Logarithmically Improved Property Regression for Crowd Counting

Usman Khan[1(✉)] and Reinhard Klette[2]

[1] Centre for Advanced Studies in Engineering, Islamabad, Pakistan
usmankhanakbar@gmail.com
[2] Auckland University of Technology, Auckland, New Zealand

Abstract. Crowd counting based on video camera recordings faces two major problems, namely inter-occlusion among the people, and perspective scaling. Though the former issue has been adequately addressed using different regression- and model-based schemes, a solution to the later problem remains an open problem so far. This paper proposes a novel scene-independent solution to perspective scaling. We show that it supports promising results. A property matrix, combining both a grey-level co-occurrence matrix and segmentation properties, is first obtained which is subsequently weighted using logarithmic relationships between pixel distances and foreground regions. We apply a Gaussian process regression, using a compounded kernel, to acquire an estimate for the crowd count. We show that results are comparable to those obtained when using more complex and costly techniques.

Keywords: Crowd counting · Perspective scaling · Grey-level co-occurrence matrix · Gaussian process

1 Introduction

Crowd density estimation of moving humans gains increasing attention in the field of video surveillance due to its importance for crowd statistics, monitoring, crowd control, and security-related observations.

Crowd counts are directly dependent upon pixel regions occupied by the crowd itself [8]; therefore, one commonly used step is *foreground extraction* using segmentation. This process is supported by having stationary cameras. A segmentation step is followed by relevant property or feature extraction tools. (A *property* is a measured value; a *feature* is defined by a location in an image and a property vector [16].) Subsequently, property regression is then used to map extracted properties into a crowd count.

Another technique used for crowd counting is a model-based template matching method [3] in which a template of a human head and shoulder moves across the scene. Most probable matches in the scene are declared as being humans. Both these methods (i.e. property regression and model-based crowd counting) are mainly focused on blob refinement and measurement.

T. Bräunl et al. (Eds.): PSIVT 2015, LNCS 9431, pp. 123–135, 2016.
DOI: 10.1007/978-3-319-29451-3_11

In the property-regression technique, perspective distortion is catered for by using two basic methods. The first method is scene-specific perspective correction based on a linear perspective-map formulation, whereas in the second method low-level features have been used. In both cases, the feature set or the property vector expand at a level where the complexity of property regression escalates exorbitantly, resulting in over-fitting. On the other hand, template scaling for the head-shoulder template-matching scheme is also confined to a specific scenario, and needs to be redefined for different scenes. Consequently, researchers have been adopting more complex methods to achieve better estimates.

The paper is structured as follows: Sect. 2 provides a brief review of related work. Section 3 outlines the proposed approach. Section 3.1 elaborates the usefulness of the selected textural feature set based on a *grey-level co-occurrence matrix* GLCM. In Sect. 3.2, techniques defined elsewhere used for segmenting extracted features, including the ViBe algorithm and the optical flow method, are briefly introduced. Section 3.3 proposes our novel approach adopted in this work for the correction of perspective distortion. Gaussian process regression, using a compounded kernel, is detailed in Sect. 4. Section 5 provides an experimental evaluation of the proposed approach, including a comparison of our algorithm with some benchmark techniques. Section 6 concludes.

2 Literature Review

The crowd estimation problem is mainly handled as a classification problem where crowd density is categorised as being low, medium, or high. A well-known example of crowd-density estimation [18] is by applying improved local binary patterns for obtaining a histogram of the considered crowd, for comparing it against model histograms by calculating their inter-distance. This supports categorising the crowd as being of very low, low, medium, high, or very high density. Crowd-count estimation, on the other hand, is a tedious problem which focuses mainly on accurately counting the number of people making up the given crowd. This problem is mainly tackled in one of the following three ways:

