Automated Process for Generating Digitised Maps through GPS Data Compression

Stewart Worrall and Eduardo Nebot
University of Sydney, Australia
{s.worrall, e.nebot}@acfr.usyd.edu.au

Abstract

This paper describes a robust method for extracting a compressed, digitised road map using GPS data. This is a useful tool for many applications, both autonomous and non-autonomous, such as localisation, prediction, task planning and collision avoidance. The motivation behind this application is to provide a small representation of the digital map so that it can be used in low power, low memory processors. It is necessary that the map is small so that the search space for finding locations is reduced, and consequently the time to search will be also reduced. An additional constraint is that the roads are not marked well, meaning that there are no lanes marked and the edges of the roads are not defined. Results are shown from a large mine, with data collected from a fleet of mining vehicles working over several days.

1 Introduction

The combination of a digitised road map, GPS, wireless communication and an onboard computer can be very useful for many vehicle automation, safety and productivity applications. This paper describes an automated method for producing a digitised road map that is highly compressed. Once generated, this map can be distributed amongst a fleet of vehicles. The current applications for this system include collision avoidance, fleet management, data collection, vehicle scheduling amongst others.

The specifications for this system include utilising a low power, low memory (inexpensive) on-board computer. The maps must be automatically distributed to the vehicles, allowing for updates when new roads are created. This is important especially for mining applications as there can be many additions and changes each year with the roads, and there can be a large number of vehicles that each require the maps to be kept up to date. This paper presents an example in a mining environment, though it can be used in other closed areas (such as ports, quarries, railways, etc).

The first section of this paper introduces existing methods for mapping roads, and determining the road curvature. The suitability of these methods are discussed in the context of this paper. The next section describes how roads are actually designed. This analysis is useful in generating a model for the road.

Section 4 describes the process of collecting and refining the GPS data. Data is automatically downloaded from the vehicles upon returning to a base station (as illustrated in Figure 2). This data includes vehicle trajectories from logged GPS data, as well as other on-board diagnostic information. A mesh network forms the communication network for the vehicle to base station communication, and also the communication between vehicles. This communication network also allows the distribution of digitised maps to the vehicle fleet. Clustering and linking techniques are used to create a set of road waypoints.

Section 5 introduces a technique for converting a set of waypoints into a set of lines and circular arcs. This is a compressed version of the GPS waypoints. The overview of this technique and a case study can be seen in Figure 1. This figure shows an overview of the process with an example of the amount of compression that can be obtained.

Throughout this paper, experimental results are presented. The data used in this paper is taken from several days of operation of five mining vehicles. Section 6 provides an analysis of the errors introduced in the compression process.

2 Existing Map Building Methods

There are existing bodies of work that use different techniques to determine the curvature in the road. The problem with the majority of these methods is that they are restricted to the special case where there are well

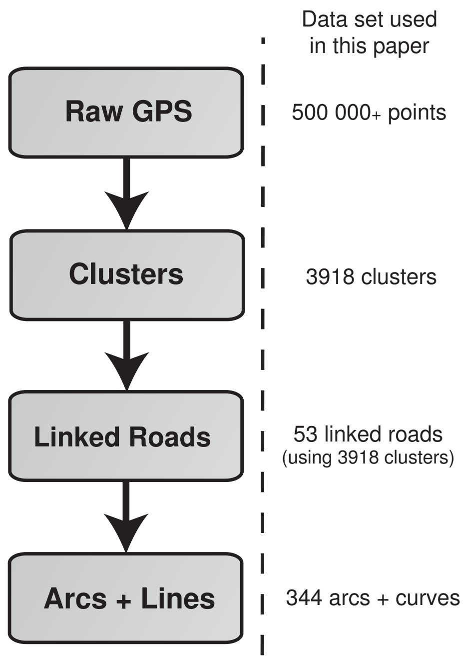

Figure 1: The process for compressing GPS coordinates into a digitised map containing arcs and lines. The dataset presented created 39.3km of road lanes, with an average curve length of 114 meters.

marked lanes on a defined road. This section explores these methods in the context of the environment specified in this paper, where roads are not so well defined.

2.1 Localised lane detection

There are several different bodies of work that try to predict the road curve using image processing [1], [2]. A vehicle with cameras and GPS can perform an inspection of the road, looking visually for lane markings and road edged. In a mining environment, there are no lanes marked, it is often difficult to determine the edge of the road, and there are often environmental problems (such as dust, fog, rain) that proclude the use of cameras.

