Cross Camera People Counting with Perspective Estimation and Occlusion Handling

Tsung-Yi Lin[1], Yen-Yu Lin[2], Ming-Fang Weng[3], Yu-Chiang Wang[2], Yu-Feng Hsu[4], Hong-Yuan Mark Liao[3]

[1] *Department of Electrical and Computer Engineering, University of California, San Diego, USA*
tsl008@ucsd.edu

[2] *Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan*
{yylin, ycwang}@citi.sinica.edu.tw

[3] *Institute of Information Science, Academia Sinica, Taipei, Taiwan*
{mfueng, liao}@iis.sinica.edu.tw

[4] *Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, Taiwan*
yufenghsu@itri.org.tw

***Abstract*—We introduce a novel approach to cross camera people counting that can adapt itself to a new environment without the need of manual inspection. The proposed counting model is composed of a pair of collaborative *Gaussian processes* (GP), which are respectively designed to count people by taking the *visible* and *occluded* parts into account. While the first GP exploits multiple visual features to result in better accuracy, the second GP instead investigates the conflicts among these features to recover the underestimate caused by occlusions. Our contributions are threefold. First, we establish a cross camera people counting system that can facilitate forensics investigation and security preservation. Second, a principled way is proposed to estimate the degree of occlusions. Third, our system is comprehensively evaluated on two benchmark datasets. The promising performance demonstrates the effectiveness of our system.**

I. Introduction

Video surveillance plays a critical role in a broad range of applications related to information forensics and security, such as law enforcement, abnormality detection, and crime prevention. Due to the recent advances in surveillance devices, huge volume of data taken by these devices has made surveillance by manual effort impossible [1], [2]. Hence there is a strong demand for automatic video surveillance systems, so that the data fragments of interest can be efficiently mined. *People counting* is one of the most important components in establishing these systems, since people counts serve as an important index for surveillance. For example, one might retrieve groups of people of a certain amount for law enforcement, monitor the degree of crowdedness for online traffic control, or detect abnormal gathering for crime prevention.

In this paper, we introduce a new approach to *cross camera* people counting. That is, the counting model learned with video data taken by some particular cameras can be adaptively applied to data taken by other cameras. We consider such flexibility valuable and practical, since in general cases we have no idea about where a camera is located and what its perspective setting is.

WIFS'2011, November 29th-December 2nd, 2011, Foz do Iguaçu, Brazil. 978-1-4577-1019-3/11/$26.00

Two major obstacles to developing cross camera people counting systems are: 1) the high diversity of the pedestrian scales and appearances; and 2) the degree of crowdedness varies from environment to environment. While the former results in the inconsistency and large variations of the extracted features, the latter leads to different degrees of partial occlusions. In this paper, we address the two unfavorable issues and propose an algorithm, called *two-pass Gaussian processes* (TPGP), for carrying out cross camera people counting. We measure the number of people in a video frame by considering two sources, the *visible* part and the *occluded* part of people. TPGP is composed of a pair of collaborative *Gaussian processes* (GP) [3], which are designed to estimate the numbers of people in the two parts respectively.

The first-pass GP aims to count people by taking the visible part of pedestrians into account. To address the variations of pedestrian scale and appearance resulting from heterogeneous cameras, we develop a method for camera perspective estimation and propose a mechanism for automatic training data collection. By properly integrating the two procedures, both the scale and appearance variations are significantly alleviated. Learning with the *calibrated* features, cross camera people counting is carried out in the first-pass GP.

The second-pass GP is designed to compensate for the underestimate in the first pass caused by occlusions. The underlying difficulty is that no visual features can be exploited from the occluded (unobservable) parts. We propose to estimate the degree of occlusions via analyzing *the differences among the predictions by diverse features*. It is motivated by the observation illustrated in Fig. 1. The blue, green, and red lines respectively indicate the numbers of people predicted by the feature of horizontal gradients, feature of vertical gradients, and ground truth. The two features lead to similar prediction when no occlusions occur. However, due to their difference in sensitivities to occlusions, the performance gaps become larger as the degree of occlusions increases. That is, the *extent of disagreement* between the two features reveals useful evidences for recovering the underestimate by occlusions, even though none of them can individually handle occlusions well.

