2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

PanNote: an Automatic Tool for Panoramic Image Annotation of People's Positions

Alberto Bacchin[1,*], Leonardo Barcellona[1,2], Sepideh Shamsizadeh[1], Emilio Olivastri[1], Alberto Pretto[1], Emanuele Menegatti[1]

***Abstract*— Panoramic cameras offer a 4π steradian field of view, which is desirable for tasks like people detection and tracking since nobody can exit the field of view. Despite the recent diffusion of low-cost panoramic cameras, their usage in robotics remains constrained by the limited availability of datasets featuring annotations in the robot space, including people's 2D or 3D positions. To tackle this issue, we introduce *PanNote*, an automatic annotation tool for people's positions in panoramic videos. Our tool is designed to be cost-effective and straightforward to use without requiring human intervention during the labeling process and enabling the training of machine learning models with low effort. The proposed method introduces a calibration model and a data association algorithm to fuse data from panoramic images and 2D LiDAR readings. We validate the capabilities of *PanNote* by collecting a real-world dataset. On these data, we compared manual labels, automatic labels and the predictions of a baseline deep neural network. Results clearly show the advantage of using our method, with a 15-fold speed up in labeling time and a considerable gain in performance while training deep neural models on automatically labelled data.**

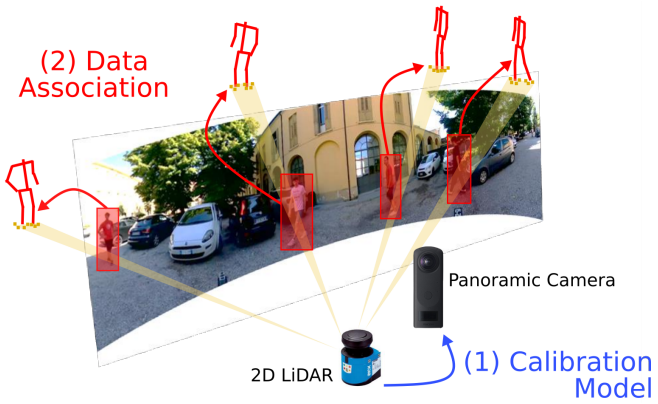

Fig. 1: *PanNote* is an automatic annotation tool for people's position in panoramic images. The proposed method includes (1) a 2D LiDAR and panoramic camera calibration model and (2) a data association algorithm. Through sensor fusion, we are able to annotate people's position in robot space without human intervention.

I. INTRODUCTION

The recent diffusion of commercial omnidirectional cameras opened new opportunities in robotics. Modern cameras are more compact and robust than the old catadioptric cameras while providing high-resolution images. The 4π steradian view is particularly valuable in autonomous driving or service robotics because people and objects cannot leave the Field of View (FoV) of the camera. If the robot is equipped with a single camera, having a wide FoV offers a significant advantage in robot navigation and people-aware behaviors. However, the lack of panoramic image datasets in addition to the difficulties in handling the geometry of the projection model and the large distortions [1] limited the applicability of these devices. Furthermore, the few available datasets, in most cases, provide annotations at the image level (e.g. bounding boxes) and do not include any information about the spatial positioning of individuals in the robot space. In fact, annotations in such a space are tougher to be manually provided compared to image-level.

To address these concerns, we propose a novel automatic labelling tool, named *PanNote*, to annotate people's real-world 2D positions in a panoramic image, alongside image-level annotations, without the need for human intervention. *PanNote* relies on 2D LiDARs, commonly used in mobile robots, to bridge between image and robot spaces. We aim to enable deep learning methods in panoramic videos for tasks like monocular position estimation and monocular people tracking [2], removing the burden of manually annotating data. Our tool can be used to collect and label custom datasets with low effort, facilitating the development of algorithms to support robot navigation in social environments through commercial panoramic cameras. As shown in Fig. 1, *PanNote* is composed of two main modules: an automatic calibration procedure to find a transformation between a 2D LiDAR and a panoramic camera and a data association module to link information between image and robot space. We also provide a sample dataset and a baseline deep learning method to demonstrate the benefits of using automatically annotated data compared to manually labelled data. The code and dataset are available at `https://github.com/bach05/PanNote.git`.

