# ACM SIGSPATIAL GIS Cup 2012

Mohamed Ali
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052 USA
mali@microsoft.com

Travis Rautman
Institute of Technology
University of Washington Tacoma
Tacoma, WA 98402 USA
brsivart@uw.edu

John Krumm
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052 USA
jckrumm@microsoft.com

Ankur Teredesai
Center for Web and Data Science
Institute of Technology
University of Washington Tacoma
Tacoma, WA 98402 USA
ankurt@uw.edu

## ABSTRACT
The 20th ACM SIGSPATIAL Conference on Advances in Geographic Information Systems (GIS) was held in November of 2012. In conjunction with this conference, we organized the conference's first competition, called the SIGSPATIAL GIS Cup 2012. The subject of the competition was map matching, which is the problem of correctly matching a sequence of noisy GPS points to roads. We describe the details of the contest, the results of the competition, and the lessons we learned in running a contest like this.

## Categories and Subject Descriptors
D.2.8 [Metrics]

## General Terms
Algorithms, Performance, Theory

## Keywords
Geographic Information Systems, Map-matching, SIGSPATIAL 2012, Competition

## 1. INTRODUCTION
The ACM special interest group SIGSPATIAL [1] addresses issues related to the acquisition, management, and processing of spatially-related information with a focus on algorithmic, geometric, and visual considerations. The scope includes, but is not limited to, GIS. Each year SIGSPATIAL holds a conference, The International Conference on Advances in Geographic Information Systems, to discuss these issues and the current research being done in the field. This year's conference was held in Redondo Beach, California in November of 2012. For the first time ever, key members of SIGSPATIAL hosted a GIS related competition called the SIGSPATIAL GIS Cup 2012. The competition opened in June and ended several weeks before the conference.

This paper discusses the SIGSPATIAL GIS Cup 2012 [2], serving as a record of the contest rules and data, its winners, and the lessons we learned in creating and running a contest like this.



By creating the SIGSPATIAL GIS Cup, we were able to define one specific problem for many different individuals or groups to work on simultaneously. We hoped focusing many people on the exact same problem created a sense of community and encouraged competitors to push themselves, and their solutions, further than they might if researching the problem separately. This sense of competition can result in overall more accurate algorithms and faster progress in the GIS field.

We chose *map matching* to be the topic of this contest. This is the problem of correctly matching a sequence of noisy GPS points to roads. As computing goes more mobile, location becomes increasingly important. For people traveling on roads, knowledge of the past, current and future road is important for both real time location based services and for retrospective analysis of routes. Map matching from GPS data is a necessary step in understanding which road a vehicle was on given the various options for a possible route. For instance, Wang et al. [3] point out map matching as a central problem for researchers analyzing the large amount of GPS data available from Chinese taxi fleets.

The problem of map matching seems simple at first. The naïve solution is to match each GPS point to the nearest road. This fails frequently enough that more sophisticated solutions are necessary as outlined in [3-5].

Map matching remains an active area of research, with no one particular dominating solution.

### 1.1 Similar Competitions
The use of a competition in a specific field of research is not unique to the SIGSPATIAL GIS Cup. One of the most well-known recent competitions was the Netflix Prize [6]. This competition was focused on the area of recommendation systems and provided a $1,000,000 prize to anyone who was able to improve upon Netflix's own movie recommendation system by 10%. Competitors were provided with a training dataset containing over 100 million ratings from over 480 thousand users. They were also provided a qualifying test set of over 2.8 million customer/movie id pairs with the ratings withheld, divided into two disjoint subsets. Root mean square error (RMSE) was used to score the submissions based on the test set. The RMSE for the first test subset was reported publicly on the contest website, while the RMSE score for the second test set was used to determine a winner for the competition. If a competitor was able to improve RMSE by 10%, they were notified by the competition

and requested to submit a description of their algorithm and all source code. This description and the source code were then evaluated by experts in the field to confirm the correctness.