1. *Counting by detection* uses a visual object detector which segments individual objects of interest. Reference [11] shows a unique crowd-segmentation method which uses Fourier descriptors for shape indexing of crowd blobs while [22] uses a generic head detector to segment and count the number of people making up the crowd.
2. *Counting-by-regression* methods do not consider solving the task by a detection of individual objects. In these methods, the number of object counts is learnt mainly through a supervised learning methodology [21]. The main interest of this technique lies in the number of people determined by foreground pixels. Regression-based techniques involve background segmentation followed by foreground-property measurements (e.g. the area of regions, or texture characteristics) and then using linear, Bayesian, support vector machine (SVM), or neural-network regression techniques to learn the human

count. An example of regression counting is reference [6] where neural-network regression is applied.
Another distinguished technique is Chan's method; for example, see [18]. Here, the problem of counting people in a moving crowd is addressed by using a Poisson regression in a Bayesian framework over low-level features extracted from the images. A crowd count is taken as an outcome of a linear function by presenting a prior distribution on the weights of selected low-level features.
In [12], the perspective distortion problem has been addressed by assigning linear weights to the *scale-invariant feature transform* (SIFT) based on interest points at different locations in the image. These weights are assigned with respect to the ratio between height of a reference individual at two different locations in the video. Subsequently, Gaussian process regression is used to map interest points into a people count.

3. A *hybrid method* uses both detection and regression techniques. For example, [20] focuses on the fact that each blob in a crowd at a different perspective should be dealt with separately. Weights are assigned to each pixel based on a perspective plot. Multiple feature extraction is the second step after which each feature is assigned a different weight based on least-square regression. An *artificial neural network* (ANN) is used to obtain a number-of-people count in each blob based on distinct features and corresponding weights assigned.

3 Proposed Approach

Our approach focuses on crowd counting by regression. It presents a novel and natural solution to the problem of perspective distortion, thereby improving results by keeping time complexity of the regression algorithm at a low level.

For counting moving people, a set of textural and geometric segment features is extracted from each frame of a video sequence. Foreground segmentation of moving people is obtained using optical flow (for an original paper, see [15], or for a recent text on optic flow, see [16]) and a variant of the ViBe algorithm [2,5]. This allowed us to improve the handling of illumination invariance, "foreground erosion", and "stationary object blindness", all known as being issues in this area. Segmentation is further improved by using morphological dilation and erosion. A fundamental segmentation feature, which is directly proportional to the crowd count, is the foreground area. Logarithmic perspective correction is applied to the foreground blob area-property with respect to its centroid location in the image. Thereafter, this property set is mapped to the crowd count using a Gaussian process regression.

The proposed method outperforms methods presented in [1,9,10,12,20] in terms of simplicity of selected features, clustering method, perspective correction, and, altogether, better results.

3.1 Textural Feature Selection

Textural content descriptors provide useful information about distinguishing characteristics of an image such as coarseness, homogeneity, or smoothness.

Gonzalez [14] categorises approaches of textural analysis or synthesis into three main classes, called structural, spectral, or statistical. [23] added a model-based class to the three classes mentioned by Gonzalez. We decided for a statistical approach, and the textural features, chosen to obtain accurate estimates of crowd counts, are derived from a *grey-level co-occurrence matrix* (GLCM). A GLCM elaborates the frequency and combination of pixels with different brightness values (e.g., see the text [16]). Common GLCM features are defined with respect to contrast, homogeneity, energy, or correlation.

3.2 Geometric Feature Extraction for Segments

Foreground segmentation of the images is carried out by using two different sets of algorithms, including background subtraction and image foreground clustering algorithms. ViBe and optical flow are the two background subtraction algorithms, whereas *Gaussian mixture model* (GMM), k-means clustering, adaptive GMM, and *hidden Markov model and expectation maximisation* (HMM-EM) frameworks have been used to classify the foreground into distinct clusters. Experimentally it was observed that the background subtraction algorithms provided faster segmentation owing to lesser complexity and comparatively better results.

