

Pedestrian Verification for Multi-Camera Detection

Scott Spurlock and Richard Souvenir

University of North Carolina at Charlotte
Charlotte, NC 28223 USA
{sspurloc,souvenir}@uncc.edu

Abstract. In this paper, we introduce an approach to multi-camera, multi-object detection that builds on low-level object localization with the targeted use of high-level pedestrian detectors. Low-level detectors often identify a small number of candidate locations, but suffer from false positives. We introduce a method of *pedestrian verification*, which takes advantage of geometric and scene information to (1) drastically reduce the search space in both the spatial and scale domains, and (2) select the camera(s) with the highest likelihood of providing accurate high-level detection. The proposed framework is modular and can incorporate a variety of existing detection methods. Compared to recent methods on a benchmark dataset, our method improves detection performance by 2.4%, while processing more than twice as fast.

1 Introduction

Detection and tracking of multiple people from video has many important applications, including automated surveillance, crowd modeling, and sports analysis. As the number of people in the scene increases, occlusions become a major challenge. Compared with single-camera approaches (e.g., [1, 2]), by making use of multiple, overlapping cameras, several recent methods [3–6] have shown robustness to occlusion in these types of crowded scenes with low-level detectors that measure 3D occupancy. Typically these approaches require a trade-off between speed and accuracy. Of recently developed approaches to multi-camera, multi-object (MCMO) detection, the most accurate involve expensive computation not suited to real-time application. The fastest methods tend to be less accurate, providing only probability maps and delaying final localization to a subsequent tracking phase.

In parallel, recent methods for pedestrian detection have shown promising results for identifying individual people in images. At low resolutions and in the presence of occlusion, however, even the best detectors perform poorly. Further, while detector speed has improved significantly in recent years, these methods are not designed to be used for multi-camera person detection in real-time at typical resolutions using the common approach of sliding windows at multiple scales and locations.

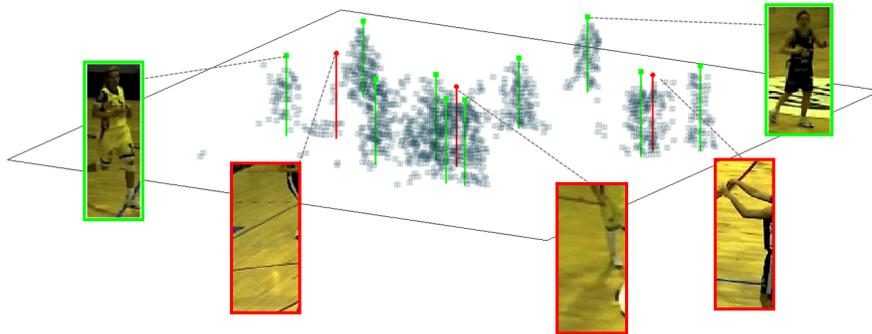


Fig. 1. MCMO detectors based on low-level features are prone to “ghosts” (red lines), or false positives, caused by shadows, occlusions, and projective effects among the true positive detections (green lines). Our method incorporates high-level image-based features from the camera(s) with the best view to verify actual people.

In this paper, we propose a hybrid approach that uses fast low-level detection and targeted high-level verification, achieving high accuracy at real-time speed. Our framework is modular, consisting of low, medium, and high-level detection steps. The modularity of the design allows our framework to incorporate new or pre-existing detector implementations as needed. With each successive step more computationally expensive than the previous, the goal is to discard as many hypotheses as possible using computationally inexpensive methods, and only use high-level detectors to *verify* uncertain earlier hypotheses. Figure 1 illustrates the idea. A low-level occupancy detector identifies 3D foreground voxels, shown as gray cuboids. A mid-level aggregation step localizes objects, finding both true detections (green lines) as well as false positives (red lines), known as ghosts. For high-level pedestrian verification, image patches are extracted corresponding to locations to be verified. The goal is for a pedestrian detector to accurately evaluate the presence of a person in the image patch. However, in a multi-camera environment, certain viewpoints may be preferable to others, in terms of the expected accuracy of the detector.

