Preliminary Concepts for a long−range Mars Rover Navigation System Prototype based on a Mars Global Terrain Database

Research Report No: 2001−695−18

James Bradley
Department of Computer Science
University of Calgary,
Calgary, Alberta, Canada

Abstract. A recent paper proposed a detailed structure for a Mars Global Terrain Database (MGTDB), for use in research on Mars, and populated with data from the Mars Global Surveyor satellite, in polar orbit around Mars. The database design allowed for 16.5 meters resolution or better everywhere, at a cost of some 2,000 Gigabytes of data storage.

An MGTDB also has another use, not envisaged in the original paper. It can be used as the key component in a Mars *rover navigation system*, to enable long−range rover navigation over the Martian terrain, and so obviate the lack of magnetic poles on Mars as a basis for navigation.

This preliminary paper sets forth the basic architecture, as well as the basic hardware and software concepts, for such a rover navigation system. It also points out how a useful research prototype might be constructed in the absence of both a finished MGTDB and a viable long−range Mars rover.

The paper also shows that some basic computer−science research would likely be needed, to determine two quite different optimum search algorithms. These algorithms are (a) a *lost−search* algorithm, which would be needed as a basis for a lost−search subsystem that could determine the rover's position when lost, and (b) a *route−search* algorithm, which would be needed as a basis for a route−search subsystem that could generate the optimum route between any two locations on the planet.

If a future satellite−based Global Positioning System (GPS) for Mars, like that in use in Earth, were ever to render the lost−search subsystem obsolete, it would still have value as a risk−preventive resource.

Key Words: algorithm, computer, graphics, Mars, Mars Global Terrain Database, navigation, rover, search, system, terrain, voice−activated system

Introduction

In a recent paper [1], the author proposed the construction of a Mars global terrain database. The structure proposed for this database was based on a relation Terrain, which contains the basic terrain data from Mars. This terrain data is obtained from the Mars Global Surveyor satellite in polar orbit around Mars [5], and engaged in a detailed survey of the surface terrain, using a laser altimeter to measure surface elevations [12]. The original justification for the database was use by scientists on Earth for exploration of the planet.

Now, however, a quite different and additional use has emerged, namely the incorporation of a MGTDB as the key component of a long−range Mars rover [4, 14] navigation system. A reliable navigation system is a problem for a long range−rover on Mars, since Mars has no magnetic poles, preventing the use of a compass [5]. However, an MGTDB−based navigation system allows accurate navigation to any location on the planet. Before we review the possibilities here we need to briefly review the basic MGTDB design.

1. Summary of the MGTDB design

The MGTDB relational [6] database design uses an arc raster grid [3], and allows for both surface−feature data and terrain data over the entire surface of Mars [13]. At the core of the database is a 2000−Gigabyte relation called Terrain that contains the terrain data. The database has a surface resolution everywhere, including at the poles, of 16.5 meters or better.

The tuple structure of the relation Terrain

Data in the Terrain relation is recorded with respect to approximately 1−second of arc almost square grid elements. However, the terrain data within such a square−second grid does not constitute a tuple of Terrain.

Each tuple of Terrain actually contains two sets of data: First, the latitude/longitude coordinates of a 1−degree curved rectangle. Second, data for each of the 1−second square grids making up a N−S running column of 900 of these grid elements. Such a 900−grid element column is thus a quarter degree of latitude long, lying within the 1−degree rectangle. It is called a N−S column grid.

Thus for each 1−degree curved rectangle there are many Terrain tuples each denoting a N−S column grid.

In the N−S direction there are four 900−grid−element N−S column grids per degree of latitude and thus four tuples. The number of tuples in the E−W direction, per degree of longitude, and thus the number of N−S column grids, varies from 3600 per degree of longitude at the equator to 15 at the poles.

Within any N−S column grid each of the 900 grid elements has a N−S side of exactly 1 second of arc, and the eastern and western sides of a N−S column grid are meridians and intersect the poles. Also, within any N–S column grid, the E–W side length of each grid element will normally vary from 1.0 down to 0.98 seconds of arc for latitudes less than 78, but from 1.0 down to 0 for the column grid running from latitude 89.75 to 90.

