STEFAN EDELKAMP AND DAMIAN SULEWSKI, TU DORTMUND
FRANCISCO C. PEREIRA AND HUGO COSTA, UNIVERSITY OF DORTMUND

COLLABORATIVE MAP GENERATION — *SURVEY AND ARCHITECTURE PROPOSAL*

INTRODUCTION

The current widespread use and higher quality of GPS devices is drastically increasing the need for up-to-date digital maps to feed better and more reliable intelligent travelling assistance services.

While on one hand the main industrial players can be seen to make large investments in order to satisfy this demand, on the other, a new trend is slowly emerging from the Internet social networking trend (sometimes referred to as 'Web 2.0') which should not be neglected. As in many other cases, the commercial approaches are superior in terms of quality and in coverage at the beginning, but the effect of a community working together is unexpectedly powerful. See, for example, the case of Wikipedia (as opposed to encyclopaedias such as Encarta or Britannica).

We more specifically refer to Collaborative Map Generation, which consists of jointly building a geographical map of a region or a city out of shared geo-referenced traces (normally GPS). Up until the present, a few efforts have been made, the OpenStreetMap project (http://www.openstreetmap.org, accessed 23 July 2008) probably being the best known. There are, however, two major hurdles to the success of this approach: the aggregation of new traces has to be done manually and, even worse, individual traces are rarely devoid of errors, in spite of the high quality of current devices.

In this collaborative approach, the input data (GPS traces) is inherently rich in terms of information about individual and social mobility. Its aggregation can provide more than a roadmap – it represents real urban mobility. In this sense, since this framework necessarily needs a set of methodologies to automate common processes such as filtering and smoothing, mode detection or statistical analysis, a very useful outcome is the deployment of such information for transport and urban planners. One could, for example, deduce mobility trends and predict possible improvements in transport networks (cf. chapter 11). Until the present, this has mainly taken place manually, sometimes also using simulators calibrated with sampled data (sometimes even from lengthy personal interviews). A GPS sharing framework in the form described here would give dramatic improvements in time efficiency.

It is thus clear that an automated Collaborative Map Generation platform also provides a wealth of information regarding individual and collective mobility patterns, provided that a contract of privacy, security and trust is provided to all contributors (as is the case in any Web2.0 platform). At an individual level, for example, the 'personal network' can be determined or better routes and transport means suggested. At an urban level, for example, congestion points, places where people stay for periods of time and transport means usage can be detected, to name a few indicators.

The first breakthrough in this context will take place happen when efficient trace aggregation is achieved. In this chapter, we intend to present the Map Making State-of-the-Art and discuss current and future prospects for the development of an automated methodology for map aggregation that takes into account the need for integration of mobility data and the social networking trend, which we believe will eventually become the main source of geographical maps. This will allow us to abstract a general architecture for a Collaborative Map Generation system and discuss in some detail the technical challenges for each module (and its current solutions). In doing so, we hope to show that as a very relevant and desirable 'side effect', a set of algorithms must be developed that will help with regard to those Transport and Urban management tasks referred to above. We address filtering, map matching, update and aggregation, steps for the construction of the maps, and some efficient algorithms and data structures that are used to compress, process and query the map once generated. The chapter closes with a number of conclusions.

STATE-OF-THE-ART

Making maps

Map making (or cartography) is a discipline that has been subject to many technical revolutions in the course of time. From angle measurement to the North star aided by sextants and telescopes to more sophisticated settings, such as aerial photography and laser range finders, cartographers have given their best to achieve maps of the highest quality. The still recent age of artificial satellites in the orbit of the Earth (which started in 1957 with Russia's Sputnik) produced far-reaching improvements. This revolution generated two important new tools for map-making: satellite imagery and the Global Positioning System (GPS). The former enabled cartographers to start making high-precision global maps and, aided by the higher definition of airplane photography and localized measurements (e.g. altitude, pressure, temperature), to quickly acquire large quantities of geographical data. With the advent of micro-computers, this high quantity and quality of databases eventually led to the emergence of Geographic Information Systems (GIS), which are nowadays fundamental for a range of applications (also see chapter 4).

Differently to satellite imagery, GPS technology reduces the referential to a single point (as opposed to entire maps given by pictures). This enables the individualised use of geo-reference and opens up a myriad of new applications, ranging from navigation to location-based services, or LBS (e.g. "where is the nearest Restaurant?", "What is the weather forecast for today?"). It is now common to find vehicles and mobile phones or PDAs with GPS receivers, and there is a dramatic increase in the number of these applications, both commercial and freeware-based (cf. chapter 5).

Again, for the cartographer, GPS has also meant serious improvements. Particularly with Differential GPS (DGPS) [1] and Real Time Kinematics (RTK) [2], which allow for sub-meter accuracy, these technicians can now guarantee a more than satisfactory accuracy in a large number of features (e.g. roads, buildings). Moreover, the popularity of GPS and handheld devices has contributed to the low prices and lightweight solutions that can now be found on the market, and which are currently highly favoured by cartography experts (Wadhwani, 2001).

At an industrial level (e.g. Tele Atlas, http://www.teleatlas.com, accessed 23 July 2008), the mappings have to be made at a systematic and intense basis, and it is therefore common to use special purpose vans with GPS (DGPS when available), odometer and cameras (Desmet, 2005). These techniques have been applied for several years (e.g. by Grejner-Brzezinska (1995)). The major problem with this solution is the need for constant updates. Whenever a change in the area is made (e.g. new roads, change in traffic direction, speed limit, etc.), an update has to be made of the map. Furthermore, these approaches tend to neglect pedestrian and alternative transport means (e.g. bike, cf. chapter 13), which would imply even more unstable maps.

