2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Using Non-Parametric Filters and Sparse Observations to Localise a Fleet of Mining Vehicles

Stewart Worrall and Eduardo Nebot
Australian Center for Field Robotics
University of Sydney, Australia
Email: {s.worrall, nebot} @acfr.usyd.edu.au

***Abstract*— Mining operations generally involve a large number of expensive vehicles, and for the efficient management of these vehicles it is very beneficial to know their location at all times. The current procedure for vehicle localisation in mines is to provide the mine with complete wireless network coverage to facilitate the broadcasting of vehicle positions. This paper examines an alternative method of localisation that does not require the expense of a radio network with full mine coverage. Two different non-parametric filter approaches are presented to estimate the location of the vehicles. A comparison of the two filters is also presented with experimental results using data collected in two operational mines.**

I. Introduction

The mining industry is very capital intensive, using a large number of high cost machines. For a mine to remain financially viable it is important that these machines are managed efficiently and effectively. *Fleet Management Systems* (FMS) are useful tools for assisting mine planners in achieving these goals by monitoring the performance of the vehicles, and detecting bottlenecks in the system.

One major component of any FMS is the ability to know the location of the vehicles in the mine at all times. A mine planner or mine planning software can use a FMS to detect, for example, when there is a long queue at a particular loader and redistribute the vehicles accordingly. Existing commercially available FMS localise the mining vehicles by providing full wireless radio coverage throughout the mine. This is done so that all trucks in the mine can continuously communicate their location and status to a central base station. The cost for such a system is prohibitive for many small to medium capacity mines, especially when the mine covers a large area, or if there is difficult terrain that is not conducive for radio coverage.

This paper is organised as follows. Section II introduces an alternative method to localise the vehicles within a mine without the expense of a full radio coverage network. A mesh network is used to facilitate communication between vehicles, allowing each vehicle to broadcast its GPS position to the other vehicles when in proximity. Section III shows how this data is used to build a computer representation of the mine.

Sections IV and V outline the filtering and prediction algorithms. This is the process where the GPS data collected by the vehicles during operation is used to update a set of localisation filters running on the base station computer. This paper presents and compares the use of two different filters to localise the vehicles in the mine.

Some initial work in this area has been done by the Washington ITS research group [1], [2]. This group presented an approach for predicting bus arrival times at a central location, by considering prior knowledge of the bus travel times and measuring bus speeds along each route. The algorithms introduced in this paper extend this approach, by using prior GPS data to build a model for propagating the vehicle location. The posterior calculation for the vehicle model is non-parametric, and consequently non-parametric filters are used for the implementation.

Finally, Section VI compares experimental results from each filter using real data collected from two mines in Western Australia. These mines have the first version of the haul truck computer system [3] installed. The new hardware described in this paper is currently being implemented in more mines in Australia.

II. Description of the System

Mine Operations

Most mining operations have a central ore collection point, with haul trucks bringing ore from different areas of the mine to this place. The main components of this process that are mentioned throughout this paper are defined below:

- *Crusher:* An ore crushing machine acting as a central collection point for the ore.
- *Loader:* A digging machine that loads ore onto the haul trucks.
- *Haul Truck:* A large vehicle used to carry ore between the loaders and the crusher, see Figure 8.
- *Haul Trip:* A round-trip where a haul truck drives from the crusher to the digger and returns to the crusher.

Each mine generally has a number of loaders, with a number of haul trucks servicing each loader.

Existing System

The basis for this system is the ad hoc communication between computers installed in a fleet of mining vehicles [3]. When two or more vehicles are in wireless range, each vehicle broadcasts its GPS location to the others. During normal operation, a haul truck will log its own GPS data, and also store the time and GPS location of any other vehicles that come into wireless range.

