This is the peer reviewd version of the followng article:

First Steps Towards 3D Pedestrian Detection and Tracking from Single Image / Mancusi, G.; Fabbri, M.; Egidi, S.; Verasani, M.; Scarabelli, P.; Calderara, S.; Cucchiara, R.. - 13232:(2022), pp. 335-346. (Intervento presentato al convegno 21st International Conference on Image Analysis and Processing, ICIAP 2022 tenutosi a ita nel 2022) [10.1007/978-3-031-06430-2_28].

Springer Science and Business Media Deutschland GmbH
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

12/10/2024 09:07

(Article begins on next page)

First Steps Towards 3D Pedestrian Detection and Tracking from Single Image

Gianluca Mancusi[1,2], Matteo Fabbri[1,3], Sara Egidi[2], Mattia Verasani[2], Paolo Scarabelli[2], Simone Calderara[1], and Rita Cucchiara[1]

[1] University of Modena and Reggio Emilia, Italy
{firstname.lastname}@unimore.it
[2] Tetra Pak Packaging Solutions S.P.A.
{firstname.lastname}@tetrapak.com
[3] GoatAI S.r.l.
{firstname.lastname}@goatai.it

Abstract. For decades, the problem of multiple people tracking has been tackled by leveraging 2D data only. However, people move and interact in a three-dimensional space. For this reason, using only 2D data might be limiting and overly challenging, especially due to occlusions and multiple overlapping people. In this paper, we take advantage of 3D synthetic data from the novel MOTSynth dataset, to train our proposed 3D people detector, whose observations are fed to a tracker that works in the corresponding 3D space. Compared to conventional 2D trackers, we show an overall improvement in performance with a reduction of identity switches on both real and synthetic data. Additionally, we propose a tracker that jointly exploits 3D and 2D data, showing an improvement over the proposed baselines. Our experiments demonstrate that 3D data can be beneficial, and we believe this paper will pave the road for future efforts in leveraging 3D data for tackling multiple people tracking.

Keywords: Multiple-Object Tracking · Synthetic Data · 3D people detection

1 Introduction

People detection and tracking in crowded scenarios are highly challenging tasks with mature literature, with applications ranging from surveillance to autonomous driving. To effectively advance the field, the community has been adopting neural networks since the advent of deep learning [14, 26, 12, 1, 3, 11].

The research history of people detection and tracking in computer vision revolves around two-dimensional approaches, where the subjects are tracked on the image plane using 2D bounding box detections. However, humans move and interact in a three-dimensional world. Not considering this critical point might lead to overcomplicated solutions to problems that could have simpler answers in a three-dimensional space. For example, disambiguating people that occlude each other might be easier in 3D than in 2D. When two people are overlapped in the 2D image plane, their spatial location is typically separable in 3D [22].

However, people detection and tracking are still addressed in crowded scenarios using 2D data only. Common trackers ignore the targets' depth and only reason regarding image coordinates. The absence of methods that leverage 3D information is mainly due to the lack of available datasets that can provide adequate annotations. Only recently, promising works on 3D people tracking have been proposed [21, 22]. However, those methods are unsuitable for crowded scenarios as they are trained on non-crowded datasets and do not handle strong occlusions effectively.

Here, we take advantage of the recently proposed MOTSynth [8] dataset for designing a first attempt to exploit 3D annotations for people detection and tracking in crowded scenarios. MOTSynth is a large, synthetic dataset intended explicitly for the tasks of pedestrian detection and tracking that has already shown outstanding generalization capabilities in real-world scenarios.

Following LoCO [6] and CenterNet [36], we propose a novel 3D people detector that, given a single RGB image, is able to predict the image plane bounding boxes, as long as the camera distances of every pedestrian in the scene. The training scheme has been carried out leveraging synthetic data only (no fine-tuning on real data), while performances are evaluated on real datasets.

The newly provided detections are then used to perform an extensive study involving multiple trackers over real and synthetic test sets. Results show that 3D data is mostly beneficial, especially for sequences with strong occlusions. To the best of our knowledge, until now, no previous attempts to exploit 3D data to tackle people detection and tracking in crowded scenarios have been proven successful. We strongly believe that leveraging the third dimension could be a key factor in solving the most critical challenges of multiple people tracking.

