Occlusion-Aware Pedestrian Detection and Tracking

Simão Correia[1], Diogo Mendes[1], Pedro Jorge[1], Tomás Brandão[1,2], Patrícia Arriaga[1,3] and Luís Nunes[1,2]

[1] ISCTE – University Institute of Lisbon, Lisbon, Portugal
[2] ISTAR-IUL – Information Sciences, Technologies and Architecture Research Center, Lisbon, Portugal
[3] CIS-Iscte – Center for Psychological Research and Social Intervention, Lisbon, Portugal

{simao_correia, diogo_amaro_mendes, pedro_antonio_jorge, tomas.brandao, patricia.arriaga, luis.nunes}@iscte-iul.pt

***Abstract*—This paper proposes an occlusion-aware mechanism, used on a framework for detecting and tracking pedestrians in videos acquired from surveillance cameras, which includes the extraction of trajectory points, estimation of walking velocities, detection of groups, and projection of the final trajectories into a 2D plan. The occlusion-aware mechanism is introduced in order to manage irregularities in the pedestrian trajectory data derived from occlusions. This mechanism is able to identify the parts of the human body that are occluded, using skeleton data generated by human pose estimation algorithms, and adjust the dimensions of the bounding boxes of the occluded pedestrians.**

***Index Terms*—object detection, multi-object tracking, trajectory extraction, pose estimation, computer vision, deep learning**

I. Introduction

Over the last few decades, with the rapid increase in computing power and the development of different Deep Learning techniques capable of autonomously learning features, many advances have been made in the field of Artificial Intelligence.

An area that has gained significant importance is Computer Vision, which covers techniques both in the image domain, such as image classification and object detection, and in the video domain, such as multi-object tracking and action recognition. However, several challenges still need to be overcome, including varying viewpoints and illumination, occlusions, and complex backgrounds.

The goal of this work is to develop a framework capable of detecting pedestrians in videos acquired from high-resolution surveillance cameras, as well as tracking them over time, so that relevant information can be extracted, including the paths taken by each pedestrian, the speed at which they walk, and the detection of groups of people. The main contribution of this study is the introduction of an occlusion-aware mechanism that is able to identify and adjust the dimensions of bounding boxes corresponding to occluded individuals.

The remainder of this paper is organized as follows: Section II provides a brief summary of the related work; Section III describes the dataset that was used in the experiments; Section IV presents an overview of the developed framework, and highlights details concerning implementation decisions as well as the algorithms used; Section V provides an analysis of the achieved results; and Section VI draws the main conclusions and suggestions for future work.

This work was partially funded by P2020 project "ECI4.0", ref. LISBOA-01-0247-FEDER-047155 and FCT - Fundação para a Ciência e Tecnologia, I.P. under project and UIDB/04466/2020 (ISTAR).

II. Related Work

The creation of multi-target multi-camera tracking datasets has become increasingly difficult due to the need to ensure data privacy. In this context, the authors of [1] introduced the Multi-Camera Track Auto (MTA) dataset, which was created using recordings made in the Grand Theft Auto V video game. The MTA dataset contains more than 2800 bots (people identities), and consists of a set of videos recorded on 6 different cameras, with more than 100 minutes of video per camera. In addition, the authors developed a baseline for the dataset, which consists of a framework with stages for detecting, re-identifying, and tracking bots, as well as calculating the distance traveled by each of them and associating trajectories.

In [2], the authors proposed the use of deep learning techniques to monitor social distancing, to prevent the increasing spread of the COVID-19 virus. Based on data acquired from video surveillance cameras, they used the YOLOv3 [3] algorithm for person detection, and the DeepSORT [4] algorithm to track each detected person over time. Furthermore, the authors compared YOLOv3 with other object detectors, namely Faster Region-based Convolutional Neural Network (Faster R-CNN) [5] and Single Shot MultiBox Detector (SSD) [6], concluding that YOLOv3 is the most efficient, in terms of speed-accuracy trade-off.

