Smart Bus Stops: Public Transport monitoring

Hugo Martínez[1*], Manel Davins[2], Irene Cardenas[3], Margarida López[4], Agusti López[5], Francisco Barelles[6] and Josep Ramon Morros[7]

1. CARNET-UPC (hugo.martinez@upc.edu), Spain
2. CARNET-UPC (manel.davins@upc.edu), Spain
3. LaSalle (irenecpalma@gmail.com), Spain
4. Transports Metropolitans de Barcelona (TMB) (mlromero@tmb.cat), Spain
5. Transports Metropolitans de Barcelona (TMB) (almarin@tmb.cat), Spain
6. Transports Metropolitans de Barcelona (TMB) (fbarelles@tmb.cat), Spain
7. Universitat Politècnica de Catalunya (UPC) (josep.ramon.morros@upc.edu), Spain

Abstract

The digitalization of public transport offers new opportunities to improve service efficiency, particularly in managing peak hours and reducing delays. A key contributor to transport inefficiency is crowd congestion at bus stops. Monitoring bus stops enables a better understanding of passenger flow and waiting times, supporting more effective planning and faster real-time response. However, current data on activity at these locations is scarce due to privacy concerns and the practical difficulties of collecting data in urban settings. To address this gap, this paper proposes a solution to monitor bus stops by equipping them with depth cameras, which capture anonymized data by only recording distance and leveraging computer vision techniques to perform people and bus detection. Results show the viability of the proposed solution to optimize bus networks and its potential replicability in other urban environments.

Keywords: *Public Transport, Bus, AI, Computer Vision, Object Detection.*

1. *Introduction*

With the target of exploring the application of cutting-edge technologies for bus network improvement the Innovation Department of the Transports Metropolitans de Barcelona (TMB), the Universitat Politècnica de Catalunya (UPC), and CARNET-UPC have collaborated on a project focused on optimizing bus stop operations through the use of artificial intelligence and sensor technology. Obtaining detailed information of the activity at bus stops can help planning bus frequency, modifying routes and giving real-time response to ongoing demand. This means allocating resources more effectively and improve overall bus service efficiency.

The primary objective of this project is to explore techniques for monitoring activity at bus stops using as objective metrics the number of people present, individual waiting times and bus arrivals. To achieve this, a sensor must be installed at bus stops to enable data acquisition. The most straightforward approach is to use traditional cameras and apply computer vision techniques to extract relevant information from RGB (Red, Green, and Blue) images, as demonstrated in previous works in urban environments (Viegas, 2019; Velastin et al., 2020; Bedmar, 2024). Numerous state-of-the-art methods have shown outstanding performance in object detection from RGB images (Zong, Song, Liu, 2023; Su et al., 2023), particularly for detecting people (Wang et al., 2023; Zheng et al., 2022) and even with multimodal data (Zhang et al., 2024). However, deploying traditional RGB cameras in public spaces raises significant concerns related to data privacy. To overcome this limitation, in this project we propose the use of depth cameras for people and bus detection. These sensors capture the distance of each element in the scene to the camera plane on a per-pixel basis, inherently providing anonymized data and eliminating the possibility of extracting sensitive visual information. To the best of our knowledge, no prior research has addressed bus stop monitoring through depth data, making this a novel and privacy-conscious approach in urban mobility.

Within this framework, we propose a pipeline that obtains competitive performance based on the current available detection models and tracking methods, analyzing their strengths and limitations within the context of the Smart Bus Stop project. Particular focus has been given to data acquisition, choosing an adequate depth camera from the possible depth acquiring technologies, finding a satisfactory camera setup and equipping various bus stops with a camera to finally obtain a high-quality dataset.

This paper is structured as follows. Firstly, the objective of the project and the aim of the research is stated. Subsequently, a comprehensive literature review on depth technology, object detection models and object tracking methods is provided. Then, the proposed pipeline and the methodology to validate it are detailed. Additionally, results for detection and tracking are presented and analyzed. Lastly, conclusions are highlighted including future research possibilities.

2. *Background*

2.1 Depth Perception

Depth perception is crucial to an array of applications like robotics, autonomous vehicles, scene understanding, 3D representation, path planning, and augmented reality, among others. For this reason, it has been a popular research topic in the field of computer vision. So far, precise depth sensing is achieved via multi-view (stereo) imaging or through depth sensors such as Time of Flight (e.g. LiDAR - Light Detection and Ranging) and Structured Light.

Depth Cameras

Depth cameras are devices that are able to determine the distance between themselves and a target or between two objects. In this project we use the Intel Realsense D455 stereo depth camera (Hübner, Hou, Iwaszczuk, 2023) which is based on stereo depth technology and has a depth range of 6 meters. Stereo depth cameras typically use two sensors spaced apart to estimate depth by comparing images from each sensor, in a similar way to human vision, see Figure 1. The wider the sensor separation, the greater the depth range. This type of camera works in various lighting conditions, including outdoors, and does not interfere with other similar cameras unlike laser emitting cameras this is one of the key reasons the D455 camera was selected.