II. RELATED WORKS

A. Panoramic Video Datasets for Position Estimation

Modern panoramic cameras represent an interesting opportunity in the realm of computer vision and robotics, thanks to the 4π FoV. In fact, previous research has explored the extension of tasks such as object detection [3] or semantic

*Corresponding author, `bacchinalb@dei.unipd.it`
[1]Author is with the Department of Information Engineering, University of Padova, Via Gradenigo 6/b 35131, Italy
[2]Author is with Politecnico di Torino, 10138 Torino, Italy.
Part of this work was supported by MIUR (Italian Minister for Education) under the initiative "PON Ricerca e Innovazione 2014 - 2020", CUP C95F21007870007

segmentation [4], [5] to panoramic images. The wide FoV is particularly valuable for tasks like tracking individuals [6], [7] and robot navigation [8]. In fact, one of the main limitations of traditional cameras is that people are likely to exit the field of view, interrupting the tracking. Despite the great advantage of panoramic cameras, the effectiveness of modern machine learning methods in people detection hinges on the availability of annotated datasets, which is very limited for panoramic videos. Moreover, for robotic navigation, estimating the positions of individuals in the robot's space, rather than just in the image space using bounding boxes, is crucial. However, providing such annotations is considerably more complex, contributing to the scarcity of data for training deep learning models in this scenario. One of the few examples is the CVIP360 dataset [9], which is annotated with bounding boxes around pedestrians and the distances from landmarks placed in the scene at known distances. By exploiting the information about landmarks, authors are able to retrieve the distance between the camera and pedestrians. However, carefully placing landmarks in the environment is not always feasible. *PanNote* does not need any landmarks in the environment, making the labelling process straightforward. It is noteworthy to highlight also the JRDB dataset [10] and its derivative, JRDB-Pose [11]. These datasets encompass manually annotated 2D bounding boxes, 3D oriented cuboids, and skeletal information. The data collection process was undertaken using a sophisticated robotic platform equipped with an omnidirectional camera, multiple RGB cameras, multiple 2D and 3D LiDAR sensors, as well as RGB-D cameras. Despite the massive number of images, around 60K annotated frames, the complexity and the costs behind the labelling procedure make the extension of these datasets to new environments very hard. Instead, the proposed method provides a simple and affordable automatic annotation tool that everyone can use to collect and label panoramic images with a minimum overhead.

B. Calibration of Panoramic Cameras and LiDARs

The fusion of the 2D LiDAR readings and the panoramic images requires the calibration of the two sensors. The calibration between cameras and laser range finders poses many challenges. To tackle these, researchers have introduced a variety of approaches aimed at enhancing calibration accuracy [12]. For example, some studies employ specific calibration patterns to find the geometrical transformations between the LiDAR reference frame and the camera reference frame. Gong *et al.* [13] used a trihedron, while Yu *et al.* [14] designed an L-shaped board. Triangular calibration boards are also employed in [15] and [16] to facilitate extrinsic calibration between cameras and laser range finders. Perreira *et al.* [17] used a sphere as calibration pattern. The center of the sphere is easy to estimate both in the camera's images and in the laser scans. The spherical object, while rolling around, assumes different positions in the field of view sensors. The estimations of the sphere center in multiple positions are then used to calculate the transformation between the sensors. However, most of these techniques exploit standard cameras whose intrinsic parameters can be estimated using well-known calibration algorithms [18]. Nevertheless, panoramic cameras cannot exploit those algorithms because they do not follow the pin-hole camera model. Scaramuzza *et al.* [1] implemented an intrinsic calibration toolbox for omnidirectional cameras, but limited to dioptric and catadioptric cameras. In fact, modern commercial panoramic cameras are based on a different technology. They are polydioptric cameras - i.e. composed of multiple image sensors - equipped with a stitching and interpolation algorithm which outputs equirectangular images. Some authors, like [19], [20], attempted to calibrate panoramic cameras with a 3D LiDAR. To the best of our knowledge, our method is the first attempt to calibrate 2D LiDARs with panoramic cameras. The advantages of 2D LiDARs are that they are notoriously cheaper than 3D LiDARs and can be easily found on mobile robots. Since *PanNote* aims to be a low-cost and easy-to-use tool, we decided to focus on the challenge of using 2D LiDARs.

