

Automatic map inference, data refinement, and machine-assisted map editing promises more accurate map datasets.

BY FAVYEN BASTANI, SONGTAO HE, SATVAT JAGWANI, EDWARD PARK, SOFIA ABBAR, MOHAMMAD ALIZADEH, HARI BALAKRISHNAN, SANJAY CHAWLA, SAM MADDEN, AND MOHAMMAD AMIN SADEGHI

Inferring and Improving Street Maps with Data-Driven Automation

STREET MAPS HELP to inform a wide range of decisions. Drivers, cyclists, and pedestrians use them for search and navigation. Rescue workers responding to disasters such as hurricanes, tsunamis, and earthquakes rely on street maps to understand where people are and to locate individual buildings.²³

Transportation researchers consult street maps to conduct transportation studies, such as analyzing pedestrian accessibility to public transport.²⁵ Indeed, with the need for accurate street maps growing in importance, companies are spending hundreds of millions of dollars to map roads globally.^a

However, street maps are incomplete or lag behind new construction in many parts of the world. In rural Indonesia, for example, entire groups of

» key insights

- Digital street maps cost hundreds of millions of dollars annually to create and maintain, but updates lag months behind new road and building construction.
- Satellite imagery and GPS data captured by road users are promising data sources for automating the map update process. Machine learning can infer road and building positions from these sources.
- To update maps with minimal error and cost, we built Mapster, which features a machine-assisted map editing framework and end-to-end machine-learning methods that directly infer vector road networks from raw data sources.

^a For example: Korosec, K. Uber will spend \$500 million on mapping to diverge from Google. *Fortune* (May 7, 2016).

villages are missing from OpenStreetMap, a popular open map dataset.³ In many of these villages, the closest mapped road is miles away. In Qatar, construction of new infrastructure has boomed in preparation for the FIFA World Cup 2022. But the rapid pace of construction means that it often takes a year for digital maps to reflect new roads and buildings.^b Even in countries such as the U.S., where significant investment has been made in digital maps, construction and road closures often take days or weeks to be incorporated into map datasets.

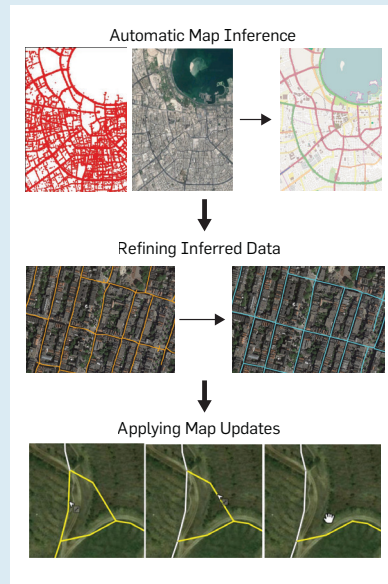
These problems arise because today's processes for creating and maintaining street maps are extremely labor-intensive. Modern street map editing tools allow users to trace and annotate roads and buildings on a map canvas overlaid on relevant data sources, so that users can effectively edit the map while consulting the data. These data sources include satellite imagery, aerial imagery, and GPS trajectories (which consist of sequences of GPS positions captured from moving vehicles). Although the data presented by these tools helps users to update a map dataset, the manual tracing and annotation process is cumbersome and a major bottleneck in map maintenance.

Over the past decade, many *automatic map inference* systems have been proposed to automatically extract information from these data sources at scale. Several approaches develop unsupervised clustering and density-thresholding algorithms to construct road networks from GPS trajectory datasets.^{1,4,7,16,21} Others apply machine-learning methods to process satellite imagery and extract road networks,^{19,22} building polygons,^{13,24} and road attribute annotations—for example, the number of lanes, presence of cycling lanes, or presence of street parking on a road.^{6,18}

However, automatic map inference has failed to gain traction in practice due to two key limitations. First, existing inference methods have high error rates (low precision), which manifest in noisy outputs containing incorrect

Figure 1. An overview of the Mapster map editing system.

First, we infer the road network from satellite imagery and GPS data. Then, we transform the map to make it look more realistic. Finally, we have an interactive system to apply map updates.



roads, buildings, and attribute annotations. Second, although prior work has shown how to detect roads and buildings from data sources, the challenge of leveraging this information to update real street map datasets has not been addressed. We argue that even with lower error rates, the quality of outputs from automatic approaches is below that of manually curated street map datasets, and semi-automation is needed to efficiently leverage automatic map inference to accelerate the map maintenance process.

At Massachusetts Institute of Technology (MIT) and Qatar Computer Research Institute (QCRI), we have developed several algorithms and approaches to address these challenges,^{2,3,14} which we combined into a new system called *Mapster*. Mapster is a human-in-the-loop street map editing system that incorporates three components to accelerate the mapping process over traditional tools and workflows: high-precision automatic map inference, data refinement, and machine-assisted map editing.