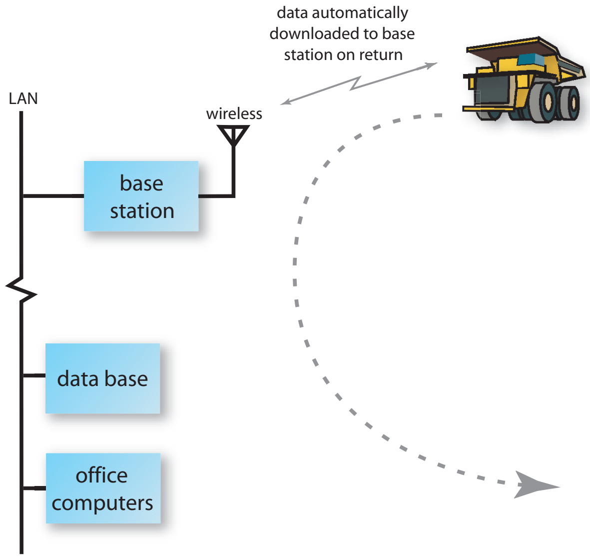

Fig. 1. A truck returns with ore collected from a loader

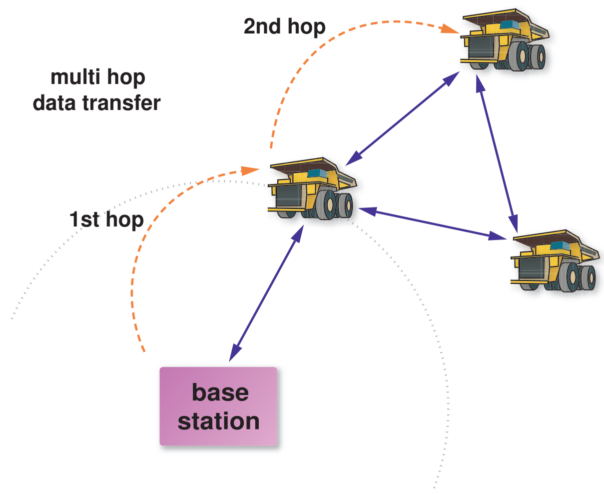

Fig. 2. Mesh network in a mining application

A *Base Station* computer is installed at the crusher, providing the opportunity to regularly download the vehicle data at the end of each haul trip. This means that when a vehicle returns to the crusher, the base station will obtain the GPS data from the haul trip, and the last known locations of other vehicles sighted during this trip. This data is used to update the localisation algorithms that run on the base station for each vehicle in the mine.

The procedure for a truck returning to the base station after collecting ore from a loader is shown in Figure 1 . The data is downloaded at the end of each trip and stored on a data base within the base station.

Mesh Network

The network connecting the haul truck computers has been updated from the previous system [3], with a focus on a robust wireless network. For the algorithms introduced in this paper it is important that a reliable communication network is established. To achieve this, a self healing and robust wireless mesh network has been implemented. The updated network selected for this project uses an implementation of the ”Optimized link state routing” algorithm (OLSR) [4]. Figure 2 shows how the multi-hop capabilities of this network are implemented using each vehicle as a node on the network.

III. Building the Map

A map of the mine is generated using the GPS data collected from the vehicles during normal operation. This is an automated process that eliminates the need for user input to the system. Surface mining operations are dynamic, with the mining area constantly moving and new roads constructed regularly. The map of the mine is updated with each new set of GPS data collected, automatically accounting for these changes to the mine layout. Another benefit of incorporating additional GPS data becomes apparent in Section IV with a broader data set to generate the velocity histograms for each road.

It is necessary at this point to define some terms used in building the computer representation of the mine:

- *Road Section:* A small section of ground that is traversable by vehicles. These are determined using collected vehicle GPS data, clumping together points with similar position and heading. Road sections only use data points that have similar headings, so for a bi-directional road there are different road sections on each side of the road.
- *Road:* A coherent chain of road sections. The road sections are linked together if they are in close proximity and have similar headings. Linking sections with similar headings means that a standard bi-directional ”road” is now considered to be two separate roads, one for each direction.
- *Intersection:* When a road abruptly changes direction, or splits into two roads, an intersection is formed.

Following these definitions, a road is a directional link between two intersections and since it is directional, each road has a start and a finish. At this point, we can borrow terminology from graph theory [5] by considering the mine layout as a digraph (directed graph) as defined in Equation 1 and shown graphically in Figure 3.

Using the digraph representation, the location of a vehicle in the mine can be measured as the distance to the start of a graph edge (road). This is beneficial since a localisation filter with one dimension only (distance) has lower computational requirements than a filter using cartesian coordinates directly from GPS data. Also, search algorithms such as the common A* search [6] can be implemented directly using this graph with distance as a cost function. The A* algorithm is used in this system for planning vehicle routes.