GMM Segmentation. GMM-based segmentation [17] distributes the image into a set of different classes, assuming that each class can be modelled as a normal distribution with a separate mean and variance. This assumption, though not accurate, provides a fairly reasonable segmented image as shown in Fig. 1, using an image from the PETs 2009 dataset [13].

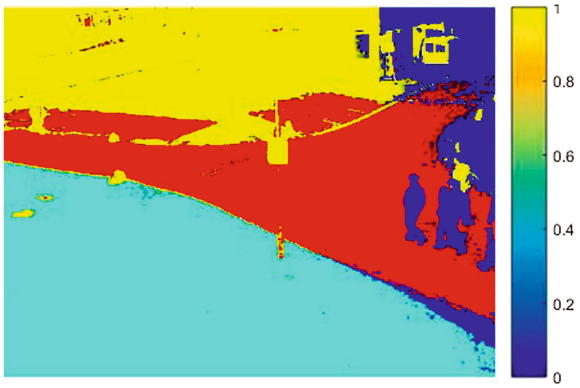

Fig. 1. Foreground segmentation using GMM for a sample from PETS 2009

Optimum result obtained for this sample of the PETs 2009 dataset is by using four classes. The algorithm does not need to be trained using background images. Segmentation results are obtained directly without the need of background subtraction. However, some of the background is still being classified as

human blobs. Another problem with the algorithm is that for each new scene, a number of classes is required to be set separately, and processing time increases with an increase in the number of classes.

ViBe Segmentation. A ViBe algorithm does not rely on one particular pixel distribution assumption such as for the GMM. It results in faster processing, provides illumination invariance by uniformly updating a background model, and has reduced mathematical complexity. Classification of a pixel as foreground or background in ViBe is based on the simple criteria of considering a disk of specified radius around the current pixel and determining the cardinality of the set of pixel values arising from intersecting the disk in the image with the disk at the same location in the background model.

Fig. 2. Foreground segmentation using a ViBe algorithm. *Left*: Input sample from PETS 2009. *Right*: Segmentation result using a ViBe algorithm

To ensure intensity invariance and ghost suppression, a background pixel model is updated randomly using a uniform distribution. The erosion of the foreground is suppressed by disallowing an already declared foreground pixel in the current frame to become a part of the background in the next frame. Finally, morphological operations [14], including opening, closing and hole filling, are used to ensure the desired level of foreground segmentation. See Fig. 2 for an example.

HMM-EM Algorithm Based on GMM and k-Means Clustering. Taking its lead from segmentation of human brain images, the HMM-EM framework proposed in [25] is an edge-prior preserving segmentation algorithm which was also used for obtaining accurate segmentation labels for the PETS-2009 dataset. In this algorithm, an initial segmented image is obtained using k-means clustering or GMM with estimated parameters including mean and variance. Subsequently, these estimates are then refined using the HMM-EM framework.

The algorithm was tested for a number of classes starting from 2 to 8. See Fig. 3 for an example.

Optical Flow Segmentation. For segmentation using optical flow, we apply the original Horn-Schunck algorithm [15]. Optic flow is the relative motion of

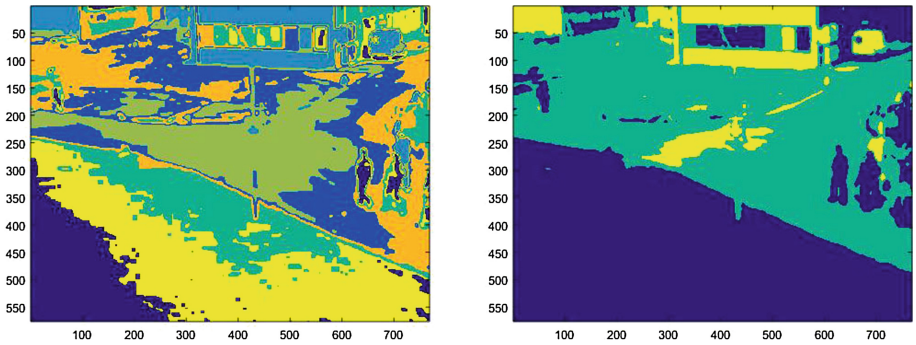

Fig. 3. Image segmentation using the HMM-EM framework. *Left*: Initial segmentation obtained using k-means clustering. *Right*: Initial segmentation obtained using GMM clustering

