scientific data

OPEN

DATA DESCRIPTOR

Pedestrian Trajectory Dataset of Public European Squares

Nils Wolff[1,4]✉, Layne Perry[1,2], Titus Venverloo[1,3], Geertje Slingerland[4], Jessica Wreyford[2], Paolo Santi[1] & Fábio Duarte[1]✉

Pedestrian trajectories are used to learn about human behavior in public space and the impact of spatial features on pedestrian flows. Currently, these trajectories are collected manually, with self-tracking devices, or with video cameras. Even when trajectories are obtained using computational techniques, such as using computer vision to trace them in space, these datasets are not made available for reproducibility or comparative studies between different locations. To close this gap, this paper makes available the data of pedestrian trajectories collected in 39 European squares. Firstly, we summarize the data collection process which was based on collecting footage from publicly available webcams. Secondly, we describe the process of trajectory extraction entailing object detection, tracking, and georeferencing. Lastly, we describe the data cleaning and validation steps that lead to the final dataset. The dataset ultimately includes 348,300 pedestrian trajectories extracted from 193 hours of video footage, collected at different times of the day, during working days and weekends, and during the Spring and Summer season.

Background & Summary

Pedestrian tracking has been used in various urban research fields, from the design of public squares[1–5] to the analysis of human behavior. Some examples can be found in crowd control[6–8], transport infrastructure optimization[8,9], or crowd animation in video productions[10].

For data collection in public places, researchers like Jan Gehl[11,12] relied on manual sketching of human trajectories. William H. Whyte[2] pioneered the use of video cameras to understand people's behavior in public spaces. By analyzing walking speeds, the distribution of pedestrians in an area, and their basic behavior (e.g., standing, sitting, or walking), researchers extracted information about the use of squares, the attractiveness of specific features, or the best design for paths and seating arrangements. Contemporary research has automated data collection and analysis using means such as computer vision[3,4,13]. The analysis of pedestrian trajectories has also been automated by giving a mathematical representation to specific usage patterns of place users[4,5].

Most available datasets on pedestrian trajectories have been created for the purpose of evaluating pedestrian tracking approaches[14–16]. Datasets for tracking evaluation come with the necessity of ground-truth data on trajectories, which makes the creation of such datasets resource intensive. Other datasets, collected for research in crowd behavior, are often limited to a specific scene[7,8,17], or even involve actors for specific crowd behavior[6,18,19].

This paper offers pedestrians' trajectories collected in 39 European squares. Data was originally collected to analyze the impact of street seating on place vitality, following the methodology of Niu *et al.*[4]. In addition, we looked at the impact of seasons and temperatures on the use of public places. But we found that one of the main gaps in the literature, even more when it comes to comparative studies in urban settings, is the availability of datasets based on common standards. This is the main contribution of our paper: to provide an open-source dataset of pedestrian trajectories in multiple public squares following standard data-collection and data-processing methods, which are also detailed here. The publication of our dataset will enrich quantitative research on the use of public squares, including the understanding of the impact of place design on its vitality, human interactions in public spaces, pedestrian modeling, or crowd behavior.

[1]Senseable City Laboratory, Massachusetts Institute of Technology, Cambridge, USA. [2]Laboratory of Geo-Information Science and Remote Sensing Wageningen University & Research, Wageningen, Netherlands. [3]Amsterdam Institute for Advanced Metropolitan Solutions, Amsterdam, Netherlands. [4]Architecture and the Built Environment, Delft University of Technology, Delft, Netherlands. ✉e-mail: wolffn@mit.edu; fduarte@mit.edu

Variable	Unit
Square area	m^2
City population	No.
Temperature - city level	°C
Windspeed - city level	m/s
Humidity - city level	%

Table 1. Square observation parameters.

Methods

The methods used for data collection and processing can be grouped into two steps; i) Square recording, describing the recording of place videos, and ii) Data processing, describing the steps needed to retrieve trajectories in real-world coordinates from video material.

Place recording. Many cities worldwide have webcams installed in public spaces. Webcam streams are available through platforms like YouTube. We reviewed two main databases of publicly available webcams[20,21], selecting webcams based on technical and methodological criteria:

- A public square had to be visible in the video.
- The stream had to be recordable through the VLC media player.
- The stream had to have a frame rate of at least 15 fps to ensure good tracking results.
- The stream had to be free of lag, compression artifacts, and pixel errors. Pedestrians had to be clearly visible in the video.

Recording times and dates. Based on the guideline "How to study public places" created by Jan Gehl[11] and the observations by William H. Whyte[2], we recorded each place in four different time slots, for 30 minutes each in June 2024.

- Morning: 8:00 – 8:30 local time
- Noon: 12:30 – 13:00 local time
- Evening: 18:00 – 18:30 local time
- Saturday: 12:30 – 13:00 local time

To also add seasonal data to our research, we additionally recorded four squares (Assisi, Biberach, Celanova, and Varberg) between 25th of March and 22nd of June 2025. Ultimately, we collected two datasets using the same methodology. A comparative dataset containing 39 squares with four recordings each was collected investigating the use of street seating. We therefore did not record places when it was raining or when events (e.g. markets and parades) changed the general use of the place. When necessary and possible, recordings were repeated on another day to ensure a full set of data. The seasonal dataset containing 4 places was collected to learn how seasonal changes influences the use of public squares, so we collected data in different weather conditions.