Our main contribution is a multi-stage, coarse-to-fine framework for MCMO detection, which includes a probabilistic model for selecting the optimal camera(s) with respect to expected detection accuracy. The targeted use of high-level verification keeps computational cost low while keeping accuracy high. We evaluate our method on a challenging benchmark dataset for MCMO detection and tracking. Our results show the efficacy of our real-time approach, outperforming recent methods in both detection accuracy and computational efficiency. Note that while our focus in this paper is on MCMO detection, the method can easily be incorporated into any end-to-end tracking system, directly benefiting tracking performance.

2 Related Work

Detecting people from images and video has been well-covered over many years [7]. Our focus is on multi-camera methods that incorporate low-level features for occupancy estimation.

Multi-Camera, Multi-Object Detection Most MCMO methods start with background subtraction (e.g., [8]) and then fuse extracted foreground silhouettes to a common 3D coordinate system or ground plane. For example, Khan and Shah [9] use homographies to warp foreground probability maps to a common reference plane and detect feet locations, while Eshel and Moses [10] detect head tops by incorporating intensity correlation in a similar homography-based framework. Fleuret et al. [5] introduce a probabilistic framework to model occupancy over a ground plane grid. Several methods [4, 11, 12] employ a 3D reconstruction approach, where occupancy is calculated over a discrete 3D grid of voxels, instead of just the 2D ground plane. These methods may detect people in the 3D space [12] or project the volumetric reconstruction to the ground plane [4, 11]. Typically, exact localization is delayed to a later tracking phase based on, e.g., graph cuts [9] or dynamic programming [5] over temporal windows.

Reducing False Positive Detections Some recent MCMO detection methods have explicitly incorporated schemes to address ghosts. Alahi et al. [3] model ground plane occupancy estimation as a sparse optimization problem. A sparsity constraint is intended to rule out false positives during the detection phase. While this method achieves high detection accuracy, the authors' implementation takes 10 seconds per frame, making it unsuitable for real-time applications. Peng et al. [6] incorporate a graphical model that explicitly encodes occlusion relationships among discretized ground-plane locations. An iterative algorithm finds the occupancy configuration that best explains the camera foreground images. The method reduces the occurrence of ghost detections due to the occlusion reasoning, but takes 3 seconds per frame in the authors' implementation. Other methods incorporate simple rules to reduce ghosts, such as fixing a priori the number of objects to be detected [13].

Our framework, which includes concepts common to MCMO methods, incorporates pedestrian verification directly into the detection stage rather than a subsequent tracking step or with ad hoc rules. The verification step relies on selecting the best viewpoints for image-based pedestrian detection. However, compared to the sliding window approach commonly employed for single image pedestrian detectors, our method drastically reduces the search space by only evaluating selected image patches. Viewed in this light, the low-level detection step provides geometric context similar to approaches (e.g., [14]) that use scene context to reduce false positives. By combining efficient low-level detection, mid-level aggregation, and targeted use of high-level verification, our framework is capable of real-time multi-person detection in multi-camera networks.

3 Base Detector

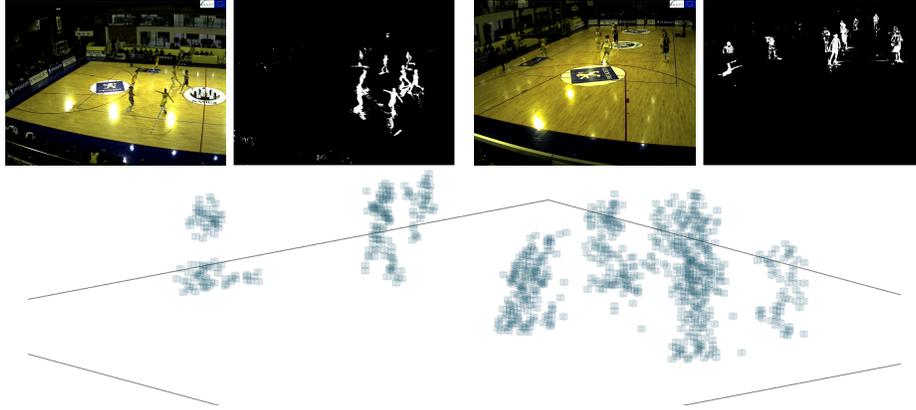


Fig. 2. Example input frames and extracted foreground silhouettes used to perform a coarse 3D reconstruction for our low-level detector.