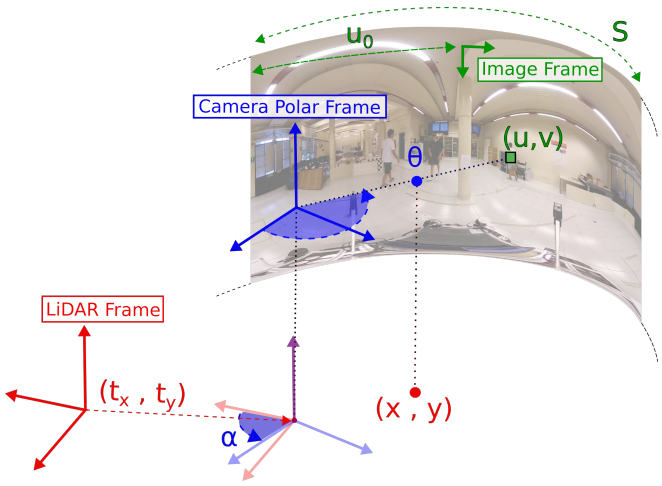

Fig. 2: Graphical representation of the simplified model we propose to project LiDAR points into a panoramic image. The LiDAR frame (red) is translated by (t_x, t_y) and converted in polar coordinate with Eq. 2. θ coordinate is enough to locate an image column. Note that the translation in z is not affecting θ. Rotation α is applied and finally, the point is projected from the camera frame (blue) into the image frame (green).

III. METHODS

A. Calibration of Panoramic Cameras and 2D LiDAR

We introduce a simple yet effective model to calibrate 2D LiDARs and commercial panoramic cameras in order to bridge between the camera and the LiDAR spaces. In the following sections, without loss of generality, we consider the robot space aligned with the 2D LiDAR space.

In standard cameras, the projection model maps 3D points from a world frame into the image plane passing through the optical center O. For panoramic cameras, we can model the projection surface as a unit sphere, centered in O. An arbitrary 3D point $A = (x, y, z)$ can be projected into

(a) (b)

Fig. 3: The image shows the calibration data collection and processing. On the left (a), the detection of the sphere performed by the one-shot detector (red) and its center (magenta). On the right (b), the detection and estimation of the circumference (red) and its center (magenta) in the LiDAR point cloud.

the point A^S on the unit sphere following the connection segment $\overline{AO}$. At this point, previous works [21], [22] tried to project A^S into the image by introducing a virtual cartesian reference frame. We, instead, propose to pass into a polar coordinate system where a sphere is implicitly mapped into a plane. The reason is as follows. In polar coordinates, the mapping between the camera coordinates (θ, ϕ, ρ) and image coordinates (u, v) can be described by Eq. 1, where K is 3×3 matrix modeling the intrinsic parameters, $[R\,|\,T]$ is a 3×4 matrix modeling a roto-translation in polar coordinates.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\,[R\,|\,T]\,\begin{bmatrix} \theta \\ \phi \\ \rho \\ 1 \end{bmatrix} \tag{1}$$

with

$$\begin{bmatrix} \theta \\ \phi \\ \rho \end{bmatrix} = \begin{bmatrix} \text{arctg2}(x, y) \\ \arccos\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) \\ \sqrt{x^2+y^2+z^2} \end{bmatrix} \tag{2}$$

Due to the non-linearity introduced in Eq. 2, finding the optimal set of parameters working directly on Eq. 1 would be hard. To simplify the problem, we can make the reasonable assumption that the 2D LiDAR plane and the θ-plane in the camera polar coordinate system are parallel (see Fig.2). As a consequence a point (x, y) in the LiDAR plane can be mapped into the corresponding vertical line identified by the column coordinate u. As we are going to see in Sec. III-B, estimating u is enough to bridge between robot and image space. We can now formulate the following relation:

$$u = S \cdot \left(\frac{\text{arctg2}(x + t_x, y + y_t) + \alpha}{2\pi} + \frac{1}{2} + u_0\right). \tag{3}$$

Note that under our assumption, the 3D roto-translation described by $[R\,|\,T]$ is reduced into a 2D roto-translation, described by a rotation α and a translation (t_x, t_y). S is the width of the panoramic image and u_0 is the normalized offset with respect to the center of the image frame, as shown in Fig. 2. Given a set of n coupled points $\mathcal{X} = \{u_i, x_i, y_i \mid i = 1...n\}$ where (x_i, y_i) a point in the 2D LiDAR plane and u_i the corresponding image column, we can find the optimal set of parameters $P = (S, \alpha, u_0, t_x, t_y)$ by solving

$$S^*, \alpha^*, u_0^*, t_x^*, t_y^* = \arg\min_P |u_i - u_i^{pred}| \tag{4}$$

where

$$u_i^{pred} = \hat{S} \cdot \left(\frac{\text{arctg2}(x_i + \hat{t}_x, y_i + \hat{t}_y) + \hat{\alpha}}{2\pi} + \frac{1}{2} + \hat{u}_0\right)$$

with $(\hat{S}, \hat{\alpha},\ \hat{u}_0,\ \hat{t}_x,\ \hat{t}_y)$ our initial guess. Under non-linear setting, optimization algorithms need a good initial guess to converge. Our initial guess is based on the following assumption: (1) $\hat{S}$ is reasonably set to the nominal resolution of the camera given by the producer; (2) $(\hat{t}_x, \hat{t}_y) = (0, 0)$ since we expect t_x and t_y significantly smaller compared to the average values of (x_i, y_i); (3) given that $u_0 \in [-0.5, 0.5]$, we perform a grid search in this interval; (4) same as before since $\alpha \in [-\pi, \pi]$.