2 Related works

Multiple Object Tracking. The problem of multiple object tracking has challenged computer vision researchers for many years. The techniques proposed are wide-ranging, and the most significant are summarized in [15]. It is possible to track any object, and there are trackers suitable for different targets. This is because tracked entities can have different types of motion and behavior. For example, ants follow a Brownian motion [10], while people walk more linearly. In this paper, we study tracking related to people in crowded scenarios, where people freely move and interact in indoor or outdoor locations.

One of the most popular trackers is SORT [2]. SORT is a barebones implementation of visual multiple objects tracking frameworks based on rudimentary data association and state estimation techniques. It serves as a baseline for more recent and sophisticated trackers. SORT is based on the well-known linear Kalman filter [24] to predict the state of the targets in the next time step. A famous extension of SORT is DeepSORT [29], which uses a deep neural network to compute re-ID features for the association step. In this work, we are not interested in exploiting visual cues, as we are more concerned with evaluating how the spatial location impacts the performance of the tracker.

More recently, multiple successful trackers have been proposed in the literature in the past few years [1, 3, 30, 31, 11, 21, 32, 18, 27, 33]. Despite the ever-growing literature in the tracking community, none of these recently proposed methods utilize 3D cues to improve the performance of their trackers.

On the other hand, some attempts have been made to exploit three-dimensional information for designing tracking methodologies. However, they assume multiple camera setups [19, 20, 13, 34], or dedicated sensors [25, 28].

Only recently, promising works that target 3D people detection and tracking have been proposed [21, 22]. However, those methods do not perform well in crowded scenarios, as they have been modeled on non-crowded datasets. For this reason, they do not handle strong occlusions effectively.

In this work, instead, we advocate for a novel approach that fully exploits 3D information for the task of multiple people tracking in particularly crowded scenarios from single RGB cameras.

Prediction of People's Distance. Predicting the camera distance of a person from a monocular image has always been challenging in computer vision, but with the advent of deep learning, it has become possible to tackle the problem more effectively. In particular, many works perform a multi-person 3D human pose estimation by predicting the distance and the pose of every person [17, 35, 9]. However, those methods are not suitable for crowded scenarios peculiar of surveillance applications.

Among the recently proposed 3D methods for assessing the people's camera distance, the paper of Fabbri *et al.* [6] is the most relevant to our work. The proposed LoCO architecture can handle 3D multi-person human pose estimation from a single image in crowded scenarios. In this work, we modified the LoCO architecture to deliver multiple 3D people detection instead of multiple pose estimation.

MOT Datasets. Visual surveillance-centric datasets aim at crowded scenarios where pedestrians are interacting and occluding each other. MOTChallenge [5] is the reference benchmark for assessing the performance of multi-object tracking methods, as it provides consistently labeled crowded tracking sequences. In particular, MOT17 [16] and MOT20 [4] provide challenging sequences of crowded urban scenes, capturing severe occlusions and scale variations. However, none of the datasets provided in the MOTChallenge suite provide 3D annotations.

Among the synthetic datasets, JTA [7], and its improved version MOTSynth [8], are the most relevant ones for surveillance applications. JTA and MOTSynth have been collected utilizing the Grand Theft Auto V video game, which simulates a city and its inhabitants in a three-dimensional world. The provided sequences are highly photorealistic while providing temporally consistent bounding boxes and instance segmentation labels, 3D poses with occlusion information and depth maps.

3 Method

The following subsections summarize the key elements of our approach. Following the tracking-by-detection paradigm, we propose splitting the problem into two distinct tasks: detection and tracking. Section 3.1 describes our proposed 3D detector, while Section 3.2 illustrates our tracking methodology.

3.1 3D People Detection

At the core of our proposal lies our 3D people detector: LoCO-Det. We modified the original architecture in Fabbri *et al.* [6] to perform 3D people detection rather than 3D pose estimation.

The original architecture of LoCO consists of two separate networks, a Volumetric Heatmap Autoencoder (VHA), which compresses three-dimensional volumetric tensors $\mathfrak{H}$, representing the 3D space into a compressed 2D code $e(\mathfrak{H})$, and a Code Predictor, which takes the RGB image I of shape $3 \times H \times W$ as input and predicts the compressed 2D codes $f(I)$.

