BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Thesis

PEOPLE COUNTING USING AN OVERHEAD FISHEYE CAMERA

by

SHENGYE LI

B.Eng., University of Electronic Science and Technology of China,
2017

Submitted in partial fulfillment of the

requirements for the degree of

Master of Science

2019

Approved by

First Reader

Janusz Konrad, PhD
Professor of Electrical and Computer Engineering

Second Reader

Prakash Ishwar, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

Third Reader

Osama Alshaykh, PhD
Lecturer and Assistant Research Professor of Electrical and Computer Engineering

Acknowledgments

First of all, I would like to thank Professor Janusz Konrad. He was not only my Master's thesis advisor, but also a generous mentor. He taught me image processing, offered me a chance to work in his group, and shared his great experience and wisdom during the last 12 months. He also devoted a lot of time and effort to reading and revising this thesis. It is safe to say that I would not have completed this thesis without his instruction and encouragement. He deserves my highest respect and appreciation.

Secondly, I would like to thank Professor Prakash Ishwar who has greatly helped me with my research. His clarity of thought, logical analysis and probing questions were instrumental for me pushing forward when the path looked foggy.

Also, I would like to thank Professor Osama Alshaykh, who served on my thesis committee.

I would also like to give credit to Ozan Tezcan, a PhD student also working on Professor Konrad's ARPA-E project. He shared with me his knowledge in the field of deep learning, and helped me improve my research skills. This project would have taken much longer had he not helped me.

Finally, my appreciation goes to my friends and family for their help, care and, of course, financial support. Particularly, I would like to credit my roommate, and also one of my best friends in Boston, Peide Liu, for easing my concerns, and powering me to live with optimism in the tough times of graduate study.

Shengye Li

PEOPLE COUNTING USING AN OVERHEAD FISHEYE CAMERA

SHENGYE LI

ABSTRACT

As climate change concerns grow, the reduction of energy consumption is seen as one of many potential solutions. In the US, a considerable amount of energy is wasted in commercial buildings due to sub-optimal heating, ventilation and air conditioning that operate with no knowledge of the occupancy level in various rooms and open areas. In this thesis, I develop an approach to passive occupancy estimation that does not require occupants to carry any type of beacon, but instead uses an overhead camera with fisheye lens (360 by 180 degree field of view). The difficulty with fisheye images is that occupants may appear not only in the upright position, but also upside-down, horizontally and diagonally, and thus algorithms developed for typical side-mounted, standard-lens cameras tend to fail. As the top-performing people detection algorithms today use deep learning, a logical step would be to develop and train a new neural-network model. However, there exist no large fisheye-image datasets with person annotations to facilitate training a new model. Therefore, I developed two people-counting methods that leverage YOLO (version 3), a state-of-the-art object detection method trained on standard datasets. In one approach, YOLO is applied to 24 rotated and highly-overlapping windows, and the results are post-processed to produce a people count. In the other approach, regions of interest are first extracted *via* background subtraction and only windows that include such regions are supplied to YOLO and post-processed. I carried out extensive experimental evaluation of both algorithms and showed their superior performance compared to a benchmark method.

Contents

List of Tables

List of Figures

List of Abbreviations

AA	Activity-Aware Method
AB	Activity-Blind Method
ACF	Aggregate Channel Features
BAS	Building Automation System
CA	Correct Acceptance
CCR	Correct Classification Rate
CNN	Convolutional Neural Network
CR	Correct Rejection
FCN	Fully Convolutional Network
FN	False Negative
FOV	Field of View
FP	False Positive
HOG	Histogram of Gradients
HVAC	Heating, Ventilation and Air Conditioning
IOU	Intersection over Union
KRR	Kernel Ridge Regression
LBP	Local Binary Pattern
MAE	Mean Absolute Error
MAP	Maximum a Posteriori Probability
MOTA	Multiple Object Tracking Accuracy
NMS	Non-Maximum Suppression
PCA	Principal Component Analysis
PDV	Person-Detection Verification
PIR	Passive Infra-Red
ROI	Region of Interest
SOR	Spatial Outlier Rejection
SVM	Support Vector Machine
TP	True Positive
WA	Wrong Acceptance
WR	Wrong Rejection
YOLO	You Only Look Once

Chapter 1

Introduction

As climate change concerns drive both social and political movements in many countries, reduction of energy consumption is one of many solutions under consideration. In the United States, among many energy uses, heating and cooling of buildings is one of the most wasteful. The US Department of Energy estimates that a considerable amount of energy is wasted in commercial buildings due to sub-optimal heating, ventilation and air conditioning (HVAC). In order to optimally control HVAC equipment, and therefore save energy, a building automation system (BAS) must be supplied with occupancy level (number of people) in all rooms and open areas. Equipped with this information, BAS can provide optimal amount of air wherever and whenever needed, without unnecessary waste.

In addition to optimal HVAC control, occupancy information can be of substantial value in space management, hospitality industry, etc. For example, in order to optimize office use, especially in high-cost real-estate markets, companies would like to know how well space is utilized in a building they rent (How often are conference rooms in use? Can their number be reduced to save on rent?). In hospitality industry (e.g., hotels), management would like to know which and when rooms are unoccupied.

Therefore, indoor space occupancy estimation has become an active topic of research. At a high level, there are two types of approaches to occupancy sensing: active methods and passive methods. Active methods require that each potential occupant carry a “beacon”, a portable device that communicates with a data collection unit mounted nearby. This mechanism necessitates the use of beacons which may be invasive, inconvenient, unreliable (when one forgets) and costly to implement, thus making this approach less attractive. As for passive methods, a common solution are PIR (passive infra-red) sensors mounted overhead, but they can only provide a binary occupancy estimate (occupied or unoccupied). Therefore, even if one person enters a conference room with maximum occupancy of 20 people, the HVAC system will deliver maximum airflow. While energy savings will be realized when the room is empty, no energy will be saved when the room is partially occupied. Clearly, a more fine-grained occupancy estimate is needed to realize further energy savings. In other words, BAS must know the number of people in a space and based on the ratio of this number to the maximum number of people allowed, adjust the airflow.

Reliable counting of people in a room by passive approach requires a sufficiently wide field of view and sufficiently high resolution of a sensing device. One such common device these days is surveillance camera, often seen at building entrances, in elevators, hallways, etc. While most surveillance cameras are side-mounted (at sufficient height in order to cover a large area) and are equipped with a lens that typically delivers a 30-90° field of view (FOV), they are not optimal for indoor applications as several such cameras may be needed to fully cover a room. An alternative are cameras with fisheye lens, sometimes also called panoramic or 360° cameras, for the fisheye lens typically provides a 360° horizontal by 180° vertical field of view. When mounted overhead, such cameras can provide a full view of the room while largely avoiding occlusions. This is very difficult and costly to accomplish with side-mounted, standard-lens cameras (several cameras are needed along with substantial post-processing to eliminate double counting of people). A single fisheye camera is often sufficient in such scenarios.

However, a fisheye camera mounted overhead captures human body in various orientations depending where the person stands in the room. The body can be seen from above (only head and shoulders of a person standing directly under the camera) or in upright, upside-down, horizontal or diagonal orientation. This impedes a direct application of standard people detection algorithms developed for the usual upright pose. An additional complication is the fact that image captured by a fisheye camera exhibits significant geometric distortions that increase close to the periphery of camera's FOV.

Many people detection algorithms, both model-based and data-driven, have been developed for side-mounted, standard-lens cameras. Among data-driven methods, those leveraging the generalization capacity of deep convolutional neural networks (CNNs) have been particularly successful. Some of these methods will be reviewed in Chapter 2. The success of CNNs in people detection has inspired me to adopt them to images captured by overhead, fisheye cameras. However, this adoption is not straightforward and I will develop two approaches that leverage state-of-the-art deep-learning object detection algorithms. In one approach, called "activity-blind method", I apply a deep-learning object detection algorithm to a sequence of rotating, partially-overlapping windows extracted from a fisheye image. Since these windows partially overlap, multiple detections may be associated with one person. In order to combat this, I developed a post-processing algorithm that fuses redundant detections and performs the final count. In another approach, called "activity-aware method", I first detect regions of interest (ROIs) to which the same deep learning algorithm is applied. I detect ROIs by means of background subtraction, a computational algorithm that detects changes in the field of view of the camera. This approach attempts to detect people only in areas that undergo change without wasting computational resources on areas that are unlikely to contain people (no change).

This thesis is organized as follows. Chapter 2 reviews the current state of the art. In Chapter 3, I present the activity-blind method for people detection based on a state-of-the-art deep learning object detection algorithm. In Chapter 4, I describe ROI detection and the activity-aware algorithm. In Chapter 5, I show a wide range of experimental results including a comparison between the activity-blind method, activity-aware method, and a benchmark method. Finally, I draw conclusions in Chapter 6.

Chapter 2

Relevant Work

There exists vast literature in the area of people detection from a single RGB image, the focus of this thesis. I will not consider people detection from depth images, stereo images or video. First, I will briefly review recent, top-performing people detection methods from an RGB image captured by a side-mounted, standard-lens camera (pinhole projection model). Then, I will describe some very recent methods applied to fisheye images captured by an overhead camera.

2.1 People detection from side-mounted, standard-lens cameras

The literature is vast in this area, so I will concentrate on 3 methods from the last decade or so that have been dominant in terms of their popularity and performance.

Arguably, the most often cited person-detection algorithm is the *Histogram of Gradients* (HOG) employed in conjunction with *Support Vector Machine* (SVM) classifier (Dalal and Triggs, 2005). This method combines model-based and data-driven approaches. The features selected to characterize human-body shape are hand-crafted and use local orientations of spatial gradients in an image. Typically, gradient orientations are quantized to 9 angles (including one bin for small-norm gradients), and then gradient norms, instead of counts, are used to construct a weighted histogram of gradient orientations. After various normalizations, a feature vector of length 36 is associated with each 16×16 block of an image. For an image with N such blocks an $N \times 36$-length vector results. Such feature vectors are then used for training a linear SVM classifier. This approach has been very successful in people detection from side-mounted cameras.

Another very successful and fast pedestrian detector is based on *aggregate channel features* (ACF) (Dollár et al., 2014). It uses 6 channels of the HOG descriptor in combination with 3 LUV luminance/color channels and a channel of normalized gradient magnitudes. These channels are used to construct a feature pyramid with 7 or 8 scales per octave. This pyramid is used as feature vector for training an AdaBoost classifier (Friedman et al., 2000). The method is very fast and allows real-time pedestrian detection. ACF counts among state-of-art people detection methods using hand-crafted features.

Among recent, fully data-driven approaches (no hand-crafted features), convolutional neural networks (CNNs) have been most successful for their generalization capacity. While several methods of this type have been developed in the last few years, including R-CNN and Faster R-CNN, the method that has perhaps attracted most attention is *You Only Look Once* (YOLO) (Redmon et al., 2016). This is a deep CNN architecture that allows object detection with excellent classification performance, and whose newer version (e.g., YOLO version 3 or YOLO v3) does it in almost real time. YOLO has inspired my work on people detection from overhead fisheye cameras.

2.2 People detection from overhead, fisheye-lens cameras

People detection from top-view, $360° \times 180°$ cameras is a much less researched area with the vast majority of papers being published in the last few years. While a side-mounted, standard-lens camera usually captures people in the upright pose, an overhead, fisheye camera records people in a wide range of poses (upright, upside-down, horizontal, diagonal or top-view head and shoulders) depending where they are in the field of view of the camera.

In a very recent method, Wang *et al.* proposed to model people as upright cylinders and derived a series of elliptic detection masks whose size diminishes with the distance from image center (Wang et al., 2007). They applied four SVM classifiers to features derived from each detection mask: HOG feature vector and LBP (local binary pattern) feature vector both from full-size masks as well as HOG and LBP features from half-size masks. Features of half-size masks are computed from the top portion of the full mask. They proposed a simple scene (room) model to help with occlusion handling. The classifiers are trained separately for each concentric layer of same-size masks using manually-extracted masks with people. During detection, they extracted elliptical detection masks in each concentric layer, and derived the visibility of masks from the scene model. The final result is a linear combination of scores from two pairs of SVMs (full- and half-size masks) for HOG and LBP features. The method has been tested and quantitatively evaluated in the context of people tracking on a private dataset of several 2-minute 480×480-pixel videos recorded in a 5m×14m room producing 50-93% MOTA (multiple object tracking accuracy) score.

In another method using HOG features, Chiang and Wang proposed to rotate a fisheye image in 4° steps, and extract a long and narrow window (Chiang and Wang, 2014). The window has height equal to half image height and extends from image center to the top boundary covering image area where people appear upright. The window's width is much smaller than its height; it is wide enough to capture a person but small enough to minimally overlap windows extracted after ±4° image rotations. From this narrow but tall window, they extracted detection windows of various sizes (to capture a range of body sizes). Subsequently, they computed HOG features from each detection window and applied Gaussian-kernel SVM to detect a person. Since multiple detections of the same person are possible due to window overlap, Chiang and Wang developed a merging procedure based on SVM scores to eliminate overlapping detections. They have manually extracted and labeled 583 windows with people and automatically captured 1,166 windows with background (from frames void of people). They used a private dataset consisting of a 2.5-hour 384×384-pixel video of a dance game and demonstrated performance of 98.3% correct classification rate (CCR) on this dataset. This method was an inspiration for me as I followed a similar strategy of extracting rotating windows but with a different classification method and merging procedure.

A Bayesian approach to people detection from fisheye cameras was proposed by Saito *et al.*; this approach leverages a maximum *a posteriori* probability (MAP) model and its subsequent maximization (Saito et al., 2010). In particular, upright human body and top-view head-and-shoulders silhouettes are modeled by means of probabilistic shape features as a function of distance from the camera. They used principal component analysis (PCA) and kernel ridge regression (KRR) to build a template model for human body and used it in MAP formulation. The proposed method was tested on a private detaset composed of 320 overhead, fisheye 640×480-pixel images of a T-shaped hallway, and delivered 86–91% CCR.

In contrast to training new classifiers with top-view, fisheye images, Krams and Kiryati proposed a method that uses ACF trained on side-view, standard-lens images (Krams and Kiryati, 2017). However, the method does not dewarp fisheye images into a panorama image, but instead dewarps features extracted from the fisheye image. This is important since image dewarping uses non-uniform spatial interpolation thus introducing spatially-variant smoothing in the dewarped image. As features used for detecting humans usually apply spatial derivative operators (e.g., to detect edges), this spatially-variant smoothing is detrimental to derivative operator's localization accuracy. Extracting features in the fisheye image and dewarping them to the panorama image avoids this issue. Krams and Kiryati also proposed a method to optimize the feature dewarping model with the explicit goal of maximizing people detection performance. They used an overhead, fisheye-view dataset called BOMNI (Demirz et al., 2012) to test their method and obtained about 0.3 miss rate at 0.1 false positive rate per image.