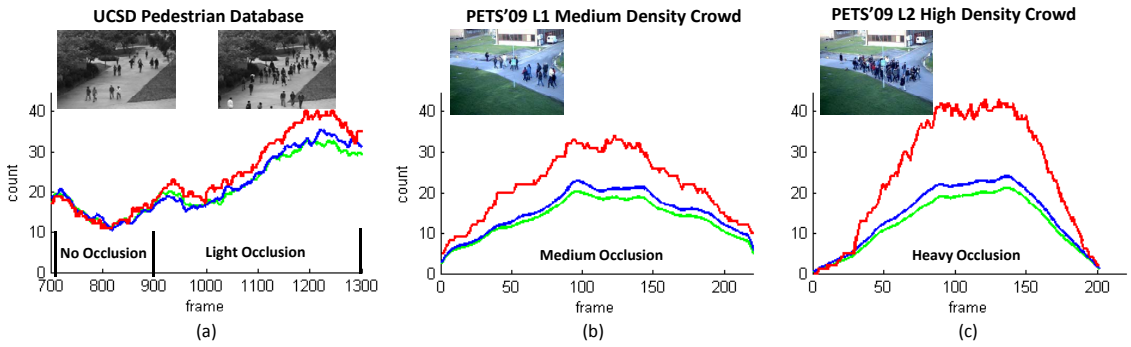


Fig. 1. Video sequences with different degrees of occlusions: (a) no occlusions and light occlusions, (b) medium occlusions, and (c) heavy occlusions. Red line gives the numbers of people in ground truth. The blue and green lines respectively indicate the numbers estimated by the two features with different levels of robustness against occlusions. The extent of their disagreement in prediction reveals useful evidences for recovering the underestimate by occlusions.

This work distinguishes itself with the following three main contributions. First, a cross camera people counting framework is established to facilitate forensics investigation and security preservation. Second, a principled way is proposed to estimate the degree of occlusions by investigating the prediction differences among diverse features. Third, our system is comprehensively evaluated on two benchmark datasets, PETS09 Database [4] and UCSD Pedestrian Database [5]. The promising performance demonstrates the effectiveness of our system.

II. Related Work

The literature about people counting is quite comprehensive. Therefore we review only some of the works closely related to our approach, and divide them into the following two categories for the ease of discussion.

A. Counting by Pedestrian Localization

People counting systems of this category work via explicitly locating the position of each person and then counting. Some of them, e.g., [6], [7], illustrate that the object (and people) counting problem can be transformed and solved via evaluating the number of clusters grouped by the similarity of texture [6] or the motion pattern [7]. However, the appearance or shape of objects cannot be clearly defined in advance. They may result in the limited performance. The trend of algorithms of this category is to employ a pedestrian detector, such as [8], [9], to count the number of people in images. These pedestrian detectors have been demonstrated that they are robust to variations caused by different illumination and poses. In addition, the performance of detection can be farther improved by modeling the geometric relations between body parts and whole body [10], [11]. The above mentioned detectors are trained with datasets where samples are of a high resolution and no occlusions are involved. Thus the performance of these detectors will severely degrade, when they are applied to find pedestrians with partial occlusions or of low resolutions. Besides, the computational cost in detection is too high to support real-time people counting.

B. Counting by Feature Regression

Methods of this category, such as [5], [12], [13], [14], [15], [16], [17], [18], estimate the numbers of people in images via extracting features from regions of crowd, which is typically yielded by background subtraction or motion segmentation. Since these methods do not need to solve the hard localization problem, they are suitable for crowd counting, especially for cases where partial occlusions occur or low resolution cameras are used. In [12], [13], [16], a set of perspectively normalized features is introduced, and the values of these features are assumed to be linear with respect to the number of people in an image. Despite the high applicability of these features, the linearity assumption fails as occlusions occur. In [14], [5], [15], some powerful regression models, e.g., neuron networks, Gaussian process, and Poisson process, are adopted to alleviate the problem by taking training data with occlusions into account. These methods however require a large number of labeled training data over different degrees of crowdedness. Further, the built model cannot achieve cross camera people counting, because the relationship between feature values and the degree of the crowdedness is highly camera-dependent.