We now have to face the issue of collecting $\mathcal{X}$. The idea is to use a pattern that can be easily detected by both a LiDAR and a camera. We embrace the approach of Pereira *et al.* [17] which is based on a spherical object, as introduced in Sec. II-B. Inspired by this idea, we propose a more advanced approach for center sphere detection. In the image domain, we designed a one-shot detector based on Faster R-CNN [23] to locate the sphere in the image through a bounding box, since this architecture has proven its efficacy in panoramic images [24]. The model is trained using an image of a sphere, devoid of background, which is randomly superimposed onto images of the environment where the calibration takes place. Since the bounding box circumscribes the sphere, the center of the sphere in the image corresponds to the center of

the bounding box detected. In the LiDAR domain, we first map the 2D point cloud into an image and perform Hough Transform to find candidate subsets of points representing the intersection between the laser scan and the sphere. We filter the candidates using prior information about the sphere radius R. Finally, we select the candidate set of points which better fit the circle equation $(x - x_c)^2 + (y - y_c)^2 = R^2$, where (x_c, y_c) is the center of the circumference drawn by the laser plane intersecting the sphere. Applying some trigonometric relationships, it is easy to calculate its center $(x_c, y_c, \sqrt{R^2 - (x_c^2 + y_c^2)})$. Inferring the center using a subset of points is more reliable than approaches [14] that directly select LiDAR points which are affected by sensibility noise. The result of the processing is shown in Fig. 3.

Overall, the whole calibration procedure is almost automatic. The user is just required to collect a few data and provide camera's nominal resolution and the sphere radius. In our experiments, we used a commercial Swiss ball with $R = 0.65m$. After the optimization process, we achieved a re-projection error of 6.6 pixels, around 0.15% of the image resolution.

B. Data Association and Automatic Labelling

In the previous section, we showed how to calibrate 2D LiDAR and a panoramic camera. After optimization, Eq. 3 allows mapping a point (x, y) from the 2D LiDAR reference frame to the image column u. Data association is then performed by detecting people in LiDAR space, remapping the detection into the image space and intersecting with people detection in the image space.

Despite people's detection in 2D LiDARs constitutes a well-established research field, state-of-the-art approaches [25] have exhibited unsatisfactory accuracy levels in our experimental evaluations. Therefore, we developed a custom detector. Without loss of generality, we assumed that the LiDAR and the camera are static and people are the only dynamic agents in the environment. Under these hypotheses, the fixed structures always produce LiDAR readings around the same point, when not occluded.

Our algorithm begins by processing the measurements from the LiDAR sensor, which are represented in polar coordinates as r and γ. We discretize this space and, for each γ value, employ a voting scheme to identify the distance $r(\gamma)_{max}$ that exhibits the highest likelihood of representing a fixed structure. We classify as fixed structures all the points r, γ such that $r - r(\gamma)_{max} \leq 0.5\ m$ and remove them from the point cloud. Finally, density clustering is applied to the residual point cloud to detect people and their positions. Each cluster is considered a candidate person, its centroid the associated position. In our experiments, we used DBSCAN [26] with $\epsilon = 0.4\ m$ and $min_{samples} = 1$. At this point, a set of possible candidates is identified in the LiDAR space.

To detect people in the panoramic image, we rely on the popular YOLOv7 [27], already used in similar settings [28]. Due to distortions, performing inference directly on the panoramic image leads to many false negatives [6], especially when the person is far. Thus, the following procedure is performed: (1) apply cube projection [29] to alleviate distortions, (2) carry out inference for each side of the cube and (3) perform back-projection to the panoramic image.

Given a set of people's detection in the LiDAR point cloud $\mathcal{L} = \{ (x_i, y_i) \mid i = 1...n \}$ and a set of detection in the image $\mathcal{I} = \{ (u_j^{min}, y_j^{min}, u_j^{max}, y_j^{max}) \mid j = 1...m \}$, we apply Eq. 3 to the elements of $\mathcal{L}$, obtaining $\bar{\mathcal{L}} = \{ u_i \mid i = 1...n \}$. We associate 2D pose (x_i, y_i) to the bounding box $(u_j^{min}, y_j^{min}, u_j^{max}, y_j^{max})$ if $u_j^{min} \leq u_i \leq u_j^{max}$. If more 2D poses are associated with the same bounding box, we keep only the closest to the sensor - i.e. $(x_i, y_i) = \arg\min_{x_k, y_k} \sqrt{x_k^2 + y_k^2}$ - since YOLO usually detect the person in the foreground when two individuals overlap. If a 2D pose (x_i, y_i) intersects more bounding boxes, it is discarded to avoid wrong association and get reliable annotations. Detections with no coupling are considered false positives and discarded. At the end of the process, the image is labelled with people's position in the image space (i.e. the bounding box) and people's position in the robot space (i.e. the LiDAR detection).