Recently, CNN-based people detection methods have been proposed for overhead, fisheye images. In one method, YOLO was modified and directly trained on full fisheye images (Nguyen et al., 2016). Unlike original YOLO working on RGB images, the authors used only grayscale channel but augmented it with foreground/background segmentation image obtained using adaptive Gaussian mixture model (Stauffer and Grimson, 1999). With a simplified network structure to compensate for the computational cost of foreground/background segmentation, the method can reach speeds needed for real-time people detection. The method was trained on one sequence from the BOMNI dataset and tested on two other sequences from the same dataset as well as on a private overhead, fisheye 704×576-pixel video. The method achieved 96-99% precision at 96-98% recall when trained on 2,000 frames from several videos and tested on other frames from the same videos. However, when tested on videos with different background or number of people, the method delivered only 50-90% precision at 48-90% recall.

Another YOLO-based people detection method was proposed by Seidel *et al.* The approach leverages YOLO trained on side-view, perspective images to detect people in fisheye images, without specific training on such images (Seidel et al., 2018). The authors proposed to extract highly-overlapping windows in order to avoid misses, and to dewarp those windows by means of an omnidirectional-to-perspective image mapping (local-area projection *via* a perspective model). The dewarped window was then fed into standard YOLO trained on perspective images. In a post-processing step, mutiple detections of the same person (due to window overlap) were fused by means of several variants of non-maximum suppression. The method was evaluated quantitatively on the PIROPO perspective and omnidirectional 600×600-pixel image dataset (del Blanco and Carballeira, 2016) and a private dataset of 1,680×1,680-pixel overhead, fisheye images. The average precision of the best-performing variant of the proposed method (non-maximum suppression with Gaussian smoothing) was 64.6% for the PIROPO dataset and 77.6% for the private dataset.

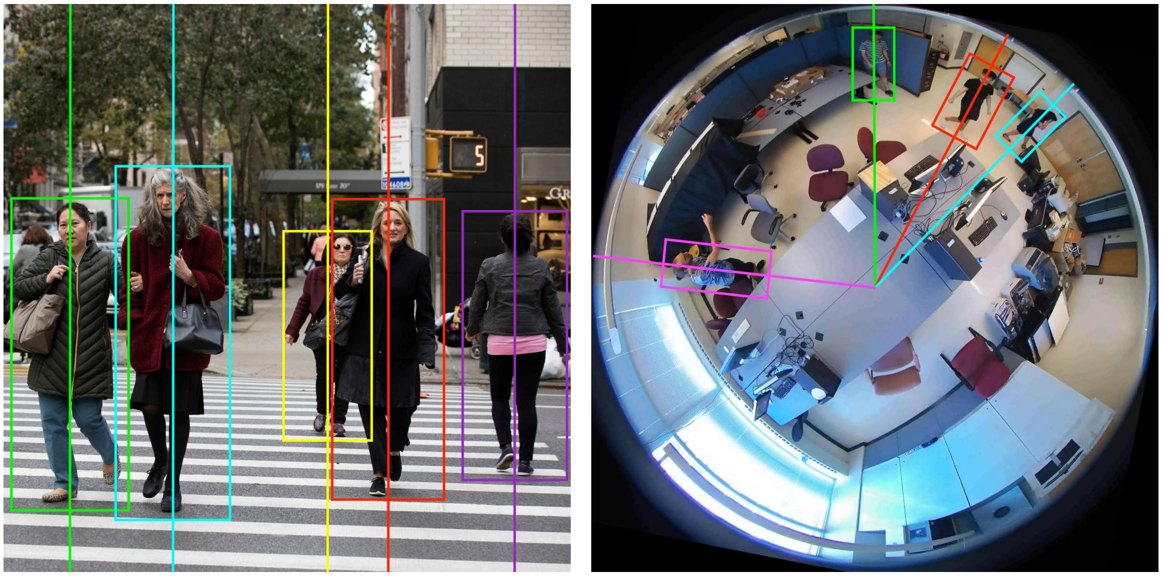
Chapter 3

Activity-Blind Application of YOLO

3.1 System overview

The success of CNNs in object detection from images captured by standard cameras (pinhole projection, perspective camera model) typically held at human-head height, has inspired me to consider CNNs for people detection from overhead cameras with fisheye lens. However, as stated earlier, the application of CNNs in this scenario is not straightforward because people in top-view images may appear in different orientations. People in image center occupy a large area and appear in the shape of an ellipsis; only head and shoulders are visible. People at field-of-view periphery appear smaller than those close to image center but their appearance is similar to those seen in images obtained by standard, side-mounted cameras, although the orientation varies. The body pose, orientation and size in other image areas change gradually. While one could apply CNNs trained on standard-view, perspective images, the performance would suffer. On the other hand, training a new network would require a vast number of overhead, fisheye images with carefully annotated human body locations.

It is easy to observe that a standing person in an image captured by a side-mounted, standard-lens camera is typically vertical to the ground. A corresponding observation can be made about an image captured by overhead, fisheye-lens camera – people standing upright are always aligned with the direction of FOV's radius (line through image center). Therefore, it is safe to say that the appearance of a standing person in the top-center portion of a fisheye image (radius aligned with the vertical axis) is similar to that in side-view, perspective images (Figure 3·1). This observation leads to the first method I am proposing. The main idea is to extract a rectangular window, that I shall call *focus window*, at the top-center of an overhead, fisheye image (Figure 3·3(a)) and apply a people detector to this window. In the next step, I propose to rotate the image by a small angle, extract the same window and apply the detector to the window with new data. Subsequent rotations make sure that windows are extracted from all parts of the image (Figure 3·3(b)). While it is the image that is being rotated, this is equivalent to rotating the focus window; I will refer to this as a rotated window. It is worth noting that since the FOV of a fisheye camera is circular, unlike in the case of perspective cameras with rectangular FOV, image rotation does not cause difficulties due to image locations moving outside of the rectangular image domain or outside locations entering into the domain of the image.



(a)

(b)

Figure 3·1: (a) People in an image captured by side-mounted, standard-lens camera are vertical to the ground. (b) People in an image captured by overhead, fisheye-lens camera are aligned with the direction of FOV's radius.

Since the people detector must be applied many, many times, a fast CNN detection method with excellent performance is essential. Given this constraint, I have opted for YOLO v3, the fastest CNN object detection method to date. This method is based on YOLO v1 (Redmon et al., 2016) and YOLO v2 (Redmon and Farhadi, 2016) and differs primarily in the implementation of various enhancements to make it faster and more precise. Since YOLO detects a wide range of objects, the first post-processing step must retain only those detections (bounding boxes) that correspond to a person with high confidence. Furthermore, weaker detections (YOLO may produce multiple overlapping detections) need to be eliminated. Also, detections close to rotated window's boundary are eliminated for they may be unreliable (a person may be only partially visible). Since the previous steps are implemented after the image has been rotated, the final detections need to be mapped back to the original location of the person in the fisheye image. This mapping may result in multiple bounding boxes being associated with one person since the rotated windows overlap. Again, multiple detections of the same person need to be paired down to a single detection. The final output of the algorithm is the total people count in an image. A block diagram of the proposed method is shown in Figure 3·2; each block in this diagram will be detailed in the remainder of this chapter.

3.2 Window selection and image rotation

The shape and size of focus window to which a people detector will be applied are two important factors to consider. In the case of images captured by an overhead, fisheye camera, the FOV is circular. Therefore, a natural approach would be to extract windows in the shape of rings, so that human bodies are of similar size in each ring window, or in the shape of a wedge, so that human bodies at the same orientation appear in one wedge window. However, a ring-shaped window is not a

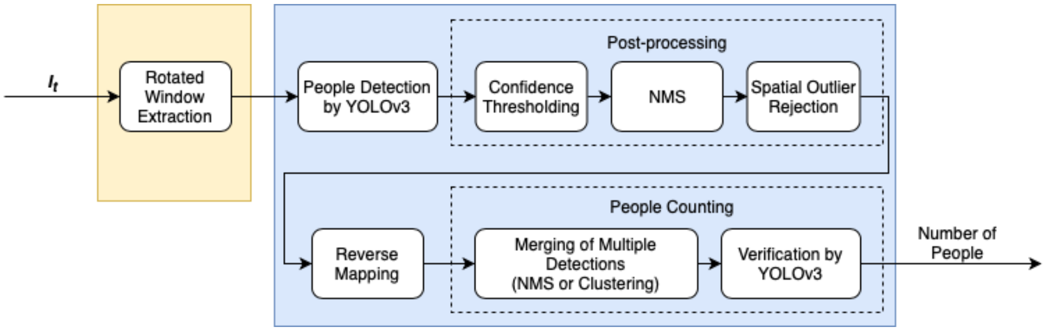
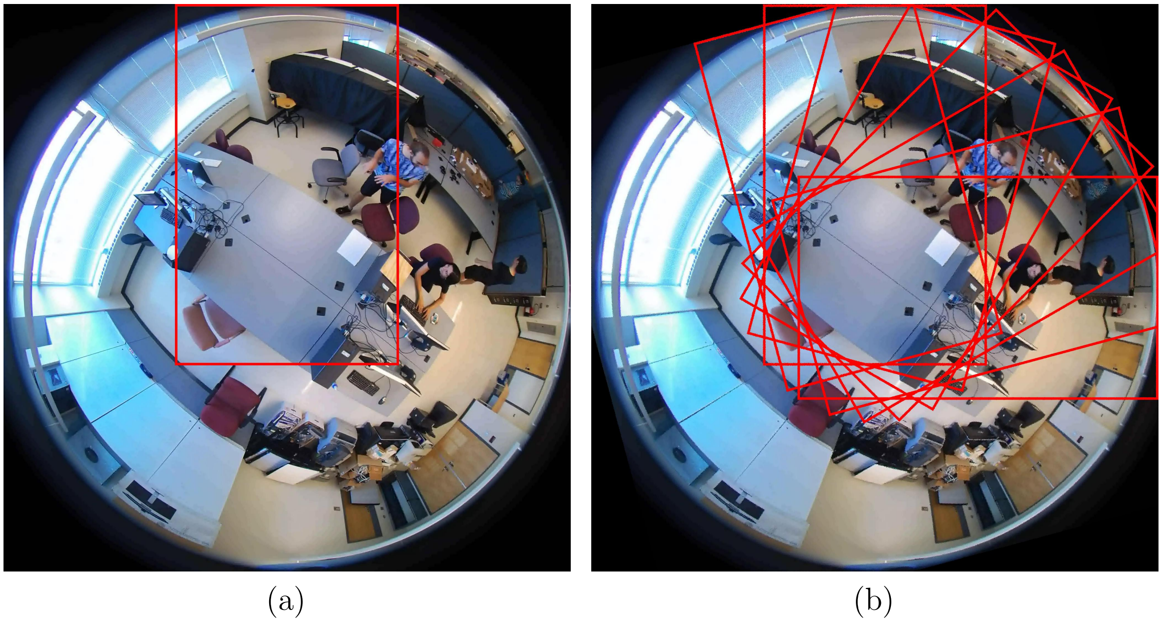

I_t
Rotated Window Extraction
People Detection by YOLOv3
Post-processing
Confidence Thresholding
NMS
Spatial Outlier Rejection
Reverse Mapping
People Counting
Merging of Multiple Detections (NMS or Clustering)
Verification by YOLOv3
Number of People

Figure 3·2: Block diagram of the activity-blind method that applies YOLO v3 to 24 rotated windows. Blocks within the blue-shaded rectangle are common to both methods: activity-blind method discussed in this chapter and activity-aware method presented in Chapter 4, while those within the beige-colored rectangle are unique to the activity-blind method.

good choice since standing people will appear in many different orientations depending where they are located in camera's FOV, thus making it more difficult for a standard people detector to work reliably. As for wedge-shaped windows, they are appropriate for people close to or at FOV periphery because body shapes look similar to those captured by a side-mounted, perspective camera. However, bodies in the central part of the image can be only partially captured due to window's tapered shape. In contrast, a rectangular window is free of these disadvantages and will be used throughout this thesis.

Unlike methods discussed in Chapter 2, that use HOG features and therefore require that a detection window contain one person occupying majority of the window, the YOLO approach was developed for object detection in full-size images. This facilitates extraction of large focus windows after image rotation. Since the Axis M3057 PLVE fisheye-lens camera, that I am using in this project, is equipped with a 6 Megapixel CMOS sensor that effectively records 2,048×2,048-pixel frames with a circular field of view, I selected a window with height of 1,300 pixels and width of 800 pixels centered horizontally and with window's top aligned with the upper boundary of the image (Figure 3·3(a)). This window placement captures human bodies in an upright or almost upright position in the upper half of the window.

As for image rotation, a small angular increment would result in extraction of many focus windows with significant overlap and the need for many applications of a people detector. In order to keep the overall complexity reasonable, I selected a 15° angular step, thus resulting in 24 rotations and extraction of 24 focus windows from each captured frame. Figure 3·3(b) shows 8 rotated focus windows, that is effective rectangular areas extracted after each image rotation.


(a)

(b)

Figure 3·3: (a) Placement of the 1,300×800-pixel focus window whose RGB values are used by people detector. (b) Rotated focus windows, that is effective rectangular areas that are captured after each image rotation.

3.3 People detection

The YOLO v3 algorithm, that I apply to every focus window, is a *Fully Convolutional Network* (FCN) consisting of 75 convolutional layers plus upsampling layers and skipconnection layers. YOLO v3 produces predictions with a stride of 32, 16, and 8, that can be considered a division of the input image into blocks of size 32×32, 16×16, and 8×8, respectively. Each block produces 3 predictions. Each prediction results in a vector of size 85 that contains coordinates of the center and the size of the bounding box, objectness score, and class scores for 80 class items included in the COCO dataset (Lin et al., 2014) on which the model was trained. The first five elements (horizontal and vertical coordinates of the center, horizontal and vertical size of the bounding box, and objectness score), are obtained through sigmoid function, and the 80 class scores are obtained from the softmax function. For an image of size $H \times W$, YOLO v3 produces $3 \times H \times W \times (\frac{1}{32^2} + \frac{1}{16^2} + \frac{1}{8^2})$ predictions in total.

