Inferring Road Maps from Global Positioning System Traces

Survey and Comparative Evaluation

James Biagioni and Jakob Eriksson

As a result of the availability of Global Positioning System (GPS) sensors in a variety of everyday devices, GPS trace data are becoming increasingly abundant. One potential use of this wealth of data is to infer and update the geometry and connectivity of road maps through the use of what are known as map generation or map inference algorithms. These algorithms offer a tremendous advantage when no existing road map data are present. Instead of the expense of a complete road survey, GPS trace data can be used to generate entirely new sections of the road map at a fraction of the cost. In cases of existing maps, road map inference may not only help to increase the accuracy of available road maps but may also help to detect new road construction and to make dynamic adaptions to road closures—useful features for in-car navigation with digital road maps. In past research, proposed algorithms had been evaluated qualitatively with little or no comparison with prior work. This lack of quantitative and comparative evaluation is addressed in this paper with the following contributions: (*a*) a comprehensive survey of the current literature on map generation; (*b*) a description of the first method for the automatic evaluation of generated maps; (*c*) a qualitative, quantitative, and comparative evaluation of three reference algorithms; and (*d*) an open source implementation of each of the three algorithms, with a 118-h trace data set and ground truth map for unrestricted use by the automatic map generation community.

Road map inference is the process of automatically producing a directed, annotated road map from GPS traces. GPS traces are typically collected opportunistically, from vehicles that are driving the roads for some other purpose. Map inference can be used to rapidly map unknown or constantly changing territory, to update and improve on existing road maps, or to quickly adapt to detours and new construction.

Common throughout the existing literature on automatic map generation is a focus on qualitative evaluation. Virtually every published paper on the topic relies on a visual inspection of the results and manually compares generated maps with existing maps or satellite imagery. Moreover, comparisons with existing work are virtually absent: out of 11 papers surveyed, only one provided any comparison with prior work (*1*). The focus on qualitative evaluation in the literature, and the lack of comparative evaluation, has made it difficult to understand the relative merits of the various proposed map inference methods.

The lack of quantitative, comparative evaluations of map inference algorithms is attributable to three missing ingredients: (*a*) sufficiently expressive and robust methods of quantitatively evaluating the accuracy of a generated map, (*b*) publicly available implementations of proposed map inference methods, and (*c*) common sets of publicly available GPS traces and ground truth maps for use in evaluations.

In this paper, the three problems identified earlier are thoroughly addressed. The research (*a*) describes the first quantitative evaluation method for map inference algorithms, (*b*) provides open source implementations of three reference algorithms, and (*c*) makes available a 118-h trace data set, plus ground truth maps, for unrestricted use by the map inference community. The data and implementations are available on the BITS laboratory website (*2*). To provide a better understanding of prior work on map inference, the following are also provided: (*a*) a comprehensive survey of the current literature on automatic map generation and (*b*) a qualitative, quantitative, and comparative evaluation of the three reference algorithm implementations.

The remainder of the paper is structured as follows: (*a*) a survey of the literature on the topic of automatic map generation, which briefly describes the various methods proposed and the evaluation performed in each paper; (*b*) a description of the proposed quantitative evaluation method; (*c*) a visual illustration of the performance of the algorithms and qualitative observations; (*d*) an analysis of the quantitative measures of the performance and a brief discussion of the implementations of three reference algorithms (*3–5*); and (*e*) conclusions.

BACKGROUND AND SURVEY OF EXISTING LITERATURE

The baseline requirement of a map inference algorithm is to automatically turn raw GPS traces into a directed and annotated graph that represents the connectivity and geometry of the underlying road network. Beyond such basic map generation, various additional objectives have been proposed, such as the extraction of detailed intersection geometries (*6*), the number and centerlines of lanes (*6*), and the speed limit and road type (*7*). This paper focuses primarily on basic map generation and only briefly mentions these other aspects.

Operational Overview

The map generation process is typically preceded by a filtering step, in which traces are checked for any irregularities with regard to the expected distance between points, the speed traveled, the acceleration,

Department of Computer Science, University of Illinois at Chicago, SEO 1120 M/C 152, 851 South Morgan Street, Chicago, IL 60607. Corresponding author: J. Biagioni, jbiagi1@uic.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2291, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 61–71.
DOI: 10.3141/2291-08

and any abrupt direction changes. Any point along a GPS trace that fails to satisfy these criteria is removed, and an interpolated point is inserted in its place.

After filtering, the approaches described in the literature can be categorized by their algorithmic foundations: the *k*-means algorithm (*8*), trace merging, or kernel density estimation (KDE) (*9*). Members of the first category perform variations on a three-dimensional (latitude, longitude, direction) *k*-means algorithm to reduce the set of GPS points to a smaller set of cluster centroids. The centroids are then linked together to form the road geometry. Trace-merging methods merge edges directly, without first reducing the set of raw GPS data. Finally, KDE-based methods first produce a KD estimate of the raw GPS points or edges, then co-opt image processing techniques to extract the road geometry from this density estimate.

Survey of Existing Literature

Table 1 is a summary of the literature on automatic map generation. Out of the 11 papers, six report *k*-means based methods, two report trace-merging methods, and three report KDE methods. As noted in the evaluation method column, the literature thus far has almost exclusively relied on a qualitative evaluation method (eyeballing); only four papers quantitatively evaluated the accuracy of the generated maps, and none made quantitative comparisons with existing work. Typically, a sample of the generated road geometry is overlaid onto a map or satellite image (ground truth), from which conclusions are drawn manually. Despite the widespread availability of ground truth maps, quantitative evaluation has been exceedingly rare in this line of work. In cases in which any type of numerical accuracy result has been reported, no comparison has been made with related work (*3, 6, 7, 11*). Similarly, in cases in which a comparison has been made with related work, only a qualitative comparison was offered, with no numbers reported (*1*).

In the following section, the three categories of map inference algorithms are introduced in more detail and the variations on each offered in the literature are briefly discussed.

Map Inference Based on *k*-Means Algorithm

The *k*-means based approach, originally described in Edelkamp and Schrödl (*3*), and illustrated in Figure 1, begins by distributing a series of cluster seeds at locations drawn from the set of trace data, with the constraint that every trace point must be within a fixed distance d and bearing difference δ of a cluster seed.

The cluster seeds are used as initial guesses; minor variations on the *k*-means algorithm are then used to find the seed locations and headings that best describe the raw traces. Once the seed locations are identified, seeds are linked to form road segments, based on the pattern of raw traces passing through the seed locations. Qualitative and quantitative evaluations of the basic *k*-means algorithm are provided later in the current paper.

Edelkamp and Schrödl, 2003 (3)

In addition to being the original *k*-means based method, the 2003 paper by Edelkamp and Schrödl further refines the road network model by fine-tuning the location of intersections, representing the road centerline by a fitted spline rather than a series of straight lines, and performing lane finding. Lane finding is done by clustering raw traces based on their respective distance offset from the road centerline.

This work was evaluated by comparing the number of lanes detected and the lane position error with a base map generated by the authors, through the use of survey-grade differential GPS equipment.