First, Mapster applies automatic map inference algorithms to extract initial estimates from raw data sources. Although these estimates are noisy,

we minimize errors by applying two novel approaches that replace heuristic, error-prone post-processing steps present in prior work with end-to-end machine learning and other more robust methods: *iterative tracing for road network inference* and *graph network annotation for attribute inference*.

Second, Mapster refines the noisy estimates from map inference into map update proposals by removing several types of errors and reducing noise. To do so, we apply conditional generative adversarial networks (cGANs) trained to transform the noisy estimates into refined outputs that are more consistent with human-annotated data.

Finally, a machine-assisted map editing framework enables the rapid, semi-automated incorporation of these proposed updates into the street map dataset. This editing tool addresses the problem of leveraging inferred roads, buildings, and attribute annotations to update existing street map datasets.

Figure 1 summarizes the interactions between these components. We included links to two videos demonstrating the execution of Mapster, along with a link to Mapster's source code (which we have released as free software).^c

In this article, we first introduce our automatic map inference approaches for inferring roads and road attributes, which achieve substantially higher precision than prior work. Then, we detail our data refinement strategy, which applies adversarial learning to improve the quality of inferred road networks. Finally, we discuss our machine-assisted map editor, which incorporates novel techniques to maximize user productivity in updating street maps. We conclude with a discussion of future work.

Automatically Inferring Road Networks

Given a base road network, which may be empty or may correspond to the roads in the current street map dataset, the goal of road network inference is to leverage GPS trajectory data and satellite imagery to produce a road network

^b An example of a subdivision in Doha, Qatar that was missing from maps for years is detailed at <https://productforums.google.com/forum/#!topic/maps/dwtCso9owlU>.

^c Iterative tracing in action: https://youtu.be/3_AE2Qn-Rdg. Machine-assisted map editing: <https://youtu.be/i-6nbuuX6NY>. Mapster source code: <https://github.com/mitroad-maps>.

map that covers roads not contained in the base map. The road network map is represented as a graph, where vertices are annotated with spatial longitude-latitude coordinates and edges correspond to straight-line road segments.

Broadly, prior approaches infer roads by dividing the space into a 2D grid, classifying whether each grid cell contains a road, and connecting cells together to form edges. Figure 2(a) summarizes this strategy. For satellite imagery, recent work obtains the per-cell classification by applying deep convolutional neural networks (CNNs) that segment the imagery, transforming the input imagery into a single-channel image that indicates the neural network's confidence that there is a road at each pixel.^{9,10,17} For GPS trajectories, several approaches perform the classification based on the number of GPS trajectories passing through each cell.^{5,8,11,20}

However, we find that these methods exhibit low accuracy in practice when faced with challenges such as noisy data and complex road topology. Figure 2(b) shows the output of prior work around a major highway junction in Chicago. Noise in the per-cell classification estimates is amplified when we connect cells together to draw edges, resulting in garbled road network maps.

Indeed, both GPS trajectory data and satellite imagery exhibit several types of noise that make robust identification of roads challenging. While GPS samples in open areas are typically accurate to four meters, in practice, due to high-rise buildings and reflection, GPS readings may be as far off as tens of meters. Correcting this error is difficult because errors are often spatially correlated—multiple GPS readings at the same location may be offset in the same way as they encounter the