Webcams available online generated streams with various resolutions, frame rates, and duration. In the first step of data processing, we down-sampled the resolution of the videos to HD (1280 x 720 pixels). When higher, the frame rate was reduced to 15 frames per second to guarantee consistency across the collected footage. To ensure a uniform video length, each stream was recorded for exactly 30 minutes.

Parameters of place recording. For each recording and for each recorded square, various general parameters have been collected (see Table 1). Using a Python script, we retrieved the weather information from World weather API[22] for each recording. To limit the analysis to the square, we have excluded adjacent streets and parking lots by creating polygons over the recorded videos. Table 1 summarizes the recording parameters, including corresponding units.

Data processing. The extraction of trajectories from the video material included three steps (see Fig. 1): object detection, object tracking, and georeferencing. This section describes the steps to extract pedestrian locations from the raw video footage (object detection), to detect humans through multiple frames (object tracking), and ultimately to transform the extracted pixel coordinates into real-world coordinates (georeferencing).

Object detection. In line with similar research in the urban domain[4,5], we have utilized the YOLO (You Only Look Once) object detection algorithm. YOLO is available as an open-source Python package and compatible with various object trackers. In addition to the selection of a specific object detection model, datasets used for training such models have a significant impact on the model's performance[23]. A model mainly trained on humans can detect humans better than a model trained on various classes, such as trees, cars, animals, and humans[24,25]. In the context of training and evaluating human detection models, the COCO dataset[26] and the crowdhuman[27] dataset stand out. YOLO comes in several versions with multiple pre-trained models based on various training datasets. For this research, we tested various combinations of versions and models[28–32]. To evaluate the models' performance, we validated the models on the crowdhuman dataset. Following Raino *et al.*[33],

Fig. 1 The steps of trajectory extraction from **a**) Raw video material: **b**) Object detection resulting in identification of pedestrians; **c**) object tracking resulting in trajectories in pixel coordinates; and **d**) georeferencing resulting in trajectories in geo-coordinates.

Model (pretrained)	Source	Precision	Recall	mAP (50)	mAP (95)
YOLOv8m	[30]	0.671	0.437	0.513	0.273
YOLOv8m_crowdhuman	[32]	0.876	0.661	0.725	0.509
YOLOv8_humancount	[28]	0.032	0.052	0.017	0.005
YOLOv8l_world	[30]	0.624	0.471	0.530	0.278
YOLOv8x-world	[30]	0.625	0.470	0.530	0.279
YOLOv9c	[48]	0.673	0.441	0.520	0.279
YOLOv10l	[31]	0.679	0.419	0.504	0.274
YOLOv10x	[31]	0.674	0.426	0.501	0.271
Model (custom trained)	**Source**	**Precision**	**Recall**	**mAP (50)**	**mAP (95)**
YOLOv10l_crowdhuman	(custom)	0.871	0.684	0.787	0.528

Table 2. Evaluation results for different YOLO object detection models.

we compared the models based on precision, recall, the mean average precision (mAP) for an "intersection over Union" (IoU) of 50 (50% of the predicted object overlap with the ground truth object) and 95 (95% of the predicted object overlap with the ground truth object).

From the comparison of different YOLO object detection models, which were pre-trained on a variety of datasets, YOLOv8m pre-trained on the crowdhuman dataset[32] stood out as the best object detection model – see Table 2. Validated on the crowdhuman dataset[27], it achieved high scores for precision and recall even for a strict IoU of 95. To utilize the speed of the newer version of YOLOv10[31] and the accuracy of a custom-trained model, the crowdhuman dataset was downloaded and adjusted to the YOLO dataset structure. YOLOv10l has been trained on it for 300 epochs. The training was computed on an NVIDIA RTX 3070 Ti with 8GB of GPU-RAM and 64GB of RAM. The process took 72 hours and resulted in significant improvements in the models' detection, as highlighted in Table 2. Compared to the pre-trained YOLOv10l model, the precision increased by 23%, and recall increased by 38%. Compared to the pre-trained YOLOv8m_crowdhuman model, the precision decreased by 0.5% while recall increased by 3.3%. Also, mAP 50 and mAP 95 showed higher scores for the custom-trained YOLOv10l model. This is why it was ultimately chosen for the research.

Object tracking. Tracking algorithms generally compare detected objects in frame n with the objects in frame $n+1$. By assigning IDs to the objects, similar objects on consecutive frames can be assigned with the same ID. To reduce the complexity of the comparison of objects in each frame, tracking algorithms estimate the trajectory of each object and only compare objects in the estimated pixel space of a frame with the detected object of a previous frame. The result is continuity between frames[34,35]. Figure 2 visualizes the object tracking process in the example of human detection.

Tracking algorithms must mitigate various challenges in the trajectory extraction process[34–38]. Firstly, they need to cope with the occlusion of detected objects. For the perfect trajectory extraction, a person must keep their ID even after being occluded by an object (e.g. a tree) for a few seconds. This means that the tracking algorithm must remember every detected object for a predefined number of frames (or predefined duration) and continue to compare these "buffered" IDs to objects in future frames. Secondly, tracking algorithms need to mitigate false positive detections of the object detection algorithms. An ID should not be given to a shadow that is falsely detected as a human for just one frame. And thirdly, IDs should not be swapped between objects with high similarity ("ID swap").