The authors of [7] proposed a system to perform pedestrian detection and tracking in surveillance videos. In order to detect each pedestrian, the authors used the YOLOv5 [8] algorithm, from which the center of the generated bounding boxes was inferred to estimate the social distance between people. As for tracking, the authors used the DeepSORT algorithm to obtain the complete trajectory of each pedestrian.

In [9], the authors proposed a modular pipeline for surveillance systems with the objective of early detection and prevention of suicide. The proposed system consisted of modules

for pedestrian detection, pedestrian tracking, pose estimation, and action recognition. For the first module, the authors used a pre-trained YOLOv5x model and performed fine-tuning using their own private dataset. Then, to track the detected people over time, they used the DeepSORT tracking algorithm, which was also employed to detect groups and generate information about the traveled trajectories. This information, together with pedestrian pose information extracted through the HRNet [10] pose estimation algorithm, was used to evaluate the occurrence of actions over a predefined duration, to infer risk behavior.

III. Dataset Description

For the purpose of this study, the VIRAT Video Dataset [11] was chosen, in particular the stationary footage from ground-based surveillance cameras, which consists of approximately 25 hours of video distributed across 16 different scenes. The videos were recorded in either 720p or 1080p, at a frame rate of 25 to 30 fps, varying according to the camera used.

Every video frame is annotated with the bounding boxes of the objects moving throughout the scene (i.e., people and cars), along with the events that occur over time. Furthermore, the homography [12] of each scene is also provided, in the form of projection matrices that allow the data acquired from the videos to be projected into the 2D floor plan.

IV. Methodology

Initially, each video is loaded and decoded into a sequence of frames, which are stored in a list. Then, for each frame in the list, an object detection algorithm is used to identify and locate pedestrians. Based on the bounding boxes generated by the object detector, a multi-object tracking algorithm is then used to assign a numerical identifier (ID) to each person over time, and an occlusion-aware mechanism is applied to correct the dimensions of the occluded bounding boxes. With these adjustments, trajectory points are extracted, and an estimate of the speed at which each person walks is calculated based on the previous points. Furthermore, taking into account aspects such as the distance between pedestrians and the scale of the bounding boxes, groups are inferred. Finally, based on the trajectory points that were registered throughout the video, a projection is created in the 2D floor plan (top view) with all the paths that were traveled by the pedestrians.

A. Pedestrian Detection

To detect each person in the video frames, object detection algorithms can be employed. The concept of these algorithms consists of identifying the objects present in a given image, returning, for each object: the bounding boxes with the coordinates of its location, its class prediction, and the confidence score associated to the detection (from 0.0 to 1.0).

You Only Look Once (YOLO), whose first version was proposed in [13], is one of the most commonly used approaches for this task. Its architecture consists of a single Convolutional Neural Network (CNN) capable of simultaneously generating multiple bounding boxes, as well as class probabilities for each of them. The main advantages over other object detectors, such as R-CNN [14] (including Fast [15] and Faster [5] R-CNN), are that it is extremely fast, can take into account the context in which the objects are in the image, and is very generalizable in how it learns the representations of each object.

Throughout the years, several versions of the YOLO algorithm have been proposed, with version 8 currently available. This study uses version 5 (YOLOv5 [8]), given that it is currently in a stable state, shows good performance, and is easily accessible through the PyTorch Hub [16] repository of pre-trained models. More specifically, we chose the YOLOv5x6 model due to its ability to detect distant people, as it operates on high-resolution images (1280x1280 pixels).

B. Pedestrian Tracking

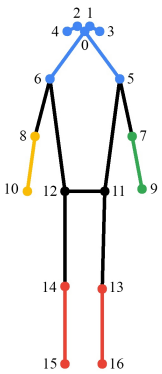
To obtain the location of each detected pedestrian over time, it is necessary to associate the bounding boxes that correspond to the same person, throughout the video frames. A numerical identifier (ID) is thus assigned to each tracked pedestrian. This is referred to as the tracking-by-detection paradigm. For this study, we decided to use the ByteTrack [17] algorithm, which is the current state of the art in multi-object tracking.