The proposed counting system belongs to the second category. However, we develop automatic mechanisms for eliminating unfavorable variations induced in heterogenous cameras. We also introduce a novel algorithm for measuring the degree of occlusions, and it makes no assumption about environment-specific properties. Thus our system can achieve cross camera people counting with high precision.

III. The Proposed TPGP Framework

In the section, we first specify the mechanisms for handling variations of scales and appearances in heterogeneous cameras, and then present the TPGP framework. An overview of the proposed system is given in Fig. 2.

A. Cross Camera Scale and Appearance Manipulation

1) Blob Representation: We represent a video frame I by a set of *blobs*, each of which is a group of spatially-connected foreground pixels. These blobs can be obtained by applying the background subtraction algorithm [19] to I and segmenting the

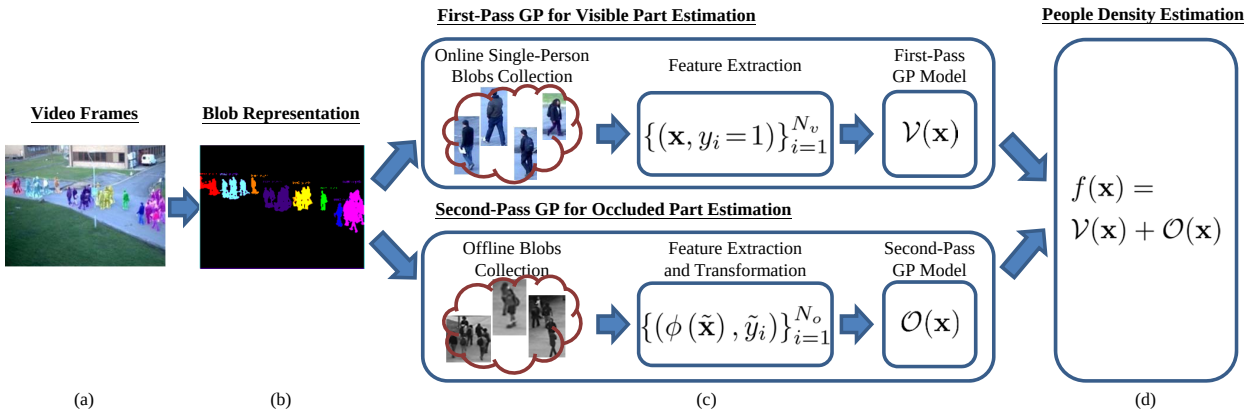


Fig. 2. The overview of the proposed TPGP framework. (a) A video frame. (b) Its blob representation. (c) & (d) The learning and the counting procedures of the two-pass GPs.

resulting foreground area into a set of connected components. Suppose total N blobs are extracted from I and each blob is described by M kinds of different features. Then image I can be expressed as $\{(\mathbf{x}_i \in \mathbb{R}^M, y_i \in \mathbb{R})\}_{i=1}^N$, where y_i denotes the number of people in the blob. In the training phase, component y_i is given, and in testing y_i is what we need to evaluate. With this representation, the number of people in I is calculated by summing over the ones estimated in the blobs. An example of the blob representation is given in Fig. 2b.

We consider blob as a natural unit for people counting, as each pedestrian typically appears within one blob, while a pedestrian can be occluded by other pedestrians residing in the same blob. Thus not only the scale and appearance normalization but also the occlusion evaluation can be performed blob-wisely. In our approach, we assume each feature adopted to characterize a blob can be applied to each pixel in the blob, and the feature value of a blob can be calculated by summing over the responses on pixels belonging to the blob. Features of this sort are proven to be very effective in people counting [5], [17], [18]. We leave the implementation details of the used features in Sec IV-A.

