OrienterNet: Visual Localization in 2D Public Maps with Neural Matching

CVPR 2023

Appendix

A. Additional results

Which map elements are most important? We study in Fig. 1 the impact of each type of map element on the final accuracy by dropping them from the input map. The classes with the largest impact are buildings and road, which are also the most common in areas covered by the training data.

Impact of the field-of-view: We study the impact of the FoV on the accuracy by cropping the images in the horizontal direction to varying degrees. Figure 2 shows the results on the MGL validation set. Reducing the FoV decreases the accuracy proportionally – a 50% smaller FoV results in half of the original accuracy.

Qualitative results: We show additional examples of single-image predictions in Fig. 3 and failure cases in Fig. 4.

B. Data processing and distribution

B.1. OpenStreetMap

Map classes: OpenStreetMap [2] exposes for each element a set of tags with standardized categories and labels according to a very rich hierarchy. We group elements into a smaller set of classes that we list in Tab. 1, resulting in 7 types of areas, 10 types of lines, and 33 types of points (nodes). Figure 7 shows the distribution of such elements for a small area. Figure 8 shows how some OSM tags are mapped to some semantic classes.

Coordinate system: The coordinates of the map elements are given in WGS84 coordinates (longitude and latitude). We convert them to a local scaled Mercator datum centered at the median camera pose of each area. This yields topocentric coordinates that are aligned with the East and North axes.

B.2. Mapillary Geo-Localization dataset

Curation process: We browsed the Mapillary platform and looked for sequences that were sufficiently recent and with the most accurate ground truth poses. We selected sequences recorded after 2017 and with cameras known for resulting in good reconstructions. These include the Xiaomi Yi Action 2K (fisheye) or GoPro Max, MADV QJXJ01FJ, or LG-R105 (spherical) cameras. We selected 12 cities, listed in Tab. 2, that have a high density of such sequences. Figure 6 shows maps overlayed with the selected sequences. Images of each location were split into disjoint training and validation sets, resulting in 826k training and 2k validation views.

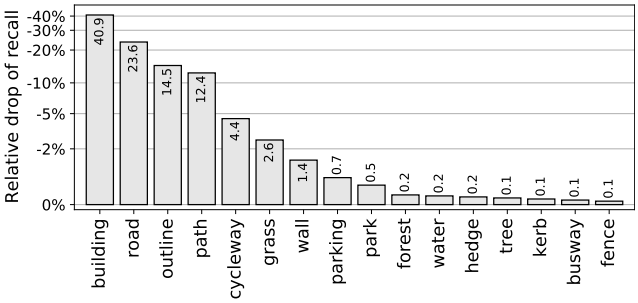

Figure 1. **Good semantics to localize.** Removing different elements from the map reveals how important they are for localization. Buildings, roads, footpaths, and cycleways are the most useful semantic classes, likely because they are also the most frequent ones.

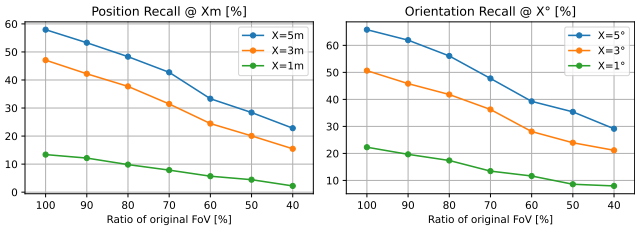

Figure 2. **Impact of the field of view** on the localization recall with the MGL validation set. Decreasing the FoV directly impairs the accuracy as fewer map elements are visible in a single image.

type	classes
areas	parking spot/lot, building, grass, playground, park, forest, water
lines	road, cycleway, pathway, busway, fence, wall, hedge, kerb, building outline, tree row
nodes	parking entrance, street lamp, junction, traffic signal, stop sign, give way sign, bus stop, stop area, crossing, gate, bollard, gas station, bicycle parking, charging station, shop, restaurant, bar, vending machine, pharmacy, tree, stone, ATM, toilets, water fountain, bench, waste basket, post box, artwork, recycling station, clock, fire hydrant, pole, street cabinet

Table 1. **List of map classes derived from OpenStreeMap data** and included in the map rasters.