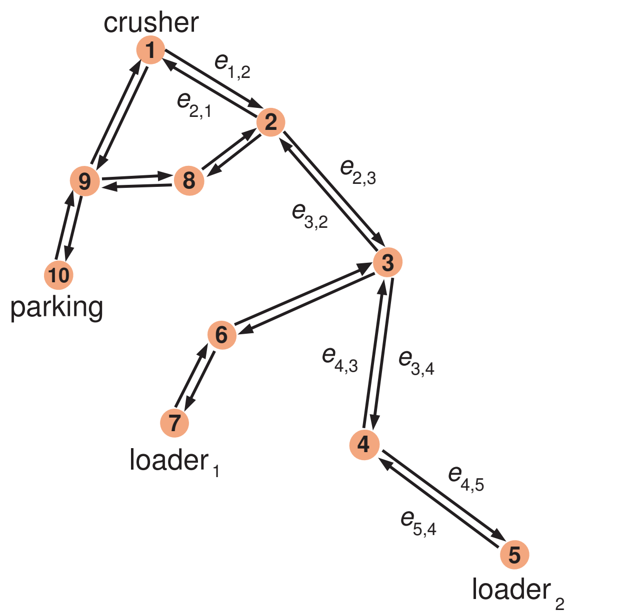

Fig. 3. Graph representation of the mine with vertices numbered and some edges (roads) labeled

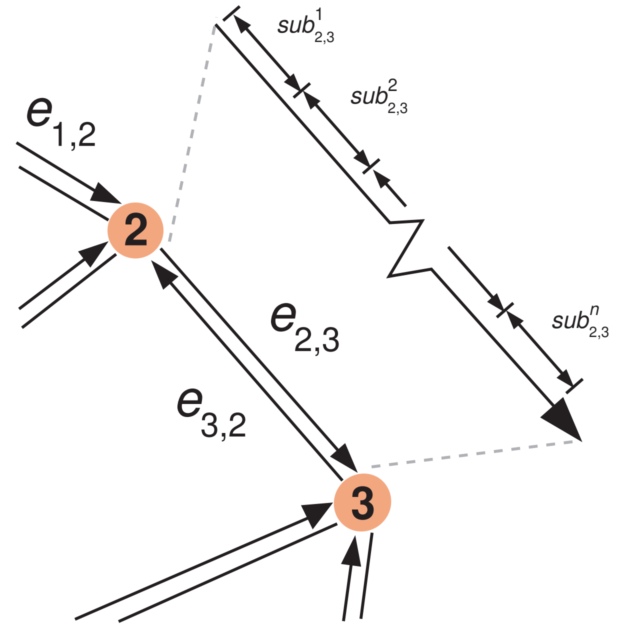

Fig. 4. Close-up of a graph edge showing the division into road sections

$$
\begin{aligned}
\text{Graph of Mine } G &= (V, E) \\
\text{vertices } V &= v_1 \cup v_2 \cup \cdots \cup v_n \\
&= \text{Set of } n \text{ Intersections} \\
\text{edges } E &= e_{1,2} \cup e_{2,3} \cup \ldots etc \\
&= \text{Set of Roads} \qquad (1)
\end{aligned}
$$

where

$e_{i,j}$ = directional link between intersection i and j

As mentioned earlier, each graph edge e is made up of a chain of road sections sub. Figure 4 and Equation 2 describes how the edge is composed:

$$e_{i,j} = sub^1_{i,j} \cup sub^2_{i,j} \cup \cdots \cup sub^n_{i,j} \qquad (2)$$

where

$sub^k_{i,j}$ = the kth road section of edge (i, j)

GPS data collected from the vehicles is used to generate a histogram of velocity for each road section in the mine. Each histogram represents the *velocity profile* for the corresponding road section, and is used by the prediction algorithm to predict the velocity of a vehicle as a function of distance. When the velocity profiles for each road section of a graph edge are shown together, a picture of the traversability of the road becomes clear as shown in Figure 5. In this figure, the x-axis represents the distance along the graph edge (road) and the y-axis represents the velocity profile for each road section.