IV. EVALUATION

In order to validate our solution, we collect a sample dataset of 4 panoramic videos, both indoor (corridor, laboratory and public hospital) and outdoor (car park). Each video has been recorded with Ricoh Theta Z1 [1] camera at 3840x1920 pixels @ 15 fps. The camera was mounted on a mobile base equipped with 2 Sick-LMS151 LiDARs [2]. One sequence has been recorded while moving the mobile platform.

A. Comparison with Manual Annotations

We manually annotated videos, one frame every 15, using a graphical tool we designed ad-hoc and ending up with 460 labelled frames. We observe an average labelling time of 35 s/image, compared to 2.2 s/image taken by the proposed automatic labelling method, with a speed up of $15\times$. To evaluate the quality of automatic annotation, we compared manual labels with automatic labels. We used two metrics to compare them. Given a set of manual labels $M = \{(\dot{x}_i, \dot{y}_i, \dot{box}_i) \,|\, i = 1..n\}$ and a set of labels automatically generated through *PanNote* $\mathcal{A} = \{(x_j, y_j, box_j) \,|\, j = 1..n\}$, we computed the RMSE as

$$RMSE = \sqrt{\frac{1}{n} \sum_{i,j=1}^{n} (\dot{x}_i - x_j)^2 + (\dot{y}_i - y_j)^2}$$

and the accuracy $\mathrm{Acc}[\mu]$ as

$$\mathrm{Acc}[\mu] = \frac{|\{(x_j, y_j)\,,\ \sqrt{(\dot{x}_i - x_j)^2 + (\dot{y}_i - y_j)^2} \leq \mu, i = j\}|}{n}$$

where μ is the desired tolerance threshold in meters. Results are shown in Table I. The average difference between

[1] https://www.ricoh360.com/theta/
[2] https://www.sick.com

Metric	indoor1	indoor2	indoor3	outdoor	**Avg**
RMSE [m]	0.086	0.023	0.048	0.021	0.044
Acc[0.75]	0.976	1.0	0.985	0.992	0.988
Acc[0.25]	0.957	1.0	0.975	0.985	0.979
Acc[0.01]	0.845	0.871	0.838	0.953	0.876

TABLE I: Comparison of *PanNote* with manual labels.

manual and automatic labels is $0.04\ m$. Since distinguishing a difference of $0.04\ m$ is difficult even for a human, we can assume that this error is smaller than the inherent uncertainty associated with manually assigned labels. In other words, the RMSE tells us that automatic labels are indistinguishable from manual labels. However, RMSE does not describe how predictions are distributed around the ground truth. On the other hand, accuracy evaluates the probability of a prediction being close enough to the ground truth, providing more valuable insights about the trustworthiness of the predictions. Taking into account that the average human step size is around $0.75\ m$, the $\mathrm{Acc}[0.75]$ indicates that 98.8% of times automatic labels can be considered correct since they fall within $0.75\ \ m$ from the manual annotation. According to [30], in the most popular dataset for computer vision the average error rate in the annotations is 3.3%. *PanNote* successfully shows an error rate of 1.2% according to $\mathrm{Acc}[0.75]$ metric, below the average error rate. For completeness, we also evaluated $\mathrm{Acc}[0.25]$ and $\mathrm{Acc}[0.01]$. The latter ensures that, in 87.67% of cases, the labels from *PanNote* are at most $0.01\ m$ far from the manual annotations.

B. Baseline Model

In Sec. IV-A, we validate the quality of the labels generated by *PanNote*. In this section, we aim to prove that the dataset generated through *PanNote* is suitable for training machine learning models. Therefore, we designed a simple baseline model to be trained on automatically labelled data. The model consists of a backbone in charge of extracting key-points from people in the panoramic image, followed by a Multi-Layer Perceptron (MLP) [31] which performs regression on key-points to predict the 2D position of the person in the robot space. In our case, the key-points are the corners of the bounding box extracted by YOLOv7 [27]. We used a MLP with $[8, 16, 32, 64, 128]$ hidden units, trained for 200 epochs with *AdamW* [32] optimizer, learning rate = 0.5, weight decay = 0.01 and early stopping after 5 epochs without improvements in the validation loss. We performed training and testing on different datasets: *Manual* contains 460 manually annotated frames; *PanNote-Red* contains the same frames of *Manual*, but labelled with *PanNote*; *PanNote-Full* contains 6828 frames automatically annotated with *PanNote*. Since frames come from videos acquired in different conditions, we performed k-fold cross-validation by training on frames from $k-1$ videos and testing on the remaining. Additionally, we introduced a small set for testing purposes only of 85 manually annotated frames, acquired while moving the mobile platform. In Tab. II the average metric values across the k folds are shown.