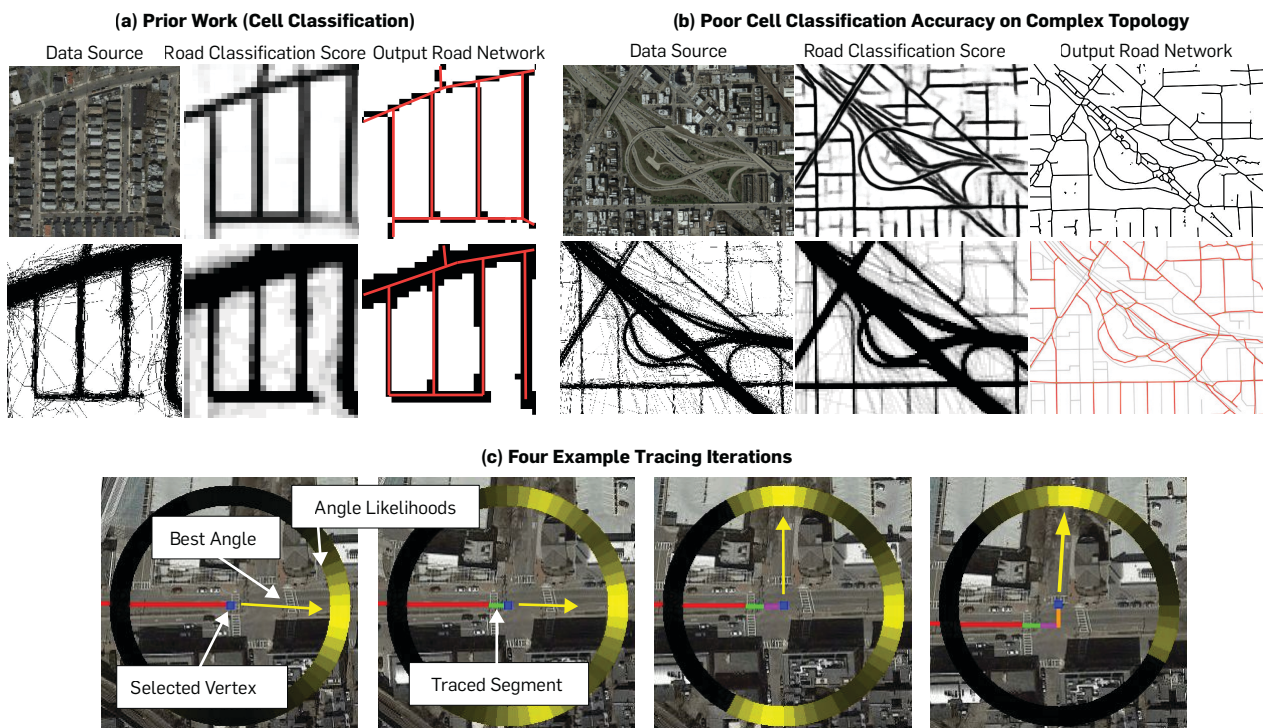
same reflection and distortion issues. Similarly, roads in satellite imagery are frequently occluded by trees, buildings, and shadows. Furthermore, distinguishing roads and buildings from non-road trails and surface structures in imagery is often nontrivial.

To substantially improve precision, we adopt an iterative road-tracing approach in lieu of the per-cell classification strategy. Our iterative tracing method mimics the gradual tracing process that human map editors use to create road network maps, thereby eliminating the need for the heuristic post-processing steps that prior work applies to draw edges based on cell classification outputs.

Iterative tracing begins with the base map, and on each iteration, it adds a single road segment (one edge) to the map. To decide where to position this segment, it uses the data

Figure 2. Old and new mapping approaches.

(a) Prior automatic mapping approaches operating on satellite imagery (top) and GPS trajectories (bottom). In the center, we show the per-cell classification scores, and on the right, the result of thresholding and cell connection. (b) These approaches produce noisy outputs around complex road topology such as highway interchanges. (c) Iterative tracing on satellite imagery. A CNN predicts the likelihood that there is a road at each of 64 angles from a vertex; higher likelihoods from the blue vertex are shown in yellow in the outer circle, and lower likelihoods are shown in black. Iterative tracing gradually expands the coverage of the map: on each iteration, it adds a segment corresponding to the highest likelihood vertex and direction.



source to compute two values for each vertex in the portion of the map traced so far: (a) a confidence that an unmapped road intersects the vertex, and (b) the most likely angular direction of the unmapped road. Then, it selects the vertex with the highest confidence and adds a segment in the corresponding direction. This procedure is repeated, adding one segment

at a time to the map, until the highest confidence for the presence of an unmapped road falls below a threshold. We illustrate the iterative tracing procedure in Figure 2(c).

We develop different approaches to compute the unmapped road confidence and direction from satellite imagery³ and from GPS trajectories.¹⁴ With satellite imagery, we compute

these through a deep neural network. We develop a CNN model that inputs a window of satellite imagery around a vertex of the road network, along with an additional channel containing a representation of the road network that has been traced so far. We train the CNN to output the likelihood that an unmapped road intersects the vertex, and the most likely angular direction of this unmapped road.

With GPS trajectories, we compute the values at a vertex based on the peak direction of trajectories that pass through the vertex. We identify all trajectories that pass through the vertex and construct a smoothed polar histogram over the directions followed by those trajectories after they move away from the vertex. Then, we apply a peak-finding algorithm to identify peaks in the histogram that have not been explored earlier in the tracing process. We select the peak direction and measure confidence in terms of the volume of trajectories that match the peak direction.

Often, both satellite imagery and GPS trajectory data may be available in a region requiring road network inference. To reduce errors and improve the map quality, we develop a two-stage approach that leverages both data sources when inferring road networks. In the first stage, we run iterative tracing using GPS data to infer segments along high-throughput roadways. Because these roads have high traffic volume, they are covered by large numbers of GPS trajectories, so they can be accurately traced with GPS data. At the same time, junctions along high-throughput roads (especially controlled-access highways) are generally more complex, often involving roundabouts and overpasses. These features make tracing based on satellite imagery challenging.