The ACM special interest group SIGKDD has been holding an annual competition for several years, focusing on the area of Knowledge Discovery in Data Mining. The most recent competition, KDD Cup 2011 [7], contained two different competition tracks. Each track was related to recommendation systems and made use of item and rating data from Yahoo! Music. Similar to the Netflix Prize, the data test sets for both tracks were divided into two disjoint, equal sets. Root mean square error (RMSE) was used to score the submissions. The RMSE from the first test set was displayed on a web based scoreboard, while the RMSE from the second test set was used to determine the winning submissions. For each contest track, the competitors with the top three scores were required to submit a manuscript detailing the competitor's algorithm and methods used to generate their submission. These manuscripts were then verified by a team of three experts in the field to confirm the solutions were reasonable. Top competitors were also invited to present their work at the 2011 KDD Conference.

The Trading Agent Competition is another KDD Cup-like competition that focuses on the area of agent-based computational economics. Internet advertising provides a substantial amount of revenue for on-line publishers. To help resolve the problem of quoting an appropriate price for individual ads, publishers have begun to use automated ad auctions. This means that a business that wishes to purchase ad space must now bid against other business for the advertising space. The basic problem for the business is how to quickly determine how much they are willing to bid based on how much revenue they expect the ad to bring in. In this competition, competitors were asked to develop an agent, using techniques such as machine learning, that was able to dynamically purchase on-line "ads" in a manner that generates the highest profit possible. To accomplish this goal, an ad auction server framework was developed to simulate the real world advertising scenario environment [8]. Each competitor developed their own agent and then connected to the specified ad auctions server hosting the competition. Often the hosting server would define a 24-hour time period for the competition. During this time period, the competitor's agent directly competed against other competitor's agents in an attempt to gain more profit than any of the others.

## 1.2 Map Matching

The programming problem for the SIGSPATIAL GIS Cup was that of map matching. The basic task of map matching is, given a data set of a vehicle's GPS trace route and a set of map data, match each GPS coordinate with the road that the vehicle was most likely traveling on at the time of the GPS point [4, 9].

### *1.2.1 Correctness*

When determining the correctness of a map matching algorithm, one must not only consider whether the algorithm was able to match each point to a road, but also consider the overall path of the matched point/road pairs. A solution should be analyzed to ensure it does not match consecutive points in the data onto roads that are logically infeasible in relation to how a vehicle could travel. In the end, a good measure of correctness is to compare the algorithm matched point/road pairs to the actual road traveled by the vehicle. It may also be appropriate to consider some sort of penalty for incorrect matches, based on how inaccurate a specific point/road pair was from the actual vehicle path.

### *1.2.2 Challenges*

When working with GIS data and map matching problems, there are a few common challenges that should be taken into account. One challenge is data sparseness. Data sparseness is related to a GPS device's sampling rate (how often a device records a coordinate). While an algorithm may do well on a GPS data set with a high sampling rate (such as one coordinate per second), the algorithm correctness may degrade substantially as the sampling rate is decreased. A second challenge is noisy data. GPS measurements are inherently noisy, with an error standard deviation of a few meters. There are often outliers in the data as well, attributed to urban canyons or other such terrestrial features. An algorithm's correctness may degrade substantially as the number of inaccurate points increases.

### *1.2.3 Running Time*

Some map matching applications must run in real time, such as in a vehicle navigation system. Even running offline, speed is still important as the amount of recorded GPS data grows. Thus, a good map matching algorithm will be forced to make tradeoffs between speed and guaranteed accuracy. These tradeoffs can be made intelligently by considering only reasonable candidate roads, based on the expected size of GPS errors and the expected behavior of the vehicle's driver.

# 2. SIGSPATIAL GIS Cup 2012 Competition Details

In this section, we present details about the competition. More specifically, we overview the data sets provided by the competition and highlight the wisdoms behind the scoring criteria that we used for evaluation. Additional information can also be found at the competition official website [2].

