Cross-Camera Knowledge Transfer for Multiview People Counting

Nick C. Tang, Yen-Yu Lin, *Member, IEEE,* Ming-Fang Weng, and Hong-Yuan Mark Liao, *Fellow, IEEE*

***Abstract*—We present a novel two-pass framework for counting the number of people in an environment where multiple cameras provide different views of the subjects. By exploiting the complementary information captured by the cameras, we can transfer knowledge between the cameras to address the difficulties of people counting and improve the performance. The contribution of this work is threefold. First, normalizing the perspective of visual features and estimating the size of a crowd are highly correlated tasks. Hence we treat them as a joint learning problem. The derived counting model is scalable and it provides more accurate results than existing approaches. Second, we introduce an algorithm that matches groups of pedestrians in images captured by different cameras. The results provide a common domain for knowledge transfer, so we can work with multiple cameras without worrying about their differences. Third, the proposed counting system is comprised of a pair of collaborative regressors. The first one determines the people count based on features extracted from intra-camera visual information, while the second calculates the residual by considering the conflicts between inter-camera predictions. The two regressors are elegantly coupled and provide an accurate people counting system. The results of experiments in various settings show that, overall, our approach outperforms comparable baseline methods. The significant performance improvement demonstrates the effectiveness of our two-pass regression framework.**

***Index Terms*—People counting, transfer learning, correspondence estimation.**

I. Introduction

The goal of *people counting* is to estimate the number of pedestrians or the density of the crowd in a monitored environment [1]–[8]. In recent years, the topic has generated a great deal of interest among researchers in many fields, e.g., image processing, computer vision, security and surveillance, because it plays an important role in a broad spectrum of real-world applications, such as video understanding, summarization, and traffic monitoring.

Despite the wide applicability of people counting, most computer-vision-based systems suffer from the following drawbacks. First, mutual occlusion among pedestrians causes significant changes in their appearances and the loss of extracted features. It often results in an underestimate of the number of people. Second, the problems caused by low-resolution or blurred images, especially for pedestrians far from the camera, usually degrade the stability of a counting system. Finally, large variations in the appearance of pedestrians and lighting conditions, as well as cluttered backgrounds, make people counting more difficult.

In this work, we propose a *multiview people counting* (MVPC) system to resolve the above issues. Specifically, multiple cameras monitor an area from different angles. Videos recorded by the cameras contain complementary information; therefore, fusing the knowledge embedded in the videos facilitates the development of a robust and accurate counting system. In this paper, we consider two issues: 1) How all visual cues captured by different cameras can be shared? 2) For each view, how can the intra-camera and inter-camera visual knowledge be combined to yield a robust and accurate counting system?

To work with cameras that have different settings, we propose a correspondence estimation algorithm that maps each segmented group of pedestrians in one view to the corresponding group in another view. We call these corresponding groups *matched blob clusters*, each of which enables knowledge to be shared between cameras. The intra-camera visual cues (captured by one camera) and inter-camera visual knowledge (transferred from other cameras) are included in each view. It follows that we present a *two-pass regression* framework for multiview people counting. Specifically, the first-pass regressor uses the visual features extracted from the intra-camera video frames to estimate the size of a crowd. Then, the second-pass regressor estimates the residual obtained in the first pass by considering the inconsistency in the knowledge provided by multiple cameras (i.e., inter-camera knowledge). Because the second pass is based on the conflicts between the predictions derived from multiple views, we formulate the training of the second-pass regressor as a transfer learning problem [9]. We investigate the properties of each matched blob cluster, i.e., having the same numbers of pedestrians in all views, with the objective of transferring useful knowledge and preventing error propagation.

The contributions of this work are as follows. First, we introduce a novel regularized regression method to evaluate the size of a crowd. The method is highly scalable because it can count people in crowds that are unseen in the training set, and it outperforms comparable techniques. Second, we present an algorithm that can estimate cross-camera correspondence accurately and combine multiple cameras with different settings by matching the blob clusters. Third, we utilize a pair of collaborative regressors to transfer cross-camera knowledge. The regressors are elegantly coupled so that intra-camera

This work was supported in part by Ministry of Science and Technology (MOST) under Grant 103-2221-E-001-026-MY2.

N. C. Tang and H.-Y. M. Liao are with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. Phone: 886-2-2788-3799 ext. 1519; Fax: 886-2-2782-4814; E-mail: nickctang@gmail.com, liao@iis.sinica.edu.tw.

Y.-Y. Lin is with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan. Phone: 886-2-2787-2392; Fax: 886-2-2787-2315; E-mail: yylin@citi.sinica.edu.tw.

M.-F. Weng is with the Smart Network System Institute, Institute for Information Industry, Taipei 105, Taiwan. Phone: 886-2-6607-3754; Fax: 886-2-6607-3511; E-mail: mfueng@iii.org.tw.

visual features and inter-camera view knowledge are considered simultaneously to derive an accurate people counting system. To the best of our knowledge, this work gives the first machine learning algorithm that integrates visual knowledge captured by multiple cameras for people counting.

II. Related Work

In this section, we review the research on topics related to the development of the proposed framework.

A. People Counting

The literature on people counting is quite comprehensive [10], [11]. However, we only consider computer-vision-based methods, which can be divided into two categories, *counting-by-detection* and *counting-by-regression* methods.

Counting-by-detection. In this category, the methods determine the number of people in an image by locating the position of each individual and then counting the total number. The methods proposed in [12]–[15] utilize various low-level features to search for human heads or moving entities. Inspired by the progress of designing powerful pedestrian detectors [16]–[19], methods in this category often employ a pedestrian detector to find people [20], [21]. Since the training samples are usually of a high resolution without occlusions, the detectors' performance deteriorates significantly when the targeted pedestrians are partially occluded or in blur images. Moreover, the computational cost of the detection stage is typically too high to support real-time responses.

Counting-by-regression. The methods in this category are relatively efficient. They estimate the size of a group by extracting low-level features to represent the corresponding region, which is usually generated by background subtraction or motion segmentation [3]–[5], [22]. However, these methods can not solve the localization problem or determine the exact number of people. They are only suitable for estimating the level of crowdedness. Following the methods that linearly map a set of perspectively normalized features to the number of people [22]–[24], nonlinear regression models, such as neural networks, Gaussian process models, and Poisson process models, have been utilized recently to enhance the performance of people counting systems [1], [3], [25], [26].

While aforementioned approaches to people counting address ROI (region of interest) counting, there exist approaches to LOI (line of interest) counting. For instance, Ma and Chan [27] presented an integral programming method for counting pedestrians crossing a line. Cong et al. [28] developed an approach to both ROI counting and LOI counting.

Despite the use of discriminative visual features and powerful machine learning techniques, the systems in the above two categories still suffer from the problems caused by occlusions, low-quality images, and large variations in the appearance of pedestrians. We try to resolve these problems by integrating information captured by multiple cameras. It is worth noting that the most similar work to our approach is probably that of Ma et al. [7], which fuses visual cues from two cameras to improve the performance of people counting. The authors emphasize reliable detection by two single-view cameras, and average the counting results provided by the two camera views. Because the approach utilizes an off-the-shelf pedestrian detector, its ability to handle occlusions may be restricted in crowded environments. In contrast to [7], we investigate the conflicts between predictions based on the views of multiple cameras in each matched blob cluster, and determine the reliability of the information derived from the views. As a result, only useful knowledge is transferred and error propagation is mitigated. We also extend our previous work [29] by defining the perspective normalization and regressor learning tasks as a joint optimization problem. It turns out that the resulting model can deal with the problem caused by the inconsistency between the training and testing data, and is more accurate than the previous one [29].

B. Correspondence between Multiple Cameras

To enable multiple cameras to share visual knowledge, we must establish their correspondence. Generally, conventional methods for determining camera correspondence can be divided into two categories: homographic-based methods and calibration-based methods.

Most homographic-based methods, such as [30]–[32], estimate plane homographies by matching salient regions across images, e.g., SIFT features [33] or people heads, and then determine the correspondence between multiple cameras. In general, these methods are sensitive to large variations in the appearance of objects, camera settings, and video qualities. To be effective, the approaches require consistent matching, but this may not be possible if the above problems occur.

The objective of calibration-based methods, e.g., [34]–[37], is to derive the model of a camera, including 1) the extrinsic parameters, i.e., the position and orientation of the camera relative to the real-world coordinate system; and 2) the intrinsic parameters, i.e., the image center, focal length, and distortion coefficients. Having a precise planar transformation between multiple cameras would facilitate people counting in crowded environments. Generally, calibration-based methods provide more precise camera transformations than homographic-based methods; hence, they are more suitable for our work.

C. Transfer Learning

Transfer learning refers to an information delivery process that tries to improve the target task by exploiting the abundant knowledge available in the source tasks. The exploration of auxiliary knowledge derived from different tasks has generated a great deal of interest among researches in the field of machine learning. The methods that utilize additional knowledge sources to accomplish a task can be divided into four categories [9]: transfer 1) by *model parameters* [38], [39]; 2) by *data instances* [40], [41]; 3) by *feature representation* [42]; and 4) by *contextual information* [43], [44]. All of these methods are based on the assumption that the data sets of the source and target tasks have the same domains for knowledge transfer. In this work, we consider the visual cue captured by each camera as an information source, and try to establish a robust counting system by sharing knowledge across cameras. To compensate for the variations resulting from heterogeneous

cameras with diverse perspective settings, we integrate the cameras by matching the blob clusters, which serve as the *common domains* for knowledge transfer.

III. The Proposed Two-pass Regression Framework

In this section, we describe the proposed MVPC system. Suppose a set of M cameras, $\mathcal{P}=\{P_m\}_{m=1}^{\mathrm{M}}$, is installed to monitor a public environment; and let $\mathcal{V}=\{V_m\}_{m=1}^{\mathrm{M}}$ be the videos recorded by the cameras. Without loss of generality, we assume that the videos are synchronized, and that each one comprises T frames. The objective is to construct a matrix $\boldsymbol{Y}=[y_{m,t}]\in\mathbb{R}^{\mathrm{M}\times\mathrm{T}}$, where $y_{m,t}$ is the predicted number of people present in $I_t^{(m)}$, the t-th frame in V_m.

