Mining GPS Data for Extracting Significant Places

G. Agamennoni, J. Nieto, E. Nebot
Australian Center for Field Robotics
University of Sydney, Australia
{gaga9329, j.nieto, nebot} @acfr.usyd.edu.au

Abstract
This paper presents a fast and robust algorithm for extracting significant places from a set of raw GPS data points. Determining such places provides valuable context information in a variety of applications such as map building, vehicle tracking and user assistance. In our case, we are interested in obtaining context information as a preliminary step towards improving mining safety. The algorithm developed is validated with experimental data sets obtained from a fleet of haulage vehicles operating in an open pit mine in Western Australia.

1 Introduction
1.1 Mining safety
Every year, mining haulage vehicles amount to hundreds of accidents worldwide [Randolph and Boldt, 1996; MSHA, 2008] (see Fig. 1), resulting in a significant number of deaths and injuries as well as substantial costs through replacement, repair and downtime. The most widely accepted approach to reducing the number of accidents is by training vehicle operators to follow a certain set of rules. This approach, however, has several shortcomings. First of all, the rules cannot cover every possible situation. Open pit mines are highly unstructured and complex environments that can exhibit wide variations due to changing weather conditions and working regimes. Second, either deliberately or not, operators do not always follow the rules. The performance of a human driver depends on his or her level of fatigue, concentration, skills, experience and training, and driving ability is often impaired by environmental factors such as heavy fog, dust and rain. Third, due to the size of haulage vehicles, drivers often have very limited visibility. Even when using rear-view mirrors, the geometry of a truck and the location of the driver’s cabin create large blind spots around the truck, which pose a great risk to light vehicles and personnel. Last of all, mines can be thought of as being composed of several distinct places, and vehicle interactions vary greatly from place to place. For instance, in a parking lot or an excavation site trucks tend to drive slowly and very close to each other, whereas along the haul roads vehicle speed and separation is typically much larger. Therefore, it is very hard to compose a fixed set of rules that can guarantee safe operation for every possible situation in all areas of the mine, without incurring an excessive penalty in productivity.

Figure 1: A few mining vehicle-related accidents.

Mining-related accidents can be divided into several categories [Randolph and Boldt, 1996]. One major type of accidents relates to vehicles veering towards the edge of the road or across the centerline. The HaulCheck system [Nebot et al., 2006] was designed to address these issues and is already functioning in several Australian open-cut mines. Another major category of accidents comprises vehicle-to-vehicle collisions, involving both haul trucks and light vehicles. In this regard, proximity and collision-avoidance systems [Worral and Nebot, 2007] were designed to detect potential danger
and provide timely warning to the operator, and have also been deployed in the field. One of the active proximity systems available integrates GPS sensors into a wireless multi-hop network, allowing the transfer of data amongst vehicles and the downloading of logged data to a base station for post-analysis. Such system is currently fully operational in various mines.

1.2 Context

Our objective is to improve mine safety by exploiting the large amounts of raw GPS data made available by the proximity systems. In order to do so, we must be able to process this raw data and extract useful high-level information, such as a measure of danger level or an indicator of the presence of potential threats. This information could then be used for risk assessment and online monitoring, as well as supervising and providing driving assistance to truck operators. The challenge, however, is how to define useful measures of threat and what information to pass onto the operators. In order to account for all forms of danger previously outlined, and to overcome the limitations of the simple rule-based safety approach, we must be able to distinguish between different situations. As an illustrative example, suppose that we impose a speed limit for vehicles working in the mine area, and furthermore we decide to issue a warning signal each time two trucks come closer than a certain range. For a haulage road, where the average truck velocity is in the order of 50 kph, a 60 kph limit with a 40 m threshold may seem suitable. However, when trucks are queuing at the excavation site for loading, they tend to crowd up around the loader. Because of the cluttered environment, a 60 kph limit would now turn out to be too large and the 40 m clearance warnings would no longer be helpful but rather annoy drivers, overburdening them with useless information. Moreover, these values may only be acceptable for ideal atmospheric and road conditions. Dense fog and narrow winding corridors would certainly require far more conservative values than a straight, wide flat road with perfect visibility.