This arrangement ensures an almost square grid element, at least to within 2%, for latitudes less than 78. Since 1 second of arc is close to 16.5 meters, it also ensures the resolution of 16.5 meters or better over the entire planet. (A N–S column grid can easily be visualized as equivalent to a stretch of highway in a N–S direction just under 15 km long
(or 0.25 latitude degrees) and about 16.5 meters wide, with sides that are meridians, converging slightly in the poleward direction, but completely convergent in the case of a column grid that ends at a pole.)

The relation structure in the MGTDB

The remainder of the database has to do with relating Mars feature–type relations to Terrain. It must always be remembered that the core relation Terrain is structured so that there is a set of tuples per 1–degree latitude/longitude rectangle, the number in the set falling with increasing latitude. (However, this is essentially transparent to users writing SQL expressions [6] to retrieve Terrain data on the basis of specific features, even though there are many different types of features, each requiring a distinct relation.)

Since all features have some attributes in common, each feature–type relation is in a 1:1 ISA relationship with a relation Feature that contains attributes common to all feature types. It is Feature that is linked to Terrain via a simple relationship relation FT. FT is estimated to be of the order of 1.0 Mbyte in size, since it is likely to have a number of tuples comparable to the number of 1–degree curved rectangles on a sphere.

In contrast, the number of tuples in Feature is equal to the number of distinct feature instances named on Mars, e.g. Ares Valles, Pavonis Mons, etc., and is likely to grow with time [5, 13]. As a result of this design, there are no Feature–type relationship attributes in Terrain, so that it is possible to add additional feature–type relations in a modular manner. Since feature–type relations, such as Crater, Valley, Mountain, and so on, are each in 1:1 ISA relationship with Feature, a feature–type tuple inherits all the properties of the Feature supertype, including relationship participation.

The Feature/FT relations can be viewed as a relationship "bus" from Terrain, to which it is possible to attach any additional feature–type relation, regardless of any additional complexity due to recursive relationship participation.

2. Concepts for a Rover Navigation system based on a MGTDB

In this section, we give a preliminary conceptual outline of the system hardware, the system architecture, as seen by a user, and the system software modules required. We begin with the hardware.

Hardware requirements.

A massively miniaturized, high capacity data store unit, capable of holding 2,000 gigabytes of data is obviously needed.

Currently there may be some problems with storage space in a mobile vehicle for a 2000–gigabyte MGTDB. However, with the prototype, as explained below, only terrain data for a limited part of the Earth's surface would be stored. By the time a manned Mars mission takes place (in the Mars opposition year and sunspot minimum year of 2016 at the earliest) it is unlikely that storage capability for a 2,000–gigabytes database will represent any problem.

A voice sensor system is required, so that a driver can issue voice commands to the navigation system's command interpreter [11].

In addition, two video screens are needed on the dash behind the steering wheel––a map screen and a perspective screen. Furthermore, a cursor control unit needs to be positioned on the steering wheel, to enable a cursor arrow to be easily moved in the X and Y directions (E–W and N–S) directions, and to be rotated easily, for pointing in any direction.

Basic system features

To get some understand of system architecture, that is, how the system would work, as seen by a user, let us imagine the system being operated on a rover expedition.

Before departure over unexplored terrain, the driver (or navigator) must inform the navigation system where on the planet the rover is currently located. It is assumed that the driver knows that (but if not, the navigation system can solve that problem with the lost–search subsystem––more about that later).

The driver begins by simply voicing the latitude and longitude coordinates for the rover location. The voice–activated command interpreter uses this information to generate a retrieval request directed at the Terrain relation [10, 11]. This request, executed by the database system, retrieves the Terrain data for a degree square about the coordinates input by the driver.

The retrieved terrain data is displayed as a map, seen from above, on one of the two display screens––*the map screen*. On the map screen a cursor also appears, dimmed. The driver then manipulates the cursor using the steering wheel cursor control, to place the cursor at the rover's location, and pointed in the direction of travel, at normal intensity. This causes the navigation system to generate a perspective view of the terrain, seen from this location, looking in that direction. This perspective view appears on the other screen––*the perspective screen*.