Train set	RMSE [m]	Acc[0.75]	Acc[0.25]	Acc[0.01]
Manual	0.785	0.688	0.201	0.035
PanNote-Red	**0.779**	0.670	0.212	0.037
PanNote-Full	0.784	**0.72**	**0.287**	**0.072**

(a) Test on manually annotated frames.

Train set	RMSE [m]	Acc[0.75]	Acc[0.25]	Acc[0.01]
Manual	0.850	0.687	0.175	0.032
PanNote-Red	0.829	0.666	0.230	0.042
PanNote-Full	**0.646**	**0.768**	**0.267**	**0.052**

(b) Test on *PanNote* annotated frames.

Train set	RMSE [m]	Acc[0.75]	Acc[0.25]	Acc[0.01]
Manual	0.658	0.751	0.267	0.071
PanNote-Red	0.679	0.756	0.253	0.054
PanNote-Full	**0.436**	**0.897**	**0.410**	**0.103**

(c) Test on manually annotated frames with moving platform.

TABLE II: Comparison of *PanNote* with manual annotations after k-fold validation. Bold highlights the best results.

Results clearly show the advantage of training with a larger number of examples that can be easily provided by the proposed method. A larger number of examples prevent overfitting. In fact, training on *Manual* and *PanNote-Red* stops on average around $22k$ iteration, while on *PanNote-Full* it keeps going until $125k$ iterations on average. Thus, leveraging automatic annotations proves beneficial, as the abundance of examples generated leads to improved generalization and higher performance. The comparison of performance at test time confirms this hypothesis. The small bias between the metrics in the two cases is probably due to some discrepancies introduced by the automatic labelling tool. In any case, all the trends are stable in both tests, proving that images annotated with *PanNote* offer a reasonable estimation of the real performance of the model.

In Sec.III-B, we claimed that labelling on static images is not a limitation. In fact, Tab. IIc reports the test results on the moving platform, demonstrating that the model can generalize to dynamic situations. Again, data generated thorough *PanNote* exhibits the best performances.

While our baseline model still exhibits some limitations in overall performance, the results are satisfying. Some errors are due to the bounding box that does not cover the whole figure of the person, leading to imprecise position estimation as visible in Fig.4. But it is also worth noticing that in other cases (e.g. Fig.4, bottom row, yellow person), the baseline model is able to predict a meaningful position even where the automatic labelling tool fails. However, beyond the performance of the model, we showed that the use of *PanNote* annotation tool facilitated the training of deep neural networks due to the large volume of data that can be generated without the need for human intervention.