visible texture w.r.t. an observer in time and space. For calculating optic flow, the Horn-Schunck algorithm uses the *intensity constancy assumption* (ICA, see [16]) that reflectance is directly proportional to surface brightness, and also that surface brightness varies continuously with time. It is known that these assumptions are limiting the accuracy of calculated optic flow, especially the ICA is harmful. However, the Horn-Schunck algorithm is time-efficient and proved to be sufficient for our purpose.

Fig. 4. Foreground segmentation using optical flow. *Left*: Input sample from the PEDS dataset. *Right*: Segmented foreground

In our experiments with data of the PEDS database [24], foreground segmentation was obtained by applying a threshold to the minimum optical flow magnitudes at a pixel in both directions. We decided for a value of 3 as a threshold for flow velocities u and v in both x and y directions (i.e. $u > 3$ and $v > 3$); this produced the desired results. For an example, see Fig. 4.

Neither ViBe nor optical flow-based segmentation provide accurate segments of visible persons. However, this is also not required for the considered application as we will verify below.

3.3 Perspective Correction

Properties of fundamental importance, drawn from the segmented foreground, are blob area and perimeter. Due to the camera pose, an individual blob entering from one side of the image and traveling to the other side, suffers from shortening of both height and width which is a result of perspective distortion. This problem has been treated as a linear problem in previous work such as in [7,8,12,20] by taking into account height-to-distance ratios to cater for perspective distortion. However, a perspective linearization approach is not scene-independent, which means that for each new scenario, camera calibration is required again. Moreover, the formulation of an inherently nonlinear problem as a linear problem is an inappropriate simplification which causes errors in the final result.

Another method, which can be used to correct perspective distortion in images, is camera calibration [4]. This method involves a conversion from the 2D camera plane into the 3D real world. The complexity of the approach requires extra processing power despite followed optimisation efforts, thus making it unsuitable for real-time applications. This method has basically not been adopted into crowd-counting algorithms.

The *inverse square law* explains the relationship between the intensity of light and the distance of a point-light source from an object which is being illuminated by this light source. The *Weber-Fechner law* (identified as being a psycho-physical law) is another important relationship between human eye intensity response and contrast sensitivity. These two laws have been tested to automate perspective correction. Experiments have shown that a Weber-Fechner law implementation provides far better results compared to the inverse square law. The perspective scaling problem has been modelled in terms of logarithmic ratios as follows:

$$R_D = \sqrt{R_{IO}^2 + R_{IA}^2} \tag{1}$$

$$A_{corr} = \frac{A_{blob}}{\log C_{blob}} \log R_D \tag{2}$$

$$P_{corr} = \frac{P_{blob}}{\log C_{blob}} \log R_D \tag{3}$$

where, R_{IO} is the reference image ordinate, R_{IA} the reference image abscissa, R_D the reference distance, A_{blob} and P_{blob} are the blob area and perimeter. C_{blob} is the distance of the blob centroid from the reference point. Obtained corrected area and perimeter variables are A_{corr} and P_{corr}, respectively.

First a reference blob is selected. The distance from a specified reference point in the image, preferably from a corner point to the reference blob's centroid, is then used as the *reference distance*. In the second step, distances are calculated

from the selected reference point to centroids of the remaining blobs in the scene, and we call those *blob-centroid distances*. A logarithmic ratio between the reference distance and a blob-centroid distance provides a weighted factor which can scale blob area or perimeter, basically normalising a blob's size to the selected reference blob's centroid location. This is equivalent to shifting foreground blobs to the same location, thereby, removing perspective distortion (Fig. 5).

