Privacy-Preserving Crowd Incident Detection: A Holistic Experimental Approach

Emmanuel Baccelli, Alexandra Danilkina, Sebastian Müller, Agnès Voisard, Matthias Wählisch

Privacy-Preserving Crowd Incident Detection: A Holistic Experimental Approach

Emmanuel Baccelli
INRIA Saclay
Île-de-France
91120 Palaiseau (CEDEX),
France
Emmanuel.Baccelli@inria.fr

Alexandra Danilkina
Freie Universität Berlin
Takustr. 9
Berlin, Germany
alexandra.danilkina@fu-berlin.de

Sebastian Müller*
Freie Universität Berlin
Takustr. 9
Berlin, Germany
sebastian.mueller@fu-berlin.de

Agnès Voisard†
Freie Universität Berlin and
Fraunhofer FOKUS
Takustr. 9
Berlin, Germany
agnes.voisard@fu-berlin.de

Matthias Wählisch
Freie Universität Berlin
Takustr. 9
Berlin, Germany
matthias.waehlisch@fu-berlin.de

ABSTRACT

Detecting dangerous situations is crucial for emergency management. Surveillance systems detect dangerous situations by analyzing crowd dynamics. This paper presents a holistic video-based approach for privacy-preserving crowd density estimation. Our experimental approach leverages distributed, on-board pre-processing, allowing privacy as well as the use of low-power, low-throughput wireless communications to interconnect cameras. We developed a multi-camera grid-based people counting algorithm which provides the density per cell for an overall view on the monitored area. This view comes from a merger of infrared and Kinect camera data. We describe our approach using a layered model for data aggregation and abstraction together with a workflow model for the involved software components, focusing on their functionality. The power of our approach is illustrated through the real-world experiment that we carried out at the Schönefeld airport in the city of Berlin.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial Databases and GIS*

General Terms

Emergency Management, Incident Detection, Algorithms, Low-Cost, Experiment

*corresponding author
†corresponding author

1. INTRODUCTION

Critical infrastructures where large number of people come together (e.g., check-in halls, boarding counters at airports, or train stations) can benefit from crowd monitoring systems to prevent dangerous situations, hence support emergency management. In such an environment, there is a continuous flow of people with no clear movement direction, and when crowd density becomes too high, experience shows that dangerous situations become more likely. Therefore, it is essential to be able to estimate and analyze in real-time crowd size and density in key areas of the critical infrastructure to be secured.

In this context, over the recent years, different types of surveillance systems have emerged. Video-based techniques, especially, are wide spread. Available systems propose various functionalities and various levels of complexity, from simple CCTVs to high performance object tracking. In this paper, we focus on large area, indoor multi-camera systems for crowd monitoring providing aggregated crowd information to a central control entity, which can then carry out further evaluations, aiming to enable online complex event detection.

The contributions of this paper are the following:

- we develop a privacy-by-design, distributed architecture and workflow for video-based crowd monitoring and analysis in critical public spaces
- we develop specific online video-processing techniques for various types of cameras including infrared and Kinect[8]
- we validate the architecture and video-processing techniques with experiments in public spaces, including live experiments in an airport

Note that the approach that we propose in this paper is flexible in that it does not depend on the number of cameras in the set-up, and can deal with a large number of sources. Moreover, the approach proposed in this paper can be qualified as online, since the video streams are processed on the fly, and the resulting crowd analysis information is immediately available at the central control – excluding information transmission delay, which is low since the information to transmit is light-weight, preprocessed data.

1.1 Deployment Model & Challenges

We consider a deployment scenario as depicted in Figure 1, where an arbitrary number of cameras look down, vertically, over the areas to monitor. This is a valid assumption, since we target indoor deployments. We assume that cameras may be placed at different heights, may have different view angles, or may be rotated w.r.t. the horizontal plane (0°, 45°, 90°, 270°).