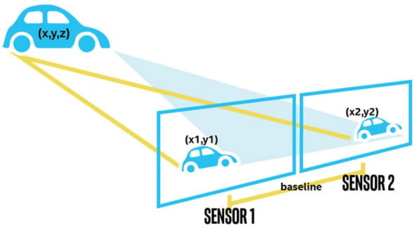

Figure 1: *System overview*

An alternative depth sensing approach is Time-of-Flight (ToF), which measures the time it takes for emitted infrared light to bounce back from surfaces to estimate depth. While ToF cameras can offer high frame rates and perform well indoors, their performance degrades significantly under direct sunlight due to interference from ambient infrared radiation. For this reason, we opted not to use ToF technology, as our data collection took place in outdoor settings where consistent depth accuracy is required.

2.2 Object Detection

Object detection is a computer vision task the goal of which is to detect and locate objects of interest in an image or video. This means identifying the position and boundaries of objects in an image (typically expressed as bounding boxes), and classifying objects into different categories.

Object detection typically involves two tasks:

1. **Object Classification:** Identifying what types of objects are present in an image (e.g., cars, people, animals).
2. **Object Localization:** Determining where these objects are located within the image by drawing bounding boxes around them.

Object detection can be generally split into 2 categories: One-stage and Two-stage detectors. The main difference between these is that two-stage detectors generate region proposals and one-stage detectors do not.

One-stage detectors are typically faster and more suitable for real-time applications, but may sacrifice some accuracy compared to two-stage methods. Also, they are not as good recognizing irregular or small objects. The most popular are: YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) and RetinaNet (Lin et al., 2017). Two-Stage Detectors usually have higher accuracy. However, they tend to be slower due to many inference steps per image and are not end-to-end trainable because they have intermediate cropping stages. Some examples are: R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015) and SPP-Net (He et al., 2015). In addition to these traditional approaches, transformer-based models have recently emerged as a new direction in object detection. DETR (Carion et al., 2020), for example, reframes object detection as a direct set prediction problem, using transformer architectures to match image features with a fixed set of learnable object queries. These models simplify the detection pipeline and enable global reasoning across the image.

In this project, we use the latest version of the one-stage detector YOLO as a pre-trained model for detection tasks, due to its state-of-the-art real-time performance, ease of use, deployment facilities, and abundant available resources.

You Only Look Once (YOLO) is a real-time object detection model that predicts bounding boxes and class probabilities directly from an image in a single pass of a convolutional neural network (CNN). YOLO processes an input image by passing it through a CNN and dividing into sections to obtain a grid, for each cell multiple bounding boxes are predicted, with confidence scores and class probabilities. To refine predictions, non-max suppression removes overlapping detections using Intersection over Union (IoU) as metric. The final output consists of detected objects with their class labels, confidence scores, and precise bounding box coordinates. This straightforward approach makes it very fast while maintaining accuracy, capable of detecting objects in real-time. YOLO has been developed in several versions and each version has been built on top of the previous one with enhanced features obtaining improved accuracy, faster processing, and better handling of small objects and occlusions.

2.3 Object Tracking

Multi Object Tracking (MOT) is the task of following several objects along the frames of a video sequence. The position of each object must be determined in each frame, and then given a unique identity across the sequence. Many of the SotA MOT methods follow the tracking-by-detection paradigm: first an object detection method is applied to successive frames to locate the objects, and then, an association step determines the detections that belong to the same object in consecutive frames.

The **Simple Online and Realtime Tracking (SORT)** (Bewley et al., 2016) algorithm is one of the most computationally efficient MOT algorithms, relying on a Kalman filter (Kalman, 1960) for motion prediction and the Hungarian algorithm (Kuhn, 1955) for detections association. SORT is designed for high-speed tracking applications and does not incorporate appearance features, which makes it lightweight but also less robust to occlusions and identity switches. Due to its reliance only on motion predictions, SORT struggles when objects are momentarily lost (not detected) due to occlusions, abrupt movements, or similar-looking objects close together.

Observation-Centric SORT (OC-SORT) (Cao et al., 2023), enhances SORT by improving robustness to occlusions and adding non-linear motion. OC-SORT introduces new key mechanisms:

1. Observation-Centric Re-Update (ORU): This method mitigates error accumulation in the Kalman filter when objects are lost by re-updating tracks using past observations.
2. Observation-Centric Momentum (OCM): Instead of relying only on noisy velocity estimates from the Kalman filter, OCM integrates direction consistency into the association step.
3. OCR is a technique that associates the last known observation of each unmatched track to all of the unmatched observations. This is an IOU association that occurs after the primary track/detection association.

ORU and OCM are effective solutions for tracking under occlusion or non-linear motion. OCR helps to prevent lost tracks. In general, OC-SORT has better performance in occlusion-heavy scenarios, making it useful for environments with high density of pedestrians, such as bus stops.