2) Perspective Modeling : Due to the effect of perspective, pedestrians closer to the camera appear larger in the image. In addition, each camera has its own settings, such as its focus length and pose. We hence need to adaptively estimate the camera perspective for cross camera people counting. To this end, the model in [20] is adopted to estimate a pedestrian's image height h given his/her bottom position v by

$$h(v) = \frac{l}{l_c}(v_0 - v), \tag{1}$$

where l_c and v_0 denote the camera height and the horizon position respectively, and l is the 3D height of the pedestrian. As the ratio of l and l_c can be treated as a constant c, parameters c and v_0 can be obtained by manually drawing the bounding boxes of two reference pedestrians. With the model, the scale of a person at an arbitrary position of the image can be estimated. The resulting *perspective normalization map* $1/h^2(v)$ is used to weight features for scale normalization.

3) Online Training Data Acquisition : The appearances of people taken by heterogeneous cameras span a wide spectrum due to factors ranging from the intrinsic properties of cameras, e.g., resolutions and image qualities, to the extrinsic environment conditions, e.g., illumination. These variations make features extracted from different cameras quite inconsistent. Hence it is infeasible to directly apply the counting model learned from one camera to another even with perspective normalization. To address this issue, we propose an online mechanism that can adaptively collect training samples and learn the counting model for an arbitrary video sequence.

Specifically, we are inspired by the fact that nowadays pedestrian detectors, such as [8] and [21], can locate people in images against these variations, because they are derived with huge amounts of training examples. Based upon an off-the-shelf pedestrian detector, we develop a coarse-to-fine algorithm (which will be described in detail in Sec IV-B) to automatically collect the training data (pedestrians) with high precision. The detected pedestrians can be regarded as a special case of the blob representation, since each of the resulting blobs contains exactly one person. Imagine that the camera is located in a new environment. After collecting sufficient blobs, say N, i.e., $D = \{(\mathbf{x}_i, y_i = 1)\}_{i=1}^N$, we are ready to learn a people counting system that accounts for the appearance variations via analyzing evidences revealed in D.

B. The TPGP Algorithm

For a given blob $\mathbf{x}$, the proposed TPGP framework will regress the number of people in this blob by

$$f(\mathbf{x}) = \mathcal{V}(\mathbf{x}) + \mathcal{O}(\mathbf{x}), \tag{2}$$

where $\mathcal{V}(\mathbf{x})$ and $\mathcal{O}(\mathbf{x})$ are designed to measure the amounts of people in the *visible* and *occluded* parts in the blobs respectively. Each of function $\mathcal{V}$ and $\mathcal{O}$ is accomplished by a Gaussian process regressor [3] in the work. Since the features of visible parts change from camera to camera, the first-pass GP $\mathcal{V}$ is learned by the automatically collected data. That is,

GP $\mathcal{V}$ is *camera-dependent*, and it utilizes the *complement* of various features to boost the prediction accuracy. Relative to $\mathcal{V}$, the second-pass GP $\mathcal{O}$ investigates the *diversity* of these features in the aspect of their robustness against occlusions, and recovers the underestimate in occluded parts. When the abilities of features against occlusions are assumed to be fixed, the GP $\mathcal{O}$ is only *feature-dependent*. It follows that GP $\mathcal{O}$ can be learned with data collected *offline*, since the adopted features remain unchanged. To better explain the above concept, we will briefly introduce how Gaussian process regression works before describing the design of the two-pass Gaussian processes.

1) Gaussian Process Regression: Gaussian process regression (GPR) adopts a Bayesian treatment in predicting the target value $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ of an input vector $\mathbf{x}$. With a set of training data $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, GPR models the posterior distribution of weight vector $\mathbf{w}$ and regresses a novel sample $\mathbf{x}$ as

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{y}^\top (K + \sigma^2 I_N)^{-1} \mathbf{k}(\mathbf{x}), \qquad (3)\\ \text{where } \mathbf{y} &= [y_1 \;\; y_2 \;\; \cdots \;\; y_N]^\top \in \mathbb{R}^N,\\ K &= [k(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}^{N\times N},\\ \mathbf{k}(\mathbf{x}) &= [k(\mathbf{x}, \mathbf{x}_1) \;\; k(\mathbf{x}, \mathbf{x}_2) \;\; \cdots \;\; k(\mathbf{x}, \mathbf{x}_N)]^\top \in \mathbb{R}^N, \end{aligned}$$

$I_N \in \mathbb{R}^{N\times N}$ is the identity matrix, and σ is a constant.