We assume the use of infrared cameras or other specific cameras such as Kinect which make personal identification difficult or impossible. Furthermore, since the cameras are placed looking vertically down on the scene, people appear in the scene only as "blobs", which can be approximated with circles on the fly, locally on the camera itself, so in effect no personal data leaves camera nodes.

We assume the areas to be monitored to be dividable along a virtual global grid of granularity one square meter, as shown in Figure 1. As we will see, this grid-based model allows to encode both spatial and temporal crowd monitoring data in a simple way.

One challenge is not only to find "blobs" that correspond to a crowd (as opposed to, for instance, some other elements in the background), but also to identify the individuals in a "blob" or in a set of "blobs". Indeed, a "blob" could correspond to groups of people or parts of a single person. In environments such as as boarding control, people behavior may be hectic, pushing each other, or stand very close to each other, falsely appearing as a single "blob", seen from above.

Another challenge comes from the multi-camera aspect: data from different cameras should be handled appropriately, especially in the cases where several cameras monitor areas that overlap. The goal is to compute a coherent global view, even faced with potentially inconsistent or duplicate inputs from different cameras.

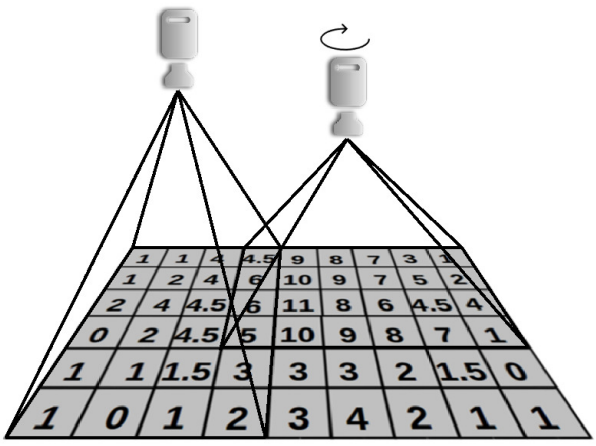

Figure 1: Deployment of the Multi-Camera Solution.

1.2 Related Work

Incident detection as part of emergency management is concerned with many issues. We focus here on video-based surveillance systems. In contrast to many video-based surveillance systems such as [15], [11] and [9], the crowd monitoring scenario that we consider contains cameras, which look strictly down on the monitored area. Consequently, no people occluding or covering each other appear in the scene, so that typical challenges such as perspective estimation, transformation or normalization are not the main focus of the solution presented in this paper. Nevertheless, splitting people silhouettes staying close to each other people is a challenging task.

As presented in [13], object re-identification in one camera view as well as through multiple cameras is often required. Works of [6] and [10] identify people in different camera observations for correct counting. We perform privacy-preserving counting and we re-identify high-level objects – the grids. Our solution does not primarily focus on people tracking, but on counting. We do not focus on motion features of the scene as in [2] or histogram of oriented gradients (HOG) as in [10] and [16]. We analyse features, which result from infrared camera properties directly – pixel intensities, which encode the warmth of monitored objects. The aim of the work described in [12] is to track people, nevertheless the approach of Probabilistic Occupancy Maps is related to the idea of density approximation which we apply. Both estimate the area which is covered by people for further analysis.

The work of [14] makes use of the additional depth sensor in Kinect for Xbox 360 cameras [7] for people detection and further tracking. However, in contrast to our approach, these studies focus on privacy-preserving approach on the contours of humans though.

This paper is structured as follows. In Section 2 we first introduce the architecture of the developed solution. This is followed by a description of the data aggregation layers and the functionality of the corresponding software components. Section 3 presents real-world and constructed experiments as well as the evaluation of counting results. Section 4 concludes the paper.