The multi-object tracker (MOT) ByteTrack[38] is closely integrated into the Python package "ultralytics" together with the YOLOv10[39]. ByteTrack is a capable tracker and has outperformed multiple available trackers[38]. ByteTrack comes with multiple hyperparameters that needed to be tuned to mitigate the challenges of occlusion and ID swap.

Applying the tracking algorithm ByteTrack resulted in videos with annotated pedestrians that were used for debugging, hyperparameter tuning, and general understanding of the scene. Furthermore, the results of the

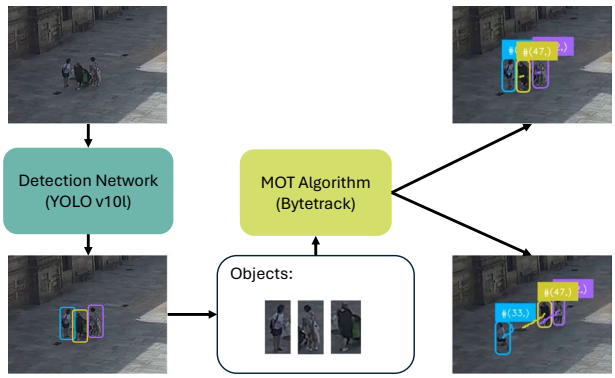

Fig. 2 A pictorial explanation the process of multi-object-tracking as adapted from Hassan *et al.*[49]. The objects detected by a detection network are labeled with unique IDs by the tracking algorithm.

Parameter name	Description	Value
track_activation_threshold	min. confidence of detection to start tracking [%]	0.30
lost_track_buffer	number of frames that is buffered for re-identification of objects	125
minimum_matching_threshold	similarity between consecutive detection [%]	0.95
minimum_consecutive_frame	number of consecutive detections to start tracking	3

Table 3. Hyperparameters for ByteTrack.

tracking were stored in a CSV file for further calculations. Optimal parameters for the use of ByteTrack in our video dataset are described in Table 3.

Georeferencing. As described in the previous section, the location of humans (in fact: their bounding boxes) are described in pixel coordinates[31,38]. Pixel coordinates do not allow for any implications about depth information (how far away) or information about walking speed. Due to the perspective of the webcams, humans with the same walking speed move fewer pixels in the back of the picture than in the front.

To extract information such as walking speed or user density, the pixel coordinates must be translated into real-world coordinates. In computer vision, this is usually done by creating a transformation matrix that maps the pixel coordinates into real-world coordinates. For such a matrix, multiple pieces of information about the camera are needed, such as lens distortion, the height of the camera, and its rotation in each direction[40]. In this research, this information was unknown, as the video material was collected from webcams available online. Therefore, an estimation of the transformation matrix had to be found. This research followed the methodology called 'homography', used in similar research by Loo *et al.*[13]. Homography suggests that two images of the same plane (in our case: the square) contain the same information, just differently transformed[41]. If multiple features can be identified in both planes, a transformation matrix (also called homography matrix) can be calculated. The homography process is less precise than working with a hardware-based transformation matrix, but the estimation of the location of humans has been proven to be precise enough for the analysis of places[13].

In this research, two images of each place were available: The image of the camera and satellite imagery from Google. Google satellite imagery is visualized in the Pseudo-Mercator projection (EPSG:3857). Using homography, the coordinates of the original image from the camera were therefore transformed into the ESPG:3857 coordinates of the satellite imagery. From this coordinate system, a transformation in any other world coordinate system is possible. Features of the square imagery could be any fixed object that was visible in both images, such as street furniture, corners of buildings, pavement, stairs, or trees. Figure 3 shows a selection of place features that were identified in pixel and real-world coordinates for the example of Wolsztyn square, in Poland.

For the selection of corresponding feature pairs, the "Georeferencer" of the open-source geo-information software QGIS[42] was used, which stores the location of each selected feature in pixels and geo-coordinates. A Python script based on the package OpenCV[43] was used to calculate the homography matrix from the extracted feature points[44]. The homography matrix was finally applied to transform every pixel coordinate into geo-coordinates. The homography matrix had to be created for each square individually.

Data Processing. To ensure a dataset with complete trajectories and reproducibility, this section shortly outlines the process of data processing. Figure 4 visualizes the relevant steps and provides detailed information about the parameters we have used.

First, trajectories with a length of less than 30 data points (duration of 2 seconds) are removed. Using homography, real-world coordinates are calculated for each data point. A sliding window is applied to smooth the trajectory, avoiding noisy data. Based on the real-world information, coordinates outside the square-boundaries are removed. We down-sampled the trajectories to two data points per second to reduce complexity. Based on these filtered data points, noise-sensitive information, such as speed or number of people per frame, is calculated

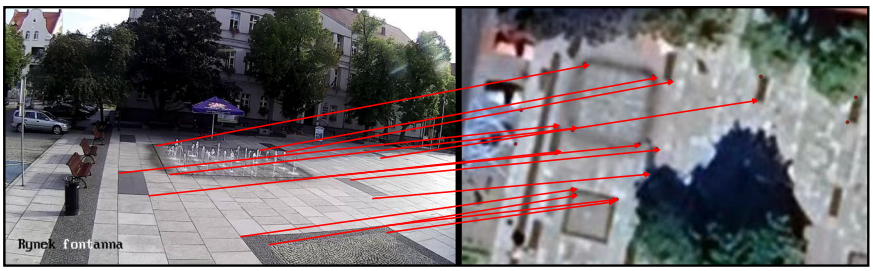

Fig. 3 The mapping of pixel coordinates (left) to real-world coordinates (right).