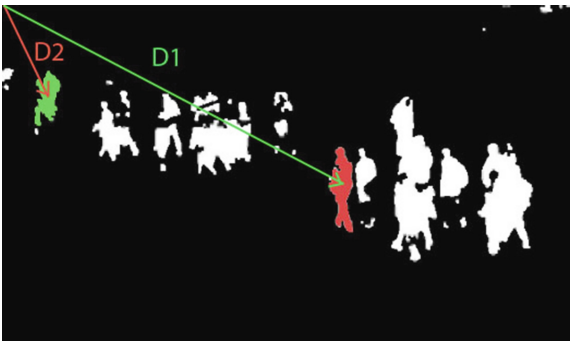

Fig. 5. D_1: Distance of the reference blob from the reference point, selected to be the image origin. D_2: Distance of another blob to the reference point selected to be the image origin. *Perspective correction*: The logarithmic ratio between D_1 and D_2 provides the required factor which is used for perimeter and area correction of the segmented blobs, i.e. virtually shifting blobs to the reference point.

4 Gaussian Process Regression with Compounded Kernel

After a representing property set is obtained for a given image I_i, it is mapped into a $D \times 1$ input vector $\mathbf{x}_i$, where D is the dimension of the collected property set. The goal is to design a function f which, if applied to the input property vector, provides the desired crowd count y_i for the given image [19]:

$$y_i = f(\mathbf{x}_i) + \varepsilon \tag{4}$$

with some error tolerance ε.

In generalisation, a set of vectors $\mathbf{x}_i$, obtained for a sequence $I_1, \ldots, I_n$ of n images, is collected into a matrix $\mathbf{X}$ of dimension $D \times n$, and we want to estimate

$$\mathbf{y} = [y_1, \ldots, y_n]^\top = f(\mathbf{X}) + \varepsilon \tag{5}$$

In case of linear regression, the function f can be expressed in terms of a weight vector as follows:

$$f(\mathbf{x}_i) = \mathbf{x}_i^\top \mathbf{w} \tag{6}$$

A Bayesian treatment of the above mentioned regression problem, applying a Gaussian distribution, assumes that noise is independent and identically distributed (i.e. Gaussian noise with zero mean and variance σ_n^2).

The likelihood of an observation is mutually independent, i.e. we have that

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) \tag{7}$$

The weight vector has zero mean with a covariance matrix $\mathbf{Q}$.

With these assumptions the posterior distribution of the weight vector $\mathbf{w}$ is as follows:

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \tag{8}$$

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = N(\mathbf{w} = \frac{1}{\sigma_n^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{y}, \mathbf{A}^{-1}) \tag{9}$$

$$\mathbf{A} = \sigma_n^2 \mathbf{X}\mathbf{X}^\top + \mathbf{Q} \tag{10}$$

Therefore, a predictive distribution of values of f, for a set of property vectors $\mathbf{x}$, can be written as follows:

$$p(f|\mathbf{x}, \mathbf{X}, \mathbf{y}) = N(\frac{1}{\sigma_n^2}\mathbf{x}^\top \mathbf{A}^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}^\top \mathbf{A}^{-1}\mathbf{x}) \tag{11}$$

The defined property-vector matrix $\mathbf{X}$ needs to be projected to a higher-dimensional property space using an unknown basis function $\phi_\mathbf{x}$ which converts a D-dimensional property vector into an N-dimensional array. Therefore, the model for f can be expressed in terms of $\phi_\mathbf{x}$ as follows:

$$p(f|\mathbf{x}, \mathbf{X}, \mathbf{y}) = N(\frac{1}{\sigma_n^2}\phi_\mathbf{x}^\top \mathbf{A}^{-1}\phi_\mathbf{X}\,\mathbf{y}, \phi_\mathbf{x}^\top \mathbf{A}^{-1}\phi_\mathbf{x}) \tag{12}$$

