix the parameter to evaluate the proposed methods on three datasets.

	Proposed	Segment	Edge	GLCM	LBP	S+E+GLCM	S+E+LBP
USCD	0.0656	0.0626	0.0579	0.0563	0.0849	0.0563	0.0587
Mall	0.0769	0.0674	0.0801	0.0674	0.0873	0.0698	0.0722

TABLE III: Mean deviation error (lower is better) of different features in the crowded scenarios.

1) The number of blocks exploration: The number of image blocks are evaluated by cross validation. Fig. 6 shows the counting results on the UCSD dataset. From the figure, we can see the results generally becomes worse as the number of rows increases. Rows 4-6 give the best counting performance. Increasing the number of columns past 5 worsens the results.

2) Results on three datasets: For the Sydney dataset, one image is taken every second over the duration for a total of 642 images. The first 400 images are in the training set while the remaining 242 form the test set. 400 images are for training and 400 image are for testing in the USCD dataset while 400 images are for training and 200 images are for testing in the Mall dataset. The results for three datasets are shown in TABLE I. The proposed method achieves the best performance on USCD dataset. 90.25% with error less than 10% shows the proposed method is effective for practical applications. This is due to the background having a single distinct color (grey, the road) from the pedestrians. The lighting condition changes throughout the frames but the color is still fairly similar to the original dominant color. These factors allowed the foreground to be extracted easily from the image. Perspective distortions are minimal which leads to lower variance in each segmented blocks. The background of Mall dataset consists of multiple dominant colors, it is difficult to extract the proper foreground.

D. Benchmarking

In this secretion we provide some benchmarking results against other regression methods. The referenced results and benchmarking criteria are from [15]. It is difficult to compare with [17], [18], [19] directly since they adopted different experimental setup and evaluation criteria. However, empirical comparison shows the proposed method achieves comparable performance with them. For this process we divided the Mall dataset and UCSD dataset into sparse and crowded datasets as the paper specified. In the USCD dataset, 400 frames are used for training and 658 for testing in sparse scenario with less than 23 people in each frame while 400 frames for training and 542 for testing in crowded scenario with more than 23 people in each frame. In the Mall dataset, sparse scenario with less than 30 people in each frame contains 400 frames for training and 572 frames for testing. Crowded scenario with more than 30 people in each frame contains 400 frames for training and 628 frames for testing.

1) Sparse scenario: Besides our proposed methods, we also list the results for SVR with different features from [15] including segment, edge, GLCM, LBP, and two combined features S+E+GLCM (the first two features and GLCM) and S+E+LBP (the first two features plus LBP). In TABLE II, it is clear to see our method outperforms the SVR method with other individual and even combined features in the two datasets, showing the great effectiveness of the proposed algorithm.

	Train: Crowded - Test Sparse		
	Mean Abs. Error	Mean Sq. Error	Mean Dev. Error
LR	1.7448	4.8034	0.1013
PLSR	2.0208	6.2892	0.1170
KRR	2.0284	6.3176	0.1172
LSSVR	2.0123	6.2202	0.1163
GPR	2.3081	7.6730	0.1330
RFR	6.0851	50.5539	0.3882
Proposed	1.3450	3.7109	0.0744
	Train: Sparse - Test Crowded		
LR	2.8811	13.0382	0.0860
PLSR	4.0934	25.4034	0.1184
KRR	4.1805	26.4459	0.1210
LSSVR	4.2304	27.2070	0.1225
GPR	3.8089	20.6921	0.1119
RFR	9.4671	134.2994	0.2681
Proposed	3.9114	23.7048	0.1223

TABLE IV: Performance of the proposed methods in the USCD dataset under unseen density.

	Train: Crowded - Test Sparse		
	Mean Abs. Error	Mean Sq. Error	Mean Dev. Error
LR	5.4959	45.9012	0.2414
PLSR	4.9877	35.0432	0.2171
LR	5.1070	36.1893	0.2225
LSSVR	5.0216	35.2623	0.2189
GPR	5.4969	39.4660	0.2389
RFR	7.1080	64.0175	0.3127
Proposed	5.6906	39.2937	0.2387
	Train: Sparse - Test Crowded		
LR	4.5360	29.5379	0.1225
PLSR	5.6625	42.8628	0.1499
KRR	5.8006	44.0924	0.1534
LSSVR	5.7704	43.6109	0.1526
GPR	6.9426	59.8687	0.1835
RFR	8.6994	95.4601	0.2276
Proposed	5.5478	45.0478	0.1768

TABLE V: Performance of the proposed methods in the Mall dataset under unseen density.

