Visual Knowledge Transfer among Multiple Cameras for People Counting with Occlusion Handling

Ming-Fang Weng
Institute of Information Science,
Academia Sinica, Taipei 115, Taiwan
mfueng@iis.sinica.edu.tw

Yen-Yu Lin
Research Center for Information Technology
Innovation, Academia Sinica, Taipei 115, Taiwan
yylin@citi.sinica.edu.tw

Nick C. Tang
Institute of Information Science,
Academia Sinica, Taipei 115, Taiwan
nickctang@gmail.com

Hong-Yuan Mark Liao
Institute of Information Science,
Academia Sinica, Taipei 115, Taiwan
liao@iis.sinica.edu.tw

ABSTRACT

We present a framework to count the number of people in an environment where multiple cameras with different angles of view are available. We consider the visual cues captured by each camera as a knowledge source, and carry out cross-camera knowledge transfer to alleviate the difficulties of people counting, such as partial occlusions, low-quality images, clutter backgrounds, and so on. Specifically, this work distinguishes itself with the following contributions. First, we overcome the variations of multiple heterogeneous cameras with different perspective settings by matching the same groups of pedestrians taken by these cameras, and present an algorithm for accomplishing cross-camera correspondence. Second, the proposed counting model is composed of a pair of collaborative regressors. While one regressor measures people counts by the features extracted from intra-camera visual evidences, the other recovers the yielded residual by taking the conflicts among inter-camera predictions into account. The two regressors are elegantly coupled, and jointly lead to an accurate counting system. Additionally, we provide a set of manually annotated pedestrian labels on the PETS 2010 videos for performance evaluation. Our approach is comprehensively tested in various settings and compared with competitive baselines. The significant improvement in performance manifests the effectiveness of the proposed approach.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing method*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*video analysis*.

General Terms

Algorithms, Theory, Experimentation.

Keywords

People counting, Transfer learning, Correspondence estimation.

MM'12, October 29–November 2, 2012, Nara, Japan.

1 Introduction

The goal of *people counting*, such as [9, 24, 27, 29, 31, 36], is to estimate the number of people or the density of crowds in a monitored environment. It is receiving recent attention in many fields, e.g., multimedia, computer vision, and video surveillance, since it plays a critical role in a broad range of applications. First of all, pedestrian counts serve as one of the most important indices for video data, and hence counting the number of people has become a key component in video understanding, retrieval, and surveillance. Further, both the long-term and short-term statistics of people counts of an environment provide useful information for strategy planning or event detection. For instance, one could measure the degree of crowdedness for traffic flow planning, analyze the benefit of the advertisement for potential revenue estimation, or identify abnormal gatherings for crime prevention. In addition, researches on people counting can be applied to adaptive device control for energy-saving purposes. For example, the number of people in a transit station is helpful in deciding how low the A/C (air conditioning) temperature should go or how many escalators should be operated.

Despite the great applicability of people counting, most of vision-based counting systems are still hindered by the following challenges. First, mutual occlusion among people yields the severe change of pedestrian appearances or the loss of extracted features. It often leads to the underestimate of the number of people. This situation becomes worse in crowded environments. Second, the problems caused by low-resolution or blur images, especially for the pedestrians far from the camera, often degrade the stability of a counting system. Further, the large variations in pedestrian appearances, and lighting conditions, or clutter backgrounds make people counting quite difficult.

In this work, we address the aforementioned issues by establishing a *multiple camera people counting* (MCPC) system, where multiple cameras are installed to monitor an identical region but with different angles of view. Videos taken by these cameras contain complementary information. It follows that fusing visual cues in these videos generally facilitate the accomplishment of a more robust and accurate counting system. An illustration of our idea is given in Figure 1, where two different views of an environment are available, while two examples of complementary visual cues catched by the two camera views are shown. In the first example, View 2 consists of useful information to compensate for the underestimate caused by occlusions in View 1. In the other, View 1 provides high-resolution visual cues to enhance the performance of estimation for View 2. So far, two questions arise: 1) How to

Figure 1: Illustration of our multiple camera people counting (MCPC) system in which visual knowledge are transferred across cameras to establish a more accurate and robust counting system.

work with multiple cameras so that all the captured visual cues can be shared across different camera views? 2) For each view, how to couple both the *intra-camera* and *inter-camera* visual knowledge to result in a more robust and accurate counting system?

To work with the multiple cameras with diverse perspective settings, we propose a correspondence estimation algorithm that maps each segmented group of moving people in one view to its corresponding group in another view. We call these matched groups *components*, upon each of which knowledge can be transferred across cameras. The identity numbers in Figure 1 index the matched components. In this situation, both the intra-camera (captured by itself) and inter-camera (transferred from others) visual cues are available in each view, and we present a two-pass regression framework to carry out the multiple camera people counting system. Specifically, the first-pass regressor measures the people count by using the visual features extracted from the intra-camera knowledge. The second-pass regressor estimates the residual yielded in the first pass by taking the conflicts among inter-camera knowledge into account. Since the second-pass regression is built on knowledge conflicts, we cast the training of the second-pass regressor as a transfer learning problem [33], in which useful knowledge transfer is encouraged while error propagation is prevented.

This work distinguishes itself with the following contributions. First, we propose to incorporate multiple heterogeneous cameras with different perspective settings via matching the components, and present an algorithm for cross-camera correspondence estimation with high precision. Second, a pair of collaborative regressors is introduced to accomplish cross-camera knowledge transfer. The two regressors are elegantly coupled so that both intra-camera and inter-camera visual knowledge are taken into account simultaneously, and lead to a more robust counting system. Third, we create a set of manually annotated pedestrian labels on the PETS 2010 videos which are publicly accessible in the research community.

2 Related Work

In this section, we review a few research topics and techniques relevant to the establishment of the proposed framework, including people counting, correspondence estimation among multiple cameras, and transfer learning.

2.1 People Counting

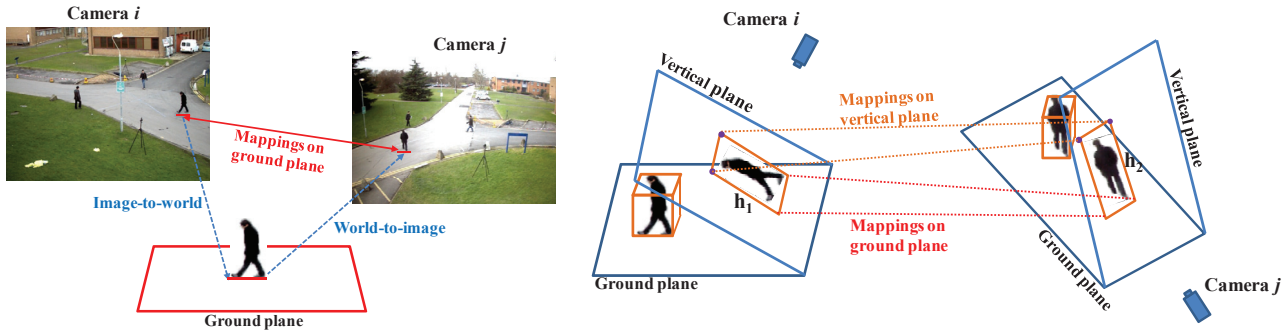
The literature of people counting is quite comprehensive [19, 44]. We focus on only the computer-vision-based methods, since they are closely related to our work. For the ease of discussion, these methods are divided into the two categories, *counting-by-detection* and *counting-by-regression*.