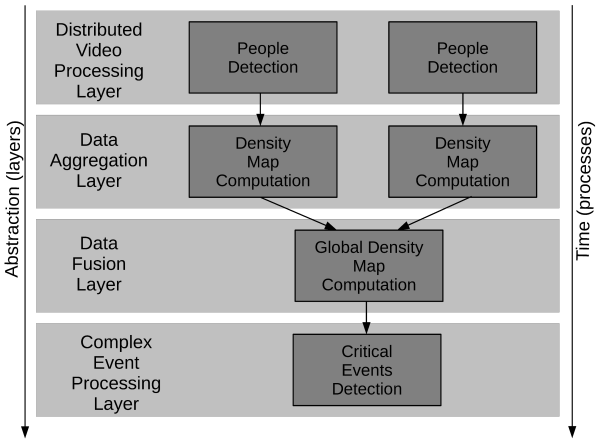

Figure 2: Overview of the Crowd Monitoring Solution.

2. ARCHITECTURE

In this section we present the overall solution for distributed privacy-preserving people monitoring system. Figure 2 gathers two main principles of the developed solution: the layered approach and the workflow approach. Layers represent various levels of data abstraction, from raw sensor data on the low abstraction layer to the complex data (on the top layer) which will eventually be processed. The four layers are represented top down in Figure 2. Workflows represent the temporal order for the execution of software components in parallel or in series.

We developed four abstraction layers from "Distributed On-Board Pre-processing" over "Data Aggregation" to the "Data Fusion", producing input for the "Complex Event Processing" layer. Each of the processing layers may contain several software components from the workflow. The software components are represented with dark gray rectangles.

On the lowest abstraction layer, the "Distributed Video Processing" layer, people should be detected. As introduced in Section 1.1, the view on the monitored scene is modeled in a grid-based manner by dividing the scene along a virtual grid. We represent a density by the number of people per grid cell as a Density Map (DM). Each camera view is modeled by a single DM, the overall view on the scene (further on the Global Spatial View) is modeled by a global DM. Single DM as an abstraction of detected people are computed on the "Data Aggregation" layer and are then processed to global DMs on the "Data Fusion" layer. Global DMs are sent to the top "Complex Event Processing" layer, where critical events could be detected.

In order to obtain the final result i. e., a global DM from distributed infrared images, each step of the workflow has to be executed at least once. Furthermore, software components can be executed in parallel for each abstraction layer, depending on the system set-up. The overall monitoring structure always contains four abstraction layers. Having n cameras in the set-up, the "Distributed Video Processing" layer will contain n "People Detection" components, the "Aggregation" layer n "Density Map Computation" components and the "Data Fusion" layer one "Global Density Computation" component. The "Complex Event Processing" layer will contain at least one module for handling critical events.

Figure 3: Raw infrared image (left), detected people (right).

The low level, "Distributed Video Processing" layer, is executed directly on the camera nodes, while the "Data Fusion" layer and "Complex Event Processing" layer are executed on centralized nodes. However, the layer for data aggregation may be implemented, either on distributed or on centralized nodes depending on available computational power. A shift of the "Data Aggregation" layer from distributed to centralized nodes has no impact on privacy: Raw sensitive video data is pre-processed on the low abstraction layer and does not leave the camera nodes and is not stored.

In this paper we focus on the first three layers of abstraction and workflow steps comprising the video monitoring system, which provides input for the high-level complex event detection of critical situations.

2.1 On-Board People Detection

In this section we present the first step of data processing, which is executed directly on the camera nodes. The aim of the software component "People Detection" is to find people in the stream of images captured by each camera and to provide information about the size of detected persons and their relative positions in the frame.

As mentioned in Section 1, privacy is an essential issue in people monitoring systems. Since cameras are looking down on the scene and since we are not interested in exact people silhouettes, we approximate people with circles. In order to compute the position and the size of circles, two challenges are to be solved. First, pixels representing the crowd should be detected as a foreground. The second challenge is to assign foreground pixels to according circles.

2.1.1 Foreground detection

We implemented two methods for infrared cameras and one method for Kinect cameras for solving the first challenge of people detection.