To address the limitations of SORT, trackers such as **DeepSORT** (Wojke, N., Bewley, A., & Paulus, D., 2017), **StrongSORT** (Du, Y, 2023) or **Deep OC-SORT** (Maggiolino, G., 2023) add deep-learning-based appearance information in addition to motion information. By using appearance features, these methods improve detection association, reducing identity switches in crowded environments. However, using deep appearance features introduces additional computational overhead, which can be a drawback in real-time systems.

3. *Method*

The objective of the project is to enable the extraction of key metrics in bus stops: waiting time, people density and bus arrivals. For this purpose, we equip bus-stops with sensors, concretely with depth cameras which guarantee data privacy. In order to extract the desired metrics from depth data we need to develop computer vision methods to detect people and keep track of each person. Having this in mind, the focus is to develop a people detection and tracking system, as depicted in Figure 2. The first stage consists of a detector based on the previously introduced YOLO model. However, the available pre-trained YOLO models operate on RGB images and require additional training to work on depth data. The second stage employs the OC-SORT tracker, which uses detection bounding boxes as input to maintain object identities across frames. Additionally, we researched complementary approaches using traditional computer vision techniques.

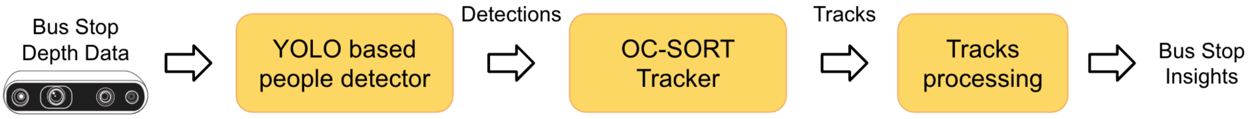

Figure 2: *System overview*

3.1 People detection

YOLO-based detector

The main approach is to use a pre-trained object detection model and use it to perform people detection in depth images. However, since the available pre-trained YOLO models are designed to detect objects in RGB images, they must be adapted to perform detection in depth data. To achieve this, the first step is to collect paired RGB and depth images using a depth camera. The pre-trained YOLO model is then applied to the RGB images to only detect people and generate people detection labels (bounding boxes coordinates). These labels, along with the corresponding depth images, are then used to fine-tune (retrain) the pre-trained YOLO model to enable detection of people in depth data, see pipeline in Figure 3. A relevant observation is that, unlike in RGB images, objects in depth data are not invariant to translation. The depth representation of an object varies depending on its position relative to the camera, in general following a predictable geometric pattern. While this provides valuable spatial cues the model can learn from, it also means that a larger and more diverse dataset is needed to ensure the model generalizes well across different distances and scene configurations.

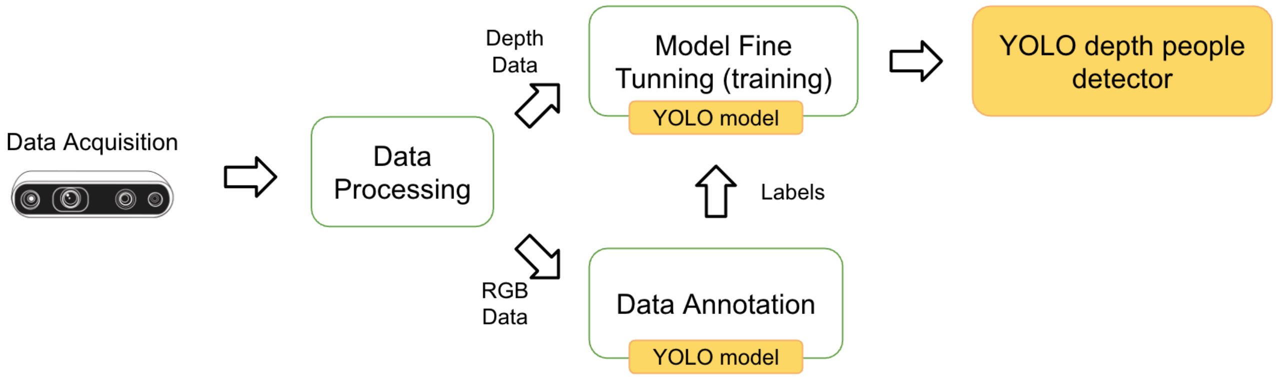

Figure 3: *YOLO fine tune pipeline*

Data Acquisition

The camera used for data acquisition is the Intel RealSense D455, which is based on stereo vision technology. This camera can record a depth stream and also a RGB stream. Both streams can be registered simultaneously and aligned. Recordings were conducted in the city of Barcelona at three different bus stops with varying numbers of people (standing and seated) and including scenarios where buses arrived at the stop, with people boarding and unboarding the buses. Each stop has a height of 2.9 m, a width of 1.7 m and a length of 4 m.

The camera was mounted on top of the bus stop at a height of 2.25 approximately (see Figure 4), providing a wide view of the bus stop. The camera was installed in a vertical orientation instead of the standard horizontal one, being rotated by 90 degrees. This was done because the camera has a wider angle of view in the horizontal direction than in the vertical; by rotating it, we effectively maximize the field of view in the vertical direction, given that road information is not as relevant in this particular scenario. Since it was mounted at a certain height, it was also tilted 45 degrees downward to better capture the bus stop.

