Depth-aware Multi-Object Tracking in Spherical Videos

Liliana Lo Presti[1][0000−0003−0833−4403], Giuseppe Mazzola[2][0000−0003−3839−9312], Guido Averna[1][0000−0002−0935−5866], Edoardo Ardizzone[1][0000−0002−3096−8253], and Marco La Cascia[1][0000−0002−8766−6395]

[1] Engineering Department, University of Palermo, Palermo, Italy
[2] Department of Humanities, University of Palermo, Palermo, Italy
liliana.lopresti@unipa.it; giuseppe.mazzola@unipa.it;

Abstract. This paper deals with the multi-object tracking (MOT) problem in videos acquired by 360-degree cameras. Targets are tracked by a frame-by-frame association strategy. At each frame, candidate targets are detected by a pre-trained state-of-the-art deep model. Associations to the targets known till the previous frame are found by solving a data association problem considering the locations of the targets in the scene. In case of a missing detection, a Kalman filter is used to track the target. Differently than works at the state-of-the-art, the proposed tracker considers the depth of the targets in the scene. The distance of the targets from the camera can be estimated by geometrical facts peculiar to the adopted 360-degree camera and by assuming targets move on the ground-plane. Distance estimates are used to model the location of the targets in the scene, solve the data association problem, and handle missing detection. Experimental results on publicly available data demonstrate the effectiveness of the adopted approach.

Keywords: multi-object tracking · equirectangular image · 360 degree videos · depth.

1 Introduction

Multi-object tracking (MOT) is the problem of analyzing a video, namely a sequence of images, and detecting the locations on the image plane of targets moving in the environment over time. The problem can be extended to 3D when a world coordinate system is known or is estimated, and is especially challenging in a crowd scenario where objects with similar appearances are difficult to discriminate from each other. In these cases, not all targets are detected due to partial/total occlusions, and false positives have to be handled. Handling occlusions can take advantage of knowing the distance of the targets from the camera. Objects close to the camera can still be detected while the farthest ones require algorithms to predict their location on the image plane.

Recently, the work in [16] showed how to estimate the distance from the camera of objects of interest given only the height of the 360° camera and the coordinates, on the spherical image, of the contact point of the target with the

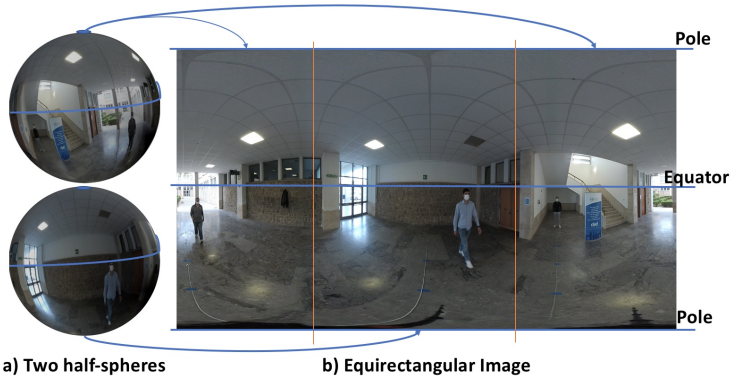

Fig. 1. The figure shows on the left two half-spheres, each one acquired by a lens in a dual lens 360°camera. On the right, the equirectangular image on which the spherical one is projected. The middle line represents the equator of the spherical image, while the uppest and lowest rows are the sphere poles.

ground plane. 360° camera devices acquire panoramic images with a view spanning 360° horizontally and 180° vertically. Recent devices are typically formed by a system of two wide-angle lenses, each of which can shoot (more than) half of the scene. In the acquired spherical images, pixels are mapped onto a sphere centered into the camera. Spherical images can be stored by applying equirectangular projections [6], after correcting the distortion introduced by lenses, if their shape is known. As shown in Fig. 1, the central row of the equirectangular is the equator, and the upper and lower rows are the poles. This projection introduces a distortion, which is more visible approaching the poles.