The common process for updating geographical maps uses GPS as well as satellite imagery. Some systems and algorithms exist (Yun et al, 2004; Gerke et al., 2004) that directly or indirectly help identify visible road geometry and associate it with geographic positions. However, such approaches still require close human attention and, above all, cannot identify several important route features (e.g. traffic direction and speed limits) or non-standard roads (e.g. off-road). They thereore still have to be complemented by careful high precision GPS data collection on ground. It is, consequently still a very slow and resource-consuming process.

Autonomous systems

Given the need and costs of constant map updating, the search for autonomous systems was the logical next step. The two main map provider companies (TeleAtlas and NavteQ) and four major car manufacturers (BMW, FIAT, Daimler and Volvo), have been involved in two projects under the coordination of the ERTICO European consortium on Intelligent Transport Systems (ITS) (ERTICO, 2007): ActMAP and FeedMAP. ActMAP (Flament, 2005) lasted for 3 years (2002-2005) and its aim was to investigate and develop "mechanisms for online incremental updates of digital map databases into vehicles. Up-to-date map components containing dynamic or static location-based content should be integrated and/or attached to the in-vehicle digital map". The final results include reference architecture for the on-line update of digital maps. In the design specification developed, the authors proposed methodologies, procedures and data formats to enable an efficient process of updating maps, from map providers to users.

The FeedMAP project expects to complete the loop by transmitting data from the client to the provider, integrating it semi-automatically into the map. This project commenced in 2006 and is expected to be finished by the end of 2008. In FeedMAP, the main aim is to "assess the technical and economic feasibility of map data correction by providing a map data feedback loop applied to a map data updating framework using the ActMAP standardized exchange formats and mechanisms". As can be seen from **Illustration 12.1**, the authors propose a complete cooperative process that takes as actors the map providers, the clients and public authorities.

Commercially, the results of these and related projects are already visible. The most salient example is TomTom Map Share technology (http://www.tomtom.com/page.php?Page=Mapshare, accessed 23 July 2008), in which the user can record traces and send them to the map server. The whole process is still essentially manual (the symbolic information, the error correction, the aggregation), and it therefore strongly resembles OpenStreetMap philosophy, mentioned in the next section.

On the academic side, Schroedl et al. (2004) fully specify a system for generating geographical maps from DGPS traces. Their approach consists of successive processing steps: individual vehicle trajectories are divided into road segments and intersections, a road centerline is derived for each segment, lane positions are determined by clustering the perpendicular offsets from it, and the transitions of traces between segments are utilized in the generation

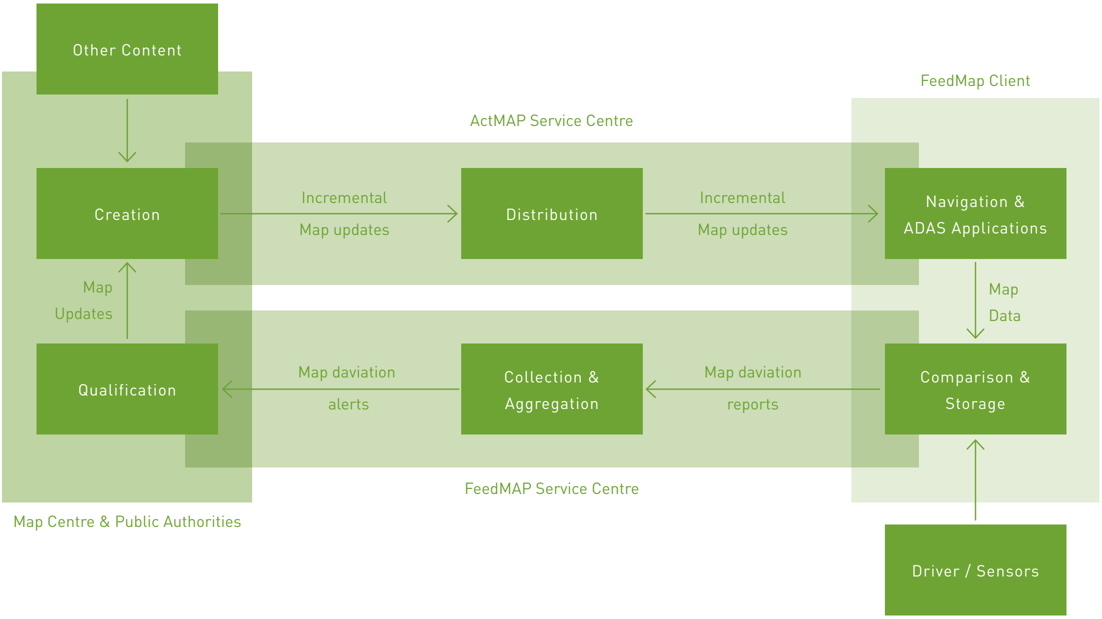


Illustration 12.1 Architecture of FeedMAP

of intersection models (Schroedl et al., 2004). Although as far as we know the most extensive as yet within academia, no further results have been published, possibly due to an industrial strategy (an associated patent was registered – US Patent nr. 6385539).

Partly a follow-up, the work of Brüntrup et al. (2005), 'Map Generation', may be the first that explicitly considers the collaborative side. In this project, the authors developed a system that incrementally generates a map of the world, starting from an 'empty' map and gradually adding new data collected from GPS traces. The system applies Artificial Intelligence search techniques to perform what is commonly known as Map Matching, the task of identifying which parts of an already existing map a given set of coordinates should correspond to. When no map segment is found, the set of coordinates should correspond to a new road (and thus aggregate it as a new segment of the map). 'Map Generation' performs reasonably well as long as the first traces for each segment are of a high quality, but has difficulties in improving an initial map with GPS errors (which is rather common in a realistic setting). In other words, it lacks a more precise feedback correction mechanism, perhaps with a 'forgetting factor'.

The observable investment in ActMAP and FeedMAP allows us to predict an improvement in the quality of maps in car navigation commercial systems in the coming years. However, the essence of these approaches (particularly the FeedMAP side) relies on having users participate and contributing. Such an approach is mainly common in Web 2.0 applications (e.g. Wikipedia, del.icio.us, MySpace, Blogspot, WikiMapia), and is rarely successful when based on a commercial relationship.