We annotated 9 different recordings for a total of 763 seconds. We used seven recordings (14,537 frames) for the training set and the remaining two (3,604 frames) for the validation set. The recordings in the validation set are from bus stops not used in the training set. For all frames, we annotated the persons’ locations using bounding boxes (70,000 in total), and their identities using a unique label for each person (25 in total). Currently, the dataset is being expanded.

Manual annotation of video sequences is very time-consuming. Moreover, annotating directly on the depth images can be challenging due to their lack of detail. To address this, we adopted a semi-supervised approach: first, we performed automatic person detection in the RGB stream using a YOLOv12x model. Since the RGB and depth images are registered, the RGB bounding boxes mark exactly the positions of the persons in the corresponding depth images.

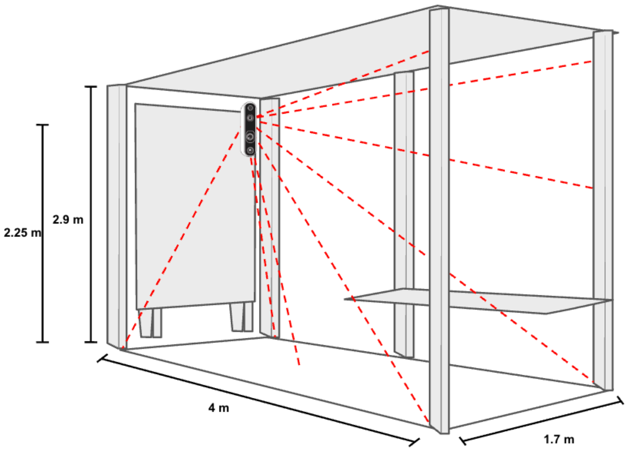

***Figure 4:** Bus stop depth camera setup*

The automatic annotation can contain errors (missed persons, false positives, bounding boxes with incorrect adjustment to the persons). If the depth detection model was trained with data containing occasional errors, the model would most probably replicate this behavior when being applied to a real scenario. The idea is to add a manual correction step after the automatic annotation, see Figure 5, using the CVAT annotation software. This semisupervised approach is much less time consuming than full manual annotation. As the predictions of the YOLO model are already very accurate, only occasional changes are needed. This correction step significantly improved detection results by reducing false positives.

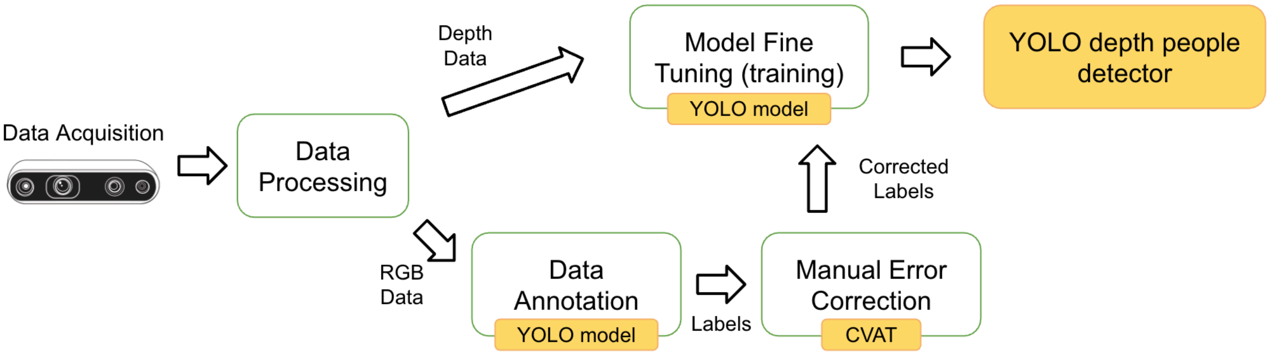

***Figure 5:** YOLO fine tune pipeline with corrected annotations step*

Point-cloud Based Detection

To complement the first detection approach that uses a pre-trained YOLO model which works using 2D data, we developed a second approach that works directly with depth data represented as point-clouds (collections of 3D points in space).

In this case, the detection pipeline processes the depth information in five stages (see Figure 6):

1. **Initial processing**: The depth data from the camera is converted into a 3D point cloud, giving us the precise location of every surface visible to the camera. Also, we filter out points that are too far away or outside the area of interest. This is achieved using the camera's intrinsic parameters and applying cropping margins to focus on the relevant area of the bus stop.
2. **Floor removal**: Identify and remove points that belong to the floor of the bus stop using RANSAC (Fischler et al, 1981). This works by finding the best-fitting plane that represents the floor and removing all points close to it. This helps simplify the following steps by focusing only on objects that rise above the ground plane.
3. **Clustering**: The remaining points are clustered based on how close they are to each other using DBSCAN (Ester, 1996). Points that are close to each other (within 7mm) are grouped together if there

are enough nearby points (at least 10). For bigger clusters that might contain multiple objects, a second, more detailed clustering is performed to separate them. Each cluster potentially represents a person or other objects in the scene.