Our pedestrian verification approach could be used with any low- or mid-level MCMO detector. In this section, we describe our base detector implementation.

3.1 Low-level Detection

As shown in Figure 2, our low-level detector performs change detection on C cameras viewing the scene in order to create a coarse 3D reconstruction of the visual hulls of moving objects. The scene volume is discretized into a voxel grid, $V = \{v_1, v_2, \dots, v_n\}$, where each voxel is identified as either background or foreground by a straightforward voting scheme:

$$v_i = \begin{cases} 1 & \text{if } \sum_k \pi(v_i, k) \geq \gamma \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\pi(v_i, k)$ indicates whether voxel v_i projects to foreground in camera k , and γ is the threshold for the number of cameras in the network that must agree for a positive voxel detection. To implement the voxel-image occupancy function, π , other MCMO detectors (e.g., [4]) employ *point sampling*, where the voxel center is projected to a single pixel in an image. For greater robustness to noisy foreground extraction, we employ *area sampling*, where the 3D extent of the voxel is projected to a bounding box in an image. Then, we define the voxel-image occupancy function as:

$$\pi(v_i, k) = \begin{cases} 1 & \text{if } \rho(v_i, k) \geq \beta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where ρ represents the proportion of pixels in the associated bounding box in image k corresponding to foreground and β is a system-specific threshold that can be tuned based on the noise level of the foreground segmentation process. The voxel-image occupancy function with area sampling can be implemented efficiently using the integral image technique [15] with the foreground mask image.

3.2 Mid-level Aggregation

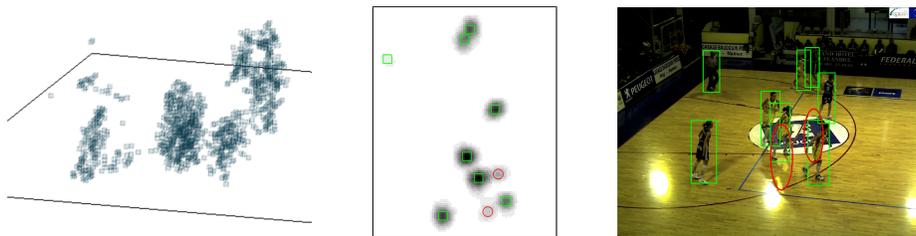


Fig. 3. Given foreground voxels (left), mean shift clustering (middle) localizes objects. For the identified cluster centers, green squares are true positives and red circles are false positives (ghosts). Note that the two ghosts are more pronounced than the correct detection of the person at the top-left. (Right) An image from a camera in the network shows the projected detections.

The next stage is aggregation of voxel detections to objects, illustrated in Figure 3. Our approach relies on mean shift clustering (MSC) [16] for this step. MSC is a non-parametric clustering approach that can find non-uniform or narrow modes in a distribution, which, in our case, correspond to potential object locations in the scene. MSC is well-suited to the problem because no prior knowledge about the number or location of objects is needed.