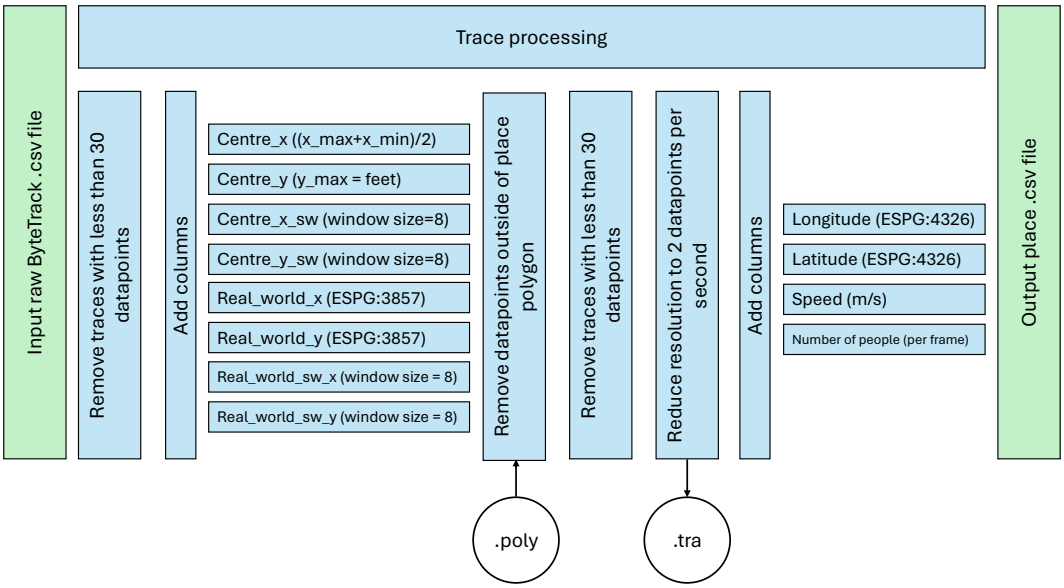

Fig. 4 Full process of data processing.

and added to the data structure. The trajectories are provided in both .csv and .geojson formats to enable various applications.

Data records

The dataset[45] is available on zenodo.org. An overview of the observed squares is provided in the file "`Place_overview.xlsx`". It describes each observed square, including access link to the recorded webcam and an individual square number ("No."). This number can be utilized to filter the dataset for a specific square. The files "`stats_season.csv`" and "`stats_comparative`" contain weather information and general information about the recordings. The dataset contains two main folders:

The folder "`01.Trajectories`" contains raw trajectory data, published in the subfolder "`01.Trajectories_raw`" containing the trajectories as extracted by ByteTrack without the processing or filtering steps. The columns of each trajectory file are described in Table 4.

The subfolder "`02.Trajectories_modified`" contains the final trajectory dataset. The modified trajectory files are described in Table 5. These trajectories are also available in geojson format.

The second main-folder "`02.Resources`" contains files that are used during the processing steps. More specifically, these are:

- The homography matrix used for each square in the folder "`01.Geotransformation`".
- Polygon files (.csv) for the real-world coordinates and the pixel-coordinates of the observed square outlines. The folder "`02.Polygons`" also includes images of these polygons.
- Pictures from each square can be found in the folder "`03.Place_images`".

Ultimately, the final dataset includes 391 recordings distributed over 39 squares in Europe entailing a total of 348,200 trajectories, obtained after cleaning and processing the original dataset. To ensure reproducibility, the dataset entails location data in pixels and real-world coordinates, as well as longitude and latitude (degree). Each trajectory has a specific ID, ensuring a clear identification of single pedestrians.

Variable name	Description
x_min	the left edge of the bounding box in pixel coordinates
x_max	the right edge of the bounding box in pixel coordinates
y_min	the top edge of the bounding box in pixel coordinates
y_max	the bottom edge of the bounding box in pixel coordinates
class_id	The ID of a class follows the definitions of the COCO dataset. In this research, always '0' for the class human
confidence	percentage of confidence that the bounding box describes the specific class
tracker_id	a unique ID for every detected human
frame_index	The video frame on which the detection was performed. For a video with 15 frames per second (fps), there is a time difference of 0.067 seconds between every frame.

Table 4. Column description for the raw trajectory files.

Variable Name	Description
x_min	The left edge of the bounding box in pixel coordinates
x_max	The right edge of the bounding box in pixel coordinates
y_min	The top edge of the bounding box in pixel coordinates
y_max	The bottom edge of the bounding box in pixel coordinates
centre_x	Calculated as (x_max + x_min)/2
centre_y	Equal to y_max
centre_x_sw	Sliding window over centre_x (window size = 8)
centre_y_sw	Sliding window over centre_y (window size = 8)
real_world_x	centre_x transformed into EPSG:3857
real_world_y	centre_y transformed into EPSG:3857
real_world_sw_x	Sliding window over real_world_x (window size = 8)
real_world_sw_y	Sliding window over real_world_y (window size = 8)
longitude	real_world_sw_y in EPSG:4326
latitude	real_world_sw_x in EPSG:4326
class_id	The ID of a class follows the definitions of the COCO dataset. In this research, always '0' for the class human
confidence	Percentage of confidence that the bounding box describes the specific class
tracker_id	A unique ID for every detected human
frame_index	The video frame on which detection was performed. For a video with 15 frames per second (fps), there is a time difference of 0.067 seconds between every frame.
speed	Speed in m/s calculated based on longitude and latitude
number_people	Number of unique tracker IDs calculated for each frame

Table 5. Column description for the modified trajectory files.

