Centroid human tracking via oriented detection in overhead fisheye sequences

Olfa Haggui, Hamza Bayd, Baptiste Magnier

Centroid human tracking via oriented detection in overhead fisheye sequences

Olfa Haggui[1] · Hamza Bayd[1] · Baptiste Magnier[1]

Abstract

Pedestrian tracking is highly relevant to the understanding of static and moving scenes in video sequences. The increasing demand for people's safety and security has resulted in more research on intelligent visual surveillance in a wide range of applications, such as moving human detection and tracking. With the great success of deep learning methods, researchers decided to switch from traditional methods based on hand-crafted feature extractors to recent deep-learning-based techniques in order to detect and track people. In this work, the topic of person detection using a Top-view moving fisheye camera is addressed using a deep learning detector combined with a centroid technique in order to track a selected person. Although the fisheye camera is a useful tool for video monitoring, most object detection techniques use classical perspective cameras, with (or without) deep learning. However, due to the distortions of fisheye images, we expect to have higher requirements and challenges on pedestrian detection using this type of device. In this paper, an end-to-end people detection learning method is proposed; it is based on a YOLOv3 detector that detects people using oriented bounding boxes. The proposed model customizes the traditional YOLOv3 for the detection of oriented bounding boxes, by regressing the angle of each bounding box using a periodic loss function. With rotation bounding box prediction, the new approach is efficient, reaching 98.1% of true detection. This detection model is combined with a centroid tracker in order to track a single person by identifying the trajectory, estimated angle of rotation and target distance. Finally, the proposed method is evaluated on a new available dataset where rotated bounding boxes represent annotations from several fisheye videos.

Keywords Human detection · Tracking · Moving fisheye camera · Deep learning · YOLOv3 · Centroid tracker

1 Introduction

In recent years, significant progress has been made in computer vision regarding people detection and tracking challenges, notably with advances in network technology. Within this context, the typical cameras used in visual surveillance include perspective and fisheye cameras. Regrettably, most of the existing research uses perspective cameras, as they generate views similar to human vision with, in addition, small image distortions. However, the main disadvantage of perspective cameras is their limited field of view. Therefore, the automatic use of algorithms for standard cameras is not directly applicable to fisheye images, as most methods in computer vision focus on narrow field-of-view cameras with mild radial distortion [17].

People detection and tracking via video frames captured by fisheye cameras have received massive attention due to a certain number of advantages in visual surveillance application such as the broad field of view. A fisheye lens enables images to be acquired with a viewing angle of approximately 180 degrees by a single camera in order to create a panoramic view of the surrounding. However, the major challenge is to take into consideration the radical distortions obtained in the image. In this context, pedestrians in a fisheye image appear in different shapes, sizes and at various orientations, such as upright, upside down, horizontal or diagonal. Unfortunately, most of the existing people detection algorithms are designed for standard/perspective camera images where people appear upright. This paper focuses on the problem of people detec-

✉ Olfa Haggui
olfa.haggui@mines-ales.fr

Hamza Bayd
hamza.bayd@mines-ales.fr

Baptiste Magnier
Baptiste.Magnier@mines-ales.fr

[1] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France

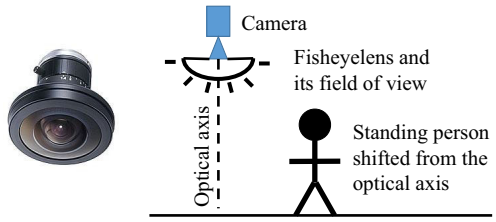


Fig. 1 Fisheye camera. On the left: fisheye lens used in our experiments: 2/3 "Format C-Mount fisheye lens 1.8 mm FL, with a horizontal field of view, 1/2" sensor for 185°. On the right: diagram of the experimental protocol

tion from video sequences recorded by top-view moving fisheye cameras, as represented in Fig. 1, right. Over the past decade, a significant improvement has been witnessed with the help of traditional handcrafted features and models based on end-to-end learning. Among traditional people detection algorithms, the most popular ones for pedestrian detection concern HOG (Histogram of Oriented Gradients) and ACFs (Aggregate Channel Features); they have been used with overhead fisheye images. For example, in [18], a human detection algorithm based on the histogram of HOG features and an SVM (Support Vector Machine) classifier are combined after rotating each search window on a radial line to the vertical reference line. Another method in [5] is based on HOG and LBP (Local Binary Pattern) features followed by a SVM classifier to model people as upright cylinders and derive a series of elliptic detection masks whose size diminishes with the distance from the image center. In [3], ACFs are trained on side-view, standard lens images for pedestrian detection without unwrapping a fisheye image into a panoramic image.

In the context of tracking with a fisheye camera, most modern trackers follow the tracking-by-detection paradigm. In fact, an off-the-shelf object detector first extracts all objects in each individual frame. Although many feature-based tracking methods have been proposed and experiments have proved their robustness and effectiveness, unfortunately, none of them are used for fisheye tracking and a more powerful feature representation is needed for fisheye target deformation. However, various traditional methods have been used to track people without the need for a detection phase. These methods are usually based on correlation filter tracking [53,54,56] (see also details in [32]), which attempt to solve the interference of fisheye deformation in target tracking since target features have been proved to be the most important factor for improving tracking performance using a feature integration method. In order to meet real-time requirements, works based on a particle filter tracking algorithm [51,52] have been proposed; they integrate a best-view selection strategy to ensure tracking consistency across single and/or multiple fisheye cameras.

Recently, with the advent of Deep Learning, numerous benchmarks and datasets have been created in order to train and evaluate people detection and tracking algorithms with high accuracy and in real time. Some algorithms are based on classification work in two stages. First, the Regions of Interest (ROIs) are detected, here detected people. This is a pre-processing step, consisting in dividing an image into several regions using basic segmentation, mainly based on colors or contours. Then, these regions are usually classified using Convolutional Neural Networks (CNNs) or SVMs. This process is very slow because every selected region must be predicted. In this context, the most popular algorithms are the Region-based Convolutional Neural Network (RCNN) and its Fast-RCNN [1] and Faster-RCNN [2,4] versions.

Instead of a selection of ROIs from the image, classes and bounding boxes (BBoxes) are predicted for the image in one run of the algorithm based on regression as in the well-known YOLO (You Only Look Once) [11] and SSD (Single-Shot multibox Detection) mobilenet [15,16] algorithms. Much research has addressed the topic of top-view person detection with a static fisheye camera, mostly YOLO-based techniques. In [14], a rotation-invariant training method is applied, using randomly rotated standard images (i.e., captured by a perspective camera), without any additional annotation to simulate people's various poses and orientations in fisheye images. Another YOLO-based people detection method adapts YOLOv3 trained on standard images [12], especially for people counting [33]. In this technique, each image is rotated in 15° steps; then, YOLO is applied to the top-center part of the image followed by post-processing to remove multiple detections of the same person. Recently, the algorithm proposed in [24] provides much faster and more accurate results than previous algorithms aimed at people detection in fisheye images, without any pre-processing. Based on a deep learning technique, its goal is to predict BBoxes of people, with a certain center and size, and also the angle of each BBox.

For human tracking, the key purpose of the techniques is to detect the person in a video sequence and sustain the tracking information in successive frames in order to find the trajectories of each detected object. In that respect, most studies use deep learning-based detection and tracking models. To detect or identify the object (i.e., person), a convolutional neural network is combined with a tracking algorithm. Generally, this type of model uses a detector to first detect the object, and then, subsequently a tracker is initialized to track the detected object. Several people-tracking algorithms, mostly YOLO-based detectors, have been proposed recently [62,63], respectively, combined with a Hungarian or SORT tracker. Another recent algorithm [59] makes use of the real-time performance of a SORT tracker and YOLO detector. Recent research has also attempted to use a deep-learning-based Long Short-Term Memory (LSTM) method

[61] to extract the target, and Convolutional Neural Network (Faster-RCNN) [57] has also been adopted in combination with Generic Object Tracking Using Regression Networks (GOTURN) architecture.

Compared with existing works, the technique proposed in this paper is the continuation of our previous paper [31] to achieve the detection and tracking of people in a complex scene recorded by top-view moving fisheye cameras. Clearly, the work proposed in [31] presents the first part of our work for detecting humans by a moving fisheye camera without any tracking processing. Therefore, in this paper, we exploit the performance of the detection method presented in the previous work to increase the degree of realism and accuracy suited to the existing tracking goal. To do so, we propose a Centroid tracking method which is combined with our detector in order to properly position the target throughout the sequence.

No constraints on people's movements are established, i.e., people can stand, sit, walk, kneel down, push objects and occlude each other for long periods of time. The proposed method does not compute the differences between images to extract moving objects, and it also runs with moving fish eye cameras. Moreover, this method does not require any camera calibration. To achieve this work, a new Top-view people detection dataset is introduced.

The main contributions of this work can be summarized as follows:

- An end-to-end neural network is proposed. It extends YOLOv3 to oriented people detection in top-view fisheye images, based on a periodic loss function for bounding box angle that facilitates the handling of a wide range of human body poses.
- A Centroid tracking method is utilized in order to locate the person's position in the image with respect to the scene coordinates.
- The angle of displacement from the center of the fisheye camera is estimated, then the distance of each person from the camera in meters and the trajectory and displacement vector.
- A new dataset for oriented people detection from top-view fisheye cameras is introduced that includes a range of challenges which can also be useful for tracking tasks.

2 Background study and related works

Based on the studies and research previously carried out in the field of person detection and tracking, we will briefly mention the traditional techniques utilized for detecting people, in particular on fisheye images. In this section, on the one hand, we will briefly mention existing detection techniques for fisheye cameras and, on the other hand, those used for tracking people by machine learning methods, more precisely Deep Learning.

2.1 Person detection in fisheye images

Detecting humans in images is a challenging task owing to their variable appearance and wide range of poses, especially with fisheye camera. As well-known traditional methods, people detection algorithms using classical feature extraction such as HOGs (Histograms of Oriented Gradients) or LBPs (Local Binary Patterns) have been applied to fisheye images [3,5,18]. Recently, an algorithm for detecting people using a single downward-viewing fisheye camera, proposed in [48], modeled people as upright cylinders and derived a series of elliptic detection masks whose size diminishes with the distance from the image center. They applied four SVM classifiers to features derived from each detection mask: HOG and LBP features from full-size and half-size masks. The final result is a linear combination of scores from two pairs of SVMs for HOG and LBP features. Another method for automated people detection using a fisheye-lens camera was proposed in [49]. Here, a probabilistic appearance model is built by means of kernel ridge regression. The features of body silhouette and head-shoulder contour are extracted from the human images taken at various distances and orientations with respect to the camera. Human detection is formulated as a Maximum A Posteriori (MAP) estimation using this model. Another technique is proposed in [37] to detect people in fisheye images; each fisheye image is rotated through small-value angles. Then, the HOG features are extracted from the top part of the image and the SVM classifier is applied to finalize the detection. Subsequently, in [38], a Near Ground Point Projection (NGPP) algorithm is used to detect various moving objects such as vehicles and pedestrians in fisheye images. This technique combines the method of detecting moving objects based on a point in the fisheye image and the method of compensating for motion based on the image region to filter out false detection results.