Counting-by-detection. Methods of this category determine the number of people present in an image by explicitly locating the position of each individual pedestrian and then counting the total number of detected ones. The work by Lin et al. [28] is a good example of this category, in which they proposed to learn head-like contours and exhaustively search for human heads. In addition to the visual features that are embedded in static images, motion features that can be extracted from sequences of video frames, e.g., dynamic textures and motion patterns, have been shown to be useful to detect individual moving entities [1, 8, 34].

These methods work well in controlled environments, but they probably result in suboptimal performance in a more general setting because most of them cannot clearly define the appearance or the shape of pedestrians in advance. The trend of this kind of systems is to employ a pedestrian detector to find people [21, 26]. While pedestrian detectors have been demonstrated to be robust to variations caused by different illumination, poses, and image sizes [12, 13, 32], the performance of detection results can be farther improved by modeling the geometric structures among body parts [41]. However, training such detectors in general requires a great number of manually collected data. Moreover, since the training samples are usually of a high resolution and no occlusions are involved, the performance of these detectors severely degrades when they are applied to find pedestrians with partial occlusions or of low resolutions. Additionally, the computational time at detection stage is typically too high to support yielding real-time responses.

Counting-by-regression. Methods of this class is relatively more efficient. They estimate the size of a group via extracting low-level features to represent the corresponding region which is typically assumed to be yielded by background subtraction or motion segmentation approaches [23, 24, 27]. Since these methods do not solve the localization problem, they are suitable for estimating the density of a crowd with a certain level of crowdedness rather than precisely evaluating the exact number of people appearing in a scene. Followed by the methods which linearly map a set of perspectively normalized features to the number of people, nonlinear regression models such as neural networks, Gaussian process, and Poisson process, recently are adopted to enhance the performance of people counting [9, 10, 36].

Despite the use of discriminative visual features and powerful machine learning techniques, people counting systems of both the two categories suffer from the problems caused by occlusions, low-quality images, and large variations of pedestrian appearances. It is worth mentioning that the most similar work to ours is that of Ma et al. [31], in which visual cues from two cameras are fused to



(a) Cross-camera ground plane matching

(b) Ground plane matching vs. Vertical plane matching

Figure 2: The proposed approach to cross-camera blob localization and matching.

deal with the problem resulting from occlusions. They combine human detection results from two cameras with the verification of the *homography constraint*. However, their system depends on the reliable results of the two single-view detection. As mentioned above, methods developed upon pedestrian detectors do not work well with partial or heavy occlusions. Thus the abilities of occlusions handling in their approach may be restricted in crowded environments. Unlike [31], we investigate the evidences revealed in the prediction confliction among multiple cameras, and cast the estimation of occlusions as an instance of transfer learning. Our system can deal with the problem of underestimate even if heavy occlusions appear in all the camera views.

2.2 Correspondence among Multiple Cameras

One critical task of synchronizing multiple cameras and realizing visual knowledge transfer is to establish the correspondence among these cameras. Conventional methods of camera correspondence estimation can be roughly divided into the following two categories.

Homographic-based Methods. In [22], Khan and Shah integrate multi-view information by introducing a *planar homographic constraint*. They first estimate plane homographies by matching the SIFT [30] features, and then discover the correspondence among multiple cameras. Eshel and Moses [15] propose to detect head blob of each person via the *multiple height planar homographies*, and alleviate the problem of performance degradation in tracking occluded objects. Arsie et al. [2] suggest to track people based on a homographic transformation between the target blob boundaries. However, methods mentioned above are sensitive to the large variation of appearances of objects, perspective settings of cameras, and video qualities by heterogeneous camcorders. This is because that homographic-based approaches rely on consistent matching. However, it is generally not available in real environments.

Calibration-based Methods. The goal of camera calibration is to estimate the model of a camera, including not only the extrinsic parameters, e.g., position and orientation relative to the real world coordinate system, but also the intrinsic parameters of the camera, e.g., image central, focal length, and distortion coefficients. Xiong and Quek [42] present an approach, in which a box with the specific pattern of dots and markers is employed, to calibrate both the intrinsic and extrinsic parameters of camera networks. Aslan et al. [3] develop a set of features based on the positions of all the detected pedestrians' heads, and automatically calibrate the extrinsic parameters of multiple cameras via these features. Tsai [38] present a two-stage approach to computing the position, orientation, and the intrinsic parameters of a camera. This approach is adopted in a wide range of applications, since it is capable of dealing with both the *coplanar* and *non-coplanar* points. Having a precise planar transformation among multiple cameras would facilitate counting people in the crowded environments. Generally speaking, calibration-based methods tend to provide more precise camera transformations than the ones by homographic-based methods, and hence are more suitable in our cases.

2.3 Transfer Learning

Transfer learning [33] refers to an information delivering process that aims to improve the target task by leveraging abundant knowledge available in the source tasks. The exploration of auxiliary knowledge drawn from different tasks has received a rapidly growing interest in the field of machine learning. The methods exploiting additional knowledge sources to benefit the accomplishment of the underlying task can be generally divided into four categories [33]: transfer by *model parameters* [14, 43], by *data instances* [7], by *feature representation* [6], and by *contextual information* [20, 40]. However, these methods are established upon the assumption that data 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 a knowledge source, and aim to establish a robust counting system via sharing knowledge across cameras.

3 Cross-camera Blob Matching

We propose to make use of visual information from different camera views to alleviate the problems, such as partial occlusions or imperfect foreground segmentations, in people counting. To work with heterogeneous cameras together with various perspective settings, one of the most important tasks is to *match* the corresponding *blobs* among these cameras, so that visual knowledge can be transferred across cameras and support the establishment of an accurate counting system.

3.1 Blob Extraction

We represent a video frame by a set of *blobs*, each of which is a group of spatially connected foreground (moving) pixels. We consider blob as a natural unit for people counting, since 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. To extract foreground blobs, we first apply the background subtraction algorithm [5] to every video frame to segment out the foreground areas. Then, spatially connected foreground pixels are clustered to yield the blobs. Each extracted blob in Figure 1 is highlighted by a specific color.

3.2 Blob Localization and Matching

Once the blobs from all the camera views are extracted, our goal is to localize and match the corresponding blobs across cameras.