Technical Validation

The technical validation of the collected data must be separated into the validation of the object detection, the validation of the object tracking, and the validation of the georeferencing.

Validation of object detection. For the evaluation of the object detection, we used the precision and recall metrics. Precision measures the accuracy of a model in positively predicting instances (here: humans). Recall measures the model's ability to identify all relevant instances. We extracted 50 random frames (while including at least one frame per square) from the total video dataset and counted for each frame the positive and negative detections, as well as the false positive and false negative detections. Based on this, we calculated precision and recall evaluating the performance of the applied object detection algorithm on the research-specific dataset.

Over all instances (humans) in the sample data, object (human) detection achieved a precision of 93.8% (Fig. 5, left) and a recall of 92.8% (Fig. 5, right).

Outliers with a precision of 0% have been observed primarily in squares with a human-like statue. When the place was not visited by a pedestrian in the observed frame, this single false-positive detection results in a precision of 0%. Outliers with 0% recall were observed in squares where a single visitor was present but not detected. Squares with more visitors resulted in very good precision and recall and therefore good object detection results.

Validation of trajectory precision. Tested on various test datasets, the object tracker Bytetrack archives a "multi-object tracking accuracy" (MOTA) of 80.3% on the pedestrian centered MOT17 benchmark[46]. The more accurate "higher order tracking accuracy" (HOTA) test results in an accuracy of 63.1% on the same dataset[38]. Due to the shear complexity of manual labeling of trajectories for ground truth labeling, we are not able to generate a ground truth dataset. Instead we approach the validation of trajectory quality in a two step process. In the first step, we inspect the quality of trajectories based on their length. In the second step, we manually extract the completeness and number of confused IDs for a subsample of 200 trajectories.

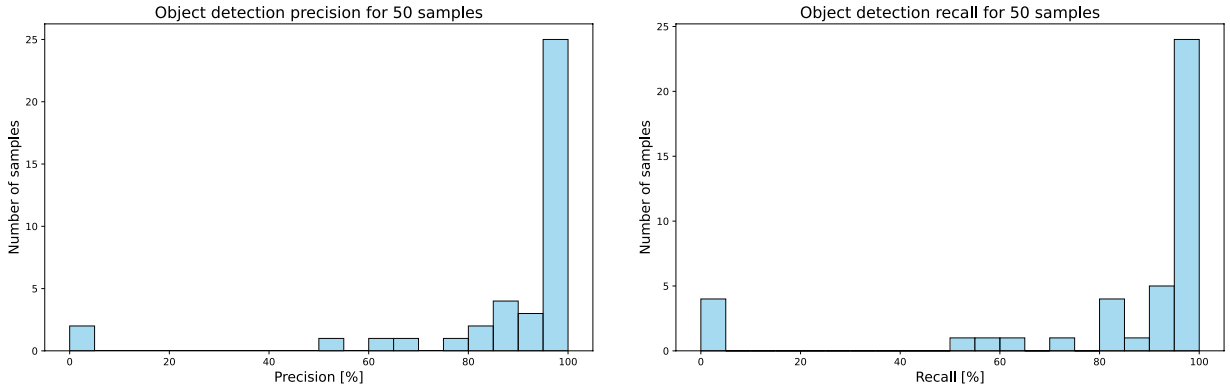

Fig. 5 Precision (left) and recall (right) of YOLOv10l for a sample of n = 50 frames.

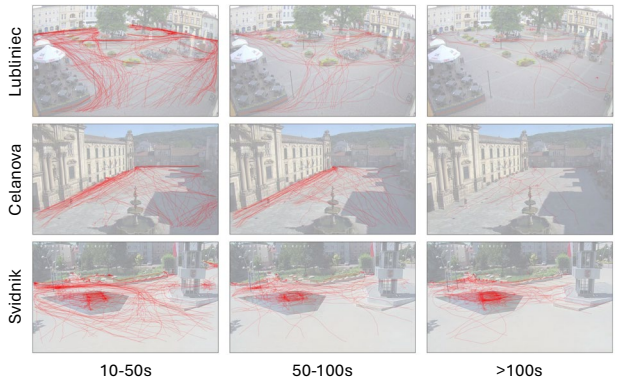

Fig. 6 Density of trajectories with a duration of 10–50s (left), 50–100s (middle), and more than 100s (right) on the example of Lubliniec (top), Celanova (middle), and Svidnik (bottom).

In the processing of trajectory data, we deleted trajectories with a duration shorter than 2 seconds, avoiding a dataset full of misclassified or re-identified pedestrians. This results in a dataset with an overall median duration of trajectories of 32 seconds and mean duration of 54 seconds. Figure 6 visualizes the quality of trajectories depending on their duration, indicating a high number of complete trajectories longer than 10 seconds. Table 6/I summarizes the abundance of trajectories per time interval. The majority of trajectories has a duration between 2 and 50 seconds. Nevertheless, the dataset also contains more than 39 thousand trajectories with a duration between 50 and 100 seconds and more than 17 thousand trajectories longer than 100 seconds.