In Figure 5, properties of the road become apparent. At the beginning of the road, the vehicles generally start from close to zero velocity. This would indicate that this road begins either with an intersection or a sharp corner. At around 300 meters along the road, there is a slow section which indicates a corner. For the remainder of the road, the vehicles travel at an almost constant velocity, with small variation.

IV. Modeling the System

State Space Model

A *trip* is defined as the journey from the base station to the loader and returning to the base station. The destination is assumed to be known before the trip commences. The trip route is calculated with an A* search algorithm using the digraph outlined in Section III. The result of the search algorithm is an ordered set of vertices and edges (intersections and roads) that comprise the complete trip. This is shown in Equation 3.

$$\text{trip} = v_1 \cup e_{1,2} \cup v_2 \cup e_{2,3} \cup \cdots \cup v_2 \cup e_{2,1} \cup v_1 \qquad (3)$$

The combination of these vertices and edges form the state space for the trip. This state space has a single dimension which is a measure of the total distance traveled from the beginning of the trip. This distance (S) is broken into vertices and edges, from which each edge is broken down again into its discrete road sections (Equation 2). The combination of these vertices and road sections form the set of elements (s_i) that make up the entire state space (S) (Equation 4).

$$
\begin{aligned}
S &= v_1 \cup e_{1,2} \cup v_2 \cup e_{2,3} \cup \cdots \cup v_2 \cup e_{2,1} \cup v_1 \\
&\text{substituting Equation 2,} \\
&= v_1 \cup \left\{ sub^1_{1,2} \cup \cdots \cup sub^n_{1,2} \right\} \cup v_2 \cup etc \qquad (4)
\end{aligned}
$$

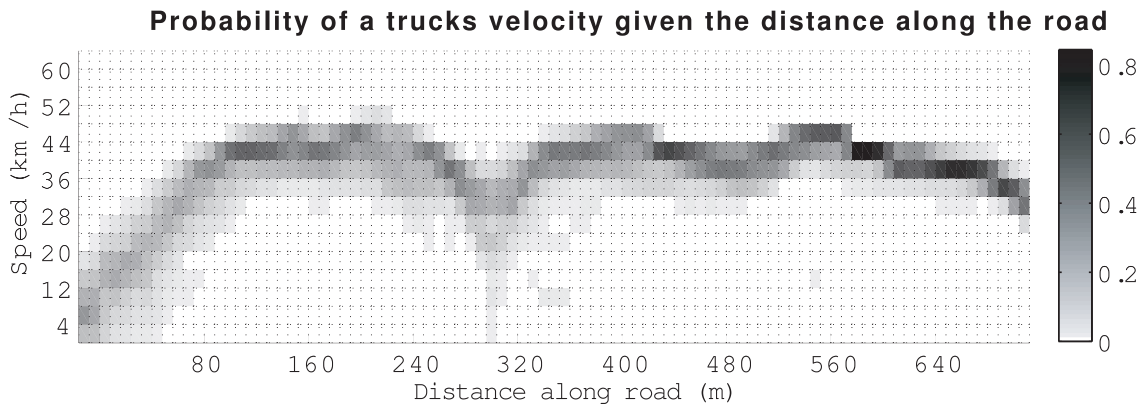

Fig. 5. An example of the velocity PDF generated

The set of elements in Equation 4 form the divisions (s_i) of the state space (S), shown in Equation 5.

$$S = s_1 \cup s_2 \cup \cdots \cup s_m \tag{5}$$

The velocity histogram was generated for each road section using the collected GPS data, as described in Section III. When normalised, this histogram represents a probability density function (PDF). This PDF gives the probability of truck velocity at a given location in the mine, shown graphically in Figure 5. Considering velocity as a function of distance leads to Equation 6;

$$\begin{aligned} \text{velocity } V &= f\left(S\right) \\ P\left(V|S\right) &= P\left(V|s_i\right) \text{ for } s_i \in S \\ &= \text{PDF shown in Figure 5} \end{aligned} \tag{6}$$

Vehicle Model

The vehicle model is based on a distance-only measure from the state space defined in Equation 5. The motion of the vehicle can then be described as shown in Equation 7;

$$\begin{aligned} \text{distance } S &= \int V \ dt \\ S_{t+1} &= S_t + V \ \Delta t \quad \text{(discrete model)} \end{aligned} \tag{7}$$