On the other hand, although they have fewer resources, the academic approaches are improving in quality and their constituents (namely the Map Matching modules) are currently extremely efficient. It has become difficult to make a technical comparison between the two, due to lack of information. From what is publicly known, industrial players are focused on improving maps for car navigation and integrating the system with other ADAS (Advanced Driver Assistance Systems), while academia is focused on general-purpose solutions (e.g. bike or pedestrian navigation).

Mapmaking and Web2.0

Collaborative projects currently exist for manually creating maps from GPS traces. The most popular and complete one is OpenStreetMap (OpenStreetMap, 2007). OpenStreetMap is a project "aimed squarely at creating and providing free geographic data such as street maps to anyone who wants them." In this project, each user can upload his/her GPS trace logs and use an editor (jOSM) to complete/correct the data (e.g. define directions, connections to other segments, add names, etc.). The resulting joint 'map of the world' can then be seen and the data can be processed for other uses (such as car navigation). In general, the major drawback of such a system is the manual effort demanded of ordinary users.

Map making and mobility analysis

The space in terms of the relationship between these systems and urban mobility analysis is still rather unfilled. Other than average speed, hot spots, points-of-interest and typical map features (e.g. gas station, parking lot, hospital, etc.), there is not much more information in the process added to the inferred route maps. OpenStreetMap representation is still open to added features and some related projects are actually adding external statistics (e.g. Stockholm GIS info, with dangerous and goods roads, nature reserves, built up areas, etc.; http://www.gisdata.se, accessed 2007), but a great deal more has to be done to achieve valuable content.

AN ARCHITECTURE FOR COLLABORATIVE MAP GENERATION

From a detailed analysis of previous work, we abstracted an architecture for Collaborative Map Generation (**Illustration 12.2**) that covers those works and that will direct our own next research steps, described in this section. We assume that the system is fully automatic, and that a base map may be initially absent (an initial empty map of the world).

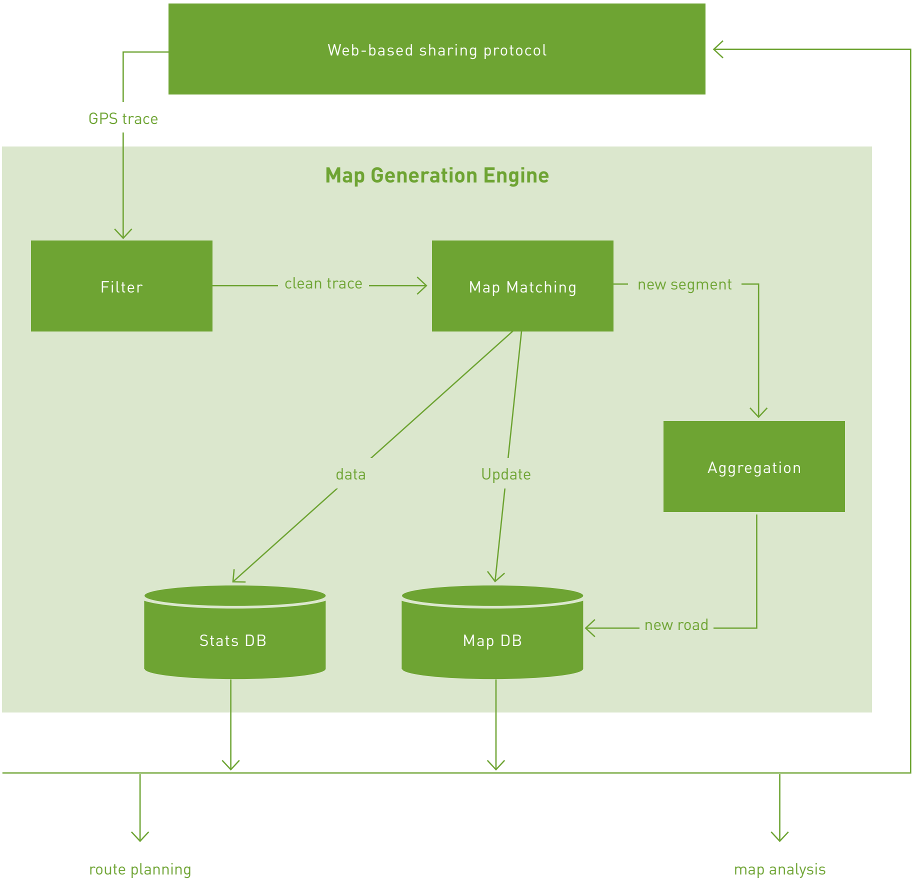


Illustration 12.2
General Architecture

As with many other Web2.0 applications, we include a *front-end interface* accessible via a Web browser. This will be responsible for feeding the Map Generation engine with GPS traces. Notice that we expect each trace to be analysed separately (as opposed to what happens in the above-mentioned work of Schroedl et al. (2004), where traces are processed in batches). Each incoming GPS trace will then be *filtered* for consistency (as in the Map Generation project by Brüntrup et al. (2005)).

The filtered trace will then be matched to the existing map in the *Map Matching* module. The parts of each trace that achieve a correct match to the map (i.e. points that placed on the already existing roads of the map) will serve to *update* the *Map*. The procedure to be used in this correction mechanism consists of averaging the centerline according to the number of previous estimates and the confidence – according to DOP [3] – of the current one. The traces that reflect a 'new road' shall be *aggregated* taking into account the intersection treatment suggested by Schroedl et al. (2004). Regardless of being updated or aggregated, the *Map Matching* module will allow inference of mobility data that will feed the statistics (*Stats*) database (e.g. average speed, time spent in locations, most probable transport means used).

External services can be connected to this platform, for example for route planning or for map analysis (e.g. visualising cycling routes, transport usage, places where people stay for long times).