Schroedl et al., 2004 (6)

Based on Edelkamp and Schrödl (*3*), the 2004 paper by Schroedl et al. describes a process for additionally refining the intersection geometry and modeling individual lanes and the transitions and turn restrictions between them. This process is accomplished by first identifying intersections and their bounding boxes. Traces through an intersection are then grouped by entry and exit points, and a

TABLE 1 Chronological Summary of Papers on Map Inference

Paper	Class	Data	Ground Truth	Evaluation Method	Features
Edelkamp and Schrödl (*3*)	*k*-means	250 synthetically perturbed traces	Generated from GPS traces	Lane error vs. amount of data	Lane finding
Schroedl et al. (*6*)	*k*-means	250 synthetically perturbed traces	Generated from GPS traces	Lane error vs. amount of data	Intersection geometry
Davies et al. (*4*)	KDE	1 million GPS points	UK ordnance survey	Eyeball vs. ground truth	na
Worrall and Nebot (*10*)	*k*-means	Traces from mining vehicles	None	Compact vs. raw	Compact representation
Guo et al. (*11*)	*k*-means	Synthetic GPS traces	None	Relative error vs. amount of data	na
Chen and Cheng (*12*)	KDE	Traces from automobiles	Google Earth	Eyeball vs. ground truth	na
Niehoefer et al. (*7*)	Trace merge	7 traces	Google Maps	Eyeball vs. ground truth, relative error vs. amount of data	Edge classification
Cao and Krumm (*5*)	Trace merge	20 million GPS points from campus shuttles	Bing Maps	Eyeball vs. ground truth, route query vs. Bing Maps	GPS trace clarification
Shi et al. (*13*)	KDE	Massive amounts of GPS traces	Google Earth	Eyeball vs. ground truth	na
Jang et al. (*14*)	*k*-means	GPS traces	Naver maps (*15*)	Eyeball vs. ground truth	na
Agamennoni et al. (*1*)	*k*-means	5 days or 15 open mine vehicle GPS traces	None	Eyeball vs. Davies et al. (*4*) and Schroedl et al. (*6*)	Principal road path

NOTE: na = not applicable.

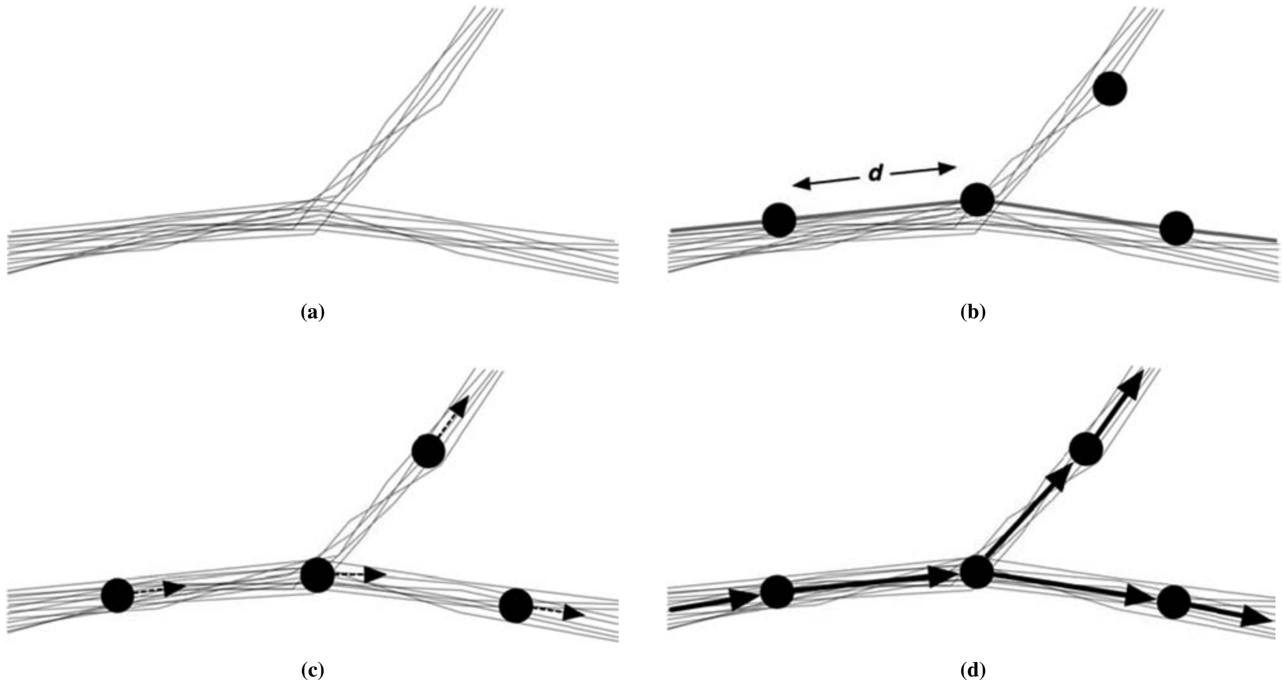

FIGURE 1 Map inference with *k*-means algorithm: (*a*) raw traces, (*b*) initial seeds, (*c*) updated means with bearing, and (*d*) final graph after means are linked.

spline-fitting technique (*16*) is used for each group to produce the final turn-lane geometry of the intersection.

This work was evaluated by visually comparing generated intersection geometries against Navteq maps (*17*). A heuristic performance optimization for this method was described by Jang et al. and evaluated by visually comparing the generated map with the map provided by the South Korean web portal Naver (*14, 15*).

Worrall and Nebot, 2007 (10)

In Worrall and Nebot, a method based on Edelkamp and Schrödl (*3*) is presented to infer a road network represented as a compact set of lines and arcs. To create a compact representation of the generated road map, the set of clusters is first segmented into regions of constant curvature. The best line or arc is then fitted to each segment. Standard regression analysis is used for regions that represent straight lines; regions that represent arcs are fitted using nonlinear least-squares fitting (*18*).

This work was evaluated by measuring the differences between the compressed map and the originally generated map. A similar approach is presented in Guo et al. (*11*).

Agamennoni et al., 2011 (1)

The approach taken by Agamennoni et al. (*1*) is similar to Schroedl et al. (*6*), but the focus is to extract "principal road paths," which are curves as defined in Hastie and Stuetzle (*19*).

In Agamennoni et al. (*20*), a visual comparison is made against the original *k*-means approach (*6*), as well as one based on KDE (*4*). A limited quantitative evaluation of this method is also presented in Agamennoni et al., in which a GPS trace data set collected by the authors is used (*20*).

Map Inference Based on Trace Merging

The trace-merging methods proposed thus far use a greedy approach, illustrated in Figure 2. Through iterations of each recorded GPS trace, edges from the raw trace are added to the map unless an edge sufficiently similar in location and bearing already exists. Should such an edge already exist, its weight is instead incremented. In postprocessing, any edges with weights below a certain threshold are removed.

Cao and Krumm, 2009 (5)

The method proposed by Cao and Krumm (*5*) precedes the standard trace-merging method with a clarification step. The "clarification step" is a type of particle simulation, in which a strong but short-range attractive force is applied to pull together nearby traces, and a weaker but long-range retractive force is used to keep traces from straying too far from their original location (*21*). This reduces the effects of GPS noise by pulling together traces that originate from the same road, thereby forming tight bands along the road centerlines.

This work was evaluated by visually comparing the generated map with satellite imagery from Bing Maps, and also by visually comparing shortest path routes from the generated map with those from Bing Maps. Qualitative and quantitative evaluations of this trace-merging algorithm are provided later in the current paper.