Tracking in equirectangular images is challenging due to the image's circularity and high resolution. The former is a consequence of the sphericity of the image and implies that a target walking around the camera and depicted close to the left/right border of the image may re-appear at the opposite right/left side of the image itself. High resolution of the image is required to represent the entire scene without losing too many details.

We propose to exploit geometrical facts about the equirectangular projection to estimate and track the targets' locations onto the ground plane by taking advantage of the estimated distances of the targets from the camera. In Fig. 2, four targets are moving around the scene. Green bounding boxes represent ground-truth information, while the colored ones are the tracker's predictions (in this case only two). The radial plot on the left shows the estimated targets' locations on the ground plane. The center of the plot is the camera location. Concentric circles are the loci of points equidistant from the camera. Circles are exactly one meter apart from each other. Numbers are identifiers (color-coded) associated with the targets and are close to their location on the ground.

To the best of our knowledge, this is the first multi-object tracker for 360° videos that explicitly estimates and uses targets' distances from the camera to track the targets. In this respect, we propose a novel depth-aware multi-object tracker that does not need any calibration procedure.

Fig. 2. At the center, an extended equirectangular image (dotted areas are included to enhance target detection). Green bounding-boxes are ground-truth, colored ones are tracker's predictions. The image shows two pedestrians far more than 15 meters from the camera. They are annotated but not detected. On the left, the plot shows the locations on the ground-plane of the two tracked objects. On the right, the plot shows the locations of the annotated objects. Pedestrians distant more than 15 meters from the camera are not shown.

The only assumption of our approach is planar motion of the targets on the ground-plane. We stress that if the camera height is known, exact targets' locations on the ground plane can be derived. This characteristic is very appealing in several applications, especially in surveillance but also in behavior understanding applications to better understand how people interact. We tested our tracker on the publicly available dataset [16] by adapting the MOT evaluation protocol to spherical images. We also implemented a baseline technique to compare with. The plan of the paper is as follow. In Sec. 2 we summarize state-of-the-art work in multi-object tracking; In Secs. 3 and 4 we present our novel approach and implementation details respectively. Finally, in Sec. 5 we discuss experimental results and in Sec.6 we present conclusions and future work.

2 Related Work

Solution of MOT problem relies on the matching, at time t, between observed objects and targets detected at time $t-1$. Information used to establish the matches can be about object appearance, 2D/3D location, or trajectory and is incrementally refined over time, including the possibility of dynamically changing the number of targets to account for objects entering/exiting in/from the scene.

Matching can be found by two main approaches [1]: frame-by-frame associations of observations to targets and deferred-logic tracking. With the former sequential tracking strategy, at each frame, the most likely associations between object detections and targets are estimated and not modified anymore. The problem is modeled as a data association (DA) one, and often linear programming techniques involving constrained optimization problems are adopted [2, 11, 12].

Global Nearest Neighbor Standard Filter [11, 1] considers all assignments within a region of interest and solves a maximal bipartite matching problem to find the best assignments between detections and targets, generally by the Munkres algorithm.

In deferred-logic tracking, observation-to-target associations are delayed until evidence of their correctness is accumulated. Multiple Hypotheses Tracking (MHT) [20] builds a hypothesis tree whose root-leaf paths represent all possible combinations of detection-target associations through time. The exponential growth of the tree is avoided by pruning heuristics, and the path with the highest likelihood is selected as the correct target track.

In this paper, we adopt a frame-by-frame association strategy. At each frame, we detect candidate targets and associate them with the identities known until the previous frame. Considering that we are processing spherical images, and we can estimate the distance from the camera of the detected targets, we solve the DA problem through the Munkres algorithm by considering the targets' depth while accounting for the image circularity.

A recent survey of MOT works using deep learning is in [5]. In general, these works [24, 23, 4] use a deep model for solving the detection problem, the Munkres algorithm to solve the DA problem, and prediction methods such as Kalman filter to track the targets through the occlusions. Deep models can additionally be used to estimate similarities [23] or model motion [4].