Let $\{\mathbf{x}_i\}$ be the set of points in \mathbb{R}^3 corresponding to the centers of the identified foreground voxels. We define the kernel density estimator [16] for occupancy at a point \mathbf{x} as

$$\hat{f}(\mathbf{x}) \propto \sum_i K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) \quad (3)$$

where $K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x})$. Here \mathbf{H} is a $d \times d$ bandwidth matrix and K is the unit flat kernel [17]

$$K(x) = \begin{cases} 1 & \text{if } \|x\|_{\infty} \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\|\cdot\|_{\infty}$ is the infinity norm, which implies an axis-aligned, box-shaped kernel with dimensions controlled by bandwidth matrix, \mathbf{H} , a diagonal matrix, where each element along the diagonal is the squared bandwidth for a dimension of

the box. For person detection, we choose \mathbf{H} to approximate the dimension of an upright person, i.e., $h_1 = h_2 = h_3/4$. While MSC implementations typically incorporate the smoothly differentiable Epanchnikov or Gaussian kernels, our choice of an axis-aligned box kernel allows for faster computation and works well in practice.

Each cluster is scored based on the proportion of foreground voxels within the bandwidth to total bandwidth volume

$$s_d(\delta_m) = \frac{\hat{f}(\delta_m)}{\prod_{j=1}^d h_j} \quad (5)$$

where δ_m is the d -dimensional cluster mean (for our application, $d = 3$). The cluster score can be thresholded to discard low-scoring detections, which often correspond to ghosts. However, care must be taken to avoid rejecting valid detections. Figure 3 shows an example where a valid detection scores lower than two ghost detections. In the next section, we describe how pedestrian detection can help distinguish between correct and incorrect detections.

4 Pedestrian Verification

In some systems [3, 5, 6], the output from low- and/or mid-level stages are directly used as output detections. However, some of these may actually be “ghosts,” or false positive detections due to shadows, reflections, or occlusions. These errors become increasingly common as crowd density increases, and, in complex scenes, significantly degrade overall system accuracy. Figure 3 shows two examples of ghost detections in red. Our high-level detection stage, pedestrian verification, is aimed at identifying and eliminating these false detections without filtering out correct detections.

4.1 Predicting Verification Accuracy

For a given cluster, represented by center location, δ_m , the 3D bounded region corresponds to an image patch in each camera. For each candidate patch, we compute the *Expected Detection Accuracy (EDA)*, $E[Q|\Theta]$, where Q is a continuous random variable representing accuracy of a pedestrian detector under the conditions encoded by the vector, Θ . Ideally, the model attributes would be features that are efficient to compute following the low-level detection phase. A recent survey [18] provides an evaluation of the performance of numerous detectors as a function of occlusion and scale. The best performing detectors work well for near-scale (at least 80 pixels high) examples, with rapid performance decrease as pedestrian size decreases. Additionally, all of the detectors were sensitive to occlusion; even partial occlusion (<35%) led to a log-average miss rate of 73% for the best detector. To estimate the predictive power of a pedestrian detector from a given viewpoint, our model incorporates occlusion, scale, and also verticality, a measure of how upright a person appears from a particular viewpoint. For a candidate location and corresponding image patches, these three features can be computed using the projection of the 3D bounding boxes.

4.2 Model Attributes

We define the bounding box for detection m projected into camera k as the (rectangular) area of pixels, r_m^k . For the candidate detection, the up vector, U_m^k , is the projection of a 3D vector pointing up along the positive Z-axis from the candidate ground location, m , to the target’s estimated height, as viewed in camera k .

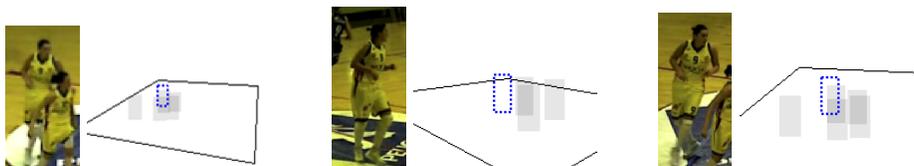


Fig. 4. The occlusion value is estimated by calculating the overlap of the candidate bounding box (dashed blue rectangle) with other, closer detections (gray boxes). For the views shown, occlusion is 0.82, 0.00, and 0.56, respectively. (Lower is better.)