In the second stage, we fill in gaps in the road network with lower-throughput residential and service roads that were missed in the first stage by tracing with satellite imagery. These roads have simple topologies and are covered by few GPS trajectories, making imagery a preferred data source.

We evaluate our method by comparing road networks inferred through iterative tracing against those inferred by prior cell-classification approaches. Figure 3 shows qualitative results in Boston, Chicago, Salt Lake City, and

Figure 3. Qualitative results comparing iterative tracing to prior cell classification approaches shows (a) road networks inferred from satellite imagery, and (b) road networks inferred from GPS trajectories. We show inferred roads in the foreground and OpenStreetMap data in the background.



Los Angeles. We use 60-cm/pixel satellite imagery from the United States Geological Survey (USGS) and Google Earth, which is available in urban areas across the world, and GPS trajectory data captured at 1 Hz from private cars. In contrast to cell classification, iterative tracing from satellite imagery robustly infers roads despite occlusion by buildings and shadows in dense urban areas. In lower-density areas such as Salt Lake City, iterative tracing performs comparably to prior work.

Cell classification from GPS trajectories produces noisy outputs at crucial but complex map features, such as highway interchanges. Despite the intricate connections at these features, iterative tracing accurately maps the interchanges.

We show quantitative results.^{3,14} The execution time of iterative tracing is practical. With satellite imagery, the asymptotic complexity is $O(l)$, where l is the total length of the inferred roads. For example, inferring roads in a 15-km² urban area requires 16 minutes on an NVIDIA Tesla V100 GPU with iterative tracing; although slower than cell classification (which completes in two minutes), it is nevertheless efficient even for large road networks, and tracing can be parallelized across multiple machines. With GPS trajectories, the asymptotic complexity is $O(l \cdot d)$, where l is the total length of the inferred roads and d is the average number of GPS points explored at each iteration. For example, iterative tracing takes 110 minutes to process 2.5 million GPS points in a 16-km² area on an AWS C5.9xlarge instance with 15% CPU utilization.

Inferring Road Attributes

Modern navigation systems use more than just the road topology—they also use a number of road metadata attributes, such as the number of lanes, presence of cycling lanes, or the presence of street parking along a road. As a result, inferring these attributes is an important part of the Mapster system.

Prior work in road-attribute inference applies an image-classification approach that trains a CNN to predict the road attributes given a window of satellite imagery around some location along the road. Then, the local prediction at each location is post-processed through



Mapster is a street map editing system that accelerates the mapping process over traditional tools and workflows with high-precision automatic map inference, data refinement, and machine-assisted map editing.