Works about tracking in 360-degree videos [17] focus mostly on single object tracking by applying deep learning strategies [15, 9] and Kalman or particle filters [19, 18, 3]. The work most similar to ours is the one in [14], which is built upon DeepSORT [22]. DeepSORT solves the DA problem by considering deep features extracted by the detector (YOLO v2). In contrast to our approach, in [14], detections are taken on image slices to account for the high image resolution. Similar to our approach, the method computes the detections on the extended frame in order to account for the image circularity. With respect to this work, we model tracking by considering the distances from the camera when tracking pedestrians, and use them to handle occlusions.

3 Modeling Targets' Locations on Spherical Images

Despite targets are detected on the equirectangular image, in our approach tracking is performed on the ground-plane. On the image plane, the target's ground-touching point is approximated by the middle point of the lower side of the bounding box detected on the image. By geometrical facts, the distance of the target from the camera is estimated and, together with the azimuth angle, provides the polar coordinates of the target on the ground in a coordinate system centered onto the camera ground location. In this section we provide details on the adopted reference systems.

3.1 360° Videos and Equirectangular Images

As shown in Fig. 1, a 360° camera device can acquire panoramic images with a view spanning 360° horizontally and 180° vertically, and can represent the surrounding environment at each shot. This is of interest in several fields such as video surveillance, robotics, and cultural heritage applications.

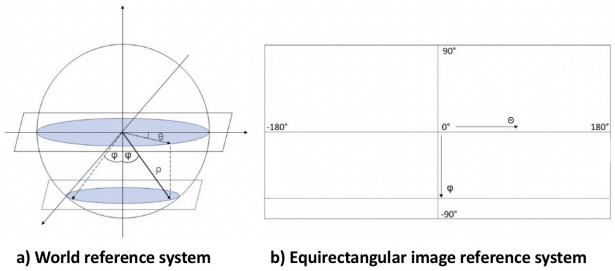

Fig. 3. The image shows the world coordinate system on the left, and the equirectangular coordinate system on the right.

Regardless of the type of 360° device, pixels of the images are mapped onto a sphere centered into the 360° camera. Equirectangular and cubic projections are often adopted to allow displaying the image on monitors or viewers [6]. While cubic projections map the spherical points onto the plane tangent to the sphere, equirectangular projection maps the whole sphere to a single image. In particular, the central row of the equirectangular image represents the sphere equator; the uppermost and lowermost rows correspond to the sphere poles. In general, each row of the equirectangular image corresponds to the intersection between the sphere and a plane parallel to the horizontal plane of the camera [16].

Pixel coordinates (x_r, y_r) on the equirectangular image represent normalized values of polar and azimuth angles of the corresponding point on the sphere surface. The angles can be recovered from the pixel coordinates by a simple rescaling and shifting such that the polar angle ϕ ranges in $[-90°, 90°]$, while the azimuth angle θ ranges in $[-180°, 180°]$ (see Fig. 3). Of course, by this projection, the radial coordinate of the spherical coordinate system cannot be preserved.

3.2 Estimating Targets' distances and locations

The algorithm to estimate the targets' distances from the camera [16] is based on pure geometrical facts and is uncalibrated, as there is no need to estimate the camera parameters. The algorithm takes advantage from a simple fact: all points of a plane parallel to the horizontal camera plane and equally distant from the camera are projected onto the same row of the equirectangular image. The only hypothesis that must hold to apply the method is that target move on the ground plane, and that the ground plane is parallel to the horizontal camera plane. Furthermore, it must be possible to measure the touching point of the target with the ground on the spherical image (the target must be visible). We define the touching point $P = (x_L, y_L)$ as the middle one in the lowest side of the bounding box enclosing the target on the equirectangular image.