In order to associate the bounding boxes, ByteTrack leverages every detection, contrary to most methods, which discard those with low confidence scores. The reason for this method to include low-confidence detections is that many result from occlusions, which does not invalidate their usefulness. Hence, detections are selected based on the confidence score, using a threshold set to 0.6. With this in mind, for each video frame, the detected bounding boxes whose confidence score is greater than the threshold, are associated with the bounding boxes predicted using a Kalman filter [18] (based on information from previous frames). This association consists of computing motion or appearance similarity (i.e., Intersection Over Union (IoU) or Re-Identification (Re-ID), respectively), and applying the Hungarian method [19] to assign the IDs based on the similarity. If the associations are unsuccessful, then the process is repeated for the detections whose confidence scores are lower than the threshold, using IoU only, in order to resolve occlusion and background detection cases.

The components used in the Kalman filter to predict the tracks, based on measured locations and prior knowledge, are: $(x, y, a, h, v_x, v_y, v_a, v_h)$, where (x, y) is the center position of the bounding box, a is its aspect ratio, h is its height, and (v_x, v_y, v_a, v_h) are the corresponding velocities.

C. Occlusion Awareness

When certain obstacles are present, such as cars, lampposts, trees, bushes, etc., the bounding boxes generated by YOLOv5 simply surround the visible area of the occluded pedestrians. However, in some tasks, namely trajectory point extraction, it is essential to have an estimate of the location of the feet of each person. To address this issue, we developed a mechanism to detect whether or not an individual is occluded and, if so, automatically adjust the dimensions of its bounding box to include the occluded area.



upper part	0 - nose 1 - left eye 2 - right eye 3 - left ear 4 - right ear 5 - left shoulder 6 - right shoulder
left arm	7 - left elbow 9 - left wrist
right arm	8 - right elbow 10 - right wrist
lower part	13 - left knee 14 - right knee 15 - left ankle 16 - right ankle

Fig. 1. Correspondence between each body joint, obtained using the HRNet algorithm, and its respective body part. The hip body joints (11 and 12) were not taken into account for occlusion awareness.

Regarding pedestrian occlusion detection, we opted to use a pose estimation algorithm, more specifically, HRNet [10]. This algorithm consists of predicting where each person is located in an image by identifying and classifying their body joints, along with a body part ID and a confidence score (from 0.0 to 1.0). HRNet follows a top-down approach, which means that it uses information derived from object detection algorithms relative to the location of each detected individual in an image. With this in mind, the skeleton of each person is divided into 4 parts: upper part, lower part, left arm, and right arm. Having the body parts established, and access to the confidence score of each body joint, we establish that if there is any body joint with a low confidence score (a threshold of 0.3 was used), then the body part to which it belongs is considered occluded. An illustration of the body part division process is depicted in Fig. 1.

Depending on which part of the body is occluded, we apply different adjustments to the dimensions of the bounding box of the occluded person. Considering that a bounding box is defined as (x, y, w, h), where (x, y) are the coordinates of the top-left corner, and (w, h) are the width and height of the bounding box, respectively, if either the left or the right arm is occluded, the dimensions are adjusted as follows: $x' = x - \frac{w}{8}$, $w' = w + \frac{w}{4}$. In addition, if the lower part is occluded, a modified sigmoid function $\sigma(\alpha)$ (whose parameters were empirically set) is calculated as a means of limiting the amount by which the height of the bounding box is increased, taking into account its aspect ratio ($\alpha = \frac{w}{h}$). Thus, when the aspect ratio of the bounding box has a small value (e.g., 0.4) the height is not increased as much as when it has a larger value (e.g., 1.0). This process can be described as follows:

$$\sigma(\alpha) = \frac{1.5}{1 + e^{-7.5(\alpha - 0.75)}} \tag{1}$$

$$h' = h + (h \times \sigma(\alpha)) \ . \tag{2}$$

On the other hand, if the upper part is occluded, the y coordinate of the top left corner is decremented with the same value by which the height was increased:

$$y' = y - |h - h'| \ . \tag{3}$$

D. Trajectory Point Extraction

Based on the IDs assigned to each person by the ByteTrack algorithm, we are able to extract additional information regarding the paths traveled over time. To this end, according to the corrected bounding boxes, we store every point corresponding to the center of the bottom side (feet position).