The problem of person detection, especially on fisheye images, has been increasingly studied in computer vision. Indeed, thanks to the results achieved by the Deep Learning technique, researchers are continuing in this direction in order to meet the needs of detection applications. Deep Learning is often used because the results of previous studies show significant progress in the detection of people on fisheye images. Among these techniques, the most used are: YOLO, SSD, R-CNN, Fast R-CNN and Faster R-CNN. Based on these algorithms, there are also studies that have been conducted in order to create new methods or improve the performance of detection results for existing methods. As an example, to build a detector for deformed objects in a fisheye image, the main idea in [39] is to train a rotation-sensitive neural network based on YOLOv3. A *theta* parameter is

therefore introduced to calculate the rotation angle of the detected object. Based on the YOLOv3 architecture, a new rotation-sensitive end-to-end fisheye image person detection method named RAPID is implemented in [40]. Also based on the YOLOv3 architecture in [41], the authors proposed a new technique of top-view pedestrian detection in fisheye images. Firstly, they make a transformation of individual perspective views form several views from one fisheye image. Instead of performing detection in these views separately, they combine them into a square composite image, on which the pedestrian detection is performed, and finish by mapping bounding box generated by the object detector from the perspective views to the fisheye frame.

This process is applied to detect pedestrians in fisheye images viewed from above (top view). To solve the computational cost problem and performance degradation caused by various transformations, a rotation-invariant training method during pedestrian detection in a fisheye image is proposed in [42]. It uses only randomly rotated perspective images without any additional annotation. In [43], the complexity of using fisheye lenses for top-view person detection is demonstrated using the Aggregate Channel Features (ACF) detector.

Alternatively, to detect people in a top-view fisheye image, in [44] the image is unwrapped before sending it to the ACF detector classifier. In [45], two methods for supervised people counting using an aerial fisheye camera were proposed. One concerns the adaptation of YOLOv3, trained on standard images applied to 24 overlapping and rotated windows. In the second method, YOLOv3 is applied to windows of interest extracted by background subtraction to produce the number of people using a fixed camera.

2.2 People tracking

A review of the literature on people tracking is well beyond the scope of this paper, so we will only mention a few examples of related works here. People tracking using overhead fisheye cameras is an emerging area with sparse literature. "Overhead" means that the camera is perpendicular to the ground and people are seen from above. As people appear differently in fisheye images than when using perspective cameras, the processing must be completed using other options. Consequently, to improve the target tracking performance, a pre-processing step is required in some approaches. An improved particle filter tracking algorithm is proposed in [51]; it is tied to fisheye camera calibration, but it was not tested with overhead acquisitions. In order to improve search accuracy, spherical projection is introduced to solve the non-linearity of pixel resolution caused by the distortions. Furthermore, the authors of [52] propose a distributed framework for multi-pedestrian object tracking across fisheye camera networks based on a particle filtering algorithm. Aiming at the distortion handling, the correlation filters enable target tracking on fisheye videos in [54], incorporated in the Kernelized Correlation Filtering (KCF) method. A discussion in [53] concerns an improved method of moving object detection and tracking in fisheye video sequences which are based on the moving blob method. This method is divided into two main steps: the first corresponds to the detection of the moving blobs through background subtraction while the second tracks the moving objects by means of the determined moving blobs. In [60], a human tracking space which expresses position and movement information is proposed, using a fisheye lens camera. The tracking space, in particular, shows a characteristic and simple pattern for the locus of human movement. Using this space, it is possible to discriminate the locus deterministically. In [55], another algorithm is proposed for 360° detection and tracking of both pedestrians and vehicles using multiple fisheye cameras. The approach starts by unwrapping the fisheye image, thereby enabling the chosen model to have overlapping fields of view between adjacent cameras. Secondly, Soft-Cascade with the ACF (Aggregated Channel Features) detector is used to detect both vehicles and pedestrians. The outputs from all views are combined with each other and, finally, the Unscented Kalman Filter tracks all obstacles by following an object moving around the vehicle through different fields of view. Another method is proposed in [56] for tracking human position and head direction from a ceiling-mounted fisheye camera. A fisheye HOG descriptor is developed as a substitute for HOG, where the human body and head are detected by the proposed descriptor and tracked to extract head area for direction estimation. Owing to the above-mentioned limitations of handcrafted based approaches, which show the weakness of the trackers because of the big distortion of fisheye images, especially when too many objects are in their periphery, many researchers have focused their works on deep learning-based approaches in order to overcome these limitations. Several approaches to person tracking have been explored using the deep learning technique with a normal embedded camera in different scenarios. The method in [64] is presented using a deep Reinforcement Learning (RL) technique, based on a single object tracker that tracks an object of interest in drone images. Also, a Convolution Neural Network (CNN) is used in [63] to detect and the Hungarian Algorithm (HA) to track detected humans in a drone image. For top-view images or video sequences which provide broad coverage of the scene or field of view, a deep learning-based person-tracking detection framework is proposed by [62]; it includes detection by YOLOv3 and tracking by a Deep SORT (Simple Online Real-time Tracking) algorithm using an IP (Internet Protocol) camera to provide a framework using 5 G infrastructure for top-view multiple people tracking. In a recent work, a CNN tracking technique for multiple people tracking in overhead-view indoor and outdoor environments is developed in [57]. This work mainly focuses on

overhead-view person tracking using a Faster Region Convolutional Neural Network (Faster-RCNN) in combination with Generic Object Tracking Using Regression Networks (GOTURN) architecture. This method, which has been yielding outstanding tracking results in recent years, is explored for person tracking using overhead views. However, few of these approaches have also used deep learning methods for human tracking using fisheye cameras. Recently, CNN-based algorithms have been applied with a fisheye camera. A method based on Regions of Interest (ROIs) and a Deep Neural Network (DNN) for real-time detection and tracking of moving and stationary object in fisheye images was proposed in [58]. The algorithm works on 4 views captured by fisheye cameras which are merged into a single frame. The moving object detection and tracking solution uses a minimal overhead system to isolate ROIs containing moving objects. These ROIs are then analyzed using a DNN to categorize the moving objects. The work in [61] is a novel network architecture based on Inception-v3, a deep-learning-based Long Short-Term Memory (LSTM) method for directly estimating human joint positions in a 3D space from 2D fisheye images. The Deep Multi-Fisheye-Camera Tracking (DeepMFCT) algorithm is proposed in [59] to identify customers and locate their corresponding positions from multiple overlapping fisheye cameras based on a single camera tracking algorithm (Deep SORT). It establishes the correlation between different single-camera tracks.

This study of the state of the art shows that there are few works on pedestrian detection in fisheye images, using traditional techniques or deep learning, nor the tracking of people using deep learning methods and conventional methods. However, regarding pedestrian detection and tracking using the deep learning technique, there are a limited number of approaches that apply to fisheye images in particular, which perform poorly. This is the main motivation to produce a pedestrian tracking technique for fisheye images using the deep learning method.

3 The proposed method

The emergence of deep learning has brought the best-performing techniques for a wide variety of tasks and challenges. Most of these challenges are centered around object detection and tracking. In recent years, Convolution Neural Network (CNN)-based architectures have shown significant performance improvements that are leading toward high-quality object detection, mainly regarding pedestrians, which reflects the performance of such models in terms of mean Average Precision (mAP) and Frames Per Seconds (FPS) on standard benchmark datasets.

In the present paper, a deep learning based framework is proposed utilizing object detection and tracking models and using a moving fisheye camera for human tracking. In order to maintain the balance between speed and accuracy, both an improved YOLOv3 and the centroid tracker (see Sect. 6) are utilized as person detection and tracking approaches while surrounding each detected person with oriented bounding boxes. The latter are utilized to compute the Euclidean distance between the centroids of the detected bounding box (BBox). To identify the target, new object centroids for each subsequent frame with an efficient computer representation are used. Each frame retrieved is processed through the object detection and tracking process. The proposed system is composed of two major modules, as shown in Fig. 2: foreground people detection and foreground tracking. They are described in detail below.

3.1 Oriented people detection via fisheye cameras

3.1.1 Fisheye camera description

Usually, omnidirectional and fisheye cameras offer panoramic views of 2π radian angles [26]. Catadioptric cameras are fitted with specific mirrors, whereas fisheye devices only use lenses; their angle of view can attain a 2π radian angle or more. Wide-angle lenses therefore capture typically warped images, creating the effect of a fisheye. Fisheye cameras are a major asset for several applications. These cameras are thus popular in many fields of computer vision, robotics and photogrammetric tasks such as navigation, localization, tracking and mapping. A fisheye camera is a camera fixed to a front lens group which appears as a single "big" lens, as shown in Fig. 1, left. This device enables far greater negative refraction power than usual lenses, greatly increasing the back focal distance and embracing wider fields of view [28]. In the context of people detection, the wide field of vision provided by these cameras makes people look inclined and distorted. Consequently, standard detection and tracking techniques are not reliable on warped images, especially with a cluttered and moving background [32]. Moreover, specific detectors for unconventional cameras are hard to design because they need a calibration stage which could be difficult to design [30]. Even though many algorithms already exist for perspective images, people detection and tracking through top-view images acquired by fisheye cameras are a recent topic involving deep learning algorithms. In particular, the advantage of these techniques is that the detection in fisheye images can work directly on the raw data without a pre-processing step, but few works have been developed at the present time. In this context, we propose a deep convolution network based on YOLOv3 for people detection without any pre-processing step.