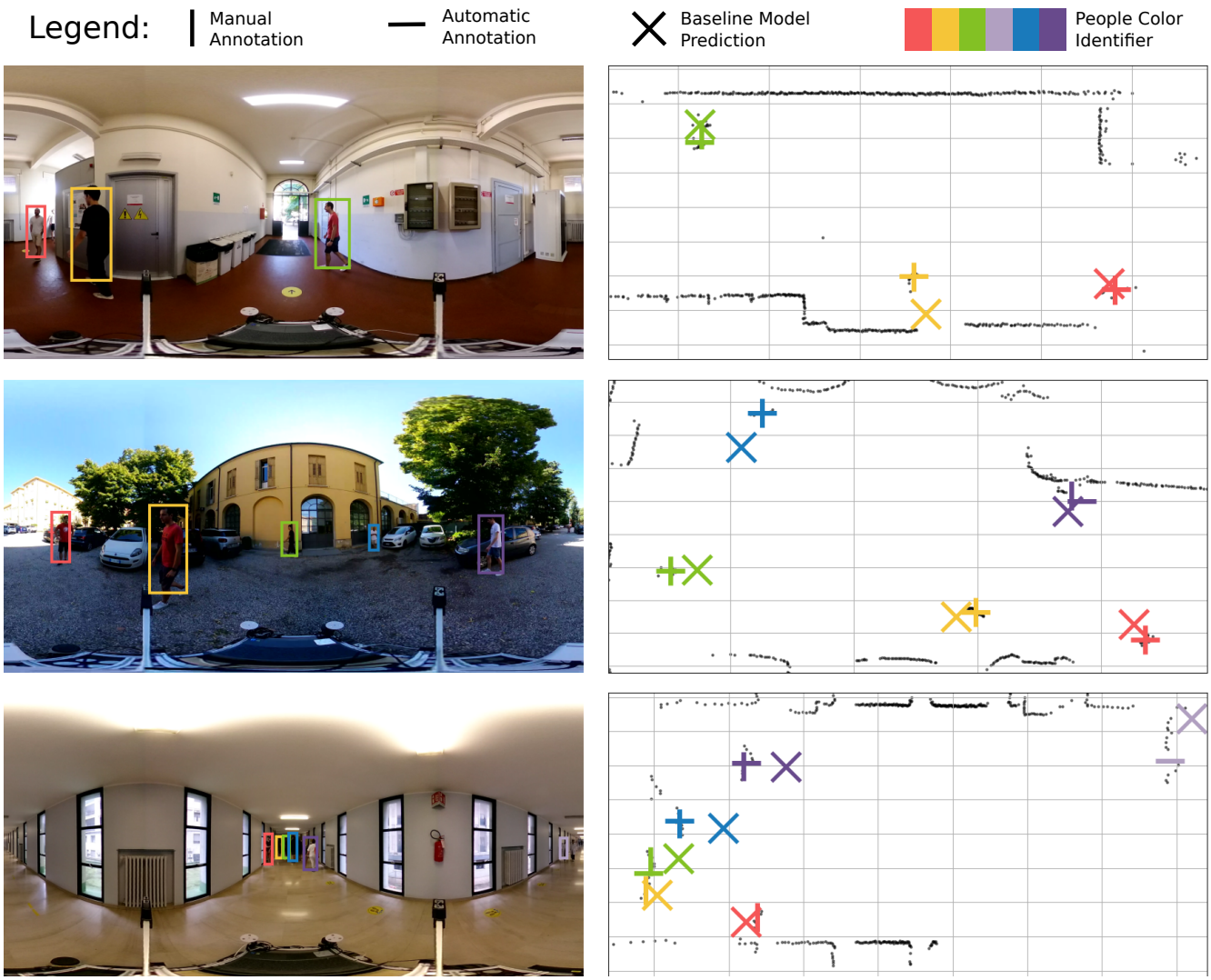

Fig. 4: The image compares manual labels (|), *PanNote*-generated labels (–) and prediction with our baseline model (×) in different scenarios. Note in the upper image, the yellow person's position is mispredicted by our baseline model due to the wrong-sized bounding box. It is also worth highlighting how the baseline model is able to retrieve a good estimation of the person position when the automatic labeller fails (bottom image, yellow and red persons).

V. CONCLUSIONS

In this work, we introduce *PanNote*, an automatic labelling tool to annotate people's 2D position in panoramic images. The contribution of this work is twofold. An effective model for panoramic cameras and 2D LiDAR calibration and a data association algorithm to include in the image annotations also the people's positions in the robot space. While the automatic labeling is performed within specific constraints, such as a static platform, our experimental evaluation on real-world videos has confirmed the effectiveness and utility of the proposed method, even when those constraints are relaxed. Notably, our method operates without the need for human intervention, simplifying the acquisition of training data for deep learning models. With this work, we aim to make available to researchers a tool to boost research on monocular people position estimation and vision-based robot navigation using modern and cheap panoramic cameras. Although 2D LiDAR-based navigation has demonstrated successful utility across numerous scenarios, challenges arise in the context of human-aware navigation, particularly with respect to potential occlusions of individuals' lower extremities. In such instances, the incorporation of panoramic vision alongside LiDAR sensors can enhance navigation system performance, ensuring greater robustness and reliability. For future works, we aim to extend the annotation tool functionality, introducing the support to 2D and 3D skeleton annotations, which may solve the issues with wrong-sized bounding boxes. We also want to refine the calibration method in order to further reduce errors. *PanNote* proved effective in this setting, but we plan to enhance performance by leveraging a tracking algorithm to exploit the spatiotemporal correlations in the observations of the pedestrians.

REFERENCES

[1] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5695–5701.

[2] K. Koide, J. Miura, and E. Menegatti, "Monocular person tracking and identification with on-line deep feature selection for person following robots," *Robotics and Autonomous Systems*, vol. 124, p. 103348, 2020.