3.4 Post-processing of detections in the focus window

Since YOLO v3 recognizes up to 80 types of objects, a mechanism is needed to reject non-people detections. Also, by its very nature YOLO v3 may detect a person with several bounding boxes that significantly overlap each other within a single focus window. In order to avoid over-counting people, only one representative among the overlapping detections should be retained. Finally, people close to the left, right or bottom boundary of the focus window may not be fully visible resulting in unlikely bounding-box shape and/or size. This will deteriorate the performance of people counting since methods to be proposed in Section 3.6 highly depend on the assumption that bounding boxes for the same person have similar location and size, despite being detected in different focus windows. Thus, detections close to the left, right or bottom focus window boundary need to be suppressed. Detections close to the top of the focus window correspond to a fully-visible body (unless occluded by furniture or other objects) by the very nature of fisheye lens properties.

The three objectives discussed above are accomplished by post-processing the raw detections in 3 steps shown in the block diagram (Figure 3·2) and described below.

1. **Confidence Thresholding**: First, only detections with “objectness” score (confidence level on the scale of 0 to 1 that the detection is a true object from one of the 80 classes) above threshold $\gamma = 0.3$ are retained. Then, out of the retained object detections only those are kept whose person-class score is the highest among all class scores.

2. **Non-Maximum Suppression (NMS)**: All people detections after confidence thresholding are sorted in descending order according to their person-class score. Then, all detections in the sorted list that have a high bounding-box overlap with the detection at the top of the list are suppressed. A high degree of overlap is declared if the *intersection over union* (IOU) between two bounding boxes:

$$IOU = \frac{area\ of\ intersection\ of\ two\ bounding\ boxes}{area\ of\ union\ of\ the\ same\ bounding\ boxes} \tag{3.1}$$

   is above threshold $\rho = 0.4$. Details of NMS implementation are shown as pseudo-code in Algorithm 1.

3. **Spatial Outlier Rejection**: In this step, the remaining bounding boxes of which any part is contained within a Δ-wide margin inside the focus window along left, right and bottom boundaries (Figure 3·4) are rejected. Since for the angular step of 15° the neighboring rotated windows overlap by 84.7%, even with a relatively wide margin Δ no part of camera’s FOV will be uncovered. For the 1,300×800-pixel focus window, I selected $\Delta = 50$.

3.5 Reverse mapping of detections

People detection results (bounding boxes) need to be mapped from the relative position within each extracted focus window to the absolute position in the full fisheye

```
Data: X                                 // The list of input bounding boxes
Result: Y                               // The list of output bounding boxes
/* Initialization                                                          */
Sort all bounding boxes in X by their confidence score in descending order;
Y = [ ] ;
/* iteration begins                                                        */
for $x \in X$ do
    if $Y$ is empty then
        Y.append($x$);
    else
        Flag = True;
        for $y \in Y$ do
            if IOU of $x$ and $y > TH$ then
                Flag = False ;
                break ;
            end
        end
        if Flag then
            Y.append($x$)
        end
    end
end
```

Algorithm 1: Pseudo-code for Non-Maximum Suppression

image. A naïve approach would be to rotate each bounding box in reverse (i.e., by -15° for a focus window obtained by +15° image rotation). However, if a person were perfectly detected (tight bounding box around body silhouette) in two focus windows obtained by image rotations differing by 15°, then the naïve approach would result in the two bounding boxes being at a 15° angle with respect to each other after reverse rotations. Since a perfect detection was assumed, the bounding boxes should have perfectly overlapped. This misalignment reduces the IOU, and may lead to sub-par results. In order to avoid this problem, the center of a bounding box is first reverse-rotated and the bounding box is aligned with FOV radius passing through the new bounding-box center. Both reverse mapping results are illustrated in Figure 3·5.

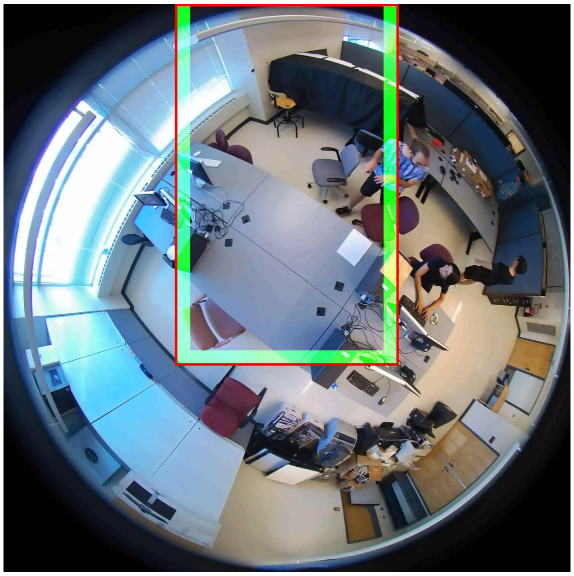


Figure 3·4: Focus window (red) and outlier margin $\Delta = 50$ (green). If a bounding box of detected person overlaps the green area, it is rejected.

3.6 People counting

The reverse mapping of all retained people detections will typically result in multiple overlapping bounding boxes despite bounding box suppression during post-processing (Figure 3·5(b)). This is due to the fact that neighboring focus windows significantly overlap and a person is detectable in several windows. In order to assure an accurate people count, duplicate person detections need to be eliminated so that one bounding box is associated with one person only. In order to accomplish this, I propose two methods. One method is based on the earlier-used NMS while the other method is based on clustering. An experimental comparison of these methods will be carried out in Chapter 5. After the elimination of detection duplicates, one extra step is applied to maximally reduce *False Positives* (FP), namely YOLO v3 is again applied to the remaining bounding boxes with the goal of verifying if the detected object is indeed a person.

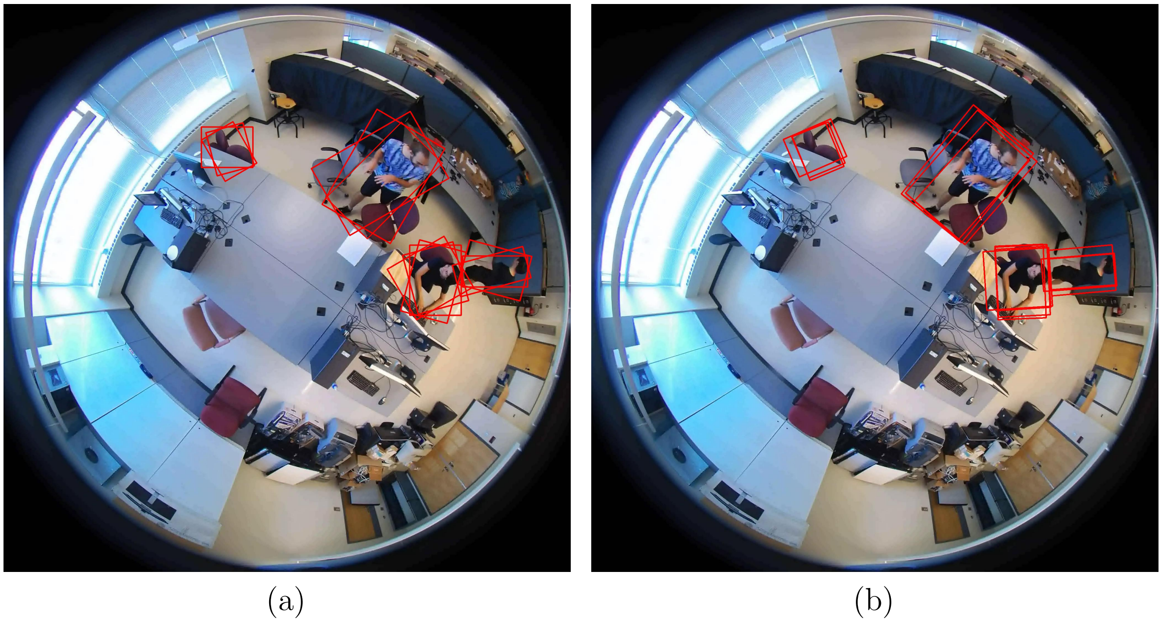
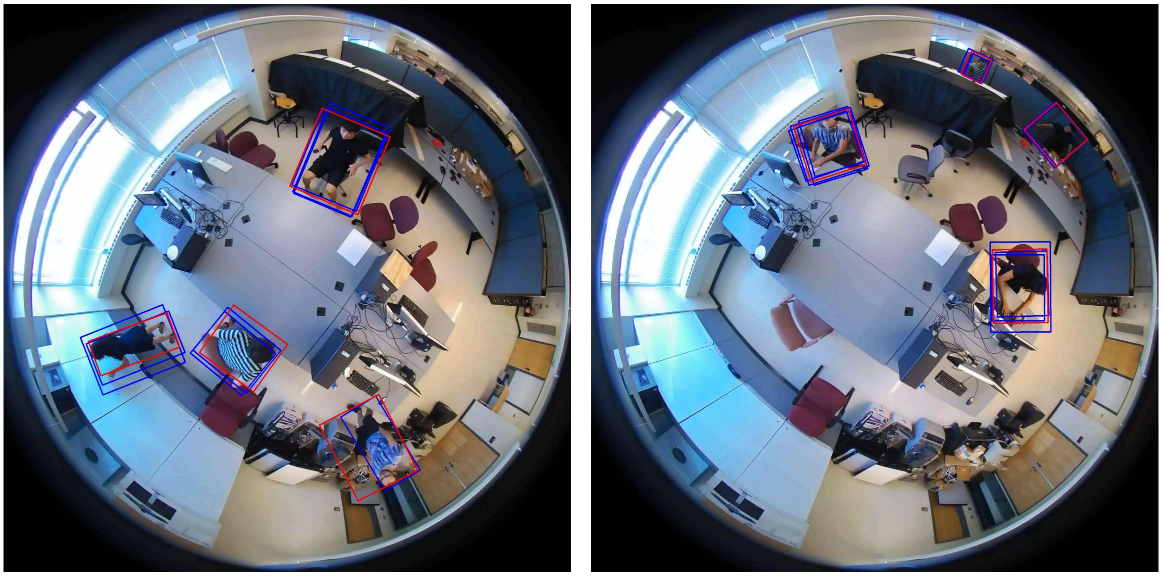


Figure 3·5: (a) Naïve approach: a bounding box is rotated in reverse by the angle of the original image rotation (reverse-mapped bounding boxes are misaligned). (b) Improved approach: the center of a bounding box is rotated in reverse by the angle of the original image rotation and the bounding box is aligned with FOV radius passing through the new center (reverse-mapped bounding boxes are aligned).

3.6.1 NMS-based merging of multiple detections

While multiple detections of the same person can result from different focus windows, the size and location of the corresponding bounding boxes should be similar (the same person may be observed in neighboring focus windows at slightly different relative angles). Based on this assumption, it is logical to apply NMS to all reverse-mapped bounding boxes (Section 3.5) with significant overlap (IOU) to retain only one representative detection. An example result of NMS-based merging is shown in Figure 3·6(a).



(a)

(b)

Figure 3·6: (a) Result of NMS-based multiple-detection merging. (b) Result of clustering-based multiple-detection merging.

3.6.2 Clustering-based merging of multiple detections

However, the NMS-based merging algorithm has several limitations. First, it uses size and location of bounding boxes only while ignoring the underlying color and texture information. This leads to difficulties in distinguishing people standing close to each other. Secondly, the outcome of NMS is as good as the detections provided by YOLO v3; NMS cannot link together two detections that are not close enough or similar enough. Since YOLO v3 has been trained on side-view, perspective images, it is natural to expect that its person localization error will increase for overhead, fisheye images. Clearly, NMS may inherit people detection errors and, therefore, cause errors in people counting.

In order to mitigate these difficulties, I propose to treat people counting as a clustering problem. I propose to cluster the location information for each detection together with the underlying structural information. More specifically, I propose a

12-dimensional feature vector composed of x, y coordinates of the center of a bounding box and a 10-bin normalized histogram of grayscale values within the box. Representing all bounding boxes (both overlapping and non-overlapping) in a full-size fisheye image as 12-dimensional vectors allows application of K-means clustering. Since one needs to provide the value of K to the algorithm and since K is the people count, I added a regularization term to the cost function to limit the number of clusters. The cost function is defined as follows:

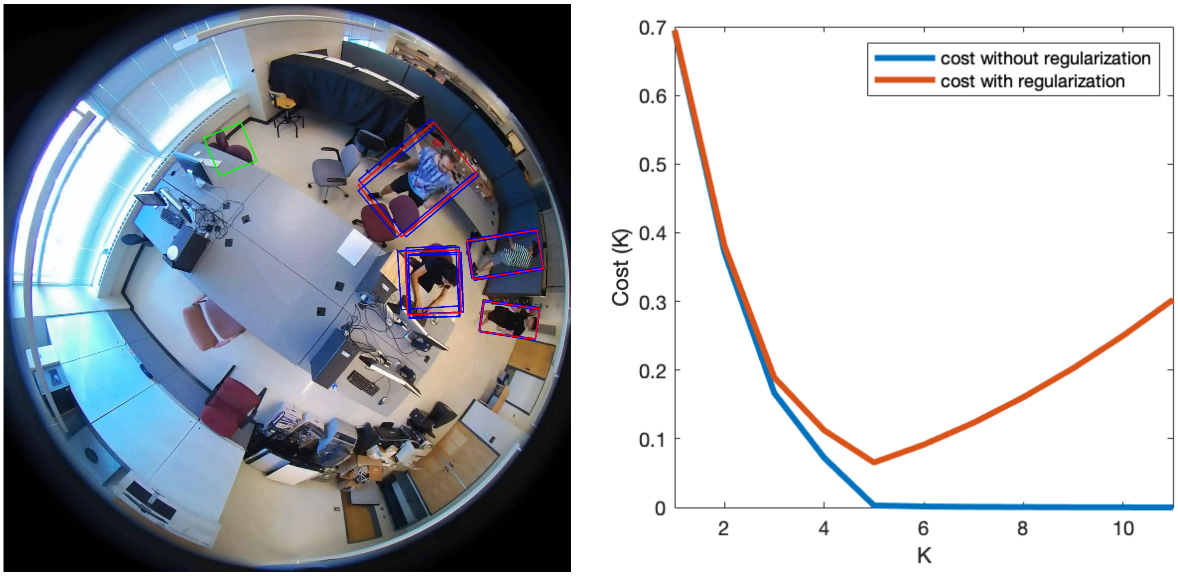
$$Cost(K) = \big(\sum_{i=0}^{K-1} \sum_{D_j \in C_i} (\boldsymbol{d}_j - \boldsymbol{\xi}_i)^2\big) + \alpha \cdot K^2, \tag{3.2}$$

where K is the number of clusters, D_j is the j^{th} bounding box, $\boldsymbol{d}_j$ is a 12-dimensional feature vector describing D_j, C_i is the i-th cluster of bounding boxes, $\boldsymbol{\xi}_i$ is the centroid of cluster C_i, and α is the regularization parameter set to $\frac{1}{400}$ in all experiments. An example of this cost function plotted against K is shown in Figure 3·7(b).

