Master's Project Summary Paper: ACM SIGSPATIAL GIS Cup 2012

Travis Rautman
Institute of Technology
University of Washington
Tacoma, WA USA
Committee Chair: Prof. Ankur Teredesai

June 1, 2012

Abstract

The 20th ACM SIGSPATIAL Conference on Advances in Geographic Information Systems will be held in November of 2012. In conjunction with this conference, we have organized the first ever competition called the GIS Cup 2012. The goal of this project was to (a) facilitate the contest by gathering the requirements of this competition and (b) develop a contest framework that will execute and score competitor submissions. The main problem under consideration is termed as the map matching problem where given a trace route of geo-location information (GPS trace) and a map consisting of all possible routes, the task is to develop approximate algorithms that will predict which routes are the most likely given inaccuracies in the GPS trace.

1 Introduction

A Geographic Information System (GIS) is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographically referenced data. The data may come from a variety of sources, including the longitude and latitude values commonly obtained from Global Positioning System (GPS) devices [2].

The ACM special interest group SIGSPATIAL [5] 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 will be held in Redondo Beach, California in November of 2012. For the first time ever, key members of SIGSPATIAL are proposing to host a GIS related competition called the GIS Cup 2012. The competition will be open in June and end several weeks before the conference.

Two key members of the SIGSPATIAL community will be members of the GIS Cup committee.

John Krumm is a senior researcher at Microsoft Research in Redmond. John has been highly involved as a co-chair for the ACM GIS for the last few years. John has a made significant contributions and extensive publications in the field, especially in the map matching domain.

Mohamed Ali is also a researcher at Microsoft, with a focus on spatial and temporal data management.

2 Project Planning

During the first quarter of this project, the GIS Cup committee met weekly to discuss issues and assign work items to each member. Items of discussion covered a wide range of topics such as data formats, allowed team sizes, deadlines and scoring algorithms. After coming to a consensus on all topics, we made the details of the competition public on the official website (HTTP://depts.washington.edu/giscup/). The website is powered by Drupal and is hosted on University of Washington servers.

In addition to finalizing the details of the competition, we also had to discuss technical details such how to obtain map data for the competition, verifying the training data and how the grading software should function.

This paper will attempt to summarize and document all decisions made by the GIS Cup committee to make this competition a reality.

3 Contest Details

Below we will discuss the details of the competition.

3.1 Map Matching

The main problem that will be focused on for the GIS Cup is that of map matching.

3.1.1 Problem Definition

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 [1, 7].

3.2 Datasets

Two different types of data have been provided to the competitors.

3.2.1 Map Data

The first set of data is map data for Washington state. The original map data was obtained from Open Street Map (OSM) [6] and put into a text file using an easy to understand format. After analyzing the data, we determined that several of the fields produced by OSM were not necessary for our competition. To reduce the chance of confusion, we removed these fields.

The map data was 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. Each row contains three values, with each value being separated by a single space. The form of a node row is: <NodeId><lat><long>
  - <NodeId>: An integer value specifying the unique identification number of the node within the road network.
  - <lat>: This value specifies the latitudinal location of the node within the road network in degrees.
  - <long>: This value specifies the longitudinal location of the node within the road network in degrees.

- 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 row contains four values, with each value being separated by a single space. The form of an edge row is: <EdgeId><from><to><cost>
  - <EdgeId>: An integer value specifying the unique identification number of the edge within the road network.
  - <from>: This value represents id of the node that is at the head of the edge. If the edge is defined as (v,w), <from>is v. These node id values correspond to the <NodeId>values in WA_Nodes.txt.
  - <to>: This value represents id of the node that is at the tail of the edge. If the edge is defined as (v,w), <to>is w. These node id values correspond to the <NodeId>values in WA_Nodes.txt.
  - <cost>: This value defines the actual cost of a vehicle to traverse from one end of the edge to the other end. It is a cost function based on length of the edge and the speed limit on the road segment the edge represents.
- 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 that the edge is representing. The file contains 1,283,540 entries, one for each edge in the network, with each entry in a single row. Each row contains a minimum of eight values, with each value being separated by a caret (ˆ). Each entry defines n different points along the edge by specifying the point's latitude and longitude values. There will be more than eight values in a single entry if the entry contains longitude/latitude information about more than just the first and last points of the edge. The form of an edge geometry row is:<EdgeId>ˆ <Name >ˆ <Type>ˆ <Length>ˆ <Lat_1>ˆ <Lon_1>ˆ ... ˆ <Lat_n>ˆ <Lon_n>
  - <EdgeId>: An integer value specifying the unique identification number of the edge within the road network. This value will match a single edge defined in the WA_Edges.txt file.
  - <Name>: This value describes the real-world name of the road segment that this specific edge represents. If no name is defined, the attribute will contain an empty string.