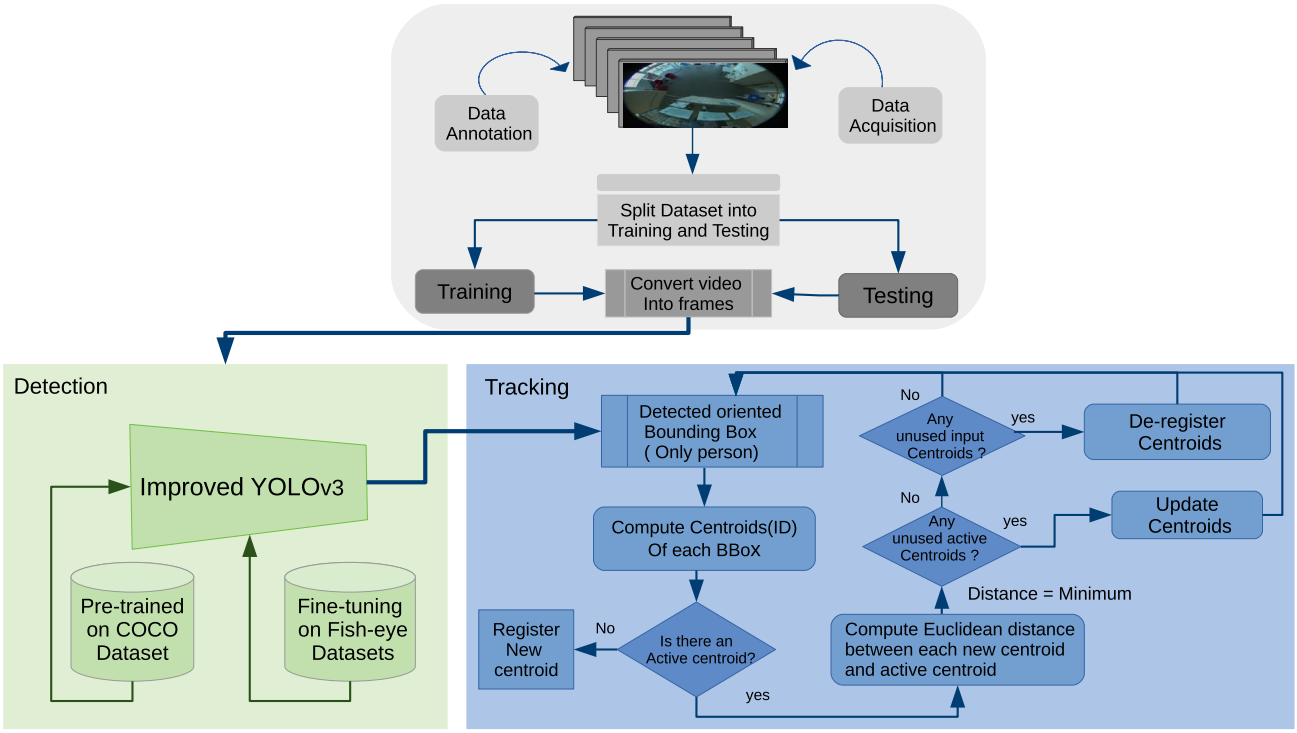


Fig. 2 Flow diagram of the proposed method

3.1.2 Top-view people detection via fisheye cameras

Overview This subsection focuses on real-time people detection using a moving fisheye camera. There are many methods for people detection and tracking using conventional cameras, as referenced to in [22]. However, people detection using fisheye cameras has barely been studied due to the complexity of such devices caused by distortion effects. Additionally, in our investigation, we particularly focus on real-time people detection in moving scenes. In recent methods, pedestrian detectors are trained using fisheye images, even though manual labeling remains a hard and time-consuming task. Another important development constraint is that this detector should be equally applicable to a visual movement sensor that is either fixed in the environment or mounted on a mobile platform (like an aerial drone). To accommodate these challenges, a CNN model based on the YOLOv3 detector is used for person detection using top-view fisheye video frames. The model is illustrated in Fig. 3. Its goal is to predict BBoxes of people, with certain center point position and size (width and height), and also the angle of each BBox. The angles of the BBoxes are an important clue for training or detection. Indeed, rectangular BBoxes encounter difficulties for object localization with different orientation angles, as produced by fisheye lenses.

The detector proposed in this paper is a full CNN with an architecture based on YOLOv3 and is configured to detect only one class, i.e., a person. In that respect, the network is structured in three parts.

Backbone The first one represents the backbone network, known as Darknet-53, trained on the ImageNet database [7]. It contains 53 convolutional layers with residual connections, each layer followed by batch normalization layer and Leaky ReLU activation. Its main goal is to extract features at different spatial resolutions; it takes an input image I and outputs a list of features (D1, D2, D3) from different parts of the network. (D1 being the highest and D3 being the lowest). Darknet-53 mainly consists of two blocks: a residual and a convolutional block. Each uses 1×1 and 3×3 successive convolutions with doubly increasing filter channels, as well as shortcut connection between the input and convolutional output with skip connections like the residual network in ResNet, as summarized in Fig. 3. Skip connection carries the input to the deeper layers, to solve the problem of network accuracy saturation which leads usually to higher training error. For this reason, the Residual Block was introduced in Darknet-53.

Feature Pyramid Network (FPN) The second part concerns the Features Pyramid Network (FPN) [8]. This network takes as input the multi-resolution features computed by our Draknet53 backbone (D1, D2 and D3) in order to extract features related to person detection. In fact, FPN contains information

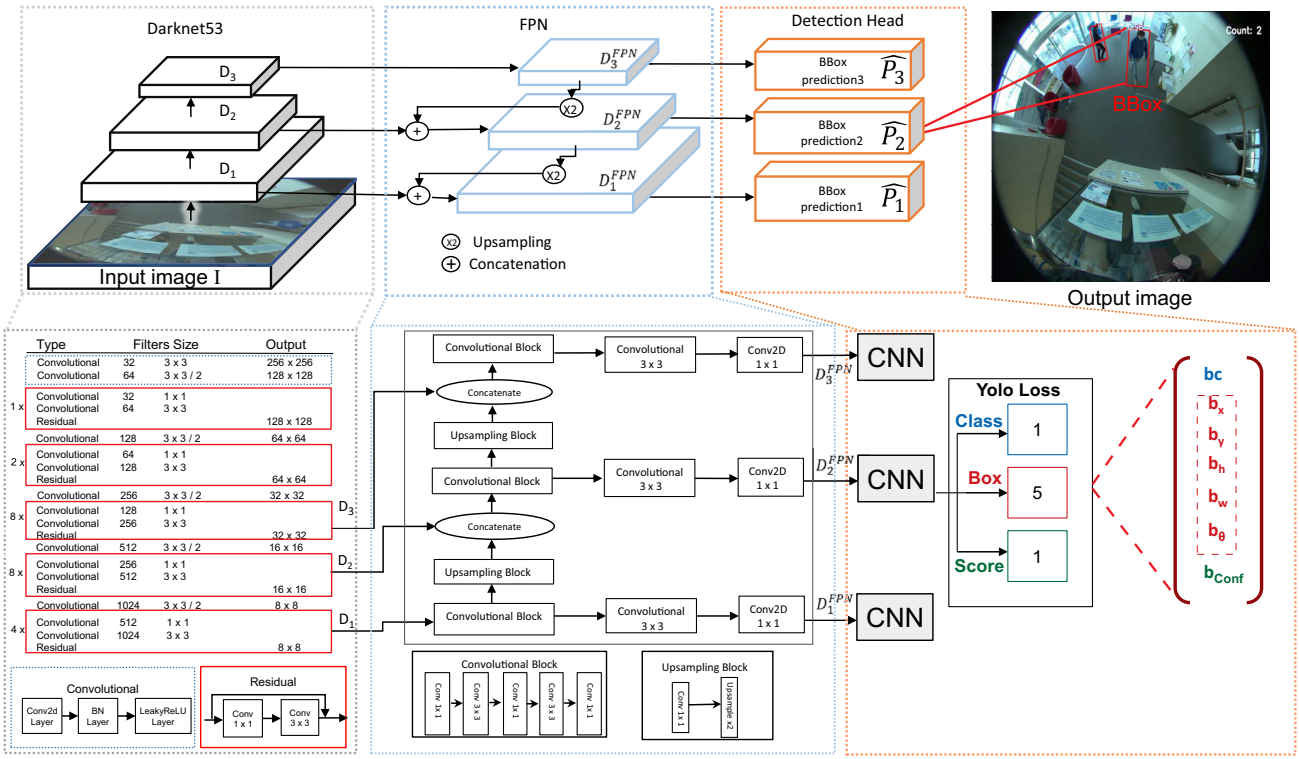


Fig. 3 Architecture of the proposed network. input: fisheye image(I), backbone (Darknet53), Features Pyramid Network (FPN), and detection head (BBox regression network), Oriented BBoxes as outputs. For the YOLO loss function, only one class, 5 parameters for each BBox and a confidence score

about small and large objects. We expect D_1^{FPN}, the output of the FPN, to contain information about small objects, and D_3^{FPN}, the output about large objects. The construction of this pyramid involves a bottom-up pathway, a top-down pathway and lateral connections as shown in the FPN block of Fig. 3. The bottom-up pathway is the feed-forward computation of the backbone ConvNet, which computes a feature hierarchy consisting of feature maps at several scales with a down-scaling step of 2. It consists of many convolution blocks, each block with many convolution layers and each layer using 1×1 and 3×3 successive convolutions. The output of the last layer of each stage will be used as the reference set of feature maps for enriching the top-down pathway by lateral connections. For the top-down pathway, the higher resolution features are up-sampled by a factor of 2—spatially coarser, but semantically stronger—from higher feature maps of pyramid levels. This is preceded by a 1×1 convolution to the corresponding feature maps in the bottom-up pathway as shown in the Upsampling block of FPN. These features are then concatenated with features from the bottom-up pathway via lateral connections. In the FPN framework, for each scale level, a 3×3 convolution filter is applied over the feature maps followed by separate 1×1 convolution for objectness predictions and boundary box regression.

Head detection Finally, the third part is head detection, building a tensor $\hat{T}_{1,2,3}$, containing information on the BBox position, including its angle of rotation. The implemented model uses a loss function combining Binary Cross Entropy (*BCE*, see Eqs. 2 and 3), as described in YOLOv3 [11] [13], and a periodic loss function that regresses the angle of each BBox, accounting for angle periodicities [9] [24,33]. Therefore, the detection of oriented objects is an extension of a general horizontal object detection.

Oriented Bounding Box Detection In fisheye images, since most targets have an orientation that is neither vertical nor horizontal, rotated object detection is essential for overhead people detection (an example is given in Fig. 5). In our case, outputs of an improved YOLOv3 [11] network are used with both horizontal location boxes and angle information, rendering the YOLOv3 module more sensitive to the angle. By introducing the oriented BBoxes, for each video frame, the predicted results of the proposed framework return six BBox parameters: the position coordinates (b_x, b_y), the BBox size (b_w, b_h) and the angle of all individuals b_θ. They are represented by a six-dimensional vector $(b_x, b_y, b_w, b_h, b_\theta, b_{Conf})$, where b_{Conf} is the predicted confidence score; it quantifies how confident the algorithm is that the target represents a human being. In addition, there is a confidence threshold,

determined by the user, but usually fixed to 0.5, and the algorithm only returns the BBoxes whose confidence score is higher than this threshold. Figure
refbbox shows the transform from the anchor to the BBox where the coordinates center (b_x, b_y) of BBox is calculated by applying a sigmoid to predicted values and adding the corner points of the corresponding grid cell. Meanwhile, the dimensions b_w and b_h of the BBox are calculated by applying a log-space transform to the predicted output dimensions and then multiplying with the anchor dimensions (p_w, p_h). The network establishes a multitask and crucial loss function (Eq. 1) inspired by that used in YOLOv3, with an additional BBox rotation angle loss to optimize the target detection. It is computed by the ground truth and the predicted result of the network:

$$Loss = Loss_{Box} + Loss_{Conf} + Loss_{Angle}, \tag{1}$$

where the Box regression loss ($Loss_{Box}$ in Eq. 2) is calculated only when the prediction box contains detected people. Confidence loss ($Loss_{Conf}$ in Eq. 3) determines whether there are people in the prediction frame. BBox rotation angle loss ($Loss_{Angle}$ in Eq. 4) determines the prediction orientation of a person. Note, that the category classification loss is not used since only one class (i.e., people) is used here. Here, $\hat{T} = (\hat{t_x}, \hat{t_y}, \hat{t_w}, \hat{t_h}, \hat{t_\theta}, \hat{t}_{conf})$ represents the transformed version of BBox predictions for each stride s_k, from which a BBox prediction $\hat{b} = (\hat{b_x}, \hat{b_y}, \hat{b_w}, \hat{b_h}, \hat{b_\theta}, \hat{b}_{conf})$ is computed as represented in Fig. 4, where (t_x, t_y, t_h, t_w) is calculated from the ground truth $b = (b_x, b_y, b_w, b_h, b_\theta, b_{conf})$. These loss functions are given by the following three formulas:

$$\begin{aligned} Loss_{Box} = & \sum_{\hat{t} \in \hat{T}^+} BCE\left(\mathscr{S}(\hat{t_x}), t_x\right) + BCE\left(\mathscr{S}(\hat{t_y}), t_y\right) \\ & + \sum_{\hat{t} \in \hat{T}^+} \left(\mathscr{S}(\hat{t_w}) - t_w\right)^2 + \left(\mathscr{S}(\hat{t_h}) - t_h\right)^2, \end{aligned} \tag{2}$$

$$Loss_{Conf} = \sum_{\hat{t} \in \hat{T}^+} BCE\left(\mathscr{S}(\hat{t}_{conf}), 1\right) + \sum_{\hat{t} \in \hat{T}^-} BCE\left(\mathscr{S}(\hat{t}_{conf}), 0\right), \tag{3}$$

with $\mathscr{S}$ the logistic sigmoid activation function, $(\hat{T}^+, \hat{T}^-)$ positive and negative samples from the predictions, respectively.