Filters

GPS traces always produce a considerable number of errors. There are many reasons for this: receiver clock errors, satellite clock errors, satellite orbit error, atmospheric effects (particularly Ionosphere and Troposphere) and multipath effect (e.g. when a signal is reflected in a building). Such error essentially affects the positioning estimation in each new calculation. This means, for example, that from one second to the next, the difference in estimation may vary in the order of meters. A study conducted by Refan and Mohammadi (2001) focused on averaging positioning estimates in a fixed point giving approximately 7 meters of standard deviation (with an amplitude of approximately 30 meters) for each axis, with a 5 minute sample of GPS points. These error values increase considerably when the receiver starts moving.

Many improvements to GPS accuracy are under research and some are already in production. In the case of DGPS and RTK, specific hardware conditions are necessary to effectively acquire accuracy improvements that are not available for low cost devices. A different technique, referred to as Dead Reckoning, consists of using other information (e.g. accelerometers, inertial sensors, assumption of correct route) to infer the precise location. These information sources

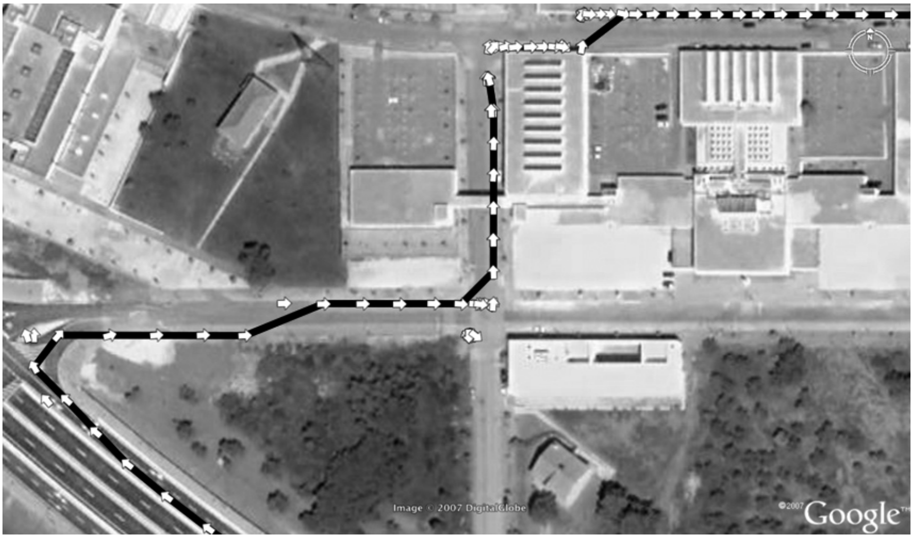


Illustration 12.3 Filter application to a GPS trace. The arrows indicate trace points, while the thick line corresponds to the filtered path proposed

are particularly useful in ‘urban canyons’ or even when no access to satellites is possible (e.g. indoors, tunnels). Although no special GPS hardware is necessary, Dead Reckoning normally uses other sensors or assumes uncertain facts, thus making the solution complex. In our case, we intend to use common off-the-shelf and low cost GPS receivers, and neither of these solutions is therefore applicable. The accuracy improvement has to be made exclusively with software post-processing of the traces. In other words, using filters.

Kalman filters, Recursive Least Squares and Linear Regression are currently being studied, with attention to car, bike and pedestrian traces. **Illustration 12.3**, for example, illustrates the result of the application of Cumulative Displacement Filter (inspired in Kalman filter). Particularly in slow movements (e.g. curves), the error can be seen to possibly strongly affect the road geometry. In terms of pedestrian traces, it has thus far been shown that Kalman related filters produce extremely bad results, while linear regression demonstrates fewer weaknesses. It is, however, noticeable that using a rule-based approach (e.g. people cannot move faster than 2m/s) allows for a reasonable first pass (without smoothing). **Illustration 12.4** shows an original pedestrian trace and a filtered one (indicating sub-traces, their beginning and ending, and places where the person stayed).

Notice that the problem we face goes beyond the common approaches: processing is made off-line (we have the entire time series, not only the ‘past’) and no auxiliary hardware resources are available.

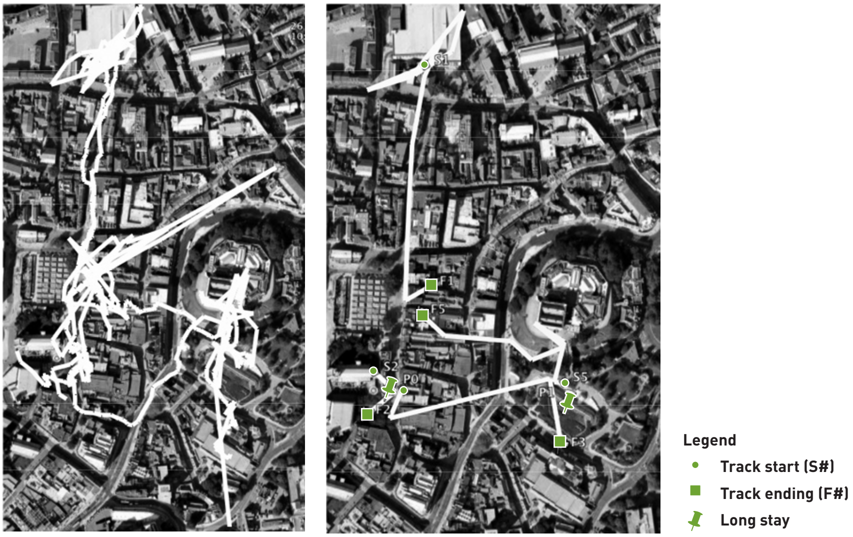


Illustration 12.4
Pedestrian trace (from Norwich, Spatial Metro project).
On the left side, the original trace data; on the right side, the filtered results

Illustration 12.5

Upper left:
Initial map (white line)
Upper right:
Recorded trace

Below:
Matching obtained with the GA (Triangles mean unmatched)