(a) Bottom boundaries of blobs (b) Their mappings

Figure 3: Blob mapping on the ground plane.

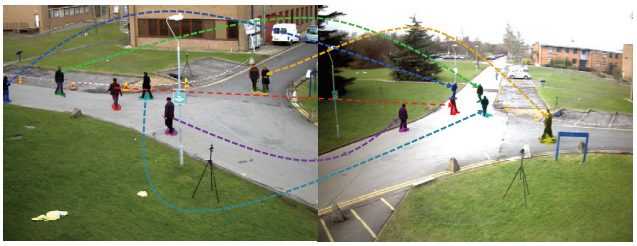


Figure 4: Blob mapping on both the ground and vertical planes.

To this end, we propose a blob matching algorithm, including the *ground plane mapping* and the *vertical plane mapping*. We illustrate the localization algorithm in Figure 2.

First, we develop a reverse variant of Tsai's camera calibration model [38], which computes the transformation from the *image coordinate system* to the *world coordinate system* on the *ground plane*, to generate the *world-to-image* coordinate transformation. Then, we detect the *bottom boundary* of each blob. To this end, the *convex hull* of each blob is generated, such that the bottom boundary of the blob can be determined by the bottom contour. Some examples of the detected bottom boundaries are shown in Figure 3(a). Each of them is drawn with one particular color. Based upon the assumption that the bottom boundary of a blob touches the ground plane, its correspondence across cameras hence can be fulfilled via Tsai's camera calibration model and its reverse variant. This procedure is summarized in Figure 2(a), in which every pixel residing on the bottom boundary in camera i is first transformed to the world coordinate system, and then transformed to the image plane of camera j. By the procedure, one can establish the mappings of the bottom boundaries of blobs between any the video frames taken by two distinct cameras. Figure 3(b) shows the mapping results where each correspondence is plotted with one distinct color. Symmetrically, we can also match blobs extracted in camera j to those in camera i. We use the set of these resulting mappings as an initial guess of the correspondence of each blob in two cameras. However, due to the imperfect blob segmentation, the bottom boundary of a blob does not always touch the ground plane in our empirical tests. Therefore, we need to validate the correctness of the estimated correspondence by using the mappings on the *vertical plane* of each blob.

For computing mappings on the vertical plane of each blob, we need to estimate the image height of a pedestrian at every position in the camera view. Based on the work by Hoiem et al. [18], we assume that the image height of a pedestrian (denoted as h) is linearly dependent on his/her bottom location (denoted as v) in the vertical position of the image, i.e.,

$$h(v) = \alpha \cdot v + \alpha_0, \tag{1}$$

where α and α_0 are the two parameters of the camera model. Thus, we adopt the procedure, described in Algorithm 1, to compute the perspective parameters of each camera via employing a off-the-shelf pedestrian detector [32]. 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.

After having the mapping of bottom boundary of each blob, we can further compute the head positions of the blob according to the estimated camera perspective model. It follows that based on both the mappings of the bottom boundary and the head position of each blob, we can calculate the *planar homographies*, project every pixel in the blob from camera i to camera j on the vertical plane, and vice versa. The relation between the ground plane matching and the vertical plane matching is given in Figure 2(b). An example of estimated mappings of blobs on the vertical planes is shown in Figure 4, where all the pixels of each blob is involved in the projection.

Algorithm 1 Automatic Perspective Parameter Estimation Algorithm. $(\alpha, \alpha_0) = \textbf{Algo}(\mathcal{D}, \tau, \mathrm{T})$. Given a set of detected windows, $\mathcal{D}$, and two parameters, τ and T, which are respectively a tolerance and the maximum number of iterations, return the camera's perspective parameters, (α, α_0).

let d_i be a detected pedestrian in $\mathcal{D}$ and v_i and h_i respectively represent the bottom position and the image height of d_i
initialize $\mathcal{D}^1 = \mathcal{D}$
for $t = 1$ **to** T **do**
  estimate parameters α and α_0 using $\mathcal{D}^t$ by minimizing $J^t(\alpha, \alpha_0) = \sum_{i=1}^{|\mathcal{D}|} (\alpha \cdot v_i + \alpha_0 - h_i)^2$
  build $\mathcal{D}^{t+1} = \{d_i | (\alpha \cdot v_i + \alpha_0 - h_i) \leq \tau,\ d_i \in \mathcal{D}^t\}$
  if $\mathcal{D}^{t+1}$ and $\mathcal{D}^t$ are the same **then**
    return perspective camera parameters, α and α_0
  end if
end for

3.3 Matched Component Extraction

After the correspondences of blobs between two cameras are established, we can group blobs into *components*, in each of which the same entities present in both the two camera views. Let's illustrate the grouping process by the example given in Figure 5. Suppose that $\{b_1^{(i)}, b_2^{(i)}, \ldots, b_6^{(i)}\}$ and $\{b_1^{(j)}, b_2^{(j)}, \ldots, b_4^{(j)}\}$ are the extracted blobs in camera i and j, respectively. A *bipartite graph* of ten nodes is then constructed. An edge between two nodes in the opposite sides is added if the two corresponding blobs are matched in any direction. Via computing the *connected components* in the graph, components $\{c_1^{(i)}, c_2^{(i)}, c_3^{(i)}\}$ and $\{c_1^{(j)}, c_2^{(j)}, c_3^{(j)}\}$ are obtained for camera i and j, respectively. Note that since each corresponding component refers to the same group of pedestrians in both the two cameras, it implies that visual cues of the component captured by the two cameras can be shared directly. That is, we in this work propose to match the components, upon each of which visual knowledge can be transferred across cameras.

4 Two-pass Regression

This section describes how the proposed MCPC system improves the accuracy of people counting by incorporating visual cues extracted from a specified camera with the knowledge shared by other collaborative cameras. Suppose that we have a set of M cameras, $\mathbf{P} = \{P_m\}_{m=1}^{\mathrm{M}}$, which monitor an environment. Let $\mathbf{V} = \{V_m\}_{m=1}^{\mathrm{M}}$ be the videos taken by these cameras respectively. Without loss of generality, we assume each video consists of T

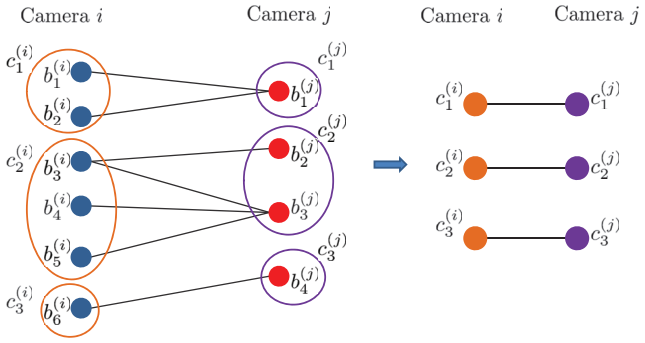


Figure 5: Matching components across cameras.

frames and all the frames are *synchronized* across different cameras. In the following, we use $I_t^{(m)}$ to denote the tth frame of V_m. Our goal here is to yield a matrix $\boldsymbol{A} \in \mathbb{R}^{\mathrm{M}\times\mathrm{T}}$ whose element $a_{m,t}$ is the prediction of the number of people present in frame $I_t^{(m)}$.