Although our approach is a counting-by-regression system that maps a set of low-level features to the people count, it differs from existing approaches because it considers intra-camera visual features and inter-camera view knowledge jointly. The proposed two-pass regression framework approximates the number of people in an image in two parts—the *regular* part and the *residual* part, as shown in Figure 1. Similar to other counting-by-regression approaches, e.g., [1], [25], the regular part infers the number of people in an image blob based on the single-view, low-level features in the same way as other counting-by-regression approaches. For various reasons, such as partial occlusions or imperfect motion segmentations, inference based on features grabbed from a single view typically yields a residual. To resolve this problem, we match blob clusters and borrow visual knowledge from a number of cameras. As the *pedestrians are identical in all the views of each matched cluster*, the clusters compensate for the variations in the cameras, and serve as a common platform to deliver additional knowledge from other views for residual estimation. The two examples in Figure 1 demonstrate how occlusion and a shadow result in underestimation and overestimation, respectively. They also show how the two undesirable effects are manipulated by the two-pass regression mechanism.

In the remainder of this section, we focus on the design of regular estimation (Section III-A) and that of residual estimation (Section III-B).

A. Regular Estimation (First-pass Regression)

It is difficult to construct a counting model that can simultaneously address two critical issues, *accuracy* and *scalability*. Accuracy means the consistency between the predictions and the ground truth. Scalability is the ability to estimate the size of a crowd in the testing stage when the information is not included in the training data.

In regular regression, we deliver the nonlinear dependence between feature responses in various scales to improve the accuracy, and train a linear regressor to enhance the scalability. The two tasks are highly correlated, and we define them as a joint optimization problem. The resulting people counting system yields very accurate results without compromising the scalability.

(a) (b) (c)

Fig. 2. Feature representation of blobs. (a) & (b) The perspective effect makes the average height of a pedestrian being a function of where he/she is in the frame. (c) Uniformly sampled scales in logarithmic space.

1) Blob Extraction and Representation: We represent a video frame by a set of blobs, each of which is a group of spatially connected foreground pixels. We apply a background subtraction algorithm [45] to segment the foreground areas. Then, spatially connected foreground pixels are clustered to produce *blobs*.

Because our people counting system operates repeatedly at every timestamp, for simplicity, we only consider frames grabbed at time t. Thus, for frame $I_t^{(m)}$ taken by camera P_m, the index t can be dropped without causing any ambiguity. Assume that frame $I^{(m)}$ contains $\hat{\mathrm{N}}$ blobs, i.e., $I^{(m)}=\left\{\left(\mathrm{x}_b^{(m)},\hat{y}_b^{(m)}\right)\right\}_{b=1}^{\hat{\mathrm{N}}}$, where $\mathrm{x}_b^{(m)}$ is the feature description of the b-th blob, and $\hat{y}_b^{(m)}$ is the number of people in that blob. Note that $\hat{y}_b^{(m)}$ is given in the training phase, and we want to estimate it in the test phase. With this representation, the number of people in $I^{(m)}$ is calculated by summing the numbers estimated in the blobs, i.e., $y^{(m)}=\sum_{b=1}^{\hat{\mathrm{N}}}\hat{y}_b^{(m)}$.

Because of the perspective effects, the average height of a pedestrian in an image depends on his/her location on the ground plane [46], as shown in Figure 2(a) and Figure 2(b). Therefore, *perspective normalization* of the features is required to make the people counting task more accurate. To this end, we uniformly sample a finite number of scales in logarithmic space, as shown in Figure 2(c). Next, we extract the features of blob $\mathrm{x}_b^{(m)}$, and assign the feature responses to the corresponding scales. The dimension of $\mathrm{x}_b^{(m)}$ is equivalent to the number of sampled scales, which is denoted as H; and the value of H is determined as a trade-off between efficiency and precision. We set the value at 30 in all the experiments.

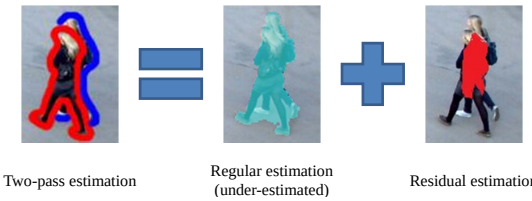
To ensure that the performance of the people counting system is satisfactory, we utilize the following low-level visual features to characterize the properties of blobs:

Area. This attribute represents the total number of foreground pixels occupied by a blob in each sampled scale. It approximates the size of moving objects in a scene.

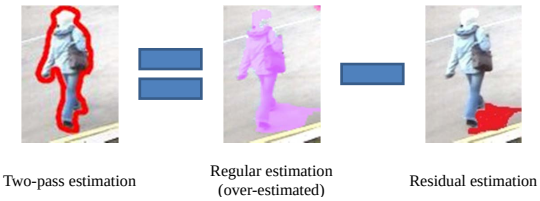
Canny edge pixels. To calculate the total number of edge pixels in each scale, we use the Canny edge detector, which captures the structural properties of crowdedness.

Oriented gradients. Each scale contains two independent features which represent the gradient magnitudes of vertical and horizontal orientations, respectively.

As the features capture diverse characteristics of a blob, we treat each one as a unique descriptor. Thus, each blob is represented by four types of descriptors of the same dimension, i.e., H.



(a) Occlusion handling



(b) Shadow handling

Fig. 1. Our two-pass regression framework is comprised of a regular estimation component and a residual estimation component. The former infers the number of people in a blob based on intra-camera low-level features, while the latter estimates the residual by exploiting inter-camera information. The two passes complement each other, and solve the difficulties of people counting, such as (a) occlusion and (b) shadows.

Algorithm 1 The proposed constrained linear regression. $(\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_\mathrm{D}) = \textbf{CLR-TRAIN}\,(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_\mathrm{D}, \mathrm{y})$. Given a set of N training examples with their D types of feature descriptors, $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_\mathrm{D}$, and a label vector y, derive a regression model comprised of D sets of parameters, $\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_\mathrm{D}$. T denotes the number of runs to be repeated.

for $i = 1$ **to** D **do**
  for $j = 1$ **to** T **do**
    Select half of training data, $\boldsymbol{X}'_i$ and y′, randomly.
    Solve the optimal $\mathrm{w}_{i,j}$ in (8) with $\boldsymbol{X}'_i$ and y′.
    Predict the whole dataset by $\mathrm{y}_{i,j} = \mathrm{w}_{i,j}^T \boldsymbol{X}_i$.
  end for
end for
Calculate the optimal combination weights $\alpha^*_{i,j}$ by applying the interior-point algorithm to solve
$\min_{\alpha_{i,j}} \left\| \sum_{i=1}^{\mathrm{D}} \sum_{j=1}^{\mathrm{T}} \alpha_{i,j} \mathrm{y}_{i,j} - \mathrm{y} \right\|_1$
s.t. $\alpha_{i,j} > 0$ for $i = 1, 2, \ldots, \mathrm{D}$ and $j = 1, 2, \ldots, \mathrm{T}$
return $\mathrm{w}_i = \sum_{j=1}^{\mathrm{T}} \alpha^*_{i,j} \mathrm{w}_{i,j}$ for $i = 1, 2, \ldots, \mathrm{D}$

2) Learning with Intra-camera Visual Features: To compensate for the perspective effect, existing approaches, such as [1] and [25], use a geometric factor or a density map to weight the features extracted from each scale, and then learn the regressor. These approaches are simple, but have two drawbacks. First, the weighting scheme is usually devised in an ad-hoc manner and is determined by our prior knowledge. For example, we assume that the edge-based features grow linearly with respect to the scale. However, prior knowledge of all the features is not usually available. Second, the local nonlinearities between the weights and the scales are ignored. Because of the segmentation and occlusion effects, the statistics of extracted raw features do not follow a global relationship exactly, so performing feature normalization and regressor construction separately could yield suboptimal results. Thus, for people counting, we train a model to consider perspective normalization and regressor training simultaneously. In the experiments, we will show that approaches based on feature weighting with powerful regression models, e.g., neural networks and Gaussian processes, still suffer from less accuracy. We will demonstrate that our method is capable of alleviating the problem.

Consider a set of training instances described by the feature $\boldsymbol{X} = [\mathrm{x}_1^{(m)} \ \ldots \ \mathrm{x}_{\hat{\mathrm{N}}}^{(m)}] \in \mathbb{R}^{\mathrm{H} \times \hat{\mathrm{N}}}$ and their labels $\mathrm{y} = [\hat{y}_1^{(m)} \ \ldots \ \hat{y}_{\hat{\mathrm{N}}}^{(m)}]^T \in \mathbb{R}^{\hat{\mathrm{N}} \times 1}$. We perform perspective normalization by weighting the features, and carry it out by solving

$$\begin{aligned} \min_{\mathrm{w}} \quad & \|\mathrm{w}^T \boldsymbol{X} - \mathrm{y}\|_1 \\ \text{subject to} \quad & \mathrm{w_i} > 0, \ \text{for } 1 \le i \le H, \end{aligned} \tag{1}$$

where $\mathrm{w} = [w_1 \ \ w_2 \ \ \cdots \ \ w_H]^T$. In Equation 1, we choose ℓ_1 norm minimization instead of ℓ_2 norm because it is less sensitive to outliers and the objective function is consistent with the performance evaluation metric for people counting. We further consider the correction of geometric distortion so as to avoid overfitting, in particular when the size of training set is small and the observed instances do not locate and cover all scales in logarithmic space. To this end, we introduce *exponential scaling law* [47] into the training procedure of the regressor. It states that the feature responses of a pedestrian depend on his/her vertical position in an image. Since the vertical space has been quantized into scales, according to the exponential scaling law, feature responses at each scale s_i, i.e., $f(s_i)$, can be represented as

$$f(s_i) = e^{\lambda s_i + \mu}, \tag{2}$$

where λ and μ are two real-valued unknowns. Their values are dependent on not only the camera perspective but also the feature characteristic. In the following, we show how to utilize this property to regularize regressor training.