2.2 Map Building using Satellite/Aerial photographs

Satellite images have been used with image processing to extract the location of the roads (an example can be seen in Vosselman [3]). There are several concerns for the use of this technique in an environment that is not well defined. First, dirt roads often do not have defined edges or lanes, making it difficult or impossible to extract the location of the roads. Secondly, the width of the road can vary along a road, meaning that no assumptions can be made for an image recognition algorithm. The colours of the road can be similar/same as the surrounding environment, again making image detection dificult. Finally, roads in mining environments change often, and satellite and aerial images are often not up to date.

2.3 GPS/image based Map Building

Some work has been carried out in the area of augmenting existing maps [4]. This involves refining the lane information on an existing map. This technique is not suitable in this case because there is no existing digital map available.

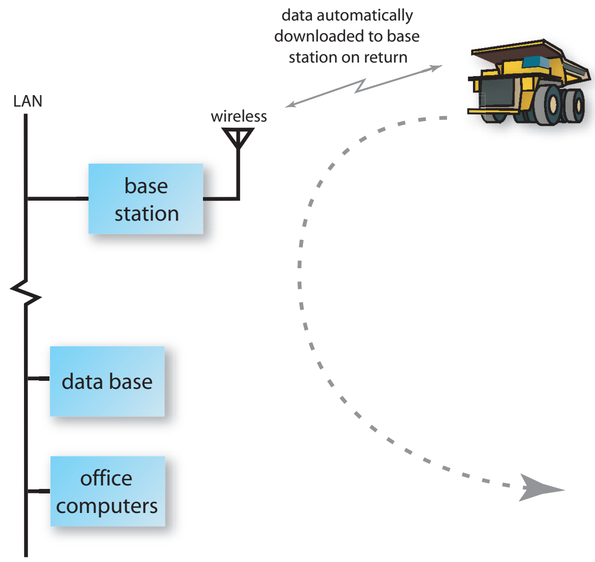

Figure 2: Automatic Process for Collection of GPS Data

3 Road Design - Building a Model

A commonly used, basic representation for a road is to mark a set of waypoints to indicate a change in road heading. This is useful for representing straight roads, though curves require a larger number of points for an accurate representation. When utilising a road map for localisation tasks, especially on a low power and memory computers, it is beneficial to reduce the search space by compressing the map. Also, by using only waypoints, some of the high level information such as the road curve radius is not available. Knowing the curve radius of the road is useful for determining safe speeds in vehicle trajectory planning and safety applications.

Analysis of road design techinques indicate that roads should consist of a "series of straights (or tangents) and circular curves" [5]. For smaller radius road curves, it is necessary to link the straight and the curve with a "transition curve" to allow for a constant change in acceleration between the two segments, generally modelled as a clothoid.

In practice, the transition clothoids form a relatively small proportion of the curve. A large curve with transitions can be represented by a set of circular arcs instead of more complex constructs such as clothoid curves. This allows for comparitively faster computation of distances, which is benefitial for small, low powered processors.

$$\begin{aligned} \text{line} &= \{x_1, y_1, x_2, y_2\} \\ \text{arc} &= \{\text{x}_{center}, \text{y}_{center}, \text{radius}, \theta_1, \theta_2\} \end{aligned} \quad (1)$$

Using this model for roads, the compression from the original GPS trace and the set of road clusters is very high. The two types of road segment (outlined in Equation 1) can be represented by four floating point terms for a line, and five terms for a circle. The line is represented by the (x, y) coordinates for each end of the line. The circle is represented by the center point, the radius and the angular range for which the circle is defined.

With this representation of the road, fitting new (potentially noisy) GPS data to the map becomes almost trivial. Searching for closest curves can be very fast using bounding box techniques, which is very important considering the constraints of the embedded processors.

4 Preprocessing

4.1 Data Collection and Storage

GPS data is collected from vehicles in operation in the mine using a wireless mesh network as illustrated in Figure 2. The vehicles operate continuously, leading to a large set of data that is useful for deriving the road lane locations. Once the data is downloaded, it is stored in an on-site server in a data base. This data base is remotely accessible allowing the data to be accessed off-site.