Niehoefer et al., 2009 (7)

The method described by Niehoefer et al. modifies the standard merging approach by adjusting the position of existing road segments when merging a new trace segment into the existing map (*7*). This technique allows the location of road segments in the base map

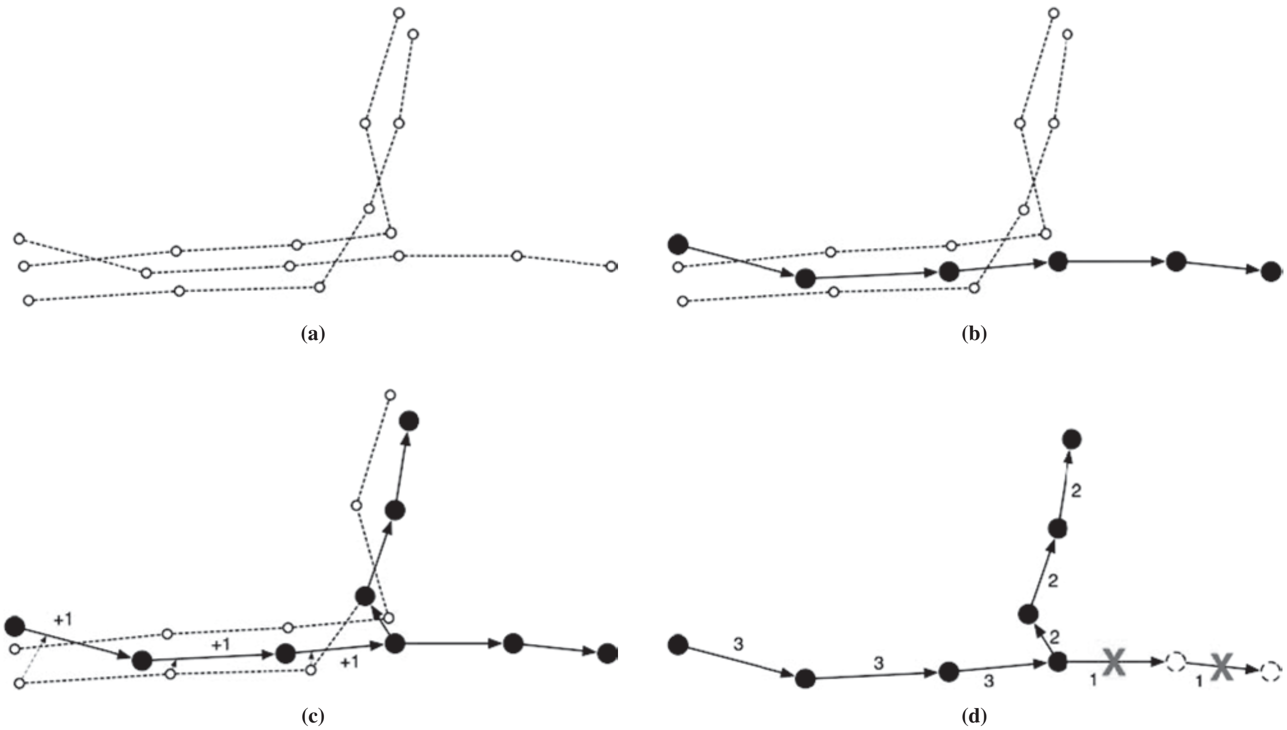

FIGURE 2 Trace-merging approach: (*a*) raw GPS traces, (*b*) new edges added, (*c*) weights of matching edges incremented, and (*d*) edges with insufficient weight removed.

to be steadily refined as more traces are added, in a similar spirit to the clarification preprocessing step used in Cao and Krumm (*5*), except here it is used during the merge procedure. The paper also describes means of automatically classifying road types, such as highways, streets, or walkways, as well as entry and exit ramps, bridges, and tunnels.

A qualitative evaluation of this work was conducted by visually comparing the generated map with the same region depicted in Google Maps. A quantitative evaluation was also performed against a reference map that was generated using all available traces, whereby the relative position error of a road segment was shown to rapidly decrease with increasing amounts of data.

Map Inference Based on KDE

Map inference methods based on KDE (*a*) compute an approximate KD estimate of trace points or edges over the area of interest (*9*), (*b*) apply a threshold to produce a binary image of the roads, and (*c*) apply a variety of methods to produce road centerlines from this binary image, as illustrated in Figure 3.

To produce the KD estimate, the geographical area of interest is divided into a two-dimensional grid of cells whose width is a fraction of the road width. Through iterations over each of the raw GPS points or edges, a two-dimensional histogram is produced that contains the number of points or edges that fall into each grid cell. The histogram is then convolved with a Gaussian smoothing function (*22*), producing an approximation of the desired KD estimate. This is an approximation because of the discretization created by the use of grid cells, and the accuracy of the approximation is inversely proportional to the grid size.

Davies et al., 2006 (4)

In the scheme proposed by Davies et al., edges are accounted for in each grid cell that they pass through by an amount proportional to the length of the line between the two points that passes through the cell in the antialiasing process (in computer graphics, "antialiasing" is the process of removing or reducing the jagged distortions in curves and diagonal lines so that the lines become more smooth) (*4, 23*).

After thresholding the density estimate (in computer graphics, "thresholding" is the process of converting an image into a binary representation whereby pixels are marked as "object" pixels if their value is greater than a certain threshold and as "background" pixels if their value is lower than the threshold), the outlines of the produced road bitmap image are extracted using a contour follower (*24*). To find the centerlines of these outlines, which are likely to coincide with the centerlines of the underlying roads, the Voronoi graph of points, evenly spaced along the contours, is produced (*25*), followed by the removal of edges that fall outside the contour or that are of insufficient length. Finally, separately produced KDEs of traces in each of the eight cardinal and ordinal directions are used to annotate each road segment with its permitted directions of travel.

This work was evaluated by visually comparing the generated map against that of a UK Ordnance Survey. Qualitative and quantitative evaluations of this KDE-based algorithm are provided later in the paper.

Chen and Cheng, 2008 (12)

The KDE-based method described by Chen and Cheng produces a point-based density estimate and takes an image-processing approach to extract the road map from the bitmap image produced after the thresholding step (*12*). Morphological dilation and closing operations