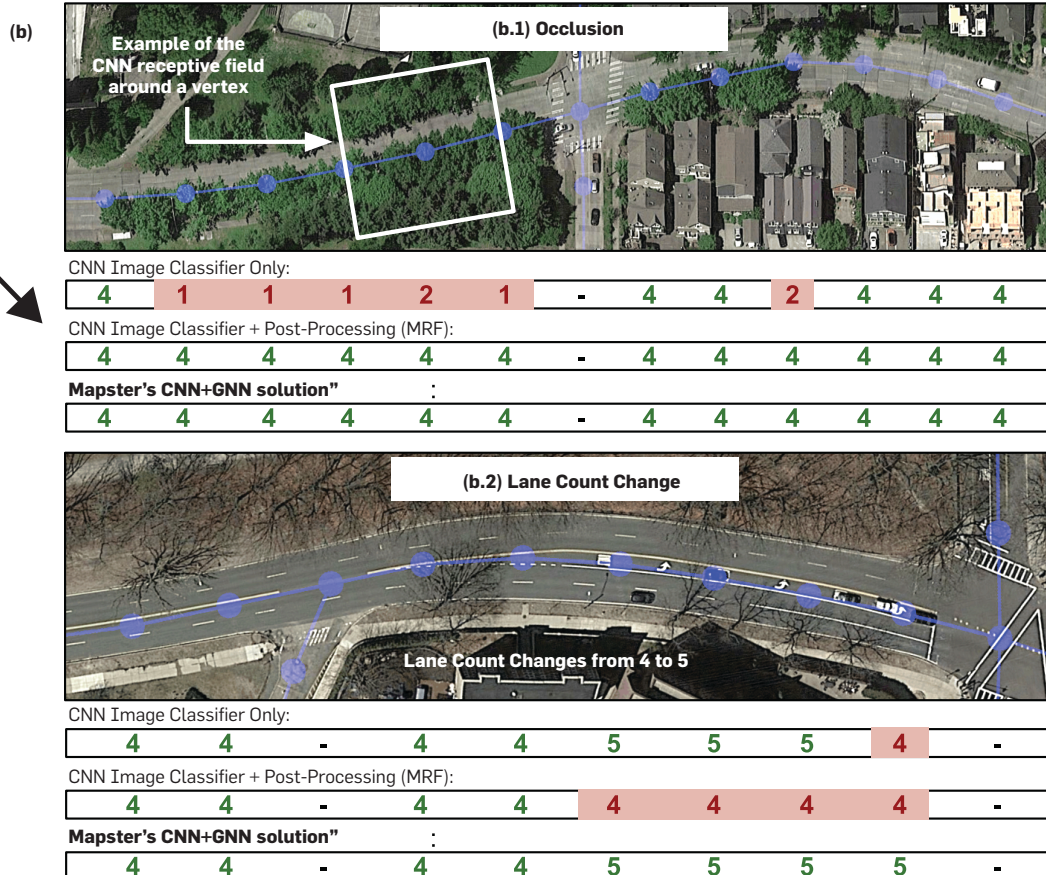
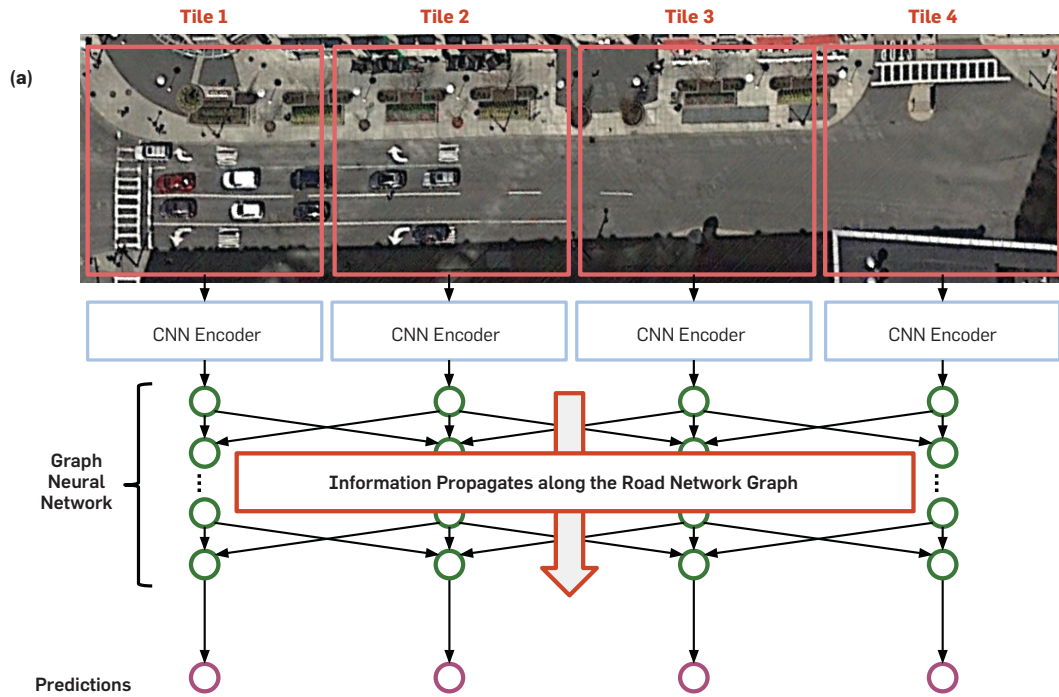
a global inference phase to remove scattered errors—for example, inference in a Markov Random Field (MRF).

This global inference phase is necessary because the CNN prediction at each location is often erroneous due to the *limited receptive field* of the CNN—in many cases, local information in the input window of satellite imagery is not sufficient to make a correct prediction. For example, in Figure 4(b.1), the road on the left side is occluded by trees. If the CNN inputs a window only from the left part of the road, it will be unable to correctly determine the number of lanes. The global inference phase corrects these errors in the CNN outputs by accounting for all the predictions along the road, as well as prior knowledge—for example, road attributes such as the number of lanes should generally be homogeneous along the road instead of varying frequently.

However, we find that global inference postprocessing is often error prone. For example, consider Figure 4(b.2), where the lane count changes from four to five near an intersection. The image classifier outputs partially incorrect labels. The global inference phase fails to correct this error; because it only accounts for predictions from the image classifier, it cannot determine whether the number of lanes indeed changes or is simply an error of the image classifier. This limitation is caused by the *information barrier* induced by the separation of local classification and global inference; the global inference phase can only use the image classifier's prediction as input, but not other important information, such as whether trees occlude the road or whether the road width changes.