4. **People detection**: Analyze each cluster's shape and size to determine if it matches the characteristics of a person. To do this, we apply Principal Component Analysis (PCA) to extract the directions of maximum variance within each cluster. The first principal component usually corresponds to the person's height (or the distance from the ground to the head in the case of a seated person) which tends to fall within a certain range and is used as a filter.
5. **Structure filtering**: Finally, remove any detections that are likely to be bus stop structures (like poles) rather than people. This is done by analyzing the shape patterns that are characteristic of these structures. For this each cluster is sliced horizontally (in 100mm segments) and the ratio between the length and width of each slice is analyzed. If the ratio is too high (>1.9), indicating a long, thin structure, we identify it as a pole of the bus stop rather than a person.

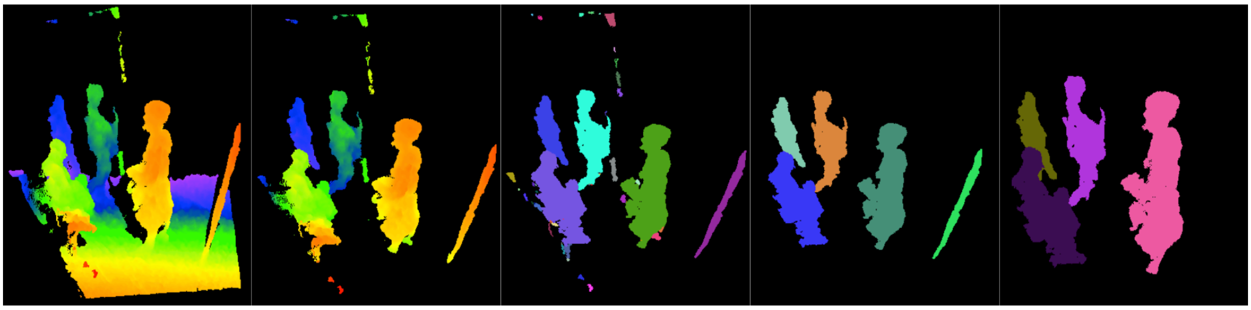

Figure 6: *Point-cloud detection steps from left to right*

After these steps, the 3D bounding boxes were extracted using the limits of clusters obtained using the directions of maximum variance. These 3D bounding boxes are then projected into the camera viewpoint to extract 2D bounding boxes that correspond to people in the stop.

3.2 People Tracking

For tracking the OC-SORT tracker is used because it relies on position and motion of detections rather than appearance features like DeepSORT, making it ideal for depth-based detection where RGB information is not available. As commented before, OC-SORT matches detections across frames using IoU and motion prediction with a Kalman filter. This motion prediction improves tracking when an object is lost (no detections) for a few frames.

The OC-SORT tracker has several tunable parameters that impact tracking performance:

- **Detection threshold** filters out low-confidence detections, which helps reduce false positives but setting this threshold too high can result in the loss of valid detections.
- **IoU threshold** controls how strictly new detections are matched to existing tracks based on spatial overlap. A high threshold ensures that only well-aligned detections are associated, minimizing identity switches but may prevent valid matches when the objects move.
- **Maximum age** determines how long a lost track is retained before deletion. This helps handling brief occlusions or missed detections. However, it can keep outdated or false tracks active for too long.
- **Minimum hits** set how many consecutive detections are needed before initializing a track, reducing false positives. It can cause valid objects to be ignored if detections are inconsistent, having false negatives in between.

- **Delta time** defines motion history window, how far back detections are considered to make new predictions. A longer delta time leads to more stable motion predictions but it may also reduce responsiveness to sudden changes in an object's speed or direction.
- **Inertia** controls the impact of motion history. High inertia leads to smoother motion trajectories and reduces jitter but it can also make the tracker less reactive to sudden motion changes.

4. Results

4.1 People Detection

There are several metrics commonly used to evaluate the performance of object detection models:

1. **Intersection over Union (IoU)**: Measures the overlap between the predicted bounding box Bp and the ground truth bounding box Bg. Higher values indicate better localization accuracy.

   $$IoU = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \quad \textbf{(1)}$$

2. **Precision**: The proportion of correctly predicted positive detections out of all positive predictions. High precision means fewer false positives.

   $$Precision = \frac{TP}{TP+FP} \quad \textbf{(2)}$$

3. **Recall**: The proportion of correctly predicted positive detections out of all actual positives. High recall means fewer missed detections.

   $$Precision = \frac{TP}{TP+FP} \quad \textbf{(3)}$$

4. **F1 Score**: The harmonic mean of precision and recall, providing a balanced measure. A high F1 score indicates a good trade-off between precision and recall.