The value of K at which $Cost(K)$ attains minimum defines the final clustering ($K = 5$ in Figure 3·7(b)). Then, x, y coordinates of each cluster's centroid are selected as the center of the representative (output) bounding box for this cluster, while the average width and height of all boxes within the same cluster define the width and height of the representative bounding box. The result of clustering-based people counting is shown in Figure 3·6(b).

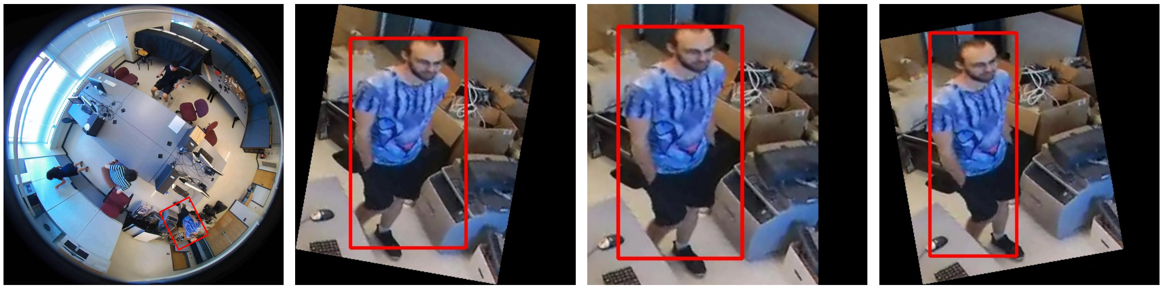
3.6.3 Verification

The final step of people counting is the reduction of false positives that may be due to erroneous detections by YOLO v3 and which may not have been eliminated by the preceding steps. I propose to accomplish this by extracting a new rectangular window around a bounding box produced by either the NMS-based or clustering-based merging algorithm, but slightly larger. Since the body pose may be at any angle, I



(a) (b)

Figure 3·7: Clustering-based merging of multiple detections: (a) input bounding boxes (blue) and resulting bounding boxes (green and red); and (b) corresponding cost function (3.2) with and without regularization for bounding boxes from (a).



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

Figure 3·8: (a) Fisheye image with a candidate detection of a person to be verified (red bounding box in bottom-right). (b-d) Rectangular window around detection from (a) rotated to upright position and further rotated by +15° and -15°. The red bounding box signifies that in each window a person was detected, thus confirming the original detection from (a).

first rotate this new window to obtain a roughly upright pose (Figure 3·8(c)), and then apply additional rotations by +15°, and -15° to account for some angular misalignments (Figure 3·8(b) and 3·8(d)). These three variants of the extracted window are passed to YOLO v3 again and then undergo confidence thresholding and NMS as detailed in Section 3.4 (no spatial outlier rejection is applied). If of the 3 results, either 2 or 3 confirm this is a person then verification is successful (Figure 3·8), and the original bounding box is accepted as showing a person; otherwise it is rejected. The final people count is the number of bounding boxes remaining after this step.

Chapter 4

Activity-Aware Application of YOLO

4.1 System overview

While the activity-blind method performs well, it is computationally complex for the whole chain of steps described in Chapter 3 needs to be applied to each of the 24 focus windows, even if no person is present. Although this is not a problem for a GPU that leverages parallelism, it is a significant obstacle for non-GPU architectures. Therefore, in this chapter I propose an activity-aware method to reduce the computational complexity of people detection from overhead, fisheye cameras. Since it is unlikely that at least one person would be present in each of the 24 focus windows used in the activity-blind method, the main idea in the activity-aware method is to identify regions of interest (ROIs) in camera's FOV, that is regions where people are likely present, and apply a people detector only to these regions. In order to detect ROIs, I apply background subtraction, a well-known methodology in the literature. The block diagram of the activity-aware method is shown in Figure 4·1. Except for ROI extraction and background model update, that I will describe in detail in this chapter, other blocks were detailed in Chapter 3.

4.2 ROI extraction

The purpose of ROI extraction is to identify which among the 24 rotated windows used in the activity-blind method are likely to contain people. To find ROIs, I propose to

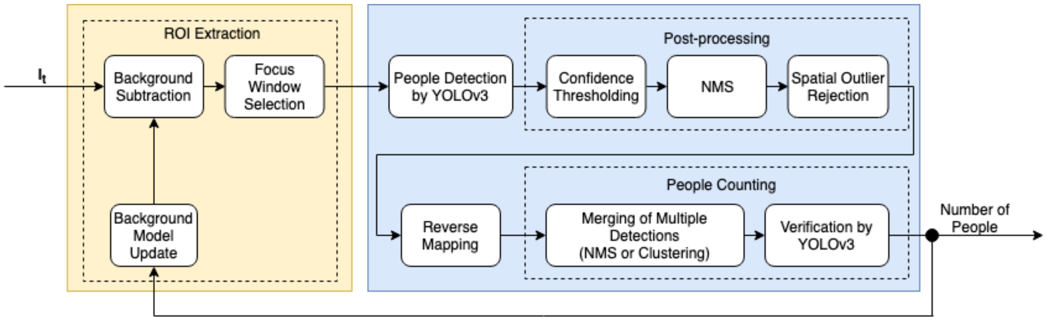


Figure 4·1: Block diagram of the activity-aware method that applies YOLO v3 only to regions of interest. Blocks within the blue-shaded rectangle are common to both methods: activity-aware method discussed in this chapter and activity-blind method presented in Chapter 3, while those within the beige-colored rectangle are unique to the activity-aware method.

detect changes in the FOV of the camera with respect to some reference FOV, often called background, for example a video frame captured in the absence of people. Certainly, a change in camera's FOV does not necessarily imply appearance of a person (it could be a moved chair), but this can substantially limit the computational complexity of the overall algorithm since people detector is applied only to windows containing ROIs. In case a window is triggered by, for example, a moved chair, the expectation is that people detector will fail to acknowledge that it contains a person.

A detailed block diagram of ROI extraction is shown in Figure 4·2. Each block from this diagram will be discussed in subsequent sections.

4.2.1 Background subtraction

Let $\boldsymbol{I}_t(x, y)$ denote a color pixel value captured by overhead, fisheye camera at time t and spatial coordinates x, y. I assume that the camera captures color in RGB space so $\boldsymbol{I}_t = [I_t^R, I_t^G, I_t^B]$, where I^* is a corresponding RGB component of the color image. Also, let $\boldsymbol{B}_t(x, y)$ denote reference background at time t and spatial location x, y. It is also a color image, so $\boldsymbol{B}_t = [B_t^R, B_t^G, B_t^B]$. The next section will describe how the

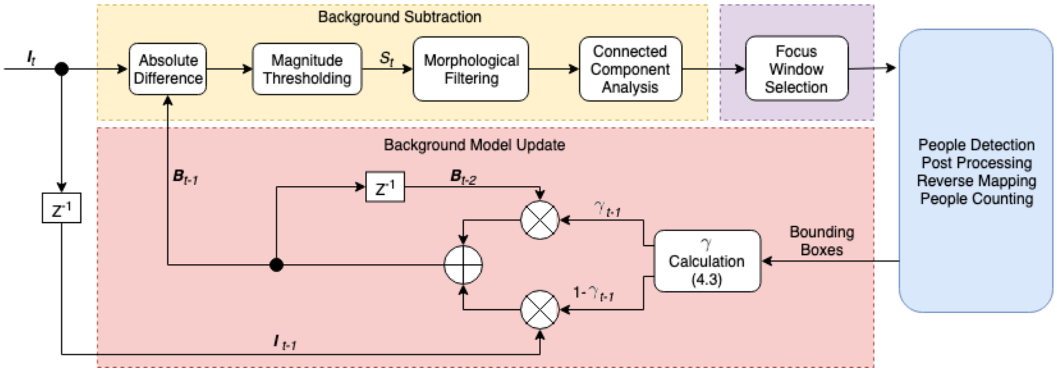


Figure 4·2: Block diagram of ROI extraction steps.

reference background is obtained.

In the first step, the following thresholding is applied to produce an initial mask of changes $S_t(x, y)$:

$$S_t(x,y) = \begin{cases} 1, & \text{if} \quad \sum_{A\in\{R,G,B\}} |I_t^A(x,y) - B_{t-1}^A(x,y)| > \theta \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where θ is a threshold. If θ is too small, the thresholding may produce too many false positives (a slightest departure from the reference background may trigger a detection), and if θ is too large, the thresholding may produce too many false negatives (some changes may be missed). Note that at time t, the current frame is compared with reference background from time $t - 1$. Furthermore, it is the sum of absolute differences for three color components that is compared against threshold θ, not each color component individually; a sufficiently large difference for red, green or blue component will trigger a detection. An example of reference background and input frame are shown in Figure 4·3.

Subsequently, two morphological operations are applied to mask $S_t(x, y)$ in order to remove outliers and make the mask compact. First, opening with a 3×3 rectangular structuring element is applied to remove tiny patches that are likely due to larger-

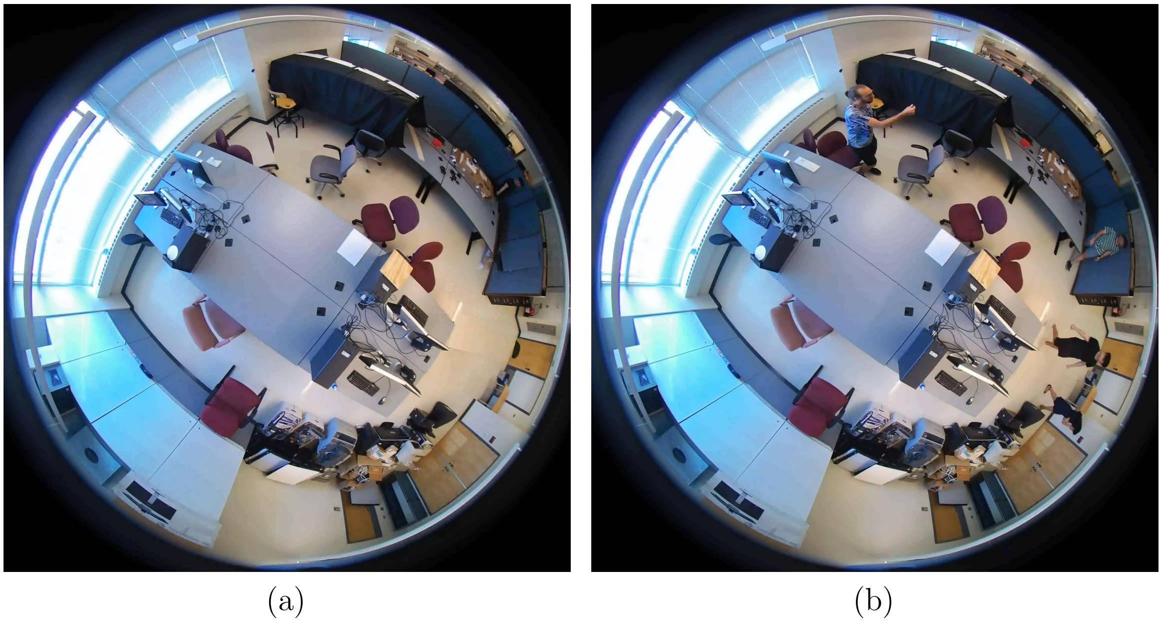


Figure 4·3: Example of: (a) reference background; (b) current frame.

amplitude noise. Then, dilation operation with a 25 × 25 elliptical structuring element is applied to expand the remaining areas of detected changes.

Following the morphological operations, connected-component analysis is performed and small-area components (with less than 3,600 pixels, equivalent to a 60×60-pixel patch) are removed. This leads to the final ROI mask.

The initial mask $S_t(x, y)$, the same mask after morphological operations and the final ROI mask with differently-colored connected components are shown in Figure 4·4 for the current and background frames from Figure 4·3.

4.2.2 Background model

The main difficulty in background subtraction is to obtain a reliable reference background to which the current frame can be compared. In general, a reference background can be modeled as a static or dynamic image.

A static background model is usually an "empty frame" captured by the camera when no people are present in camera's FOV. However, it is difficult to capture such

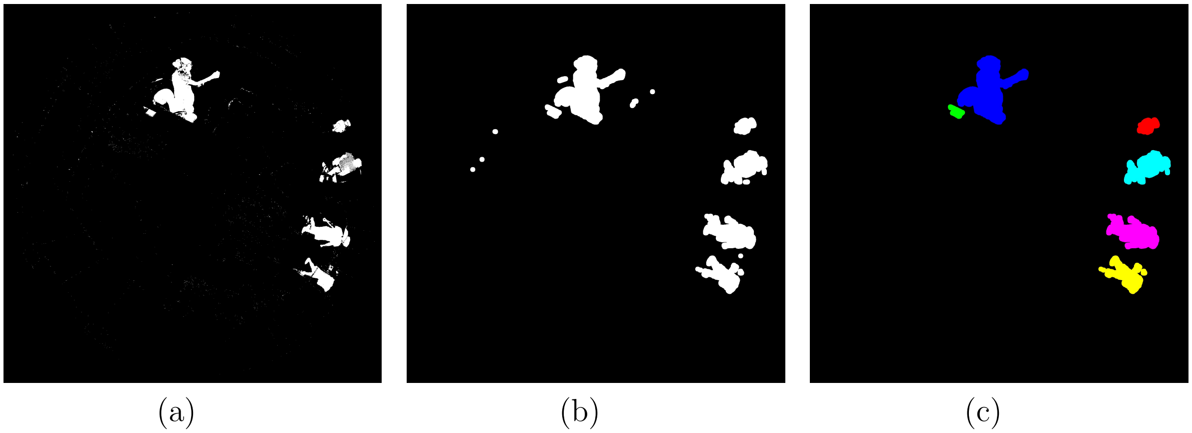


Figure 4·4: (a) Initial mask $S_t(x, y)$ after background subtraction (4.1); (b) the same mask after morphological filtering; (c) final ROI mask after connected component analysis and area thresholding (each connected component is shown in different color).

a frame automatically in real-life situations, unless side information is available (PIR sensor not triggered for hours, strict time schedule of events, etc.) Furthermore, a static background model, by its very nature, cannot reflect changes in the background, such as illumination variations or furniture movement. This, in turn, may lead to unnecessarily large ROIs (lots of false positives). For example, if the current frame is captured under different illumination than that of the background, then the difference in illumination may be detected as change; in extreme case, this may cover the whole FOV. Similarly, if a piece of furniture is moved between the time of background capture and current frame capture, both past and new location of the furniture may be detected as changed areas.

Dynamic background models have been developed primarily to deal with illumination changes. Typically, such models use statistics derived from recent frames to update model's state. If illumination changes on a time scale longer than the derived statistics, then background subtraction can work reliably even in the presence of significant illumination variations. However, dynamic background models can easily get contaminated if a person stays still or almost still on a time scale of the derived statistics; the person will become part of the background causing false change detections later.