Occlusion In order to estimate an occlusion ratio for each detection based on the other (potential) detections in the scene, we adapt the painter’s algorithm [19] from computer graphics. The idea is to order the detections by proximity to the camera center, and project a synthetic bounding box, r_j^k , into a 2D accumulator for each detection that is closer to the camera than r_m^k . The occlusion ratio measures the overlap of other (potential) detections with the candidate location:

$$s_o(m, k) = \frac{\left| \left(\bigcup_j r_j^k \right) \cap r_m^k \right|}{|r_m^k|} \quad (6)$$

where $|\cdot|$ is the number of pixels in the box. Figure 4 shows an example of how occlusion is calculated for three different views of one example detection from the scene depicted in Figure 3.

Verticality Typically, pedestrian detectors are trained on examples containing mostly upright (vertical) people. So, rather than incur the cost of training many detectors or applying a warp to each image patch, we estimate how upright a person at a given 3D location will appear from a particular view. Verticality is computed as:

$$s_u(m, k) = \left\langle \frac{U_m^k}{\|U_m^k\|}, \frac{I_m^k}{\|I_m^k\|} \right\rangle \quad (7)$$

where I_m^k is a vector pointing in the up direction (along the positive Y axis) in the image and $\langle \cdot, \cdot \rangle$ indicates the inner product.

Height One of the features most correlated with pedestrian detection accuracy is the pixel height of the pedestrian [18]. The height, in pixels, of a projected object is simply the magnitude of the projected up vector, $s_h(m, k) = \|U_m^k\|$.

4.3 Model

Given a set of training examples, we compute, for each attribute, the expected accuracy (true positive, true negative) using a binary logistic regression model. That is, we compute $E[Q | s_x]$ for each attribute. To model the joint expectation for a given image patch, we make the Naive Bayes assumption of conditional independence between the features. This gives:

$$E[Q | \Theta] \propto E[Q | s_o] \cdot E[Q | s_u] \cdot E[Q | s_h] \quad (8)$$

Figure 5 shows some examples of the expected detection accuracy evaluated for selected patches. In the next section, we show how this value can be used to compare multiple image patches of the same object detection to select the best view(s) for pedestrian verification in a real-time MCMO detection framework.



Fig. 5. The estimated detection accuracy (Equation 8) for selected image patches. The first patch depicts an ideal case (unoccluded, upright, and near-field). The remaining patches show examples of slight occlusion, smaller height, moderate occlusion, and non-verticality, respectively.

5 Results

We evaluated our method on the APIDIS dataset¹, which contains footage from a basketball game captured by 7 calibrated, pseudo-synchronized cameras. The dataset contains people of similar appearance and heavy occlusions, as well as shadows and reflections on the court. In order to compare results with other

¹ <http://www.apidis.org/>

recent work [3–6], we followed the most common protocol of measuring performance within the bounds of the left side of the basketball court, which is covered by the most cameras. For quantitative evaluation, we used precision and recall, where a true positive is a detection whose estimated location projects onto the ground plane is within a person-width of the ground truth, a false positive is a detection unmatched to an actual person, and a false negative is a missed detection.

5.1 Implementation Details

We set the minimum number of cameras for voxel occupancy voting, γ , to 3, and the foreground ratio threshold, β , to 0.25. For mean shift clustering, the bandwidth was $45 \times 45 \times 180$ cm. In the 3D occupancy grid, each voxel covered 10 cm^3 . For change detection, our method uses a GPU implementation of adaptive background subtraction [20].