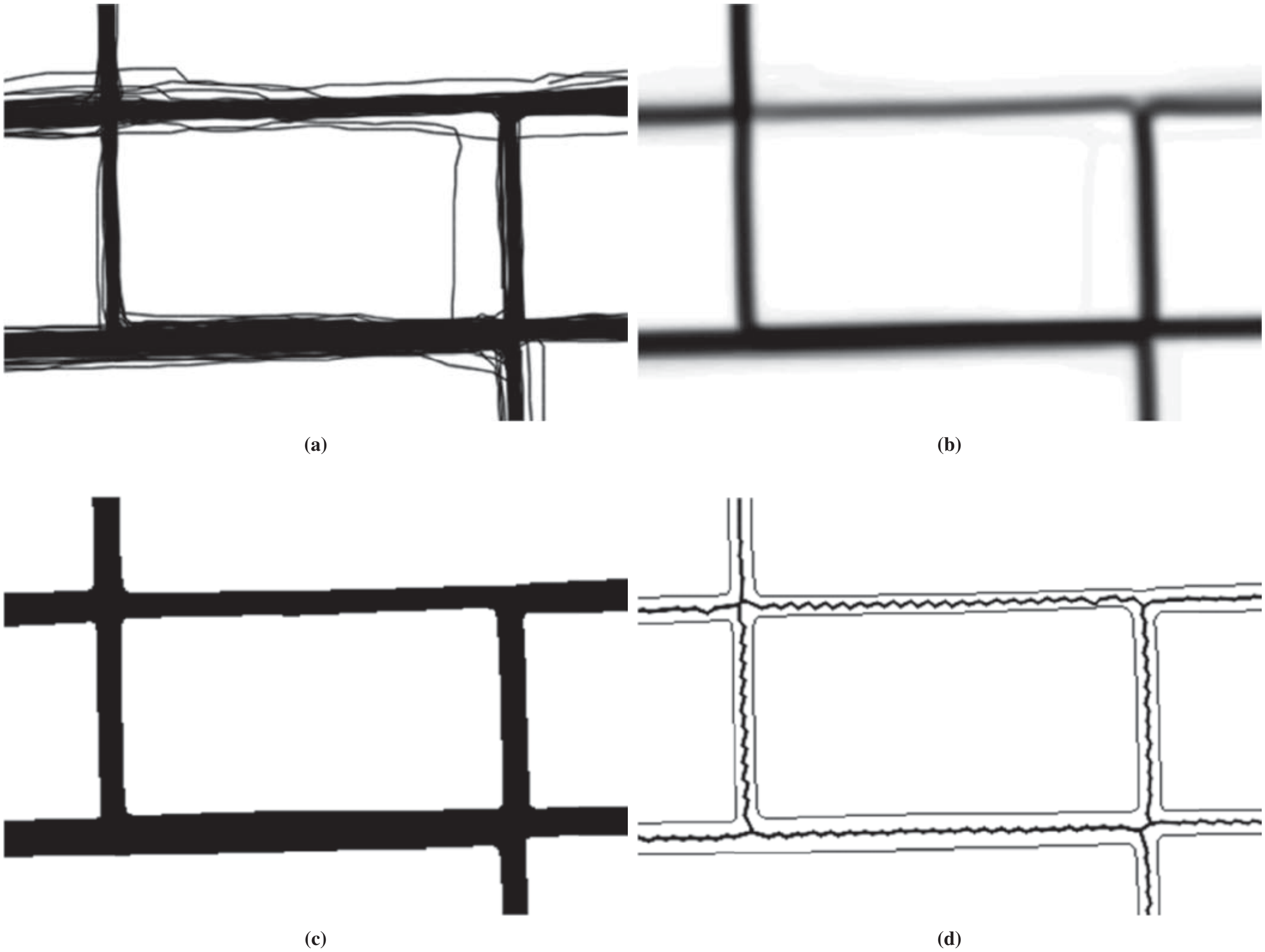

FIGURE 3 Map inference based on KDE (*4*): (*a*) raw traces, (*b*) KDE computed, (*c*) threshold applied and road or nonroad cells identified, and (*d*) road outlines and centerlines computed.

are used to produce a smooth and contiguous image of the road boundaries. Then, a thinning operation deletes all pixels on the boundaries of the pattern until only a skeleton remains along the road centerlines, which is then converted into road segments. A very similar method was proposed by Shi et al. (*13*). Both methods were evaluated by visually comparing the generated maps with images from Google Earth.

ROBUST QUANTITATIVE EVALUATION OF GENERATED MAPS

In this section, a method of evaluating the accuracy of a map, with respect to a ground truth map, is described. This is a necessary requirement for a quantitative, comparative evaluation of competing map inference algorithms.

The accuracy of a generated map depends on two primary aspects of the map: geometry and topology. The geometry of the map describes the geographical location of the roads, and the topology describes the interconnections between the roads.

A large body of work looks at the problem of comparing graphs (*26*). This work includes exact methods for testing for graph isomorphism (*27*), and inexact methods for measuring graphs' editing distance (*28*). However, the problem of measuring the similarity of graphs is fundamentally different from that of measuring the similarity of maps: crucially, graphs and their nodes and edges lack any notion of geographical location. Thus, although graph similarity algorithms are able to measure the degree of topological similarity, the graphs' nodes and edges can be freely transposed to find the closest match. In a map, however, the geographical location of the nodes and edges must be taken into account, in addition to their topological relationships.

To simultaneously measure the geometric and topological similarity of maps, a method is proposed based on the following idea: starting from a random street location, explore each map outward within a maximum radius. This produces two sets of locations, which are essentially spatial samples of a local map neighborhood. Through a comparison of the two sets of samples and repeated sampling of the maps, in this fashion, a robust measure of the difference between the two maps is obtained. If one of the maps is the ground truth, this difference represents the accuracy of the other map.

The operation of the map comparison algorithm presented in this paper is depicted in Figure 4. First, a start location is selected at random from the ground truth map. This point is marked with an X

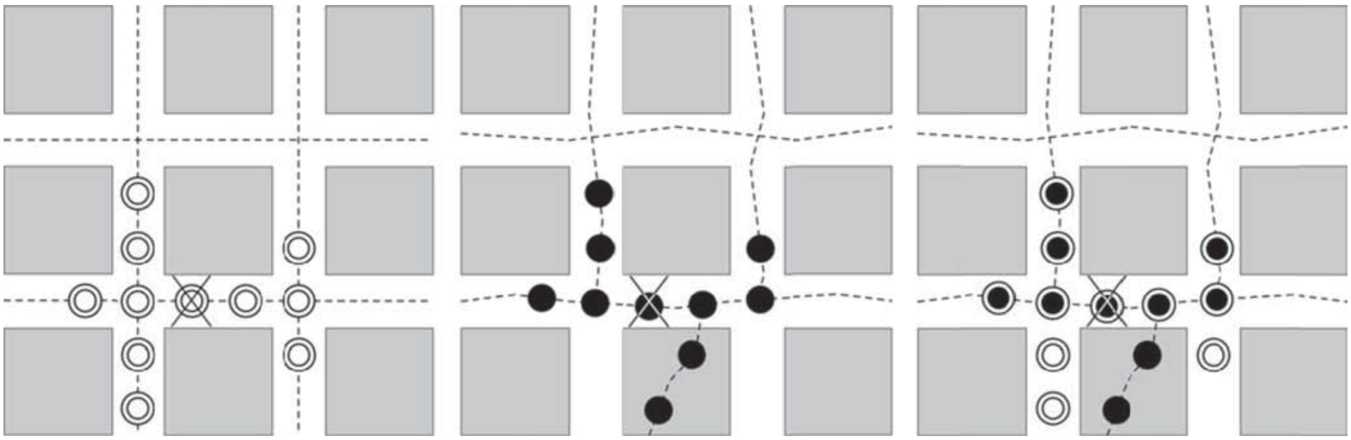

FIGURE 4 Overview of map comparison algorithm: (*a*) holes dropped at fixed intervals (ground truth map), (*b*) marbles dropped at fixed intervals (generated map), and (*c*) marbles fill holes where maps overlap.

in Figure 4*a.* From the start location, all road segments within a small matching distance d are followed and virtual holes are dropped at fixed intervals until a maximum radius r from the start location is reached or a previously followed segment is encountered. When an intersection is encountered, all connecting road segments that lead away from the start location are followed, as turn restrictions and one-way streets allow. Restricting the process to segments leading away from the start location elegantly de-emphasizes unlikely driving patterns, such as U-turns, and improves the robustness of the map comparison operation. This process is then repeated, starting from the road segments in the generated map within matching distance d of the start location. Through implementation of the same procedure in Figure 4*b,* virtual marbles are dropped at fixed intervals out to a radius r.

Intuitively, if a marble lands close to a hole, it falls in. This represents the matching process. As illustrated in Figure 4*c,* marbles that are too far from a hole remain where they land, and holes with no marbles nearby remain empty. In the figure, holes that are filled correspond to matched locations, at which the geometry and topology of the two maps overlap. Unmatched marbles correspond to spurious parts of the generated map, and holes that remain empty correspond to parts of the ground truth map that are missing in the generated map. Counting the number of unmatched marbles and empty holes quantifies the accuracy of the generated map with respect to the ground truth according to two metrics: (*a*) the proportion of spurious marbles and (*b*) the proportion of missing locations (empty holes):

$$\text{spurious} = \frac{\text{spurious\_marbles}}{\text{spurious\_marbles} + \text{matched\_marbles}}$$

$$\text{missing} = \frac{\text{empty\_holes}}{\text{empty\_holes} + \text{matched\_holes}}$$

To produce a combined performance measure from these two values, the well-known F-score is used (*29*), which is computed as follows:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \cdot \frac{(1 - \text{spurious})(1 - \text{missing})}{(1 - \text{spurious}) + (1 - \text{missing})}$$

The higher the F-score, the closer the match. Sampling the maps locally is an important aspect of this approach as it provides the ability to capture the connectivity of the maps at a very detailed level (i.e., as they would be traveled), allowing the topological similarity to be measured. Conversely, if a global approach to this problem were taken, in which every edge in the ground truth map was simply covered with holes, and every edge in the generated map with marbles, yielding just two sets to be matched, the process would capture the geometric similarity between maps; however, it would fail to capture local topological similarity, a crucial aspect of overall map similarity. Repeated local sampling at randomly chosen locations yields an accurate view of local geometry and topology throughout the map.