In order to minimize the impact of illumination variations, I opt for a dynamic background model but to avoid model contamination by a static person I propose to leverage the result of people detection in the background update mechanism as described below.

The dynamic background model is updated as follows:

$$\boldsymbol{B}_t(x, y) = \gamma_t(x, y) \cdot \boldsymbol{B}_{t-1}(x, y) + (1 - \gamma_t(x, y)) \cdot \boldsymbol{I}_t(x, y), \tag{4.2}$$

where $\gamma_t(x, y)$ is an indicator whether a pixel at time t and location x, y belongs to a bounding box associated with a detected person:

$$\gamma_t(x, y) = \begin{cases} 1, & \text{if } (x, y) \text{ belongs to bounding box of a detected person,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

Clearly, at time t people detection is performed first (so that indicator γ_t can be computed) and then the background is updated. The update mechanism (4.2) uses the indicator γ_t (4.3) to decide at each location (x, y) whether to use the current image value at time t as the new background (pixel belongs to the background) or to use the previous background value at time $t-1$ (pixel belongs to bounding box of a detected person). As long as a person is detected at location (x, y), some previous background value from time $t-2$, $t-3$, ... will be used, that is the bounding box with a person will not get included in the background model. This prevents background contamination common to many dynamic background models, as elucidated above. Since background locations outside of a person's bounding box get immediately updated by the current image value, the model is robust to illumination changes that are challenging for static background models. Illumination changes may be impactful only if a detected person remains static on a time scale longer than that of illumination variations. Clearly, the proposed update mechanism offers benefits of both static and dynamic background models.

There exists a practical issue of initializing the model. Since for the very first frame of a fisheye video ($t = 0$), there is no past information available (whether past frames or past people detections), the background is initialized as zero: $\boldsymbol{B}_0(x, y) = 0,\ \forall(x, y)$. No other processing is applied.

In subsequent frames, in order to update background $\boldsymbol{B}_t$ (4.2) the indicator γ_t is needed and this, in turn, requires background subtraction to detect ROI (Figure 4·4) followed by people detection applied to this ROI (activity-aware method). However, since for frame at $t = 1$ background subtraction uses background $\boldsymbol{B}_0$ (4.1), which is set to 0, no reliable ROI can be computed. Therefore, at $t = 1$ the activity-blind method with 24 rotated windows (Chapter 3) is applied to find bounding boxes with people. Once a more reliable background is obtained at $t = 1, 2, 3, ...$, background subtraction is used to find ROI areas and followed by people detection applied selectively in those areas only (Section 4.2.3).

Figure 4·5 shows an example of background model update. While $\boldsymbol{B}_0$ in Figure 4·5(a) is black (zero values), $\boldsymbol{B}_1$ shows the background scene except for bounding boxes where 4 people were detected (bounding boxes are black since $\boldsymbol{B}_0$ is zero-valued). However, after 20 updates, as people have moved out, the black rectangles get filled in by image values from frames $\boldsymbol{I}_2, \boldsymbol{I}_3, ...$

4.2.3 Focus window selection

Having found the final ROI in camera's FOV (Figure 4·4), I could build a focus window around each component. However, individual connected components may be inaccurate and may contain an incomplete person (or incomplete several people). I could select a larger window but it is unclear how large. Instead, I propose to use

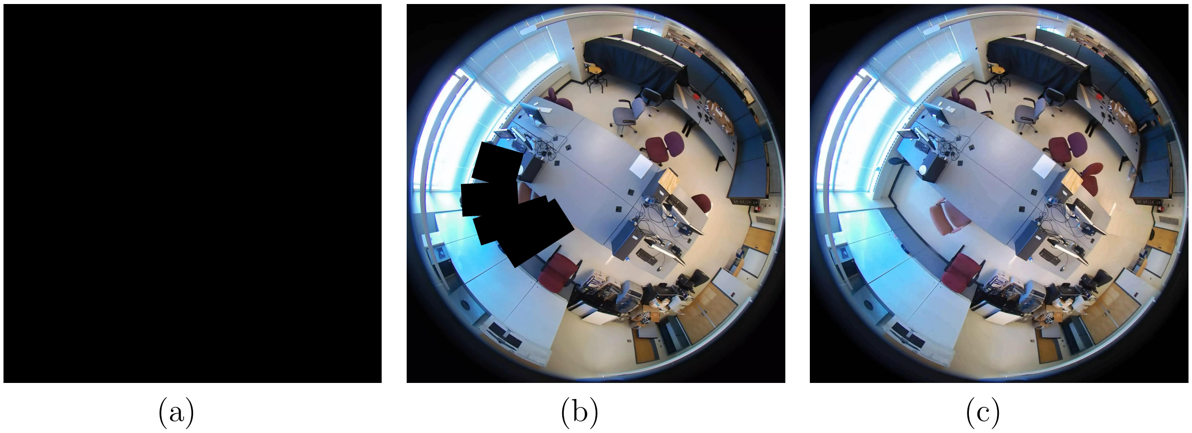


Figure 4·5: (a) Initial background model $\boldsymbol{B}_0$; (b) background model $\boldsymbol{B}_1$ after the first update; (c) background model $\boldsymbol{B}_{20}$ after 20 updates.

one or more of the 24 rotated windows used in the activity-blind method (Figure 3·3) and apply the same methodology to each window as in Chapter 3. An additional advantage of using large windows is that all connected components at a similar angular position are likely to be processed in one people-detection step.

In some cases, a connected component may be too large for one single window to fully contain it. In other cases, a connected component can be located off center and extend beyond the window. These two cases are challenging for people detection.

I propose the following algorithm to address these issues. In order to ensure full coverage of a connected component by a rotated window of width W and height H, I define the *central part* of this window as a rectangle of width $W_c < W$ and height H. A set of focus windows will be selected to cover all connected components as described below.

First, centroid C of a connected component (Figure 4·4(c)) is calculated. Let O denote the center of camera's FOV and $R_i, i = 1, ...24$ the center of each of the 24 rotated windows. Window number $k = \min_i \angle(\vec{OC}, \vec{OR}_i)$ is selected as the focus window for this connected component.

If the connected component exceeds the left boundary of the central part of the

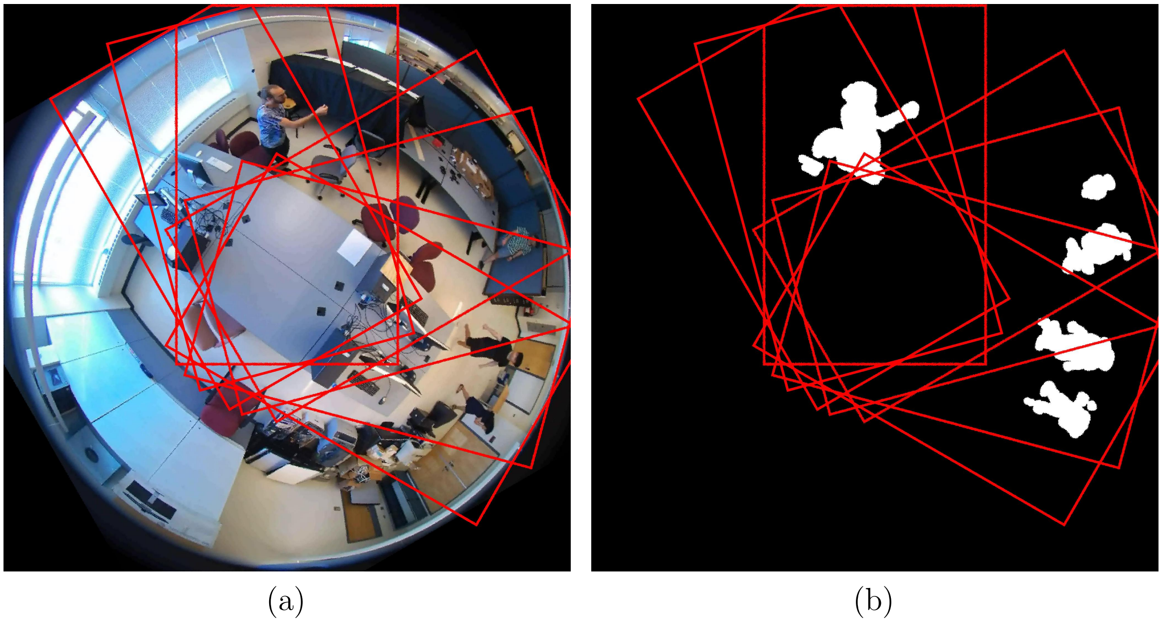


Figure 4·6: Focus windows for the current frame from Figure 4·3: (a) overlaid on the current frame; (b) overlaid on the connected components of final ROI.

focus window, the neighboring window counterclockwise is added to the final window selection. Then, the same check is performed for the newly-added window. The process is repeated until the connected component does not exceed the left boundary of the central part of the last newly-added window. A similar procedure is applied to the right boundary and neighboring windows in the clockwise direction.

The above steps are repeated for all remaining connected components. Then, people detection by YOLO v3, post-processing, reverse mapping and people counting are applied to all focus windows selected above (see block diagram in Figure 4·1). An example of selected focus windows for the background and input frames from Figure 4·3 are shown in Figure 4·6.

Chapter 5

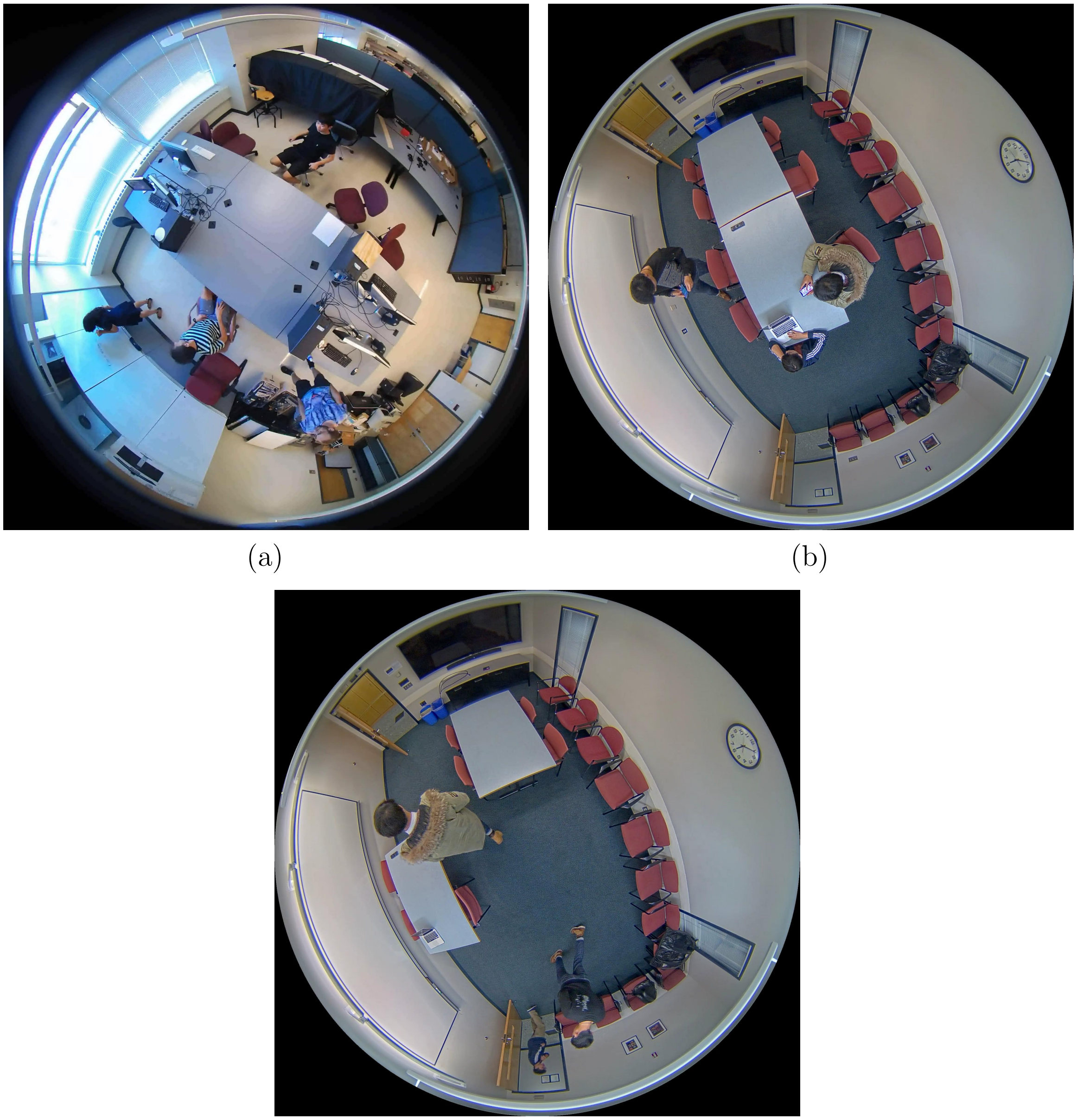
Experimental Results

5.1 Experimental setup

A series of experiments have been conducted to evaluate the proposed activity-blind method (AB) presented in Chapter 3 and activity-aware method (AA) presented in Chapter 4. First, experiments demonstrating the need for spatial outlier rejection and verification of detections (Figures 3·2 and 4·1) will be described. Secondly, a quantitative comparison between NMS-based and clustering-based merging of multiple detections will be presented. Then, the activity-aware and activity-blind methods will be compared in terms of people counting accuracy and computational efficiency. Finally, the activity-aware algorithm will be compared with a benchmark method.

In order to facilitate experimental evaluation, I collected one overhead, fisheye video in a lab and two videos in a conference room (Figure 5·1). The videos were captured by two Axis M3057 PLVE cameras (3,072×2,048-pixel resolution) installed on the ceiling. The three videos depict several people moving around in each space.

In all example frames in this section, blue bounding boxes are at the output of reverse mapping, while green ones are after merging of multiple detections (Figures 3·2 and 4·1). Since NMS-based merging selects one of the reverse-mapping results, the resulting green bounding box also belongs to the results of reverse mapping (it covers a blue bounding box). In contrast, clustering-based merging produces a bounding box whose location and size are averages of locations and sizes of all bounding boxes from the same cluster, respectively. Clearly, this means that the bounding box resulting



(a)

(b)

(c)

Figure 5·1: (a–c) Sample frames from three test videos.

from clustering-based merging does not belong to the results of reverse mapping (an additional green bounding box is created). After the verification step (second application of YOLO v3), red bounding boxes show the final detection result.