Our approach is an instance of the *counting-by-regression* systems that map a set of unintelligible features to the prediction of the people count, but distinguishes itself by jointly considering both intra-camera and inter-camera knowledge. Specifically, the proposed two-pass regression framework is motivated by the observation that the number of people in an image can be approximated by two different parts—the *regular* part and the *residual* part, as illustrated in Figure 6. The estimations of the two parts are carried out by two regressors respectively. The estimation of the regular part is an inference on the number of people within an image blob according to its own low-level features, just like what a common counting-by-regression approach does. Due to many unavoidable effects such as the failure in foreground segmentation (like shadows), partial occlusions, and the likes, the inference based on low-level features is typically imperfect, and a residual is hence yielded. In our approach, we borrow the knowledge captured by the collaborative cameras to measure the residual, and cast it as an instance of the transfer learning problem.

In this work, both the tasks of the regular estimation and the residual estimation are accomplished by the multi-kernel extension of support vector regression [39]. In the following, a brief introduce to multi-kernel support vector regression is first given. The designs of the two regressors are then depicted respectively.

4.1 Multi-kernel Support Vector Regression

In view of the large variations in crowd appearance, it is usually difficult to find a single feature to well characterize blobs. Therefore, we turn to seek a set of weights to linearly and optimally combine the given features in a unified domain. To this end, we leverage *multiple kernel learning* (MKL) [4, 17, 25, 35, 37] to fuse a set of base kernel matrices or functions, each of which is created based on a specific kind of features, and to derive the regressor simultaneously.

Given D different feature descriptors, the kernel bank $\{K_d\}_{d=1}^{\mathrm{D}}$ can be constructed for data by the corresponding kernel functions $\{k_d(\cdot,\cdot)\}_{d=1}^{\mathrm{D}}$. Fusing data characteristics captured by multiple feature descriptors now can be achieved via kernel matrix combination, i.e.,

$$K = \sum_{d=1}^{\mathrm{D}} \beta_d K_d, \text{ s. t. } \beta_d \geq 0, \tag{2}$$

where β_d is the ensemble coefficient of base kernel K_d (or the weight for the dth kind of feature descriptors). Consequently, the multi-kernel support vector regressor f from a set of labeled training data $\{(\mathrm{x}_n, y_n \in \mathbb{R})\}_{n=1}^{\mathrm{N}}$ can typically be expressed by

$$\begin{aligned} f(\mathrm{x}) &= \sum_{n=1}^{\mathrm{N}} \alpha_n y_n k(\mathrm{x}_n, \mathrm{x}) + b \\ &= \sum_{n=1}^{\mathrm{N}} \alpha_n y_n \sum_{d=1}^{\mathrm{D}} \beta_d k_d(\mathrm{x}_n, \mathrm{x}) + b. \end{aligned} \tag{3}$$

The task of MKL is to determine the optimal values of the sample coefficients $\{\alpha_n\}_{n=1}^{\mathrm{N}}$, the kernel weights $\{\beta_d\}_{d=1}^{\mathrm{D}}$, and the offset b. Actually, various objective functions, e.g., the *structural risk function* with ℓ_1- or ℓ_2-norm regularization, together with different optimization techniques, e.g., *semi-definite programming* (SDP) or *semi-infinite linear programs* (SILPs) are developed to accomplish the task. In this work, we adopt *SimpleMKL* [35] to learn the regressors due to its efficiency and effectiveness.

4.2 First-pass Regular Regression

The first-pass regressor works, including training and predicting, on extracted blobs. We describe the representation of blobs and the learning of the regressor as follows.

4.2.1 Blob Representation

Since knowledge for people counting is only shared among frames taken at the same time, for simplicity we consider only frames at time t. Thus, for frame $I_t^{(m)}$ taken by camera P_m, the index t can be dropped without ambiguity. As mentioned previously, each frame can be expressed as a set of blobs, i.e., $I^{(m)} = \{(\mathrm{x}_b^{(m)}, y_b^{(m)})\}_{b=1}^{\mathrm{n}_b}$, where $\mathrm{x}_b^{(m)}$ is the feature representation of the bth blob while $y_b^{(m)}$ is the number of people in the blob. In the training phase, $y_b^{(m)}$ is given while in the testing phase, it is exactly what we attempt to estimate. With this representation, the number of people in $I^{(m)}$ can be calculated by summing over the ones estimated in the blobs, i.e., $a_m = \sum_{b=1}^{\mathrm{n}_b} y_b^{(m)}$.

4.2.2 Learning with Intra-camera Visual Features

We have implemented three representative features to characterize the properties of blobs, including

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

Canny edge pixels. We calculate the total number of edge pixels, located by Canny edge detector, included in the blob to capture the structural properties of crowdedness.

Oriented gradients. The feature consists of two independent values representing the gradient magnitudes of vertical and horizontal orientations, respectively.

As these features capture diverse characteristics, we treat each one as a unique descriptor. Thus, each blob is represented by four different descriptors, each of which correspond to a specific kernel. To reduce the influence of perspective effects, all adopted features are normalized by applying geometric distortion correction coefficients [9] to approximate the real scale in the scene.

We adopt the multi-kernel support vector regression to implement the regular estimation in the first stage of our MCPC system. We learn the $\mathcal{F}^{(m)}$ by using a number of manually labeled blobs and the resulting four kernels. In the stage, training data are collected within a single camera, without referencing any information out of the camera. Besides, we also derive four additional support vector regressors $\{\mathcal{F}_d^{(m)}\}_{d=1}^{\mathrm{D}}$, where $\mathcal{F}_d^{(m)}$ is learned with the same training blobs, but only the dth kernel is considered. The value of

(a) Illustration of occlusion handling

(b) Illustration of shadow handling

Figure 6: Our two-pass regression framework consists of a regular estimation and a residual estimation. While the former infers the number of people of a blob based using intra-camera low-level features, the later estimates the residual by exploiting inter-camera information. Fusing both together compensates for the difference between the ground truth and the estimated value, thus yielding higher accuracy than either.

D is four here. All the procedures described above are repeated for each camera P_m.

4.3 Second-pass Residual Regression

The goal of this stage is to estimate and recover the yielded residual in the first pass by borrowing the visual knowledge captured by other collaborative cameras. The second-pass regressor works upon the *matched components*, upon which the additional inter-camera knowledge are transferred.