Gaussian process, like SVM, is a kernel machine. The inner product of each pair of data embedded in some high-dimensional feature space can be efficiently computed via kernel functions, such as

- *Linear kernel*: $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$.
- *RBF kernel*: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2)$.
- *Polynomial kernel*: $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^c$.

2) On Designing the First-Pass GP $\mathcal{V}(\mathbf{x})$ *:* The first-pass GP $\mathcal{V}(\mathbf{x})$ aims to count the number of people through their visible parts. For dealing with the scale and appearance variations of visible parts taken by heterogeneous cameras, the training dataset, $D_v = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_v}$, of $\mathcal{V}(\mathbf{x})$ is collected adaptively via the procedure described in Sec III-A3. Based upon the nature of the sliding-window detector, each blob $\mathbf{x}_i$ is composed of exactly one person (i.e., $y_i = 1$) and contains no occluded areas, which will be accounted for in the second-pass GP. With D_v, the first-pass GP is then established

$$\mathcal{V}(\mathbf{x}) = \mathbf{w}_v^\top \mathbf{k}_v(\mathbf{x}). \qquad (4)$$

Refer to (3) for the formulation of $\mathbf{w}_v^\top \mathbf{k}_v(\mathbf{x})$. The linear kernel function is used in the construction of $\mathbf{w}_v$ and $\mathbf{k}_v(\mathbf{x})$.

Note that although each blob in D_v consists of only one person, the learned GP $\mathcal{V}(\mathbf{x})$ could be used to predict blobs of more than one person. This is because the values of the adopted features are calculated by summing over pixels belonging to the blob. Thus if no partial occlusions occur, feature values are expected to grow linearly with respect to the number of people in the blob. Gaussian process with a linear kernel can nicely couple with this idea, since it can be checked that $c\mathcal{V}(\mathbf{x}) = \mathcal{V}(c\mathbf{x})$ for any constant c.

3) On Designing the Second-Pass GP $\mathcal{O}(\mathbf{x})$*:* The second-pass GP $\mathcal{O}$ is designed to deal with the underestimate caused by occlusions. As shown in Fig. 1, we observe the differences of two features in their abilities of accounting for occlusions. Although none of these features can well handle occlusions, the underestimate can be approximated by using *extrapolation* method with respect to the outputs of different features.

Suppose a set of blobs, $D_o = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_o}$, is given. Note that each of these blobs here can contain either one person or multiple people, and occlusions may or may not occur. We begin the design of $\mathcal{O}$ by introducing a term called *occlusion rate* for applying $\mathcal{V}$ to each blob $(\mathbf{x}_i, y_i)$, i.e., ,

$$\tilde{y}_i = \frac{y_i - \mathcal{V}(\mathbf{x}_i)}{y_i}. \qquad (5)$$

Then each feature d is normalized such that different features will have similar scales:

$$\tilde{\mathbf{x}}_i(d) = \frac{\mathbf{x}_i(d)}{\mathbf{m}(d)}, \text{ where } \mathbf{m} = \frac{\sum_{i=1}^{N_o} \mathbf{x}_i}{\sum_{i=1}^{N_o} y_i}, \qquad (6)$$

where $\tilde{\mathbf{x}}_i(d)$ denotes the dth dimension of vector $\tilde{\mathbf{x}}$, and $\mathbf{x}_i(d)$ and $\mathbf{m}(d)$ are similarly defined.

To implement the extrapolation of the outputs of various features, a feature transformation is carried out for each $\tilde{\mathbf{x}}_i$ by

$$\phi(\tilde{\mathbf{x}}_i) = [|\tilde{\mathbf{x}}_i(d_1) - \tilde{\mathbf{x}}_i(d_2)|], \text{ for } 1 \le d_1 < d_2 \le M, \qquad (7)$$

where M is the number of the used features.