   $$F1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \textbf{(4)}$$

5. **Mean Average Precision (mAP)**: The mean of the Average Precision across all classes. The Average Precision is computed as the area under the precision-recall curve. A higher mAP indicates better overall detection performance.

   $$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad \textbf{(5)}$$

YOLO-based People Detection

The proposed YOLO-based detection generally works well and provides stable detections (see Figure 7) that the OC-SORT tracker can follow and consistently identify across frames. However, this method is not particularly robust to occlusions and may occasionally miss to detect people on motorbikes or pedestrians walking by. To address this, a preprocessing step is needed to filter out depth values using a distance threshold, as well as a postprocessing step to remove detections that appear on the road or outside the bus stop. These additional steps, however, reduce the generalizability of the solution, as their parameters need to be adjusted for each specific bus stop. In Table 1 we observe the mAP, precision and recall of this approach using the pretrained YOLOv12m version. In general, although some misdetections occur, the current detector performance would allow to extract people density in the stop accurately.

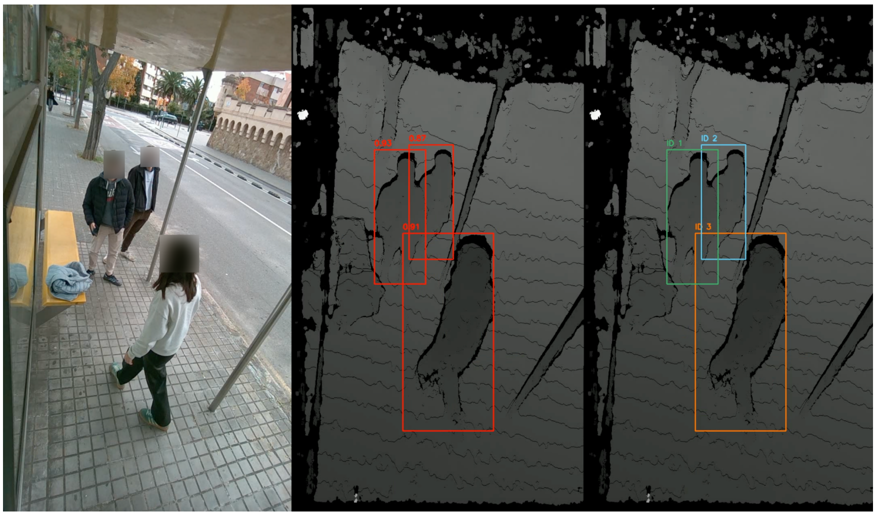

Figure 7: *YOLO approach detection and tracking results. From left to right: RGB, detections and tracks.*

Table 1: *Performance of YOLO-based person detection*

mAP	Precision	Recall	F1-Score
0.9882	0.9689	0.9547	0.9617

Point-cloud People Detection

Conducting a qualitative analysis, see Figure 8, we observed that for the same bus stop recording, this method is more robust against occlusions compared to the YOLO-based approach as we are relying in 3D information instead of just 2D images. Moreover, it is rotation-invariant, as PCA is applied to each cluster to extract sizes and distinguish people.

With this method, people are identified based on the shape and proportions of people, which leads to false positives, particularly when vehicles approach the stop. This happens because due to limitations of the depth sensor, vehicles are not perceived as smooth surfaces but rather as highly noisy ones, leading to the appearance of irregular shapes that may be detected as people. However, knowing the specific areas where vehicles pass and where buses stop allows for manual filtering of these false detections. This also enables the recognition of vehicles, making it possible to recognize when a bus is stopping if a vehicle remains in the stop area more than a certain number of frames when a bus stops the “bus stop” tag is displayed as seen in Figure 8.

However, detections tend to be noisier since bounding boxes are inferred based on object size and fluctuate between frames due to camera noise, which negatively impacts the tracking performance. As a result, when using these detections together with the OC-SORT tracker it requires careful parameter tuning to achieve reasonable tracking performance although it remains prone to errors when occlusions happen. Also, one of the key advantages of the YOLO approach is its fast inference, whereas the point-cloud method takes a much bigger time to process every frame, not achieving real-time performance.

The main steps of this approach, such as initial processing, floor removal, and clustering, are generalizable and can be useful in future solutions.