To break the information barrier, we develop a hybrid neural network architecture, as in Figure 4(a), which combines a CNN with a graph neural network (GNN). The CNN extracts local features from each segment along a road. Then, the GNN propagates these features along the road network. End-to-end training of the combined CNN and GNN model is key to the success of the method: rather than rely on hand-crafted, error-prone, post-processing heuristics that only operate over CNN predictions, our method instead *learns*

Figure 4. (a) The hybrid neural-network architecture proposed in this work. (b) Examples on inferring the number of lanes. In each image, blue lines show the road network. The number of lanes predicted by the CNN Image Classifier and Mapster on each segment are shown along the bottom of each figure. Output numbers in green are correct predictions while red numbers are incorrect predictions.



the post-processing rules as well as the required CNN features directly from the data. The use of the GNN in our system eliminates the *receptive field limitation* of local CNN image classifiers, and the combination of CNN and GNN eliminates the *information barrier* present in prior work. This allows the model to learn complex inductive rules that make it more robust in the face of challenges, such as occlusion in satellite imagery and the partial disappearance of important information, as seen in Figure 4(b).

Refining Inferred Data

Although iterative tracing improves substantially over prior grid-cell classification approaches, it nevertheless creates road networks with noisy features that clearly distinguish them from human-drawn maps. Iterative tracing is particularly effective at capturing road network topology but leaves geometrical abnormalities, such as the examples of off-center junction vertices and noisy road curvature in Figure 5(a).

To refine the map and rectify these abnormalities, we use a model—based on cGANs¹²—to learn the realistic road appearance. These networks learn to realistically reproduce complex, image-to-image transformations and have been successfully applied to many tasks, including adding color to black-and-white images and transforming photos taken in daylight to plausible nighttime photos of the same scene.¹⁵

An obvious transformation to learn for map inference is transforming satellite imagery or representations of GPS trajectories into road networks, where the output image contains lines corresponding to road segments. However, we find that the learning problem is too difficult under this strategy, and the cGAN model fails to learn to robustly identify roads. Instead, it primarily learns to produce arbitrary lines that resemble lines in the ground-truth road-network representation.

Thus, our cGAN model instead inputs not only satellite imagery or GPS trajectory data, but also a representation of the road network produced by iterative tracing. It outputs a refined road network representation where abnormalities in the input network

have been corrected. By providing this initial road network, we reduce the complexity of the transformation and thereby assist the cGAN to learn the transformation, especially early in the training process. Incorporating the initial road network representation derived from iterative tracing into the cGAN was a crucial insight that made training the adversarial model feasible.

Our cGAN architecture consists of two components: a *generator* that produces refined road networks given an initial road network and a data source, and a *discriminator* that learns to dis-

tinguish refinements made by the generator from road networks in the ground truth dataset. This network is adversarial because we train the generator and discriminator with opposing loss functions: the discriminator minimizes its classification error at distinguishing ground truth and generated (refined) road networks. In contrast, the generator learns by maximizing the discriminator's classification error. Thus, in effect, we train the generator by having it learn to fool the discriminator into classifying its generated road network as ground truth.

Figure 5. (a) Road networks before refinement, in orange, contain geometric abnormalities. The refined road networks, shown in blue, clean up these noisy features. (b) Our human-in-the-loop map-editing system employs a pruning algorithm to eliminate residential and service roads and focus the user on adding major roads to the map. Pruned roads are shown in purple and the remaining roads in yellow.



The initial road network provided to the generator enables the cGAN model to learn to produce realistic road networks. At first, the generator simply copies the road network from its input to its output to deceive the discriminator. However, once the discriminator learns to better distinguish the iterative-tracing road networks from hand-drawn, ground truth roads, the generator begins making small adjustments to the roads so that they appear to be hand-drawn. As training continues, these adjustments become more robust and complete.

Figure 5(a) shows the outputs of our cGAN in blue, both on the geometrical abnormalities introduced earlier and on a larger region in Minneapolis, MN. Again, we use 60-cm/pixel satellite imagery for this evaluation. While refinement does not substantially alter the topology of the road network, the cGAN improves the geometry so that inferred roads resemble hand-mapped roads. These geometry improvements help to reduce the work needed to integrate inferred data into the street map.

Machine-Assisted Map Editing

To improve street map datasets, the inferred road network derived from iterative tracing and refinement steps must be incorporated into the existing road network map. Fully automated integration of the inferred road network is impractical: inferred roads may include errors even after refinement, so adding all the inferred roads to the map dataset would degrade the its precision.

Instead, we developed a human-in-the-loop map-editing framework that enables human map editors to quickly validate automatically inferred data.² On initialization, our machine-assisted map editor builds an overlay containing the inferred road segments. Users interact with the overlay by left- and right-clicking to mark segments as correct or incorrect. Thus, rather than trace roads through a series of repeated clicks along the road, when a correctly inferred segment already covers the road, users of the machine-assisted editor can rapidly add the road to the map with a single click on the inferred segment in the overlay.