The propagation model for the prediction algorithm comes from the combination of Equation 6 and 7, shown below;

$$\begin{aligned} S_{t+1} &= S_t + V \ \Delta t \quad \text{(discrete model)} \\ P(S_{t+1}) &= P(S_t) * P\left(V|s_i\right) \text{ for } s_i \in S \\ \text{and with } \Delta t &= 1 \end{aligned} \tag{8}$$

Equation 8 shows that the posterior is given by the convolution ($*$) of the prior distance PDF with the velocity PDF. This equation will be used as a basis for both localisation filters introduced later in this paper.

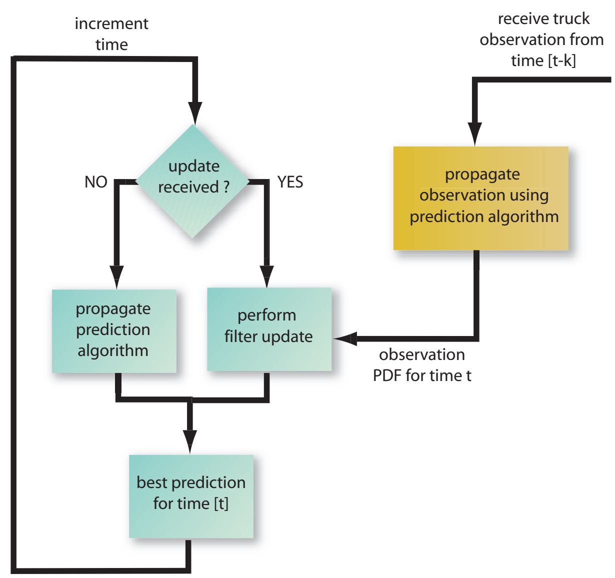

Fig. 6. Overview of filter/prediction algorithm

V. Filter Overview

The localisation filter has been implemented using non-parametric filters due to the complexities and non-linearities inherant in this problem. These complexities arise because the posterior shown in Equation 8 cannot easily be expressed in functional form due to the $P\left(V|s_i\right)$ term. As a consequence, a gaussian filter cannot be applied without changing the model. Non-parametric filters do not rely on the posterior being in a functional form [7], making them ideal for this situation.

A separate filter is maintained on the base station computer for each vehicle. The filter corresponding to each vehicle is initialised when the vehicle leaves the reception range of the base station. These filters are later updated with observations from returning vehicles. Figure 7 shows an overview of how the base station receives the updates. In this example, *truck 2* observes *truck 1* somewhere in the mine. *Truck 2* returns to the base station at time t_{now} and reports the last known location of *truck 1* to the base station. This observation was made at the time t_{now-k}, and before it can be used as a filter update this observation must first be propagated forward to time t_{now}. This is done using the same algorithm from the prediction stage of the filter.

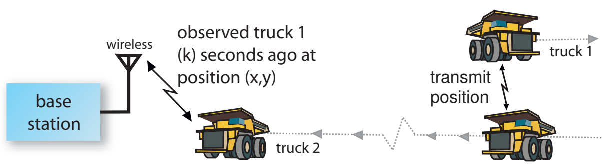

Fig. 7. The base station receives delayed observations from returning vehicles

Two different non-parametric filters have been implemented, a particle filter and a histogram filter.

A. Particle Filter

A particle filter operates by creating a set of location hypotheses (particles) that are propagated using a vehicle model. This model does not need to be in a functional form, it can be non-linear and non-parametric. For this problem, each particle is propagated using Equation 7, where the velocity of each particle is a random sample from the distribution $P\left(V|s_i\right)$.

The basis of a particle filter is that the set of particles are approximately distributed as $P(S)$ [8]. As the number of particles increase, the set of particles more closely approximates the real distribution. The number of particles used in the filter is selected to balance accuracy with computational requirements, this parameter can be tuned to suit different applications.

For a particle r representing a vehicle distance hypothesis existing within the state space S:

$$\begin{aligned} r_{t+1} &= r_t + \int V \; dt \\ &= r_t + \Delta S \quad \text{(discrete model)} \end{aligned} \tag{9}$$

The PDF of the vehicle position is given by:

$$P(S) \approx \{r_1, r_2, \ldots, r_n\} \tag{10}$$

where $n =$ the number of particles.