  - <Type>: This value describes the type of road that is represented by the edge. Some common values are:
  - <Length>: This value is the length, in meters, of the edge.
  - <Lat_1>: This value is the latitude of the first point of the edge. If the edge is defined as (v,w), <Lat_1>is the latitude value of v.
  - <Lon_1>: This value is the longitude of the first point of the edge. If the edge is defined as (v,w), <Lon_1>is the longitude value of v.
  - .... <Lat_i><Lon_i>....: The latitude and longitude values for several points between the first and the last points of the edge. These points are optional and the number of optional points varies according to the geometry of the represented edge.
  - <Lat_n>: This value is the latitude of the last point of the edge. If the edge is defined as (v,w), <Lat_n>is the latitude value of w.
  - <Lon_n>: This value is the longitude of the last point of the edge. If the edge is defined as (v,w), <Lon_n>is the longitude value of w.

3.2.2 Training Data

The second set of data is training samples of GPS traces from across Washington state. The training data set provided for the GIS Cup contains 20 files: 10 input files and 10 output files. Competitors will use these files to test their algorithms before submitting to the competition.

- Input Files: A single input file contains a GPS trace route of an individual trip. Each row of an input file represents a single GPS reading in the form of: <Time>,<Latitude>,<Longitude>
  - <Time>: Represents the number of seconds since the start of the trip.
  - <Latitude>: Represents latitudinal location of the GPS reading in degrees.
  - <Longitude>: Represents longitudinal location of the GPS reading in degrees.

- Output Files: The output files are provided to allow contestants to train and test their submissions. 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 in the form of: <Time>,<EdgeId>,<Confidence>
  - <Time>: Represents time of the original GPS reading as given in the input file.
  - <EdgeId>: The identifier of the edge that the GPS reading matches to. Note that value of <EdgeId>must be one of the <EdgeId>values in the WA_Edges.txt and WA_EdgeGeometry.txt files
  - <Confidence>: 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. 0.70 (as an example) means that the algorithm is 70% confident that output reading is correct. In practice, the confidence value is important because various application would reason about that value before taking decisions using the map matched result.

3.3 User Submissions

Each participant is expected to submit:

1. A single .zip file that contains the original Source code and any dependencies. A readme.txt file should be included for any special instructions on how to compile the submitted code. Submission of the source code is mandatory to ensure originality of the submitted work.

2. A single executable file named: "mapmatch.exe". The mapmatch.exe accepts three command line parameters. The usage of the mapmatch.exe program is as follows: mapmatch [RoadNetworkInfo_Path] [Input_Path] [Output_Path]

   - RoadNetworkInfo_Path: Specifies the directory that contains three text files (WA_Nodes.txt, WA_Edges.txt and WA_EdgeGeometry.txt).

These files contain the road network of Washington State in the U.S. The format of these files is detailed under the Road Network Information section.

- Input_Path: Specifies the directory that contains n text files. Each file is a single test case and contains a series of recorded GPS readings recorded in Washington State in the U.S. The files are named as: input_01.txt to input_n.txt, where input_i.txt is the file that contains the ith test case. Examples of input test cases can be found in the Training Data Sets section.
- Output_Path: Specifies the directory where the program is expected to place the output files. For each input file "input_i.txt" the mapmatch.exe program has to generate a corresponding output file named "output_i.txt".

To accept competitor submissions, we decided to use Microsoft's Academic Conference Management Service [4]. The Conference Management Toolkit (CMT) is a free conference management service sponsored by Microsoft Research. While CMT is capable of handling a complex work flow of an academic conference, our use case is a bit more simple. We have used CMT to designate each member of the GIS Cup committee as a conference chair, so that we each have access to user submissions. Each competitor is asked to create an account in CMT and then follow the instructions on how to submit a competition entry. The original intended purpose of this tool was to accept conference paper submissions. This meant that we had to adjust a few of the default settings to allow competitors to submit the required .zip files.