We developed an approach for homogeneous static backgrounds as presented in Figure 3 showing an original infrared frame on the left hand side. However, the frame is never stored and is depicted as an illustration of the scene. We model the scene with one probability distribution of gray levels building a histogram. People are then represented by values around the maximum peak in the histogram. By cutting tails around the peak, gray levels for foreground pixels are given. A tail starts where the histogram falls below 10 percent of the peak height. In order to achieve adaptivity to changes in the scene, we refill the histogram and detect peaks for each successive frame. We denote this approach "HIST", for histogram-based approach.

The second approach for infrared cameras from [5] is based

on the idea of background subtraction. We apply this approach as a reference implementation for people detection. Each input frame is compared to a background model. The difference are foreground pixels. The model of the background is based on the assumption of static backgrounds. Each pixel of the background is modeled by a mixture of three to five Gaussian distributions. The weights in the mixture are proportional to the time during which the gray level was observed in the scene. We denote this adaptive approach "MOG" for Mixture of Gaussians.

The third approach for Kinect cameras exploits information from the built-in depth sensor. We compute the range where people typically may be found depending on the height of cameras in the set-up and look for local maxima in the depth map within this range. Then we mark pixels within a predefined interval around local maxima in vertical, horizontal and vertical directions as a foreground and apply dilation operation, in order to represent people's heads.

2.1.2 *From Foreground to People*

The second challenge in the image processing flow is to find single persons in the foreground. First, we find regions of the foreground, which are directly connected or neighboring. Connected pixels represent parts people but not necessarily whole persons. The connected-component labeling [3] was used for connected regions exploration. In order to find parts of persons belonging to one object, we cluster them applying common density-based techniques [4].

People who appear in the scene very close to each other may wrongly be detected as one person. Therefore, we split clusters that exceed a dedicated cluster size threshold. The threshold depends on the camera height for the current set-up which is set once during the initialization phase and system deployment.

The last step is to approximate clusters representing people with circles. Therefore, we count pixels belonging to each cluster as the circle area and compute the circle radius then. The coordinates of the center of gravity of each cluster represent the position of a circle in the frame.

The output of the software component for people detection on the lowest abstraction level is a list of detected people represented as circles, position of each detected circle and its radius. The stream of these lists is then sent to the next abstraction level "Data Aggregation" (see Figure 2).

2.2 Data Aggregation on Density Maps

The software component "Density Map Computation" on the "Data Aggregation" layer derives a stream of DMs from a stream of people positions for each of camera view separately. Simple counting of detected people may be not sufficient in cases where persons stay directly on the grid separations. The grid is given by the height and the camera angle and is computed once during the system deployment. Figure 4 shows a grid of one camera view cutting people represented by colored circles.

In such cases, we count people proportionally to the area they occupy in the cell. Therefore, a geometrical approach for cutting a circle by a grid was developed. First, for each

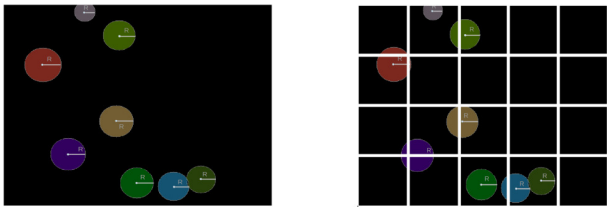

Figure 4: Detected people (left), virtual grid crossing detected people (right).

circle divided by a grid, parts belonging to different cells $i, i = 1, ...n$ are identified. Areas A_i and the area occupied by a whole person $A = \sum_{i=1}^{n} A_i$ are computed in the second step. The person is counted to a corresponding grid cell i according to proportion $\frac{A_i}{A}, i = 1, ..n$.

We worked out sixteen possible geometrical cases and sixteen corresponding formulas for automated computation of A_i. Having the proportions for cut persons, the DM is computed as a sum of parts of people to according cells.