Preprocessing: We discard sequences with poor reconstruction statistics or high overlap with OSM building footprints. We subsample the sequences such that frames are spaced by at least 4 meters. We undistorted fisheye images into pinhole cameras. We resampled each 360 panorama into 4 90°-FOV perspective views at equally-distributed yaw angles with a random offset constant per sequence. We query OSM data for each city and create tiles of our raster representation at a resolution Δ=50 cm.

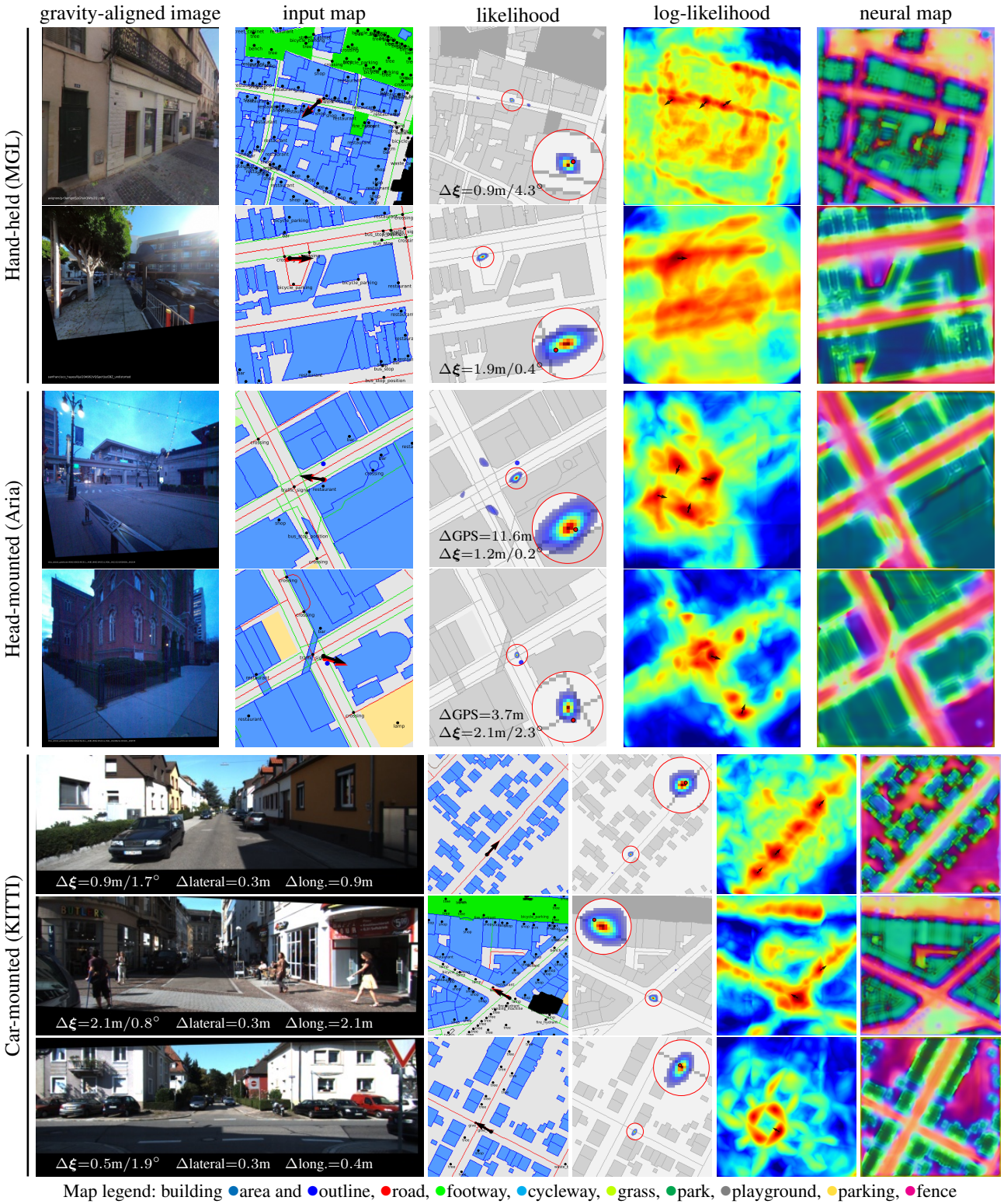

Figure 3. **Additional qualitative results for single-image localization.** We again show the ground truth pose (green arrow) and the predicted pose (black arrow). For Aria data, we also show the noisy GPS measurement as a red dot.