As shown in Fig. 4, the distance d of the object of interest from the camera can be estimated as [16]:

$$d = h_c \cot \alpha \tag{1}$$

where h_C is the camera height and α is the angle between the camera plane passing through the sphere equator and the line through the camera center and

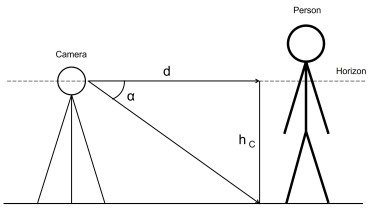

Fig. 4. The figure shows how the distance of the target from the camera can be estimated. h_c is the camera height (in meters). α is the angle between the camera horizontal plane and the line passing through the target's ground-touching point and the camera center. d is the distance of the target from the camera and can be estimated by trigonometrical equations.

the target's touching point (see Fig. 4). With this formulation, the distance d is the length of one of the two catheti of the resulting right triangle, and is related to the other one by the trigonometric formula 1.

The angle α is estimated from the point P on the equirectangular image:

$$\alpha = \frac{\frac{h}{2} - y_L}{\frac{h}{2}} \cdot 90^\circ \tag{2}$$

where h is the equirectangular image height (in pixels).

For each target, we can model its location considering the polar coordinates (d, θ) of its touching point. We then model the location of the target in Cartesian coordinates as $l = (d\cos\theta, d\sin\theta)$ on the ground-plane.

4 Proposed MOT Algorithm

Our tracking strategy relies on a simple and very common MOT strategy that aims at updating the targets' locations by associating the targets' predicted locations with the ones provided by a pre-trained pedestrian detector. Hence, the main steps in our algorithm are: pedestrian detection, prediction of the targets' location, and data association. Differently than other works, in our algorithm locations are modeled directly on the ground and expressed in a coordinate system centered in the camera's ground-touching point. Another advantage of our formulation is that it naturally accounts for the circularity of the equirectangular image due to the adopted reference coordinate system.

4.1 Pedestrian Detection

Similar to [14], we run a state-of-the-art detector directly on the equirectangular image. Of course, deformations affect the detection process but we have experimentally found that the loss of accuracy is negligible. Since we are interested in pedestrians, we ignore objects from other classes.

A simple approach to account for the image circularity is expanding the image as shown in Fig. 2 and removing duplicated bounding boxes. Let us assume that the image is originally $W \times H$ while the extended image is $W' \times H'$. Duplicated

bounding boxes are removed by shifting all of them to the left by W. Then, those with positive coordinates on the image plane are retained and matched with the non-shifted ones by intersection-over-union. Finally, the most internal ones are selected. In our experiments, we have adopted the Faster-RCNN[8] detector.

4.2 Modeling and Predicting Targets' Locations

We model targets' trajectories as discrete-time linear dynamical systems. The state of the system represents the location of the target on the ground-plane and its velocity $s_t = (x_t, y_t, v_t^x, v_t^y)$, modeled as continuous variables. We use Kalman filter (KF) to make the state evolve over time in two steps: prediction and update steps. The prediction step allows to make the state evolve over time based on the knowledge of the past state. At each discrete time, the state s_{t-1} is linearly combined to generate the new state s_t by also accounting for some Gaussian noise $w_{t-1} \sim N(0, \Sigma_s)$. In our model, no external control signal is needed and we assumed targets move based on a uniform linear motion model.

The update step uses the difference between predictions and observations to refine the state estimate.

KF is a recursive filter where prediction and update steps alternate to progressively refine the current state estimation. In particular, as detailed in [7], the filter can be viewed as a weighted average estimator where noise covariance matrices and Kalman-Gain matrix are iteratively computed to refine the state estimations. Whenever an observation is unavailable, no updating step is performed and KF is used to predict the target location. In the long run, this can of course yield to drifting of the tracker.

4.3 Data Association

Since we use a frame-by-frame association approach, we need to associate detections with known targets at each frame. We also need to decide when a new target has to be included in the pool of tracked objects, and when a target is exiting the scene. Associations are estimated by using ground-plane coordinates found through the estimates of targets' distances from the camera. As already explained, this approach naturally accounts for the image circularity.

As usually done, we model MOT as a minimum weight matching problem in bipartite graphs. In a bipartite graph, vertices are grouped into two disjoint sets such that no two vertices within the same set are adjacent. A matching is a subset of edges of the graph that do not share common vertices. The minimum weight matching has minimal sum of the edge weights.