## 2.1 Datasets

The SIGSPATIAL GIS Cup has put tremendous effort in preparing the following data sets and making them available for future research. We hope that these data sets will help researchers in various institutes with ground truth data.

### *2.1.1 Road Network Information*

The contest provided a simplified version of the road network information for Washington State, USA. The original map data is obtained from Open Street Map (OSM) [10]. The road network information is broken into three separate files:

- *WA_Nodes.txt*: This text file contains the nodes of the road network. The file defines 535,452 nodes, with each row representing a single node represented as an integer node ID and a latitude/longitude coordinate.
- *WA_Edges.txt*: This text file contains the edges of the road network. The file defines 1,283,540 edges, with each row representing a single edge. Each edge is defined by an integer edge ID, its two end nodes from the file above, and the traversal cost in time.
- *WA_EdgeGeometry.txt*: This text files contains the geometry data of each edge in the road network. The edge geometry makes a best attempt to define the polyline of the actual road an edge represents. The file contains 1,283,540 entries, one for each edge in the network, with each entry in a single row. Each edge in this file is defined by an edge ID that corresponds to the edge IDs in the file above. The edges in this file are further defined by a name, type (e.g. motorway, primary, secondary, tertiary), length in meters, and a polyline consisting of latitude/longitude coordinates.

### 2.1.2 Training Data

The SIGSPATIAL GIS Cup provided training samples that include multiple GPS traces from across Washington State. The training data set provided for the SIGSPATIAL GIS Cup contains 20 files: 10 input files and 10 output files. Competitors used these files to train, test and benchmark their algorithms before submitting to the competition.

- Input Files: A single input file contains a GPS trace of an individual trip. Each row of an input file represents a single, time stamped latitude/longitude coordinate measured from an actual GPS receiver in a vehicle. The time stamp gave the number of elapsed seconds since the beginning of the trip.
- Output Files: The output files are provided to allow contestants to test their submissions in terms of correctness. They also serve as an example of the required output file format that is expected of the submitted executable. Each row of an output file represents a single map matched GPS reading giving the time stamp, edge ID, and confidence.

The confidence value is a real number between 0.00 and 1.00 that indicates the confidence of the map matching algorithm about the correctness of the map matched GPS reading. 1.00 means that the algorithm is 100% percent confident that the output result is correct. 0.00 means that the algorithm is totally uncertain about the correctness of its output result. In practice, the confidence value is important because various applications would reason about that value before taking decisions using the map matched result.

## 2.2 Scoring Criteria

The scoring criteria were crafted to (a) evaluate both the accuracy and efficiency of the submissions and to (b) discourage the algorithms from "*guessing*". In real life applications, map matching is performed by GPS devices while the driver is on the road or is batch processed for analysis purposes by the server over huge logs of GPS data. In either case, efficiency in terms of execution time is as crucial as accuracy. Moreover, an incorrect match (due to guessing) can be more damaging than declaring "a no-match found". Hence, we require each map matched point to be associated with a confidence.

The grading formula was

$$\begin{bmatrix}(sum\ of\ confidence\ for\ correct\ lines) - \\ (sum\ of\ confidence\ for\ incorrect\ lines)\end{bmatrix} / (running\ time)$$

The grading algorithm works as follows: If the result of a map matched GPS reading is correct, the participant earns one point weighted by the program's declared confidence about this result. If the result of a map matched GPS reading is incorrect, the participant loses one point, again after being weighted by the confidence value. Considering the confidence value in the grading formula encourages participants to do their best effort in estimating the confidence value. A high confidence value for an incorrect result would result in a higher deduction in the grade. Finally, the grade is weighted by the total execution time of the program to consider both accuracy and efficiency. All test cases contain real world GPS data recorded using a GPS logger at a variable sampling rate. The sampling rate varies between one second and 30 seconds. This variable sampling rate is intended to test the resiliency of the submitted map matching algorithm under sparse GPS traces.