However, due to oscillations in the dimensions of the bounding boxes, that occur due to the presence of certain obstacles or the manner in which each person moves, the extracted points end up showing some irregularities. To solve this issue, a smoothing method is applied. It consists of computing the moving average of the 5 most recent points (if less than 5, only the available points are considered). A recursive approach was followed, which means that the smoothing process applied to the current point considers the smoothing corrections applied to the previous.

In addition, as mentioned in Section III, the dataset includes information regarding the homography of each scene, in the form of projection matrices. This allows the transformation of the pixel coordinates in each video frame into their equivalent projections in the 2D floor plan. Therefore, for each generated trajectory point, its equivalent projection in the 2D floor plan is also stored.

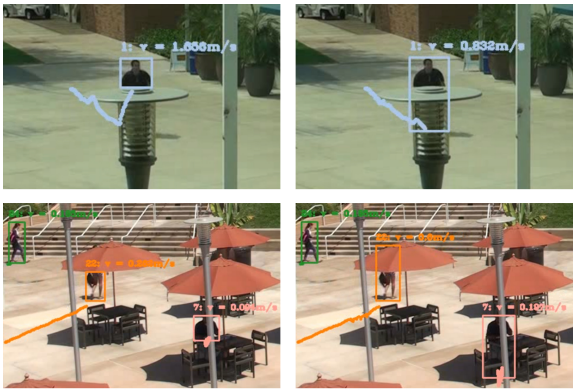
E. Velocities Calculation

In order to estimate the walking speed of each pedestrian, in a first approach, we tried to use the velocity information of the central position of the bounding box (vx, vy) provided by the Kalman filter, one of the components used in the ByteTrack algorithm, as reported in Section IV-B. However, since these values are inferred from the pixel coordinates of the video frames, the walking speed of pedestrians far from the camera turned out to be much lower than those close to it. To solve this problem, we decided to use the coordinates projected into the 2D floor plan, as a way to obtain a better approximation of the actual speed. It was calculated based on the distance traveled by each pedestrian in one second.

F. Group Detection

Another functionality that we found useful was to check if a detected pedestrian is alone or belongs to a group of people. To do so, based on the two bounding boxes of a pair of people, we analyze two criteria that need to be mutually confirmed. The first criterion is whether the distance between the centers of the two bounding boxes is less than 1.35 times the value of the maximum width. The second criterion is to measure the difference between the scales of the two bounding boxes, that is, to confirm that the result of dividing the area of the larger bounding box by the area of the smaller bounding box is less than 1.5.

After all pairs are associated, a check is performed with the goal of joining all pairs with common elements into a single group, marking it with a bounding box.



(a) (b)

Fig. 2. Examples of applying the proposed occlusion-aware mechanism: (a) without occlusion awareness (b) with occlusion awareness.

(a) (b)

Fig. 3. Example of reapplying the HRNet algorithm to the corrected bounding boxes (only body joints with a confidence score over the threshold of 0.3 are drawn): (a) before occlusion awareness (b) after occlusion awareness.

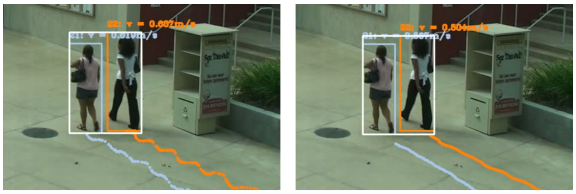
V. Results

A. Occlusion Awareness

As reported in Section IV-C, the occlusion-aware mechanism was developed to detect whether a person is occluded, and to adapt the dimensions of the bounding boxes to include the occluded area. Based on the results of several experiments, we were able to confirm its effectiveness in estimating the actual body boundaries of occluded pedestrians. Some examples are illustrated in Fig. 2.