[3] S.-H. Chou, C. Sun, W.-Y. Chang, W.-T. Hsu, M. Sun, and J. Fu, "360-indoor: Towards learning real-world objects in 360deg indoor equirectangular images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[4] K. Yang, J. Zhang, S. Reiss, X. Hu, and R. Stiefelhagen, "Capturing omni-range context for omnidirectional segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 1376–1386.

[5] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang, "Pass: Panoramic annular semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 4171–4185, 2020.

[6] A. Bacchin, F. Berno, E. Menegatti, and A. Pretto, "People tracking in panoramic video for guiding robots," in *International Conference on Intelligent Autonomous Systems*. Springer, 2022, pp. 407–424.

[7] J. Long, J. Mei, and G. Ma, "Egocentric two-frame pedestrian trajectory prediction algorithm based on a panoramic camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[8] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *Proceedings of the 2020 Conference on Robot Learning*, vol. 155. PMLR, 2021, pp. 671–681. [Online]. Available: https://proceedings.mlr.press/v155/anderson21a.html

[9] G. Mazzola, L. Lo Presti, E. Ardizzone, and M. La Cascia, "A dataset of annotated omnidirectional videos for distancing applications," *Journal of Imaging*, vol. 7, no. 8, p. 158, 2021.

[10] R. Martín-Martín, M. Patel, H. Rezatofighi, A. Shenoi, J. Gwak, E. Frankel, A. Sadeghian, and S. Savarese, "Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6748–6765, 2023.

[11] E. Vendrow, D. T. Le, J. Cai, and H. Rezatofighi, "Jrdb-pose: A large-scale dataset for multi-person pose estimation and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4811–4820.

[12] K. Liao, L. Nie, S. Huang, C. Lin, J. Zhang, Y. Zhao, M. Gabbouj, and D. Tao, "Deep learning for camera calibration and beyond: A survey," 2023.

[13] X. Gong, Y. Lin, and J. Liu, "3d lidar-camera extrinsic calibration using an arbitrary trihedron," *Sensors*, vol. 13, no. 2, pp. 1902–1918, 2013.

[14] L. Yu, M. Peng, Z. You, Z. Guo, P. Tan, and K. Zhou, "Separated calibration of a camera and a laser rangefinder for robotic heterogeneous sensors," *International Journal of Advanced Robotic Systems*, vol. 10, no. 10, p. 367, 2013.

[15] S. Debattisti, L. Mazzei, and M. Panciroli, "Automated extrinsic laser and camera inter-calibration using triangular targets," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 696–701.

[16] X. Xu, L. Zhang, J. Yang, C. Liu, Y. Xiong, M. Luo, Z. Tan, and B. Liu, "Lidar–camera calibration method based on ranging statistical characteristics and improved ransac algorithm," *Robotics and Autonomous Systems*, vol. 141, p. 103776, 2021.

[17] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple lidars and cameras on autonomous vehicles," *Robotics and Autonomous Systems*, vol. 83, pp. 326–337, 2016.

[18] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[19] A.-I. García-Moreno, J.-J. Gonzalez-Barbosa, F.-J. Ornelas-Rodriguez, J. B. Hurtado-Ramos, and M.-N. Primo-Fuentes, "Lidar and panoramic camera extrinsic calibration approach using a pattern plane," in *Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 104–113.

[20] Z. Miao, B. He, W. Xie, W. Zhao, X. Huang, J. Bai, and X. Hong, "Coarse-to-fine hybrid 3d mapping system with co-calibrated omnidirectional camera and non-repetitive lidar," *IEEE Robotics and Automation Letters*, vol. 8, pp. 1778–1785, 2023.

[21] C. Geyer and K. Daniilidis, "Paracatadioptric camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 687–695, 2002.

[22] X. Gong, Y. Lv, X. Xu, Y. Wang, and M. Li, "Pose estimation of omnidirectional camera with improved epnp algorithm," *Sensors*, vol. 21, no. 12, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/4008

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, vol. 39, no. 06, pp. 1137–1149, 2017.

[24] F. Deng, X. Zhu, and J. Ren, "Object detection on panoramic images based on deep learning," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, 2017, pp. 375–380.

[25] D. Jia, A. Hermans, and B. Leibe, "DR-SPAAM: A Spatial-Attention and Auto-regressive Model for Person Detection in 2D Range Data," in *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[26] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996, p. 226–231.

[27] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[28] T. Wang, B. Chen, N. Wang, Y. Ji, H. Li, and M. Zhang, "Zero-shot obstacle detection using panoramic vision in farmland," *Journal of Field Robotics*.

[29] N. Greene, "Environment mapping and other applications of world projections," *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 21–29, 1986.

[30] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[31] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5, pp. 183–197, 1991.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.