4.3.1 Component Representation

A video frame can also be represented by a set of components once we conduct the blob matching algorithm on all synchronized frames. Since we know that the corresponding components on different views refer to a group of the same entities, the number of people in each of them should be identical. The conflict of predictions among multiple views reveals that residual occurs. Based on this fact, knowledge like intra-camera estimation results shared among matched components can be directly adopted without requiring any further transformation or adaption. Suppose that frame $I^{(m)}$ consists of $\tilde{\mathrm{n}}_c$ components. Its *component representation* can be expressed as $\{(\mathrm{z}_c^{(m)}, \tilde{y}_c^{(m)})\}_{c=1}^{\tilde{\mathrm{n}}_c}$, where $\tilde{y}_c^{(m)}$ is the residual of component c yielded in the first pass, and $\mathrm{z}_c^{(m)}$ is the feature representation. Similar to the first pass, $\tilde{y}_c^{(m)}$ is given in training, and is what we need to measure in prediction. For the ease of discussion, the definitions of $\tilde{y}_c^{(m)}$ and $\mathrm{z}_c^{(m)}$ are given in the following section.

4.3.2 Learning with Inter-camera Visual Knowledge

Since a component is composed of a set of blobs, its residual in the training phase can be precisely computed. Let's illustrate this with an example component $\mathrm{z}^{(m)}$, taken by camera P_m, that consists of blobs $\{(\mathrm{x}_b^{(m)}, y_b^{(m)})\}_{b=1}^{N}$. The residual of $\mathrm{z}^{(m)}$ in the first stage can be defined as

$$\tilde{y}^{(m)} = \sum_{b=1}^{N} y_b^{(m)} - \sum_{b=1}^{N} \mathcal{F}^{(m)}(\mathrm{x}_b^{(m)}). \tag{4}$$

We now design the feature representation of component $\mathrm{z}^{(m)}$ by considering inter-camera knowledge. To begin with, the people counts of $\mathrm{z}^{(m)}$ predicted by support vector regressors $\{\mathcal{F}_d^{(m)}\}_{d=1}^{\mathrm{D}}$ are evaluated, i.e., ,

$$\mathbf{v}(\mathrm{z}^{(m)}) = [\mathcal{F}_1^{(m)}(\mathrm{z}^{(m)}) \ \cdots \ \mathcal{F}_D^{(m)}(\mathrm{z}^{(m)})]^\top \in \mathbb{R}^{\mathrm{D}}, \tag{5}$$

$$\text{where } \mathcal{F}_d^{(m)}(\mathrm{z}^{(m)}) = \sum_{b=1}^{N} \mathcal{F}_d^{(m)}(\mathrm{x}_b^{(m)}). \tag{6}$$

For the matched components taken by other cameras, we similarly have $\{\mathbf{v}(\mathrm{z}^{(m)})\}_{m=1}^{\mathrm{M}}$. Since component $\mathrm{z}^{(m)}$ refers to a group of the same entities taken by these cameras, the people counts estimated $\{\mathbf{v}(\mathrm{z}^{(m)})\}_{m=1}^{\mathrm{M}}$ can be compared directly. Further, the yielded conflict provides strong evidences to infer the residual. We hence develop the following four kinds of descriptors for component $\mathrm{z}^{(m)}$, i.e.,

Cross-camera Conflict. This descriptor directly captures the conflict between camera P_m and other cameras w.r.t. each of the adopted visual feature, i.e.,

$$\mathrm{z}^{(m)}.cc = \frac{\sum_{m'=1, m' \neq m}^{\mathrm{M}} \mathbf{v}(\mathrm{z}^{(m')})}{\mathrm{M}-1} - \mathbf{v}(\mathrm{z}^{(m)}). \tag{7}$$

Negative Trimmed. We are motivated by the observation that due to the camera angle relative to the motion direction of pedestrians or the distance to the monitored environment, people counts predicted from some camera tend to be underestimated. In the situation, only the positive part in Equation (7) are useful in residual estimation. This descriptor is hence defined as

$$\mathrm{z}^{(m)}.nt = \max(\mathrm{z}^{(m)}.cc, \mathbf{0}). \tag{8}$$

Positive Trimmed. Symmetrical to $\mathrm{z}^{(m)}.nt$, we also have descriptor

$$\mathrm{z}^{(m)}.pt = -\min(\mathrm{z}^{(m)}.cc, \mathbf{0}). \tag{9}$$

Intra-camera Conflict. Motivated by the fact that the sensitivities of visual features to occlusions are different, the prediction conflict among these features is also to recover the residual caused by occlusions [29]. This descriptor is designed as

$$\mathrm{z}^{(m)}.ic = [\mathcal{F}_i^{(m)}(\mathrm{z}^{(m)}) - \mathcal{F}_j^{(m)}(\mathrm{z}^{(m)})], \text{for } 1 \leq i < j \leq \mathrm{D}. \tag{10}$$

In the training phase, we match the blobs across cameras, and obtain a set of components. By measuring the residual in Equation (4) and extracting the four features in Equation (7) $\sim$ Equation (10) for each matched component, a support vector regressor, $\mathcal{S}^{(m)}$, with four kernels as input can be derived.

4.4 On People Counting

After the training procedure, the final regressors of the two passes $\{(\mathcal{F}^{(m)}, \mathcal{S}^{(m)})\}_{m=1}^{\mathrm{M}}$ are available. In the testing phase, suppose that we are given frame taken by camera P_m, and it is of blob representation $\{\mathrm{x}_b^{(m)}\}_{b=1}^{\mathrm{n}_b}$ and component representation $\{\mathrm{z}_c^{(m)}\}_{c=1}^{\tilde{\mathrm{n}}_c}$. Our approach estimates the number of people in this frame by

$$y_m = \sum_{b=1}^{\mathrm{n}_b} \mathcal{F}^{(m)}\left(\mathrm{x}_b^{(m)}\right) + \sum_{c=1}^{\tilde{\mathrm{n}}_c} \mathcal{S}^{(m)}\left(\mathrm{z}_c^{(m)}\right). \tag{11}$$

We complete this section by concluding that the proposed MCPC system integrates not only the intra-camera visual features but also the inter-camera visual knowledge, and alleviates the problems caused by occlusions, low-quality images, or imperfect background

segmentations. In the experiments, we show that it significantly outperforms single-view systems and the systems that simply average the outcomes of multiple views. Besides, it supports real-time people counting.

5 Experiments and Results

In this section we present the estimation performance of the proposed system and compare this with other competitive approaches. We first describe the experimental settings and then discuss the results of various methods.

5.1 Experimental Settings

To evaluate the performance of the proposed MCPC system, we conduct experiments on the PETS 2010 Benchmark Data [16]. This data set consists of a collection of videos captured by several stationary camcorders. These camcorders are located at distinct positions and set up to monitor the same scene. We select three representative video clips of two views, denoted as SPARSE, MEDIUM, and HEAVY, respectively. While SPARSE and MEDIUM are two videos with sparse individuals and small groups of pedestrians (i.e., less or moderate occlusions), HEAVY generally contains highly crowded groups (i.e., heavy occlusions). Since these selected videos span a wide spectrum of occlusion levels, collecting them together forms a good test bed to measure the effectiveness of occlusion handling for people counting. The details of the experimental data are given in Table 1.