This situational awareness aspect of mine safety can be handled by extracting context information. Context, in our case, is related to the vehicle’s location within the mine and its activity. For instance, a particular vehicle’s context may be described by stating that it is currently at the main excavation site (location), being loaded with copper ore (activity). Recent work on data mining [Liao et al., 2005a] has made substantial progress in extracting this type of context information. In a series of related articles, the authors described an experiment in which they attached a wearable off-the-shelf GPS unit to a person and processed the collected data to infer his or her activities, along with what they called the set of significant places. In their case, significant places were defined as geographic areas that were meaningful to the user, such as residential address, workplace, friend’s houses and shopping centers. Activities were also inferred from the data and associated with different places. In our case, and for our particular application, the significant places in an opencast mine environment would be those areas in which vehicle interactions are qualitatively different (this will be further discussed in Subsection 2.1). Specific examples would include haulage roads, intersections, excavation fronts, crushers, parking lots and maintenance bays, and their corresponding activities could then be labeled as traveling, crossing intersection loading or queuing, dumping, parked and repairing. Nevertheless, for the time being we will not address the problem of recognizing activities since it is not as crucial to our safety application. Instead, we shall focus on determining significant places in a robust and reliable manner, and defer activity recognition to future work.

1.3 Related work

The ability to distinguish between different significant places is the first step towards a comprehensive approach to address mining safety. Extracting significant places from GPS traces has received much attention in the last few years [Ashbrook and Starner, 2003; 2002; Hariharan and Toyama, 2004; Kang et al., 2005; Liao et al., 2005b], although most of these approaches were aimed at applications such as personal hand-held computers and assistance for cognitively-impaired individuals. In these approaches, the algorithm learnt the places by looking at the person’s position and monitoring the time he or she spends at a certain place. Although some of these algorithms are quite involved, their basic underlying principle is the same. Roughly, it consists of identifying positions where the person stays for a certain minimum amount of time \( t_{\text{min}} \) (e.g., 10 minutes or longer), and then clustering them to merge spatially similar points. This simple method, which has several drawbacks, is the one used in [Liao et al., 2005b]1. There are several variations of this method. For instance, some authors [Ashbrook and Starner, 2002] determine the time threshold in an automated fashion by examining the number of places as a function of \( t_{\text{min}} \), while others consider several time scales [Hariharan and Toyama, 2004] and even use varying position scales to construct a place hierarchy [Kang et al., 2005].

1.4 Contributions of this paper

In this paper we present an alternative algorithm for determining significant places. Even though it is intended

---

1Although the authors use a very simple criterion for place extraction, the activity recognition algorithm is quite involved. See [Liao et al., 2005a] for a detailed treatment.
for a completely different application, the proposed algorithm attempts to improve previous approaches in a few aspects. While previous work on the subject focuses on length of stay, our algorithm takes several other variables into account to decide whether a particular location is significant or not. The basic idea is that each sample is scored with a nonnegative value that reflects the “significance” of the vehicle’s current location. Samples which are highly significant are then linked together to form connected graphs, and these graphs are then transformed into simple polygons which constitute the final places. The only parameter that must be set by the user is the mapping from samples to scoring values. Defining such a mapping is straightforward, as will be shown in the following subsections, and allows for a great deal of flexibility. Due to its simplicity, the algorithm also exhibits a high degree of robustness to noise and parameter variations.

This paper is organized as follows. Section 2 is devoted to the derivation of the algorithm. In Subsection 2.1 we give a qualitative, though more detailed, description of the significant places we are interested in finding. Subsection 2.2 deals with the question of defining a suitable mapping and Subsection 2.3 describes the linking stage and how to extract the polygons. Finally, in Section 3 experimental results are presented and discussed, and in Section 4 a few conclusions are drawn, along with comments on possible directions for future work.

2 Finding significant places

2.1 Significant places

To the best of our knowledge, most previous methods for significant location extraction were oriented towards personal user applications. These methods yield a set of regions, represented as geometric shapes such as ellipsoids or polygons, that contain locations that are meaningful to the user. These locations can be the person’s home, workplace, friend’s place or a shopping mall, to name a few. For our present application to the mining environment, the situation is somewhat different. In some aspects it is simpler because our agents, the mining vehicles, typically have much fewer significant places than an average person. Also, they do not switch mode of transportation like a human, which can travel on foot, car or bus, would do. On the other hand, the problem can be more complicated. First of all, unlike a city road network, open pit mines are highly unstructured and changing environments. Haulage roads and excavation sites can differ greatly from one another in terms of shape and size, and their geometric features often shift over time. Second, despite the reduced number of places, mining vehicles do not necessarily follow a predictable routine. Haulage trips are sometimes entirely determined by the client’s requests, especially in small-scale quarries that face intermittent or highly variable levels of demand. Finally, we will be processing data from several agents at a time. This introduces the added difficulty of fusing information from different GPS sources simultaneously. And the quality of this information varies according to the number of visible satellites, their geometric configuration and propagation phenomena such as multipath and fading effects.