## 3. Submissions Testing Process

At the close of the competition, we received 31 entries. Next we will discuss the process of running and testing these submissions.

## 3.1 Submission Verification and Testing Environment Setup

To participate in the competition, each competitor's submission was required to contain three components:

1. An executable file compiled from their source code. The executable was required to be runnable on the Windows 7 64-bit operating system with the command line parameters specifying the location of the required data set files.
2. The original source code of the submissions. This was required to verify the originality of all submissions.
3. A "Read Me" text file. In this file competitors were asked to specify any special requirements to run their submission, such as required compilers or frameworks.

After verifying that all submissions contained the required components, the next step was to ensure we were able to run all the executable files. As the competitors were able to write their submissions in any language of their choosing, we were required to ensure certain requirements were available on the testing machine. The most common requirements were Java, .NET framework, Python and Matlab Compiler Runtime.

After all required resources were installed on the testing machine, we ran each submission with the map data and a very short input file. If any of the submissions were not able to successfully run, the competitor was notified and given the opportunity to submit a single updated submission.

## 3.2 Datasets

When testing the submissions, we chose to use the same Washington state map data that was previously provided to the competitor.

To test the submissions we evaluated the them using the originally provided 10 input files, as well as several variations of these files. These are outlined next.

The first variation is a simple validation test using a file that contained only 25 GPS points. The small size of the input file allowed us to get an idea of how efficient each submission was in loading the large map data files.

The second variation of these files was to reduce the sampling rates. The original files all have one second sampling rates. To test how the well the submissions would deal with reduced sampling rates, we created 9 new sets of input files from the original input files. Each new set of input files increased the sampling interval incrementally by 1 second. This gave us 10 sets of input files with sampling rates going from 1 to 10 seconds.

While ideally we would have tested with completely different input files, the process of collecting and verifying the correctness of each file proved to be too time intensive to make this feasible.

To create our verification output files, we ran our input files through a map matcher based on an algorithm outlined in a recently developed map matching algorithm [11]. We overlaid the results onto a map. This allowed us to visually follow each trace and confirm or correct any inaccuracies found in the output.

To generate the grading data for all the submissions, we ran them through 12 different test runs:

**Table 1: Performance results for top five entries comparing runtime in milli-seconds, % accuracy and overall score**

| Competitor | Map Load Time* | 10 files, 1s Sampling** | 10 files, 5s Sampling** | 10 files, 10s Sampling** | 100 files, 1s Sampling** |
|---|---|---|---|---|---|
| #1 - "An Efficient Algorithm for Mapping Vehicle Trajectories onto Road Networks" | (2592, 4) | (1646, 97.45%, 8.32) | (1259, 94.25%, 2.03) | (1339, 95.16% 0.97) | (2213, 97.45%, 61.90) |
| #2 - "Quick Map Matching Using Multi-Core CPU" | (1683, 2) | (2080, 98.79%, 6.77) | (1823, 98.75%. 1.54) | (1885, 98.54%, 0.74) | (5135, 98.79%, 27.43) |
| #3 - "Effective Map-matching on the Most Simplified Road Network" | (1881, 3) | (2053, 95.54%, 6.40) | (1911, 84.53%, 1.04) | (1983, 87.83%, 0.55) | (4255, 95.54%, 30.90) |
| #4 - "Fast Viterbi Map Matching with Tunable Weight Functions" | (8282, 8) | (6912, 98.37%, 2.02) | (5174, 98.33%, 0.54) | (5447, 98.89%, 0.25) | (14563, 98.37%, 9.59) |
| #5 - "Concurrent Topological Map Matching" | (17231, 11) | (18877, 98.28%, 0.73) | (14384, 95.74%, 0.18) | (14279, 97.37%, 0.09) | (39095, 98.28%, 3.56) |

* (runtime (ms), rank) ** (runtime (ms), % correct, score)

1. 1 file, containing only 25 lines
2. 10 runs with 10 files (files for each run used a different sampling rate between 1 and 10 seconds)
3. 100 files with 1 second sampling

Only submissions which were able to complete all 12 test runs were considered in the final grading.