We began accepting submissions at the end of May and will continue to accept them through August 1st.

3.4 Submission Scoring Criteria

By comparing the output files of competitors submission to our hidden solution files, we will be able to calculate a score for each submission.

The basic grading algorithm is 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 ensure that the runtime across each submission can be compared, we will ensure that all submissions are graded in the same testing environment with no unessential background processes running.

4 Design of the Grading System

Before any development grading system began, time was spent designing how the system should function. Initial concepts of the system's required entities were sketched on paper. After a rough design was completed, the concept was transformed into a class diagram using Visio. This diagram is useful in gaining an understanding of how the different classes in the grading system interact with each other.

5 Implementation Details

After the initial design was completed and documented. It was time to begin development.

The system was developed using Visual Studio 2010 and written in C#. A basic Windows Form GUI was created to allow the user to easily choose the directories that contained required files to grade all submissions. As each competitor's entry will be named "mapmatch.exe", each executable file must be in its own file to avoid duplicate file names in the same directory. When choosing the folder containing competitor submissions, the software will confirm the directory structure is correct.

After the required directories are selected, the grading can begin. The basic steps of grading are as follows:

1. Load all solution files into the proper data structures

2. Create competitor objects to represent each submission

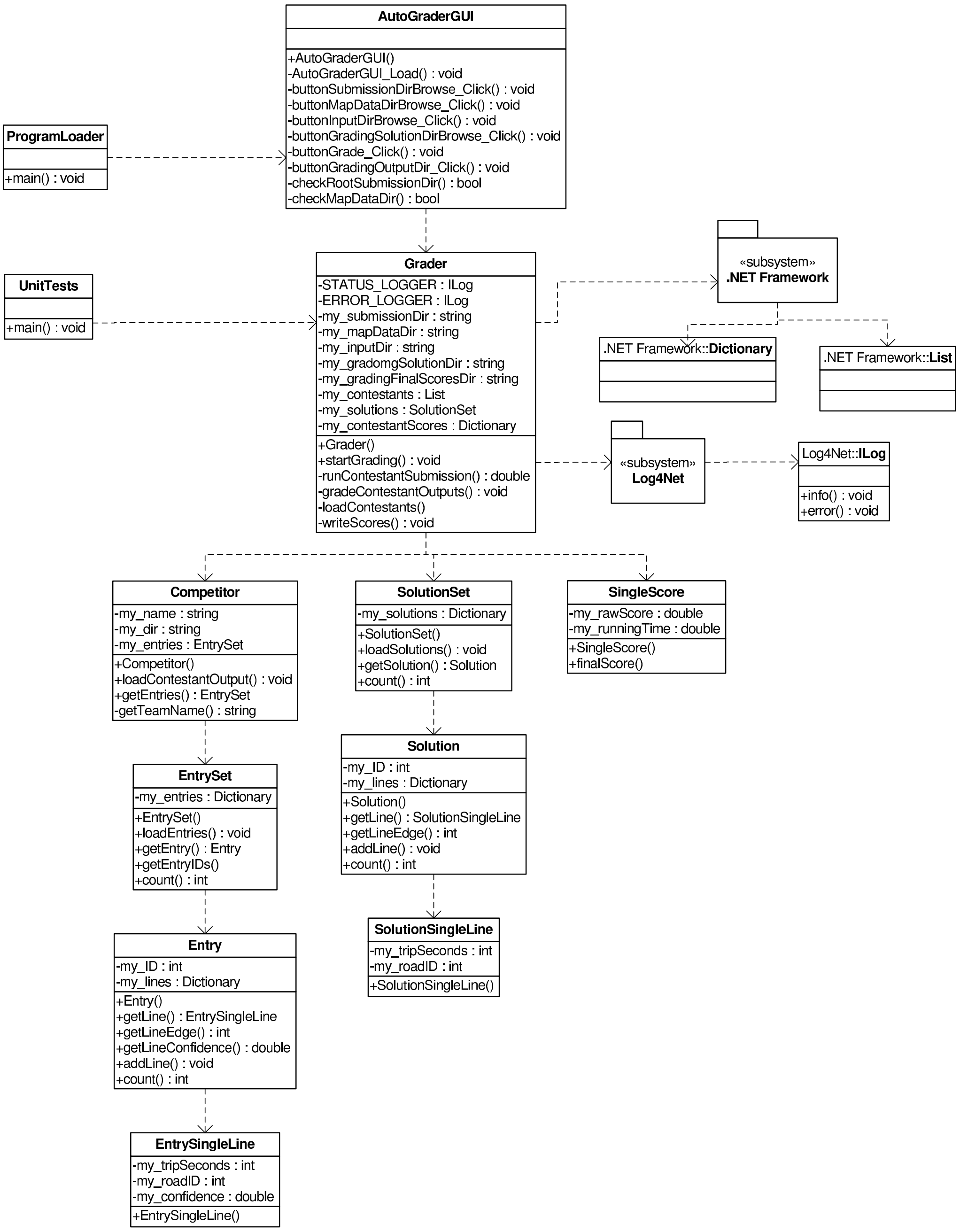

Figure 1: Grading System Class Diagram

3. Individually, and sequentially, run each competitor submission as a separate process. Directories specified earlier in the GUI are used to pass the required command line parameters to each executable submission.

4. Wait for the executable to exit, while keeping note of the program's running time. If desired, the individual running the grading software may update the app.config file of the software with a timeout value so that each submission is not allowed to run indefinitely. If a timeout value is specified, and the submission exceeds it, the executable's process is killed and the submission will not receive a grade.

5. After the executable has exited, the output files are loaded into the proper data structures.

6. When the output files are properly loaded, the actual grading can begin. Each output file is compared against it's corresponding solution file, and the scoring algorithm described above is used to determine a score.

7. Each competitor's final score is recorded into a data structure and after all scoring is complete, the results are sorted and written into a text file.

To assist with trouble shooting any issues that may occur in the submissions, as well as record the ongoing status of the grading system while it is running, we made use of the logging framework Log4Net [3]. Log4Net allowed us to record the progress of the grading system as it loaded solutions, ran submission and graded outputs. It also allowed us to record error is such cases as the submission not running as expected or the output containing unexpected values.