Considering a crowd of people in a scene, the size of the crowd stays constant no matter which scale the crowd belong to. The prior information can be taken into account to relate the regressor parameters, w. That is, $w_i f(s_i) = c$ for each scale s_i, where constant c is the crowd size. By taking the logarithm of both sides, we obtain

$$\ln(w_i) = \ln\left(\frac{c}{f(s_i)}\right) = \ln(c) - \lambda s_i - \mu. \tag{3}$$

Since λ and μ are variables to be derived, for the ease of representation we simply let two new variables replace $-\lambda$ and $\ln(c) - \mu$ in the equation, i.e., $\lambda \leftarrow -\lambda$ and $\mu \leftarrow \ln(c) - \mu$. Considering all scales, we have

$$\begin{aligned} \lambda s_1 + \mu &= \ln(w_1) \\ &\vdots \\ \lambda s_i + \mu &= \ln(w_i) \\ &\vdots \\ \lambda s_\mathrm{H} + \mu &= \ln(w_\mathrm{H}). \end{aligned} \tag{4}$$

Since a perfect solution to Equation 4 that fits best the local nonlinearity of the training samples may not exist, we seek a solution that allows a small amount of error, i.e.,

$$\left\| \boldsymbol{S} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} - \ln(\mathrm{w}) \right\|^2 < b, \text{ where } \boldsymbol{S} = \begin{bmatrix} s_1 & 1 \\ \vdots & \vdots \\ s_h & 1 \end{bmatrix}, \quad (5)$$

where $\ln(\mathrm{w}) = \begin{bmatrix} \ln(w_1) & \ln(w_2) & \dots & \ln(w_H) \end{bmatrix}^T$. By normal equations, the optimal λ^* and μ^* would be

$$\begin{bmatrix} \lambda^* \\ \mu^* \end{bmatrix} = \left(\boldsymbol{S}^T \boldsymbol{S}\right)^{-1} \boldsymbol{S}^T \ln(\mathrm{w}). \quad (6)$$

By substituting λ^* and μ^* for λ and μ in Equation 5, the inequality can be rewritten as

$$\|\boldsymbol{E} \ln(\mathrm{w})\|^2 < b, \text{ where } \boldsymbol{E} = \boldsymbol{S}\left(\boldsymbol{S}^T \boldsymbol{S}\right)^{-1} \boldsymbol{S}^T - \boldsymbol{I}. \quad (7)$$

The fitting of data for regressor learning in Equation 1 and the geometric regularization guided by the exponential scaling law in Equation 7 are considered jointly, which leads to

$$\begin{aligned} \min_{\mathrm{w}} \quad & \|\mathrm{w}^T \boldsymbol{X} - \mathrm{y}\|_1 \\ \text{subject to} \quad & \|\boldsymbol{E} \ln(\mathrm{w})\|_2 < b, \ \mathrm{w_i} > 0, \ \text{for } 1 \le i \le H. \end{aligned} \quad (8)$$

The constrained optimization problem in Equation 8 is nonlinear, so we use the *interior-point* algorithm [48] to solve it. Because an initial solution is required in optimization, we use package *CVX* [49] to solve the convex problem. We also introduce a mechanism to avoid trapping in a local minimum. Instead of using the whole training dataset, we solve the problem in Equation 8 with different subsets. We randomly select half of the training data in each run. As a result, we obtain base regressors. We repeat the above process for each feature descriptor. First-pass regressor is a linear combination of the base regressors, and it is derived by minimizing the loss of the training samples. The training procedure of this constrained linear regression method is summarized in Algorithm 1.

For each camera P_m, we train a regressor via Algorithm 1, and define a camera-specific function $\mathcal{F}^{(m)}$ to determine the number of people in a blob. To evaluate a blob with multiple feature representations, $\mathrm{x}^{(m)} = \{\mathrm{x}_d^{(m)}\}_{d=1}^{\mathrm{D}}$, the size of the regular part of the blob can be computed by

$$\mathcal{F}^{(m)}\left(\mathrm{x}^{(m)}\right) = \sum_{d=1}^{\mathrm{D}} \mathrm{w}_d^T \mathrm{x}_d^{(m)}. \quad (9)$$

Note that $\mathcal{F}^{(m)}$ is determined by the training data collected from one camera. Information captured by the other cameras is not referenced. Besides $\mathcal{F}^{(m)}$, we also derive D additional regressors $\{\mathcal{F}_d^{(m)}\}_{d=1}^{\mathrm{D}}$ independently. $\mathcal{F}_d^{(m)}$ is learned with the same training blobs, but only the d-th feature descriptor is considered. The reason of deriving $\{\mathcal{F}_d^{(m)}\}_{d=1}^{\mathrm{D}}$ will be clarified later.

B. Residual Estimation (Second-pass Regression)

When multiple cameras with various perspective settings are used, one of the most important tasks is to correlate the cameras, so that view-specific knowledge can be transferred between collaborative cameras. In the second-pass regression phase, the objective is to estimate the residual, which could not be derived in the first-pass regression phase.

1) Blob Localization and Matching: After extracting the blobs from different camera views, we try to localize and match the corresponding blobs. For this task, we propose a blob matching algorithm for *ground plane mapping* and *vertical plane mapping*.

Figure 3(a) illustrates the procedure of ground plane mapping. We first compute the convex hull of each blob in P_i to determine its bottom boundary. Figure 4(a) shows some examples of detected bottom boundaries highlighted in different colors. Then, based on the assumption that a blob's bottom boundary touches the ground plane, its correspondence across cameras can be derived by the *image-to-world* and *world-to-image* coordinate transformations. This two transformations are based on a reverse variant and the original of Tsai's camera calibration model [36], respectively. While the details of Tsai's model are given in [36], we describe its reverse variant, derived by us, in the supplementary APPENDIX for more thorough explanation. In addition, in our implementation, the calibration between P_i to camera P_j is done in advance.

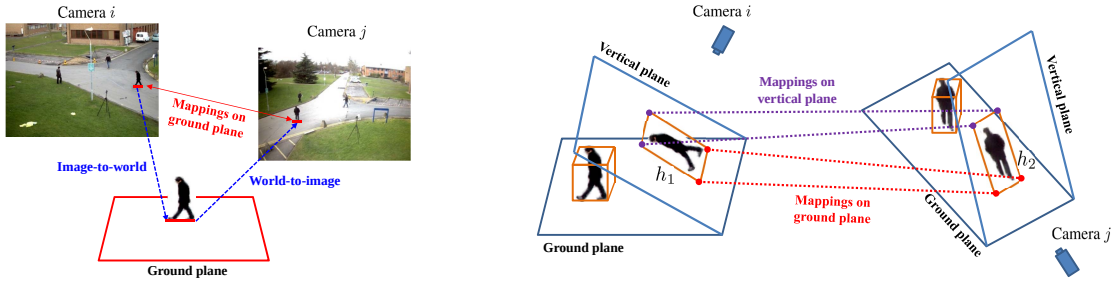
Figure 4(b) shows the bottom boundary correspondence results, every pixel on the bottom boundary in camera P_i are mapped to the image plane of camera P_j. It follows that the mappings between blobs taken by different cameras are obtained. We use the set of mappings to make an initial estimation of the correspondence between two cameras.

However, in our empirical tests, the bottom boundary of a blob does not always touch the ground plane due to imperfect blob segmentation. We further establish mappings on the *vertical plane* of each blob, which is illustrated in Figure 3(b), to validate the correctness of the estimated correspondence. The image height of a pedestrian at every position is required to compute mappings on the vertical plane of each blob. Based on the work by Hoiem et al. [46], the image height of a pedestrian (denoted as h) is assumed to be linearly dependent on his/her bottom location (denoted as v) in the vertical position of the image, i.e.,

$$h(v) = \alpha \cdot v + \alpha_0, \quad (10)$$

where α and α_0 are the two parameters of the camera model. They control the linear dependency between the image height and the image location of a pedestrian. We adopt the procedure described in [29] to determine the values of the two parameters of each camera via employing an off-the-shelf pedestrian detector [50]. On the one hand, the detected pedestrians can be used to estimate the camera model. On the other hand, the estimated model can filter out false detections. The two steps are done alternately until convergence. It is worth mentioning that our approach, developed based upon the camera model [46], assumes that pedestrians in the scene have similar heights. Namely, their image heights are only dependent on their locations in the image.

After having the bottom boundary of each blob, the head positions of the blob are available according to the estimated camera perspective model. We then take both the mappings of the bottom boundary and the head positions of each blob,



(a) Cross-camera ground plane matching (b) Ground plane matching vs. Vertical plane matching

Fig. 3. The proposed approach to cross-camera blob localization and matching.

(a) (b)

Fig. 4. Blob mapping on the ground plane: (a) the bottom boundaries of the blobs; (b) their mappings.

(a) (b)

Fig. 5. Blob mapping on the ground and vertical planes from (a) Camera P_i to (b) Camera P_j.

and apply the *direct linear transformation* [51] to calculate the *planar homographies*. It follows that every pixel in the blob can be projected across cameras. An example of the estimated mappings of blobs on the vertical planes is shown in Figure 5.

2) Matched Blob Cluster Extraction: After establishing the correspondence between the blobs derived from two cameras, we can group the blobs into *clusters* so that each cluster contains minimal and identical entities that are present in both camera views. The example in Figure 6 illustrates the grouping process. Suppose that $\{b_1^{(i)}, b_2^{(i)}, \ldots, b_7^{(i)}\}$ and $\{b_1^{(j)}, b_2^{(j)}, \ldots, b_5^{(j)}\}$ are the two sets of blobs extracted from camera P_i and P_j, respectively. First, we construct a bipartite graph of twelve nodes, as shown in Figure 6(a). An edge is added between two nodes on opposite sides if the corresponding blobs are matched in any direction. By computing the connected components in the graph, we obtain blob clusters $\{c_1^{(i)}, c_2^{(i)}, \ldots, c_4^{(i)}\}$ and $\{c_1^{(j)}, c_2^{(j)}, \ldots, c_4^{(j)}\}$ for cameras P_i and P_j, respectively. Note that each corresponding component refers to the same group of pedestrians in both images, which implies that the feature distributions in both views, as well as pedestrian counts, can be compared directly.