Because supervised learning approaches require labeled examples, however, to the best of our knowledge, few proper benchmarks along with groundtruth annotations are publicly available in the computer vision or multimedia community. Therefore, we have put manual effort into annotating videos and obtained a large set of pedestrian labels on the PETS 2010 videos. To fuel people counting research, particularly using multiple cameras, these data are released and can be downloaded from our project website[1].

For performing quantitative evaluations for people counting, we divide each video clip into a non-overlapping training and testing set. We use the preceding half of frames of each video as training data and report the performance by testing on the remaining frames. In experiments, we adopt mean absolute error (MAE) as the criterion to measure the performance. Furthermore, average MAE, the average of multiple MAE over all evaluation videos, is used to reflect the overall performance of people counting methods.

To examine the stability of people counting approaches, we conduct experiments with four different settings, including using one video only (HEAVY), combinations of two videos (SPARSE +HEAVY and MEDIUM+HEAVY), and using all three videos (SPARSE +MEDIUM +HEAVY) as training data. Therefore, we can evaluate the performance sensitivity to the changes of the diversity of training data. Since a few parameters of support vector regression and multiple kernel learning have significant influence on performance of established systems, we thus build our MCPC system with the optimal parameters, selected from a reasonable parameter space via five-fold cross validation. The automatically tuned parameters include the epsilon in loss function and the penalty cost to errors in LIBSVM [11], and the regularization term in SimpleMKL [35].

5.2 Results

We first evaluate the accuracy of our blob correspondence estimation algorithm by manually and visually checking the consistency of the matched components between two views. For the ease of validation, we assign the same color and the same number to both of the matched components in two views, as shown in Figure 8. Then, we calculate the percentage of correct matches over the whole matches. From Table 1, we observe that the proposed algorithm is able to obtain sufficiently accurate results, ranging from 96.4% to 98.5% of component matching precision, over three videos with different crowdedness levels.

To understand what the MCPC system advances single-camera people counting systems, we use the estimation results yielded in the first stage as our baseline (BS-MKL). We have also implemented a Gaussian process method and a neural network method, similar to the state of the art approaches proposed by Chan et al. [9] and Ryan et al. [36], respectively, to count the number of people. By using the same low-level (local) features, we thus establish another two sets of estimation results, i.e., BS-GP and BS-NN. Note that these three approaches merely use intra-camera visual cues and do not involve any knowledge transfer among cameras. In our current survey, the only MCPC system related to ours is a fusion approach that combines human detection results from multiple cameras [31]. As mentioned in section 2, however, this method is not suitable to handle the scenes with dense crowds like the PETS 2010 datasets used in our experiments. To make meaningful comparisons, therefore, we develop two variants of the approach to fuse complementary information. We introduce two fusion manners, frame-based fusion (FRAME-AVG) and component-based fusion (COMPO-AVG), which averagely combine the estimation results of synchronized frames and matched components, respectively. Finally, we refer to our MCPC system as OURS.

Table 2 displays the overall estimation errors (MAE) on SPARSE, MEDIUM, and HEAVY when using our MCPC system (OURS) and comparisons with the BS-MKL, BS-GP, BS-NN, FRAME-AVG, and COMPO-AVG methods. When taking into account only intra-camera visual features, the BS-MKL outperforms the BS-GP approach in most settings based on average MAE. The only two exceptions are on View 1 when using only HEAVY as training data and on View 2 when using all SPARSE, MEDIUM, and HEAVY as training data. This shows the effectiveness of applying multiple kernel learning techniques to people counting when we have a number of features. Furthermore, we observe that the performance gains of the multiple-camera systems using either of average fusion approaches is modest since this kind of approaches definitely performs an interpolation of the estimations given by different views. Although this fashion may yield higher accuracy, in most cases errors are propagated, thus degrading performance. In contrast, our MCPC system is able to handle this difficulty and achieves significant improvement. Overall, OURS reduces the estimation errors of the first pass (BS-MKL) ranging from 15.8% $\left(\frac{1.49-1.77}{1.77}\right)$ to 78.6% $\left(\frac{1.21-5.66}{5.66}\right)$, and the ones of frame-based fusion (FRAME-AVG) ranging from 13.8% $\left(\frac{3.07-3.56}{3.56}\right)$ to 74.8% $\left(\frac{1.00-3.97}{3.97}\right)$, in terms of relative improvement.

Figure 7 shows the number of people frame-by-frame on the three testing videos, including manually annotated groundtruth and estimations yielded by BS-MKL, BS-GP, BS-NN, FRAME-AVG, COMPO-AVG, and OURS, respectively, in two experimental settings. From these figures, we note that the proposed MCPC generally gains significant performance on HEAVY, while the improvement on SPARSE is not obvious in a few settings. There are two possible reasons. First, occlusions are seldom observed in the SPARSE video. Therefore, people counting with a single camera may be sufficient. Second, the appearance variation of small groups of pedestrians is usually modest, whereas highly occluded crowds could represent a large variation. Thus, complementary information and inter-camera knowledge are more useful to handle highly crowded scenes (heavy occlusions) than to handle sparse scenes. To better explain our observation, we show several examples of video frames with the estimations on extracted components in Figure 8. In these examples, one can observe that diverse properties are captured by different cameras.

[1] http://research.twnct.net/MCPC_MM2012/

Table 1: Description of the PETS 2010 datasets used in our experiments.

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	

Table 2: Summary of estimation errors of people counting on three videos with different occlusion levels, when applying the proposed MCPC system (OURS) and comparisons with the first-pass estimation results (BS-MKL), Chan et al.'s approach (BS-GP) [9], Ryan et al.'s approach (BS-NN) [36], and two average fusion methods (FRAME-AVG and COMPO-AVG). Note that the BS-MKL, BS-GP, and BS-NN are three methods which do not involve knowledge transfer, while others exploit complementary information perceived in multiple camera systems. Additionally, these quantitative results are reported in either MAE or average MAE.