The basis function is unknown; however, fortunately there is no need to find the exact basis function. Instead the kernel trick is used to obtain the best possible approximation of a predictive distribution f. We suppose that $\mathbf{K}$ is the kernel matrix on the unknown basis function matrix $\phi_\mathbf{X}$. For the property vector basis function $\phi_\mathbf{x}$ it is represented as $\mathbf{k}$. With the help of kernel trick, instead of utilizing the property vector for functional prediction, approximation of the kernel is used instead. After a few mathematical manipulations of Eq. (12), $\mathbf{K}$ is represented as

$$\mathbf{K} = \phi_\mathbf{X}\,\mathbf{Q}\,\phi_\mathbf{X}^\top \tag{13}$$

$$\mathbf{k} = \phi_\mathbf{x}\,\mathbf{Q}\,\phi_\mathbf{x}^\top \tag{14}$$

Let $\mathbf{X}^*$ be the test set property matrix for the training set matrix $\mathbf{X}$, and let $\mathbf{I}$ be the identity matrix. The training and test set property matrices contain property vectors for each image arranged in columns. For the property matrices,

the predictive distribution of the output vector $\mathbf{f}$, containing the desired crowd count, can be represented as follows:

$$\begin{aligned} p(\mathbf{f}|\mathbf{x},\mathbf{X},\mathbf{y}) = N\big[&\mathbf{K}(\mathbf{X}^*,\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X}) \\ &+ \sigma_n^2\mathbf{I}]^{-1}\mathbf{y}, \mathbf{K}(\mathbf{X}^*,\mathbf{X}^*) \\ &- \mathbf{K}(\mathbf{X}^*,\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X}) \\ &+ \sigma_n^2\mathbf{I}]^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}^*)\big] \end{aligned} \tag{15}$$

The kernel function can be a single covariance function, or a combination of covariance functions. In this work, a compounded kernel function has been used. To cater for local and global nonlinear trends, a sum of constant, linear and exponential covariance functions has been used to formulate a compounded kernel:

$$k(x,x^*) = {\sigma_0}^2 + \sum_{i=1}^{D} \sigma_i^2 x_i x_i^* \exp \frac{-||x_i - x_i^*||}{2l^2} \tag{16}$$

In order to estimate the best possible hyper-parameters σ_0^2, σ_i^2, and $\bar{x}$, and the characteristic length scale l, maximum likelihood estimation is used by maximising the logarithmic function

$$\begin{aligned} \log p(\mathbf{f}|\mathbf{X}) = &-\frac{1}{2}\mathbf{y}^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \\ &- \frac{1}{2}\log|\mathbf{K} + \sigma_n^2\mathbf{I}| - \frac{n}{2}\log 2\pi \end{aligned} \tag{17}$$

Maximum likelihood estimation provides the best possible estimate for the unknown hyper parameters. For relevant details, see [19].

5 Experiments and Results

In order to evaluate the algorithm, experiments were conducted on two different datasets, the PETS 2009 dataset and the Peds-1 dataset, with an intention to establish scene-independence for the algorithm.

For both datasets, half of the frames have been used for training purpose whereas the other half has been used for testing. The training set of images has been annotated by manual counting. The testing part remained fully automated with no requirement of manual interaction for specifying a *region of interest* (ROI), to segment a scene into different regions, or any linear interpolation for scene-specific perspective corrections or separate identifications of pedestrians walking towards or away from the camera. This, altogether, characterises our algorithm as being automated in distinction to benchmark techniques published in [7–9,12].