3) Blob Cluster Representation: A video frame can also be represented by the set of matched blob clusters. Since the corresponding clusters of different views refer to the same entities, the number of people in each cluster should be identical. The conflict between predictions based on multiple views indicates that a residual occurs. Therefore, knowledge like intra-camera estimation results shared among matched clusters can be used directly without further transformation or adaption. Suppose frame $I^{(m)}$ contains $\tilde{\mathrm{N}}$ matched blob clusters, it can be expressed as $\{(\mathrm{z}_c^{(m)}, \tilde{y}_c^{(m)})\}_{c=1}^{\tilde{\mathrm{N}}}$, where $\tilde{y}_c^{(m)}$ is the residual of the c-th blob cluster derived in the first pass, and $\mathrm{z}_c^{(m)}$ is the feature representation. As in the first pass, $\tilde{y}_c^{(m)}$ is given in the training phase, we need it to make predictions. The residual $\tilde{y}_c^{(m)}$ and the feature representation $\mathrm{z}_c^{(m)}$ are defined in the next subsection.

4) Learning with Inter-camera Knowledge: Consider a matched blob cluster $\mathcal{C}^{(m)} = (\mathrm{z}^{(m)}, \tilde{y}^{(m)}) = \{(\mathrm{x}_b^{(m)}, \hat{y}_b^{(m)})\}_{b=1}^{\mathrm{N}}$ taken by camera P_m and comprised of N blobs. The residual of $\mathcal{C}^{(m)}$ in the first stage is defined as

$$\tilde{y}^{(m)} = \sum_{b=1}^{\mathrm{N}} \hat{y}_b^{(m)} - \sum_{b=1}^{\mathrm{N}} \mathcal{F}^{(m)}(\mathrm{x}_b^{(m)}). \tag{11}$$

The first term in the right-hand side of Equation 11 is the ground truth of the people count in $\mathcal{C}^{(m)}$; and the second term is the prediction in the first pass. We use inter-camera knowledge to design the feature representation of cluster $\mathcal{C}^{(m)}$. Initially, the people count of $\mathcal{C}^{(m)}$ estimated by all the developed unifeature regressors $\{\mathcal{F}_d^{(m)}\}_{d=1}^{\mathrm{D}}$ is evaluated, i.e.,

$$\mathrm{v}(\mathrm{z}^{(m)}) = [\mathcal{F}_1^{(m)}(\mathrm{z}^{(m)}), \mathcal{F}_2^{(m)}(\mathrm{z}^{(m)}), \cdots, \mathcal{F}_{\mathrm{D}}^{(m)}(\mathrm{z}^{(m)})]^T \in \mathbb{R}^{\mathrm{D}\times 1} \tag{12}$$

where the value of each element of the vector is given by

$$\mathcal{F}_d^{(m)}(\mathrm{z}^{(m)}) = \sum_{b=1}^{\mathrm{N}} \mathcal{F}_d^{(m)}(\mathrm{x}_b^{(m)}), \text{ for } d = 1, 2, \ldots, \mathrm{D}. \tag{13}$$

This blob cluster is matched across cameras; therefore, we have $\{\mathrm{v}(\mathrm{z}^{(m)})\}_{m=1}^{\mathrm{M}}$. Because $\mathcal{C}^{(m)}$ refers to the same entities, the estimated people counts, $\{\mathrm{v}(\mathrm{z}^{(m)})\}_{m=1}^{\mathrm{M}}$, can be compared directly. In our previous work [6], we show that the conflicts between predictions based on different features by a single camera enable us to infer the residual caused by occlusions. Here, we generalize the concept for multiple cameras, and develop the following four descriptors for cluster $\mathcal{C}^{(m)}$.

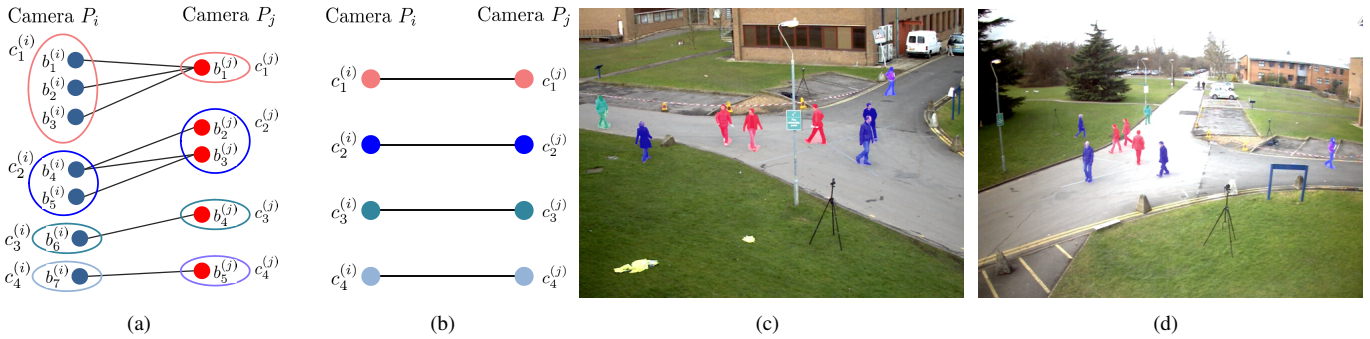


Fig. 6. The procedure for matching blob clusters across cameras. (a) A bipartite graph constructed for two camera views that contain seven and five blobs respectively. (b) Matching blob clusters by computing the connected components. (c) and (d) The resulting blob clusters, each of which is highlighted by the same color in both images. Note that, in both images each cluster contains the same group of pedestrians.

Cross-camera Conflict. For each adopted visual feature, this descriptor captures the conflict between camera P_m and the other cameras directly, i.e.,

$$\mathrm{z}^{(m)}.cc = \frac{\sum_{m'=1,m'\neq m}^{\mathrm{M}} \mathrm{v}(\mathrm{z}^{(m')})}{\mathrm{M}-1} - \mathrm{v}(\mathrm{z}^{(m)}). \quad (14)$$

Negative Trim. The people counts of some cameras tend to be underestimated because of the camera angle relative to the motion direction of the pedestrians or the distance to the monitored environment. In such cases, the positive part of Equation 14 is useful for residual estimation. Hence, this descriptor is defined as follows:

$$\mathrm{z}^{(m)}.nt = \max(\mathrm{z}^{(m)}.cc, 0). \quad (15)$$

Positive Trim. Similar to $\mathrm{z}^{(m)}.nt$, we have the descriptor

$$\mathrm{z}^{(m)}.pt = -\min(\mathrm{z}^{(m)}.cc, 0). \quad (16)$$

Intra-camera Conflict. The visual features have different degrees of sensitivity to occlusions, so extrapolation on the conflicts between predictions based on the features can recover the residual. This descriptor is defined as follows:

$$\mathrm{z}^{(m)}.ic = [\mathcal{F}_i^{(m)}(\mathrm{z}^{(m)}) - \mathcal{F}_j^{(m)}(\mathrm{z}^{(m)})], \text{ for } 1 \leq i < j \leq \mathrm{D}. \quad (17)$$

In the training phase, we match the blobs across cameras, and obtain a set of blob clusters.

After measuring the residual in Equation 11 and extracting the above four descriptors for each matched cluster, it becomes a task of *multiple kernel learning*, such as [52], [53], with four kernels (one for each descriptor) here. In this work, we use *SimpleMKL* [53] to derive a *multi-kernel support vector regressor*, $\mathcal{S}^{(m)}$, by taking the four kernels as input.

At the end of the training procedure, we have the final regressors of the two passes $\{\mathcal{F}^{(m)}, \mathcal{S}^{(m)}\}$ in each camera view m. In the testing phase, suppose there is a frame taken by camera P_m, and its blob representation $\{\mathrm{x}_b^{(m)}\}_{b=1}^{\tilde{\mathrm{N}}}$ and cluster representation $\{\mathrm{z}_c^{(m)}\}_{c=1}^{\tilde{\mathrm{N}}}$. Our approach estimates the number of people in the frame by

$$y_m = \sum_{b=1}^{\hat{\mathrm{N}}} \mathcal{F}^{(m)}\left(\mathrm{x}_b^{(m)}\right) + \sum_{c=1}^{\tilde{\mathrm{N}}} \mathcal{S}^{(m)}\left(\mathrm{z}_c^{(m)}\right). \quad (18)$$

IV. Experiments and Results

In this section, we evaluate the performance of the proposed framework and compare it with that of existing approaches. First, we describe the experimental settings, and then discuss the quantitative results of the compared methods. Finally, comprehensive studies on some components of this framework are carried out.

A. Experimental Settings

Video data. To evaluate the performance of the proposed system, we conducted experiments on the PETS 2010 Benchmark dataset [54], which contains videos captured by several stationary camcorders set up in different positions to monitor the same environment. All of video frames used as training and testing data were picked from *regular flow* of *S0* in PETS 2010. We chose three representative video clips of two views, denoted as SPARSE, MEDIUM, and HEAVY respectively. The SPARSE and MEDIUM videos only have a few individuals and small groups of pedestrians (i.e., minor or moderate occlusions), and the HEAVY video contains densely packed groups (i.e., heavy occlusions). Note that the monitoring regions by the different camera views do not fully overlap. For the ease of comparing multi-view and single-view people counting approaches in the experiments, we considered only video frames whose people counts in the different views are almost the same. Specifically, a synchronous frame was selected if $|pc_1 - pc_2| / \min(pc_1, pc_2) \leq 0.2$, where pc_1 and pc_2 are the numbers of people counts of this frame in the two views, respectively. The selected videos span a wide spectrum of occlusion levels; therefore, they provide a good test bed to assess how effectively our people counting approach handles occlusions. As the ground truth, we used the publicly available manually-annotated pedestrian labels for this benchmark dataset[1] [29]. Details of the experimental data are given in Table I.

Experimental setup. For quantitative evaluations of the compared people counting approaches, we divided each video clip into non-overlapping training and test sets. We used the first half of frames of each video as training data and assessed an approach's performance on the remaining frames. To determine

[1]Downloaded from http://research.twnct.net/MCPC/.

TABLE I
DESCRIPTION OF THE PETS 2010 VIDEO CLIPS USED IN OUR EXPERIMENTS AND THE PERFORMANCE OF THE PROPOSED BLOB MATCHING ALGORITHM.