Our map editor has two additional



Modern navigation systems use more than just the road topology—they use road metadata attributes, such as the number of lanes, presence of cycling lanes, or the presence of street parking along a road. Inferring these attributes is an important part of Mapster.



features to improve validation speed. First, the interface includes a “prune” button that executes a shortest-path-based pruning algorithm to eliminate minor residential and service roads from the overlay, leaving only major arterial roads. This functionality is especially useful when mapping areas where the existing road network in the street map dataset has low coverage. In these areas, adding every missing road to the map may require substantial effort, but the quality of the map could be improved significantly just by incorporating major roads. The pruning algorithm is effective at helping users focus on mapping these unmapped major roads by reducing information overload. We show an example pruning result in Figure 5(b). Purple segments are pruned, leaving the yellow segments that correspond to major inferred roads.

Pruning is most useful in low-coverage areas, but for high-coverage areas, we developed a teleport button that pans users to a connected component of inferred roads. In high-coverage areas, only a small number of roads are missing from the map, and identifying an unmapped road requires users to painstakingly scan the imagery one tile at a time. The teleport button eliminates this need, allowing users to jump to a group of missing roads and immediately begin mapping them.

Future Work

Although Mapster greatly reduces map creation workload and maintenance, automatically inferred street maps still have more errors than maps created by professional mapmakers. Narrowing this gap requires advances in machine-learning approaches for automatic map inference. Next, we detail several promising avenues for improving inference performance.

First, better neural-network architectures can improve the performance of automatic mapping. So far, the design of the neural-network architectures in Mapster are mostly inspired by best practices in general computer vision tasks, such as image segmentation and image classification. However, map inference tasks have unique characteristics, such as the strong spatial correlation in satellite imagery. Thus, improved neural-network architectures that are specialized

for mapmaking tasks may yield higher accuracy. For example, instead of using raw images or GPS traces as input, node and edge embeddings garnered from unsupervised tasks on extremely large map datasets could be used.

Second, end-to-end loss functions are a promising avenue to directly learn desired properties of the output road network. However, these desired properties, such as high precision and high recall over roads, generally can only be computed from the road network graph. As a result, the objectives are nondifferentiable since the graph can only be extracted from the probabilistic output of a machine-learning model through non-differentiable functions. Thus, both prior work and our iterative-tracing approach learn to build a graph indirectly: prior work minimizes the per-cell classification error, and in iterative tracing on satellite imagery, we minimize the difference between the predicted angle of an untraced road and the correct angle on each tracing step. Nevertheless, end-to-end training has been shown to improve performance in other machine-learning tasks and could improve the accuracy of automatic map creation and maintenance.

Several techniques have been proposed for optimizing non-differentiable metrics. For example, reinforcement-learning techniques can search for optimal policies under non-differentiable rewards. Alternatively, it may be possible to train an additional neural network, which takes the intermediate map representation—for example, road segmentation—as input and predicts the value of evaluation metrics. This additional neural network acts as a differentiable approximation of the evaluation metrics and can be leveraged to train a model to infer road networks that score highly on the metric.

In addition to potential improvements in machine-learning techniques, incorporating new data sources would also extend the capability of Mapster. Two particularly promising data sources are drone imagery and dashboard-camera video. Drone imagery enables on-demand image sensing—for instance, if we find the road structure in a region is unclear from satellite imagery and GPS data, we can reactively assign drones to collect aerial images over that region. Video from

dashboard cameras enables inferring several additional street map features, such as street names, business names, road signs, and road markers.

Conclusion

The world has approximately 64 million kilometers of roads and the road network is growing at a rapid pace as major countries such as China, India, Indonesia, and Qatar gather economic momentum. Street maps are important. However, creating and maintaining street maps is very expensive and involves labor-intensive processes. As a result, although a lot of effort and money has been spent in maintaining street maps, today's street maps are still imperfect and are frequently either incomplete or lag behind new construction. Mapster is a holistic approach for applying automation to reduce the work needed to maintain street maps. By incorporating automatic map inference with data refinement and a machine-assisted map editor, Mapster makes automation practical for map editing, and enables the curation of map datasets that are more complete and up to date at less cost. 