Map Matching

For the task of map matching, we need a method that takes advantage of the topology of both the new trace and of the map constructed so far. Furthermore, the offline nature of the system allows for preference on precision over performance (of course, within reasonable limits). A genetic algorithm is being designed that evolves each potential match according to minimization of distance, penalization of gaps and incorrect topology. For readers unfamiliar with genetic algorithms (GAs), we will hereby summarize the concept: in a GA, several solutions to a problem are generated randomly at the beginning (the initial population); according to a fitness evaluation, a portion of them can be selected to generate the following generation. These solutions can be crossed with each other, providing their 'genetic material' to new individuals, and can be subject to 'genetic mutation'. This process is thus repeated iteratively, generation after generation. After a number of generations a satisfactory solution is found, and the algorithm stops having found the best solution ('individual') so far [4]. In our case, a solution, or 'individual', is composed of a sequence of 'point-to-curve' matches (the 'genes'). Thus, in theory, each individual could have as many genes as points in the trace. To limit this complexity involved, we use a segmentation of the trace according to the idea of Chawathe (2007), in which a trace is divided into shorter parts bound by points with high confidence matches. Our fitness function consists of the weighted sum of average, maximum and minimum distances, the sum of the gap size (subsequences of unmatched points) and the sum of the jump size (when two consecutive points in a trace match to different roads). **Illustration 12.5** gives an example of the initial map obtained from previous traces (a), the trace as evaluated (b), and the GA result (c).

This algorithm will then be compared to others: Depth First Search (Brüntrup et al., 2005), Frechet curves (Brakatsoulas et al., 2005), Least Squares Estimation (Blewitt and Taylor, 2002), and Multiple Hypothesis (Marchal et al., 2005). Again, the experiments will be directed towards bike, car and pedestrian traces.

Aggregation

Whenever a trace is found that does not match the existing map, we have to consider a possible new road in the map. There are essentially two approaches to aggregation: incremental and batch-based. In the incremental approach, we determine the points of intersection of the new set of segments with the existing map. In the batch-based approach, we recalculate the entire sub-network that fits the area of traversal of the new trace (by recovering all the original traces).

Incremental Aggregation

An incremental approach for map generation has the advantage of time-efficiency, as compared to the batch-based approaches. For this reason, it can also be used on-line. However, due to errors in normal GPS receivers, particular care has to be taken not to propagate them.

The algorithm we currently have from the Map Generation project (Brüntrup et al., 2005) subdivides the world into tiles and for each new trace, the *trace processor* interpolates new trace nodes. In the main processing loop, three modules *walk* along the trace. The *scan module* is the first module and scans the environment of the trace for nodes that are candidates for merging. After each scan the *AI module* decides which of the nodes should actually be used. Finally the *apply module* uses these results to merge the nodes with the trace (which would actually correspond to an *update*, not to an *aggregation*) or create new nodes on the map. In **Illustration 12.6**, an example can be seen of the original traces provided (a) and the resulting portion of the map (b). An improvement for the inference results for off-roads and roundabouts applies genetic algorithms to tune parameters (Scholz 2006). The dynamically generated map can serve a series of applications, particularly for navigation and providing up-to-date statistical information about the roads (e.g. average speed/congestion). In **Illustration 12.7**, we can see a sequence of map 'snapshots' in time; the map grows as new traces arrive.
The map matching algorithm can already provide us with a *triage* that separates two different

Illustration 12.6
Left:
Set of traces around a roundabout
Right:
Aggregated map, obtained from the traces

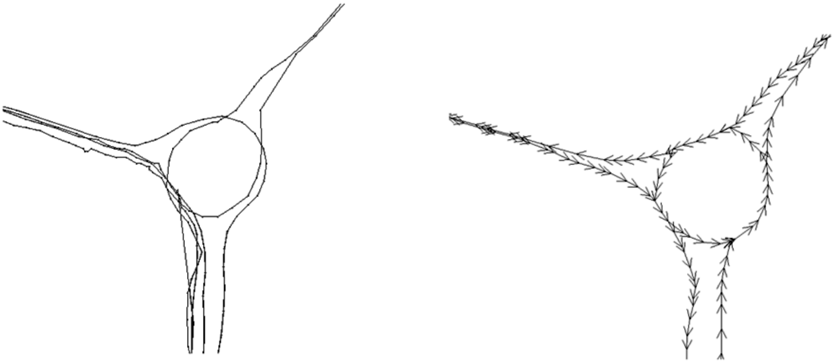
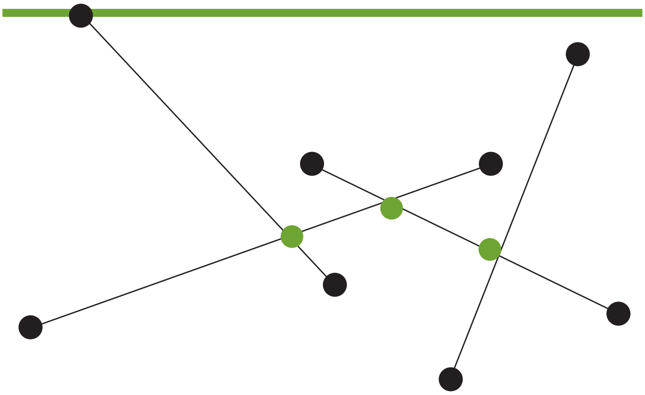


Illustration 12.7
Incremental Map Construction

Illustration 12.8
Segment Intersection of Batched-based Map Construction



tasks: updating the tiles with corrections to existing maps (Update phase), and adding entirely new segments to the map, which represent new roads (Aggregation phase).

The quality of the map will be sensitive to the successive updates and aggregations of traces. Although each new pass will provide correction, a bias towards the most recent ones will be inevitable. Such a problem can be attenuated if we apply batch-based construction periodically, by using all received traces so far (kept in the database) and applying them at once. This is set out in the next section.