Video Clips	SPARSE		MEDIUM		HEAVY	
	View 1	View 2	View 1	View 2	View 1	View 2
Total number of frames	360		190		40	
Minimal number of pedestrians in a frame	4	4	4	4	40	40
Maximal number of pedestrians in a frame	8	8	17	19	41	41
Mean of the number of pedestrians	6.8	6.8	10.7	11.9	40.5	41.0
Standard deviation of the number of pedestrians	0.9	1.1	3.8	4.9	0.5	0.2
Total number of segmented blobs	1854	1554	744	670	195	197
Precision of blob matching (%)	98.5		96.4		98.0	

the stability of the approaches, we conducted experiments in four settings; specifically only SPARSE, only MEDIUM, only HEAVY, and a mix of them (denoted as MIXED in the following discussion) are used as training data. Hence, we were able to evaluate 1) how changes in the training data affect an approach's performance; 2) the generalization ability of downscaling (i.e., training on large crowds, while testing on smaller crowds) and upscaling (i.e., training on small crowds, while testing on larger crowds); and 3) the ability of dealing with the complex mix of crowd sizes.

Evaluation criteria. In the experiments, we used the *mean absolute error* (MAE) as the criterion to measure the performance of the people counting methods on a single test video. Furthermore, we use the average MAE to reflect the overall performance of the compared systems.

Parameter determination. There are two main parameters in the proposed method, one in the first pass and the other in the second pass. We empirically determine a suitable value for the upper bound in Equation 8 and demonstrate that it is insensitive to the final performance. Note that the reported results, except those for the parameter sensitivity, are based on the same parameter, i.e., $b = 2^{-5}$ in all the experiments. In addition, we set the regularization parameter of the SimpleMKL package [53] to the optimal value, which is selected from a reasonable parameter space via three-fold cross validation.

B. Experimental Results

Accuracy of blob matching. We evaluated the accuracy of our blob correspondence estimation algorithm by manually checking the consistency of the matched components in two views, and calculated the percentage of the returned matches that are correct. The results in Table I show that, for the three videos with different levels of crowdedness, the proposed algorithm yields 96.4% to 98.5% accuracy for cluster matching. Some of the matching results are shown in Figure 8, where we assigned the same color and the same number to identify the matched clusters in the two views.

Baseline and comparison. We compared the proposed method with a Gaussian process method that is similar to Chan et al.'s state of the art approach [1]. We used a radial basis function and a linear function as kernels to learn a nonlinear estimation model and a linear estimation model respectively. Therefore, we have two sets of estimation results (RBF and LIN) to compare with our first-pass regression results (FPR). We used Gaussian processes, a kernel machine, to derive the regressors in both baselines RBF and LIN. The above approaches only use intra-camera visual cues; they do not transfer knowledge between different camera views. To determine if the proposed MVPC system outperforms conventional single-camera people counting systems, we used FPR as our baseline and applied the designed residual regressor to it. In the literature, the only MVPC system is a fusion approach that combines the results of detecting humans in multiple frames taken by different cameras [7]. However, the method is not suitable for handling scenes with dense crowds like the data used in our experiments. Therefore, we developed a variant of the approach in [7] to fuse complementary information. The variant *averages* the estimation results by FPR in the two views, and we denoted it as AVG; and the final results yielded by our two-pass regressor are denoted as TPR.

Overall performance. Table II lists the estimation errors (MAEs) made on the SPARSE, MEDIUM, and HEAVY videos by the proposed first-pass regression approach (FPR) and the two-pass regression framework (TPR). It also shows the results of the RBF, LIN, and AVG methods. If we only consider intra-camera visual features, FPR outperforms the RBF and LIN approaches in most settings. Comparing the results in View 1, both baselines RBF and LIN perform much poorly in View 2. The main reason could be that the variation of pedestrian heights in View 2 is obviously larger than that in View 1 due to the camera perspective, thus increasing the difficulties of counting. It is worth noting that FPR does not suffer from the problem, since it considers the exponential scaling law to regularize regressor learning. This case shows the advantage of our approach FPR, which joints perspective normalization and regressor learning. The results also demonstrate that FPR can 1) more accurately estimate the number of people in the test frames that contain similar sizes of crowds in the training ones; and 2) estimate downscaling and upscaling effectively.

Furthermore, Table II shows that the performance gains of multiview people counting systems that use an average fusion approach are modest because the approach interpolates the estimation results derived from different camera views and errors are often propagated, thereby degrading the performance. While the performance of the average fusion approach AVG improves on one view, the errors may increase on the other. In contrast, our approach casts visual knowledge transfer as a learning problem, and tries to deliver useful information

TABLE II
THE ESTIMATION ERRORS, MAE, OF VARIOUS APPROACHES WITH FOUR DIFFERENT SETTINGS OF TRAINING DATA.

Training Data	Test Data	View 1					View 2				
		RBF	LIN	FPR	AVG	TPR	RBF	LIN	FPR	AVG	TPR
SPARSE	SPARSE	1.35	0.86	0.52	0.68	**0.52**	0.65	**0.46**	0.78	0.54	0.64
	MEDIUM	6.08	1.94	2.20	2.78	**1.70**	7.09	4.51	5.75	5.17	**4.10**
	HEAVY	39.80	10.11	10.86	10.47	**8.22**	40.15	10.73	10.08	10.47	**5.09**
	MIXED	15.74	4.30	4.53	4.65	**3.48**	15.96	5.23	5.54	5.39	**3.28**
MEDIUM	SPARSE	1.28	0.81	0.77	0.76	**0.56**	2.08	3.31	1.45	**0.60**	0.72
	MEDIUM	4.22	2.34	2.41	**0.93**	1.12	7.64	**0.83**	1.33	2.22	1.06
	HEAVY	39.52	11.13	10.01	**6.79**	8.96	39.84	**2.83**	3.58	6.79	4.57
	MIXED	15.01	4.76	4.40	**2.83**	3.55	16.52	2.32	**2.12**	3.20	**2.12**
HEAVY	SPARSE	5.71	4.80	4.40	3.66	**2.29**	9.48	18.49	**2.90**	3.64	3.41
	MEDIUM	**1.40**	3.65	2.63	4.37	3.69	8.90	11.41	3.72	**2.06**	2.73
	HEAVY	1.24	3.16	**0.72**	0.98	0.75	4.93	6.92	2.48	**0.89**	0.99
	MIXED	2.78	3.87	2.58	3.00	**2.24**	7.77	12.27	3.04	**2.20**	2.38
MIXED	SPARSE	1.15	2.74	**0.62**	1.39	0.85	0.86	0.95	2.12	1.31	**0.85**
	MEDIUM	1.94	1.36	1.05	1.20	**0.99**	4.25	1.26	1.14	1.30	**1.07**
	HEAVY	9.04	0.98	2.93	1.33	**0.63**	15.79	11.72	0.85	1.31	**0.45**
	MIXED	4.05	1.70	1.53	1.31	**0.82**	6.97	4.64	1.37	1.31	**0.79**
OVERALL		9.39	3.66	3.26	2.95	**2.52**	11.80	6.12	3.01	3.03	**2.14**

and avoid error propagation simultaneously. It turns out that TPR reduces the MAEs in both the two views. Comparing with baseline AVG, our approach TPR leads to relative improvement 14.5% $\left(\frac{2.95-2.52}{2.95}\right)$ on View 1 and 29.3% $\left(\frac{3.03-2.14}{3.03}\right)$ on View 2.

Figure 7 shows the number of people frame-by-frame in the three test videos, including the manually annotated ground truth and the estimations made by RBF, LIN, FPR, AVG, and TPR in the three experiments. We observe that the performance gain of the proposed TPR is significant on HEAVY, while the improvement on SPARSE is moderate. There are two possible reasons for this outcome. First, the video frames of HEAVY often contain occlusions. Therefore, people counting based on a single camera is not sufficient, and information from the view captured by other cameras is valuable for residual compensation. Second, variations in the appearance of small groups of pedestrians is usually modest, but the variations may be large in highly occluded crowds. Thus, complementary information and inter-camera knowledge are more useful for handling densely crowded scenes than sparse scenes. To illustrate this point, we provide several examples of video frames with the estimations based on extracted clusters in Figure 8.

C. Comprehensive studies

In the subsection, we further evaluated the performance of a few individual components in the framework.

Effect of parameters. To evaluate the performance of the constrained linear regression algorithm, we conducted experiments by using the same settings on a range of values (from $2^{-7.5}$ to $2^{-0.5}$) of parameter b in Equation 8. First, the impact of randomness in Algorithm 1 is considered. Because the algorithm randomly selects instances as training data at each iteration and yields different results, each experiment was performed five times to evaluate the first-pass regular part.

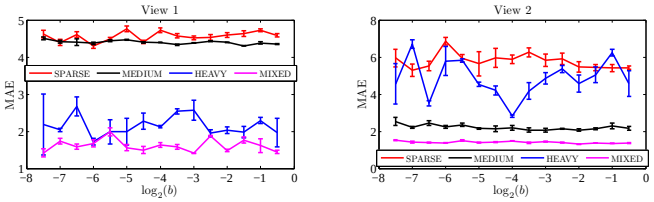
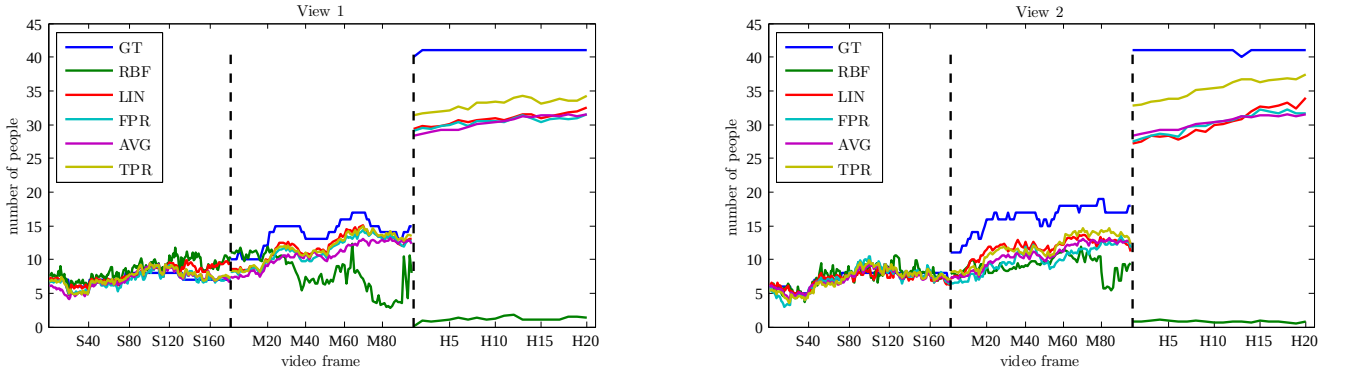