4.2 Clustering

The data compression process begins by clustering the GPS data into regions of similar position with similar headings. By restricting clusters to contain similar headings, the distinction is made between GPS points from vehicles travelling on opposing sides of the roads. This allows the cluster regions to be larger than the width of the road, while ensuring that the clusters will contain GPS points from one direction on the road only. The necessity of this can be seen from the raw GPS data in Figure 3, where the GPS points form almost a 'cloud' of points, especially around the intersections. There is noise in the GPS readings, and clustering helps filter out the noise by taking the average of many points. This has the effect that for more points added, there is a reduction in noise.

$$\begin{aligned} \text{Set of Clusters } S &= \{c_1, c_2, ..., c_n\} \\ \text{where each cluster } c_i &= \{x, y, \theta\} \end{aligned} \quad (2)$$

The clusters are stored so that when new GPS data is collected from the vehicles, it can be incorporated into the existing cluster set. For new roads, data will be collected by the vehicles travelling in these new areas. This data will be added to the existing cluster set after it is downloaded upon the vehicles return.

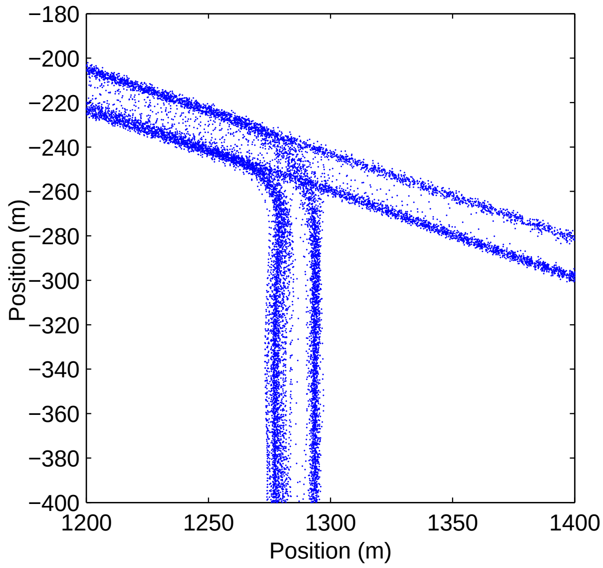

Figure 3: Raw Data Automatically Downloaded from a set of Vehicles

4.3 Linking

The second step in the process is to link the clusters (as indicated in Figure 4) together to form a coherent chain, in effect forming the road structure. For each cluster, the most likely 'next cluster' in the road is calculated using a metric to assess the suitability of nearby clusters.

The metric is a function of the distance between clusters (ΔS), bearing of the other cluster relative to the test cluster heading (ϕ), and the difference in heading between the clusters ($\Delta\theta$). Each of these measures are weighted using the parameters α, β and γ and added to give the metric value. These parameters can be tuned to suit the application in the case of different road widths, GPS noise, etc.

$$\text{metric } = \alpha\Delta\text{S} + \beta\phi + \gamma\Delta\theta \quad (3)$$

Each cluster is linked forward to the most suitable nearby cluster based on the defined metric, which is the next along the road. While each cluster will only have one of these 'forward' links, it is possible for two clusters to link forward to the same cluster. This can happen for a number of reasons such as if there are two clusters close together, or if there is an intersection. The cluster with multiple 'backward' links (as just described) can resolve this ambiguity by selecting the cluster with the lower metric value. The final result of this process is that each

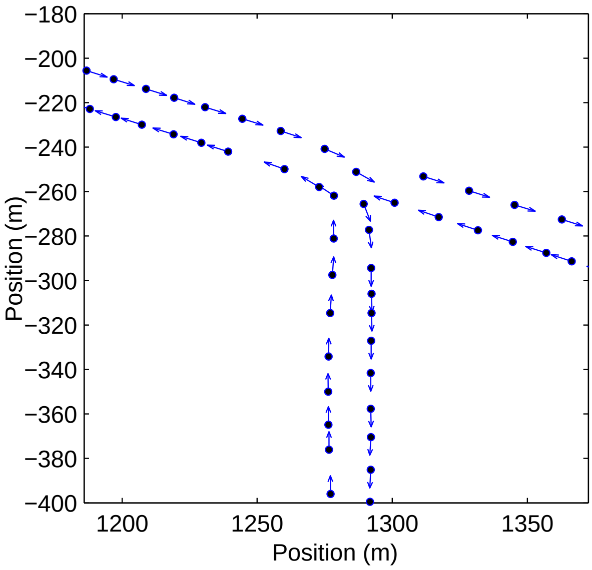

Figure 4: Clusters Shown as Vectors (x,y,heading)

cluster can be linked only to a maximum of one forward and one backward link, as outlined in Figure 5.