Cameras compute and send one density map per second to the next abstraction "Data Fusion Layer" (see Figure 2). Modeling data with Density Maps allows the use of typical surveillance systems communication infrastructure, with low requirements on communication throughput capacity. In our experiments, for convenience, we used wired connections between cameras and central node. However, since DMs can be coded efficiently (roughly one float per square meter cell, coded each over 32 bits), a back-of-the-envelope computation indicates that even a camera monitoring a large area (e.g., 1000 square meters) would only send 32kbit/s data throughput, which is easily achievable with low-power wireless communication, with off-the-shelf hardware.

2.3 Data Fusion to a Global Spatial View

The software component "Global Density Computation" constructs a grid-based Global Spatial View (GSV) from distributed sources. We define camera views as well as the Global Grid by their rectangle size and coordinate of the left upper corner in the global system of coordinates. First, an automatic configuration of the system, i.e., generation of the GSV, is ensured. The coordinate of the GSV is given by the most left and top placed camera. The size of the rectangle is derived from parameters of the most right and down placed camera. We compute the number of cameras "looking" at each cell of the GSV. In order to be able to represent any shape of the GSV with a rectangle, we note cells with no camera "looking" at it with the number zero. An example of an automatically computed GSV is depicted in Figure 5 on the right hand side.

During the system runtime, streams containing lists with density maps from all the cameras are synchronized: a GSV is computed once a second using rounded timestamps for each list. Therefore, the maximum delay between data leaving the camera node and data processing is measured with millisecond precision. For each second the component waits for n timestamped lists, where n is the number of cameras in the set-up. The bottleneck of this approach is dropped output of the whole component in case of missing input from a camera node.

Having precomputed properties of the GSV, the component knows how many cameras are "looking" on each cell. First, the coordinates of cells for local DMs of rotated cameras are transformed to the global system of coordinates. For cells with only one input source, the number of people from local DM can be directly accepted. For cells "seen" by more than one camera people should not be counted repeatedly and therefore we average the counting result. It is possible to trust the output of one cameras more than the others by different weighting of their counting result for a weighted mean.

The component is able not only to compute the GSV for the number of people automatically for any number of rotated cameras, but also to visualise the result in real time. The visualization was used for evaluation of results which we present in the following section.

3. EXPERIMENTS AND EVALUATION

In this section, we present an experimental evaluation of the people counting solution. We run three data processing layers from Figure 2 on real hardware. The software component for people detection is implemented in OpenCV and exploits provided data types for efficient processing of image data. The computation of local and global DMs is realized in Python. In order to minimize the data transfer, we physically co-locate the "Distributed Video Processing" and the "Data Aggregation" layers i.e., both layers run on each distributed node. We used a hand custom hardware based on an Intel z530 Atom processor, to produce DMs at the rate of one DM per second.

We conducted two series of experiments. The first series of experiments were carried out in the Schönefeld airport in Berlin, at the occasion of a public airshow gathering large crowds (ILA 2014), in the context of the SAFEST project [1]. The second series of experiments were carried out at the Freie Universität Berlin, in a busy, central hallway. For both experimental setups, we used the exact same cameras. In both setups, we recorded hours of video sequences and ran the workflow described in Section 2. Manual, offline comparison of the sequence of global density maps against the recorded video streams allowed us to estimate errors between the results of our workflow and the ground truth. The results below are based on the analysis of 100 randomly selected snapshots of the video streams, compared to the corresponding global density map that was computed online. The result of the comparison for each snapshot provides a data point, for which we computed relative and absolute errors as reported below. By relative error, we mean the overall uncertainty of the counting result, in percentage. By absolute error we mean the deviation of the counting result from the ground truth in number of people present in the snapshot.

3.1 Live Experiments in the Schönefeld Airport

The set-up in the Schönefeld airport in Berlin included two Kinect cameras and one infrared camera installed at the height of 5.7 meters monitoring a 70 square meter area, placed directly above the crowd, similarly to the scenario shown in Figure 1. The area covered by cameras, their rotation and overlapping pattern are shown in Figure 5: the infrared camera covers the whole scene and is rotated 180° in the global system of coordinates, while Kinect cameras cover only partial, overlapping areas, and are rotated 270° and 0° respectively.