As we have already stated in Subsection 1.2, significant places divide the mine into areas in which agents interact in qualitatively different ways. These areas can be further grouped into two major distinct classes according to whether the interactions occur at high speeds or at low speeds, which is a relevant criterion for our current application. High-speed areas would comprise haulage roads, and intersections where typical inter-agent separation is large, whereas low-speed areas would include all other locations, since agents tend to bunch up regardless of whether they are queuing, dumping or parked. To distinguish between high-speed and low-speed regions is the focus of the next subsection, although we emphasize that finding low-speed regions is not the only possible application for the algorithm. Loading sites, crushers and parking lots can be distinguished as separate entities by defining a different mapping and using vector scoring values (see Subsection 2.2).

2.2 Scoring

In this subsection we assume the data for all of the vehicles (henceforth the agents) has already been preprocessed and shaped into a suitable format. Although preprocessing is an important step, it is fairly involved and will not be discussed in this paper due to lack of space. Throughout this subsection and the next we assume that the data for each agent is given by two ordered sets \( \{x_1, \ldots, x_N\} \) and \( \{\phi_1, \ldots, \phi_N\} \). The first set contains position vectors \( x_n \in \mathbb{R}^2 \) and is called the \( \text{trace} \), and the second contains feature vectors \( \phi_n \in \mathbb{R}^M \) which summarize any additional information about the agent at time \( n \). The feature vector can be composed of raw data obtained directly from GPS readings, such as speed and heading values, as well as other values derived from the data. For instance, useful features may be the agent’s rate of turn, the local path curvature and the distances to other agents. All these quantities can be estimated from the raw data using an appropriate mapping or filter and appended to \( \phi_n \). Initially, however, we will include only instant speed \( s_n \) as a feature, since this alone is enough for the purpose of extracting low-speed regions. At the end of this subsection we will discuss the necessary modifications for finding other types of places, such as loading or excavation sites. But for the time being we will replace \( \phi_n \) with \( s_n \).

Given the sequence of features \( \{s_n\} \), we now compute
a corresponding set of scores \( \{y_n\} \). These scores will lie in the interval \([0, 1]\) and must reflect how significant the corresponding sample is. For example, for the case of identifying low-speed areas, we could define

\[
y_n = f(y_n) = \begin{cases} 
1, & s_n < s_{max} \\
0, & s_n \geq s_{max}
\end{cases}
\]

where \( s_{max} \) is a fixed positive constant. The scoring function \( f \) is simply a step function which compares the agent’s current speed against a speed threshold and assigns a high significance whenever the agent is traveling below \( s_{max} \). Although this does not work poorly in practice, it has been found that a much better alternative is to use a relay filter. That is, instead of a unique threshold \( s_{max} \) we pick two speeds \( s_1 \) and \( s_2 \), with \( s_1 < s_2 \), and calculate \( y_n \) as

\[
y_n = f(y_n, y_{n-1}) = \begin{cases} 
1, & y_{n-1} \leq s_1, s_n \leq s_1 \\
\frac{y_n - y_{n-1}}{s_2 - s_1}, & s_1 < y_{n-1} < s_1 < s_n < s_2 \\
0, & s_n \geq s_2
\end{cases}
\]

(1)

The scoring function then behaves like a relay, thus reducing high-frequency clutter due to noise.

Once the scores have been computed for the traces, the next step is to segment them. Given the sets \( \{x_n\} \) and \( \{y_n\} \), we select the set of positions whose corresponding score is unity,

\[ z = \{x_n : y_n = 1\} \subset \{x_n\}. \]

This set contains all the positions for the current agent which belong to low-speed areas. An example of such a set can be seen in Fig. 2. Here, the agent’s instant speed is plotted in blue and the speed thresholds \( s_1 = 5kph \) and \( s_2 = 10kph \) are shown as dashed red lines. Green is used to highlight the samples with unit score, that is, the samples whose position vector belong to \( z \). From the figure it is clear that these samples are not always contiguous, and it is convenient to split \( z \) into several segments,

\[ z_{n:m} = \{x_{n+1}, \ldots, x_{n+m}\} \subset z. \]

Each segment then contains a sequence of consecutive position vectors, that is, vectors with no “jumps” in the sample index. In Fig. 2 there would be two segments, namely \( z_{4306:23} \) and \( z_{43578:21} \), corresponding to the two green lines.