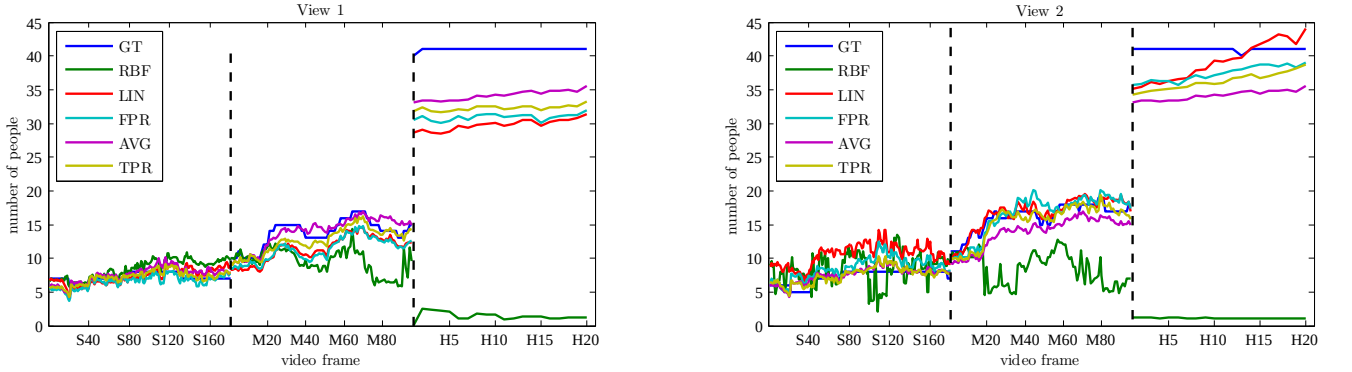
Fig. 9. The means and standard deviations of MAEs of the proposed approach with various values of the parameter b in Equation 8.

Figure 9 shows the means and the standard deviations of the performance of FPR on the test videos when SPARSE, MEDIUM, HEAVY, and MIXED are used as training data. The standard deviations of the errors are quite small in the cases of learning from the MIXED set, which varies from 0.0454 to 0.1303 in View 1 and from 0.0062 to 0.1891 in View 2. The results demonstrate that 1) the proposed algorithm stably generates models; and 2) the randomness does not have a large impact on the algorithm's performance. Then, we assessed the impact of the specified values of b. As shown in Figure 9, the results indicate that good performance is achieved with a wide range of b, i.e., $2^{-6} \sim 2^{-2}$. This finding confirms that the performance of the proposed method is stable.

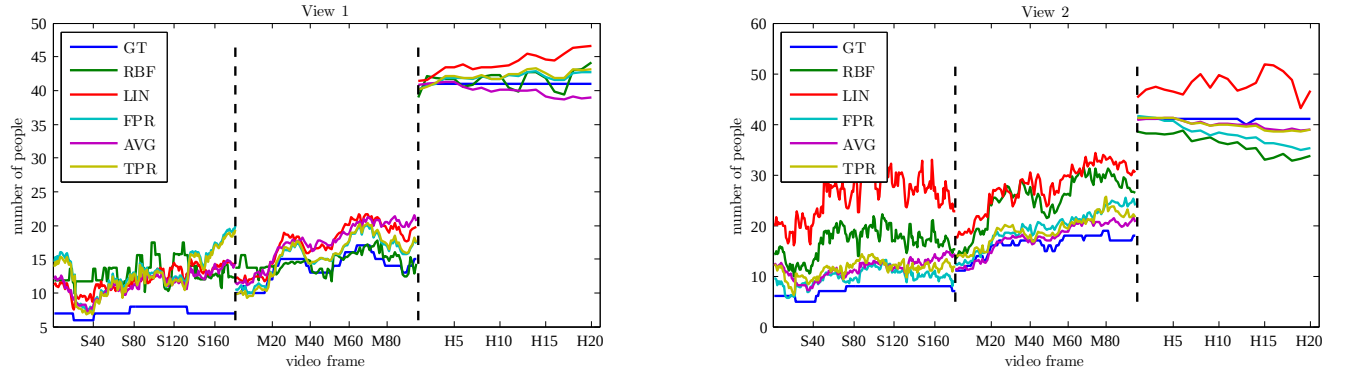
Effect of the exponential scaling law. In this study, we evaluated the benefit of introducing the exponential scaling law into people counting. We considered MIXED as training data in this set of experiments, because it stands for a general scenario. The first-pass regressor FPR, which adopts the exponential scaling law via Equation 8, is compared with two different methods. The first one is the regressor learned with Equation 1. It is the same as FPR except that the exponential scaling law is not taken into account. The second one is the method given in [1], which learns the regressor by weighting the features and respecting perspective deformation. Note that all the methods



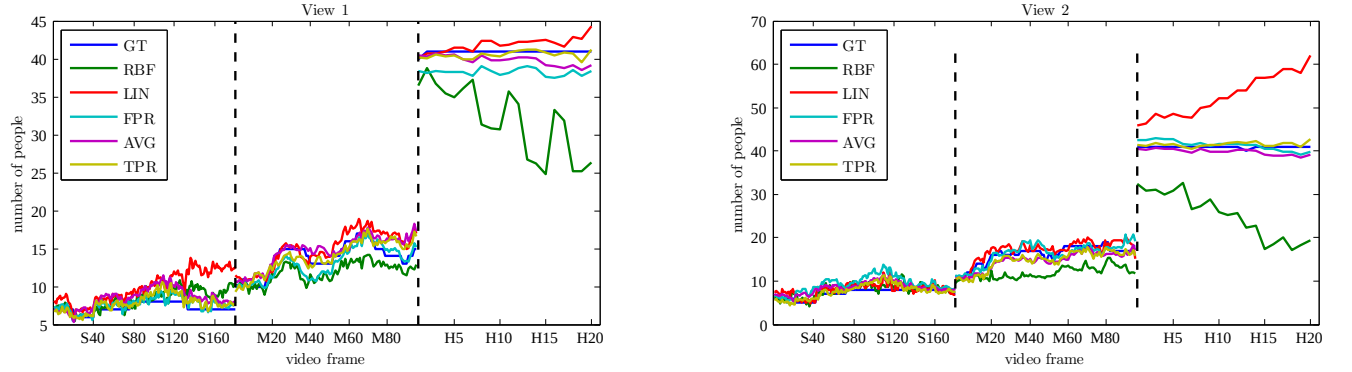
(a) Using SPARSE as training data



(b) Using MEDIUM as training data



(c) Using HEAVY as training data



(d) Using MIXED as training data

Fig. 7. The estimated number of people in frames of the four test videos (SPARSE, MEDIUM, HEAVY, and MIXED) reported with the ground truth. Note that the RBF, LIN, and FPR methods only use intra-camera visual features, while AVG and TPR utilize additional knowledge derived from different cameras.

View 1						View 2					
ID	GT	RBF	LIN	FPR	TPR	ID	GT	RBF	LIN	FPR	TPR
1	41	1.1	31.5	31.3	34.2	1	40	0.8	30.8	31.1	36.6

(a) A sample frame of HEAVY. The results are yielded by estimation models trained on SPARSE.

View 1						View 2					
ID	GT	RBF	LIN	FPR	TPR	ID	GT	RBF	LIN	FPR	TPR
1	13	12.6	15.6	13.4	13.7	1	15	21.7	27.4	18.3	15.7
2	3	2.6	4.6	3.5	3.5	2	3	2.5	1.5	0.3	2.1

(b) A sample frame of MEDIUM. The results are yielded by estimation models trained on HEAVY.

Fig. 8. Sample frames selected from the videos used in our experiments. The quantitative results reported on the components demonstrate the effectiveness of our first-pass regression approach and the proposed two-pass regression system.

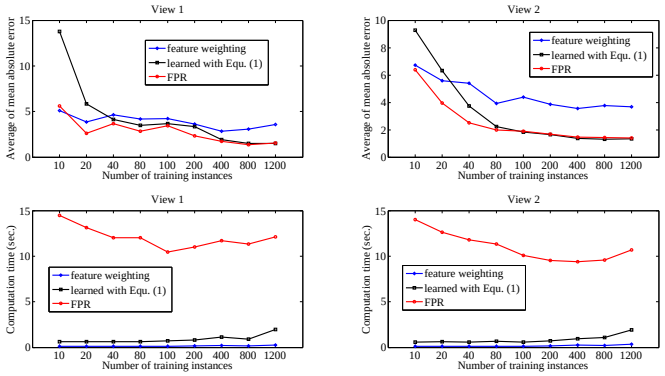


Fig. 10. (**Upper**) The average MAEs of three methods in View 1 and View 2, when different numbers of training data are available. (**Lower**) The corresponding running time in the training phase.

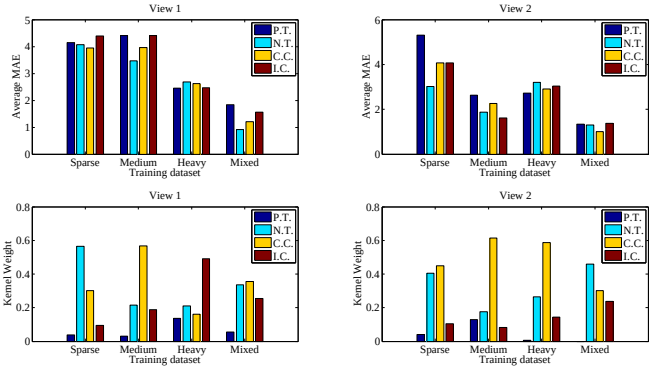


Fig. 11. (**Upper**) The individual performance of the four feature types adopted in the second-pass regression. (**Lower**) Their corresponding kernel weights optimized by SimpleMKL.