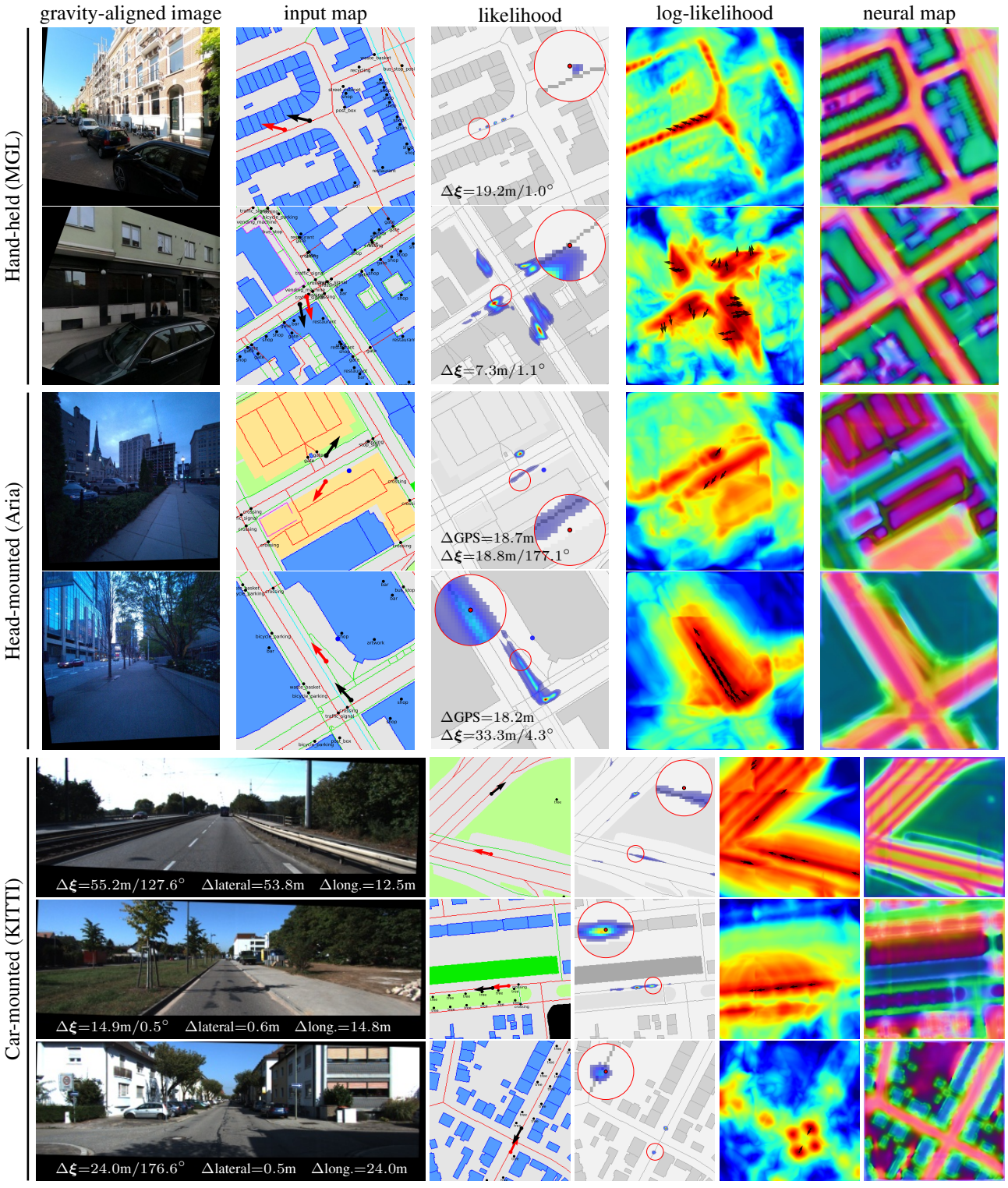

Figure 4. **Failure cases of single-image localization.** Localizing a single image often fails when the environment lacks distinctive elements, when they do not appear in the map, or when such elements are repeated, making the pose ambiguous. Since OSM is crowd-sourced, the level of detail of the map is not consistent and widely varies. For example, trees are registered in some cities but not in others.

Country	City	# sequences	# images
USA	San Francisco	1013	207.6k
Netherlands	Amsterdam	57	72.9k
Germany	Berlin	59	54.6k
Lithuania	Vilnius	381	111.5k
Finland	Helsinki	91	55.4k
Italy	Milan	156	46.2k
France	Paris	136	68.4k
	Montrouge	159	33.3k
	Le Mans	111	27.4k
	Nantes	171	62.5k
	Avignon	160	75.2k
	Toulouse	86	39.6k

Table 2. **Distribution of locations** from which we built the MGL dataset. We selected cities that are well covered by both Mapillary and OpenStreetMap.

B.3. Aria datasets

Recording: We recorded data with Aria devices [1] at 3 locations in Seattle (Downtown, Pike Place Market, Westlake) and at 2 locations in Detroit (Greektown, Grand Circus Park). In each location, we recorded 3 to 5 sequences following the same trajectories, for a duration of 5 to 25 minutes varying by location. Each device is equipped with a consumer-grade GPS sensor, IMUs, grayscale SLAM cameras, and a front-facing RGB camera, which we undistort to a pinhole model.

Evaluation: We associate GPS signals captured at 1Hz with undistorted 640×640 RGB images keyframed at 3 meters. This resulted in 2153 frames for Seattle and 2725 frames for Detroit. For each evaluation example, the map tile is centered around the noisy GPS measurement. Because of large differences in GPS accuracy due to urban canyons, we constrain the predictions within 64 m of the measurement for Seattle and 24 m for Detroit.

Comparison to feature matching: Algorithms based on 3D SfM maps require mapping images, whose quality and density have a large impact on the localization accuracy. Differently, OrienterNet can localize in areas not covered by such images as long as OSM data is available. This makes any fair comparison difficult.

Geo-alignment: Evaluating the localization within world-aligned maps requires the geo-alignement between each device pose and the global reference frame. We first co-register all trajectories of each location by minimizing visual-inertial SLAM constraints, which yields consistent poses in a local reference frame that is gravity-aligned and at metric scale. We then find the global 3-DoF rigid transformation by fusing all GPS signals with the predictions of OrienterNet for each image. To do, we perform iterative truncated least-squares, annealing the outlier threshold from 5m to 1m.