With the transformed data $\{(\phi(\tilde{\mathbf{x}}_i), \tilde{y}_i)\}_{i=1}^{N_o}$, a Gaussian process regressor can be learned. Note that what we learn is the occlusion rate, instead of the underestimate caused by occlusions. Thus with the definition of (5), if the regression value of a given blob $\mathbf{x}$ is $\tilde{y}$, its output in this pass would be

$$\mathcal{O}(\mathbf{x}) = \frac{\tilde{y}}{1 - \tilde{y}} \mathcal{V}(\mathbf{x}). \qquad (8)$$

Considering the proposed two-pass procedure in (2), the estimated number of people in novel blob $\mathbf{x}$ will be

$$f(\mathbf{x}) = \mathcal{V}(\mathbf{x}) + \mathcal{O}(\mathbf{x}) = \frac{1}{1 - \tilde{y}} \mathcal{V}(\mathbf{x}). \qquad (9)$$

IV. Implementation Details

In this section we are going to detail the implementation of two components, i.e., feature extraction and pedestrian detection, that we have adopted in our system.

A. Feature Extraction

Many features such as segmentation area [5] and number of interest points [22] have been proven to yield promising results on vision-based people counting. However, to make the proposed TPGP algorithm work effectively, we prefer to use a set of features which varies linearly with the number of people in blobs when there are no serious occlusions between people, as well as lead to inconsistent estimates of crowd density when occlusions occur. Thus, we select the following three representative features to characterize the properties of blobs in our current implementation.

TABLE I
SUMMARY OF EXPERIMENTAL SETUPS AND PERFORMANCE COMPARISON OF OUR TPGP WITH TWO OTHER APPROACHES.

	Occlusion Handling				Cross Camera Settings	
	S1-S2	S2-S3	S3-S2	train-test (UCSD)	UCSD-PETS09	PETS09-UCSD
Training data	PETS09.S1	PETS09.S2	PETS09.S3	UCSD.train	UCSD.train	PETS09.S2
Testing data	PETS09.S2	PETS09.S3	PETS09.S2	UCSD.test	PETS09.S2	UCSD
Chan et al. [15]	2.31	—	—	—	—	—
Kong et al. [14]	—	—	—	2.07	—	—
Ryan et al. [16]	—	—	—	1.53	—	—
Lempitsky et al. [18]	—	—	—	1.70	—	—
MAGP	2.49	5.50	7.12	**1.42**	22.42	15.81
FPGP	6.65	11.60	6.65	2.81	6.65	3.28
TPGP	**1.71**	**4.20**	**1.62**	1.50	**3.90**	**2.39**

Area: total number of foreground pixels occupied by the blob, roughly reflecting the volume size of moving objects in the scene.

Canny edge pixels: total number of edge pixels, located by Canny edge detector, contained in the blob, capturing the structural properties of crowdedness.

Gradient orientations: a six-bin histogram of oriented gradients (HOG), representing the gradient magnitudes of six orientations evenly spaced from 0° to 360°.

By concatenating all the extracted features, each crowd blob is thus represented by an eight-dimensional feature vector. To correct the perspective distortion in 2D images, the pixels in the features are normalized by applying the geometric distortion correction model described in Sec III-A2 to approximate the real scales in 3D scenes.