Figure 6, left, reports about a comparison between estimated crowd count and ground truth using the Peds-1 dataset. The ground truth or actual crowd count is indicated by green markers whereas the estimated count is shown by red dots. For obtaining a better idea about acquired result validity, mean square

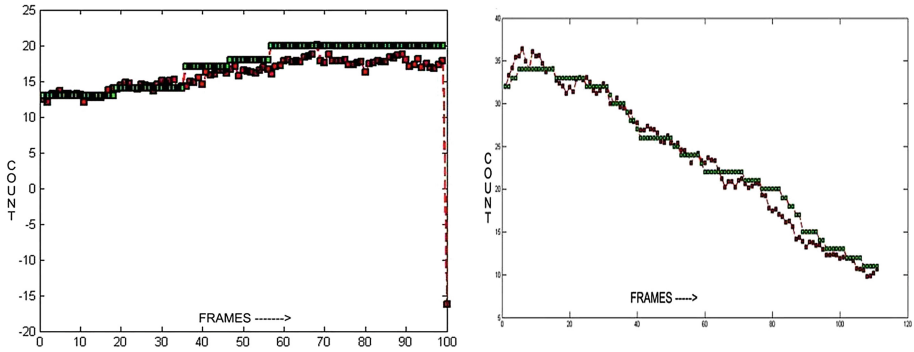

Fig. 6. Comparisons between ground truth and estimated counts (green dots represent ground truth, red dots represent estimated results). *Left*: PEDS dataset results. *Right*: PETS 2009 dataset results (Color figure online)

error, mean relative error, and mean absolute errors have been calculated using the proposed algorithm. The results are comparable to [7] but better than those reported in [12].

By using the logarithmic perspective scaling, *Gaussian process regression* (GPR) based crowd estimation results have improved compared to [7]. The minimum MSE obtained for the *scene motion class* in [7] is 3.654 using GPR, whereas by using the proposed algorithm, the MSE has been reduced to 3.093 which is comparable to 2.910, the result obtained when using a more complex *Bayesian Poisson regression* (BPR) method and a *digital terrain model* (DTM) segmentation in [7].

PET-2009 dataset results (i.e. manually annotated ground truth versus estimated count) are shown in Fig. 6, right. The red dots show estimated counts whereas the green dots represent the ground truth which has been manually annotated. The performance for the PETS-2009 dataset was also evaluated using mean-square error, mean absolute error, and mean relative error. See Table 1, which also summarises comparisons against benchmark methods.

Table 1. Examples of results

Error type	Mathematical notation	PETS 2009	Hajer Fradi	PEDS-1 (GPR)	*Chan's* (BPR)
MSE	$\frac{1}{N}\sum_{i=1}^{N}(EC_i - GT_i)^2$	1.422	-	3.093	2.910
MRE	$\frac{1}{N}\sum_{i=1}^{N}\frac{(EC_i - GT_i)^2}{GT_i}$	0.046	0.071	0.079	-
MAE	$\mathrm{A}\frac{1}{N}\sum_{i=1}^{N}\frac{\lvert EC_i - GT_i\rvert}{GT_i}$	0.951	1.38	1.458	1.308

6 Conclusions

This paper introduced a scene-independent perspective correction methodology for estimating numbers of moving crowds. Our overall algorithm, starting with foreground segmentation and ending with crowd estimation, is completely automated and caters for background and environment variance. The proposed algorithm is inherently parallel, thus implementable in FPGA or DSP hardware for real-time implementation. Image segmentation used in this approach does not cater for objects other than humans like a passing-by vehicle or large animals moving across the scene; such events may disturb the actual count. However, the problem can be catered using a suitable segmentation method with the ability to filter out unnecessary objects. The algorithm has been tested with the reported results on pedestrians moving on a road with negligible traffic, and can be recommended for comparable scenarios.