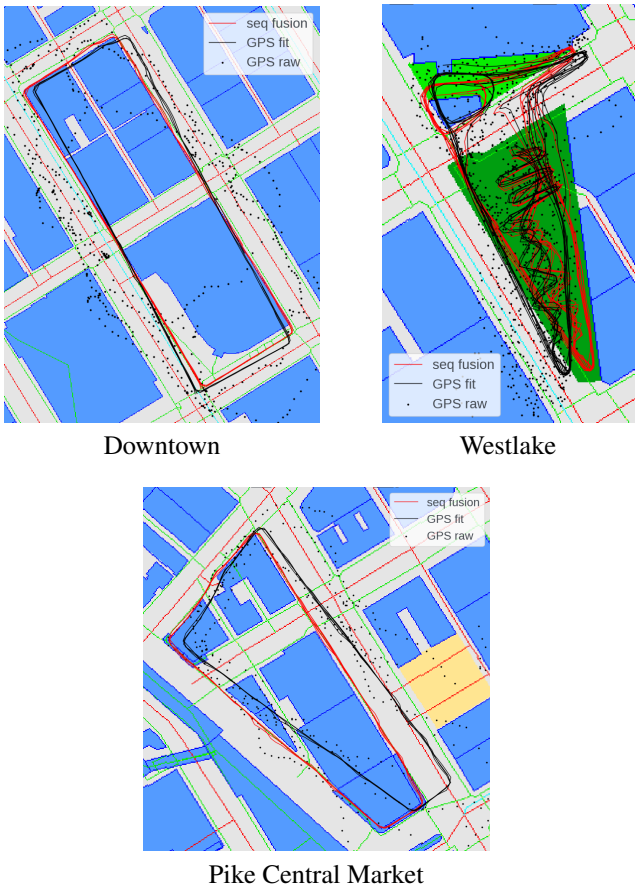

Figure 5. **Pseudo-ground truth alignment of Aria sequences** for the 3 locations in Seattle. Fusing GPS signals and OrienterNet predictions across all images of all sequences is more robust than relying on GPS alone.

We show visualizations of this alignment in Fig. 5. While GPS signals are too noisy to reliably fit a transformation, OrienterNet provides complementary and accurate local constraints. We visually check that the final alignment error is lower than 1m, which is sufficient for our evaluation.

C. Implementation details

Orienter-Net: To save GPU memory, we use only $K{=}64$ rotation bins at training time but increase it to $K{=}512$ at test time. BEV and map features have $N{=}8$ channels. To avoid overfitting, we found it critical to use replicate padding in the map-CNN Φ_{map} and to apply data augmentation to the raster map by randomly flipping and rotating it. We also use replicate padding in the BEV-to-map matching operation to avoid biasing the predictions near the map boundaries. The scale boundaries σ_{min} and σ_{max} are set to 2^1 and 2^9, respectively. For the median focal length $f = 256$px of the MGL training set, this corresponds to a depth interval of $[0.5\,\text{m}, 128\,\text{m}]$.

Training images are resized to 512×512 pixels. When evaluating on KITTI and Aria data, images are resized such that their focal length is $f = 256$px. We train with a batch size of 9 over 3 V100 GPUs with 16GB VRAM each. We select the best model checkpoint with early stopping based on the validation loss.

Retrieval baseline: The work of Samano *et al*. [3] infers a global descriptor for each map patch. This is inefficient when considering densely sampled areas. We can equivalently predict a dense feature map $\mathbf{F}$ in one CNN forward pass, which is similar to the recent work of Xia *et al*. [5] for satellite imagery. We then correlate the global image descriptor with $\mathbf{F}$ to obtain the pixelwise log-score $\mathbf{M}$. To predict a rotation, Φ_{map} computes 4 feature maps $\mathbf{F}_{\text{N}}, \mathbf{F}_{\text{S}}, \mathbf{F}_{\text{E}}, \mathbf{F}_{\text{W}}$ for the 4 N-S-E-W directions, from which we can linearly interpolate map features for any number of rotation bins This yields a 3D $W\times H\times K$ pose volume $\mathbf{M}$ as for OrienterNet. We found this approach much more efficient than re-computing map features for different map orientations.

Refinement baseline: We follow the official implementation of Shi *et al*. [4] with multi-level optimization at each iteration and warm restart. We supervise the longitudinal, lateral, and angular offsets with an L2 loss at each iteration and scale. The map data and branch are identical to OrienterNet.