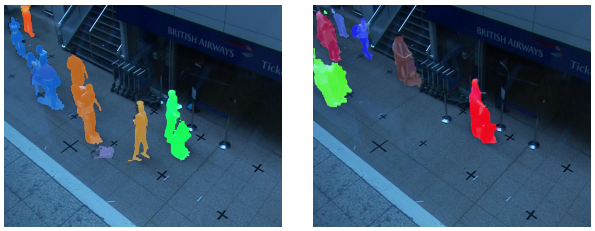


Fig. 12. Two examples of the detected blobs on a clip of PETS2007-REASON.

work with the same features in the experiments. Figure 10 plots the average MAEs as well as the training time of the three methods, when different numbers of training instances are used for both View 1 and View 2. The proposed FPR consistently yields better results than the other two methods, especially when fewer training instances are available. It validates that the scaling law can regularize the learning procedure, and compensate for the lack of training data. As the size of training data becomes larger, the task of perspective normalization can be done more reliably. Therefore, the performance gain gradually shrinks to almost zero. It is worth noting that the performance gains of FPR in View 2 is more significant, since the variation of pedestrian heights is much larger there due to the camera perspective. FPR takes longer training time in the iterative optimization procedure, but it is still within 15 seconds in the experiments. In the testing phase, the running time of our approach and the method using only Equation 1 are almost the same, since both the two approaches use the learned weight vector $\mathbf{w}$, via either Equation 1 or Equation 8, of the same dimension.

Features in the second-pass regression. In this work, we suggested four types of features to learn a residual estimator in the second pass; they are respectively termed as *Positive Trim* (P.T.), *Negative Trim* (N.T.), *Cross-camera Conflict* (C.C.), and *Intra-camera Conflict* (I.C.). We conducted two experiments to know their individual contributions to the second-pass regression. First, four second-pass regressors, each of which is learned with one specific type of features, are derived, and their performance in terms of MAE is shown in the top row of Figure 11. Although feature types N.T. and C.C. returned with better results, there is no significant performance difference among the four types of features. Second, their corresponding kernel weights, optimized by SimpleMKL, are plotted in the second row of Figure 11. It can be observed that feature types N.T. and C.C. get higher weights in most cases, while feature type P.T. is almost ignored. This may result from that P.T. is less complementary to the other feature types. It is known that the optimized kernel weights in multiple kernel learning are related to not only their individual powers but also their diversity.

TABLE III
PERFORMANCE EVALUATION ON THE VIDEO OF A CHALLENGING SCENE.

Camera	RBF	LIN	FPR	AVG	TPR
Veiw 1	26.0	9.6	14.4	9.5	3.0
Veiw 4	17.2	6.0	6.4	9.2	4.0

Blob extraction sensitivity study. In this study, we examined the performance sensitivity to the accuracy of blob extraction in challenging scenes using the PETS2007-REASON [2] dataset. In this dataset, the video clips film a near view of a departure lobby. The challenges are its poor video quality and some non-pedestrian moving objects such as luggage and trolleys, leading to error counting of each extracted blob. Based on the blob extraction results, we evaluate the performance of FPR as well as two recently proposed approaches, i.e., Gaussian process regression [1] and neural networks [25] for comparison. The experimental setups are the same as those used on PETS 2010. Namely, the first half of frames of each video serve as training data, while the rest as test data. The MAE over all test frames yielded by Gaussian process regression, neural networks, and FPR are 1.81, 0.99, and 0.85, respectively. The result show that our FPR yields superior performance to the feature weighting approaches which coordinate powerful machine learning tools to learn a people counting model. Moreover, as revealed in Figure 12, we note that fusing feature weights together with the regressor, the approach couples well with the extracted blobs in the cases where clutter backgrounds or partial occlusions due to the camera perspective appear.

Robustness in challenging scenes. To evaluate the robustness of the proposed method in challenging cases, we conducted an additional experiment on a synchronous clip captured by two cameras whose official labels are View 1 and View 4 in PETS 2010. We used the video frames with timestamps 14-06 in *regular flow* of *S0*. The challenges of the videos result from the large discrepancy between the two camera views as well as the heavy occlusions in View 4, as shown in Figure 13. Therefore, the estimated sizes of matched blobs in the different views were usually inconsistent. Following the previous experiments, we took the first half of frames as training data and evaluated the performance on the rest. Only video frames with nearly the same people counts in the two views were selected. Thus, the training and evaluation frames used in this experiment are different from those in the previous experiments. As displayed in Table III, the MAE of FPR for View 1 and View 4 are 14.4 and 6.4, respectively; the method FPR outperforms the RBF baseline significantly and is worse than the LIN approach. As for the performance degradation of FPR, we considered the main reason is that there are fewer training data in this experiment, but there are more optimization variables to be determined in FPR than baseline LIN. Nevertheless, our approach TPR can effectively leverage the information from the two camera views, remarkably reduce the counting errors in the first pass, and achieve promising results.

[2]http://www.cvg.rdg.ac.uk/PETS2007/.

(a) View 1 (b) View 4

Fig. 13. Sample frame of the detected blobs in View 1 and View 4 on a clip of PETS2009.

V. CONCLUSIONS

To resolve the difficulties of people counting, we have proposed a multiview system that transfers knowledge between multiple cameras. The contribution of this work is threefold. First, we propose a technique for counting the number of people in images captured by multiple cameras. Integrating corrections of the perspective effect and estimation of the size of a crowd, makes the intra-camera visual features more effective and yields an accurate and scalable counting model. Second, to explore inter-camera knowledge, we present a two-pass regression framework that has shown promise in exploiting and adapting heterogeneous information to handle the difficult aspects of people counting, such as mutual occlusions, imperfect foreground segmentations, and shadows. Finally, we match blobs to compensate for the variations among cameras, and propose a blob matching algorithm that derives a set of consistent entities from different views. The algorithm ensures that knowledge sharing is successful. Furthermore, the results of experiments on the PETS 2010 benchmark dataset demonstrate that our first-pass regression method and two-pass regression framework enhance the performance of people counting significantly.

APPENDIX

Suppose that $f^{(i)}$ and $k^{(i)}$ denote the focal length and the radial lens distortion coefficient of camera P_i, respectively. Let $\left(\sigma_x^{(i)}, \sigma_y^{(i)}\right)$ represent the center of radial lens distortion, and let $S_x^{(i)}$ represent the image scale factor which accounts for any uncertainty caused by imperfection of hardware timing and digitization. The reverse variant that transforms from the image coordinate system of camera P_i to the world coordinate system is described step by step as follows.

1) Transform computer image coordinate $\left(u_f^{(i)}, v_f^{(i)}\right)$ into real image coordinate $\left(u_d^{(i)}, v_d^{(i)}\right)$ by

$$u_d^{(i)} = \frac{\left(u_f^{(i)} - \sigma_x^{(i)}\right) \cdot d_x^{(i)}}{S_x^{(i)}} \quad \text{and} \tag{19}$$

$$v_d^{(i)} = \left(v_f^{(i)} - \sigma_y^{(i)}\right) \cdot d_y^{(i)}, \tag{20}$$

where $d_x^{(i)}$ and $d_y^{(i)}$ are the distances between adjacent sensor elements in x and y directions, respectively.

2) As indicated in [36], to avoid causing numerical instability, only radial distortion is considered for machine

vision application. In other words, we only need one term here. Therefore, we transform distorted image coordinate $\left(u_d^{(i)}, v_d^{(i)}\right)$ into un-distorted one $\left(u_n^{(i)}, v_n^{(i)}\right)$ by

$$u_n^{(i)} = u_d^{(i)} \cdot \left(1 + k^{(i)} \cdot \left({u_d^{(i)}}^2 + {v_d^{(i)}}^2\right)\right) \quad \text{and} \quad (21)$$

$$v_n^{(i)} = v_d^{(i)} \cdot \left(1 + k^{(i)} \cdot \left({u_d^{(i)}}^2 + {v_d^{(i)}}^2\right)\right). \quad (22)$$

3) Compute the transformation from undistorted image coordinate $\left(u_n^{(i)}, v_n^{(i)}\right)$ to its real world coordinate (u_w, v_w) on the desired ground plane using an inverted perspective projection with pinhole camera geometry, i.e.,

$$u_w = \frac{1}{\lambda^{(i)}} \left(v_n^{(i)} \begin{bmatrix} r_2^{(i)} \\ r_8^{(i)} \end{bmatrix}^T \begin{bmatrix} T_Z^{(i)} \\ -T_X^{(i)} \end{bmatrix} + u_n^{(i)} \begin{bmatrix} r_8^{(i)} \\ r_5^{(i)} \end{bmatrix}^T \begin{bmatrix} T_Y^{(i)} \\ -T_Z^{(i)} \end{bmatrix} - f^{(i)} \begin{bmatrix} r_2^{(i)} \\ r_5^{(i)} \end{bmatrix}^T \begin{bmatrix} T_Y^{(i)} \\ T_X^{(i)} \end{bmatrix} \right) \quad \text{and} \quad (23)$$

$$v_w = -\frac{1}{\lambda^{(i)}} \left(v_n^{(i)} \begin{bmatrix} r_1^{(i)} \\ r_7^{(i)} \end{bmatrix}^T \begin{bmatrix} T_Z^{(i)} \\ -T_X^{(i)} \end{bmatrix} + u_n^{(i)} \begin{bmatrix} r_7^{(i)} \\ r_4^{(i)} \end{bmatrix}^T \begin{bmatrix} T_Y^{(i)} \\ -T_Z^{(i)} \end{bmatrix} - f^{(i)} \begin{bmatrix} r_1^{(i)} \\ r_4^{(i)} \end{bmatrix}^T \begin{bmatrix} T_Y^{(i)} \\ T_X^{(i)} \end{bmatrix} \right), \quad (24)$$

where the scalar $\lambda^{(i)}$ is defined as

$$\lambda^{(i)} = v_n^{(i)} \begin{bmatrix} r_1^{(i)} \\ r_2^{(i)} \end{bmatrix}^T \begin{bmatrix} r_8^{(i)} \\ -r_7^{(i)} \end{bmatrix} + u_n^{(i)} \begin{bmatrix} r_5^{(i)} \\ r_4^{(i)} \end{bmatrix}^T \begin{bmatrix} r_7^{(i)} \\ -r_8^{(i)} \end{bmatrix} - f^{(i)} \begin{bmatrix} r_1^{(i)} \\ r_2^{(i)} \end{bmatrix}^T \begin{bmatrix} r_5^{(i)} \\ r_4^{(i)} \end{bmatrix}, \quad (25)$$

while

$$R^{(i)} = \begin{bmatrix} r_1^{(i)} & r_2^{(i)} & r_3^{(i)} \\ r_4^{(i)} & r_5^{(i)} & r_6^{(i)} \\ r_7^{(i)} & r_8^{(i)} & r_9^{(i)} \end{bmatrix} \quad \text{and} \quad T^{(i)} = \begin{bmatrix} T_X^{(i)} \\ T_Y^{(i)} \\ T_Z^{(i)} \end{bmatrix} \quad (26)$$

are the rotation matrix and translation vector defined in [36], respectively.