$$Loss_{Angle} = \sum_{\hat{t} \in \hat{T}^+} L_{Angle}\left(\hat{b_\theta}, b_\theta\right), \tag{4}$$

such that for a given range (α, β), $\hat{b_\theta} = \alpha \cdot \mathscr{S}(\hat{t_\theta}) - \beta$, which is the prediction of b_θ and the function L_{Angle} is defined by

$$L_{Angle}\left(\hat{b_\theta}, b_\theta\right) = R\left(\text{mod}\left[\hat{b_\theta} - b_\theta - \frac{\pi}{2}, \pi\right] - \frac{\pi}{2}\right),$$

with "mod" representing the modulo operation and R a symmetric regression function.

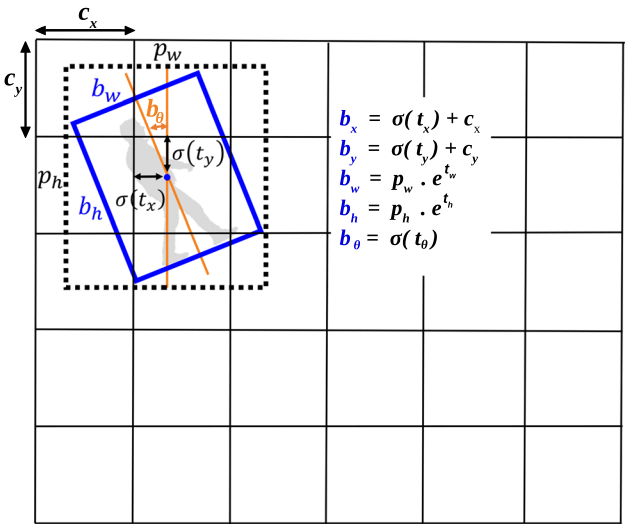


Fig. 4 Oriented Bounding Box (BBox) with its tied parameters

3.2 Implementation details

3.2.1 Dataset description

Numerous benchmarks and datasets have been created in order to train and evaluate people detection algorithms for fisheye images. Most of the existing public fisheye datasets are annotated by an aligned BBox. In this work, a dataset of overhead fisheye images with oriented BBoxes is needed for each person aligned with its orientation in the image. However, various challenges are reported on dealing with fisheye images, mainly spatial and temporal illumination variations, occlusions and various body poses. Additionally, the appearance is different when people are walking right under the camera or at the periphery of the fisheye image, and the image resolution is low near the borders, regularly disrupting the detection. To overcome these challenging scenarios of different videos captured from a moving fisheye camera, a new dataset has been collected and annotated. It is called Oriented Bounding Boxes from Moving Fisheye cameras (OBBMF) and consists of 6 videos (see Table 1 for details). Clearly, the new dataset contains many more frames and human objects, and also includes challenging scenarios, which do not exist in the other datasets. Furthermore, experiments were performed using three public datasets, MW-18Mar[1], HABBOF[2], CEPDEOF[3], in addition to our dataset in order to fit and evaluate the effectiveness of our proposed method.

[1] http://www2.icat.vt.edu/mirrorworlds/.

[2] http://vip.bu.edu/projects/vsns/cossy/datasets/habbof/.

[3] http://vip.bu.edu/projects/vsns/cossy/datasets/cepdof/.

Table 1 Descriptions of video sequences and their tied challenges

Video	#Person(s)	#Frames	fps	Description/challenges
Stairs	2	500	48	Person go up and down the stairs with rotational movement of camera
Parking	2	536	48	Person walking, body camouflage with the scene
Window	2	534	48	Person in a top-view position and nonuniform illumination
Workshop	5	530	48	More than 4 walking and sitting in a large space
Entrance1	2	543	48	Person walking and sitting in center and boundary of image
Entrance2	1	567	48	Walking activity in a reception room with Top-view challenge

3.2.2 Data acquisition

In our case, the data were collected both indoors and outdoors in the CERIS laboratory of the IMT Alès research center. The videos were collected with a fisheye camera (Basler ace acA1300-200uc) facing down at 48 fps, where one or several persons adopt various poses such as walking and sitting under the camera, there are also considerable body occlusions present, and people go up and down the stairs with rotational movement. Then, a number of frames are generated from these video clips which contains new scenes with some challenging scenarios such as illumination, camera rotation and motion in the center (Top-view), which are generally unavailable in the standard literature.

3.2.3 Data annotation and file conversion

Data labeling is an essential step in a supervised machine learning task requiring significant manual work. To annotate our new dataset, person regions are manually annotated with rotated BBoxes. In order to do so, the MVTec Deep Learning Tool[4] is used. It is a very useful deep learning target detection and labeling tool, but it generates a *hdict* file which cannot be directly used in other deep learning tools. It is therefore converted into a *txt* file in the first step using a small application with C# and HalconDotNet-V. 19.11.0 and finally reconverted to the data format of YOLOv3 as a *json* file. Technically, first, the main axis of the rectangle is drawn to define the orientation of the person in the scene. Then, the width and height of the person in pixels are defined. Consequently, each BBox is represented by five parameters:

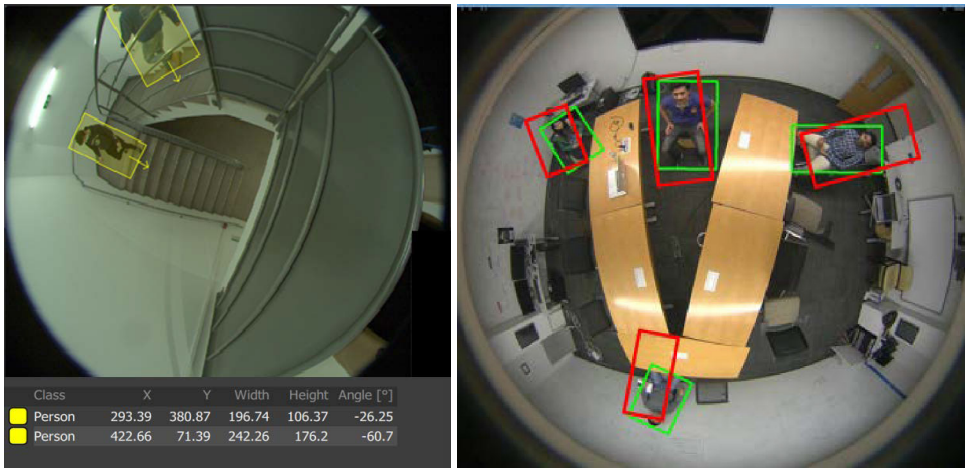
- (x, y): coordinates of the BBox center,
- w and h: width and height of the BBox, respectively,
- θ: clockwise rotation angle from the vertical axis.

Furthermore, Fig. 5(a) and (b) represents these parameters in a rectangle BBox. The whole of our data set is represented by more than 12,000 OBBMF. However, some annotations in the MW database contain errors in the angles, as shown in Fig. 5(b). Consequently, these annotations were manually modified to improve the relevance of this database. It provides a very efficient model for human being detection.

3.2.4 Training datasets

Concerning training, a pre-trained Darknet53 model was used as a starting point, which is initialized with ImageNet pre-trained weights for faster training. Also, in order to train the proposed detector, we used one of the largest datasets, MS COCO, [34] which is commonly utilized as a general object detection benchmark, because it contains various appearances of people. Our network was trained end to end by optimizing the cross entropy loss function by updating weights using "Stochastic Gradient Descent" (SGD) [36] with a momentum of 0.9 for more than 50,000 iterations (one iteration contains 128 images). The SGD represents an iterative optimization technique [35] and is most widely used in the field of deep learning to minimize the loss function to search hyper-parameters. This algorithm calculates the gradient and makes the update of the network parameters by means of the training subset, which is called the mini-batch. Each gradient evaluation using the mini-batch is defined as an iteration. At each iteration, the algorithm takes one step to minimize the loss function. The complete progress of the training algorithm over the total training set using mini-batches is called an epoch. During experiments, the initial learning rate is set to 0.001 and the weight decays to 0.0005. The mini-batch size is set to 16, and the network is trained for 500 epochs. We set the learning rate factor to 0.0001 with the same SGD parameters and batch size to fine-tune the parameters of the network. All the processes were trained on multiple cross fisheye datasets for more than 8000 iterations from weights pre-trained in ImageNet using COCO. For these two networks, the images were resized into 608×608 pixels and fed into the network until the loss was saturated. Our model was conducted on an NVIDIA Quadro P5000 GPU accelerator (Pascal architecture). It includes 2560 CUDA cores with 16 GB GDDR5 memory. The host is an Intel® Xeon® CPU E5-1620 V4 processor with 4 cores.

[4] https://www.mvtec.com/products/deep-learning-tool.