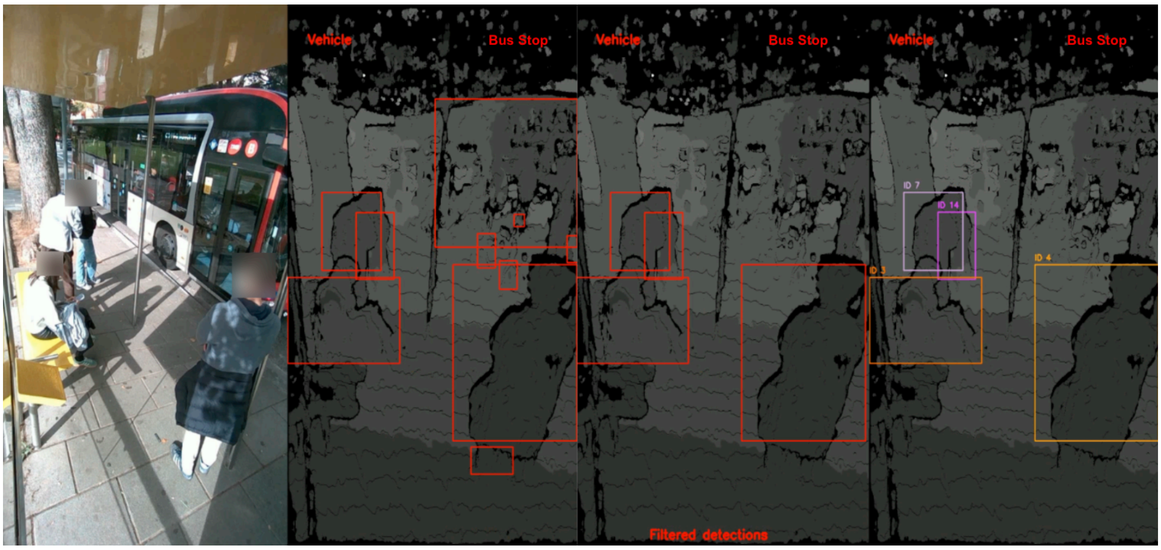

***Figure 8**: Point-cloud approach detection and tracking results containing a vehicle detection; From left to right: RGB, detections, filtered detections and tracks*

4.2 People Tracking

To evaluate tracking performance in this project, we adopt the **Higher Order Tracking Accuracy (HOTA)** metric (Luiten et al., 2021), commonly used to evaluate multiple object tracking tasks. HOTA measures how well the trajectories of matching detections align, and averages this over all matching detections, while also penalizing detections that do not match.

HOTA can be thought of as a combination of three different metrics (detection, association and localization) and calculates a score for each using an IoU (intersection over union) formulation.

$$IoU = \frac{|TP|}{|TP|+|FN|+|FN|} \qquad \textbf{(6)}$$

The localization measures the spatial adjustment between a predicted and a ground-truth detection. It is calculated as the ratio of the overlap (intersection) between the two detection bounding boxes and the total area covered by both of them (union). Localization Accuracy (LocA) is obtained by averaging the Loc-IoU over all pairs of predicted and ground-truth detections.

$$LocA = \frac{1}{|TP|}\sum_{c \in TP} Loc - IoU(c) \qquad \textbf{(7)}$$

The detection measures the alignment between the set of all predicted detections and the set of all ground-truth detections. To define which detections between the set of predicted and ground-truth detections are intersecting a localization threshold is defined. However, one predicted detection may overlap with more than one ground-truth detection. To determine single matches between predictions and ground truths the Hungarian algorithm is used. The matching pairs are called True Positives (TP), predicted detections without match are False Positives (FP) and ground-truth detections without match are False Negatives (FN). By finding computing these values over the whole datasets we obtain the Detection Accuracy (DetA):

$$DetA = Det - IoU = \frac{|TP|}{|TP|+|FN|+|FP|} \qquad \textbf{(8)}$$

Association measures evaluate a tracker’s ability to maintain consistent identities over time by correctly linking detections that belong to the same individual. Given a set of ground-truth tracks with known identities, and predicted tracks from the tracker, association is assessed by matching detections, typically using the Hungarian algorithm, and measuring how well the predicted track aligns with the corresponding ground-truth track across frames. The intersection between two tracks is the number are the True Positive Associations (TPA), left detections in the predicted track are False Positive Associations (FPA) and remaining detections in the ground-truth track False Negative Associations (FNA), see Figure 2. By averaging the Association IoU we obtain the Association Accuracy (AssA):

$$AssA = \frac{1}{|TP|}\sum_{c\in TP} Ass - IoU(c) = \frac{1}{|TP|}\sum_{c\in TP} \frac{|TPA(c)|}{|TPA(c)|+|FNA(c)|+|FPA(c)|} \quad \textbf{(9)}$$

Then by tacking these three components and combining them into a single metric we obtain HOTA where α is the localization threshold and the HOTA is calculated over a range of α values:

$$HOTA_\alpha = \sqrt{DetA_\alpha \cdot AssA_\alpha} = \sqrt{\frac{\sum_{c\in TP_\alpha} Ass-IoU_\alpha(c)}{|TP_\alpha|+|FN_\alpha|+|FP_\alpha|}} \quad \textbf{(10)}$$

$$HOTA = \int_0^1 HOTA_\alpha \, d\alpha \quad \textbf{(11)}$$

To evaluate tracking performance, we first used the YOLO-based detector to generate initial detections, which were then processed by the OC-SORT tracker to produce tracking results. We then compared these tracking results to the ground truth dataset using the TrackEval Python library, and the resulting metrics. After this process we obtained the metrics shown in Table 3. The HOTA, DetA and AssA values indicate a good performance.

These results align with the qualitative analysis where we visualized a bus stop recording detections and tracks (see Figure 8). In the recordings we observe how in some cases a same person can be identified multiple times (multiple tracks) which mainly happens when detections are temporarily lost, normally due to occlusions.