References

[1] Project Aria. https://about.facebook.com/realitylabs/projectaria/, 2022. 4

[2] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017. 1

[3] Noe Samano, Mengjie Zhou, and Andrew Calway. You are here: Geolocation by embedding maps and images. In *ECCV*, 2020. 5

[4] Yujiao Shi and Hongdong Li. Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In *CVPR*, 2022. 5

[5] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Visual cross-view metric localization with dense uncertainty estimates. In *ECCV*, 2022. 5

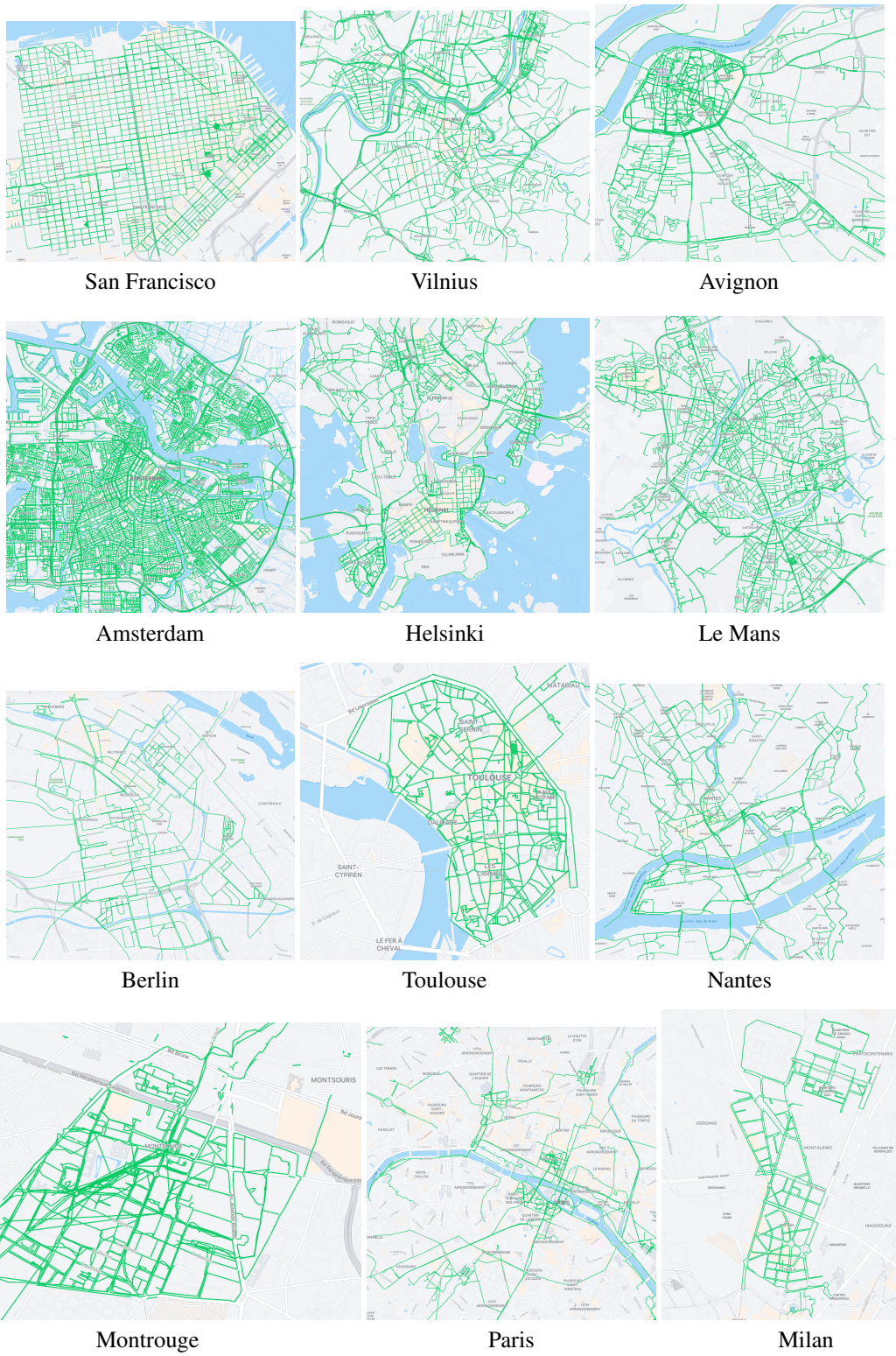

Figure 6. **Selected sequences of our MGL dataset** across 12 cities. Screenshots taken from the Mapillary platform browser.