In our framework, the vertices of one set of the bipartite graph represent the targets' locations predicted at time t by Kalman filter, while the vertices in the other set represent the locations of the pedestrian found by the detector on the image plane and converted into ground coordinates as detailed in Sec. 3.2. Each vertex of the first set can potentially match any other vertex in the second set. Edge weights measure the dissimilarity between targets and detections considering both their locations on the ground plane and the appearance features. Appearance features are extracted by using the first flattened layer of a

pre-trained CNN (ResNet-50 [10] trained on Imagenet in our experiments), but any other appearance descriptor may be used as well.

We fuse appearance features and locations by considering that the two data vectors have different lengths and values varying in different ranges. Given a target with a predicted location $P_t^T \in R^2$ and a detected bounding box with estimated touching point $P_t^D \in R^2$ at time t, the dissimilarity $l(T, D)$ between the two locations is defined as

$$l(T, D) = \frac{1}{2}||P_t^T - P_t^D||_1. \tag{3}$$

Assuming the appearance features of the target and the detection are $F_{t-1}^T \in R^L$ and $F_t^D \in R^L$ respectively, the dissimilarity $f(T, D)$ between the two feature vectors is defined as

$$f(T, D) = \frac{1}{L}||F_{t-1}^T - F_t^D||_1. \tag{4}$$

In the above dissimilarity scores, L1-norm is used since it is more meaningful and efficient than L2-norm in case of high-dimensional data. We compute $l(T, D)$ and $f(T, D)$ for all pairs of n targets and m detected bounding boxes. To make the L1-norm values comparable, we estimate the z-scores of the two sets of $n \times m$ norm values, $z^l(T, D)$ and $z^f(T, D)$ respectively. Hence, the final dissimilarity score for a pair is obtained as the minimum between $z^l(T, D)$ and $z^f(T, D)$. With this score, the tracker associates a target to a detection or because the locations are close or because the appearance features are similar.

4.4 Trackers' birth, death and updating

At each frame, two cases can arise: new targets enter the scene (or false positives are detected) or some targets exit the scene (or are not detected, for instance due to occlusions). To account for such cases, we adopt the strategy in [11], and augment the two sets of vertices in the bipartite graph with "fake" vertices. The vertices added to the target set represent potentially new targets entering the scene. The vertices added to the detection set account for missing. By including these fake vertices, each set will count $n + m$ nodes.

With these additional vertices, it is necessary to set a default weight value for the fake edges. This value is especially important for the success of the tracking strategy since it represents an error tolerance on the dissimilarity score of the matches and, hence, defines the search area on the ground-plane and in the appearance feature space. We have experimentally found that a value of 2 works pretty well. This value is the superior limit of the 95% of the confidence interval of the estimated z-scores.

Once the data association problem is solved, we use the associations to update the targets' locations and the corresponding Kalman filters. When no association is found, we maintain the Kalman filter prediction as target's location. We keep the target alive for T frames ($T = 90$ in our experiments) such that the target can be tracked through potential occlusions. However, this strategy increases the risks of tracking false positives. To limit this issue, when a new target identity is

discovered, we wait for K frame ($K = 3$ in our experiments) before adding this new target to the pool of tracked objects.

5 Experimental results

We performed tests on the publicly available CVIP360 dataset [16]. The dataset includes two sets of videos. The first set includes 11 videos acquired indoor, while the second one includes 6 videos collected outdoor. Overall, the dataset includes about 18K frames with more than 50K annotated bounding boxes.

We use the CLEAR metrics [13], including MOTA, FP, FN, ID-switches to assess the tracking quality. We also report the IDF1 score [21] that evaluates the identity preservation ability and focuses more on the association performance.

Performance have been evaluated with a modified version of the py-motmetrics library. Since images are circular, we assessed tracking on the ground-plane.

Our baseline method, *Baseline (image)*, implements a standard MOT technique by using the same detector and data association technique as our tracker but models trajectories on the image plane. It is not able to account for the image circularity. In case of missing, the target's location is not updated.