Overall, the results are very promising but in the final tracking results there are some identity switches which would make the waiting time metric not accurate enough. This is why in section 5 we propose several future work directions.

Table 3: *Performance of tracking using OC-SORT and YOLO-based person detections*

Approach	GT identities	Tracked identities	HOTA	AssA	DetA	LocA
OC-SORT (GT detections)	5	18	82.965	72.391	95.145	96.947
OC-SORT (YOLO detections)	5	16	82.456	72.951	93.256	96.971

5. Conclusions

In conclusion, this paper has studied the viability of using depth cameras and computer vision technology for bus stop monitoring. Its general conclusion indicates that this approach is viable and solves the privacy issue of installing traditional RGB cameras in public spaces.

In this paper, we conducted data acquisition at bus stops using a depth camera, recording under varying environmental conditions. We then implemented automatic people annotation using a pretrained object detection model (YOLO), followed by manual correction to produce a high-quality dataset containing depth images and bounding boxes corresponding to people in each frame. Using this dataset, we fine-tuned the pretrained YOLO model to obtain a depth-based people detection model. The resulting detections were then used for tracking with a state-of-the-art tracker (OC-SORT), which leverages detection coordinates at each time step to predict subsequent positions and consistently assign unique IDs across frames.

The depth-based people detector achieves strong performance, enabling accurate computation of people density at bus stops. In addition, classical computer vision techniques have proven effective when working directly with point clouds, allowing for reliable bus detection, an approach that could be further explored in future research. However, despite promising results, the current tracking system still suffers from occasional identity switches during occlusions. As a result, tracking cannot yet be reliably used to compute individual waiting times, since the tracking inconsistencies would lead to inaccurate measurements for some people.

Future directions of work include expanding the dataset to improve model generalization across more diverse stops enhancing detection performance by fusing point cloud-based methods with YOLO-based detections and using 3D information to incorporate a top-down perspective along with corresponding bounding boxes to more effectively handle occlusions.

6. Acknowledgements

This work has been partially supported by Transports Metropolitans de Barcelona through Smart Bus Stop project. Additional funding was provided by the projects Ajuda RED2024-154236-E and PID2020-117142GB-I00 [DeeLight project], funded by MICIU/AEI (10.13039/501100011033), and the SMARTIN project under grant agreement No. 101203240. The authors wish to formally acknowledge these projects for their role in supporting the broader research context in which this study has been conducted. We also thank Llorenç Sastre Galmés for annotating part of the recordings and for providing an improved YOLO trained model.

7. References-Bibliography

Bedmar Martínez, A. (2024). Analysis of users at bus stops with computer vision (Bachelor's thesis, Universitat Politècnica de Catalunya).

Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016, September). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464-3468). Ieee.

Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S. R., & Koltun, V. (2024). Depth pro: Sharp monocular metric depth in less than a second. arXiv preprint arXiv:2410.02073.

Cao, J., Pang, J., Weng, X., Khirodkar, R., & Kitani, K. (2023). Observation-centric sort: Rethinking sort for robust multi-object tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9686-9696).

Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

Hübner, P., Hou, J., & Iwaszczuk, D. (2023). Evaluation of Intel Realsense D455 Camera Depth Estimation for Indoor Slam Applications. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 48, 1207-1214.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2), 83-97.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., & Leibe, B. (2021). Hota: A higher order metric for evaluating multi-object tracking. International journal of computer vision, 129, 548-578

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

Velastin, S. A., Fernández, R., Espinosa, J. E., & Bay, A. (2020). Detecting, tracking and counting people getting on/off a metropolitan train using a standard video camera. Sensors, 20(21), 6251.

Viegas, M. M. (2019). Smart bus stop: people counting in a multi-view camera environment (Master's thesis, Universidade do Algarve (Portugal)).

Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., ... & Qiao, Y. (2023). Internimage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 14408-14419).

Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE

Zhang, Y., Zeng, W., Jin, S., Qian, C., Luo, P., & Liu, W. (2024, September). When pedestrian detection meets multi-modal learning: Generalist model and benchmark dataset. In European Conference on Computer Vision (pp. 430-448). Cham: Springer Nature Switzerland

Zheng, A., Zhang, Y., Zhang, X., Qi, X., & Sun, J. (2022). Progressive end-to-end object detection in crowded scenes. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 857-866).

Zong, Z., Song, G., & Liu, Y. (2023). Detrs with collaborative hybrid assignments training. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6748-6758).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In European conference on computer vision (pp. 213-229). Cham: Springer International Publishing.

Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., & Meng, H. (2023). Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, *25*, 8725-8737

Maggiolino, G., Ahmad, A., Cao, J., & Kitani, K. (2023, October). Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification. In *2023 IEEE International conference on image processing (ICIP)* (pp. 3025-3029). IEEE.

Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24, 6 (June 1981), 381–395.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. 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 (KDD'96). AAAI Press, 226–231.