CONSTRUCTION OF GROUND TRUTH

To measure the accuracy of the generated map, an accurate ground truth map is needed for comparison. The ground truth map was based on the OpenStreetMap database (*30*). However, these map data contained many road segments that were not visited by the vehicles in the study data set. Therefore, the ground truth map was restricted to those street segments that were actually traversed by a vehicle in the study data set. This reduced ground truth map reflected the most accurate road topology that could be inferred from the available traces.

QUALITATIVE EVALUATION

In this section, the road maps generated by three representative algorithms, one from each of the three classes described in the literature review, are qualitatively evaluated. To make this comparison, the algorithms were first implemented as described in their respective papers. For k-means, the algorithm by Edelkamp and Schrödl (“the Edelkamp algorithm”) is used (*3*). For trace merging, the algorithm by Cao and Krumm (“the Cao algorithm”) is used (*5*), and for KDE-based map inference, the algorithm by Davies et al. (“the Davies algorithm”) is used (*4*).

Evaluation Data

For trace data, 118 h of GPS traces from the University of Illinois at Chicago campus shuttles were used. In addition to traveling around campus, these shuttles pass through two different areas that contain relatively tall buildings and significant GPS error. Figure 5*a* provides

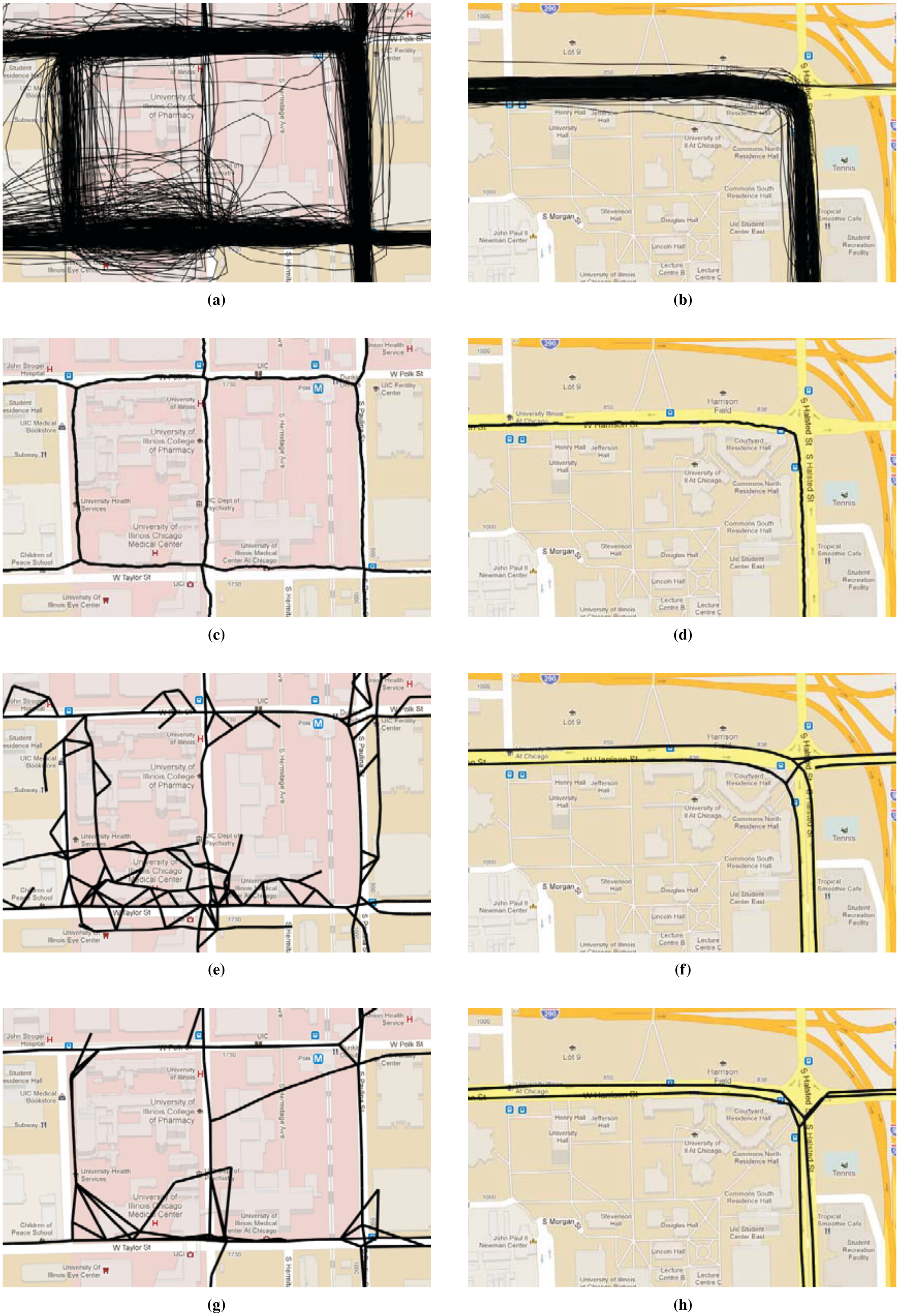

FIGURE 5 Raw data and results from three implemented algorithms in areas with high and low GPS error: (*a*) raw data, high-error sample; (*b*) raw data, low-error sample; (*c*) Davies algorithm, high-error sample; (*d*) Davies algorithm, low-error sample; (*e*) Cao algorithm, high-error sample; (*f*) Cao algorithm, low-error sample; (*g*) Edelkamp algorithm, high-error sample; and (*h*) Edelkamp algorithm, low-error sample.

an example of the distribution of the traces and GPS errors found in the study data set for one of these areas of high GPS error. In this evaluation, the entire data set was studied, as was a subset of the data drawn from an area of low-rise buildings where there is very little GPS error (see the sample in Figure 5*b*). Because GPS error can be a problem for map inference algorithms, the partitioning of the data in this way allows the performance of the algorithms to be tested and compared on both a realistic and a somewhat idealized data set. The road maps generated by each algorithm are depicted in Figure 5, *c* to *h.*

Davies et al. (*4*)

Visually, the algorithm from Davies et al. produces superior maps in areas with large GPS error. Because this method avoids treating GPS traces individually and, instead, uses them in aggregate to find the road boundaries, the method is able to create one road from a large collection of relatively diverse traces. This aspect of the algorithm is illustrated in the difficult high-error case in Figure 5*c,* in which the algorithm accurately extracted the road topology without adding any extraneous edges. In the low-error case in Figure 5*d,* the result is largely the same. However, note the absence of the less-traveled road segment on the right. This absence illustrates the fundamental trade-off of this algorithm: any given threshold value must compromise between introducing noise in high-density areas and losing infrequently traveled edges in low-density areas.