To quantitatively evaluate each method's performance, I manually labeled each computed result by identifying the following:

- True Positives (TPs):
  - people detected correctly with a single detection per person (as per subjective evaluation), or
  - people detected correctly but with multiple detections, i.e., if P detections (red bounding boxes) are associated with the same person (subjectively-evaluated overlap of a red bounding box with person's body), then 1 detection is deemed a true positive and $P - 1$ detections are considered to be false positives.
- False Positives (FPs):
  - objects detected as people by mistake (e.g., chair detected as a person), or
  - multiple detections of the same person, i.e., if P detections (red bounding boxes) are associated with the same person (subjectively-evaluated overlap of a red bounding box with person's body), then $P - 1$ detections are considered to be false positives and 1 detection is deemed a true positive.
- False Negatives (FNs): people missed during detection.

Three performance measures: *Recall*, *Precision*, and *F-score*, are calculated for each frame as follows:

$$Recall = \frac{TP}{TP + FN} \tag{5.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.3}$$

Then, each metric's mean μ and standard deviation σ are computed from single-frame metrics and reported in tables in this chapter.

All experiments were conducted on Intel Xeon 2.3GHz CPU and Tesla K80 GPU provided by Google Colaboratory. The computing times reported in this chapter are for this environment.

5.2 Evaluation of the proposed methods

5.2.1 Impact of spatial outlier rejection on people counting

As mentioned in Chapter 3, spatial outlier rejection (SOR) is needed to eliminate bounding boxes near the boundary of each rotated window since person's body may be only partially visible in this window. In order to demonstrate that this step is essential for high detection accuracy, I conducted experiments with and without SOR for both activity-blind method and activity-aware method, in each case using NMS-based merging of multiple detections. The results are shown in Table 5.1.

Table 5.1: People-counting performance of the activity-aware and activity-blind method with and without SOR

Method	TP	FP	FN	*Recall*		*Precision*		*F-score*	
				μ	σ	μ	σ	μ	σ
AA with SOR	504	14	43	0.92	0.16	0.98	0.07	0.94	0.11
AA without SOR	513	170	34	0.94	0.15	0.81	0.19	0.84	0.14
AB with SOR	519	71	28	0.95	0.12	0.91	0.14	0.92	0.10
AB without SOR	526	659	21	0.96	0.10	0.48	0.16	0.62	0.12

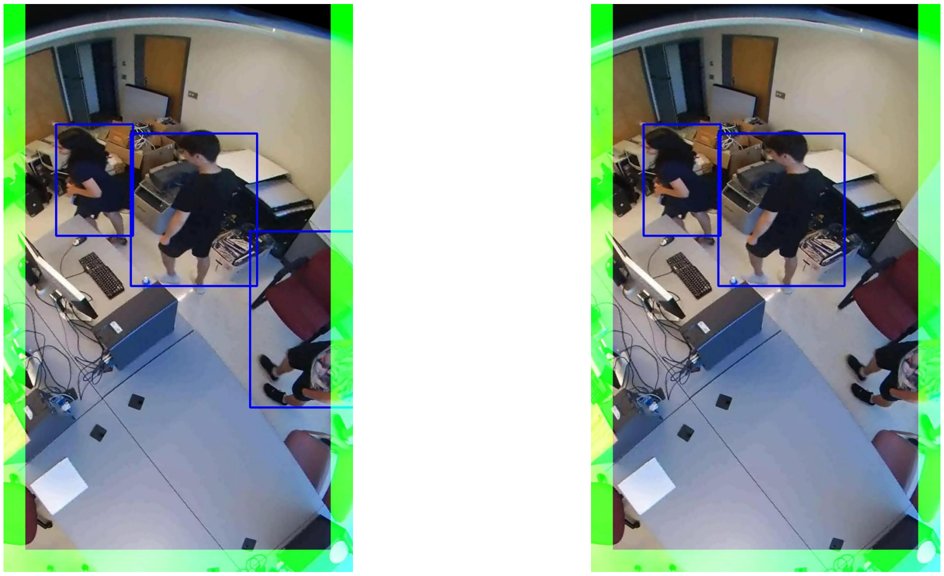
It is clear that *Precision* and *F-score* for both methods without spatial outlier rejection decrease significantly, which is caused by spurious FP detections. One example of an FP occurrence is shown in Figure 5·2. Note the partially-visible person at the right edge of the focus window in Figure 5·2(a) - no spatial outlier rejection is applied and the person is counted (likely, because the bounding box is much larger than same-person's bounding box in the neighboring rotated window and merging of multiple detections is not effective). However, when SOR is applied (the detected bounding box overlaps the green margin area) the bounding box is rejected (Figure 5·2(b)) and the person is not counted in this window.

The activity-aware method without SOR performs better than the activity-blind method without SOR, primarily because there are fewer focus windows in the activity-aware method if only a few people are sparsely scattered. In this case, it is less likely on average that people are close to focus window boundary and thus partially visible. However, if the number of people increases, and thus people are closer to each other, the number of partially visible bodies will likely increase as well; FP will increase and *Precision* and *F-score* will decrease. Clearly, spatial outlier rejection is a necessary step in people detection using overlapping windows.

Figure 5·3 shows sample frames produced by the activity-blind and activity-aware method with and without SOR. While without SOR many bounding boxes that contain part of a person's body cannot be merged with larger bounding boxes for the same person due to size difference, this is not the case when SOR is applied since partial-body bounding boxes at focus window margin are rejected resulting in reduction of false positives.

5.2.2 Impact of person-detection verification on people counting

Person-detection verification (PDV) was proposed in Chapter 3 to reduce false positives when counting people. While YOLO v3 is applied to a 1,300×800-pixel focus window in the earlier stages, at the output of the merging step much smaller bounding boxes are available around likely person detections. A second application of YOLO v3



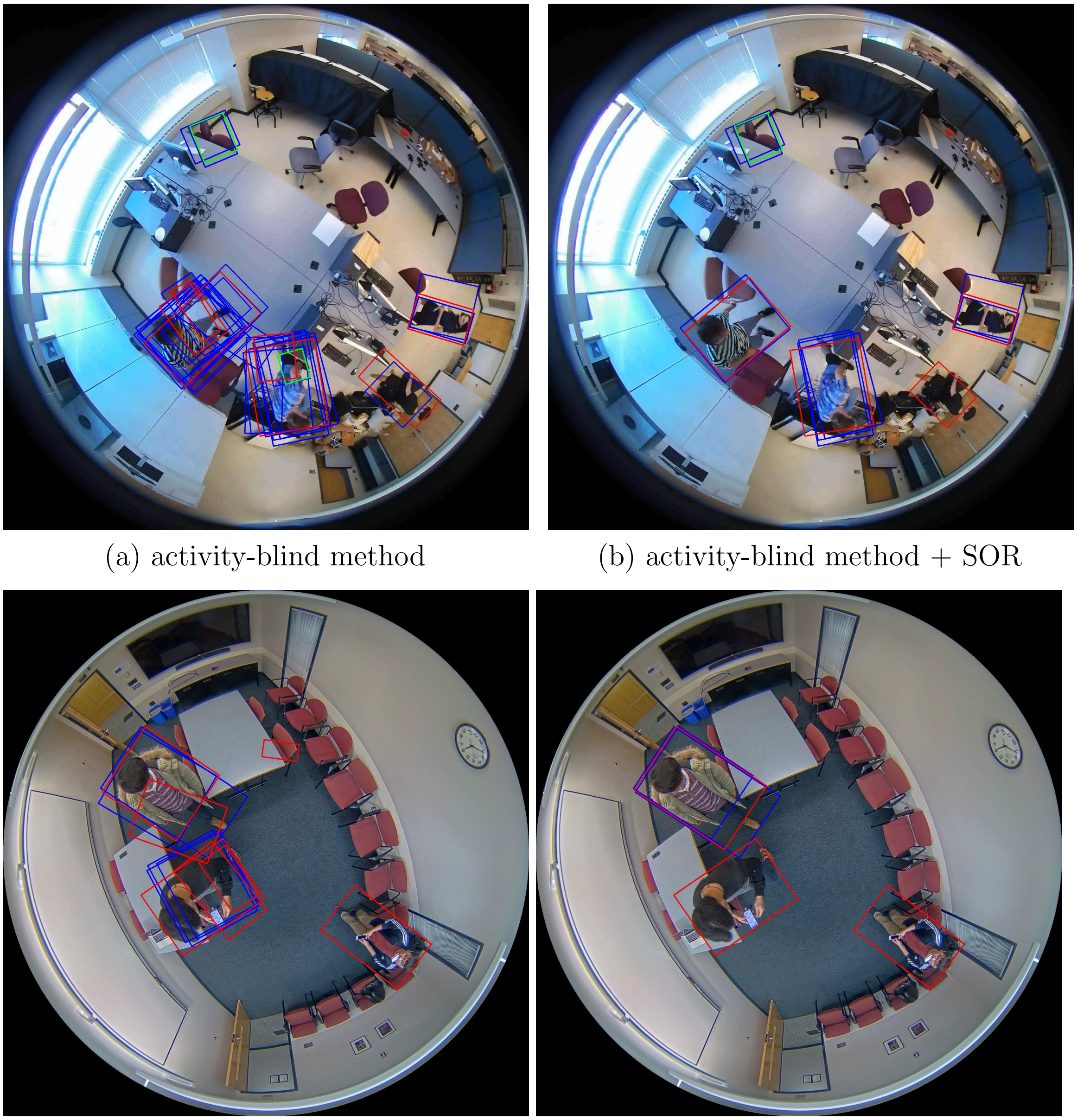
(a) (b)

Figure 5·2: The impact of spatial outlier rejection (SOR) in a focus window (margin area is in green): (a) without SOR, a partially-visible person is counted; (b) after applying SOR, a partially visible person is excluded.

to each bounding box is expected to confirm/reject person detection with improved reliability due to focus on a much smaller window than in the first YOLO v3 application. In order to verify this, I have performed experiments with and without PDV for both activity-blind and activity-aware method using NMS-based merging of multiple detections.

Table 5.2: People-counting performance of the activity-aware and activity-blind method with and without PDV

Method	TP	FP	FN	*Recall*		*Precision*		*F-score*	
				μ	σ	μ	σ	μ	σ
AA with PDV	504	14	43	0.92	0.16	0.98	0.07	0.94	0.11
AA without PDV	516	20	31	0.94	0.13	0.97	0.09	0.95	0.09
AB with PDV	519	71	28	0.95	0.12	0.91	0.14	0.92	0.10
AB without PDV	532	170	15	0.98	0.08	0.79	0.16	0.86	0.11



(a) activity-blind method

(b) activity-blind method + SOR

(c) activity-aware method

(d) activity-aware method + SOR

Figure 5·3: Sample frames produced without SOR by: (a) activity-blind method; (c) activity-aware method. Many bounding boxes containing only part of a person's body cannot be merged with those containing a complete person due to excessive difference in size, thus causing false positives. Sample frames produced with SOR by: (b) activity-blind method; (d) activity-aware method. By using SOR, bounding boxes containing part of a person are eliminated, so the correct number of people is produced.

As can be seen in Table 5.2, PDV has a strong influence on people-counting performance of the activity-blind method as both *Precision* and *F-score* are substantially reduced in the absence of PDV (since the FP value dramatically increases while the TP value increases only a little). Examples of a miss (reduction of TP value) and a simultaneous correct rejection of a false positive detection (chair), both due to the application of PDV, are shown in Figure 5·4.

As for the activity-aware method, however, PDV slightly reduces both the *F-score* and *Recall* because the number of false negatives is already small, and thus even few erroneous rejections of person detections can reduce both metrics. The underlying reason for this phenomenon is that the ROI extraction step excludes areas of little/no change, and thus people detectors in the activity-aware method are not likely to be applied to the areas where most false positives occur in the activity-blind method. However, when more people are present in camera's FOV, false positives are more likely and the PDV step should be beneficial.

5.2.3 Performance of person-detection verification on its own

While Table 5.2 shows the impact of PDV on the accuracy of the overall people counting, it does not capture the performance of PDV itself. As we have stated in Section 5.1, if there are P final person detections (red bounding boxes) associated with the same person (subjectively-evaluated overlap of a red bounding box with person's body), then $P - 1$ detections are considered to be false positives and 1 detection is deemed a true positive. Although these false positives justly contribute to the lowering of the *Precision* metric in people counting, they are still correct detections from the standpoint of PDV (indeed, a person was present in a bounding box). Therefore, in order to assess PDV performance on its own, I considered the following 4 cases of input versus decision:

- Correct Acceptance (CA): input bounding box with a person is accepted,

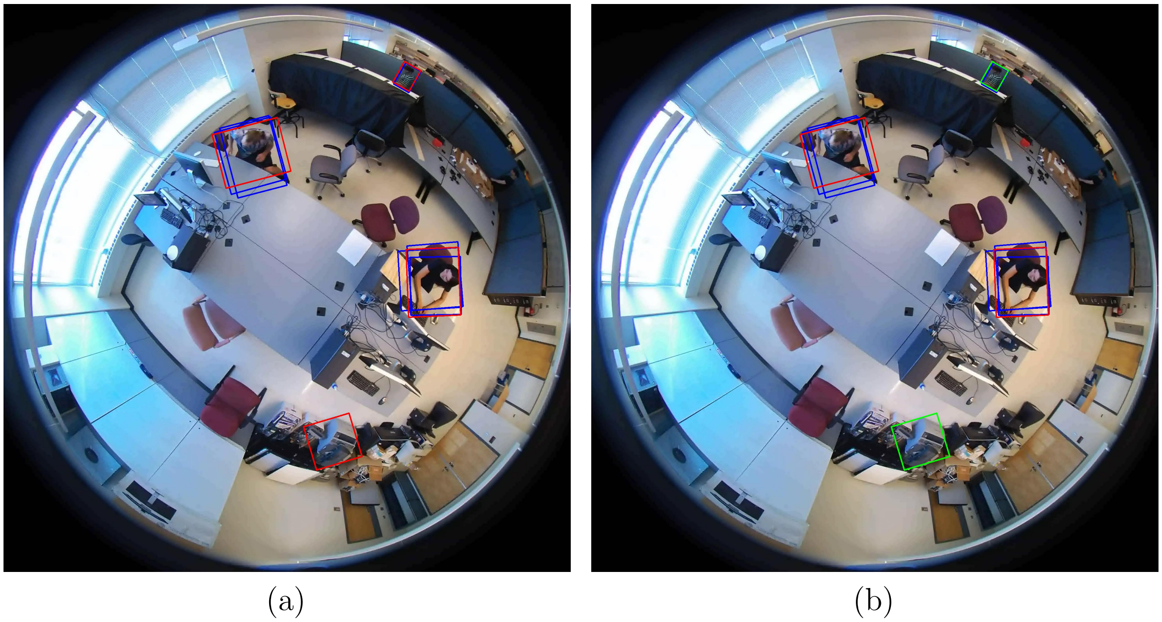


Figure 5·4: Sample detection result obtained by the activity-blind method: (a) without PDV (all detections from the first application of YOLO v3 are accepted in the final output); (b) with PDV (a far-away person by the partition is rejected by mistake, while a falsely-detected chair at the bottom is correctly rejected).