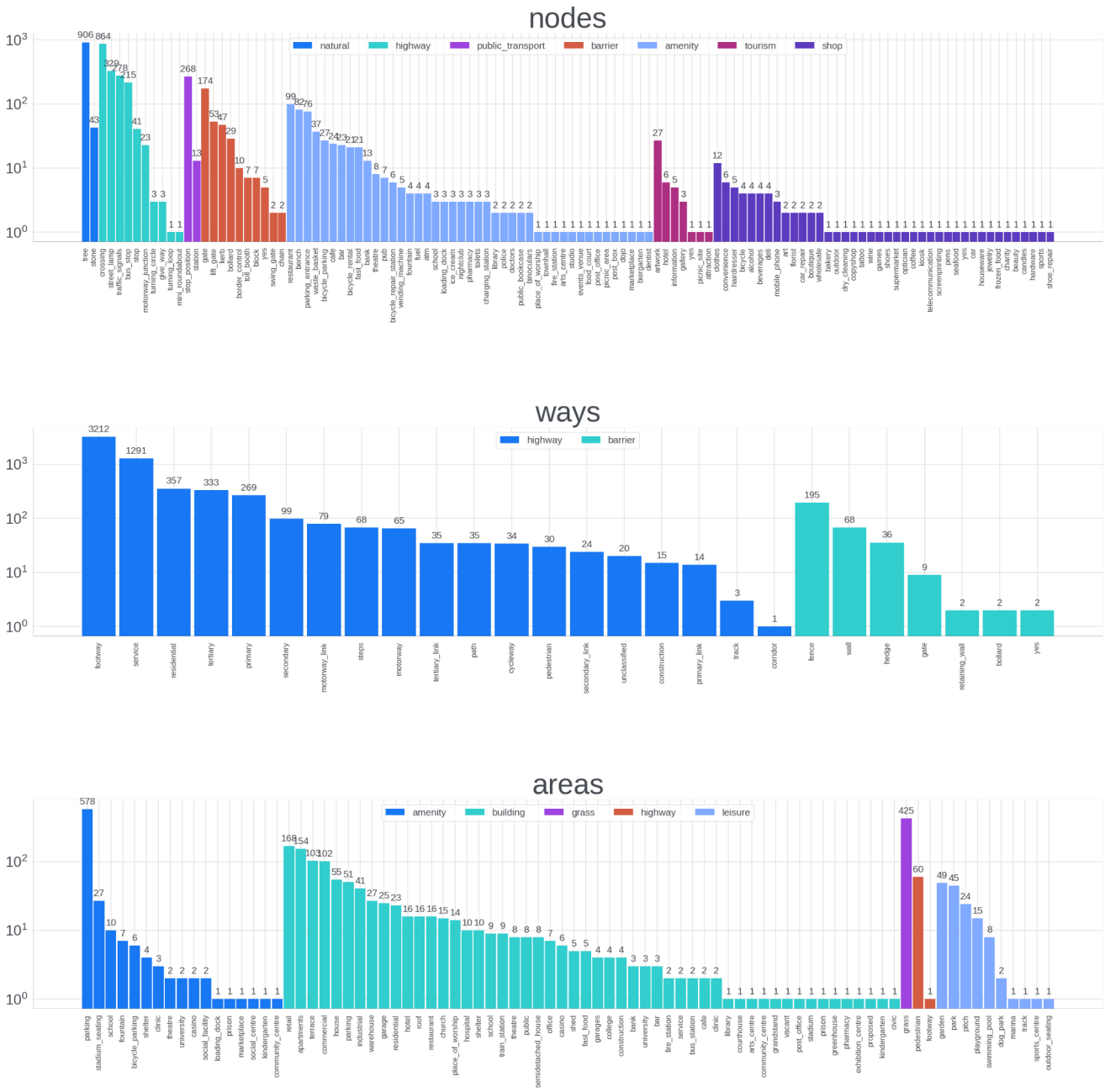

Figure 7. **Distribution of OSM elements** for the city of Detroit: points (top), lines (middle), polygons (bottom). We group them by label and order them by frequency.