VHA is trained with the ground truth three-dimensional volumetric tensors $\mathfrak{H}$ to minimize the reconstruction loss. Once the VHA training converges, the 2D codes $e(\mathfrak{H})$ obtained from the VHA encoder are used as supervision signals for the Code Predictor, which is trained using RGB images. At testing time, the 2D codes $f(I)$ computed by the Code Predictor are fed to the VHA decoder to obtain the original three-dimensional volumetric representation $\tilde{\mathfrak{H}} = d(f(I))$.

In the original version of LoCO, 14 volumetric tensors are predicted, one for each human joint. Our idea, inspired by the CenterNet [36] approach and based on LoCO, is to predict three volumetric tensors instead of 14 volumetric heatmaps. More specifically, our representation consists of one tensor for the centers, one tensor for the widths, and one tensor for the heights of the bounding boxes. The new input of the modified VHA architecture takes the following structure:

$$\mathfrak{M} = [\boldsymbol{C}, \boldsymbol{S_w}, \boldsymbol{S_h}]^T$$

where $\boldsymbol{C}$ is the tensor containing the centers of the bounding boxes, $\boldsymbol{S_w}$ and $\boldsymbol{S_h}$ are tensors containing, in the same positions of the corresponding centers, the value of width and height of the bounding box.

Given a person k, we define the 2D bounding box $\boldsymbol{b}_k = (u_k, v_k, w_k, h_k)$, where u_k and v_k are the image plane coordinates of the center, w_k and h_k are the width and height respectively. For each person, we also define the camera distance $d_k = \sqrt{x_k^2 + y_k^2 + z_k^2}$ where x_k, y_k, and z_k are the 3D coordinates of the person position.

We further define a 2.5D location $\boldsymbol{p}_k = (u_k, v_k, d_k)$ where $u_k \in \{1, ..., H'\}$ and $v_k \in \{1, ..., W'\}$ are respectively the row and column indexes of the center pixel on the image plane, and $d_k \in \{1, ..., D\}$ is the quantized distance from the camera. In our experiments, we used $D = 316$, $H' = 136$ and $W' = 240$.

For $\boldsymbol{C}$, the value $\mathfrak{c}_k$ at a generic location $\boldsymbol{p}$ is obtained by centering a Gaussian with sigma $\sigma = 2$ in $\boldsymbol{p}_k$:

$$\mathfrak{c}_k(\boldsymbol{p}) = e^{-\frac{\|\boldsymbol{p} - \boldsymbol{p}_k\|^2}{\sigma^2}}$$

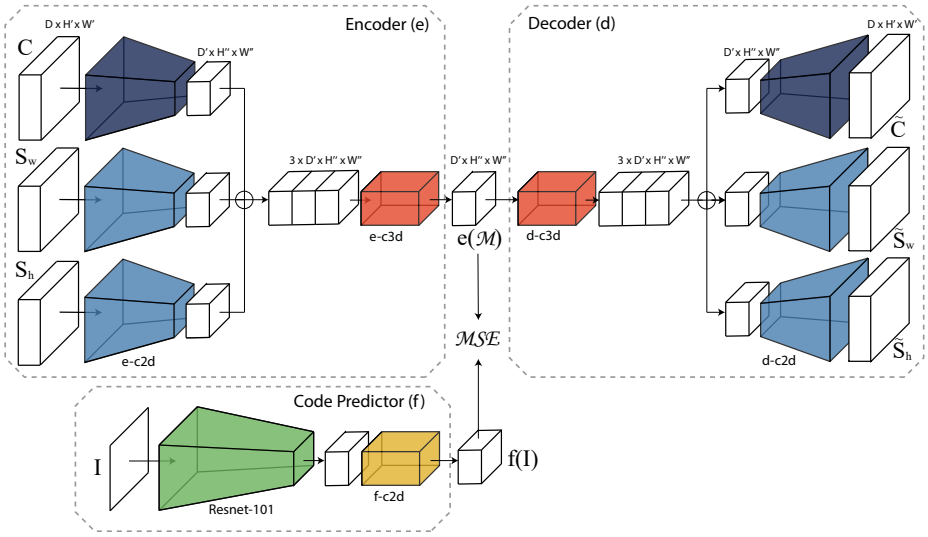


Fig. 1. Schematization of the proposed LoCO-Det pipeline. At training time, the Encoder e produces the compressed representation $e(\mathfrak{M})$ which is used as ground truth from the Code Predictor f. At test time, the intermediate representation $f(I)$ computed by the Code Predictor is fed to the Decoder d for the final output. In our case, $H' = H/8$ and $W' = W/8$.