Once the image-to-world transformation between $\left(u_f^{(i)}, v_f^{(i)}\right)$ and (u_w, v_w), and the world-to-image transformation between (u_w, v_w) and $\left(u_f^{(j)}, v_f^{(j)}\right)$ are determined, every pixel on the bottom boundary in camera P_i can be mapped to the image plane of camera P_j.

REFERENCES

[1] A. Chan, Z. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.

[2] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos, "Estimating pedestrian counts in groups," *Computer Vision and Image Understanding*, vol. 110, no. 1, pp. 43–59, 2008.

[3] A. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting," in *Proc. Int'l Conf. Computer Vision*, 2009.

[4] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010.

[5] Y.-L. Hou and G. K. H. Pang, "People counting and human detection in a challenging situation," *IEEE Trans. Systems, Man, and Cybernetics, A*, vol. 41, pp. 24–33, 2011.

[6] T.-Y. Lin, Y.-Y. Lin, M.-F. Weng, Y.-C. Wang, Y.-F. Hsu, and H.-Y. M. Liao, "Cross camera people counting with perspective estimation and occlusion handling," in *Proc. Int'l Workshop Information Forensics and Security*, 2011.

[7] H. Ma, C. Zeng, and C. X. Ling, "A reliable people counting system via multiple cameras," *ACM Trans. Intelligent Systems and Technology*, vol. 3, no. 2, pp. 1–22, 2012.

[8] Y. Zhou and J. Luo, "A practical method for counting arbitrary target objects in arbitrary scenes," in *Proc. Int'l Conf. Multimedia and Expo*, 2013.

[9] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[10] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, and L.-Q. Xu, "Crowd analysis: A survey," *Machine Vision and Applications*, vol. 19, pp. 345–357, 2008.

[11] J. Jacques Junior, S. Musse, and C. Jung, "Crowd analysis using computer vision techniques," *IEEE Signal Processing Magazine*, vol. 27, pp. 66–77, 2010.

[12] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. Systems, Man, and Cybernetics, A*, vol. 31, pp. 645–654, 2001.

[13] V. Rabaud and S. Belongie, "Counting crowd moving objects," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2006.

[14] N. Ahuja and S. Todorovic, "Extracting texels in 2.1D natural textures," in *Proc. Int'l Conf. Computer Vision*, 2007.

[15] G. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2006.

[16] N. Dalal and B. Triggs, "Histogram of oriented gradient for human detection," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

[17] P. Felzenszwalb, D. McAllester, and D. Ramananand, "A discriminatively trained, multiscale, deformable part model," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.

[18] B. Wu and R. Nevatia, "Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses," *Int'l J. Computer Vision*, vol. 82, no. 2, pp. 185–204, 2009.

[19] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.

[20] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.

[21] M. Jones and D. Snow, "Pedestrian detection using boosted features over many frames," in *Proc. Int'l Conf. Pattern Recognition*, 2008.

[22] D. Kong, D. Gray, and H. Tao, "Counting pedestrians in crowds using viewpoint invariant training," in *Proc. British Conf. Machine Vision*, 2005.

[23] N. Paragios and V. Ramesh, "A MRF-based approach for real-time subway monitoring," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2001, pp. 1034–1040.

[24] R. Ma, L. Li, W. Huang, and Q. Tian, "On pixel count based crowd density estimation for visual surveillance," in *Proc. Conf. Cybernetics and Intelligent Systems*, 2004.

[25] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Proc. Conf. Digital Image Computing: Techniques and Applications*, 2009.

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

[27] Z. Ma and A. Chan, "Crossing the line: Crowd counting by integer programming with local features," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013.

[28] Y. Cong, H. Gong, S.-C. Zhu, and Y. Tang, "Flow mosaicking: Real-time pedestrian counting without scene-specific learning," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.

[29] M.-F. Weng, Y.-Y. Lin, N. C. Tang, and H.-Y. M. Liao, "Visual knowledge transfer among multiple cameras for people counting with occlusion handling," in *Proc. ACM Conf. Multimedia*, 2012.

[30] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505–519, 2009.

[31] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.

[32] D. Arsié, B. Schuller, and G. Rigoll, "Multiple camera person tracking in multiple layers combining 2D and 3D information," in *Proc. of M2SFA2*, 2008.

[33] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] Y. Xiong and F. Quek, "Meeting room configuration and multiple camera calibration in meeting analysis," in *Proc. ACM Conf. Multimodal Interfaces*, 2005.

[35] C. Aslan, K. Bernardin, and R. Stiefelhagen, "Automatic calibration of camera networks based on local motion features," in *Proc. of M2SFA2*, 2008.
[36] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
[37] J. Liu, R. Collins, and Y. Liu, "Surveillance camera autocalibration based on pedestrian height distributions," in *Proc. British Conf. Machine Vision*, 2011.
[38] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. ACM Conf. Multimedia*, 2007.
[39] L. Duan, I. Tsang, D. Xu, and S. Maybank, "Domain transfer SVM for video concept detection," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2009.
[40] S. Bickel, M. Bruckner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *Proc. Int'l Conf. Machine Learning*, 2007.
[41] F.-J. Chang, Y.-Y. Lin, and M.-F. Weng, "Cross-database transfer learning via learnable and discriminant error-correcting output codes," in *Proc. Asian Conf. Computer Vision*, 2012.
[42] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2005.
[43] Y.-G. Jiang, J. Wang, S.-F. Chang, and C.-W. Ngo, "Domain adaptive semantic diffusion for large scale context-based video annotation," in *Proc. Int'l Conf. Computer Vision*, 2009.
[44] M.-F. Weng and Y.-Y. Chuang, "Cross-domain multicue fusion for concept-based video indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1927–1941, 2012.
[45] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Processing*, vol. 20, no. 6, pp. 1709–1724, 2011.
[46] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2006.
[47] P. Dollar, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *Proc. British Conf. Machine Vision*, 2010.
[48] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Math. Program.*, vol. 107, no. 3, pp. 391–408, 2006.
[49] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," 2011.
[50] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.
[51] Y. Abdel-Aziz and H. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," in *Proc. the Symposium on Close-Range Photogrammetry*, 1971.
[52] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1147–1160, 2011.
[53] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.
[54] J. Ferryman and A. Ellis, "PETS2010: Dataset and challenge," in *Proc. Conf. Advanced Video and Signal Based Surveillance*, 2010.

Nick C. Tang received the B.S. and M.S. degrees from Tamkang University, Tamsui, Taiwan, in 2003 and 2005, respectively. He also received the Ph. D. degree from Tamkang University in 2008. Currently, he is a Postdoctoral Fellow with the Institute of Information Science, Academia Sinica, Taiwan. His research interests include image and video analysis, computer vision, computer graphics, and their applications.

Yen-Yu Lin received the B.S. degree in information management, and the M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, in 2001, 2003, and 2010, respectively. He is currently an Assistant Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. His current research interests include computer vision, pattern recognition, and machine learning. He is a member of the IEEE.

Ming-Fang Weng received the B.S. degree and M.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1998 and 2000 respectively, Ph.D. degree from National Taiwan University in 2010, all in computer science and information engineering. He was a Postdoctoral Fellow in the Institute of Information Science, Academia Sinica, Taiwan, and is currently a Principal Engineer in the Institute for Information Industry, Taiwan. His research interests include digital content analysis, image/video information retrieval, computer vision, and multimedia applications.

Hong-Yuan Mark Liao received his Ph.D degree in electrical engineering from Northwestern University in 1990. In July 1991, he joined the Institute of Information Science, Academia Sinica, Taiwan and currently, is a Distinguished Research Fellow. He has worked in the fields of multimedia signal processing, image processing, computer vision, pattern recognition, video forensics, and multimedia protection for more than 25 years. During 2009-2011, he was the Division Chair of the computer science and information engineering division II, National Science Council of Taiwan. He is jointly appointed as a Professor of the Computer Science and Information Engineering Department of National Chiao-Tung University and the Department of Electrical Engineering and Computer Science of National Cheng Kung University. During 2009-2012, he was jointly appointed as the Multimedia Information Chair Professor of National Chung Hsing University. Since August 2010, he has been appointed as an Adjunct Chair Professor of Chung Yuan Christian University. Since August 2014, he has been appointed as an Honorary Chair Professor of National Sun Yat-sen University. He received the Young Investigators' Award from Academia Sinica in 1998; the Distinguished Research Award from the National Science Council of Taiwan in 2003, 2010 and 2013; the National Invention Award of Taiwan in 2004; the Distinguished Scholar Research Project Award from National Science Council of Taiwan in 2008; and the Academia Sinica Investigator Award in 2010. His professional activities include: Co-Chair, 2004 International Conference on Multimedia and Exposition (ICME); Technical Co-chair, 2007 ICME; General Co-Chair, 17th International Conference on Multimedia Modeling; President, Image Processing and Pattern Recognition Society of Taiwan (2006-08); Editorial Board Member, IEEE Signal Processing Magazine (2010-13); Associate Editor, IEEE Transactions on Image Processing (2009-13), IEEE Transactions on Information Forensics and Security (2009-12) and IEEE Transactions on Multimedia (1998-2001). He has been a Fellow of the IEEE since 2013 for contributions to image and video forensics and security. Currently, he also serves as IEEE Signal Processing Society Region 10 Director (Asia-Pacific Region).