Cao and Krumm (*5*)

The clarification preprocessing step performed in the algorithm developed by Cao and Krumm helps to reduce GPS noise before the map generation and, in noise-free areas, results in cleanly generated maps, as demonstrated in Figure 5*f.* However, the clarification has its limits in areas with high GPS error, in which spatially dispersed traces are unable to become tightly banded. As a result, when the trace-merging method is applied to the clarified data, residual noise results in the spurious roads seen in Figure 5*e.* Although this algorithm attempts to prune spurious roads after map generation, its efforts are largely futile in areas of high GPS noise in which edge volume is widely distributed.

Edelkamp and Schrödl (*3*)

The algorithm in Edelkamp and Schrödl creates road segments by joining clusters based on the underlying trace data and works well in areas with low GPS noise, as can be seen in Figure 5*h.* However, this cluster-joining method is easily led astray by GPS noise and results in spurious roads being produced, as illustrated in Figure 5*g*. Because this algorithm does not attempt to prune spurious roads after generation, all of these roads remain in the final road map. However, the implementation described here does not include intersection refinement and lane finding. These features would likely have improved the results on the low-error data set but would have had little or no effect on the high-error data set.

QUANTITATIVE EVALUATION

In this section, the robust map comparison method described earlier is used to quantitatively evaluate the three representative algorithms described in the previous section. Parameter selection and implementation are also discussed.

Main Quantitative Results

Figure 6 shows the main results and shows the performance of the three algorithms for a varying matching threshold. The matching threshold is the allowable distance between a hole and a marble. A detailed discussion of these results follows.

Figure 6*a* shows the performance of the three algorithms as tested on the full data set. For small matching thresholds, the fine-grained spatial accuracy of the algorithm is significant. The Davies and Cao algorithms both underperform the Edelkamp algorithm at a 5-m matching threshold. For the Davies algorithm, this performance is explained by the fact that the algorithm generates a single, bidirectional centerline, whereas the others produce one edge in each direction. On a wide road, the distance between the centerline and the center of the road in one direction may well exceed 5 m. A similar problem is exhibited by the Cao algorithm, in which lanes in opposite directions are artificially spread apart to improve legibility. This introduces a slight error, which shows up as a poor matching performance for a 5-m threshold.

In the low-error data set, all of the algorithms unsurprisingly achieved a higher *F*-score (see Figure 6*b*). However, although the Davies algorithm continued to outperform the Edelkamp algorithm at 15- and 25-m matching thresholds, the prevalence of wide, median-separated two-way roads in this data set allowed the Edelkamp algorithm to outperform the Davies algorithm at both the 5- and 10-m matching thresholds.

Addressing Directionality Finding in Davies et al. (4)

In contrast with the other two approaches, the Davies algorithm generates a single centerline for each street, with a directionality annotation. As mentioned earlier, the authors in Davies et al. use a technique that measures the direction of travel through each grid cell to extract the permitted directions of travel. However, in the testing described in this paper, this method did not fare well. Many streets that were bidirectional in the ground truth were not represented as such in the inferred map. To correct for this problem, two different modifications to the Davies algorithm were made.

First, the direction-finding component of the Davies algorithm was discarded, making every road bidirectional. Although this modification allowed bidirectional roads whose directionality had been incorrectly inferred to be correctly represented, the modification also introduced the problem of incorrectly representing one-way streets as bidirectional. On the full data set, which included a large number of one-way streets, this trade-off resulted in a performance decrease, as can be seen in Figure 6*c*. In the low-error data set, however, which consisted predominantly of bidirectional roads, the bidirectional Davies implementation achieved a notable performance gain (see Figure 6*d*).

As an alternative to the directionality-finding technique in Davies et al. (*4*), the GPS traces were matched onto the inferred bidirectional map through the use of Viterbi map matching (*31*). The resulting sequences of road segments were then used to infer road directionality. The relative performance of the three techniques is shown in Figure 6, *c* and *d.* The line marked “Davies” is identical to the one in Figure 6, *a* and *b.* On top of an already strong performance, this map-matching technique gives the KDE-based algorithm by Davies et al. a significant advantage over the alternative approaches.

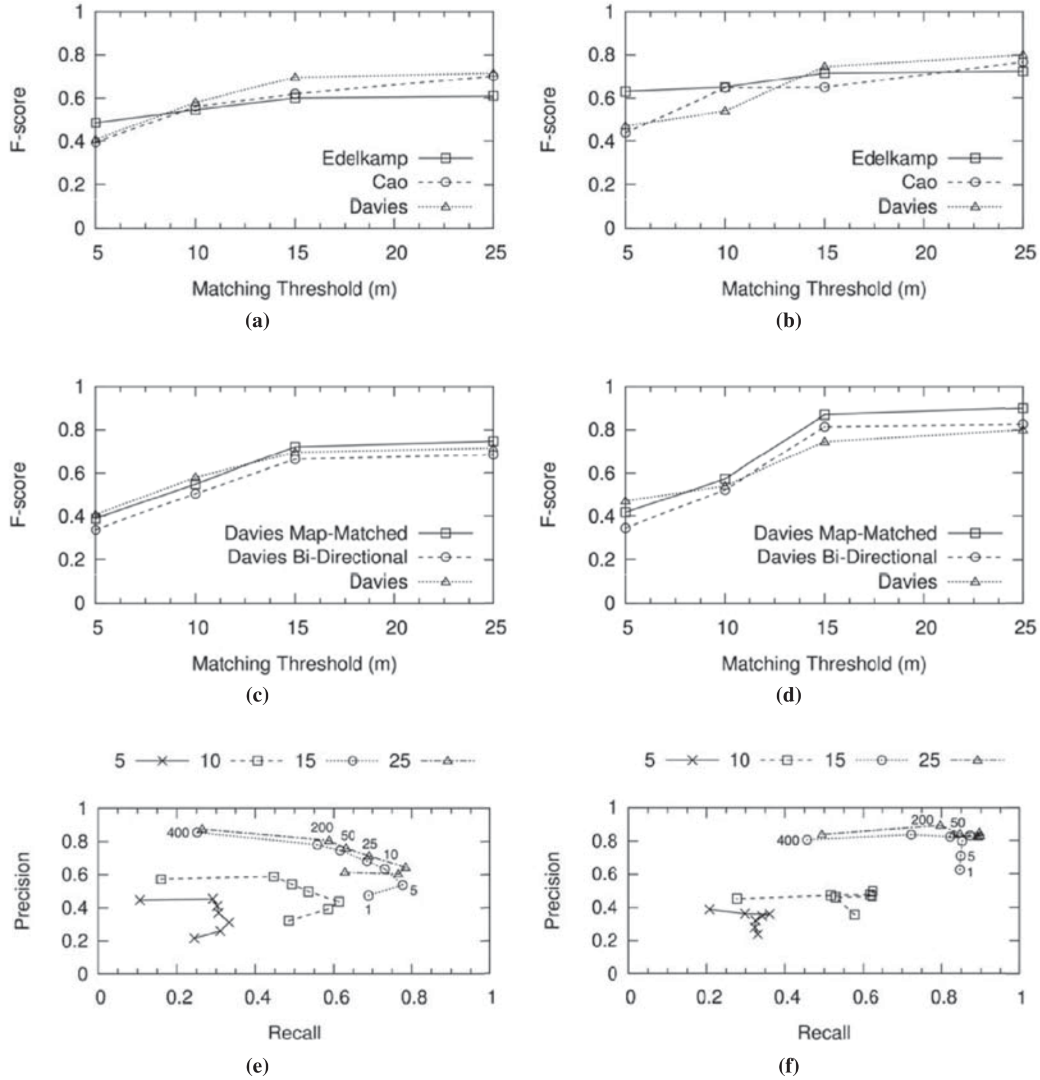

FIGURE 6 Results of algorithm analysis: (*a*) *F*-score, three algorithms, full data set; (*b*) *F*-score, three algorithms, low-error data set; (*c*) *F*-score, three Davies algorithms, full data set; (*d*) *F*-score, three Davies algorithms, low-error data set; (*e*) precision and recall, Davies map-matched algorithm, varying density threshold, full data set; and (*f*) precision and recall, Davies map-matched algorithm, varying density threshold, low-error data set.