- Wrong Acceptance (WA): input bounding box with a non-person object is accepted,
- Correct Rejection (CR): input bounding box with a non-person object is rejected,
- Wrong Rejection (WR): input bounding box with a person is rejected.

Clearly, PDV produces bounding boxes claimed to contain people. Those that indeed contain a person are correct decisions and those that do not (contain another object claimed to be a person) are errors. The counts of both types of acceptances and rejections for the activity-blind and activity-aware algorithms on 3 videos tested are shown in Table 5.3. Ideally, one would like PDV to produce the fewest wrong acceptances and wrong rejections. Clearly, PDV applied to the output of the activity-aware method wrongly accepts only 7 bounding boxes without a person and erroneously rejects 13 bounding boxes with a person. However, it correctly rejects only 5 bounding boxes without a person. This suggests that the activity-aware method already performs well and any potential improvement due to PDV is small. To the contrary, in the case of the activity-blind method PDV correctly rejects 97 bounding boxes without a person but wrongly accepts 53 such boxes. It also erroneously rejects 15 bounding boxes with a person. Since the activity-blind method produces many spurious person detections by applying YOLO to 24 focus windows, many incorrect person detections result and most of them are rejected by PDV.

Table 5.3: Performance of person detection verification for activity-aware and activity-blind methods after NMS-based merging

	CA	WA	CR	WR
PDV in AA	511	7	5	13
PDV in AB	537	53	97	15

Among the total of 1,238 detections fed by both algorithms into the verification step (1,076 correct person detections and 162 incorrect person detections produced by the merging step), 92.90% are correctly accepted or rejected. In more detail, 97.40% of correct person detections at input are correctly accepted by PDV, and 62.96% of incorrect person detections at input are correctly rejected.

These results indicate that the proposed person-detection verification step performs well in confirming correct person detections, and only a few correct detections are erroneously rejected. As for rejecting incorrect person detections, there exists significant room for improvement.

5.2.4 Impact of NMS-based and clustering-based merging of multiple detections on people counting

The step of merging multiple detections using either NMS-based approach or clustering is evaluated in terms of performance in Table 5.4 and in terms computing time in Table 5.5 (column: "Merging of detections"). Since the number of bounding boxes undergoing merging may be quite dramatically different in activity-aware and activity-blind method, they are compared separately.

Table 5.4: People-counting performance of the activity-blind and activity-aware methods using either NMS-based or clustering-based merging of multiple detections.

Method	TP	FP	FN	*Recall*		*Precision*		*F-score*	
				μ	σ	μ	σ	μ	σ
AA with NMS	504	14	43	0.92	0.16	0.98	0.07	0.94	0.11
AA with clustering	502	15	45	0.92	0.16	0.98	0.07	0.94	0.10
AB with NMS	519	71	28	0.95	0.12	0.91	0.14	0.92	0.10
AB with clustering	516	55	31	0.95	0.12	0.92	0.13	0.93	0.11

It is clear from Table 5.4, that people-counting performance is similar when applying either NMS- or clustering-based merging, whether in the activity-blind or activity-aware method. However, clustering-based merging is much slower than NMS-based clustering. This is caused by the fact that in order to learn the number of clusters, K-means clustering needs to be applied a number of times to feature vectors for all bounding boxes resulting from the reverse mapping. Furthermore, it is time-consuming to form a feature vector as a 10-bin normalized histogram of grayscale values needs to be computed in each bounding box.

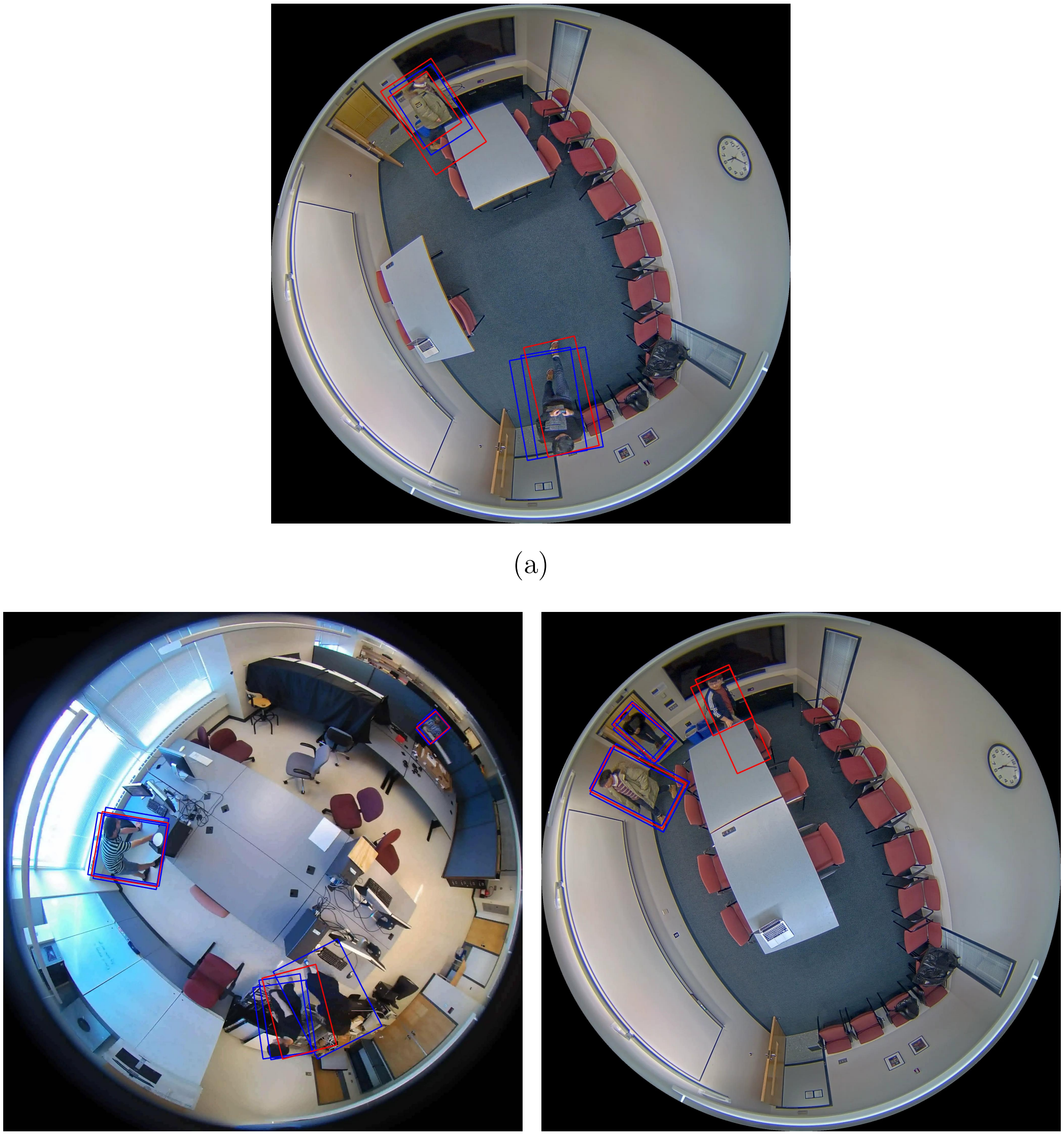
As mentioned in Chapter 3, people counting using NMS-based merging may not be able to merge bounding boxes with significant difference in size, and test results support this statement (Figure 5·5(a)). However, typically NMS-based merging does not need to deal with this problem because of reliable people detection (tight bounding boxes around human bodies).

As for clustering-based merging, its performance is more limited. First, the size and location of a bounding box produced in this case is an average of sizes and locations of all bounding boxes in the same cluster. Therefore, this box is not an actual outcome of people detection and may contain no person. This strategy is likely to generate more false negatives because such a bounding box may be located between two people (with partial overlap of each person) and will be easily rejected in the subsequent verification step (Figure 5·5(b)). Secondly, when there is a great difference in size, two bounding boxes may cover areas with different grayscale content, although they include the same person. Clearly, these bounding boxes cannot be merged together as well (Figure 5·5(c)).

5.2.5 Comparison of activity-blind and activity-aware method

In this section, I compare the activity-blind method with the activity-aware method in terms of performance/complexity trade-off. Table 5.4 shows the people-counting performance for both methods with either NMS-based or clustering-based merging of multiple detections, while Table 5.5 shows the mean computing time μ and standard deviation σ for various stages of processing and for each full algorithm, computed over the three test videos used in this study.

It is clear from Table 5.5 that the activity-aware method has a significant advantage over the activity-blind method in terms of computing time. However, its people-counting performance slightly lags behind that of the activity-blind method. It can be seen from Table 5.4 that the activity-aware method has fewer true positives than the activity-blind method and, correspondingly, more false negatives. This is caused by the fact that most people appear in a single focus window in the activity-aware method (focus window is constructed around ROI). However, even if a person



(a)

(b)

(c)

Figure 5·5: (a) Example of NMS-based clustering fails to merge two red bounding boxes at the top that have significantly different sizes and yet contain the same person. (b) Example of clustering-based method wrongly merging two people with similar appearance (black clothing) and standing close to each other (red bounding box at the bottom), thus generating a bounding box with two incomplete bodies; (c) Example of clustering-based method unable to merge two bounding boxes due to large difference in grayscale value (person and table).

Table 5.5: Computing time (in seconds) comparison of the activity-blind and activity-aware method with either NMS-based or clustering-based merging of multiple detections.

Method	ROI extrac.		Detection in focus windows		Merging of detections		PDV		Full algorithm	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
AA+NMS	0.74	0.31	3.50	2.18	1.17	0.50	1.52	0.37	6.95	2.55
AA+clust.	0.74	0.32	3.46	2.10	1.47	0.75	1.52	0.37	7.19	2.71
AB+NMS	-	-	14.20	0.32	1.53	0.57	1.67	0.50	17.41	1.08
AB+clust.	-	-	14.17	0.21	2.92	2.51	1.72	0.55	18.82	2.74

is in window's center, he/she may not be upright and detection by YOLO v3 may fail. In contrast, in the activity-blind method a person will appear in several neighboring focus windows, in each at a slightly different angle and in some with fully-visible body, so that there are more chances for detection. Therefore, the activity-blind method outperforms the activity-aware method in terms of TP and FN and, consequently, has higher *Recall*. Two sample results are shown in Figure 5·6.

While the activity-blind method is more reliable in finding a person (higher TP, lower FN, and thus higher *Recall*), the activity-aware method performs better in terms of FP (spurious detections) and *Precision*. In consequence, both methods have similar *F-score*, but the activity-aware method is much faster, and therefore it is difficult to choose between them. If the detection speed is critical and some misses can be tolerated, then the activity-aware method would be more appropriate. If the number of true detections and as few misses as possible are the most important factors, then the activity-blind method is more suitable at the cost of some spurious detections.

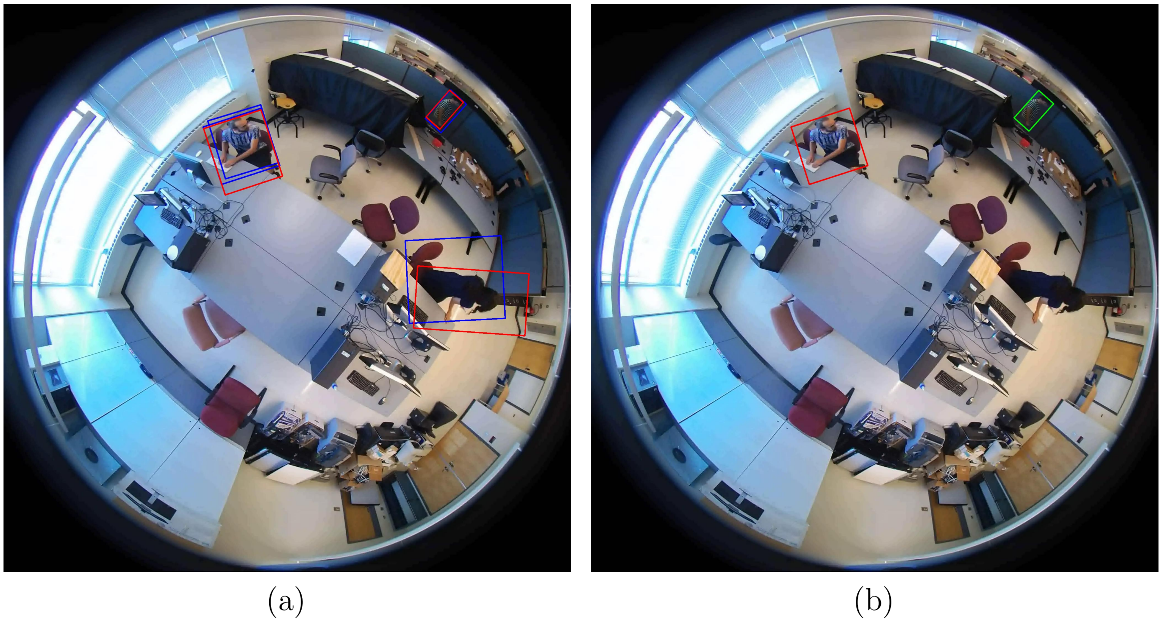


Figure 5·6: (a) The activity-blind method correctly detects a person in right-center in two neighboring focus windows; (b) The activity-aware method misses the same person.

5.3 Comparison with a benchmark method

As a benchmark method, I selected an approach in which YOLO v3 is well-matched to the appearance of most occupants. The main idea is to transform a fisheye image into an almost side-view image by means of dewarping (Courbon et al., 2012). Dewarping transforms a round FOV of a fisheye image into a rectangular FOV of a panoramic image in which people standing away from the fisheye camera look undistorted and upright (Figure 5·7). With people standing upright (except those under the camera, who will be severely distorted), YOLO v3 should perform very well.

I apply YOLO v3 to the whole image without any cropping, decimation or interpolation. However, I zero-pad the dewarped image to produce a 2,176×2,176-pixel image for processing by YOLO v3. In post processing, I apply confidence thresholding and NMS with the same threshold values ($\gamma = 0.3, \rho = 0.4$) as in both methods I had proposed.