6 Independent Study

To help us test out our grading system, we organized a GIS Cup independent study during the second half of the project (Spring quarter of 2012). While much interest was initially shown by several students, only one ended up registering for the independent study. However, this allowed us to work very closely with the student throughout the quarter. We initially spent time discussing the map matching problem and several papers that discussed solutions to the problem. After we felt the student had a good understanding of the problem, we let him decide which solution(s) he would like to implement. The student determined that he would initially develop a baseline submission, that did not make a very strong attempt at solving the map matching problem, but would then iteratively update this solution. Along with submitting several iterations of his solution, the student also agreed to submit a small number of solutions that were intended solely to test the grading system and make sure it would not break when running actual competition submissions.

7 Final Status

As of June 2012, the competition is currently active and submissions are being accepted. While no official entries have been submitted yet, we have received several emails from interested participants with questions about the competition. Additionally, 51 competitors have subscribed to our ”GIS Cup News” email list.

In November, we will present the results of our competition at the 20th ACM SIGSPATIAL Conference on Advances in Geographic Information Systems. The top three competitors will also be allowed present their solutions in a special conference workshop and display posters in the poster session.

Initial development and testing of the grading system was completed in May. The submissions received from the independent study student have successfully been run by the system and graded accordingly.

8 Summary

Over the course of this project we have created a world-class software development competition from the ground up. We envisioned competition ideas, wrote specifications, collected/formated and verified data, developed a website and designed/engineered a robust grading software system. It has been an exciting process and we are eager to begin receiving submissions and present the competition winners in the fall.

It is our hope that this competition will help to build up the GIS community and that future GIS Cups will continue to grow in scope and participation.

References

[1] David Bernstein and Alain Kornhauser. An introduction to map matching for personal navigation assistants, 1996.

[2] Kenneth Foote and Margaret Lynch. Geographic information systems as an integrating technology: Context, concepts, and definitions, 2000. http://www.colorado.edu/geography/gcraft/notes/intro/intro.html.

[3] Apache Software Foundation. Apache log4net website. http://logging.apache.org/log4net/.

[4] Microsoft Research. Microsoft's academic conference management service website. http://cmt.research.microsoft.com/cmt/.

[5] SIGSPATIAL. Sigspatial official acm website. http://www.sigspatial.org/.

[6] OpenStreetMap United States. Open street map us official website. http://www.openstreetmap.us/.

[7] Christopher White, David Bernstein, and Alain Kornhauser. Some map matching algorithms for personal navigation assistants, 2000.