The particle velocity, shown as $V|s_i$, is a random sample taken from the distribution $P(V|s_i)$. Assuming a constant velocity throughout the segment s_i during the timestep Δt, the ΔS from Equation 9 now becomes:

$$\Delta S = (V|s_i) \cdot \Delta t \quad \text{(within a segment)} \tag{11}$$

If the vehicle travels through multiple road sections during a given timestep, ΔS becomes the sum of the distances travelled in each road section. In practice, at the end of each time step the vehicle location is likely to be somewhere between the start and end of a road section. The algorithm stores the final location at the end of each time step, the remaining distance is used for the next iteration. The particle distance can take any value within the state space S, meaning that the state space representation is continuous.

The particle filter is initialised by creating n particles with the location set to the last known position of the vehicle. When an observation (z) is received from time t_{now-k} (as described earlier in this section), this observation is propagated forward to time t_{now}. This is done by creating a new set of particles at the point of the observation, and propagating these particles forward k time steps. The resulting particles are considered to be approximately distributed as the observation distribution $P(z|S)$, in the same way that the predicted distribution is approximated in Equation 10.

The update of this filter is done by fusing the observation distribution $P(z|S)$ with the predicted distribution $P(S_t|S_{t-1})$.

B. Histogram Filter

The second non-parametric filter considered in this paper is a histogram filter. The continuous state space S is divided into n evenly spaced segments. A $1 \times n$ histogram is assigned to represent the vehicle distance PDF with a histogram bin for each segment of the state space. As the number of segments n increase, the discretised state space approaches the continuous state space and the discretising error is reduced. n is a parameter that can be tuned to suit the application, a balance is required between accuracy and computational requirements.

Using a histogram to represent the vehicle distance PDF allows the convolution defined in Equation 8 to be solved numerically. To do this, we first need to construct a compatible velocity PDF. Recall from Section III that the velocity PDF is constructed as a normalised histogram. If we consider a Δt of 1 second, then we get Equation 12;

$$\begin{aligned} V &= \frac{d}{dt}(S) \\ &= \frac{\Delta S}{\Delta t} \quad \text{for a discrete model} \\ &= \Delta S \quad \text{for } \Delta t = 1 \\ P(V|s_i) &= P(\Delta S|s_i) \text{ for } s_i \in S \end{aligned} \tag{12}$$

The convolution from Equation 8 now becomes:

$$P(S_{t+1}) = P(\Delta S|s_i) * P(S_t) \tag{13}$$

The velocity PDF $P(V|s_i)$ is a normalised histogram representing the likelihood of a velocity given the location in the mine. For a velocity PDF with $1ms^{-1}$ divisions, this can be expressed as:

$$P(V|s_i) = \begin{bmatrix} P(V = 0ms^{-1}|s_i) \\ P(V = 1ms^{-1}|s_i) \\ \vdots \\ P(V = v_{max}ms^{-1}|s_i) \end{bmatrix}$$

Similarly, the histogram represented by $P(\Delta S|s_i)$ is equal to the velocity PDF for $\Delta t = 1$. This normalised histogram describes the likelihood of a change in position (ΔS) over one second given a location in the mine. This can be expressed as follows:

$$P(\Delta S|s_i) = \begin{bmatrix} P(\Delta S = 0m|s_i) \\ P(\Delta S = 1m|s_i) \\ \vdots \\ P(\Delta S = \Delta S_{max}m|s_i) \end{bmatrix}$$

If the size of the divisions in the state space histogram $P(S)$ are the same size as the divisions of the Δs histogram $P(\Delta S|s_i)$, the algorithm for the numerical solution to Equation 13 becomes:

for $i = 0$ **to** n,

$$\begin{aligned} P(s_i) &= P(\Delta S = 0m|s_i) \cdot P(s_i) \\ P(s_{i+1}) &= P(\Delta S = 1m|s_i) \cdot P(s_i) \\ P(s_{i+2}) &= P(\Delta S = 2m|s_i) \cdot P(s_i) \\ &\vdots \\ P(s_{i+\Delta S_{max}}) &= P(\Delta S = \Delta S_{max}|s_i) \cdot P(s_i) \end{aligned}$$

end for (14)

The values shown in Equation 14 for each $P(s_i)$ are then summed to create the posterior.