Figure 5: Airport experiment: live snapshot of the central control (left), camera coverage of the area (right). Camera 1 is the infrared camera, cameras 2 and 3 are Kinect cameras.

Table 1: Mean relative and absolute errors $\delta x/\Delta x$ for airport experiments with infrared and Kinect cameras.

	Infrared	Kinect	Global Spatial View
HIST+Kinect	6%/0.15	21%/0.68	16%/0.42
MOG+Kinect	4%/0.1	19%/0.47	14%/0.47

In these experiments, we compare the precision obtained with HIST on the infrared camera (see Section 2.1.1) to the precision obtained with MOG on the infrared camera (with additional Kinect cameras, in both cases). The results are shown in shown in Table 1. As we can observe, the precision of counting results is similar for both background subtractions algorithms. Because the homogeneity of the background and mostly no other objects other than people and chairs were in the scenes, the background modeling was simple and similar in both cases. More complex cases, where people are too close to each other (e.g. holding hands), are a potential source of errors. We could verify that the system corrects detection of individuals in such situations. Nevertheless, the precision obtained by the Kinect cameras is very disappointing. Our hypothesis is that the height of 5.7 meters was operating too close to the recommended range for Kinect, which is approx. 4 meters [8] (despite the offset due to people's height which we estimated at 1.7 meters on average).

3.2 Live Experiments in a Busy University Hallway

In order to verify our hypothesis that Kinect range was exceeded in the airport experiments and to involve all of the available hardware for more general evaluation, we conducted another series of experiment in a busy, central university hallway. We placed both Kinects and one of the infrared cameras at the more conservative height of 3.5 meters and another infrared camera at the height of 8 meters. In this setup, the views of all cameras are overlapping and are included in the area of the higher infrared camera. The results are presented in Table 2. We observe a much improved precision for the Kinect. Furthermore, the accuracy of the Global Spatial View is similar to the accuracy of each single camera, which confirms that our system does not multiply errors coming from different cameras monitoring the same area. The Global Spatial View even eliminates errors by intelligently leveraging multiple sources of information sources.

Table 2: Mean relative and absolute errors $\delta x/\Delta x$ for hallway experiments with two infrared (IR) cameras and two Kinects.

	Result $\delta x/\Delta x$
IR	6.9% / 0.2
Kinect	6.4% / 0.14
IR +Kinect	6.5%/ 0.15
2*IR+2*Kinect	6.6% / 0.19

In order to verify this claim, we constructed a set-up of only one infrared and one Kinect cameras. We notice that when errors coming from the infrared camera running HIST (row 1 of Table 2) tend to be corrected in the Global Spatial View (row 3 of Table 2), using the additional information coming from the Kinect camera (row 2 of Table 2). However, the correction has only a small impact on overall results. Infrared camera covers an area that only partially overlaps the area covered by the Kinect. Consequently the errors can happen in areas that are not covered by the Kinect and can thus not be corrected.

In fact, the global error does not significantly increase with number of erroneous sources (compare row 3 and 4 of Table 2). However, the counting accuracy depends on the camera placement accuracy: the global grid has an accuracy of one meter, while the cameras are placed using existing infrastructure and the accuracy of their coordinates is measured in centimeters. The rounding of camera coordinates i.e., cutting the borders of each camera view in order to fit it into the global grid and different heights of cameras in the set-up, have an impact on counting which can become slightly inaccurate.

However, we verified that errors around 6% are achieved with this approach, which is reasonable. Furthermore, we note that enhancing the Kinect camera (or equivalent) to accommodate bigger range/height is not inconceivable, at small price. Thus, our experiments confirm the relevance of the architecture, algorithms and basic hardware setup we proposed in this paper.