We present ablation studies by constructing our tracker step-by-step. The tracker *Ours (image, circularity)* tracks targets on the image plane and accounts for the image circularity by an ad-hoc matching process. Duplicated bounding boxes on the extended frame are not filtered out but are used to solve data association by retaining the minimum dissimilarity score. This approach requires more comparisons but accounts for the image circularity. This method is relatively close to [14], except for the detection strategy, as explained in section 2. In our ablation study, we compare trackers on equal terms of pedestrian detector. We detect pedestrians on the whole image, focusing on the tracking strategy.

The tracker *Ours (ground)* tracks on the ground-plane and can naturally account for the image circularity. Duplicated bounding boxes are not used for data association. The tracker *Ours (ground + KF)* uses Kalman filter to track targets in case of missing detection. Results of the above techniques are presented with and without using appearance features.

5.1 Results

Tables 1 and 2 show our experimental results in indoor and outdoor videos respectively. The tables report average values of the metrics over the test videos. FP, FN and ISDW are average raw values (not percentage). We note that the number of FP and FN (missing) in the various experiments are similar because they are run on equal terms of detector. In indoor videos, best results are achieved by modeling tracking on the ground.

Despite the appearance model is weak, it contributes to a small improvement of the performance. In outdoor videos, all methods are comparable. Without appearance features, KF improves the IDF1 score but not the MOTA due to a small increase of the ID-switches. In outdoor videos, appearance does not help the tracker and, while MOTA scores are comparable, IDF1 decreases. This might

Table 1. Results on the indoor video dataset

Tracker	IDF1↑	MOTA↑	FP↓	FN↓	IDSW↓
Baseline (image)	92.78	93.22	163.09	49.27	41.18
Ours (image, circularity)	92.86	94.95	74.36	49.27	35.09
Ours (ground)	93.66	**96.78**	61.73	49.27	7
Ours (ground + KF)	**94.21**	96.10	61.72	49.27	**5.73**
Baseline (image + appearance)	93.82	95.88	60.45	49.63	18.54
Ours (image + appearance, circularity)	93.82	95.89	60.45	49.64	17.36
Ours (ground + appearance)	94.00	96.05	61.82	49.82	**8**
Ours (ground + appearance + KF)	**94.61**	**96.19**	53.63	49.81	10.18

Table 2. Results on the outdoor video dataset

Tracker	IDF1↑	MOTA↑	FP↓	FN↓	IDSW↓
Baseline (image)	86.5	88.8	81.16	317.67	67.33
Ours (image, circularity)	86.83	88.18	120.67	338.33	58.3
Ours (ground)	86.6	**89.21**	47.5	393.3	**12.3**
Ours (ground + KF)	**89.52**	89.69	33.5	393.33	12.83
Baseline (image + appearance)	**88.6**	**89.62**	49	337.5	34.3
Ours (image + appearance, circularity)	88.30	89.38	49	365.67	24.17
Ours (ground + appearance)	85.8	88.90	64	393.22	16
Ours (ground + appearance + KF)	87.28	89.43	49.33	393.33	**15.17**

be due to the fact that generally appearance features are sensitive to illumination variations. The decreases of the performance when using ground-plane coordinates in outdoor videos might be due to the uncertainty in the touching-point estimate on the image plane, especially for the farthest pedestrians. Indeed, in outdoor, distance from the camera can be more than 10 meters. One of such cases is shown in Fig. 2 (pedestrian with ID 3). The detector may be unable to detect the pedestrian and, if it does, the bounding box is not much reliable.

Despite the limitations of the detection step, all experiments show that tracking on the ground-plane decreases the number of ID-switches. In practice, especially in case of occlusions, is much easier for the tracker to solve ambiguities.

6 Conclusions and Future Work

This paper proposes a MOT tracker that estimates targets' locations on the ground-plane by using the distance of the targets from the 360 degrees camera. Such distance is estimated by the method in [16], which is uncalibrated and works under mild hypotheses. Targets' locations on the ground-plane are used to solve data association and model trajectories. We experimentally found that measuring the location on the ground is required to properly measure performance of any tracker in 360°videos. Our preliminary results show that, when the targets' touching point can be reliably estimated on the image, ground-plane coordinates improve tracking especially in case of occlusions.