It is possible to re-obtain the coordinates Q_C of the centers as follows:

$$Q_C = \bigcup_{n=1,\dots,N} \{\boldsymbol{p} : \mathfrak{c}_n(\boldsymbol{p}) > \boldsymbol{p}' \quad \forall \boldsymbol{p}' \in \mathfrak{N}_{\bar{p}}\} \tag{1}$$

where $\mathfrak{N}_{\bar{p}}$ is the 6-connected neighborhood of $\bar{\boldsymbol{p}}$, i.e. the set of coordinates $\mathfrak{N}_{\bar{p}} = \{\boldsymbol{p} : \|\boldsymbol{p} - \bar{\boldsymbol{p}}\| = 1\}$ at unit distance from $\bar{\boldsymbol{p}}$.

For $\boldsymbol{S_w}$ and $\boldsymbol{S_h}$, the values $\mathfrak{s}_{\mathfrak{w}\,k}$ and $\mathfrak{s}_{\mathfrak{h}\,k}$ at a generic location $\boldsymbol{p}$ are obtained by centering a sphere in $\boldsymbol{p}_k$. In our experiment, we used a diameter of 5. The values inside the spheres are w_k and h_k respectively.

We modified the VHA architecture in LoCO by reducing the e-c2d and d-c2d blocks from 14 to 3. The blocks associated to width and height have their weight shared. We further modified both e-c3d and d-c3d by replacing the two 3D convolutions in each block with one 3D convolutional layer. Fig. 1 top depicts our modified version of the VHA, called VTA (Volumetric Tensor Autoencoder). Finally, the VTA is trained by minimizing the loss function defined as follows:

$$L = \lambda_c L_{\text{MSE}}(\boldsymbol{C}, \tilde{\boldsymbol{C}}) + \lambda_w \; L_{\text{MASK}}(\boldsymbol{S}_w, \tilde{\boldsymbol{S}}_w) + \lambda_h \; L_{\text{MASK}}(\boldsymbol{S}_h, \tilde{\boldsymbol{S}}_h)$$

where L_{MSE} is the MSE loss function, L_{MASK} is the masked MSE loss function and $\lambda_c \in \mathbb{R}$, $\lambda_w \in \mathbb{R}$, $\lambda_h \in \mathbb{R}$ are scaling weights.

The Code Predictor, Fig. 1 bottom, is composed by a pre-trained feature extractor (convolutional part of Resnet-101), and a fully convolutional block

(f-c2d) composed of four convolutions. We trained the Code Predictor by minimizing the MSE loss between $f(I)$ and $e(\mathfrak{M})$, where $\mathfrak{M}$ is the ground truth representation associated with the image I.

During evaluation, the $Q_{\tilde{C}}$ coordinates obtained from the predicted $\tilde{\boldsymbol{C}}$ are used to look up the corresponding width and height values in $\boldsymbol{S_w}$ and $\boldsymbol{S_h}$. Finally, by applying the pinhole camera model to the computed locations $Q_{\tilde{C}}$, we are able to obtain the 3D coordinates in the standard camera system.

3.2 Tracking in 3D

Our first attempt in addressing multiple people tracking with 3D data consists in extending the SORT baseline. To this end, we modified the representation and the motion model used to propagate a target's identity into the next frame. As for SORT, we approximate the inter-frame displacements of each object with a linear constant velocity model. The state s of each target is modeled as:

$$s = (x, y, z, h, w, \dot{x}, \dot{y}, \dot{z})$$

where x, y, and z are the 3D standard camera system coordinates, h and w represent the height and width of the 2D bounding box. When detection is associated with a target, the detected 3D coordinates and bounding box sizes are used to update the target state and the velocity components are solved optimally via a Kalman filter. If no detection is associated with the target, its state is simply predicted without correction using the linear velocity model.

In assigning detections to existing targets, each target's bounding box geometry is estimated by projecting the 3D coordinates into the image plane and using width and height to create the predicted 2D bounding box. Thus, predicting its new location in the current frame. Following SORT, the assignment cost matrix is then computed as the intersection-over-union (IoU) distance between each detection and all predicted bounding boxes from the existing targets. The assignment is solved optimally using the Hungarian algorithm.