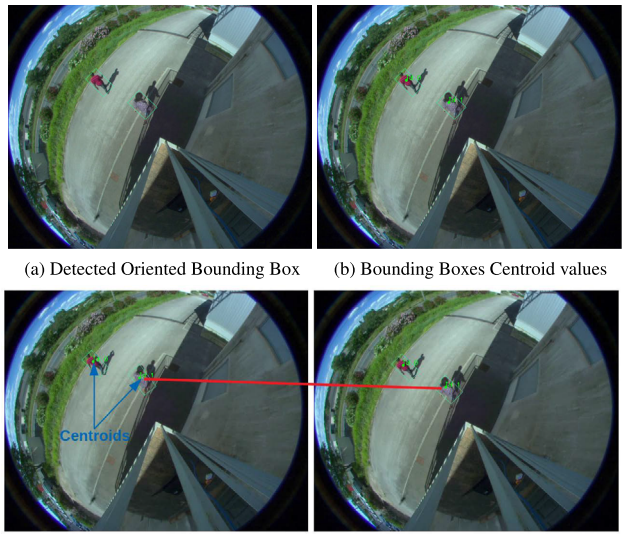
(a) Frame annotation with BBox angles

(b) Frame and BBoxes of MW-R dataset

Fig. 5 Examples of annotated frames with BBox rotation angle. In (b), a frame and its BBoxes from MW-R dataset (the red BBoxes are the initial tied to the MW-R dataset, whereas the green correspond to the corrected values)

3.3 Top-view people tracking via centroid

After performing people detection, correctly detected people are retained based on their class IDs (unique IDentify) and detection scores. The second step in this work is to track detected persons in order to locate their position in the image with respect to the scene coordinate. As tracking is intended for associating target object that appears on sequential video frames with ID preserved, it is different with object tracking with goal to classify object that appears in a frame to each corresponding category. Many recent methods are evolving with regard to tracking, both in Machine Learning and Deep Learning, and also Correlative Filter based trackers such as MOSSE [67], KCF [60] and CSRT [68] with higher accuracy and the ability to be run in stand-alone mode. This tracking method is commonly faster as it has knowledge about previous object location to determine the next probable location, it is better at handling occlusion due to its predictive nature, and it can preserve identity by harnessing location information.



(a) Detected Oriented Bounding Box

(b) Bounding Boxes Centroid values

(c) Euclidean distance between Centroid values

Fig. 6 Example of two successive frames at time $t-1$ and t in (a) and (b), respectively. The bounding boxes of moving persons are specified by IDs (centroids), and the red line represents the displacement of the centroid between each pair of existing object centroid and input object centroid

3.3.1 Centroid tracker

Centroid tracking is chosen in this work due to its lightness. The method is highly applicable and has the advantage of being relatively rapid even without using GPU. As the target objects in this research project are in movement and with different scenarios, and using a moving fisheye camera, it is best to keep the person detection run for every frame.

The centroid tracking method will accept the bounding box (BBox) produced from our object detector. Once the BBox coordinates are calculated, the centroid [65,66] must be computed. More precisely, there are the center (x, y) coordinates of the BBox for each detected object in every single frame. For every subsequent frame in the video stream, computing object centroids are applied; however, instead of assigning a new unique ID to each detected object, we first need to determine whether the new object centroids can be associated with the old object centroids. Basically, to assign the ID, the first initial set of BBoxes presented we will assign them unique IDs.

We first need to determine whether we can associate the new object centroids with the old object centroids. For any new centroid location, commonly caused by a new object that was not visible/detected in the previous frame, a new ID will be assigned. As the camera moves and the views shift, some people that were detected will disappear or no longer be detected. The ID for any disappeared BBox will be de-registered, so when the same person becomes visible again later, he/she might get assigned a new, different ID.

Figure 6 demonstrates the accepting of a set of BBox coordinates and computing the centroid. To accomplish this process, the Euclidean distances presented in Eq. 5 are computed between each pair of existing BBox centroids and the new centroids. This process is repeated by updating the coordinates (x, y) of existing objects. Usually, the primary assumption of centroid tracking algorithm is that a given object will potentially move in between subsequent frames, the distance between the centroids for frames at time $t-1$ and t will be smaller than all other distances between objects. Therefore, if we choose to associate centroids with minimum distances between subsequent frames, hence we can build our object tracker. In the event that there are more input detections than existing objects being tracked, we need to register any new object. This simply means that we add the new object to our list of tracked objects by assigning it a new object ID. Then, we must store the centroid of the BBox coordinates for that object and repeat the pipeline of steps for every frame in our video stream. From Fig. 6, two persons are detected in the image. This example illustrates the motion regions for a set of point tracks of a person at two time instances $t-1$ and t. The points p_1 of coordinates (x_1, y_1) and p_2 of coordinates (x_2, y_2) give the centroid location of moving object in the two different images at two time instance $t-1$ and t.

$$Ed\,(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \tag{5}$$

4 Experimental results and evaluations

4.1 Evaluation metrics

The detection system returns a list of detected BBoxes in an image. The match of a detected BBox and the ground truth is rated by asserting an overlap area of more than 50%. To quantitatively evaluate the performance of the proposed network, the statistical analysis of $Precision$, $Recall$, F-score and Average Precision (AP) is performed as the evaluation metrics. With TP, FP and FN denoting the number of true positives, false positives and false negatives in a video, $Precision$ means the percentage of the correctly detected persons (TP) over all the detected persons $(TP + FP)$:

$$Precision = \frac{TP}{TP + FP}. \tag{6}$$

Meanwhile, $Recall$ is the ability of a model to find all the objects. It associates with the correct predictions among all the positive cases, which means the percentage of the correctly detected:

$$Recall = \frac{TP}{TP + FN}. \tag{7}$$

Hence, $Precision$ and $Recall$ are considered to be common evaluation metrics,

and the F-score combines the two:

$$F = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}. \tag{8}$$

Finally, Average Precision (AP) is the area under the Precision–Recall curve: $AP = \int_0^1 F(x)\,\mathrm{d}x$. Therefore, the closer the evaluation scores of both F and AP are to 1, the more the detection is qualified as suitable. On the contrary, a score close to 0 corresponds to poor detection of persons.

4.2 Detection benchmark results

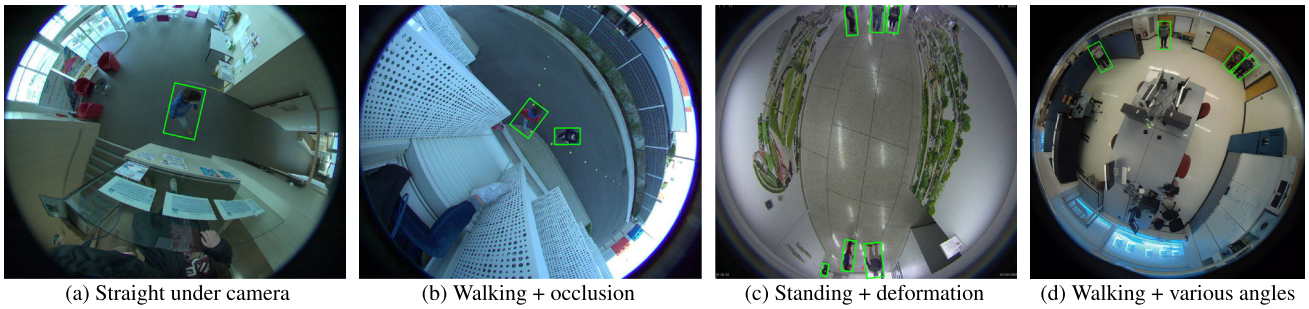
Various experiments are presented to analyze the performance of the proposed method. We fine-tuned the algorithm trained on COCO with various cross datasets from MW-R, CEPDEOF, HABBOF and OBBMF. Hence, we cross-validate on these datasets, i.e., two datasets are used for training; then, they are tested on another dataset. For example, a cross dataset trained on MW-R + HABBOF is tested on OBBMF, and inversely. Table 2 shows the detection performance of our method on each video in the OBBMF dataset obtained by fine-tuning with cross-validation1 (index $cross1$) and cross-validation2 (index $cross2$), respectively. Our model with

608×608 resolution achieves impressive performance, despite using videos captured from a moving fisheye camera. The proposed method performs outstandingly with acceptable convergence behavior in several experiments carried out with various cross validation datasets.

The overall performance metrics obtained with our model and with our OBBMF dataset are evaluated using AP, $Precision$, $Recall$ and F-measure. As shown in Table 2, the new algorithm performs efficiently on ordinary videos with a score more than 0.95 for AP. However, more complex scenes (moving camera, low light, marked shadows, etc.) remain challenging. Figure 7 shows sample results applied to the four datasets where detections are nearly perfect in a range of scenarios, such as various body poses, orientations and diverse background scenes. However, some scenarios, such as people's images on a projection screen, in low light and with marked shadows, remain challenging. For this reason, we expanded our training datasets by cross validating various samples from all the used datasets, in order to improve the resulting performance model and prevent it from over-fitting.

Table 2 Performance comparison of our method with cross validation $cross1$ and $cross2$ for each video in OBBMF dataset

	Performance metric					
	AP_{50}	AP_{75}	AP_{90}	Precision	Recall	F-measure
$Stairs_{cross1}$	0.857	0.478	0.467	0.810	0.808	0.852
$Workshop_{cross1}$	0.785	0.306	0.396	0.818	0.649	0.756
$Window_{cross1}$	0.891	0.511	0.502	0.901	0.687	0.765
$Parking_{cross1}$	0.864	0.501	0.490	0.903	0.762	0.894
$Entrance1_{cross1}$	**0.970**	0.576	0.564	0.972	0.936	0.963
$Entrance2_{cross1}$	0.686	0.432	0.556	0.791	0.692	0.637
$stairs_{cross2}$	0.911	0.432	0.544	0.901	0.911	0.839
$Workshop_{cross2}$	0.929	0.402	0.661	0.849	0.912	0.918
$Window_{cross2}$	0.916	0.642	0.706	0.791	0.892	0.883
$Parking_{cross2}$	**0.971**	0.557	0.691	0.913	0.992	0.926
$Entrance1_{cross2}$	0.896	0.530	0.656	0.891	0.892	0.837
$Entrance2_{cross2}$	0.686	0.432	0.556	0.893	0.992	0.902



(a) Straight under camera (b) Walking + occlusion (c) Standing + deformation (d) Walking + various angles

(e) Walking + rotation (f) Various angles and poses + far (g) Different poses (h) Low-light scenario

Fig. 7 Detection results of our benchmark on sample frames in different scenarios and challenge, including various poses, orientations and background scenes. Green boxes are predicted BBox (true positives, i.e., matching of a detected BBox and the ground truth with an overlap area of more than 50%)

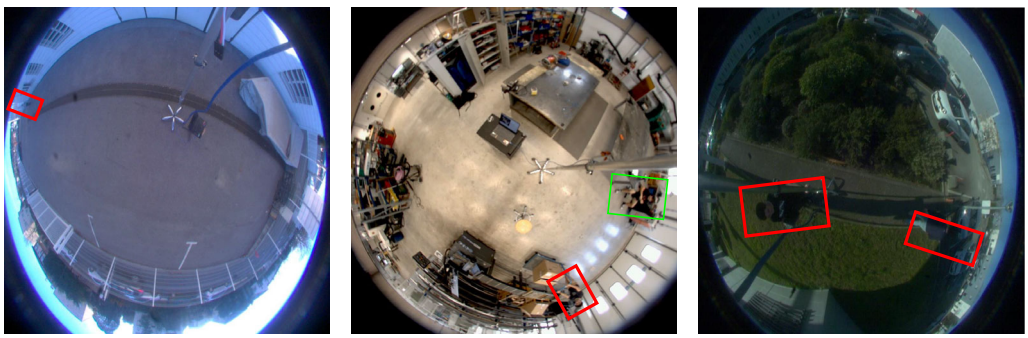
The resulting cross dataset was split: 70% of data used for the training stage and 30% for testing.