## 3.3 Results

After completing all 12 test runs, we carefully examined the results and selected the top-5 winning submissions. The following aspects were taken into account when determining the winning entries. Table 1 summarizes a sampling of the test run data for each winning competitor. The column "Map Load Time" represents the running time for the first test run (1 file, containing 25 lines). By only having a single small file for matching, this running time is effectively used to show how quickly each competitor was able to load and process the map data files. The rank in this column is the overall rank of the competitor in the full set of all 31 entries. The final four columns summarize the running time, percentage correct and overall score for each competitor across several different test runs.

# 4. Lessons Learned

One difficulty in the format of our competition was requiring competitors to submit an executable file instead of simply having them submit their output files (as other competitions have done). A large number of the submissions either had issues initially running, or were not able to complete all the test runs without crashing or hanging. The submissions that failed to run often did not output any details as to what the issue was, which made it difficult to report back to the competitor what they needed to fix in their single opportunity to submit an updated submission.

We also found, that by emphasizing execution time as a grading criterion, some competitors spent much effort reducing their program's time to read and parse the input files, which was not central to the problem of map matching. More related, but still not central, were competitors' attempts to reduce the time to find roads near the test GPS points with spatial indexing schemes.

# 5. Conclusion

Over the course of this contest we highlighted the need for developing effective solutions to interesting geospatial challenges. As the pioneering competition in this domain we focused our attention on a very basic yet critical problem faced by a variety of navigation systems today, viz. map matching. We created a competition infrastructure from the ground up, so various teams could submit solutions, and developed a robust test harness that can grade various map matching solutions to the ground truth and evaluate performance along various dimensions such as correctness, speed and sampling gap. We envisioned the competition idea, wrote the specifications, collected/formatted and verified the data, developed a website in addition to the contest grading software system. In a short span of 2 months we received over 30 submissions. It is our hope that this competition will help foster the growth of the GIS community and that future SIGSPATIAL GIS Cups will continue this tradition and grow in scope and participation.

# 6. Acknowledgements

We thank nVIDIA and Microsoft for their generous support and acknowledge the support of ACM SIGSPATIAL Executive Committee and the 20th SIGSPATIAL Conference Organizing Committee.

# 7. REFERENCES

1. ACM SIGSPATIAL. Available from: http://www.sigspatial.org/.
2. ACM SIGSPATIAL Cup 2012. Available from: http://depts.washington.edu/giscup/home.
3. Wang, Y., et al., Challenges and Opportunities in Exploiting Large-Scale GPS Probe Data. 2011, HP Laboratories.
4. Bernstein, D. and A. Kornhauser, An Introduction to Map Matching for Personal Navigation Assistants. 1998, New Jersey Institute of Technology, Newark: Newark, NJ.
5. Quddus, M.A., W.Y. Ochieng, and R.B. Noland, Current Map Matching Algorithms for Transport Applications. Transportation Research Part C: Emerging Technologies, 2007. 15(5): p. 312-328.
6. Netflix Prize. Available from: http://www.netflixprize.com/.
7. KDD Cup. Available from: http://www.kdd.org.
8. TAC/Ad Auctions. Available from: http://aa.tradingagents.org/.
9. White, C.E., D. Bernstein, and A.L. Kornhauser, Some Map Matching Algorithms for Personal Navigation Assistants. Transportation Research Part C: Emerging Technologies, 2000. 8(1-6): p. 91-108.
10. OpenStreetMap. Available from: http://www.openstreetmap.org/.
11. Newson, P. and J. Krumm, Hidden Markov Map Matching Through Noise and Sparseness, in 17th ACM SIGSPATIAL Conference on Advances in Geographic Information Systems (GIS '09). 2009: Seattle, WA USA. p. 336-34