In future work we will study how to improve the detection of farthest objects in order to have a better estimation of the touching point. We will also improve appearance and motion models to handle occlusions.

7 Acknowledgement

This research was partially funded by MIUR grant number PRIN I-MALL 2017BH297_004 and Italian PON IDEHA grant ARS01_00421.

References

1. Betke, M., Wu, Z.: Data association for multi-object visual tracking. Synthesis Lectures on Computer Vision **6**(2), 1–120 (2016)
2. Castanon, D.A.: Efficient algorithms for finding the k best paths through a trellis. IEEE Transactions on Aerospace and Electronic Systems **26**(2), 405–410 (1990)
3. Chen, G., St-Charles, P.L., Bouachir, W., Bilodeau, G.A., Bergevin, R.: Reproducible evaluation of pan-tilt-zoom tracking. In: 2015 IEEE International Conference on Image Processing (ICIP). pp. 2055–2059. IEEE (2015)
4. Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., Yu, N.: Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In: Proc. of the Int. Conf. on Computer Vision. pp. 4836–4845 (2017)
5. Ciaparrone, G., Sánchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., Herrera, F.: Deep learning in video multi-object tracking: A survey. Neurocomputing (2019)
6. Corbillon, X., Simon, G., Devlic, A., Chakareski, J.: Viewport-adaptive navigable 360-degree video delivery. In: 2017 IEEE international conference on communications (ICC). pp. 1–7. IEEE (2017)
7. Funk, N.: A study of the kalman filter applied to visual tracking. University of Alberta, Project for CMPUT **652**(6) (2003)
8. Girshick, R.: Fast r-cnn. In: Proc. of the Int. Conf. on computer vision. pp. 1440–1448 (2015)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proc. of Int. Conf. on computer vision. pp. 2961–2969 (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Huang, T., Russell, S.: Object identification in a bayesian context. In: IJCAI. vol. 97, pp. 1276–1282. Citeseer (1997)
12. Jiang, H., Fels, S., Little, J.J.: A linear programming approach for multiple object tracking. In: Conf. on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
13. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
14. Liu, K.C., Shen, Y.T., Chen, L.G.: Simple online and realtime tracking with spherical panoramic camera. In: 2018 IEEE International Conference on Consumer Electronics (ICCE). pp. 1–6. IEEE (2018)
15. Lo Presti, L., La Cascia, M.: Deep motion model for pedestrian tracking in 360 degrees videos. In: International Conference on Image Analysis and Processing. pp. 36–47. Springer (2019)
16. Mazzola, G., Lo Presti, L., Ardizzone, E., La Cascia, M.: A dataset of annotated omnidirectional videos for distancing applications. Journal of Imaging **7**(8), 158 (2021)
17. Mi, T.W., Yang, M.T.: Comparison of tracking techniques on 360-degree videos. Applied Sciences **9**(16), 3336 (2019)
18. Monteleone, V., Lo Presti, L., La Cascia, M.: Pedestrian tracking in 360 video by virtual ptz cameras. In: 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI). pp. 1–6. IEEE (2018)
19. Monteleone, V., Lo Presti, L., La Cascia, M.: Particle filtering for tracking in 360 degrees videos using virtual ptz cameras. In: International Conference on Image Analysis and Processing. pp. 71–81. Springer (2019)

20. Reid, D.: An algorithm for tracking multiple targets. IEEE transactions on Automatic Control **24**(6), 843–854 (1979)
21. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European conference on computer vision. pp. 17–35. Springer (2016)
22. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
23. Xu, J., Cao, Y., Zhang, Z., Hu, H.: Spatial-temporal relation networks for multi-object tracking. In: Int. Conf. on Computer Vision (October 2019)
24. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J.: Poi: Multiple object tracking with high performance detection and appearance feature. In: ECCV. pp. 36–42. Springer (2016)