On the one hand, Table 3 clearly shows the improved performance with large-scale datasets, exceeding 90% of the AP. On the other hand, the $P--R$ ($Precision--Recall$) curve of the best model on the test set is plotted in Fig. 9; it shows the trade-off between $Precision$ and $Recall$ as the threshold score of the model changes. $Recall$ should increase to guarantee that all the persons are detected. However, as $Recall$ increases, it is common for some scenarios, such as distortion, camera movements, low light and marked shadows to reduce $Precision$. Ideally, the upper right corner of the curve should reflect 100% $Recall$ and $Precision$, often impossible to obtain in real scenarios. Finally, $P-R$ curve illustrates why the AP of our model is high. Indeed, $Precision$ remains close to 100% for $Recall$ values as high as 85%, with a 0.98% for the optimal point.

In that respect, our method outperforms the alternatives for all challenging scenarios, but still has its limitations, as shown by the metrics discussed above. In frames (a), (b) and (c), respectively, of Fig. 8, two false negatives are presented. The one in (a) is not detected when the person walks to the very edge of the image and appears too small (few pixels). The frame in (b) produces a true positive and a false positive

Table 3 Performance tuning evaluation of our method with multi cross validation for each video in each dataset

	Performance metric					
	AP_{50}	AP_{75}	AP_{90}	Precision	Recall	F-measure
Lab2	**0.980**	0.863	0.673	0.977	0.875	0.883
stairs	0.936	0.717	0.598	0.960	0.925	0.873
Lunch2	**0.971**	0.254	0.446	0.976	0.969	0.973
Meeting2	**0.978**	0.720	0.594	0.977	0.957	0.967
Workshop	0.942	0.817	0.655	0.942	0.925	0.892
MW-R18	0.941	0.690	0.514	0.955	0.899	0.894



(a) People too far (on the right) (b) Cluttered background and colors + illumination problems (c) Backlight problem.

Fig. 8 Examples of top-view fisheye images where people are not detected. Green boxes are true positives; red boxes are false positives

Table 4 Performance comparison of our method and previous state-of-the-art methods in fisheye dataset

References	Dataset	Method	Precision(%)
Chiang et al. [18]	–	HOG+SVM+ Image calibration	90.01
Nguyen et al. [19]	Bomni	ConvNets-based YOLO	87.73
	Bomni	YOLO-Tiny	54.95
Wang et al. [20]	Bomni	Mask-RCNN	76.82
Li et al. [33]	HABBOF	YOLOv3 + post-processing	88.11
Tamura et al. [14]	MW-R, PIROPO, Bomni	YOLOv2 + Rotation data augmentation	86.30
Duan et al. [40]	MW-R, HABBOF, CEPDOF	YOLOv3 + end-to-end rotation-aware	95.10
Ours	OBBMF, MW-R, HABBOF, CEPDOF	Oriented BBox-YOLOv3	**98.01**

as shown by a red box. It misses people due to the cluttered background, colors and also lighting problems. During this experiment, this same person was detected some frames before but not in this precise scenario. Usually, the proposed detector fails when people are confused with the background; this is a common detection problem that occurs for detectors in general. Another example of a false negative appears in (c), caused by black clothes on a dark background. In addition, sometimes a false positive erupts at the edge of the image where cars appear distorted. Nevertheless, looking at the other scenarios where the images are almost semantically intelligible, our detector was able to properly detect all people present in the frame.

The results in Table 4 compare the detection performances obtained using state-of-the-art techniques and indicates the overall precision of our approach. The reported performances prove its efficiency compared to all the referenced methods. It also provides a performance comparison with handcrafted feature-based methods. Our method performed better than Li et al. [33] and Tamura et al. [14] techniques and slightly

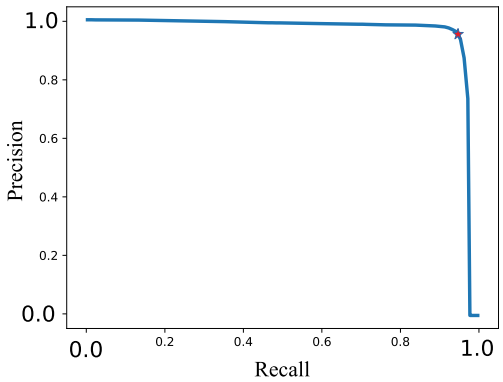


Fig. 9 Precision–recall curve of the best model on the test set. The precision remains close to 100 % for recall values as high as 85%. The optimal point (the closer to the upper-right corner) is at 0.981

better than the method of Duan et al. [40]. Actually, state-of-art techniques performed poorly, although they are based on the combination of the detector YOLO and pre- and/or post-processing stages (data augmentation, calibration, rotation, etc.), which is not the case for our method. However, our people detector outperforms the other algorithms by a large margin on four datasets, which include challenging scenarios, such as various body poses and occlusions. In conclusion, the proposed method works well in both simple and challenging cases. Otherwise, training the model and fine-tuning hyper-parameters on a large-scale dataset are very efficient to obtain high performance levels on a small dataset.

4.3 Centroid tracking results

4.3.1 Estimation of distance (Depth) between the center of the fisheye camera and the detected person

Once the people in the frame have been correctly identified and tracked, our goal is to estimate the real distance of each person from the camera in meters. This principle relies on a two-step process: i) projection of the camera center in each frame using a located marker and ii) distance estimation between the center of the fisheye camera and the detected person. An approach based on the calculation of the Euclidean distance to estimate the distance in meters between the center of the fisheye camera and the target is shown in Fig. 10. In our case, the altitude of the fisheye camera is fixed at 5 ms with a top-view field. An interpolation of the distance was carried out based on the evolution of the curve of the Euclidean distance as shown in Fig. 11. Then, a set of intervals was fixed depending on the ground truth in order to convert this distance in meters.

4.3.2 Estimated trajectory and displacement vector

In order to follow the trajectory of the target in the image plan, a displacement vector is created; this vector represents the set of center points of the BBox in each frame. It is through these vectors that the trajectory is traced. The size of the vector is not limited;, it depends only on the number of frames in the video. In the presented results, 100 points are chosen for visualization and understanding.

Figure 12 shows the target moving in a straight line and the target in a zigzag pattern (going from left to right and back again in the image plan).

4.3.3 Estimation of the angle of displacement from the center of a fish-eye camera

This subsection presents an approach for estimating the displacement angle of the target in a semi-circle. It is based on three points $p_{i,i\in\{1,2,3\}}$ with their tied coordinates $(x_i, y_i)_{i\in\{1,2,3\}}$:

- $p_1(x_1, y_1)$: the center of the fisheye camera,

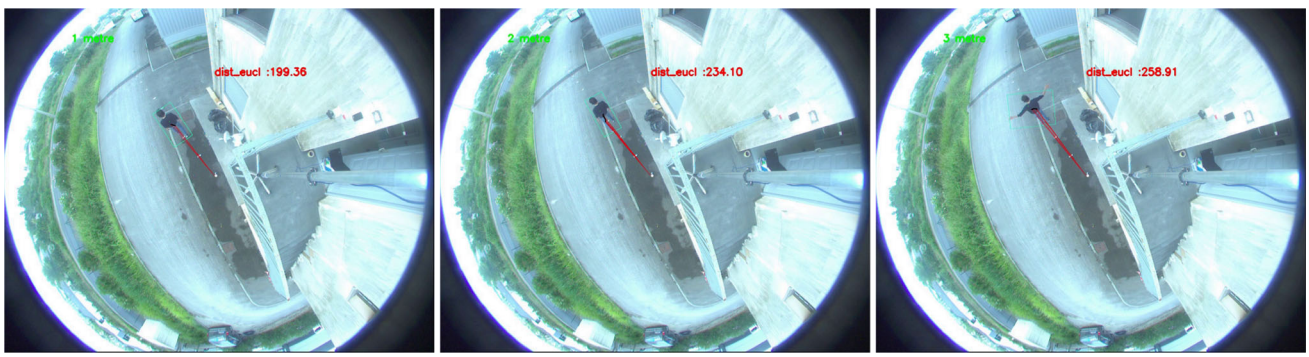


Fig. 10 Distance estimation approach between fisheye and target (1 m, 2 m and 3 m)

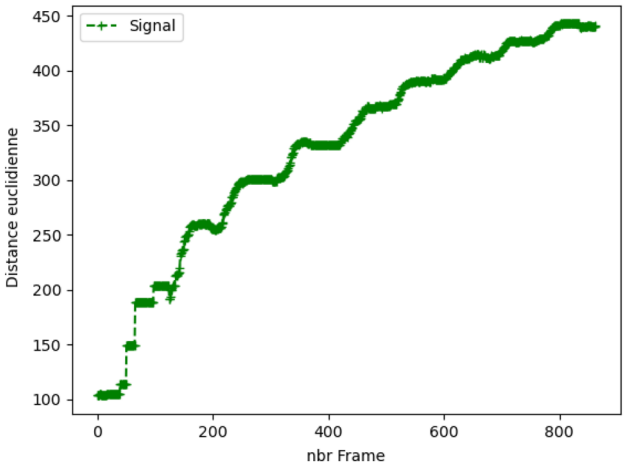
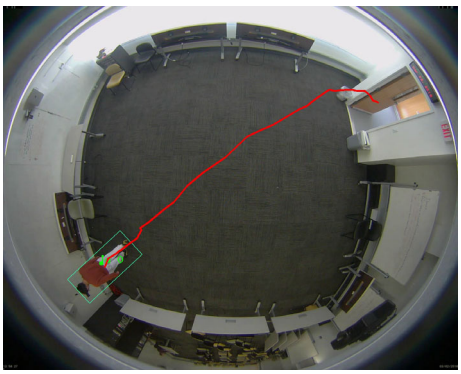
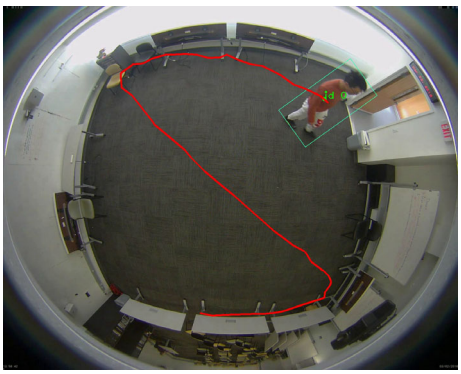


Fig. 11 Evolution of the distance according to the number of frames



(a) Straight line



(b) Zigzag line

Fig. 12 Estimation of displacement trajectory

- $p_2(x_2, y_2)$: the initial point of the first BBox of the detected target (i.e., the center of the BBox),
- $p_3(x_3, y_3)$: the center of the BBox (of the moving target).