Remaining Challenges

Some challenges remain, even with the map-matching improvements made to the Davies algorithm. Primarily, the choice of a density threshold for the Davies algorithm is made globally across the entire map. If the chosen threshold value is too low, excess noise is added to the map in the form of spurious roads. If the value is too high, portions of the map with relatively low density are treated as spurious and removed, resulting in missing roads. This behavior is illustrated in Figure 6, *e* and *f*. It can be seen that as the density threshold increases, the proportion of spurious edges decreases (i.e., precision increases), and the proportion of missing edges first decreases (i.e., recall increases) as excess noise is removed and then increases (i.e., recall decreases) as low-density roads are pruned from the map. It is believed that this insight is the key to further improvements to the KDE-based method: only marginal improvements will be made as long as the constant threshold used in the current algorithm remains.

Parameter Sensitivity and Implementation Details

Each of the algorithms described earlier includes several tuning parameters. A sensitivity analysis was conducted to determine which of those parameters most significantly affect the performance by fixing the values of one parameter and allowing all the others to vary. Some of the results of this analysis are discussed in the following sections.

Implementation of Davies et al. (4)

The implementation of the Davies algorithm closely follows the description given in the paper by Davies et al. (*4*), with the additional modifications discussed in the section on addressing directionality finding in the algorithm. The parameters for this algorithm are the cell size, the density threshold, the kernel bandwidth, and

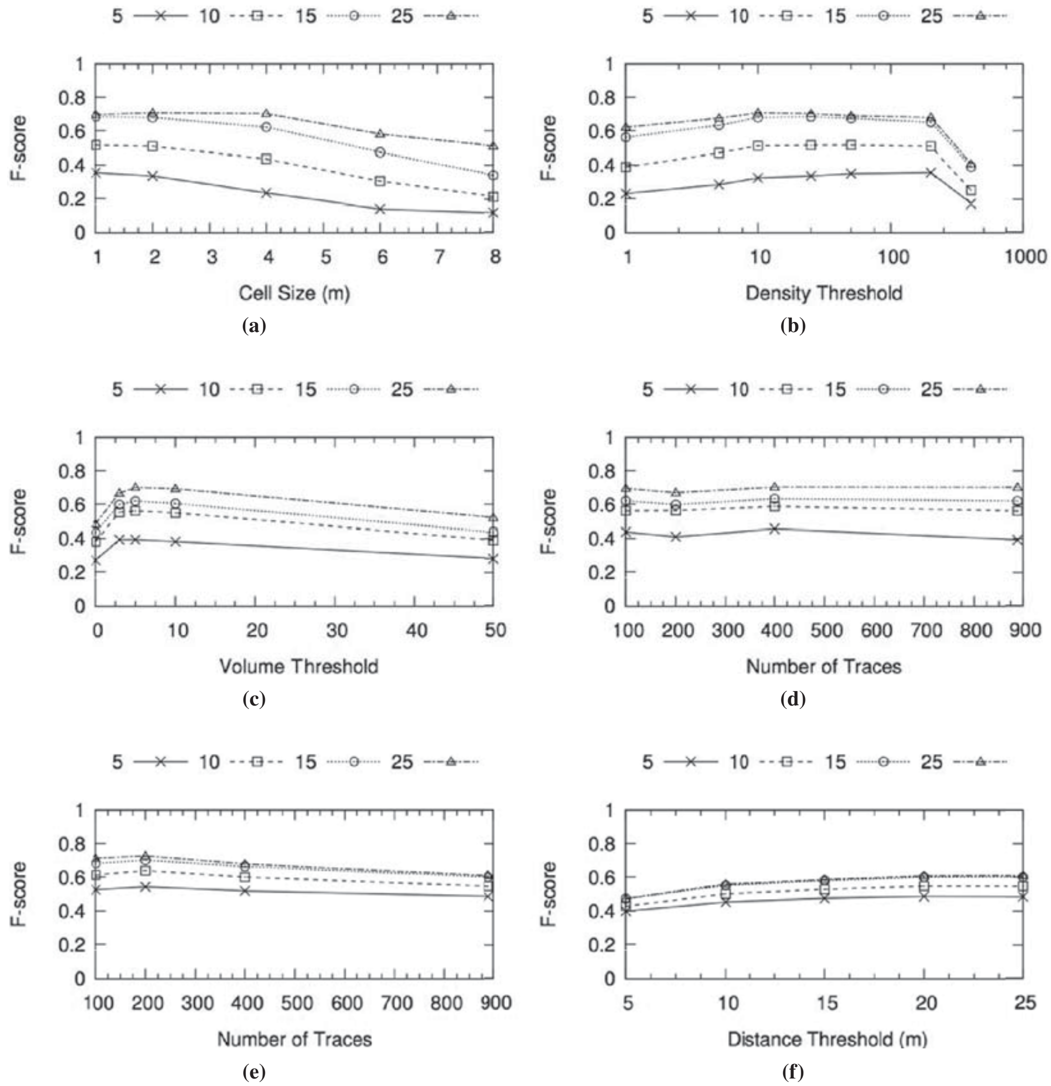

FIGURE 7 Parameter sensitivity testing over several matching distance thresholds: (*a*) cell size parameter (Davies map-matched algorithm), (*b*) density parameter (Davies map-matched algorithm, density threshold = number of traces per cell), (*c*) volume parameter (Cao algorithm, volume threshold = number of traces), (*d*) number of traces used (Cao algorithm), (*e*) number of traces used (Edelkamp algorithm), and (*f*) distance parameter (Edelkamp algorithm).

the number of traces used to generate the map. Figure 7, *a* and *b,* show this algorithm's sensitivity to cell size and density threshold. Figure 7*a* shows that a smaller cell size will always produce a better result with a fixed kernel bandwidth, as this simply improves granularity. Figure 7*b* shows that an increase in the density threshold increases accuracy as noise is removed from the map. However, this process will eventually lead to low-density roads being removed from the map and result in a decrease in accuracy.

Implementation of Cao and Krumm (5)

The implementation of the Cao algorithm follows the description in Cao and Krumm closely (*5*), with map generation being preceded by clarification. The parameters for this algorithm are the edge volume threshold, the location distance limit, the location bearing difference threshold, and the number of traces used to generate the map. Figure 7, *c* and *d*, show this algorithm's sensitivity to the edge volume threshold and the number of traces used to generate the map. As shows in Figure 7*c* (threshold range 0–5), an increase in the edge volume threshold, which is used to prune spurious edges from the generated map, increases map accuracy, as spurious edges are removed. However, if the pruning is too aggressive (i.e., the volume threshold is set too high), legitimate roads may end up being incorrectly removed, which decreases the accuracy (as seen in Figure 7*c,* threshold range 5–50). In Figure 7*d,* the performance decreases with the number of traces used to generate the map; this is a result of the increased noise that comes with a larger data set and this algorithm's inability to overcome that error.

Implementation of Edelkamp and Schrödl (3)

The implementation of the Edelkamp algorithm follows the description in Edelkamp and Schrödl (*3*), with the exception that no intersection refinement is performed. The parameters for this algorithm are the cluster seed interval, the intracluster bearing difference threshold, the intracluster distance threshold, and the number of traces used to generate the map. Figure 7, *e* and *f*, show this algorithm's sensitivity to the number of traces and the intracluster distance threshold parameters. Figure 7*e* shows that performance decreases with the number of traces used to generate the map, because, similarly to the Cao algorithm, the larger data set includes more noise, which this algorithm is unable to overcome. Also, Figure 7*f* data show that performance improves with increasing intracluster distance, as this increases the clusters' resistance to noise in the GPS traces.