If the cursor is at the correct spot on the map screen, and is pointing in the direction of travel of the rover, then the perspective on the perspective screen should exactly match the driver's view of the terrain ahead, through the front rover window. If the match is not exact, the driver can experiment with moving the cursor until the match is exact.

As the rover moves forward, the driver can head for features marked on the map screen, adjusting the cursor position, every so often, to generate a new perspective of the terrain ahead. In this way the driver can always be sure of the rover's position, with respect to the map on the map screen. This will enable the driver to navigate successfully to any location on the planet.

It the driver, by chance, is forgetful of the perspective screen, and finds that the rover is in a location that does not at all match the perspective on the perspective screen (the "lost" condition, where the driver is lost), the navigation system has a subsystem (based on a *lost−search* algorithm) to determine the rover's location.

The driver would voice the coordinates of the last known location of the rover, or an approximate location based both measurement of local time versus Airy−0 time [8, 9] This will likely be not more than a few degrees (a degree on Mars is 60 km.) from where the rover is actually located. When a Lost−search command is entered by the lost driver, the system will read in the actual perspective view through the front window, as currently recorded by a digital camera on the rover [7], and pointed ahead. It will then search the terrain database within a few degrees of the location input, until it finds a match. When this match is found, the perspective appears on the perspective screen, matching the view through the front window, and the corrected map appears in the map screen with the cursor sitting at the rover's location, and pointed in the direction ahead.

Advanced systems features−−route determination

There are no roads on Mars, and the surface is covered with dangerous precipices, crevasses, mountains, escarpments, craters and more. So to get from A to B, a route that circumvents impassible terrain must be worked out.

This route could be worked out manually by the rover driver, using the terrain database as a source of terrain information. Alternatively, the navigation system, if endowed with sufficient intelligence, could determine the optimum route. This feature would work as follows.

The driver vocally enters the coordinates for both A and B. The system will use a file of impossible route characteristics, and a file of undesirable route characteristics, to determine the route. Before the route search begins, the driver may enter some additional impossible and undesirable route characteristics records, or even delete some existing undesirable route characteristics records, since what is undesirable with one trip may be acceptable with another.

An example of an impossible route characteristics record would be a precipice, or a crevasse. An example of an undesirable route characteristics record might be a gradient exceeding one in five for more than 50 meters, and so on. The driver may also enter route search goal characteristics, such as shortest route, or route with least total ascent, or least total descent, and so on. The default would likely be the shortest route.

The system would now undertake a search of the terrain data base using the A and B coordinates, and the impossibility, undesirability and goal data, generate one or more routes that satisfy the requirements, and select the optimum according to the goal requirements.

Software requirements

Many software components are needed to build this system. Beginning with the basic subsystems, we need the MGTDB, and the database system software to manage it. Obviously also, we need voice activation software for the command interpreter, for accepting voice commands and converting them to navigation system commands.

Then we need map generation software capable of generating a map of the Mars terrain, of a specific scope, at a specific resolution, at specific latitude–longitude coordinates, on input of scope data and location coordinates to the system, as well as resolution requirements

We also need perspective generation software capable of generating a ground–level perspective of the Mars terrain at specific latitude–longitude coordinates and in a specific direction, on input of location and direction coordinates to the system (from the cursor control unit).

We obviously also need the software to accept input data from the steering wheel cursor control unit, and convert it to systems commands.

The lost–search software subsystem could be a problem, since nothing like it is likely to have been developed before. At this point, it is not clear exactly what the algorithm behind it would have to be, but since search optimization is involved, it is likely that it would require a research project, prior to its construction, to determine the appropriate lost–search algorithm.

Finally, we would need a route–search software subsystem This subsystem is likely to be the most sophisticated of all of the subsystems. How this system would work in detail can at present be only guessed at. Obviously, it would have to generate various routes, check them for compliance with impossibility and undesirability requirements, and then select on the basic of optimum compliance. However, the number of possible routes involved in any route search is likely to be very large, and the route generation time for any route is likely to be long. Thus some method of reducing the number of routes generated would be needed. Clearly, a research project would be needed to determine an optimum route generation algorithm on which to base this subsystem.