To estimate the value of the orientation angle of the moving target, a trigonometric function $arctan$ is applied. Considering the gradient of the line m_1 between the center of fisheye camera and the initial point of the target, m_2 represent the gradient of the line between the camera center and the centroid of the moving target. They are computed by:

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{and} \quad m_2 = \frac{y_3 - y_1}{x_3 - x_1}.$$

Now, let θ be the value of the orientation angle tied to the moving target. It is computed by the following equation:

$$\theta = \arctan\left(\frac{m_2 - m_1}{1 + m_2 \cdot m_1}\right). \tag{9}$$

Figure 13 shows the results of experiments in order to estimate the angle of orientation between the center of the fisheye camera and the target person detected. We compared the results obtained of the angle calculated by our method to the one calculated by our oriented BBox detector. The similarity between the two values of the angle clearly proves the performance of our method.

4.4 Ablation experiments

In this section, we present various ablation experiments to analyze how each part of our method individually contributes to the overall performance.

4.4.1 Impact of angle loss functions

To analyze the impact of the loss functions, we compare our proposed loss function with a baselines: standard YOLOv3 loss without using the Loss_angle function. We perform the same experiment using AP_{50} metrics of the testing on our dataset for both Loss_standard and with the use of Loss_angle. As reported in Table 5, the Loss_angle achieves the best performance compared to the standard loss function, with a score of 0.971.

4.4.2 Training/Fine_tuning analysis

As mentioned in Sec. 4.2, the training started using weights pre-trained on the COCO dataset. We fine-tuned the algorithm trained on COCO with cross validation1 (index cross1) and cross validation2 (index cross2) and with multicross from the MW-R, CEPDEOF, HABBOF and OBBMF datasets. We chose to train on the datasets in this order, as the context in which the network will be applied is expected to be closer to the environment we obtained images from. To quantify how close we were to overfitting the model, we observed the rate at which the IoU increased. Figure 14 represents the evolution of the IoU during training for 50k iteration with COCO datasets. It shows better learning performances, especially at the end of training where it begins converging much sooner, which makes the fine_tuning faster with the proposed crossed data especially at the start of training, where it begins converging much sooner.

Table 5 Comparison of our proposed loss function and the standard loss function, namely Loss_angle and function Loss_standard, respectively

Loss function	AP_{50}
Loss_standard	0.617
Loss_angle	0.971

4.4.3 Detected angle versus tracked angle

To analyze the effect of the orientation angles, the results obtained for the angles calculated by our method are compared to the ones estimated by our oriented BBox detector. As presented in the second row of Table 6, three videos with different displacement trajectories were used to evaluate the performance of the predicted angles. To do so, the RMSE (Root Mean Square Error), the maximum error and the STD (Standard Deviation) are calculated for each video sequence between the detected angle and the one predicted by the tracker. The original signals are presented in Fig. 15a–c. As reported in Table 6, the similarities between the two angles clearly demonstrate the performance of the centroid tracker. Indeed, the RMSE is calculated for each video sequence between the detected angle and the one predicted by the tracker. The error evaluation is very low and tends slightly toward zero. The STD score is also low: around 5 degrees, which proves the effectiveness of the proposed approach, especially with the circular line scenario where the STD tends toward 3 degrees. Finally, even though sometimes there is suddenly a strong detection error (Maximum error $\approx 54°$), the person is usually well detected in the following frame, illustrating the robustness of the proposed method.

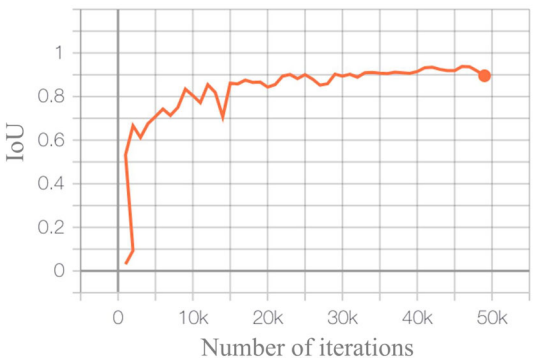
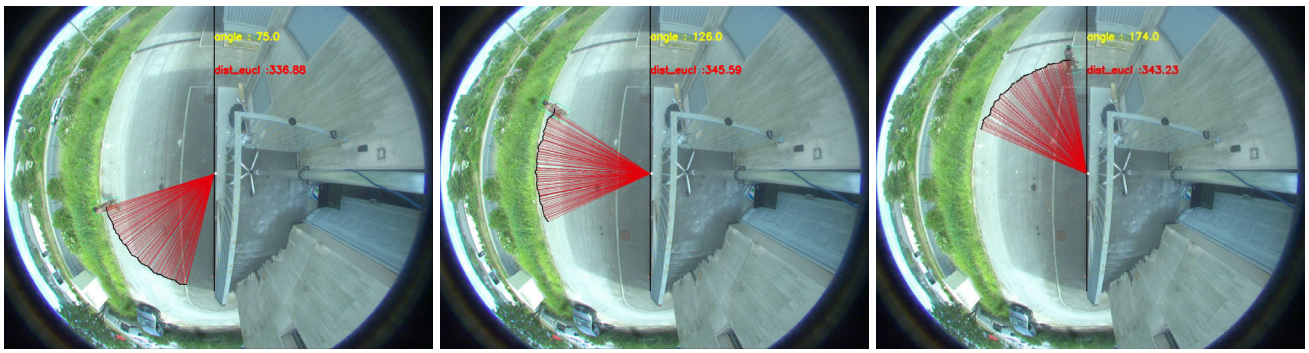


Fig. 14 Evolution of IoU during training for 50k iterations

5 Conclusion

An approach is proposed in this paper to detect and track people in top-view fisheye images, using moving cameras. It is based in the first stage on a pre-trained deep CNN architecture, extended from the YOLOv3 detector. Then, the bounding boxes generated by the detector are fed to the centroid based tracking algorithm. It tracks each person by computing the centroid height of each person in consecutive frames; the trajectory and the distance of the person compared to the camera can be determined. The system shows very promising results as we tested it using different real-world scenarios streaming directly from the camera. Experimental results confirm that people are extracted in an indoor environment using videos streaming from a moving fisheye camera with a high level of AP. Our approach eliminates the need for pre-processing and/or data augmentation, by considering oriented bounding boxes. The limitation of this method is with the centroid tracking algorithm. The centroids of the object must lie close together between subsequent frames or the ID number might be switched due to overlapping of one object by another. However, this problem does not overly affect our system. Finally, a new dataset of videos was created concerning human detection using a moving top-view fisheye camera; the great interest is that the ground truths are also available online. For future works, we

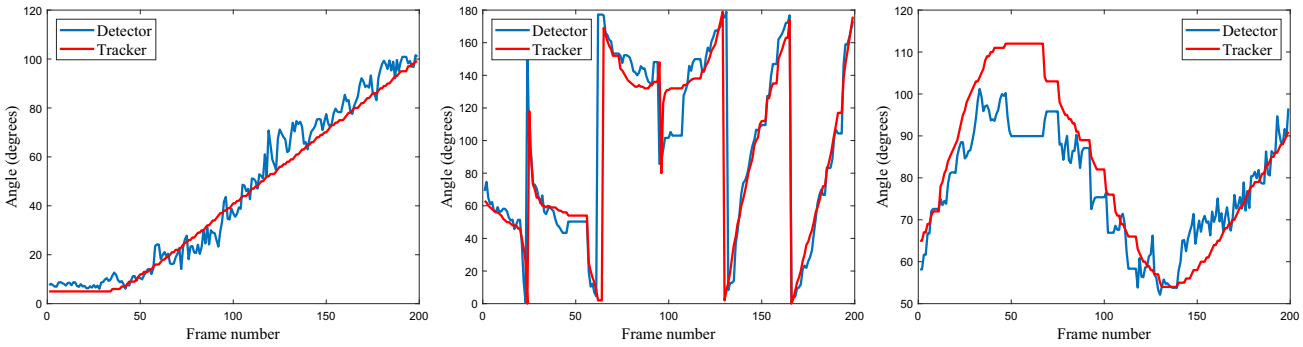


(a) 75 ° angle (b) 126 ° angle (c) 174 ° angle

Fig. 13 Estimation of the angle of displacement from the center of a fish-eye camera

Table 6 Evolution comparison between the detected angle and tracked angle with different displacement trajectories, statistics tied to Fig. 15

Videos	Displacement trajectory	RMSE	Max_Error	STD
Parking	Circular line	0.38	18.79	3.35
MW-R18	Zigzag line	0.65	21.95	6.29
Entrance	Straight line	0.77	53.62	7.70



(a) Video 1: circular trajectory (b) Video 2: Zigzag trajectory (c) Video 3: Straight line

Fig. 15 Evolution of the angles between the detected and tracked angles, respectively, with different displacement trajectories

plan to expand the proposed model with various real-world applications, especially for real-time human detection and tracking using overhead fisheye videos captured and treated automatically from an aerial drone.

Declarations

Conflict of interest Olfa Haggui, Hamza Bayd, and Baptiste Magnier declare that they have no conflict of interest. This work is part of the 'MOVCAP' project, funded by the European Regional Development Fund, region Occitanie in France.

References

1. Lin, H., Kong, Z., Wang, W., Liang, K., Chen, J.: Pedestrian detection in fish-eye images using deep learning: combine faster R-CNN with an effective cutting method. In: Proceedings of the 2018 International Conference on Signal Processing and Machine Learning, pp. 55–59 (2018)
2. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2017)
3. Demirkus, M., Wang, L., Eschey, M., Kaestle, H., Galasso, F.: People detection in fish-eye top-views. In: VISIGRAPP (5: VISAPP), pp. 141–148 (2017)
4. Saeidi, M., Arabsorkhi, A.: A novel backbone architecture for pedestrian detection based on the human visual system. Vis. Comput. **38**(6), 2223–2237 (2022)
5. Wang, T., Chang, C.W., Wu, Y.S.: Template-based people detection using a single downward-viewing fisheye camera. In: 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 719–723. IEEE (2017)
6. Srisamosorn, V., Kuwahara, N., Yamashita, A., Ogata, T., Shirafuji, S., Ota, J.: Human position and head direction tracking in fisheye camera using randomized ferns and fisheye histograms of oriented gradients. Vis. Comput. **36**(7), 1443–1456 (2020)
7. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
8. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
9. Ding, J., Xue, N., Long, Y., Xia, G.S., Lu, Q.: Learning ROI transformer for oriented object detection in aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2849–2858 (2019)
10. Li, S., Tezcan, M.O., Ishwar, P., Konrad, J.: Supervised people counting using an overhead fisheye camera. In: 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–8. IEEE (2019)
11. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
12. Zhang, H., Hu, Z., Hao, R.: Joint information fusion and multi-scale network model for pedestrian detection. Vis. Comput. **37**(8), 2433–2442 (2021)
13. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
14. Tamura, M., Horiguchi, S., Murakami, T.: Omnidirectional pedestrian detection by rotation invariant training. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1989–1998. IEEE (2019)
15. Yu, J., Seidel, R., Hirtz, G.: OmniPD: one-step person detection in top-view omnidirectional indoor scenes. Curr. Direct. Biomed. Eng. **5**(1), 239–244 (2019)
16. Wei, L., Cui, W., Hu, Z., Sun, H., Hou, S.: A single-shot multi-level feature reused neural network for object detection. Vis. Comput. **37**(1), 133–142 (2021)
17. Kumar, V.R., Eising, C., Witt, C., Yogamani, S.: Surround-view fisheye camera perception for automated driving: overview, survey and challenges. arXiv preprint arXiv:2205.13281