For these experiments both Δt and ΔS were set to 1. This means that the velocities are rounded off to the nearest $1ms^{-1}$ and the state space is rounded to the nearest metre. For long haul trips, it is possible that n can be greater than 10000 (i.e. the trip length is greater than 10 km). During experimentation, a standard desktop PC was used to run 8 simultaneous histogram filters with $n > 10000$ much faster than real time. It seems from these experiments that the values for Δt and ΔS are suitable for this problem.

This process can be further optimised, since it is not necessary to run the algorithm shown in Equation 14 for values of $P(s_i) = 0$. In this case, the value on the RHS of the equation will always be zero.

The filter is initialised by setting a value of $P(s_i) = 1$ for a vehicle with a last known location of s_i. The filter update is similar to the particle filter. For an observation received at time t_{now-k}, a new histogram is generated with an initial value $P(s_j) = 1$ where the observed location is s_j. This observation histogram is propagated forward k times, producing an observation PDF for time t_{now}. The update is then obtained by taking the multiplication of the observation PDF with $P(S)$. This is possible since the two histograms are both normalised, and of equal length.

VI. Implementation and Results

Both of the non-parametric filters introduced in Section V were implemented and tested using experimental data collected from two mines in Western Australia. The tests involved four trucks driving to a single loader and the results focus on the outgoing trip only. Further research into the effect of loader delay uncertainties and methods for

Fig. 8. A haul truck fitted with GPS and wireless network

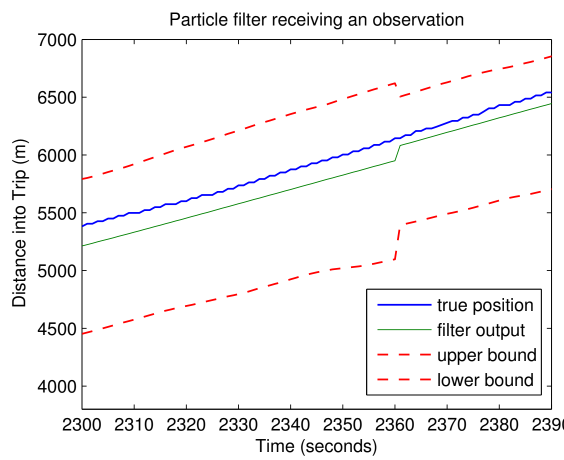

Fig. 9. Particle filter output after an observation ($\approx 4\sigma$ bounds)

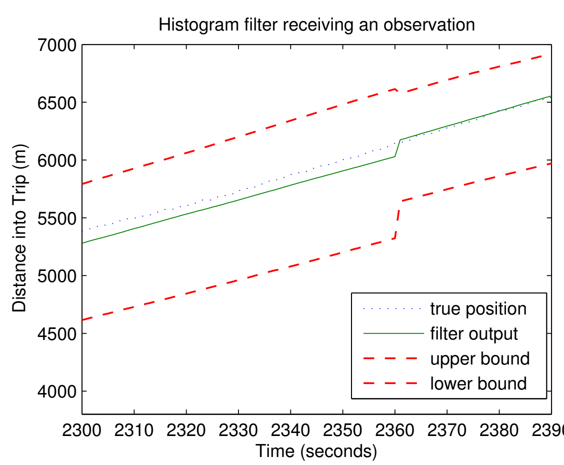

Fig. 10. Histogram filter output after an observation ($\approx 4\sigma$ bounds)

Truck number	trip status	current distance (km)	percentage of total	upper bound (m)	lower bound (m)
13	travelling to loader 1	5.45	35.5%	+561	-755
19	returning from loader 1	10.37	67.6%	+512	-626
21	travelling to loader 1	1.71	11.2%	+228	-456
23	nearby	0	100%	+0	-0