3.0 Possible Prototype Construction

Fortunately, we do not actually need a working MGTDB or a Mars rover to build a prototype of the Mars rover navigation system. As mentioned, a manned mission to Mars is not likely before 2016, so that there is time to do the basic research and development work prior to completing construction of these two component prototypes.

A terrain database of some uneven part of the Earth's terrain, preferably terrain without much vegetation, such as arctic, ant–arctic or desert regions, would be suitable for a prototype. Instead of a Mars rover, with all of its attendant engineering problems––pressurization, temperature, rough–terrain suspension, and power problems––an all–terrain military vehicle could be used to try out the system on Earth.

Research and development team

No one person could possibly have all the skills needed to construct this system. At least one of each of the specialists listed below would be needed.

A computer engineer would be needed for the hardware components, a software engineer for the overall software system, a database specialist for the database and database system components, a graphics specialist for map and perspective displays, and an algorithm specialist to develop the lost–search and route–search algorithms.

4.0 Conclusion

A Mars rover navigation system for the surface of Mars is a possibility. In fact it is difficult to see how long–range rover navigation on the surface of Mars, given the lack of a road system, would be possible without such a rover navigation system.

The rover navigation system, in its essentials, would consist of a massively miniaturized MGTDB, a map screen and a perspective screen on the dash, and a cursor control on the steering wheel. The navigation system would generate a map, on the map screen, of the terrain in which the rover is located. If the driver positions the cursor on the map screen to match the rover's position, and points the cursor in the direction in which the rover is headed, the system would generate a perspective of the terrain ahead. This should match the perspective seen by the driver through the front window. Period use of this system, say every 10 miles or so, would enable the driver to navigate accurately on the planet.

The system could undertake a lost search, when the driver is lost, to find a perspective matching that sensed by an image sensor at the front of the rover, and so determine the rover's position. It could also use a route–search feature, to determine the best route from A to B, thus avoiding impassible terrain.

The only factor that could alter the requirements for a rover navigation system, as laid out in this preliminary paper, would be the installation of a Mars navigational satellite system, to enable a Mars Global Positioning System, like that in use on Earth, to become a reality.

A rover equipped with a GPS receiver would not need to undertake a lost search, for example, since rover location could always be determined from the positions of three GPS satellites, using the standard triangulation procedure. However, the other features of the system described here would still be very useful for navigation. And even if a GPS were available one day for Mars, making the lost search feature obsolete, retaining a lost search feature in the navigation system would still be useful as a risk−preventive resource [2], in the event of a GPS satellite failure.

References

1. Bradley, J. An efficient modularized database structure for a high resolution column−gridded Mars Global Terrain Database, Software−Practice and Experience 29(5), 1999, pp437−456

2. Bradley, J. *Elimination of Risk in Systems*, Tharsis Professional Books, 2002.

3. Burrough, P. A., McDonald R., *Principles of Geographic Information Systems*, Oxford University Press, England.

4. Bresina J., et al. Increased Flexibility and Robustness of Mars Rovers, Proceedings of the 5th International Symposium on AI, Robotics, and Automation in Space, June, 1999.

5. Cattermole, P. J. *Mars*, Oxford University Press, 2001

6. Date, C. J., *Database Systems*, 7th Edition, Addison Wesley, 2000.

7. Eisenman, A., et al. Mars Exploration Rover Engineering Cameras. In SPIE Remote Sensing, September 2001, Toulouse, France.

8. Gangale, T., Martian Standard Time, J. of Interplanetary Society, June 1986

9. MacKenzie, B., Metric Time for Mars, Proc. 3rd Case for Mars Conference, University of Colorado, Vol 75, 1987, p559.

10. Sauer, C. H., and Chandy, K. M., *Computer System Performance Modeling*, Prentice−Hall, 1981.

11. Silberschatz, A, Peterson, J. L., *Operating Systems Concepts*, 6th Ed, Addison Wesley,

12. Smith, J. R., *Introduction to Geodesy*, Wiley, 1997.

13. USGS, *Atlas of Mars,* Version 1.0.9, USGS Information Service, Denver Co., 1995

14. Washington, R., et al. Autonomous Rovers for Mars Exploration, Proceedings of the 1999 IEEE Aerospace Conference, Aspen, CO, 1999.