4 Experiments

The following subsections describe the experiments and results obtained in this work. In section 4.1 the implementation details of the LoCO-Det architecture and the related detection results are presented, while in section 4.2 the experiments carried out with different trackers and the results obtained in the comparison between 2D and 3D are discussed.

4.1 LoCO-Det Results and Implementation Details

As explained in section 3.1, the VTA is trained by taking as input 3 volumetric tensors, one containing the centers of the bounding boxes, one containing the width values, and one containing the height values. During the VTA training, affine transformations, i.e. scaling and translation, are applied to the input

Dataset	Detector	MODA	MODP	AP	Rcll	Prcn	TP	FP	FN	F1	FAR
MOT17	LoCO-Det (3D)	59.00	77.85	0.57	**68.36**	87.96	**45387**	6215	**21006**	**76.93**	1.17
(train-set)	Yolov3 (2D)	**60.14**	**82.82**	**0.61**	64.58	**93.57**	42875	**2946**	23518	76.42	**0.55**
MOTSynth	LoCO-Det (3D)	50.71	79.27	0.50	57.26	89.72	3110471	356203	2321461	69.91	1.03
(test-set)	Yolov3 (2D)	**58.35**	**87.66**	**0.53**	**59.87**	**97.52**	**3266731**	**83170**	**2189196**	**74.19**	**0.24**

Table 1. Metrics computed on MOT17 and MOTSynth datasets to evaluate the LoCO-Det detector after projecting the 3D detections into the 2D image plane. Comparisons are made against Yolov3, which directly provides 2D detection.

Observations	Tracker	MOTA ↑	HOTA ↑	IDF1 ↑	IDSW ↓	FP ↓	FN ↓
Ground-Truth	NN-2D	88.9	58.5	47.9	205592	732713	90095
	NN-3D	**90.6**	**76.7**	**67.7**	**65850**	**717358**	**90412**
LoCO-Det	NN-2D	28.6	16.1	11.7	267344	265950	6103199
	NN-3D	**28.9**	**16.7**	**12.5**	**258167**	**260773**	**6091272**

Table 2. The Nearest Neighbor (NN) tracker on MOTSynth test set evaluated with the detections provided by LoCO-Det and ground truth detections.

tensors. During the Code Predictor training, the affine transformations of the augmentation applied to the volumetric tensors $\mathfrak{M}$ are also used to the corresponding image I.

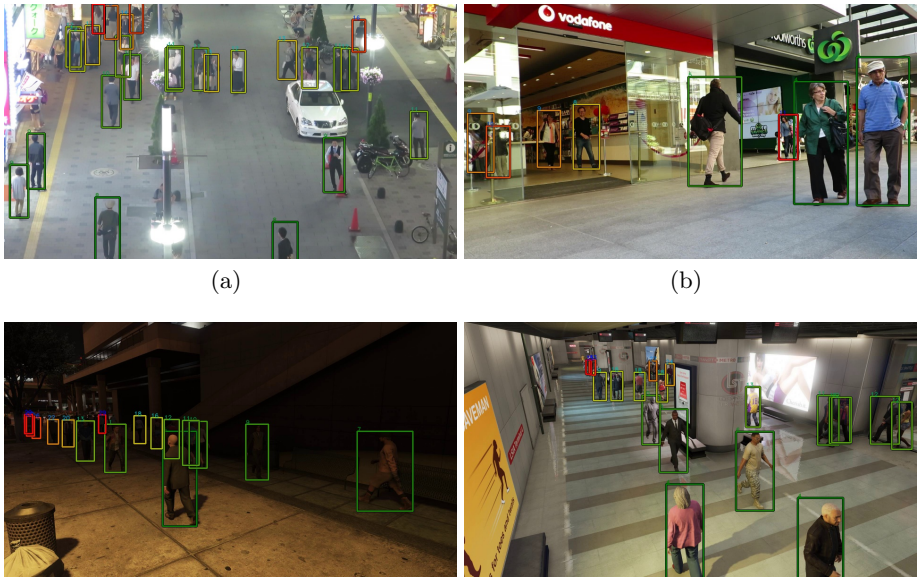
MOTSynth pedestrians with less than 20% of visible joints are ignored. If the head joint is visible, we consider the annotation even if it does not have 20% of visible joints. We utilized an ADAM optimizer to train the Code Predictor with a learning rate of 0.0001 and a batch size of 12. The Code Predictor takes as input the MOTSynth images halved to 960×540.