4. CONCLUSION

Emergency management needs a reliable detection of abnormal situations. In this paper, we presented a holistic approach for multi-camera privacy-preserving crowd density computation. The developed solution for infrared and Kinect cameras detects and counts people in real-world indoor scenarios and provides an online surveillance system with a stream of aggregated grid-based density information.

We developed a layered model for data aggregation and a workflow model for execution of software components. Both allow flexible system modeling and employment of various system set-ups composed of several distributed and centralized nodes.

For each of the processing layers we developed a set of software components. The presented solution for crowd density estimation comprises people detection on the lowest onboard processing layer, computation of local densities, and fusion of single views to a global density on higher aggregation layers. The result is sent to the event detector on the top aggregation layer for further evaluation.

By pre-processing and aggregating data at each processing step, not only counting accuracy is achieved, but also people privacy is preserved and lightweight data transfer is insured between the software components.

Furthermore, we ran experiments in a real-world environment in order to evaluate our solution and we confirmed the appropriateness of the developed approach. At this current state of research, we are able to provide an online global density information from distributed cameras of two types and visualize the result.

Acknowledgements

The research reported in this paper was sponsored by the German Federal Ministry of Education and Research (BMBF) and the French Agence Nationale de la Recherche (ANR) grant, as part of the Franco-German research project SAFEST. Part of this work was carried out within the European Union (EU) City.Risks research project.

5. ADDITIONAL AUTHORS

Additional authors: Gabriel Hege (Daviko GmbH, email: `hege@daviko.com`) and Mark Palkow (Daviko GmbH, email: `palkow@daviko.com`)

6. REFERENCES

[1] E. Baccelli, G. Bartl, A. Danilkina, V. Ebner, F. Gendry, C. Guettier, O. Hahm, U. Kriegel, G. Hege, M. Palkow, H. Petersen, T. C. Schmidt, A. Voisard, M. Wählisch, and H. Ziegler. Area & Perimeter Surveillance in SAFEST using Sensors and the Internet of Things. In *Workshop Interdisciplinaire sur la Sécurité Globale (WISG2014)*, Troyes, France, Jan. 2014.

[2] A. B. Chan, Z. John, and L. N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, June 2008.

[3] M. B. Dillencourt, H. Samet, and M. Tamminen. A general approach to connected-component labeling for arbitrary image representations. *J. ACM*, 39(2):253–280, Apr. 1992.

[4] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[5] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems*, volume 5308, 2001.

[6] V. Kettnaker and R. Zabih. Counting people from multiple cameras. In *IEEE International Conference on Multimedia Computing and Systems, ICMCS 1999, Florence, Italy, June 7-11, 1999. Volume II*, pages 267–271, 1999.

[7] Kinect Camera. http://www.xbox.com/en-US/xbox-360/accessories/kinect/KinectForXbox360. 2010.

[8] Kinect Sensor. http://msdn.microsoft.com/en-us/library/hh438998.aspx. 2012.

[9] T.-Y. Lin, Y.-Y. Lin, M.-F. Weng, Y.-C. Wang, Y.-F. Hsu, and H.-Y. Liao. Cross camera people counting with perspective estimation and occlusion handling. In *Information Forensics and Security (WIFS), 2011 IEEE International Workshop on*, pages 1–6. IEEE, 2011.

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

[11] T. T. Santos and C. H. Morimoto. Multiple camera people detection and tracking using support integration. *Pattern Recognition Letters*, 32(1):47 – 55, 2011. Image Processing, Computer Vision and Pattern Recognition in Latin America.

[12] T. Teixeira and A. Savvides. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on*, pages 36–43, Sept 2007.

[13] X. Wang. Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3 – 19, 2013. Extracting Semantics from Multi-Spectrum Video.

[14] L. Xia, C.-C. Chen, and J. Aggarwal. Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15–22, June 2011.

[15] D. B. Yang, H. H. González-Baños, and L. J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 122–129. IEEE, 2003.

[16] C. Zeng and H. Ma. Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2069–2072. IEEE, 2010.