Batch-based Aggregation

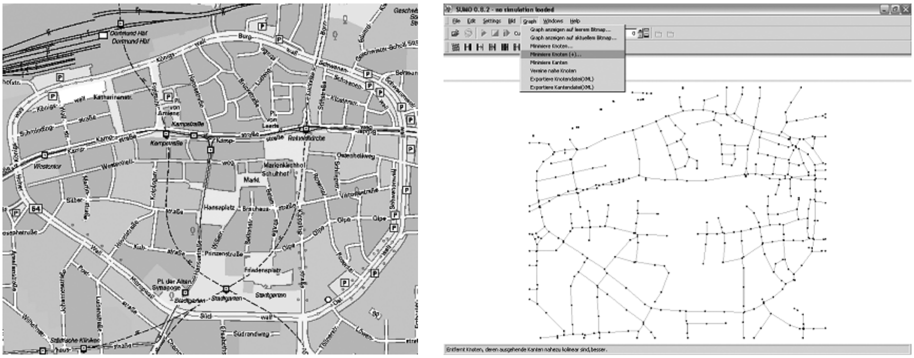
In a batch-based map construction algorithm, the traces are taken at once and organised in clusters (and sub-clusters) in order to make a *travel graph,* which is the embedded overlaid set of GPS traces together with the according intersections. To compute the superimposed graph, the sweep-line segment intersection algorithm (Bentley and Ottmann, 1979) is adapted (see **Illustration 12.8**). As opposed to the original algorithm, the generated graph is weighted and directed. At the intersections, the newly generated edges inherit direction, distance and time from the original data points. In typical travel networks, the number of edges is proportional to the number of nodes, as the node degree is bounded by a small constant. If it can be said that the graph is *planar* – as is often the case – the number of edges is linear in the number of nodes. Once the travel graph is built, many nodes of degree 2 remain. For computing the shortest paths these nodes can be merged by adding the distances of adjacent edges. Actually, only start, end and segment intersections remain, reducing the space complexity of the graph.

Batch-based map construction thus proposes to build the map 'all at once' from a batch of traces. This is only possible in an off-line basis. As said above, we propose the use of this method periodically (applied to a tile at a time) in order to 're-scan' the graph with all traces obtained over a period of time. Other methods can be applied, such as constructing a Base Map from (satellite or raster map) images.

Base Map Construction

In addition to satellite images, as referred to above, we may also use a method to extract calibrated road topology from raster maps to provide a *base map* for the collaborative map generation process. In many areas of interest, detailed vector maps are scarcely available and in some regions of the world, vector maps are not available at all. On the other hand, low-cost raster maps are frequently accessible. The images are often calibrated with respect to some form of global coordinate system to be translated to GPS. A bitmap is taken and different graphics filters used to infer the road geometry. We propose an aggregation algorithm that extracts road fragments and constructs a graph of the road network. To evaluate the proposed

Illustration 12.9
Base Map Generation (a) and inclusion in a Traffic Simulation Tool (b)



Entfernt Knoten, deren ausgehende Kanten nahezu kolinear sind,besser.

Illustration 12.10
Point Localisation Structures Quadtree

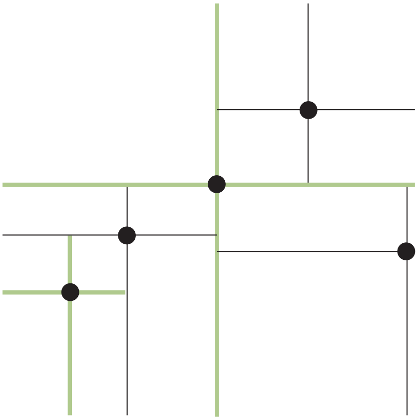
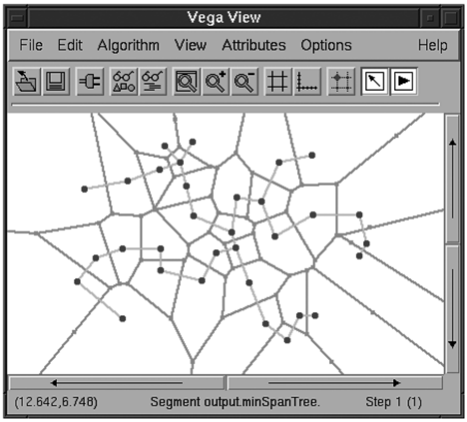


Illustration 12.11
Point Localisation via Voronoi Diagram.



(12.642,6.748) Segment output.minSpanTree. Step 1 (1) algorithms, the approach was integrated into SUMO (**Illustration 12.9**), a state-of-the-art traffic simulation tool for urban mobility (Drodzynski et al. 2007).

Efficient Map Representations

A map is beneficial to a user only if it can rapidly answer location queries. Before a query on the map based on given start and goal locations can be processed, their nearest corresponding entry nodes have to be found, i.e. efficient map matching in real time has to be carried out. For a set of queries, this is best accomplished by an assisting point localisation structure that supports nearest neighbour information.

Tile Regions

If the map is organized in form of tiles, we first have to find the set of tiles that a trace is located in, e.g. by looking at its bounding box of coordinates and retrieving this set for further processing. This has the advantage of allowing the integration of data to be distributed as far as the affected tiles do not overlap. The drawback is that a uniform distribution of fine-grained tiles is memory inefficient. One of the most interesting dynamic data structures for rapidly storing and retrieving tile information is *Quadtrees* (Finkel and Bentley, 1974), a balanced tree with children NW, NE, SW, and SE at each node (**see Illustration 12.10**).

Voronoi Regions

Another apparently suited data structure for nearest neighbour localization is the *Voronoi diagram* (Voronoi 1907). The structure consists of Voronoi regions $V(p)$ such that all points in the interior of $V(p)$ are nearer to p than to any other point in the point set (see **Illustration 12.11**). A search structure can be associated on top of the diagram or by its geometric dual, the Delaunay triangulation. A randomised construction for the triangulation and the associated search structure is presented by Berg et al. (1997).