References

1. Albiol, A., Silla, M.J., Albiol, A., Mossi, V.: Video analysis using corner motion statistics. In: Proceedings of IEEE International Workshop PETS, pp. 31–37 (2009)
2. Barnich, O., Droogenbroeck, M.: ViBe: A universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. **20**, 1709–1724 (2011)
3. Fu, H., Ma, H., Xiao, H.: Real-time accurate crowd counting based on RGB-d information. In: Proceedings of ICIP, pp. 2685–2688 (2012)
4. Beardsley, P., Murray, D.: Camera calibration using vanishing points. Int. J. Comput. Vis. **4**, 127–139 (1992)
5. Barnich, O., Droogenbroeck, M.M.V., Paquot, O.: Background subtraction: experiments and improvements for ViBe. In: Proceedings of CVPR Workshop Change Detection, pp. 32–37 (2012)
6. Garcia-Bunster, G., Torres-Torriti, M., Oberli, C.: Crowded pedestrian counting at bus stops from perspective transformations of foreground areas. Comput. Vis. IET **6**, 296–305 (2012)
7. Chan, A., Vasconcelos, N.: Counting people with low-level features and Bayesian regression. IEEE Trans. Image Process. **21**, 2160–2177 (2012)
8. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: counting people without people models or tracking. In: Proceedings Computer Vision Pattern Recognition, pp. 1–7 (2008)
9. Chan, A.B., Morrow, M., Vasconcelos, N.: Analysis of crowded scenes using holistic properties. In: IEEE International Workshop Performance Evaluation Tracking Surveillance (2009)
10. Conte, D., Foggia, P., Percannella, G., Tufano, F., Vento, M.: A method for counting people in crowded scenes. In: Proceedings of IEEE International Conference Advanced Video Signal Based Surveillance, pp. 225–232 (2010)
11. Dong, L., Parameswaran, V., Ramesh, V., Zoghlami, I.: Fast crowd segmentation using shape indexing. In: Proceedings International Conference Computer Vision, pp. 1–8 (2007)
12. Fradi, H., Dugelay, J.L.: People counting system in crowded scenes based on feature regression. In: Proceedings of European Signal Processing Conference, pp. 27–31 (2012)

13. Fraile, R.: IEEE International Workshop Performance Evaluation Tracking Surveillance (PETS 2009) (2009). www.cvg.reading.ac.uk/PETS2009/
14. Gonzales, R.C., Wintz, P.: Digital Image Processing, 3rd edn. Prentice Hall, Upper Saddle River (2009)
15. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
16. Klette, R.: Concise Computer Vision. Springer, London (2014)
17. Lalit, G., Thotsapon, S.: A Gaussian-mixture-based image segmentation algorithm. Pattern Recogn. **31**(3), 315–325 (1998)
18. Mousavi, S.M., Shahdi, S.O., Abu-Bakar, S.A.R.: Crowd estimation using histogram model classification based on improved uniform local binary pattern. Int. J. Comput. Electr. Eng. **4**, 256–259 (2012)
19. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
20. Ryan, D., Denman, S., Fookes, C.: Crowed counting using multiple local features. In: Proceedings of Digital Image Computing: Techniques Applications, pp. 81–88 (2009)
21. Shimosaka, M., Masuda, S., Fukui, R., Mori, T., Sato, T.: Counting pedestrians in crowded scenes with efficient sparse learning. In: Proceedings of Asian Conference Pattern Recognition, pp. 27–31 (2011)
22. Subburaman, V.B., Descamps, A., Carincotte, C.: Counting people in the crowd using a generic head detector. In: Proceedings of IEEE International Conference Advanced Video Signal-Based Surveillance, pp. 470–475 (2012)
23. Tuceryan, M., Jain, A.K.: Texture analysis. In: The Handbook of Pattern Recognition and Computer Vision, pp. 207–248 (1998)
24. UCSD: UCSD Anomaly Detection Dataset (2013). http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm
25. Wang, Q.: HMRF-EM-image: implementation of the hidden Markov random field model and its expectation-maximization algorithm, arXiv: 1207.3510 [cs.CV] (2012)