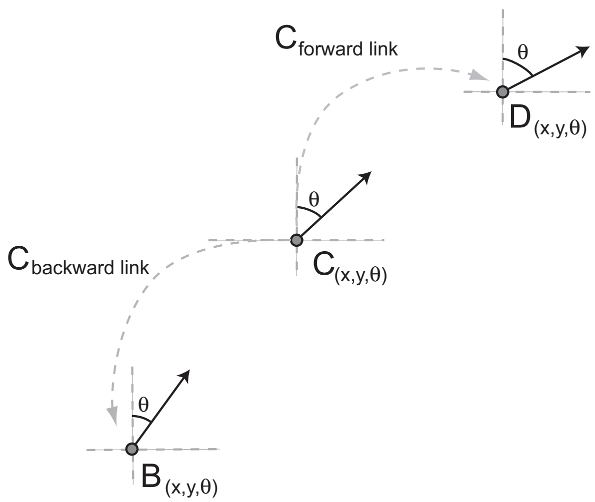

Figure 5: Each cluster can link to a maximum of one Forward cluster and one Backwards cluster

A road is a coherent set of linked clusters that begins with a cluster with no backward link and one forward link, and ends with a cluster containing one backward link and no forward link. The links in between the start and the end must contain exactly one forward and one backward link.

5 Generating the Lines and Arcs

5.1 Finding Sections of Constant Curvature

The extraction of the arcs and lines from the cluster data is a two step procedure. The distinction is first made between groups of linked clusters that have similar curvature. This is done by first calculating cluster heading per unit distance, as shown in Figures 6 and 7. In the graphs, a horizontal line represents a straight line, where the average change in heading is zero. The non-horizontal lines represent a curve in the road, where a steeper gradient corresponds to a sharper road curve.

This data needs to be segmented into areas of constant curvature, which can be difficult depending on the noise introduced from the GPS data and the clustering process. A two stage thresholding process was implemented to overcome the noise. The first point is selected as the start point, then the end point is moved iteratively further away from the start point. For each iteration, a straight line (as on the graphs in Figures 6 and 7) is drawn between the start an the end point, and the square of the error to each point in between is summed. When the sum of the squared error is greater than the first, coarse threshold, the iterations are stopped. The next step is to step the end point back towards the start point until the squared error is less than a second, finer threshold. The aim of this process is to set the coarse threshold high enough to filter out the noise, and the fine threshold small enough so that the error is sufficiently small. These thresholding values need to be tuned to suit the application.

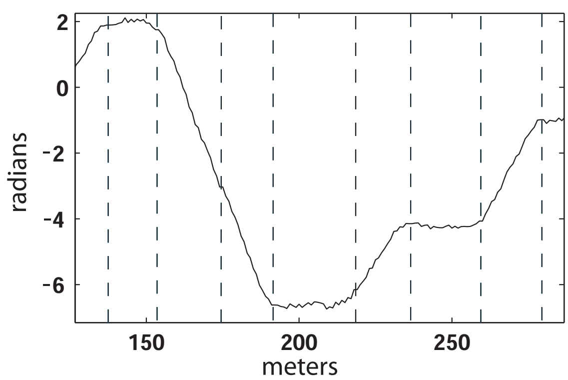

Figure 6: Relative cluster heading measured against distance. The dotted lines indicate the segmentation of the areas of constant curvature.

5.2 Non-Linear Least Squares Fitting

The second step in the extraction of arcs and lines from the cluster data is to perform a regression analysis to

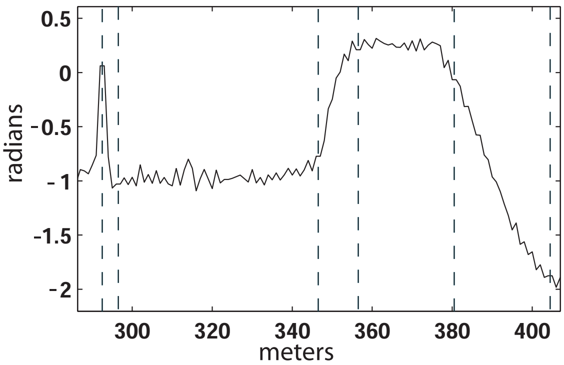

Figure 7: Another example of relative cluster heading measured against distance. Again, the dotted lines indicate the segmentation of the areas of constant curvature.

fit the best line or arc to each segment determined in the previous subsection. For the straght lines, this is a standard regression analysis. The arcs are more difficult since circles are non-linear. The fitting technique used for this analysis is known as non-linear least squares fitting [6].