Regarding the extraction of trajectory points, in cases where the lower part of the body was occluded, our method was able to predict the location of the feet of the occluded person, so as to improve the accuracy of the resulting trajectory. However, as the occlusion awareness mechanism takes into account the aspect ratio, and moving the arms influences the width of the bounding boxes, this created some noise in the trajectory. In addition, given that this method was developed specifically for pedestrian occlusions, in some cases, such as cyclist occlusions (where they usually lean forward and therefore the width of the bounding boxes is larger), it did not perform as well.

It is also worth mentioning that as the confidence scores of body joints are used to infer whether a body part is occluded, this may lead to some false positives, although their impact is not significant. On the other hand, when reapplying the HRNet algorithm to the corrected bounding boxes, the pose estimation results were improved, as illustrated in Fig. 3.



(a) (b)

Fig. 4. Example of applying the trajectory smoothing method to irregularities caused by gait movement: (a) without smoothing (b) with smoothing (5-point window).

(a) (b)

Fig. 5. Example of applying the trajectory smoothing method to irregularities caused by obstacles: (a) 5-point window (b) 10-point window.

B. Trajectory Smoothing

The smoothing method was applied to reduce the impact of oscillations on the trajectory points. To assess its effectiveness, we performed an analysis according to the factors that caused the oscillations. These include the way each person walks and the occurrence of occlusions.

Regarding the oscillations caused by the manner in which each person walks, the smoothing method was able to achieve good results in mitigating them. Fig. 4 shows an example of these results, with a window of 5 trajectory points.

However, when addressing irregularities caused by the presence of obstacles, a smoothing window of 5 trajectory points is not sufficient. Thus, we attempted to apply a window of 10 trajectory points. This eventually reduced the irregularities in the trajectories, although given that only past points are used, this caused a significant delay in the extracted points, implying that they no longer matched the location of the pedestrians. A comparison between using a window with 5 and 10 trajectory points is illustrated in Fig. 5.

VI. Conclusion

In this study, we developed a system for detecting and tracking pedestrians in outdoor spaces, through videos captured by high-resolution surveillance cameras.

The experiments conducted on the VIRAT dataset confirmed the effectiveness of the proposed occlusion-aware mechanism in detecting occluded body parts and adjusting the dimensions of the respective bounding boxes, thus improving the accuracy of the extracted trajectory points in the presence of occlusions.

The current version of the proposed mechanism only considers previously extracted trajectory points to smooth the current points. Thus, for future work, we aim to further improve the trajectory smoothing method by also considering information extracted from future frames.

REFERENCES

[1] P. Köhl, A. Specker, A. Schumann, and J. Beyerer, "The mta dataset for multi target multi camera pedestrian tracking by weighted distance aggregation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 4489–4498.

[2] N. S. Punn, S. K. Sonbhadra, S. Agarwal, and G. Rai, "Monitoring covid-19 social distancing with person detection and tracking via fine-tuned yolo v3 and deepsort techniques," *arXiv:2005.01385*, 2020.

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[6] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 21–37.

[7] Y. Wang and H. Yang, "Multi-target pedestrian tracking based on yolov5 and deepsort," in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2022, pp. 508–514.

[8] G. Jocher *et al.*, "ultralytics/yolov5," Zenodo, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[9] X. Li, S. Onie, M. Liang, M. Larsen, and A. Sowmya, "Towards building a visual behaviour analysis pipeline for suicide detection and prevention," *Sensors*, vol. 22, no. 12, 2022.

[10] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5686–5696.

[11] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, 2011, pp. 3153–3160.

[12] E. Dubrofsky, "Homography estimation," *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, vol. 5, 2009.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[15] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[16] "Pytorch hub," PyTorch Foundation, 2023. [Online]. Available: https://pytorch.org/hub/

[17] Y. Zhang *et al.*, "Bytetrack: Multi-object tracking by associating every detection box," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 1–21.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[19] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.