Some results of LoCO-Det compared to Yolov3 [23] are shown in Table 1. Both architectures are trained exclusively on MOTSynth and evaluated on MOTSynth and MOT17.

Performance is measured using 2D detection metrics, as MOT17 only provides 2D annotations. From the table, it can be seen that both Yolov3 and LoCO-Det perform comparably on MOT17. However, on MOTSynth, Yolov3 outperforms LoCO-Det. The reason is because Yolov3 is a much more robust 2D detector. Additionally, LoCO-Det has a more complex task to solve: estimating the camera distances of every pedestrian in the scene. Some qualitative results on MOT17 and MOTSynth are depicted in Fig. 2.

4.2 Tracking 2D vs 3D

To study how 3D trackers can be better than 2D trackers, we decided to utilize one of the simplest trackers, the Nearest Neighbor tracker (NN). The reason behind this adoption is that we first want to study the association performance of the tracker in the 3D space, neglecting the next state prediction performance.



(a) (b)

(c) (d)

Fig. 2. Qualitative results of LoCO-Det. The bounding box distance from the camera increases as the color of the bounding box changes from green to red. Images (a, b) are taken from MOT17. While (c, d) are taken from MOTSynth.

Nearest Neighbor tracker. The NN tracker considers the current frame bounding box centers and computes the distance to the previous frame's bounding box centers. Bounding boxes at a minimum distance will be part of the same tracklet. We implemented one version that utilizes the bounding box centers in 2D and another version that utilizes them in 3D.

The results in Table 2 show results obtained on MOTSynth. In the ideal case, i.e., utilizing ground truth detections, the best-performing NN tracker is the one that exploits 3D information. This indicates that, by improving the detector, the 3D tracker can significantly outperform its 2D counterpart. By replacing the ground truth detection with the ones obtained with LoCO-Det, the 3D tracker still exceeds the 2D one. Using the observations provided by LoCO-Det and evaluating on MOT17, we see in Table 3 that the 3D tracker, even in the real case, can outperform the 2D tracker.

Table 3 similarly shows the performances on MOT17, divided by sequence. The only sequence of MOT17 where the 2D tracker significantly outperforms the 3D tracker is sequence 4, which is captured with a high angle of view. In this case, the 3D tracker does not bring a significant advantage, as people rarely occlude each other.

SORT 3D. We defined the SORT tracker as our baseline, which we compared against our first 3D adaptation called SORT-3D. Here, the Kalman filter per-

	Tracker	MOTA ↑	HOTA ↑	IDF1 ↑	IDSW ↓	FP ↓	FN ↓
MOT17-02	NN-2D	18.7	22.4	17.3	**342**	1077	**13672**
	NN-3D	**18.8**	**23.9**	**18.9**	345	**1042**	13699
MOT17-04	NN-2D	**51.5**	**32.6**	**26.9**	**1087**	**1767**	20211
	NN-3D	50.7	28.3	21.6	1388	1931	**20109**
MOT17-05	NN-2D	39.9	**37.4**	**34.5**	341	336	**3478**
	NN-3D	**40.9**	35.1	33.8	**295**	**288**	3504
MOT17-09	NN-2D	59.6	34.1	30.2	185	**50**	1911
	NN-3D	**61.5**	**41.1**	**36.7**	**144**	54	**1851**
MOT17-11	NN-2D	**33.5**	23.5	19.1	358	451	7725
	NN-3D	34.2	**27.7**	**23.3**	**299**	**428**	**7721**
MOT17-13	NN-2D	60.4	43.8	37.5	186	250	3298
	NN-3D	**61.0**	**46.0**	**40.4**	**156**	**249**	**3278**

Table 3. Results of the NN tracker on MOT17, divided by sequence.

forms the prediction in the 3D space, neglecting w and h, which are only updated with the new observations. The Hungarian algorithm is applied to a cost matrix created using 3D Euclidean distances between the centers of the targets and the new observations. The maximum distance between two centers is set to 2 meters. Tracklets are deleted after being lost for 11 frames. For a tracklet to be valid, it must be matched to an observation at least twice.