As was already stated, the class of significant places that can be found with this algorithm is not necessarily limited to low-speed regions. Other types of places can be identified by incorporating more features and by defining an appropriate scoring function. As an example, suppose we wish to be able to identify a mine’s loading sites. These regions are characterized by the presence of a loader, which generally serves one truck at a time. When a trucks enters a loading site it parks relatively close to the excavation front and remains immobile while it is being filled. Meanwhile, the loader travels back and forth collecting ore at the excavation front and dumping it onto the truck. This means that during the course of the loading operation both agents remain relatively close to each other. Also, although the loader moves continuously, it’s speed is substantially lower than when it is traveling along one of the haul roads.

Figure 3 shows the values of some selected features during a loading operation, which lasts from sample 30 to 130, approximately. The upper figure shows the speeds of both loader and truck while loading is in progress, and the lower figure shows the distance between them. A quick look reveals that all of the time the agent’s speeds are below 10kph and that they remain less than 20m
2.3 Linking

Once the segments have been extracted, they are linked together to form graphs and finally transformed into polygons. But before they are linked, they are first quantized with a suitable resolution. The reason for doing so is that all mining vehicles are at least 2m long, and hence a resolution smaller than 2m is unnecessary. Most times even 5m is enough for practical purposes, since greater refinements do not improve the final results. Therefore, we create a uniform grid in $\mathbb{R}^2$ with spacing $\delta$ and then snap every point of every segment to this grid.

Then, for each (quantized) segment $z_{n,m}$, we construct a graph with one node corresponding to each of its elements. Edges are introduced between elements that are consecutive, that is, an edge is created between every pair $x_i, x_j$ such that $|i - j| = 1$. Edges are also introduced where the segment intersects itself, or in other words between any pair $x_i, x_j$ such that $x_i = x_j$ and $|i - j| > 1$. Repeating this procedure for each segment yields a multitude of small separate graphs. These graphs are then further fused together to form a super graph. The node set of this super graph is obtained as the union of the nodes of all individual graphs. The edge set, on the other hand, is carried over from the smaller graphs and then augmented by linking nodes with identical positions.

Once all graphs have been fused, they can be transformed into polygons. This is done by first finding the connected subgraphs of the super graph and then tracing the outer boundaries of the subgraph’s nodes. Figure 4 shows a few subgraphs obtained by this procedure, using a resolution of $\delta = 2m$. The nodes are plotted as red dots and the edges as blue lines connecting them, and the boundaries corresponding to each subgraph appear in green. This is what the final significant places look like.

3 Experimental results

Before diving into the experimental results, we present the data that was used as input to the algorithm. The data set was collected from the database of one of the quarries that is currently fitted with a mesh network. It corresponds to an entire day of operation of the quarry and comprises roughly around 130 thousand samples. The position values of the data set spread over an area of more than 4 million square kilometers, and hence plotting the entire set is impractical. Instead, Figures 5 and 6 focus on small regions of the mine which are the most interesting for our current application. The position traces are superimposed on an aerial photograph of the quarry.

In both figures, yellow circles were drawn by hand to mark a few low-speed areas, whose positions were determined based on external information about the mine layout. In Fig. 5 the places that have been highlighted are an ore crusher, a parking lot, a maintenance depot and a pond, which serves as a water supply. In Fig. 6 the three circles mark a large loading area, a washing facility and a pickup point for collecting water. All these places have been marked because, unlike loading sites or stockpiling areas, they do not change over time. The only exception is the large loading area shown in the left of Fig. 6, which is visited frequently enough to be easily identified as a significant place. These marks provide information that will allow us to assess the results generated by the algorithm, at least qualitatively.