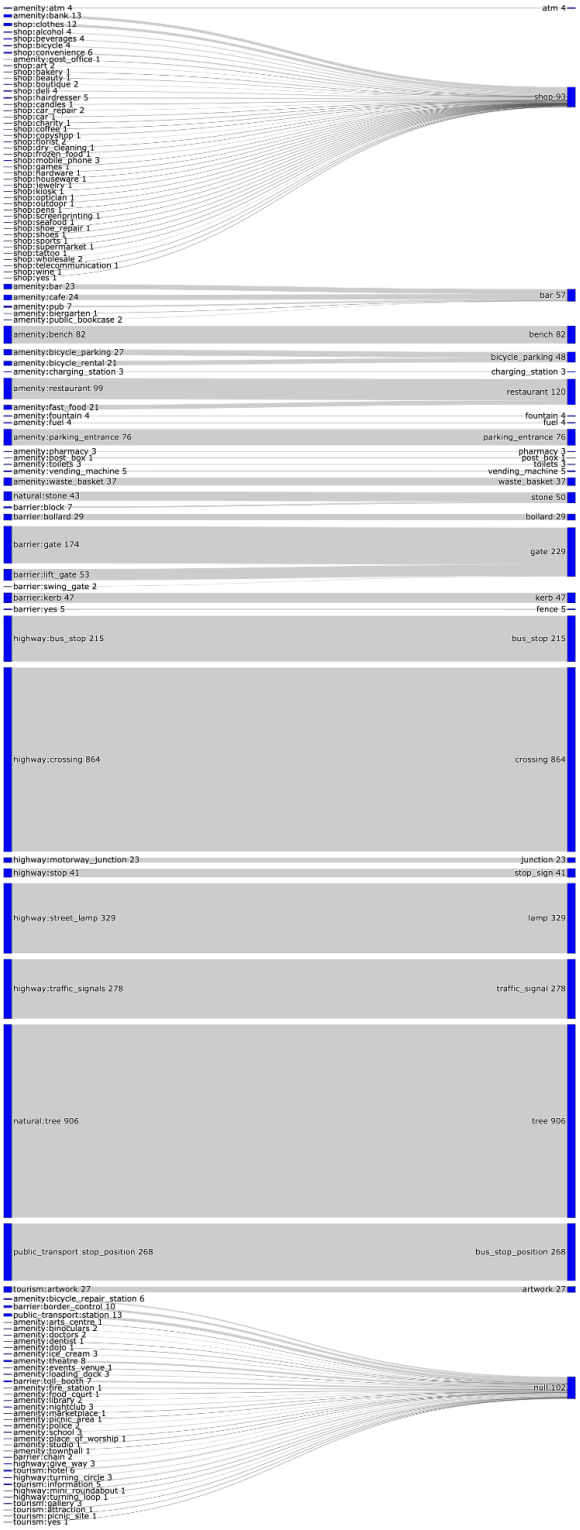

Figure 8. **Mapping between OSM tags and raster classes**. We show a subset of the classes and the number of corresponding elements for the city of Detroit.