The non-linear least squares fitting technique involves finding an objective function which is to be minimised, indicating the quality of the fit. The objective function is described in Equation 4. Levenberg-Marquardt Algorithm was used to minimise this objective function, as described in [6].

$$
\begin{aligned}
J(x,y,r) &= \sum \left(\sqrt{(x_i-x)^2+(y_i-y)^2}-r\right)^2 \qquad (4)\\
\frac{\partial d_i}{\partial x} &= -(x_i-x)/(d_i+r)\\
\frac{\partial d_i}{\partial y} &= -(y_i-y)/(d_i+r)\\
\frac{\partial d_i}{\partial r} &= -1
\end{aligned}
$$

Objective Function (J) and set of derivatives for the Levenberg-Marquardt Algorithm

The iterative Levenberg-Marquardt Algorithm requires a good first estimate of the center and radius of the circle. This can be achieved by taking the first and last point of each set of points, and finding the point where their perpendicular vectors intersect. Even with noisy data, this can provide a sufficient estimate of the center of the circle and radius.

6 Results

Figure 8 shows the result of the process introduced in this paper. The lines and arcs are plotted on an image of the area from where data was collected. The lines match very well, and the notches in the lines indicate where one line/arc finished and another begins. It should be noted here that some of the longer curves were broken into two or three sections. This is due to the transition curves, and shows that they are sufficiently modelled by a number of circular arcs.

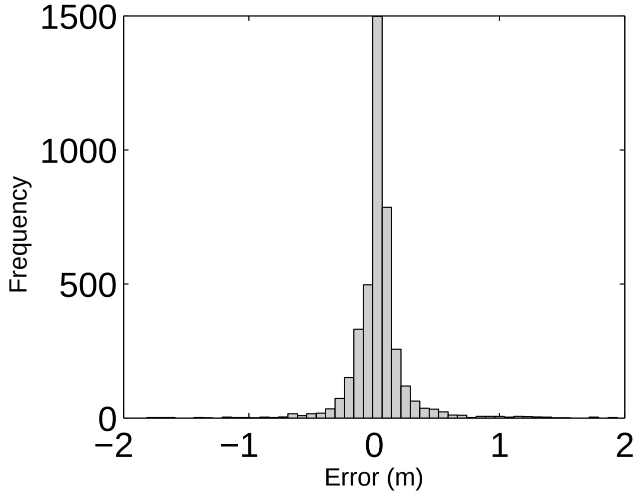

Figure 8: Histogram of the error between the original clusters and the set of lines and circle arcs

The quality of the compression can be seen in Figure 8. This histogram illustrates the amount of error between the thousands of clusters and the corresponding lines and arcs. The compression ratio can be seen in Figure 1. The vast majority of the clusters were located within 0.5 meters of the resulting arcs and lines that make up the compressed digitised map. This is sufficient for all applications mentioned in the introduction of this paper.

7 Conclusion

This paper introduced a technique for generating a minimal digitised road map. The process started by collecting data from a fleet of vehicles, and the result was a set of lines and arcs that make up a road map. It was shown in this paper that the compression from a set of road points (clusters) to the set of arcs and lines had minimal error in the large test data set presented. This technique was used in a mining environment, and was successful even with no defined road edges or lane markings.

Acknowlegements

This work is supported by CRC Mining and the ARC Centre of Excellence program, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

[1] S. E. J. Goldbeck, B. Huertgen and L. Kelch, "Lane following combining vision and dgps," *Journal of Image and Vision Computing*, vol. 18, no. 5, pp. 425–433, April 2000.

[2] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," in *Proc. Intelligent Vehicles Symposium.*, Oct 1994, pp. 109–114.

[3] G. Vosselman and J. D. Knecht, "Road tracing by profile matching and kalman filtering," 1995, pp. 265–274.

[4] S. Rogers, P. Langley, and C. Wilson, "Mining gps data to augment road models," in *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining.* New York, NY, USA: ACM Press, 1999, pp. 104–113.

[5] R. T. Underwood, *The Geometric Design of Roads.* South Melbourne: Macmillan Company of Australia, 1991.

[6] C. M. Shakarji, "Least-squares fitting algorithms of the nist algorithm testing system," *Journal of Res. Natl. Inst. Stand. Technol.*, vol. 103, no. 6, pp. 633–641, November 1998.