To assess the trajectory quality, namely the completeness and confusion of IDs, we randomly sampled 200 trajectories across different time frames (duration of 2-10s, 10-50s, 50-100s, and >100s) as summarized in table 6/I. The sampling was executed in two steps, where in the first step only 10% of all trajectories per time frame and recording are kept, while ensuring at least 5 trajectories per subset. In a second step, 50 trajectories are randomly selected for each time frame. This results in a diverse sample-set ensuring the representation of various locations. For each sampled trajectory we created an annotated video clip to allow a manual comparison of the tracking result with the recorded pedestrian. In line with the object detection accuracy, 9 % of the sampled IDs are false positive (FP) classifications such as trees or statues. We observed that the tracker fails to consistently re-identify seating or standing pedestrians. This is likely due to the architecture of MOT algorithms like Bytetrack which heavily rely on information coming from motion[38]. Re-identification of pedestrians, e.g. after occlusion, is therefore less likely for stationary pedestrians. As shown in Table 6/II 0% of the stationary trajectories are classified as complete while more than 56% of the moving trajectories are classified as complete. To account for this, we divided the sample into stationary pedestrians (seating or standing) and moving pedestrians.

We then rated each sampled ID based on two features; completeness, and ID merges. Completeness describes to what extent an ID covered the full trajectory of a pedestrian (also called ID recall). An ID describing the full trajectory of a pedestrian in a square is considered complete. When an ID only describes a fraction of a trajectory, we estimated the full trajectory and state the covered fraction. When an ID describes multiple pedestrians (called “ID merge"), we estimate the completeness for the pedestrian that is assigned to the ID for the longest period. We account for ID merges by total number (how often was an ID assigned to an additional pedestrian) and per trajectory (How many IDs were assigned to one or multiple other pedestrians). Table 6/III summarizes the results. The highest completeness is found for trajectories with a duration between 50 and 100 seconds. Naturally, long trajectories include more seating pedestrians, reducing the average completeness and increasing

Trajectory duration:	2-10s	10-50s	50-100s	>100s	All
I. Sample Information					
Total Number Trajectories	118,315	173,430	39,062	17,447	348,254
Sampled Trajectories (n)	50	50	50	50	200
7D2Valid Trajectories (n_{valid})	43	43	48	44	182
Stationary Trajectories (Count)	13	10	13	15	51
Moving Trajectories (Count)	30	37	35	29	131
II. Proportion of Complete Trajectories					
All Samples	25.6%	48.9%	52.1%	34.1%	40.7%
Stationary Samples	0.0%	0.0%	0.0%	0.0%	0.0%
Moving Samples	36.7%	62.2%	71.4%	51.7%	56.5%
III. Tracking Quality					
Average Completeness (ID Recall %)	49.3%	78.3%	91.1%	86.9%	77.4%
Total ID Merges (Count)	5	3	5	14	27
Trajectories with ≥1 ID Merge (Count)	4	2	4	10	21
Proportion of Samples with ID merges	13.3%	8.1%	11.4%	34.4%	16.0%

Table 6. Trajectory validation results for a sample of $n = 200$ split across four duration time frames.

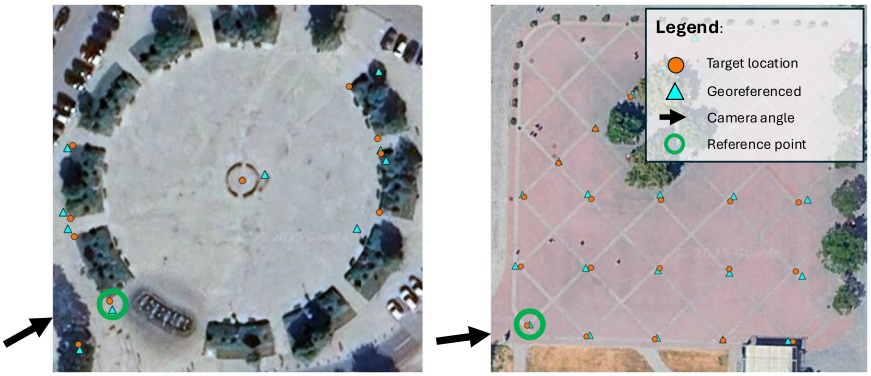

Fig. 7 Accuracy of georeferencing for Ozórkow (left) and Zlín (right).

Square	avg. Error_ns	avg. Error_ew	avg. Error_total	Error_range_total
Zlín	0.11m	0.7m	0.3m	0-2.2m
Ozórkov	1.44m	1.59m	2.58m	1.5-10.9m

Table 7. Validation of georeferencing using the example of Zlín and Ozórkow.

the chance of ID merges. Shorter trajectories tend to be incomplete, indicating a higher share of ID swaps (multiple IDs are assigned to one pedestrian). Overall, 56.5% of the sampled, moving trajectories are complete and 84% of the sampled IDs describe not more than one pedestrian.

Validation of georeferencing. The georeferencing of pixel coordinates is based on the visual methodology of homography. Ultimately, the homography matrix is an approximation based on given cues (features in world-, and pixel-coordinates). The quality of the geotransformation is therefore highly dependend on the availability of these cues on the observed squares. To assess the quality of the georeferencing, we selected a set of features in pixel coordinates and transformed them using the same homography applied to the trajectory data. The georeferencing error was then quantified by comparing the transformed feature locations with their corresponding ground-truth positions.