References

- Ahmed, M., Karagiorgou, S., Pfoser, D., and Wenk, C. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica* 19, 3 (2015), 601–632.
- Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., and Madden, S. Machine-assisted map editing. In *Proceedings of the 26th ACM SIGSPATIAL Intern. Conf. on Advances in Geographic Information Systems* (2018), 23–32.
- Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., and DeWitt, D. RoadTracer: Automatic extraction of road networks from aerial images. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition* (2018), 4720–4728.
- Biagioni, J. and Eriksson, J. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *J. of the Transportation Research Board* 2291, 1 (2012).
- Biagioni, J. and Eriksson, J. Map inference in the face of noise and disparity. In *Proceedings of the 20th Intern. Conf. on Advances in Geographic Information Systems*, ACM, (2012), 79–88.
- Cadamuro, G., Muhebwa, A., and Taneja, J. Assigning a grade: Accurate measurement of road quality using satellite measurements. In *Proceedings of NIPS Workshop on Machine Learning for the Developing World* (2018).
- Cao, L. and Krumm, J. From GPS traces to a routable road map. In *ACM SIGSPATIAL* (2009), 3–12.
- Chen, C. and Cheng, Y. Roads digital map generation with multi-track GPS data. In *Proceedings of the 2008 Intern. Workshop on Geoscience and Remote Sensing*.
- Cheng, G., Wang, Y., Xu, S., Wang, H., Xiang, S., and Pan, C. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Trans. on Geoscience and Remote Sensing* 55, 6 (2017), 3322–3337.
- Costea, D. and Leordeanu, M. Aerial image geolocalization from recognition and matching of roads and intersections. In *Proceedings of the British Machine Vision Conf.* (2016).
- Davies, J.J., Beresford, A.R., and Hopper, A. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing* 5, 4 (2006), 47–54.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems* (2014), 2672–2680.
- Hamaguchi, R. and Hikosaka, S. Building detection from satellite imagery using ensemble of size-specific detectors. In *The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2018).
- He, S., Bastani, F., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., and Madden, S. RoadRunner: Improving the precision of road network inference from GPS trajectories. In *Proceedings of the 26th ACM SIGSPATIAL Intern. Conf. on Advances in Geographic Information Systems* (2018).
- Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A.A. Image-to-image translation with conditional adversarial nets (2016).
- Liu, X., Biagioni, J., Eriksson, J., Wang, Y., Forman, G., and Zhu, Y. Mining large-scale, sparse GPS traces for map inference: Comparison of approaches. In *Proceedings of the 18th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining* (2012).
- Mattyus, G., Luo, W., and Urtasun, R. DeepRoadMapper: Extracting road topology from aerial images. In *Proceedings of the IEEE Intern. Conf. on Computer Vision* (2017), 3438–3446.
- Mattyus, G., Wang, S., Fidler, S., and Urtasun, R. Enhancing road maps by parsing aerial images around the world. In *Proceedings of the IEEE Intern. Conf. on Computer Vision* (2015), 1689–1697.
- Mnih, V. and Hinton, G.E. Learning to detect roads in high-resolution aerial images. In *European Conf. on Computer Vision*, Springer (2010), 210–223.
- Shi, W., Shen, S., and Lu, Y. Automatic generation of road network map from massive GPS, vehicle trajectories. In *Proceedings of the 12th Intern. IEEE Conf. on Intelligent Transportation Systems* (2009).
- Stanojevic, R., Abbar, S., Thirumuruganathan, S., Chawla, S., Filali, F., and Aleimat, A. Robust road map inference through network alignment of trajectories. In *Proceedings of the 2018 SIAM Intern. Conf. on Data Mining* (2018).
- Wegner, J.D., Montoya-Zegarria, J.A., and Schindler, K. Road networks as collections of minimum cost paths. *ISPRS J. of Photogrammetry and Remote Sensing* 108 (2015), 128–137.
- Yin, A. A mapathon to pinpoint areas hardest hit in Puerto Rico. *The New York Times* (October 2017).
- Zhao, K., Kang, J., Jung, J., and Sohn, G. Building extraction from satellite images using mask r-cnn with building boundary regularization. In *The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2018).
- Zielstra, D. and Hochmair, H.H. Comparative study of pedestrian accessibility to transit stations using free and proprietary network data. *J. of the Transportation Research Board* 2217, 1 (2011), 145–152.

Favyen Bastani is a Ph.D. student at MIT Computer Science and Artificial Intelligence Lab (CSAIL), CSAIL, Cambridge, MA, USA.

Songtao He is a Ph.D. student at MIT CSAIL, CSAIL, Cambridge, MA, USA.

Satvat Jagwani is a master's student at MIT CSAIL, CSAIL, Cambridge, MA, USA.

Edward Park is a master's student at MIT CSAIL, CSAIL, Cambridge, MA, USA.

Sofiane Abbar is a machine-learning software engineer at Facebook, U.K.

Mohammad Alizadeh is an associate professor of Electrical Engineering and Computer Science at MIT CSAIL, CSAIL, Cambridge, MA, USA.

Hari Balakrishnan is a professor of Electrical Engineering and Computer Science at MIT CSAIL, CSAIL, Cambridge, MA, USA.

Sanjay Chawla is the research director of the Data Analytics department at Qatar Computer Research Institute (QCRI), Doha, Qatar.

Sam Madden is a professor of Electrical Engineering and Computer Science at MIT CSAIL, CSAIL, Cambridge, MA, USA.

Mohammad Amin Sadeghi is a research scientist at QCRI, Doha, Qatar.