Algorithm Run Time

The run time of these algorithms varies dramatically as a result of differences in algorithmic complexity. Particularly, the Cao algorithm suffers in dense neighborhoods, in which it exhibits quadratic complexity. On a subset of 100 traces, the Cao algorithm finished in 2.5 h, the Edelkamp algorithm in 73 s, and the Davies algorithm in 8 s. The map-matched version of the Davies algorithm finished in 106 s. On the full set of 899 traces, the Cao algorithm required 2.5 days, the Edelkamp algorithm 15 min, and the Davies algorithm 25 s. The map-matched version of the Davies algorithm finished in 14 min.

CONCLUSION

Robust quantitative evaluation methods and a rigorous comparison with prior work are important tools for furthering any field of scientific inquiry. With the new tools presented here, three algorithms from the literature were compared. Overall, the algorithm by Davies et al. (*4*) was found to significantly outperform the others under a variety of conditions. Through the quantitative evaluation method presented in this paper, opportunities for further improvement to this algorithm were identified. Some of these improvements were implemented and evaluated here and yielded a further, significant performance improvement. The new tools offered in this paper are expected to help bring significant advances to the study of map inference from GPS traces.

ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation, as well as by the Transportation Research Board of the National Academies through Transit Innovations Deserving Exploratory Analysis Project-65: A Context-Aware Transit Navigator.

REFERENCES

1. Agamennoni, G., J. Nieto, and E. Nebot. Robust Inference of Principal Road Paths for Intelligent Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 12, No. 1, 2011, pp. 298–308.
2. BITS Networked Systems Laboratory. http://bits.cs.uic.edu/. Accessed July 22, 2011.
3. Edelkamp, S., and S. Schrödl. Route Planning and Map Inference with Global Positioning Traces. In *Computer Science in Perspective* (R. Klein, H.-W. Six, and L. Wegner, eds.), Vol. 2598 of *Lecture Notes in Computer Science,* Springer, Berlin, 2003, pp. 128–151.
4. Davies, J. J., A. R. Beresford, and A. Hopper. Scalable, Distributed, Real-Time Map Generation. *IEEE Pervasive Computing,* Vol. 5, No. 4, 2006, pp. 47–54.
5. Cao, L., and J. Krumm. From GPS Traces to a Routable Road Map. *Proc., 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems,* Seattle, Wash., Association for Computing Machinery, New York, 2009, pp. 3–12.
6. Schroedl, S., K. Wagstaff, S. Rogers, P. Langley, and C. Wilson. Mining GPS Traces for Map Refinement. *Data Mining and Knowledge Discovery,* Vol. 9, No. 1, 2004, pp. 59–87.
7. Niehoefer, B., R. Burda, C. Wietfeld, F. Bauer, and O. Lueert. GPS Community Map Generation for Enhanced Routing Methods Based on Trace-Collection by Mobile Phones. *Proc., 1st International Conference on Advances in Satellite and Space Communications,* Colmar, France, IEEE, New York, 2009, pp. 156–161.
8. Mitchell, T. *Machine Learning,* 1st ed. McGraw-Hill Education, New York, 1997.
9. Scott, D. W. Kernel Density Estimators. In *Multivariate Density Estimation: Theory, Practice, and Visualization,* John Wiley & Sons, Inc., Hoboken, N.J., 2008, pp. 125–193.
10. Worrall, S., and E. Nebot. Automated Process for Generating Digitised Maps Through GPS Data Compression. *Proc., Australasian Conference on Robotics and Automation,* Brisbane, Australia, Australian Robotics and Automation Association, Sydney, Australia, 2007.
11. Guo, T., K. Iwamura, and M. Koga. Towards High Accuracy Road Maps Generation from Massive GPS Trace Data. *Proc., International Geoscience and Remote Sensing Symposium,* Barcelona, Spain, IEEE, New York, 2007, pp. 667–670.
12. Chen, C., and Y. Cheng. Roads Digital Map Generation with Multi-Track GPS Data. *Proc., International Workshop on Education Technology and Training and International Workshop on Geoscience and Remote Sensing,* Shanghai, China, Vol. 1, IEEE, New York, 2008, pp. 508–511.
13. Shi, W., S. Shen, and Y. Liu. Automatic Generation of Road Network Map from Massive GPS, Vehicle Trajectories. *Proc., Conference on 12th International Intelligent Transportation Systems,* St. Louis, Mo., IEEE, New York, 2009, pp. 1–6.
14. Jang, S., T. Kim, and E. Lee. Map Generation System with Lightweight GPS Trace Data. *Proc., 12th International Conference on Advanced Communication Technology,* Gangwon-Do, South Korea, Vol. 2, IEEE, New York, 2010, pp. 1489–1493.
15. Naver. http://dev.naver.com/openapi/apis/map/. Accessed Nov. 14, 2011.
16. Piegl, L., and W. Tiller. *The NURBS Book,* 2nd ed. Springer-Verlag, New York, 1997.
17. NAVTEQ Maps and Traffic. http://www.navteq.com/. Accessed July 22, 2011.
18. Wong, S. S. M. *Computational Methods in Physics and Engineering.* World Scientific, Singapore, 1997.
19. Hastie, T., and W. Stuetzle. *Principal Curves.* Technical report 151. University of Washington, Seattle, 1988.
20. Agamennoni, G., J. I. Nieto, and E. M. Nebot. Technical report: Inference of Principal Road Paths Using GPS Data. Technical report. Australian Centre for Field Robotics, University of Sydney, Australia, 2010. http://www-personal.acfr.usyd.edu.au/jnieto/Research-files/PRPInferenceReport2010.
21. Khanna, M. *Introduction to Particle Physics.* Prentice-Hall, New Delhi, India, 2004.
22. Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, 2009.
23. Leler, W. J. Human Vision, Anti-Aliasing, and the Cheap 4000 Line Display. *Proc., 7th Annual Conference on Computer Graphics and Interactive Techniques,* Seattle, Wash., Association for Computing Machinery, New York, 1980, pp. 308–313.
24. Yokoi, S., J. Toriwaki, and T. Fukumura. An Analysis of Topological Properties of Digitized Binary Pictures Using Local Features. *Computer Graphics and Image Processing,* Vol. 4, No. 1, 1975, pp. 63–73.
25. Aurenhammer, F. Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys,* Vol. 23, No. 3, 1991, pp. 345–405.
26. Conte, D., P. Foggia, C. Sansone, and M. Vento. Thirty Years of Graph Matching in Pattern Recognition. *International Journal of Pattern Recognition & Artificial Intelligence,* Vol. 18, No. 3, 2004, pp. 265–298.
27. Cormen, T. *Introduction to Algorithms.* MIT Press, Cambridge, Mass., 2001.
28. Bunke, H. On a Relation Between Graph Edit Distance and Maximum Common Subgraph. *Pattern Recognition Letters,* Vol. 18, No. 8, 1997, pp. 689–694.
29. Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data.* Springer, New York, 2007.
30. OpenStreetMap. http://www.openstreetmap.org/. Accessed July 22, 2011.
31. Thiagarajan, A., L. Sivalingam, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan. *Proc., 7th ACM Conference on Embedded Networked Sensor Systems,* Berkeley, Calif., Association for Computing Machinery, New York, 2009, pp. 85–98.

The Geographic Information Science and Applications Committee peer-reviewed this paper.