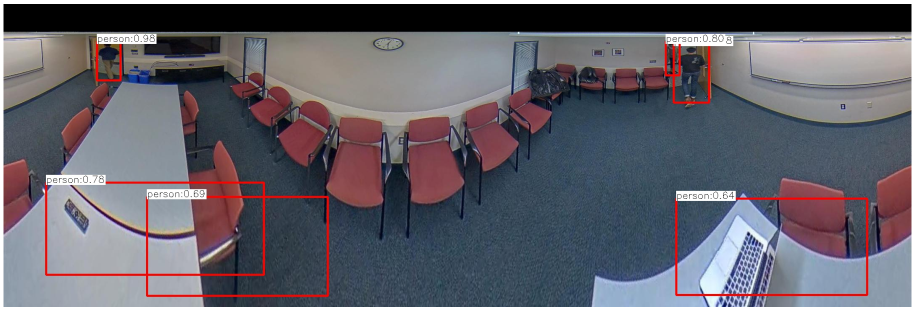


Figure 5·7: Dewarped fisheye image (2,160×720 pixels) and sample detection results produced by the benchmark method.

Table 5.6 shows performance metrics of the benchmark algorithm in comparison to the activity-aware and activity-blind methods both implemented with NMS-based merging of multiple detections. In FOV periphery, the benchmark method performs well as can be seen at the top of Figure 5·7. Across all three test videos there are few misses (FNs) in this area. As already mentioned, human bodies are upright and look undistorted after dewarping, thus contributing to good YOLO v3 performance. Benchmark method's relatively high score in terms of *Recall* supports this statement. Most false negatives for the benchmark method concentrate at the bottom of the panoramic image (center of fisheye FOV), where a strong distortion occurs due to dewrapping. Furthermore, there is a huge number of false positives in this area (see, for example, the bottom of Figure 5·7) resulting in very low *Precision.* In addition to dewarping distortion, the lack of person-detection verification (making the method a one-shot try) is a contributor to its weak performance.

Table 5.6: People-counting performance of NMS-based activity-blind and activity-aware methods *versus* the benchmark method.

Method	TP	FP	FN	*Recall*		*Precision*		*F-score*		*MAE*	
				μ	σ	μ	σ	μ	σ	*fr.*	*pe.*
AA with NMS	504	14	43	0.92	0.16	0.98	0.07	0.94	0.11	0.37	0.12
AB with NMS	519	71	28	0.95	0.12	0.91	0.14	0.92	0.10	0.37	0.12
Benchmark	473	916	74	0.87	0.19	0.36	0.11	0.50	0.12	4.90	1.54

In addition to performance metrics used earlier, I calculated the Mean-Absolute Error (MAE) per frame and also per person using equations (5.4) and (5.5), respectively. N is the total number of frames in the three test videos, GT_i is the ground-truth number of people in frame number i, while FP_i and TP_i are the numbers of False Positives and True Positives in frame number i, respectively.

$$MAE_{frame} = \frac{1}{N} \sum_{i=0}^{N-1} |TP_i + FP_i - GT_i| \tag{5.4}$$

$$MAE_{person} = \frac{\sum_{i=0}^{N-1} |TP_i + FP_i - GT_i|}{\sum_{i=0}^{N-1} GT_i} \tag{5.5}$$

The MAE values shown in the last two columns of Table 5.6 confirm a superior performance of the proposed algorithms compared to the benchmark (an order of magnitude lower MAE).

Finally, a normalized histogram of people-counting error $TP_i + FP_i - GT_i$ is shown in Figure 5·8 for each method. Clearly, the spread of histograms for the proposed methods is much smaller than that for the benchmark method. The activity-aware method undercounts by 1 fairly often, but overcounts by 1 much less often. This is due to the fact that, occasionally, only one focus window is selected to cover a changed area (obtained from background subtraction) so the chance of a miss is not negligible. The miss chance is reduced when multiple focus windows overlap the changed area. The method's absolute error rarely exceeds one. In case of the activity-blind method, undercounts and overcounts by 1 are equally likely since every person is captured by at least 2 focus windows (often more), thus reducing the chance of a miss. Again a larger departure from the true count is rare. It is a different story for the benchmark method as it heavily overcounts as is also obvious from Table 5.6 (916 false positives).

Clearly, the proposed activity-blind and activity-aware methods handily outperform the benchmark and suggest that this is a good strategy for applying CNN-based

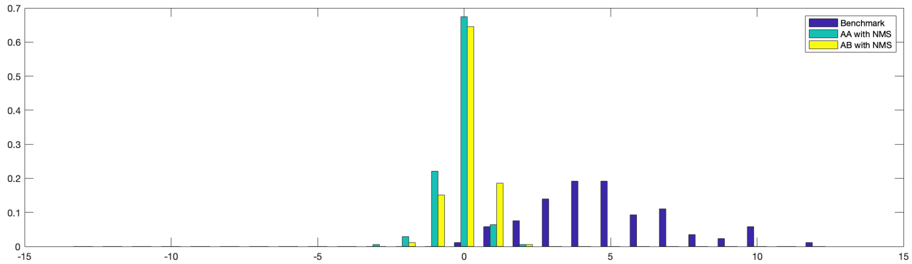


Figure 5·8: Normalized histogram of the people-counting error per frame ($TP_i + FP_i - GT_i$) for the benchmark, AA with NMS, and AB with NMS methods.

people detection trained on standard images (from side-view, pinhole-lens cameras) to overhead, fisheye images. A remaining weakness of the proposed methods is the fact that for people close to the center of FOV, whose head and shoulders are the only visible parts of the body, misses easily occur, as shown in Figure 5·9. To address this weakness, one possible approach is to train another CNN on images with this scenario and apply it only to FOV center, or to train only one CNN model on a diverse set of training data captured by overhead fisheye cameras.

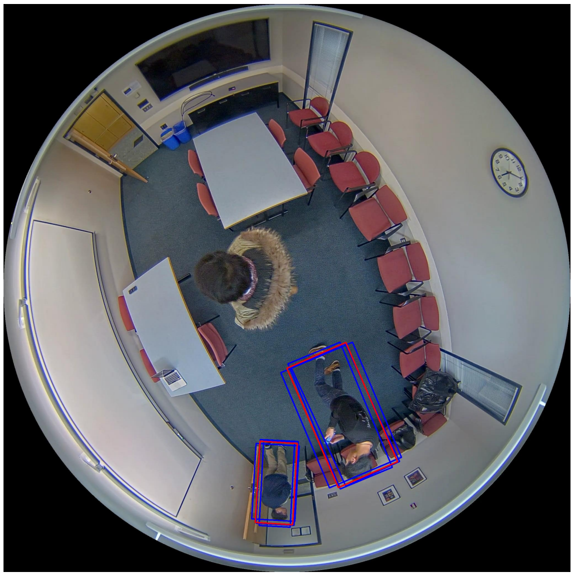


Figure 5·9: Example of a missed detection by the activity-blind method when a person stands directly under a fisheye camera.

Chapter 6

Conclusions

In order to supply HVAC system with a room's occupancy estimate, and thus facilitate energy savings, I developed an approach to people counting using an overhead, fisheye camera that leverages YOLO v3, a top-performing CNN-based object detection method. Rather than re-training YOLO v3 with overhead, fisheye images, I proposed two approaches, activity-blind and activity-aware, that use the original YOLO v3 model trained on side-mounted, standard-lens cameras. The basis for this approach is the following similarity between overhead, fisheye images and side-view, standard-lens images: people standing upright away from the camera are always vertical to the ground in side-view images and always aligned with the radial direction in fisheye images.

The activity-blind method rotates a fisheye image by 15° at a time, and applies people detector to a focus window extracted from the central upper part of the image, where a person appears upright (or approximately upright). This results in 24 focus windows to which a people detector is applied. In contrast, the activity-aware method begins by extracting a region of interest by means of background subtraction followed by selecting only some focus windows (out of the 24 windows) where people are likely to appear. This leads to a reduced computational complexity of the activity-aware method compared to the activity-blind method.

To deal with multiple detections produced by YOLO v3 in rotated, and largely overlapping, focus windows, I proposed spatial outlier rejection and merging of multiple detections (NMS-based and clustering-based) as a post-processing step. The final step I proposed is the verification of merged detections by applying YOLO v3 second time but only around the detected bounding boxes. This reduces false positive detections since the area considered is much smaller than in the first application of YOLO v3.

I verified the usefulness of spatial outlier rejection and person-detection verification experimentally. Spatial outlier rejection helped significantly reduce false positives with only a small reduction in true positives. As for person-detection verification, it also reduced false positives thus helping improve the F-score, although occasionally it would reject a true positive by mistake and too often retain a false positive. Although this seems to indicate that the idea of verification is useful, a better verification strategy is needed to maximize the number of true positive detections.

I compared experimentally the NMS-based and clustering-based merging of multiple detections. It turns out that clustering-based merging does not offer expected advantages: it is more complex computationally than the NMS-based merging, and tends to merge multiple detections into fewer clusters. Clustering-based merging has difficulty with merging different-size bounding boxes containing the same person, primarily because of the histogram-based feature vector that may capture appearance of a person in a smaller bounding box and that of the person and the background in a larger bounding box. Clearly, clustering-based merging needs to be further investigated. First, a better feature vector than grayscale histogram needs to be considered. Secondly, a better strategy to find the optimal number of clusters is needed.

An experimental comparison of the computational complexity showed that the activity-aware method significantly reduces detection time for the three test videos used, while producing fewer false positives than the activity-blind method. However, the activity-blind method produces more true positives and fewer false negatives (misses), which may be important for maintaining air quality – in the case of many misses, HVAC system may provide too little air.

In the final set of experiments, I compared both proposed methods with a benchmark method. Both the activity-blind and activity-aware method significantly outperformed the benchmark method in terms of *Recall*, *Precision*, and *F-score* values. The proposed methods proved to be capable of at least as many true-positive detections as the benchmark method, while generating only a fraction of false-positive detections.

In conclusion, I proposed a methodology for people counting from overhead, fisheye images using a CNN object detector trained on side-view, standard-lens images. I developed two methods using this methodology, and both of them proved reliable with up to 0.95 recall, 0.98 precision and 0.94 F-score on three test videos. With this performance, both methods are viable candidates for practical indoor occupancy estimation.

In the future, several improvements to the proposed methodology can be envisaged. First, person-detection verification needs further investigation since the current approach too often accepts false person detections. Second, a better clustering-based method is needed for merging multiple detections. The current use of a 10-bin grayscale histogram and bounding-box location as the feature vector has proved lacking in terms of performance. One possible solution is to include more features. Another one is to find a better distance metric to compare grayscale patterns in bounding boxes so that interference from spurious part of a bounding box, such as the background, has less of an impact.

References

Chiang, A.-T. and Wang, Y. (2014). Human detection in fish-eye images using hog-based detectors over rotated windows. In *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6.

Courbon, J., Mezouar, Y., and Martinet, P. (2012). Evaluation of the unified model of the sphere for fisheye cameras in robotic applications. *Advanced Robotics*, 26:947–967.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *International Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society.

del Blanco, C. and Carballeira, P. (2016). The PIROPO database (people in indoor rooms with perspective and omnidirectional cameras). http://sites.google.com/site/piropodatabase.

Demirz, B. E., Ari, ., Erolu, O., Salah, A. A., and Akarun, L. (2012). Feature-based tracking on a multi-omnidirectional camera dataset. In *2012 5th International Symposium on Communications, Control and Signal Processing*, pages 1–5.

Dollár, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

Krams, O. and Kiryati, N. (2017). People detection in top-view fisheye imaging. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Nguyen, V. T., Nguyen, T. B., and Chung, S. (2016). Convnets and agmm based real-time human detection under fisheye camera for embedded surveillance. In *2016*

International Conference on Information and Communication Technology Convergence (ICTC), pages 840–845.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Redmon, J. and Farhadi, A. (2016). YOLO9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*.

Saito, M., Kitaguchi, K., Kimura, G., and Hashimoto, M. (2010). People detection and tracking from fish-eye image based on probabilistic appearance model. In *International Symposium on Communications, Control and Signal Processing (ISCCSP)*. IEEE.

Seidel, R., Apitzsch, A., and Hirtz, G. (2018). Improved person detection on omnidirectional images with non-maxima supression. *arXiv:1805.08503*.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252 Vol. 2.

Wang, J.-M., Chen, S.-W., Cherng, S., and Fuh, C.-S. (2007). People counting using fisheye camera. In *Proceedings of the Conference on Computer Vision, Graphics and ImageProcessing*, pages 808–813.

CURRICULUM VITAE

Shengye Li

Education

- **MS** in Electrical Engineering
  Boston University (BU), 2019

- **B.Eng.** in Electronic Science and Technology
  University of Electronic Science and Technology of China (UESTC)

Research and Projects

- **People Counting Using Fisheye Camera, BU**
  As a member of Prof. Konrad's research group, worked on occupancy estimation using overhead, fisheye cameras.

  - Proposed algorithms that leverage deep-learning object detection models trained on side-view, standard-lens images to detect people in overhead, fisheye images.
  - Proposed a method to improve the performance of people detection using strongly overlapping windows.
  - Proposed a dynamic background update mechanism to obtain background model robust to illumination variations.

- **Photo Enhancement through Flash/No-Flash Exposure, BU**
  Implemented an image enhancement algorithm using a pair of photos of the same view taken with and without a flash.

  - Developed a detail-preserving de-noising module and a detail-inheriting module to create a new image with benefits of both flash and no-flash photos.
  - Developed modules for correcting red-eye effect and poor white balance both due to flash photography.

- **Image Foreground/Background Segmentation, BU**
  Developed an application to segment images into background and foreground layers.

  - Implemented image segmentation using GMM clustering and network-flow method achieved by push-relabel algorithm.

  - Proposed and implemented an OVO segmentation method to segment multiple layers.

- **Smart Wheelchair, BU**
  Designed a smart robot-arm system to identify and automatically press a handicap button (team project).

  - Proposed and coded an algorithm to identify and spatially locate target button using TensorFlow and Google Cloud Vision API, achieving precision of 86.7%.
  - Designed and produced a robot arm with 5 degrees of freedom and wrote programming code to control it.

- **Human Health Monitoring System, UESTC**
  Designed a smartphone-based WBAN system model for real-time EEG and ECG signal monitoring.

  - Developed corresponding Android application to collect and analyze data from measuring nodes, and to send alert message through Multimedia Message Service.
  - Established system's energy-consuming model, introduced energy harvesting mechanism, and finally proposed and tested a self-maintained energy model.

- **Facial Recognition based on Fractional Time-Frequent Features, UESTC**
  Developed an application for facial recognition based on fractional time-frequency features (team project).

  - Designed and implemented a pre-processing block to calibrate input images.
  - Designed a classifier to recognize faces based on extracted features.