The experiments in Table 4 show that with an ideal detector, i.e., using ground truth observations, SORT-3D is much better than SORT, especially regarding IDSW. However, using the observations from LoCO-Det, SORT-3D is outperformed by the traditional implementation of SORT (2D version). This is because SORT associates targets and new observations utilizing the IoU. The IoU allows the association step to be aware not only of the location of people but also of their shape. Instead, during the associations, SORT-3D only considers the distance between 3D centers that might have noise.

SORT 2D + 3D. To take the most out of both 2D and 3D information, we designed a tracker that predicts the new state in the 3D space, then moves into 2D space for the association phase, as explained in section 3.2. The IoU threshold used in our experiments is 0.15.

Our newly proposed SORT 2D+3D tracker gets better results than the classic version of SORT, as shown in Table 5. The improvement, although not pronounced, demonstrates that 3D data can be a viable solution to tackle multiple people tracking with another perspective.

Detector	Tracker on MOTSynth	MOTA ↑	HOTA ↑	IDF1 ↑	IDSW ↓	FP ↓	FN ↓
Ground-Truth	SORT 2D (IoU-based)	86.7	68.6	59.8	134033	705092	396998
	SORT 3D (distance-based)	85.7	**76.2**	**67.4**	**56263**	910784	**356740**
	SORT 2D+3D (IoU)	**86.8**	69.7	60.9	131011	**701964**	397139
LoCO-Det	SORT 2D (IoU-based)	**29.0**	21.8	18.1	160415	181044	**6266115**
	SORT 3D (distance-based)	27.2	19.9	16.6	198500	234060	6344827
	SORT 2D+3D (IoU)	**29.0**	**21.9**	**18.3**	**159389**	**180777**	6267953

Table 4. Comparison on MOTSynth between SORT 2D based on IoU association, SORT 3D based on Euclidean Distance association, and SORT 2D+3D which performs the Kalman filter predictions in 3D and project the bounding boxes in 2D, performing an IoU association.

Tracker on MOT17-train	MOTA ↑	HOTA ↑	IDF1 ↑	IDSW ↓	FP ↓	FN ↓
SORT 2D (IoU-based)	**41.6**	40.7	37.8	1512	**3189**	**60889**
SORT 3D (distance-based)	39.3	35.8	32.5	2337	3766	62008
SORT 2D+3D (IoU)	**41.6**	**41.0**	**38.1**	**1497**	3210	60928

Table 5. Comparison on MOT17 using the same trackers of Table 4

5 Discussion

In this paper, we proposed an effective way of addressing 3D people detection from single images. We trained our novel LoCO-Det on synthetic data only without fine-tuning on real data and providing state-of-the-art results on real data.

We further show that 3D information can be beneficial, especially when people are occluding each other, as ID switches are consistently higher when relying on 2D data only.

Finally, we showed a first attempt at designing a competitive tracker able to perform 3D reasoning.

6 Acknowledgements

Partially supported by the PREVUE "PRediction of activities and Events by Vision in an Urban Environment" project (CUP E94I19000650001), PRIN National Research Program, Italian Ministry for Education, University and Research (MIUR), by ROADSTER "Road Sustainable Twins in Emilia Romagna" project, International Foundation Big Data and Artificial Intelligence for Human Development, and by Tetra Pak Packaging Solutions S.P.A.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. "Tracking without bells and whistles". In: *ICCV*. 2019.

[2] Alex Bewley et al. "Simple Online and Realtime Tracking". In: *2016 IEEE International Conference on Image Processing (ICIP)* (Sept. 2016).

[3] Guillem Brasó and Laura Leal-Taixé. "Learning a neural solver for multiple object tracking". In: *CVPR*. 2020.

[4] Patrick Dendorfer et al. "Mot20: A benchmark for multi object tracking in crowded scenes". In: *arXiv preprint arXiv:2003.09003* (2020).

[5] Patrick Dendorfer et al. "Motchallenge: A benchmark for single-camera multiple target tracking". In: *International Journal of Computer Vision* 129.4 (2021), pp. 845–881.

[6] Matteo Fabbri et al. "Compressed volumetric heatmaps for multi-person 3d pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7204–7213.

[7] Matteo Fabbri et al. "Learning to detect and track visible and occluded body joints in a virtual world". In: *ECCV*. 2018.