training data	testing data	View 1						View 2					
		BS-MKL	BS-GP	BS-NN	FRAME-AVG	COMPO-AVG	OURS	BS-MKL	BS-GP	BS-NN	FRAME-AVG	COMPO-AVG	OURS
HEAVY	SPARSE	5.94	5.71	3.10	4.86	4.77	**1.80**	15.64	18.09	13.42	4.84	5.00	**4.72**
	MEDIUM	5.98	3.40	3.74	3.79	3.30	**2.26**	10.11	11.51	7.42	2.52	3.00	**2.42**
	HEAVY	**1.24**	2.11	1.28	3.33	3.33	1.72	6.50	5.77	5.64	3.33	3.33	**2.06**
	OVERALL	4.39	3.74	2.71	3.99	3.80	**1.93**	10.75	11.79	8.82	3.56	3.77	**3.07**
SPARSE+HEAVY	SPARSE	1.75	3.78	1.74	**1.15**	1.51	1.24	0.97	1.94	0.91	**0.72**	0.82	0.88
	MEDIUM	1.80	1.85	1.73	1.75	1.81	**1.60**	1.63	1.84	1.76	1.59	**1.46**	2.41
	HEAVY	1.84	2.12	1.82	5.02	5.02	**1.08**	8.25	7.36	6.89	5.02	5.02	**2.19**
	OVERALL	1.80	2.59	1.76	2.64	2.78	**1.31**	3.62	3.71	3.19	2.44	2.43	**1.83**
MEDIUM+HEAVY	SPARSE	2.04	5.83	2.13	4.05	4.64	**1.38**	6.13	7.15	6.96	4.03	3.45	**0.73**
	MEDIUM	1.68	3.16	1.73	3.53	2.84	**1.04**	3.99	4.95	4.10	1.55	1.94	**1.02**
	HEAVY	1.82	1.44	1.45	4.33	4.33	**0.58**	6.86	6.88	6.34	4.33	4.33	**1.89**
	OVERALL	1.85	3.48	1.77	3.97	3.93	**1.00**	5.66	6.33	5.80	3.30	3.24	**1.21**
SPARSE+MEDIUM+HEAVY	SPARSE	1.64	3.85	1.75	**1.31**	1.73	1.95	1.58	1.96	1.38	1.11	**0.68**	1.32
	MEDIUM	1.57	1.71	1.57	2.75	2.17	**1.37**	3.24	2.34	2.09	1.20	1.35	**1.02**
	HEAVY	2.10	1.65	1.79	5.71	5.71	**1.16**	9.33	7.82	7.98	5.71	5.71	**1.20**
	OVERALL	1.77	2.40	1.70	3.26	3.20	**1.49**	4.71	4.04	3.82	2.67	2.58	**1.18**

Our approach effectively makes use of these properties and leads to salient improvement.

6 Conclusions

In this paper, we have introduced a multiple camera people counting system based on the spirit of transfer learning to improve the weakness of using a single camera. This work has three main contributions. The first is an exploration of inter-camera knowledge. We developed a two-pass regression framework which has been shown effective in adapting heterogeneous information for people counting. Second, a novel blob matching algorithm was proposed to obtain a set of consistent entities among cameras, thus leading to the success of knowledge sharing. Finally, we released sets of manually annotated pedestrian labels based on the PETS 2010 Benchmark Data, which are considered as a valuable resource for multiple camera people counting research.

Acknowledgments

This work was partially supported by the National Science Council of Taiwan, R.O.C., under grants NSC100-2221-E-001-013-MY3, NSC100-2631-H-001-020, and NSC100-2218-E-001-004.

7 References

[1] N. Ahuja and S. Todorovic. Extracting texels in 2.1D natural textures. In *Proc. of IEEE ICCV*, 2007.

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

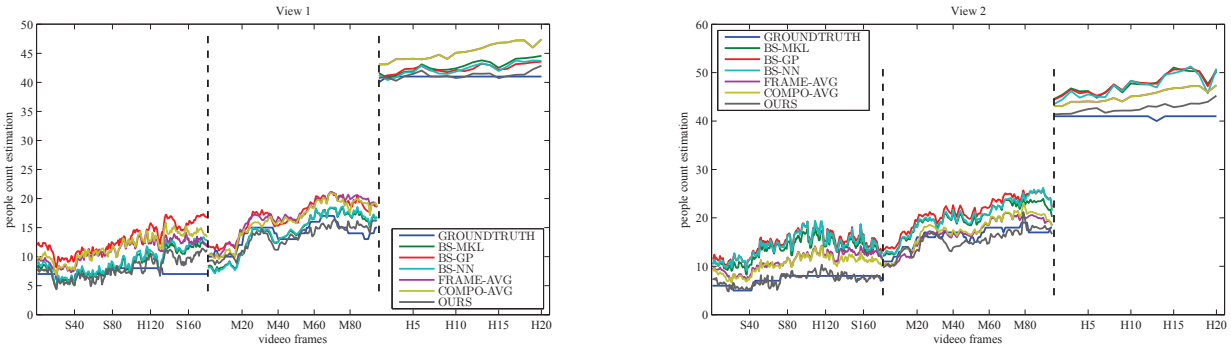
[3] C. Aslan, K. Bernardin, and R. Stiefelhagen. Automatic calibration of camera networks based on local motion features. In *Proc. of M2SFA2*, 2008.

[4] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proc. of ICML*, 2004.

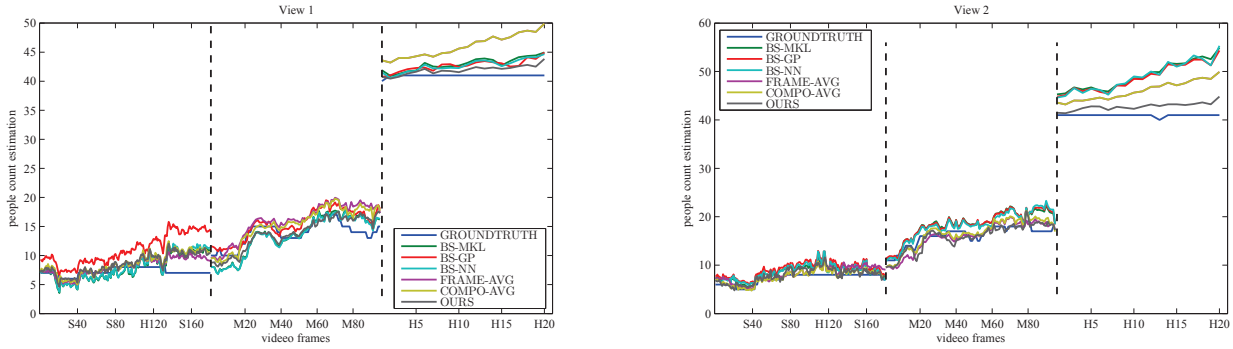
[5] O. Barnich and M. V. Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.*, 20(6):1709–1724, 2011.

[6] E. Bart and S. Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *Proc. of IEEE CVPR*, 2005.

[7] S. Bickel, M. Bruckner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proc. of ICML*, 2007.



(a) Using MEDIUM and HEAVY as training data



(b) Using all SPARSE, MEDIUM, and HEAVY as training data

Figure 7: The number of people of individual frames on the testing data reported with groundtruth, first-pass estimation (BS-MKL), Chan et al.'s approach (BS-GP) [9], Ryan et al.'s approach (BS-NN) [36], two average fusion methods (FRAME-AVG and COMPO-AVG), and our MCPC system (OURS). Note that the BS-MKL, the BS-GP, and the BS-NN are three methods which use only intra-camera visual features while others utilize additional knowledge among collaborative cameras.