Before running the algorithm, the parameters were set as in Fig. 2, specifically $s_1 = 5kph$ and $s_2 = 10kph$. Additionally, a minimum length of 30 seconds was imposed for all segments. This avoids intersections, where agents slow down for making a turn, to be counted as significant locations. Also, places which were only visited once were discarded. This is because quite frequently trucks stop midway along a haul road to make repairs. The trucks park at the side of the road and may stay there for several minutes. Since we do not wish to count such places as significant, and since trucks rarely stop for repairs exactly in the same spot twice, we discard them by retaining only places in which agents have stopped on more than one occasion.

The resulting polygons, calculated with a resolution...
of $\delta = 5\text{m}$, are shown in green in Figures 7 and 8. Here, the data is plotted in two different colors to show where the agents have stopped. Most part of the data appears blue; these are samples for which the agents are traveling at high speeds. In contrast, the segments (see Subsection 2.2) are plotted as red dotted lines. This means that every time an agent slows down below $5\text{kph}$, its position is plotted in red until it reaches $10\text{kph}$ again.

It can be seen from these figures that the algorithm has managed to extract most of the places indicated beforehand. The crusher found by the algorithm matches quite closely the region circled in Fig. 5, and the same happens with the maintenance bay, the pond, the large loading site and the water pickup point. The parking lot, on the other hand, has not been recognized as a single region but as three small polygons. The reason for this is that there are no traces liking the polygons together, since they correspond to three separate parking spots. The washing facility also appears to have been wrongfully detected. The thin red polygon found in the middle of Fig. 8 does not lie within the yellow circle. But this is because the gravel was not being dumped inside the facility itself. Instead, it was accumulated in a nearby stockpile, which the algorithm did recognize as a significant place.

So far we have described the matches between the findings of the algorithm and the locations that were known beforehand from outside information. There still remains the question of what are the rest of the green polygons in Figures 7 and 8. These regions could be either loading sites, stockpiles or even other types of places not accounted for. Moreover there are a few polygons that do not appear in these figures, such as the ones shown in Fig. 9. Most likely, these areas are also loading sites, but at this point we cannot be sure. In order to confirm this, we decided to run the algorithm again, this time using a scoring function specifically tailored to recognize loading sites. Such a function was already described in the last paragraph of Subsection 2.2, and we chose relay functions for both the speed and the distance between agents. The parameters for the speed function were identical to the previous run, namely $s_1 = 5\text{kph}$ and $s_2 = 10\text{kph}$, and the thresholds for the distance function were set to $d_1 = 10\text{m}$ and $d_2 = 20\text{m}$. The grid resolution was left fixed at $\delta = 5\text{m}$.

The results for the second run are shown in Fig. 10, where the same color coding for the data is used. Because the total number of polygons was much smaller than for the previous run, plotting them in separate figures was wasteful of space. Instead, all the polygons were
fitted into a single figure consisting of three horizontal tiles. The upper tile corresponds to the middle part of Fig. 7, the middle tile to the left part of Fig. 8 and the lower tile corresponds to Fig. 10.

From the results in Fig. 10 it can be seen that several loading sites were found, including the large one already marked in Figures 6 and 8. It can also be seen that a small fraction of the parking lot was identified as a loading site. This occurred because of trucks and loaders coming close to each other while they were parking. In order to distinguish between parking areas and true loading sites we would need to give a more thorough characterization of a loading site. This would require greater detail in the definition of the scoring function and is left for future work.

4 Conclusions

In this paper we presented an algorithm for finding the significant places in a mining environment in an autonomous manner. We worked with two types of places, namely low-speed areas and loading sites, and showed how to characterize them in mathematical terms via a scoring function. The algorithm is very fast because the calculations involved in both the scoring and linking of the traces are very simple. Also, it can easily be cast into an incremental version, allowing online updates to be made as soon as new traces become available. Comparing the results with qualitative knowledge about the mine layout indicated that the algorithm performs well. However, there are still improvements to be made regarding the definition of the places and the scoring criteria.

5 Future work

As for future work directions, we will focus mainly on further developing the scoring criteria. At the end of Subsection 2.2 we mentioned a few of the many possibilities for defining the scoring functions. Moreover, there are numerous other methods for scoring traces. One possible alternative could be to use unsupervised learning techniques such as curve clustering or curve alignment techniques [Gaffney and Smyth, 2005] to process the segments, or to extract time series motifs [Patel et al., 2002] directly from the traces. The aligned curves, or the motifs, could then be used to characterize significant places of interest.

References

Mining Health, Safety and Research, Blacksburg, VA: Virginia Polytechnic Institute and State University, 29–38(9) January 1996.