Routing

Even though routing first seems to be unrelated to the map construction process, many search enhancements can be pre-processed and included in the map. Searching for the shortest route in the inferred graph can be sufficiently accomplished by a single run of the single-source shortest paths *algorithm of Dijkstra*.

Many modern navigation systems either provide their services through Internet portals, so that portable devices access large databases through communication with a server, or rely on portable devices with limited capacities. In the following, we address efficient algorithms and data structures to reply to frequent queries. Most of the algorithms exhibit the fact that the graph is embedded in the plane, so that refined geometric information on the set of all possible shortest paths can be associated to nodes or edges.

A* Search

Heuristic search is a well-known technique for reducing the number of expansions for a shortest path. This technique of *goal direction* includes an additional node evaluation function h into the search. The estimate that is applied to accelerate route planning measures the straight-line distance to the set of goal nodes. In this case, A* mimics Dijkstra's algorithm by changing the edge weights from $w(u,v)$ to $w(u,v)+h(v)-h(u)$ together with offset $h(s)$ given at the start node s (see Illustration 12.12). If one is interested in the shortest path in more terms than those of mere distance, the heuristic has to be adapted.

Illustration 12.12 Effect of Straight-line Distance Heuristics on the Search Space.

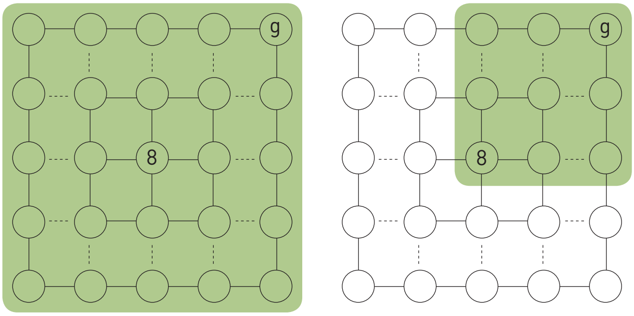
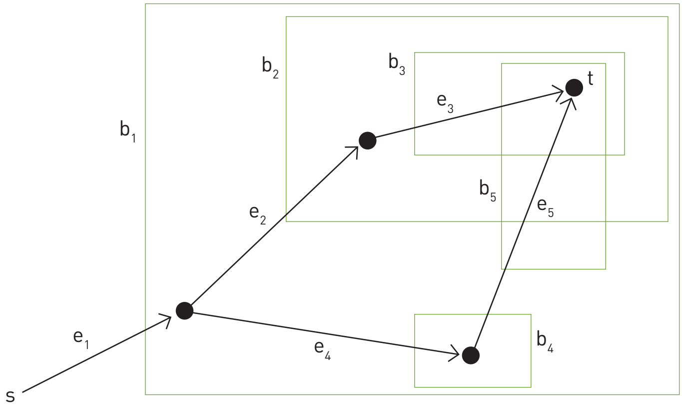


Illustration 12.13 Enhancing the Search with Geometric Containers.



Geometric Containers

Another possibility for decreasing the size of the search space is to ignore some of the neighbour points. The neighbours (or more precisely the incident edges to these neighbours) that can be safely ignored are those that are not on a shortest path to the target. In a pre-processing step, for each edge, we store the set of nodes that can be reached on a shortest path that starts with this particular edge. While running Dijkstra's algorithm or A*, we then refrain from inserting edges into the search queue that are not part of a shortest path to the target. The problem is the quadratic amount of space required to store this information, which is not available even for contracted graphs. Hence, we do not remember the set of nodes that can be reached on a shortest path for an edge, but approximations of it, so-called *containers*. Incorporating such geometric pruning preserves the completeness and optimality of a route planning algorithm, since at least one shortest path from the start to the goal is preserved.

Graph Rewriting

It can be advantageous to modify the routing graph simply to accelerate shortest path queries. One option is to apply abstraction to the graph structure by contracting nodes, to obtain a hierarchy of graph layers. More refined techniques have been established that reach out up to recent developments. A very influential approach is to insert *transit nodes* into the map to separate the base graph from the *highway graph* (Bast et al., 2007). Speed-ups of factors 100 and more have been obtained.

CONCLUSIONS

In this chapter, we have seen a smooth introduction to rising requests for collaborative and automated map construction on a set of GPS traces recorded by a host of individual devices. Besides the emergent roadmap, such methodology allows the elicitation of important information regarding individual and collective mobility. Examples include the inference of individual and crowd pedestrian networks or urban space use, correlated with time. By themselves, the statistics obtained from such a voluntarily built database can be substantial, for example for better calibration for simulations and to obtain detailed Origin-Destination matrices.

In terms of motifying individuals, collective map making is an answer to many important challenges for generating highly accurate, multi-modal and up-to-date maps. Automatically pre-processed and adapted positioning data can be used to accelerate the usability cycle on top of low cost devices. Maps in current use e.g. for navigation on the device are extended and refined on-the-fly by recording and integrating new traces.

Our working prototype (see **Illustration 12.14**) illustrates the advantage of the collaborative approach by comparing the aggregated map (a) on top of Google Earth (b), Google Street Map (c – notice the big error) and OpenStreetMap (d).

Besides surveying projects and existing solutions, this chapter reflected many algorithmic details as a portfolio to build a working architecture. We start with filtering of data, turn to incremental map aggregation via segment intersection, then to batch-based map construction, and the construction of a quick base map from raster map images. Moreover, for enhanced use of the map in a route planning system we show how to organize it for location queries and annotate it for accelerated shortest path search.

Illustration 12.14a-d
Collaborative Map as an Overlay on top of Google and OpenStreetMap.