2) Crowded scenario: The result for crowed scenario is shown in TABLE III. In the UCSD dataset, there is a dark region to the left of the region of interest that is visually distinct from the main color. This means any pedestrians in this region will be misclassified as the background, resulting slightly worse performance than other methods. However, it can still outperform the method with segment as feature. In the Mall dataset, considering the difficulty in correctly segmenting the image and the simplicity of feature adopted, the proposed method can still achieve comparable mean deviation error to the average benchmarking result.

3) Generalization to unseen density: The goal is to determine the performance of a method when tested under a density that it was not previously trained on. This means the method is trained on a sparse dataset then tested on a crowded dataset, and vice versa. [15] does not provide a performance comparison between different features for this category. However the performance using different regression models were provided. Note that the combined feature: segment, edge and LBP is adopted in [15].

The proposed method shows very promising result in the USCD dataset in TABLE IV. When trained in a crowded scenario, it can generalize the result to a sparse scenario. This method in fact performs better than all benchmarking methods in this category. The method does encounter difficulty in generalizing from sparse to crowded as other methods

do, this is most likely due to the region towards the left of the region of interest that could not be segmented properly. However, it still outperforms the method LSSVR which adopts complicated features and shares the same regression method. In the Mall dataset the proposed method is unable to generalize in both scenarios shown in TABLE V. This is due to difficulties in segmenting images with multiple background colors and lighting conditions changes. However, the proposed can still achieve excellent performance considering other regression methods apply more complicated features.

V. Conclusion and Future Work

In this paper, we proposed a regression based people counting method. This method took color segmentation result as input and divided the foreground image into blocks for effective counting. This method was fast and achieved robust performance. Experiments on three datasets showed that the proposed method achieved comparable results to other regression methods. Further study is to employ more sophisticated segmentation methods and combine more features to improve the results.

References

[1] Y.-L. Hou and G. K. Pang, "People counting and human detection in a challenging situation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 24–33, 2011.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, Jun. 2005, pp. 886–893.

[3] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on riemannian manifolds," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, Jun. 2007, pp. 1–8.

[4] M. Inen, M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, *Computer Vision Using Local Binary Patterns*. Springer, 2011, vol. 40.

[5] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral Channel Features," in *Proc. British Mach. Vis. Conf.*, 2009.

[6] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 32–39.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 4, no. 34, pp. 734–761, 2012.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2008, pp. 1–8.

[9] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *Proc. Int. Conf. Comput. Vis.*, 2009.

[10] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, June 2008, pp. 1–8.

[11] H. Rahmalan, "Application of invariant moments for crowd analysis," Jan. 2010.

[12] A. Marana, S. Velastin, L. Costa, and R. Lotufo, "Estimation of crowd density using image processing," in *IEE Colloquium on Image Processing for Security Applications*. IET, 1997, pp. 11–1.

[13] A. N. Marana, L. da Fontoura Costa, R. Lotufo, and S. A. Velastin, "Estimating crowd density with minkowski fractal dimension," in *Proc. IEEE Conf. Acoust., Speech, Signal Processing*, vol. 6, 1999, pp. 3521–3524.

[14] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, no. 6, pp. 610–621, 1973.

[15] C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*, ser. The International Series in Video Computing, 2013, vol. 11, pp. 347–382.

[16] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2013, pp. 2547–2554.

[17] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1324–1332.

[18] Y. Zhou and J. Luo, "A practical method for counting arbitrary target objects in arbitrary scenes," in *Proc. IEEE Conf. Multimedia and Expo*, 2013, pp. 1–6.

[19] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting." in *Proc. British Mach. Vis. Conf.*, vol. 1, no. 2, 2012, p. 3.

[20] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[21] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2010.

[22] N. Krahnstoever and P. R. Mendonca, "Bayesian autocalibration for surveillance," in *Proc. Int. Conf. Comput. Vis.*, vol. 2, 2005, pp. 1858–1865.

[23] J. Liu, R. T. Collins, and Y. Liu, "Surveillance camera autocalibration based on pedestrian height distributions," in *Proc. British Mach. Vis. Conf.*, 2011, p. 144.

[24] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 909–926, 2008.

[25] A. Chan and N. Vasconcelos, "Counting people with low-level features and bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, 2012.

[26] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, 2013, pp. 2467–2474.