5.2 Pedestrian Detector Evaluation

Our method supports most image-based pedestrian detectors. We evaluated four commonly-used, pre-trained detectors: HOG [21], VJ, based on the Viola-Jones cascade classifier [15], and the Dollár et al. [22] detector, trained with the INRIA dataset [21] (DOLLAR-INRIA) and the CalTech dataset [23] (DOLLAR-CALTECH). The Viola-Jones detector is a cascade classifier trained specifically on upper body examples [24], while the others are trained to identify full-body pedestrians. Each of these pedestrian detectors provides a detection score, and a threshold is commonly applied to obtain the final result. For a set of image patches containing both positive (people) and negative (background) examples, we computed the ROC curve across a range of thresholds for each detector and used the Area Under the Curve (AUC) measure as a basis for comparison. HOG, VJ, DOLLAR-INRIA, and DOLLAR-CALTECH achieved 0.65, 0.56, 0.60, and 0.67, respectively. Overall, DOLLAR-CALTECH performed the best, and, unless otherwise specified, is the implementation we employed for subsequent experiments. These values are much higher than would be expected from the typical approach of pedestrian detection of sliding windows across multiple image scales and locations. Beyond the efficiency concerns, this approach leads to many false positives and false negatives. However, with a fixed location and scale (i.e., an image patch corresponding to a particular 3D location), such detectors can be quite accurate. This phenomenon was noted in a recent survey [18], which found that classifier performance on image patches is only weakly correlated with detection performance on full images.

5.3 Pedestrian Verification

To evaluate the effect of pedestrian verification on MCMO detection, we implemented the base detector, and performed experiments applying pedestrian

verification from multiple cameras. We tested two schemes: (1) using the top- k cameras, and (2) selecting a variable number of cameras based on predicted accuracy. To combine the results from multiple cameras, the k detector scores are averaged, weighted by the expectation (Equation 8), prior to thresholding. In the variable-camera scheme, all cameras with an EDA above .9 are included in the ensemble.

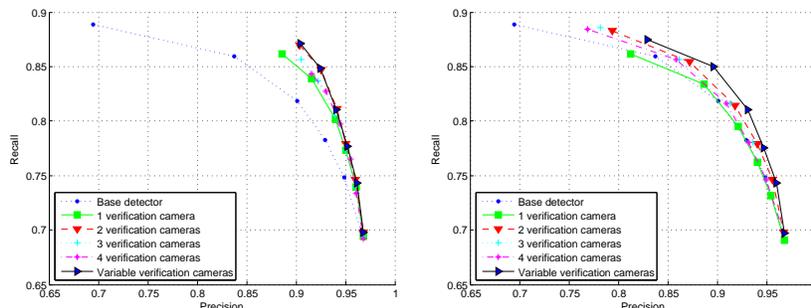


Fig. 6. Precision-recall curves for base detection with pedestrian verification with DOLLAR-CALTECH (left) and HOG (right) using both fixed and variable number of camera schemes.

Figure 6 shows precision-recall curves for various verification schemes on the APIDIS dataset for two pedestrian detectors (HOG and DOLLAR-CALTECH). While increasing from $k = 0$ to $k = 2$ cameras improves the overall performance, adding a third or fourth camera does not. This result suggests that, for this particular dataset, there are many instances where two of the available cameras provide complementary suitable views of a particular location, but additional viewpoints are neither helpful and perhaps contradictory. Overall, using a variable number of cameras for each location performed best, although the effect is more pronounced with the HOG detector than with DOLLAR-CALTECH. On average, the variable scheme resulted in 2.56 image patches evaluated for each candidate location.

5.4 Comparison with Other MCMO Methods

Table 1 compares the results of our method with several recently published approaches on the APIDIS dataset. For each method, the precision, recall, F-score, and frames-per-second (FPS) are shown. Excluding our method, the speed-accuracy trade-off is evident across the related approaches. To the best of our knowledge, our method using $k = 2$ verification cameras (base detector + verification) outperforms all other reported detection results on the APIDIS dataset, while performing at real-time speeds. Note that our detection method outperforms approaches that also incorporate tracking. Figure 7 shows some examples from this experiment.



Fig. 7. The top two frames show examples of our correctly identifying the presence of multiple people (green rectangles). The bottom two frames show challenging cases where the selected pedestrian detector failed for a given patch (red oval).

Method	Precision	Recall	F-Score	FPS
Base detector	0.84	0.86	0.85	13.13
Base detector + verification	0.93	0.85	0.89	10.40
Alahi [3]	0.92	0.82	0.87	0.1
Peng [6]	0.90	0.84	0.87	0.33
Posseger [4] (with tracking)	0.88	0.79	0.83	4.42
POM+KSP [5, 25] (with tracking)	0.80	0.73	0.76	0.03
POM [5]	0.51	0.63	0.56	80.70

Table 1. Comparison of our method and several recent approaches using precision and recall rate on the APIDIS dataset. POM+KSP results are taken from [4] and POM results are from [3].