Base map layers are regularly updated which sometimes changes the orientation of the base map to the reference points. To avoid the impact of this characteristic on the assessment, we calculate the distance between all transformed feature locations to one reference point (See Fig. 7). Comparing the resulting distances (north to south, east to west, and total) to the corresponding distances of the ground truth locations, allows us to calculate the error of the georeferencing. Table 7 shows the accuracy for two different squares. The square of Zlín is paved with a grid like structure which increases the quality of georeferencing. As visible in Fig. 7, georeferenced and ground truth location are relatively close together. In contrast, the square in Ozórkow only provides irregular features for georeferencing, which results in larger errors, especially farther away from the camera.

Usage Notes

The processing steps performed on this dataset were intended to reduce the number of faulty trajectories. However, object tracking comes with limitations that also affect the quality of this dataset. These limitations are the confusion of IDs and fragmentation due to occlusion or faulty object detection. We modified the hyperparameters of ByteTrack to overcome these limitations as much as possible. Nevertheless, this dataset ultimately contains fragmented trajectories, trajectories of human shadows, and swapped or merged IDs. The validation of trajectory quality revealed low completeness and ID consistency for short or stationary trajectories. We generally did not see a big difference in tracking quality between the observed squares. Inherently, crowded scenarios come with more occlusion, and therefore with a higher number of confused IDs. For many research questions, these limitations are not relevant. For others, these limitations might affect the research results. By providing the full dataset, code, and descriptor, we aim to enable customized research based on this dataset.

Data availability

The dataset[45] is available on zenodo.org.

Code availability

The Python code containing the preprocessing steps of the video files, the object-detection- and tracking and the processing of the trajectory files can be found on GitHub and is published on Zenodo[47].

Received: 28 July 2025; Accepted: 22 January 2026;
Published online: 10 February 2026

References

1. Gehl, J. *Life between Buildings* (Van Nostrand Reinhold, 1987).
2. Whyte, W. H. *The Social Life of Small Urban Spaces* (Project for Public Spaces, 1980).
3. Hou, J., Chen, L., Zhang, E., Jia, H. & Long, Y. Quantifying the usage of small public spaces using deep convolutional neural network. *PLOS ONE* **15**, e0239390 (2020).
4. Niu, T. *et al.* Small public space vitality analysis and evaluation based on human trajectory modeling using video data. *Building and Environment* **225**, 109563 (2022).
5. Ceccarelli, G., Messa, F., Gorrini, A., Presicce, D. & Choubassi, R. Deep learning video analytics for the assessment of street experiments: The case of Bologna. *Journal of Urban Mobility* **4**, 100067 (2023).
6. Boomers, A. *et al.* Pedestrian Crowd Management Experiments: A Data Guidance Paper. *Colective Dynamics* 1–57, https://doi.org/10.48550/arXiv.2303.02319 (2023).
7. Lerner, A., Chrysanthou, Y., Shamir, A. & Cohen-Or, D. Context-Dependent Crowd Evaluation. *Computer Graphics Forum* **29**, 2197–2206 (2010).
8. Strigel, E., Meissner, D., Seeliger, F., Wilking, B. & Dietmayer, K. The Ko-PER intersection laserscanner and video dataset. *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014* 1900–1901, https://doi.org/10.1109/ITSC.2014.6957976 (2014).
9. Boltes, M. & Seyfried, A. Collecting pedestrian trajectories. *Neurocomputing* **100**, 127–133 (2013).
10. Shao, W. & Terzopoulos, D. Autonomous pedestrians. *Graphical Models* **69**, 246–274 (2007).
11. Gehl, J. & Svarre, B.*How to study public life* (Island Press, Washington, 2013).
12. Gehl, J. *Cities for People* (Island Press, 2013).
13. Loo, B. P. Y. & Fan, Z. Social interaction in public space: Spatial edges, moveable furniture, and visual landmarks. *Environment and Planning B: Urban Analytics and City Science* **50**, 2510–2526 (2023).
14. Chavdarova, T. *et al.* WILDTRACK: A Multi-camera HD Dataset for Dense Unscripted Pedestrian Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 5030–5039, https://doi.org/10.1109/CVPR.2018.00528 (2018).
15. Ristani, E., Solera, F., Zou, R., Cucchiara, R. & Tomasi, C. Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **9914 LNCS**, 17–35 (2016).
16. Sun, P. *et al.* DanceTrack: Multi-Object Tracking in Uniform Appearance and Diverse Motion. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2022-June**, 20961–20970 (2021).
17. Pellegrini, S., Ess, A., Schindler, K. & Van Gool, L. You'll never walk alone: Modeling social behavior for multi-target tracking. *Proceedings of the IEEE International Conference on Computer Vision* 261–268, https://doi.org/10.1109/ICCV.2009.5459260 (2009).
18. Adrian, J., Seyfried, A. & Sieben, A. Crowds in front of bottlenecks at entrances from the perspective of physics and social psychology. *Journal of the Royal Society Interface* **17**, https://doi.org/10.1098/rsif.2019.0871 (2020).
19. Usten, E., Ugering, H. L. & Sieben, A. Pushing and Non-pushing Forward Motion in Crowds: A Systematic Psychological Observation Method for Rating Individual Behavior in Pedestrian Dynamics. *Collective Dynamics* **7**, 1–16 (2022).
20. WorldCam Webcams from around the globe. https://worldcam.eu/webcams/europe.
21. Live HD Streaming Webcams from around the globe https://www.webcamtaxi.com/en/ (2024).
22. Free Weather API - WorldWeatherOnline.com. https://www.worldweatheronline.com/weather-api/.
23. Shao, S. *et al.* Objects365: A large-scale, high-quality dataset for object detection. *Proceedings of the IEEE International Conference on Computer Vision* **2019-October**, 8429–8438 (2019).
24. Girshick, R. *et al.* Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1**, 5000 (2014).
25. Li, J. *et al.* Domain Adaptation based Object Detection for Autonomous Driving in Foggy and Rainy Weather. *IEEE Transactions on Intelligent Vehicles* 1–12, https://doi.org/10.1109/TIV.2024.3419689 (2023).
26. Lin, T.-Y. *et al.* Microsoft COCO: Common Objects in Context. In Fleet, D., Pajdla, T., Schiele, B. & Tuytelaars, T. (eds.) *Computer Vision,Äı ECCV 2014*, 740–755, https://doi.org/10.1007/978-3-319-10602-1_48 (Springer International Publishing, Cham, 2014)
27. Shao, S. *et al.* CrowdHuman: A Benchmark for Detecting Human in a Crowd https://doi.org/10.48550/arXiv.1805.00123 (2018).
28. J3lly-Been. Yolov8-HumanDetection https://github.com/J3lly-Been/YOLOv8-HumanDetection (2024).
29. Jocher, G. YOLOv5 by Ultralytics https://github.com/ultralytics/yolov5 (2020).
30. Jocher, G., Chaurasia, A. & Qiu, J. Ultralytics YOLO https://github.com/ultralytics/ultralytics (2023).
31. Wang, A. *et al.* YOLOv10: Real-Time End-to-End Object Detection https://doi.org/10.48550/arXiv.2405.14458 (2024).
32. Wangjianan. Graduation_Project_MOT https://github.com/wangjianan-123/Graduation_project_MOT (2023).
33. Rainio, O., Teuho, J. & Klén, R. Evaluation metrics and statistical tests for machine learning. *Scientific Reports 2024 14:1* **14**, 1–14 (2024).
34. Ciaparrone, G. *et al.* Deep learning in video multi-object tracking: A survey. *Neurocomputing* **381**, 61–88 (2020).
35. Luo, W. *et al.* Multiple object tracking: A literature review. *Artificial Intelligence* **293**, 103448 (2021).