[8] G. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *Proc. of IEEE CVPR*, 2006.

[9] A. Chan, Z. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proc. of IEEE CVPR*, 2008.

[10] A. Chan and N. Vasconcelos. Bayesian Poisson regression for crowd counting. In *Proc. of IEEE ICCV*, 2009.

[11] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:1–27, 2011.

[12] N. Dalal and B. Triggs. Histogram of oriented gradient for human detection. In *Proc. of IEEE CVPR*, 2005.

[13] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proc. of IEEE CVPR*, 2009.

[14] L. Duan, I. Tsang, D. Xu, and S. Maybank. Domain transfer SVM for video concept detection. In *Proc. of IEEE CVPR*, 2009.

[15] R. Eshel and Y. Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Proc. of IEEE CVPR*, 2008.

[16] J. Ferryman and A. Ellis. PETS2010: Dataset and challenge. In *Proc. of IEEE AVSS*, 2010.

[17] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, 12:2211–2268, 2011.

[18] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *Proc. of IEEE CVPR*, 2006.

[19] J. Jacques Junior, S. Musse, and C. Jung. Crowd analysis using computer vision techniques. *IEEE Signal Process. Mag.*, 27:66–77, 2010.

[20] 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. of IEEE ICCV*, 2009.

[21] M. Jones and D. Snow. Pedestrian detection using boosted features over many frames. In *Proc. of ICPR*, 2008.

[22] S. M. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(3):505–519, 2009.

[23] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos. Estimating pedestrian counts in groups. *Comput. Vis. Image Underst.*, 110(1):43–59, 2008.

[24] D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. In *Proc. of BMVC*, 2005.

[25] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

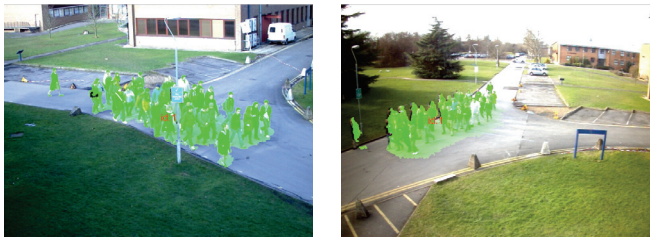
[26] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. of IEEE CVPR*, 2005.

[27] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[28] S.-F. Lin, J.-Y. Chen, and H.-X. Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Trans. Syst., Man, Cybern. A*, 31:645–654, 2001.

[29] T.-Y. Lin, Y.-Y. Lin, M.-F. Weng, Y.-C. Wang, Y.-F. Hsu, and

View 1						View 2					
ID	GT	MKL	GP	AVG	OURS	ID	GT	MKL	GP	AVG	OURS
1	41	44.7	44.9	48.0	42.2	1	41	51.4	50.4	48.0	42.5



(a) An example frame of HEAVY. The results are yielded by those models learned from the combination of SPARSE and HEAVY.

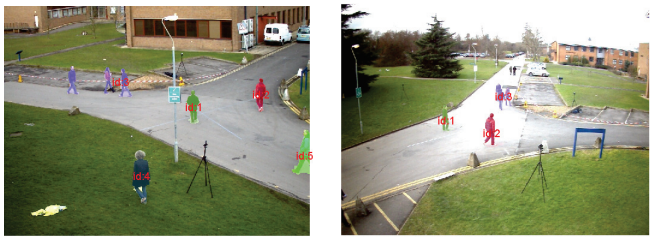
View 1						View 2					
ID	GT	MKL	GP	AVG	OURS	ID	GT	MKL	GP	AVG	OURS
1	9	10.2	9.8	12.7	10.9	1	12	15.7	16.2	12.7	11.9
2	4	5.2	5.8	5.6	4.9	2	4	5.4	5.1	5.6	5.1
—	—	—	—	—	—	3	1	1.1	1.3	1.1	1.1

(c) An example frame of MEDIUM. The results are yielded by those models learned from all three SPARSE, MEDIUM, and HEAVY.

View 1						View 2					
ID	GT	MKL	GP	AVG	OURS	ID	GT	MKL	GP	AVG	OURS
1	7	9.2	10.8	12.3	6.8	1	7	15.4	16.4	12.3	8.2
2	1	1.0	1.6	1.4	1.0	2	1	1.9	2.0	1.4	1.3

(b) An example frame of SPARSE. The results are yielded by those models learned from the combination of MEDIUM and HEAVY.

View 1						View 2					
ID	GT	MKL	GP	AVG	OURS	ID	GT	MKL	GP	AVG	OURS
1	1	0.7	1.0	0.8	1.0	1	1	0.9	1.0	0.8	1.1
2	1	1.1	1.5	1.2	1.4	2	1	1.3	1.4	1.2	1.4
3	3	2.3	3.7	2.8	3.4	3	3	3.3	3.5	2.8	3.7
4	1	0.9	1.4	0.9	0.9	—	—	—	—	—	—
5	1	0.7	1.3	0.7	0.7	—	—	—	—	—	—



(d) An example frame of SPARSE. The results are yielded by those models learned from HEAVY.

Figure 8: Several sample frames selected from the videos used in our experiments. The quantitative results reported on components demonstrate the effectiveness of our MCPC system.

H.-Y. M. Liao. Cross camera people counting with perspective estimation and occlusion handling. In *Proc. of IEEE WIFS*, 2011.

[30] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[31] H. Ma, C. Zeng, and C. X. Ling. A reliable people counting system via multiple cameras. *ACM Trans. Intell. Syst. Technol.*, 3(2):1–22, 2012.

[32] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proc. of IEEE CVPR*, 2008.

[33] S. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[34] V. Rabaud and S. Belongie. Counting crowd moving objects. In *Proc. of IEEE CVPR*, 2006.

[35] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *J. Mach. Learn. Res.*, 9:2491–2521, 2008.

[36] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Crowd counting using multiple local features. In *Proc. of DICTA*, 2009.

[37] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *J. Mach. Learn. Res.*, 7:1531–1565, 2006.

[38] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robot. Autom.*, 3(4):323–344, 1987.

[39] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[40] M.-F. Weng and Y.-Y. Chuang. Cross-domain multi-cue fusion for concept-based video indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 99(PrePrints), 2011.

[41] B. Wu and R. Nevatia. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *Int. J. Comput. Vision*, 82(2):185–204, 2009.

[42] Y. Xiong and F. Quek. Meeting room configuration and multiple camera calibration in meeting analysis. In *Proc. of ICMI*, 2005.

[43] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *Proc. of ACM MM*, 2007.

[44] B. Zhan, D. Monekosso, P. Remagnino, S. Velastin, and L.-Q. Xu. Crowd analysis: A survey. *Mach. Vision Appl.*, 19:345–357, 2008.