Precision, recall, and framerate (FPS) numbers for the other methods are taken from results reported in the respective papers. Timing numbers, in particular, may not be directly comparable due to differences in hardware and other implementation details. Our method was implemented in C++ with OpenCV and deployed on a 2.5 GHz PC with 8 GB RAM and a Tesla C2075 GPU. For the base detector with verification, processing time is roughly 73%, 6%, and 21% for low-, medium-, and high-level detection, respectively.

6 Conclusions

We presented a framework for multi-camera, multi-object detection. Our multi-stage approach incorporates fast low-level detection and more accurate high-level pedestrian detection to verify uncertain hypotheses. The method is agnostic to any specific implementation of the base detector or verification method. This hybrid approach was shown to be effective in experiments on a challenging dataset, achieving state-of-the-art performance at real-time speeds. For the future, we plan to investigate a cost-sensitive scheme to choose which techniques to deploy in which situations to allow the speed-accuracy tradeoff to be explicitly controlled, depending on the requirements of the system.

References

1. Nakajima, C., Pontil, M., Heisele, B., Poggio, T.: Full-body person recognition system. *Pattern recognition* **36** (2003) 1997–2006
2. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **30** (2008) 1198–1211
3. Alahi, A., Jacques, L., Boursier, Y., Vandergheynst, P.: Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision* **41** (2011) 39–58
4. Possegger, H., Sternig, S., Mauthner, T., Roth, P.M., Bischof, H.: Robust real-time tracking of multiple objects by volumetric mass densities. In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE* (2013) 2395–2402
5. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30** (2008) 267–282
6. Peng, P., Tian, Y., Wang, Y., Huang, T.: Multi-camera pedestrian detection with multi-view bayesian network model. In: *BMVC.* (2012) 1–12
7. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. *ACM Comput. Surv.* **38** (2006)
8. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition. Volume 2.* (1999)
9. Khan, S., Shah, M.: Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Trans. on Pattern Analysis and Machine Intelligence* (2008) 505–519

10. Eshel, R., Moses, Y.: Homography based multiple camera detection and tracking of people in a dense crowd. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2008) 1–8
11. Liem, M., Gavrilu, D.M.: Multi-person tracking with overlapping cameras in complex, dynamic environments. In: *British Machine Vision Conference (BMVC)*, British Machine Vision Association (2009) 199–218
12. Canton-Ferrer, C., Casas, J.R., Pardàs, M., Monte, E.: Multi-camera multi-object voxel-based monte carlo 3d tracking strategies. *EURASIP Journal on Advances in Signal Processing* **2011** (2011) 1–15
13. Delannay, D., Danhier, N., De Vleeschouwer, C.: Detection and recognition of sports(wo)men from multiple views. In: *International Conference on Distributed Smart Cameras*. (2009) 1 – 7
14. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. *International Journal of Computer Vision* **80** (2008) 3–15
15. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Volume 1., IEEE (2001) 1–511
16. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **24** (2002) 603–619
17. Cheng, Y.: Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **17** (1995) 790–799
18. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34** (2012) 743–761
19. Hearn, D., Baker, P., Carithers, W.: *Computer Graphics With OpenGL*. Prentice Hall (2011)
20. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters* **27** (2006) 773–780
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Volume 1., IEEE (2005) 886–893
22. Dollár, P., Appel, R., Kienzle, W.: Crosstalk cascades for frame-rate pedestrian detection. In: *Computer Vision–ECCV 2012*. Springer (2012) 645–659
23. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE (2009) 304–311
24. Kruppa, H., Castrillon-Santana, M., Schiele, B.: Fast and robust face finding via local context. In: *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*. (2003) 157–164
25. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33** (2011) 1806–1819