ACKNOWLEDGEMENTS

The authors would like to thank DFG for its support in connection with the project ED74/3. Moreover, the work was also supported by the running of the map generation software by Björn Scholz and the use of the filters and aggregator by Nuno Pereira.

NOTES

1 In Differential GPS (DGPS), each earth-fixed station, which extremely accurately knows its position, detects the error of the incoming GPS signal and propagates a correction to the surrounding receivers through radio communication

2 In Real Time Kinematics (RTK), the philosophy is similar to DGPS, but corrections are made to the *carrier phase* measurements (as opposed to the messages contained in it).

3 Dilution Of Precision – An estimate of GPS quality based on satellite alignment with respect to the receiver

4 For more information on this subject we suggest, for example, Goldberg (1989)

REFERENCES

- Bast, H., Funke, S., Sanders, P. and Schultes, D. (2007) Fast routing in road networks with transit nodes, *Science Magazine*, April 2007.
- Bentley, J.L. and Ottmann, T.A. (1979) Algorithms for reporting and counting geometric intersections, *Transactions on Computing*, 28, pp. 643–647.
- Berg, M., Kreveld, M van., Overmars, M. and Schwarzkopf, O. (1997) Computational Geometry, *Algorithms and Applications*, Springer Verlag, New York.
- Blewitt, G. and Taylor, G. (2002) Geoffrey Blewitt and George Taylor. Mapping Dilution of Precision (MDOP) and map-matched GPS, *International Journal of Geographical Information Science*, 16(1): 55-67.
- Brakatsoulas, S., Pfoser, D., Salas, R. and Wenk, C. (2005) On map-matching vehicle tracking data, *Proceedings of the 31st international conference on Very large data bases (VLDB)*, Norway, 2005.
- Brüntrup, R, Edelkamp, S., Jabbar, S., and Scholz, B. (2005) Incremental map generation with GPS traces, *Proceedings of Intelligent Transportation Systems*, IEEE Publications.
- Chawathe, S. (2007) Segment-Based Map Matching, *Intelligent Vehicles Symposium*, 2007 IEEE, pp. 1190-1197.
- Drodzyinski, M., Edelkamp, S., Gaubatz, A., Jabbar, S., and Liebe, M. (2007) On Constructing a Base Map for Collaborative Map Generation and its Application in Urban Mobility Planning, *IEEE Conference on Intelligent Transportation Systems*, IEEE Press, pp. 678-683.
- Edelkamp, S., Jabbar, S. and Willhalm, T. (2005) Geometric travel planning, *IEEE Transactions on Intelligent Transportation Systems*, 6(1):5-16.
- ERTICO (2007) *ERTICO ITS Europe*, http://www.ertico.com, accessed 23 July 2008.
- Finkel, R. and Bentley, J.L. (1974) Quad trees: a data structure for retrieval on composite keys, *Acta Informatica* 4(1): 1-9.
- Flament, M. (2005) *ACTMap* Final Report, ERTICO, D 1.2. http://www.ertico.com/en/subprojects/actmap/public_documents/, accessed 23 July 2008.
- Gerke, M., Heipke, Ch. and Busch, A. (2004) Automated image-based verification of road databases, *Proceedings of 7th AGILE Conference on Geographic Information Science,* 29 April - 1 May 2004, Heraklion, Greece.
- Goldberg, D. (1989) Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston (Mass.).

- Grejner-Brzezinska, D. (1995) Positioning Accuracy of the GPSvan, *Proceedings of the 52nd Annual National Technical Meeting of the Institute of Navigation*, Palm Springs (California), pp. 657-665.
- Jabbar, S. (2003) *GPS-based navigation in static and dynamic environments*, Master's thesis, Universität Freiburg.
- Krajewicz, D., Hartinger, D., Hertkorn, G., Mieth, P., Rössel, C., Zimmer, J., and Wagner, P. (2003) Using the road traffic simulation (SUMO) for educational purposes. In: S.P. Hoogendoorn, S. Luding, P.H.L. Bovy, M. Schreckenberg and D.E. Wolf, eds, *Traffic and Granular Flow (TGF) 2003*, Springer, Berlin Heidelberg.
- Marchal, F., Hackney, J. and Axhausen, K.W. (2005) Efficient map-matching of large GPS data sets - Tests on a speed monitoring experiment in Zurich, *TRB annual meeting*, Washington D.C., January 2005.
- OpenStreetMap (2007) OpenStreetMap website. http://www.openstreetmap.org/ as of April 2007.
- Otto, H., Beuk, L., Aleksic, M., Meier, J., Loewenau, J., Flament, M., Guarise, A., Bracht, A., Capra, L., Bruns, K. and Sabel, H. (2004) *ACTMap Specification*, ERTICO. http://www.ertico.com/en/subprojects/actmap/public_documents/
- Refan, M.H. and Mohammadi, K. (2001) Point averaging of the position components, before and after S/A is turned off, *Asian GPS Conference,* India.
- Scholz, B. (2006) *Automatic inference of street maps based on GPS traces*. Master's thesis (in German), Universität Dortmund, Germany.
- Schroedl, S., Wagstaff, K., Rogers, S., Langley, P. and Wilson, C. (2004) Mining GPS Traces for Map Refinement, *Knowledge Discovery and Data Mining*, 9: 59-87.
- Voronoi, G. (1907) Nouvelles applications des paramètres continus à la théorie des formes quadratiques, *Journal für die Reine und Angewandte Mathematik*, 133: 97-178.
- Wadhwani, A. (2001) Recent advances in mobile GPS/GIS mapping technology. In: MAP India 2001. *GIS Development Newsletter*, gisdevelopment.net.
- Yun, L., Uchimura, K. and Wakisaka, S. (2004) Automatic Extraction of main road from Ikonos satellite imagery based on fuzzy reasoning, *Proceedings of the 8th International Conference on Applications of Advanced Technologies in Transportation*, no. 325, (CD-ROM), Beijing, 26-28 May 2004, pp. 641-646.