[8] Matteo Fabbri et al. "MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?" In: *International Conference on Computer Vision (ICCV)*. 2021.

[9] Taosha Fan et al. "Revitalizing Optimization for 3D Human Pose and Shape Estimation: A Sparse Constrained Formulation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[10] Deborah M. Gordon, Richard E. Paul, and Karen Thorpe. "What is the function of encounter patterns in ant colonies?" In: *Animal Behaviour* 45.6 (1993), pp. 1083–1100. ISSN: 0003-3472.

[11] Yanru Huang et al. "Sqe: a self quality evaluation metric for parameters optimization in multi-object tracking". In: *CVPR*. 2020.

[12] Chanho Kim, Fuxin Li, and James M. Rehg. "Multi-object Tracking with Neural Gating Using Bilinear LSTM". In: *ECCV*. 2018.

[13] Oh-Hun Kwon, Julian Tanke, and Juergen Gall. "Recursive Bayesian Filtering for Multiple Human Pose Tracking from Multiple Cameras". In: *Proceedings of the Asian Conference on Computer Vision*. 2020.

[14] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. "Learning by Tracking: Siamese CNN for Robust Target Association". In: *CVPR Workshops* (2016).

[15] Wenhan Luo et al. "Multiple object tracking: A literature review". In: *Artificial Intelligence* 293 (2021), p. 103448.

[16] Anton Milan et al. "MOT16: A Benchmark for Multi-Object Tracking". In: *arXiv preprint arXiv:1603.00831* (2016).

[17] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image". In: *ICCV*. 2019, pp. 10133–10142.

[18] Jiangmiao Pang et al. "Quasi-Dense Similarity Learning for Multiple Object Tracking". In: (June 2021).

[19] Nam Trung Pham, Weimin Huang, and S. H. Ong. "Probability Hypothesis Density Approach for Multi-camera Multi-object Tracking". In: *Computer Vision ACCV 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[20] Kha Gia Quach et al. "DyGLIP: A Dynamic Graph Model With Link Prediction for Accurate Multi-Camera Multiple Object Tracking". In: *CVPR*. June 2021, pp. 13784–13793.

[21] Jathushan Rajasegaran et al. "Tracking People by Predicting 3D Appearance, Location & Pose". In: *ArXiv* abs/2112.04477 (2021).

[22] Jathushan Rajasegaran et al. "Tracking People with 3D Representations". In: *NeurIPS*. 2021.

[23] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[24] Donald B. Reid. "An Algorithm for Tracking Multiple Targets". In: *IEEE Transactions on Automatic Control* 24 (1979), pp. 843–854.

[25] Seiichi Sato et al. "Multilayer lidar-based pedestrian tracking in urban environments". In: *2010 IEEE Intelligent Vehicles Symposium*. IEEE. 2010, pp. 849–854.

[26] Jeany Son et al. "Multi-object Tracking with Quadruplet Convolutional Neural Networks". In: *CVPR*. 2017.

[27] Pavel Tokmakov et al. "Learning to Track with Object Permanence". In: (2021).

[28] Xinshuo Weng et al. "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning". In: *CVPR*. 2020, pp. 6499–6508.

[29] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. "Simple Online and Realtime Tracking with a Deep Association Metric". In: *ICIP*. IEEE. 2017, pp. 3645–3649.

[30] Yihong Xu et al. "How To Train Your Deep Multi-Object Tracker". In: *CVPR*. 2020.

[31] Junbo Yin et al. "A unified object motion and affinity model for online multi-object tracking". In: *CVPR*. 2020.

[32] Fangao Zeng et al. "MOTR: End-to-End Multiple-Object Tracking with TRansformer". In: *arXiv preprint arXiv:2105.03247* (2021).

[33] Yifu Zhang et al. "ByteTrack: Multi-Object Tracking by Associating Every Detection Box". In: *arXiv preprint arXiv:2110.06864* (2021).

[34] Yuxiang Zhang et al. "4D association graph for realtime multi-person motion capture using multiple video cameras". In: *CVPR*. 2020, pp. 1324–1333.

[35] Ce Zheng et al. "3D Human Pose Estimation With Spatial and Temporal Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 11656–11665.

[36] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. "Objects as points". In: *arXiv preprint arXiv:1904.07850* (2019).