18. Chiang, A.T., Wang, Y.: Human detection in fish-eye images using HOG-based detectors over rotated windows. In: IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6. IEEE (2014)
19. Van Tuan, N., Nguyen, T.B., Chung, S.T.: ConvNets and AGMM based real-time human detection under fisheye camera for embedded surveillance. In: International Conference on Information and Communication Technology Convergence (ICTC), pp. 840–845. IEEE (2016)
20. Wang, T., Hsieh, Y.Y., Wong, F.W., Chen, Y.F.: Mask-RCNN based people detection using a top-view fisheye camera. In: International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 1–4. IEEE (2019)
21. Wang, T., Chang, C. W., and Wu, Y. S.: Template-based people detection using a single downward-viewing fisheye camera. In: International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 719-723. IEEE (2017)
22. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2011)
23. Krams, O., Kiryati, N.: People detection in top-view fisheye imaging. In: 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2017)
24. Duan, Z., Tezcan, O., Nakamura, H., Ishwar, P., Konrad, J.: RAPiD: rotation-aware people detection in overhead fisheye images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 636–637 (2020)
25. Chiang, S.H., Wang, T., Chen, Y.F.: Efficient pedestrian detection in top-view fisheye images using compositions of perspective view patches. Image Vis. Comput. **105**, 104069 (2021)
26. Scaramuzza, D., Ikeuchi, K.: Omnidirectional Camera (2021)
27. Magnier, B., Comby, F., Strauss, O., Triboulet, J., Demonceaux, C.: Highly specific pose estimation with a catadioptric omnidirectional camera. In: 2010 IEEE International Conference on Imaging Systems and Techniques, pp. 229–233. IEEE (2010)
28. Kumler, J.J., Bauer, M.L.: Fish-eye lens designs and their relative performance. In Current developments in lens design and optical systems engineering. Int. Soc. Opt. Photon. **4093**, 360–369 (2000)
29. Hansen, P., Corke, P., Boles, W.: Wide-angle visual feature matching for outdoor localization. Int. J. Robot. Res. **29**(2–3), 267–297 (2010)
30. Boui, M., Hadj-Abdelkader, H., Ababsa, F.E., Bouyakhf, E.H.: New approach for human detection in spherical images. In *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 604–608. IEEE (2016)
31. Haggui, O., Bayd, H., Magnier, B., Aberkane, A.: Human Detection in Moving Fisheye Camera using an Improved YOLOv3 Framework. IEEE MMSP,: IEEE 23rd International Workshop on Multimedia Signal Processing, Oct 2021. Tampere, Finland (2021)
32. Haggui, O., Agninoube Tchalim, M., Magnier, B.: A Comparison of openCV algorithms for human tracking with a moving perspective camera. In: 9th European Workshop on Visual Information Processing (EUVIP), 2021, pp. 1–6. https://doi.org/10.1109/EUVIP50544.2021.9483957.
33. Li, S., Tezcan, M.O., Ishwar, P., Konrad, J.: Supervised people counting using an overhead fisheye camera. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–8. IEEE (2019)
34. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, pp. 740–755. Springer, Cham (2014)
35. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT 2010, , pp. 177–186. Physica-Verlag HD (2010)
36. Ramezani-Kebrya, A., Khisti, A., Liang, B.: On the generalization of stochastic gradient descent with momentum. arXiv preprint arXiv:2102.13653 (2021)
37. Chiang, A.T., Wang, Y.: Human detection in fish-eye images using HOG-based detectors over rotated windows. In: IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6. IEEE (2014)
38. Yu, H., Liu, W., Zhang, S., Yuan, H., Zhao, H.: Moving object detection using an in-vehicle fish-eye camera. In: 2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM). IEEE, pp. 1–6 (2010)
39. Chen, Z., Georgiadis, A.: Learning rotation sensitive neural network for deformed objects detection in fisheye images. In: 2019 4th International Conference on Robotics and Automation Engineering (ICRAE). IEEE, p. 125–129 (2019)
40. Duan, Z., Tezcan, O., Nakamura, H., Ishwar, P., Konrad, J.: RAPiD: rotation-aware people detection in overhead fisheye images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 636–637 (2020)
41. Tamura, M., Horiguchi, S., Murakami, T.: Omnidirectional pedestrian detection by rotation invariant training. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1989–1998. IEEE (2019)
42. Tamura, M., Horiguchi, S., Murakami, T.: Omnidirectional pedestrian detection by rotation invariant training. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1989–1998. IEEE (2019)
43. Demirkus, M., Wang, L., Eschey, M., Kaestle, H., Galasso, F.: People detection in fish-eye top-views. In : VISIGRAPP (5: VISAPP), pp. 141–148 (2017)
44. Krams, O., Kiryati, N.: People detection in top-view fisheye imaging. In: 2017 14th IEEE International conference on advanced video and signal based surveillance (AVSS), pp. 1–6. IEEE (2017)
45. Li, S., Tezcan, M.O., Ishwar, P., Konrad, J.: Supervised people counting using an overhead fisheye camera. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–8. IEEE (2019)
46. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
47. Baek, I., Davies, A., Yan, G., Rajkumar, R.R.: Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 447–452. IEEE (2018)
48. Wang, T., Chang, C.W., Wu, Y.S.: Template-based people detection using a single downward-viewing fisheye camera. In: 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 719–723. IEEE (2017)
49. Saito, M., Kitaguchi, K., Kimura, G., Hashimoto, M.: Human detection from fish-eye image based on probabilistic appearance model. Trans. Soc. Instrum. Control Eng. **49**(3), 319–325 (2013)
50. Wang, W., Gee, T., Price, J., Qi, H.: Real time multi-vehicle tracking and counting at intersections from a fisheye camera. In: 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 17–24. IEEE (2015)
51. Xiaozhe, W., Wei, L., Changkai, W., Shuying, Z., Tak, U.K., Yunzhou, Z.: An improved particle filter tracking algorithm for fisheye camera. In: 2016 Chinese Control and Decision Conference (CCDC), pp. 329–332. IEEE (2016)
52. Vandewiele, F., Bousetouane, F., Motamed, C.: Occlusion management strategies for pedestrians tracking across fisheye camera networks. In: 7th International Conference on Distributed Smart Cameras (ICDSC). IEEE, pp. 1-6 (2013)
53. Jianhui, W., Guoyun, Z., Shuai, Y., Longyuan, G., Mengxia, T.: Study the moving objects extraction and tracking used the moving

blobs method in fisheye image. In: Chinese Conference on Pattern Recognition, pp. 255–265. Springer, Berlin (2014)
54. Huang, C., Zhou, X., Chen, S., Shao, Z., Li, Y. F.: Correlation filter based fisheye video target tracking with adaptive weighted feature integration. In: 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 543–548. IEEE (2017)
55. Bertozzi, M., Castangia, L., Cattani, S., Prioletti, A., Versari, P.: 360 detection and tracking algorithm of both pedestrian and vehicle using fisheye images. In: 2015 IEEE Intelligent Vehicles Symposium (iv), pp. 132–137. IEEE (2015)
56. Srisamosorn, V., Kuwahara, N., Yamashita, A., Ogata, T., Shirafuji, S., Ota, J.: Human position and head direction tracking in fisheye camera using randomized ferns and fisheye histograms of oriented gradients. Vis. Comput. **36**(7), 1443–1456 (2020)
57. Ahmad, M., Ahmed, I., Khan, F. A., Qayum, F., Aljuaid, H. (2020) Convolutional neural network-based person tracking using overhead views. Int. J. Distrib. Sensor Netw. **16**(6), 1550147720934738 (2020)
58. Baek, I., Davies, A., Yan, G., Rajkumar, R.R.: Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 447–452. IEEE (2018)
59. Sio, C.H., Shuai, H.H., Cheng, W.H.: Multiple fisheye camera tracking via real-time feature clustering. In: Proceedings of the ACM Multimedia Asia, pp. 1–6 (2019)
60. Kubo, Y., Kitaguchi, T., Yamaguchi, J.I.: Human tracking using fisheye images. In: SICE Annual Conference 2007, pp. 2013–2017. IEEE (2007)
61. Chen, C.C., Wu, C.M., Shen, I.C., Chen, B.Y.: A deep learning based method for 3D human pose estimation from 2D fisheye images. In: Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, pp. 1–2 (2018)
62. Ahmed, I., Ahmad, M., Ahmad, A., Jeon, G.: Top view multiple people tracking by detection using deep SORT and YOLOv3 with transfer learning: within 5G infrastructure. Int. J. Mach. Learn. Cybern. **12**(11), 3053–3067 (2020)
63. Nguyen, H.D., Na, I.S., Kim, S.H., Lee, G.S., Yang, H.J., Choi, J.H.: Multiple human tracking in drone image. Multimed. Tools Appl. **78**(4), 4563–4577 (2019)
64. Ozer, S.: Visual object tracking in drone images with deep reinforcement learning. IEEE International Conference on Pattern Recognition, pp. 10082–10089 (2021)
65. Rosebrock, A.: Simple object tracking with OpenCV. PyImageSearch (2018)
66. Venkateswarlu, R., Sujata, K.V., Rao, B.V.: Centroid tracker and aimpoint selection. In: Acquisition, Tracking, and Pointing VI. International Society for Optics and Photonics, Vol. 1697, pp. 520–529 (1992)
67. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2544–2550. IEEE (2010)
68. Lukezic, A., Vojir, T., Cehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. IEEE Conference on Computer Vision and Pattern Recognition, pp. 6309-6318 (2017)

Olfa Haggui is a research fellow (postdoc) in computer vision and image processing at the University of IMT Mines Ales. She holds a PhD in computer science from MINES ParisTech (Paris, France) and ENISO National School of Engineers (Sousse, Tunisia) since June 2020. She received her engineering degree in 2011 and MSc in 2015; her research interests include image processing, computer vision, high-level program transformations for optimizing algorithm implementation on modern architectures, parallel computing, computer architecture (GPU, multi- and many-core architectures), machine learning, deep learning, algorithm optimization and parallel languages.

Hamza Bayd is a PhD student at EuroMov Digital Health in Motion, IMT Mines Ales, France. His PhD thesis focuses on the analysis of multi-scale synchronization during motion and music. He obtained a master's degree in embedded systems and computer vision at the National School of Applied Sciences of Tangier and in signal imaging and audio-video applications at the Faculty of Science and Engineering of Toulouse III. His research interests include computer vision, object detection architecture, signal processing, deep learning, music information retrieval, bio-mechanics, motion capture and synchronization.

Baptiste Magnier received his PhD degree from the Ales School of Mines, in 2011. He is currently an Associate Professor EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Ales, Ales, France. His research interests are in low-level feature extraction in image processing like edge/ridge detection and corner extraction. His researches contributed to the progress in color image steganalysis and, recently, object tracking. Moreover, he has developed anisotropic diffusion approaches involving partial differential equations for image segmentation or restoration.