B. Pedestrian Detection

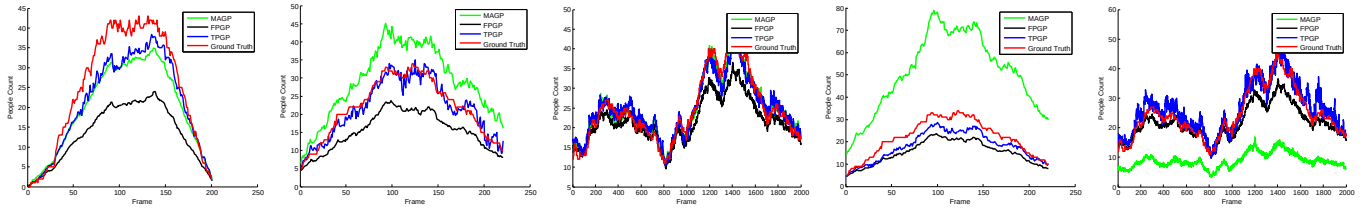
To cope with the issue of variations in image resolution, we use HOG features and an SVM to build three pedestrian detectors which are suitable to classify 16×32, 32×64, and 48×96 detection widows, respectively. Similar to the configurations suggested by Dalal and Triggs [8], we divided each window into 8×16 cells and each cell consists of either 2×2, 4×4, or 6×6 pixels depending on the window sizes. Therefore, each window has 105 blocks and is represented by a $3{,}780$-dimensional vector. At training stage, a linear SVM model with probabilistic outputs is learned from a set of manually labeled examples. During detection, we crop windows with proper width and height based on the perspective prior in a sliding fashion over the images and ignore those windows contain less than 50% foreground pixels for efficiency. Each extracted window is then resized to the nearest supported resolutions; its HOG feature is extracted and a probabilistic score is generated to indicate the possibility of the window containing a pedestrian. We obtain the final detection results by selecting the window with the maximum probability if overlaps occur.

V. EXPERIMENTAL RESULTS

We conducted experiments on two public data sets, PETS09 Database [4] and UCSD Pedestrian Database [5], each of which consists of videos collected from stationary camcorders with various crowd densities. The PETS09 set contains three video sequences, denoted as S1, S2, and S3. While S1 and S2 represent the videos with normally crowded groups of moving people (i.e., with medium occlusions), S3 is the one with highly crowded groups (i.e., with heavy occlusions). For UCSD set, we follow the protocol by Chan et al. [5], and divide the dataset into a training set (UCSD.train) of 800 frames and a test set (UCSD.test) of $1{,}200$ frames. Due to the high diversity of camera poses, frame sizes, and monitoring scenes, the videos in the two datasets indeed provide a good test bed to measure the effectiveness of our method on cross camera settings. In the following, we focus on evaluating the performance of TPGP on two aspects, occlusion handling and cross camera counting. The details of these setups are specified in Table I. In the experiments, we use the ground truth as well as the regions of interest provided in the released packages.

We compare our approach TPGP with two baselines, i.e., MAGP (manually annotated GP) and FPGP (first-pass GP) respectively. While MAGP learns regression models using manually annotated training data, FPGP is what we describe in Sec III-B2. Comparing with MAGP explores the effectiveness of our approach in cross camera counting, while comparing with FPGP probes the effectiveness in occlusion handling. We adopt *mean absolute error* (MAE) as the criterion for performance measure. The quantitative results of MAGP, FPGP, and TPGP, together with the ones of the state-of-the-art systems [14], [15], [16], [18], are reported in Table I.

From Table I, it can be observed that our approach TPGP outperforms both the MAGP and FPGP in most cases. The only exception is MAGP on the UCSD dataset. That is because only light occlusions appear in the dataset, and there is no significant difference between the training and testing data. From the quantitative results, the proposed approach shows good ability on occlusion handling with a wide range of extents of occlusions. In particular, our TPGP yields the predictions which are very close to the ground truth in the S3-S2 experiment, whereas the other two baselines do not work well. It demonstrates the effectiveness of our method in occlusion handling via exploring the conflicts among different features. Furthermore, since MAGP cannot handle the cases



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

Fig. 3. Performance comparison among TPGP, MAGP, and TPGP on PETS09 and UCSD data sets. Experiments of occlusion handling: (a) S2-S3, (b) S3-S2, and (c) train-test(UCSD). Experiments of cross camera counting: (d) UCSD-PETS09 and (e) PETS09-UCSD.

where the training and test data do not share similar properties, it can only be applied to few scenarios. By contrast, as we exploit both the camera-independent features from training data and camera-dependent features on testing environment, our approach indeed accomplishes cross camera counting.