36. Chu, Q. *et al*. Online Multi-object Tracking Using CNN-Based Single Object Tracker with Spatial-Temporal Attention Mechanism. *Proceedings of the IEEE International Conference on Computer Vision* **2017-October**, 4846–4855 (2017).
37. Wen, L. *et al*. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding* **193**, 102907 (2020).
38. Zhang, Y. *et al*. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **13682 LNCS**, 1–21 (2021).
39. Home - Ultralytics YOLO Docs https://docs.ultralytics.com/ (2024).
40. Hartley, R. & Zisserman, A.*Multiple view geometry in computer vision*, 2 edn. https://doi.org/10.1017/CBO9780511811685 (Cambridge University Press, 2004)
41. Hartley, R., Gupta, R. & Chang, T. Stereo from uncalibrated cameras. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1992-June**, 761–764 (1992).
42. QGIS Development Team. QGIS Geographic Information System http://www.qgis.org. (2024).
43. Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* https://opencv.org/ (2000).
44. OpenCV: Basic concepts of the homography explained with code. https://docs.opencv.org/4.x/d9/dab/tutorial_homography.html.
45. Wolff, N. & Perry, L. Pedestrian Trajectory Dataset of Public European Squares https://doi.org/10.5281/zenodo.18267205 (2025).
46. Dendorfer, P. *et al*. MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking. *International Journal of Computer Vision* **129**, 845–881 (2021).
47. Wolff, N. kaktusracing/pedestrian_trajectories: 2025-Dec-08_code_pedestrian_trajectories (v1.0.1). https://doi.org/10.5281/zenodo.18266937 (2025).
48. Wang, C.-Y., Yeh, I.-H. & Mark Liao, H.-Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Leonardis, A. *et al*. (eds.) *Computer Vision, Ät ECCV 2024*, 1–21 (Springer Nature Switzerland, Cham, 2025).
49. Hassan, S., Mujtaba, G., Rajput, A. & Fatima, N. Multi-object tracking: a systematic literature review Multimedia Tools and Applications **83**(14), 43439–43492, https://doi.org/10.1007/s11042-023-17297-3 (2024).

Acknowledgements

The authors would like to thank AMS Institute and all members of the MIT Senseable City Consortium (including A2A, Abu Dhabi's Department of Municipal Transportation, City of Laval, City of Rio de Janeiro, Consiglio per la Ricerca in Agricoltura e l'Analisi dell'Economia Agraria, Dubai Future Foundation, FAE Technology, UnipolTech, Hospital Albert Einstein, Seoul AI Foundation, Sondotécnica Engenharia, Toyota Woven City) for supporting this research.

Author contributions

N.W., J.W., and G.S. defined the research and data collection plan, N.W. and L.P. collected the trajectory data and processed the data supervised by J.W., G.S. and T.V. The manuscript was prepared by N.W. and supervised by F.D., and P.S.. F.D., P.S., and T.V. jointly revised the paper and dataset.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to N.W. or F.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.