Fig. 3 plots the frame-by-frame predictions on a few experiments, yielded by MAGP, FPGP, and TPGP along with the ground truth. We observed that the improvement on PETS09 (Fig. 3(a)–(b)) are more significant than the one on UCSD (Fig. 3(c). Since people are seriously occluded in PETS09 and our TPGP method is able to adequately handle occlusions by estimating the occlusion rate, we can provide accurate estimation against these variations under various degrees of crowdedness. When occlusions occur, FPGP generally underestimates; however, this estimation is successfully compensated by the second-pass GP of the proposed method.

As illustrated in Fig. 3(d)–(e), TPGP also yields better performance than MAGP and FPGP in cross camera settings. Although FPGP can automatically collect the training data from the target camera to adaptively learn models, it is limited in dealing with the occlusion issues since the pedestrian detector only finds non-occluded subjects. By contrast, TPGP is able to simultaneously handle both the occlusion and appearance variations, thus achieving 41% $((6.65-3.90)/6.65)$ and 27% $((3.28-2.39)/3.28)$ relative improvement over FPGP on PETS09 and UCSD respectively.

VI. Conclusions

We have presented an effective framework TPGP that adapts itself to the variations resulting from heterogenous cameras, and carries out cross camera people counting. In TPGP, the numbers of people in the visible and occluded parts of pedestrians are respectively estimated by a couple of collaborative Gaussian processes. The proposed approach is comprehensively evaluated on two challenging benchmark databases. The promising experimental results consolidate the usefulness of our approach. The established system distinguishes itself with four important properties: automatic training data collection, occlusion manipulation, real-time computation, and cross camera counting. These properties make it quite suitable for a broad range of applications related to forensics investigation and security preservation.

Acknowledgments. The work is supported in part by grants NSC 100-2218-E-001-004, NSC 97-2221-E-001-019-MY3, and ITRI project A301ARY220.

References

[1] K. Franke and S. Srihari, "Computational forensics: An overview," in *Int'l Workshop Computational Forensics*, 2008.
[2] M. Worring and R. Cucchiara, "Multimedia in forensics," in *ACM Conf. Multimedia*, 2009.
[3] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[4] PETS'09 Database, "http://www.cvg.rdg.ac.uk/PETS2009/."
[5] A. Chan, Z. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Computer Vision and Pattern Recognition*, 2008.
[6] N. Ahuja and S. Todorovic, "Extracting texels in 2.1d natural textures," in *Int'l Conf. Computer Vision*, 2007.
[7] V. Rabaud and S. Belongie, "Counting crowd moving objects," in *Computer Vision and Pattern Recognition*, 2006.
[8] N. Dalal and B. Triggs, "Histogram of oriented gradient for human detection," in *Computer Vision and Pattern Recognition*, 2005.
[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition*, 2009.
[10] P. Felzenszwalb, D. McAllester, and D. Ramananand, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition*, 2008.
[11] B. Wu and R. Nevatia, "Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection response," *Int'l J. Computer Vision*, vol. 82, no. 2, 2009.
[12] A. Davies, J. Yin, and S. Velastin, "Crowd monitoring using image processing," *IEE Electronic and Communications Engineering J.*, 1995.
[13] R. Ma, L. Li, W. Huang, and Q. Tian, "On pixel count based crowd density estimation for visual surveillance," in *Int'l Conf. Cybernetics and Intelligent Systems*, 2004.
[14] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *British Conf. Machine Vision*, 2005.
[15] A. Chan, M. Morrow, and N. Vasconcelos, "Analysis of crowded scenes using holistic properties," in *Int'l Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.
[16] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Digital Image Computing: Techniques and Applications*, 2009.
[17] A. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *Int'l Conf. Computer Vision*, 2009.
[18] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010.
[19] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, 2006.
[20] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," in *Computer Vision and Pattern Recognition*, 2006.
[21] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Int'l Conf. Computer Vision*, 2003.
[22] A. Albiol, M. Silla, and J. Mossi, "Video analysis using corner motion statistics," in *Int'l Workshop on Performance Evaluation of Tracking and Surveillance*, 2009.