TABLE I
EXAMPLE OF A VEHICLE LOCATION TABLE

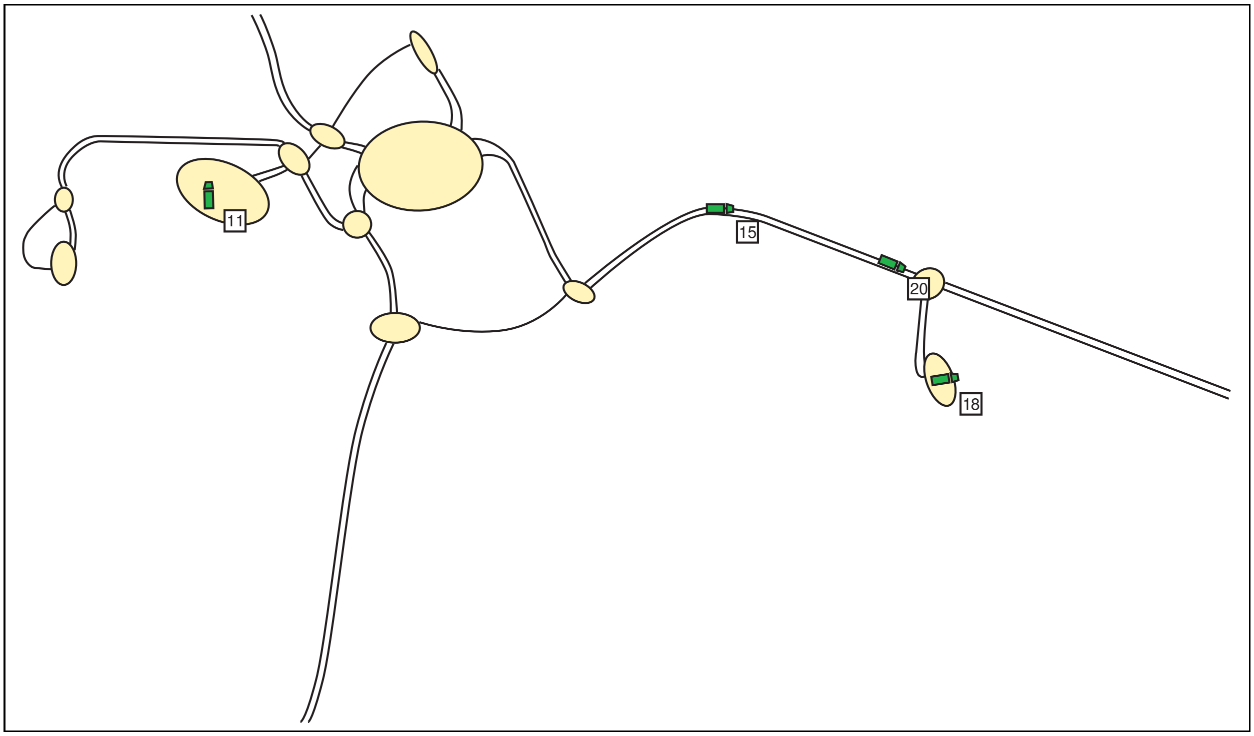

Fig. 11. Screenshot showing a computer representation of the mine with vehicles. The lines represent roads, the ovals represent the intersections.

propagating this information is required before the return trip can be considered.

The results from these tests were very positive, with the true vehicle location sitting within the uncertainty bounds of each filter for the majority of the time. Unexpected events, such as a vehicle driving very slowly, are corrected in the filter when a passing vehicle transports this information back to the base station. If more vehicles drive the same haul route, the filters will be updated more frequently and, consequently, the average uncertainty of the filters will be reduced.

There was very little difference between the outputs of the two filters implemented. Figures 9 and 10 show the output from the two filters using the same dataset, and receiving the same observation update. The variation between these outputs was due to the particle filter taking random samples from the velocity PDF. These random samples mean the particle filter will produce a slightly different output each time it is run.

The other difference between the two filters is that the histogram filter ran approximately 3 to 4 times faster during both the prediction stage and during an update. Since the histogram filter was both faster and had less variation than the particle filter, the outcome of this experiment was that the histogram filter will be implemented in the mine.

When a truck drives consistently faster than the average velocity, the updates provide a correction for the filter. Figure 12 shows this situation, where the dotted line (the true location) is close to the upper bounds of the filter estimate. An observation is received from the base station and the point of update is marked in this figure. At this point, the estimated location jumps towards the true location as expected.

Another important part of this system is the user interface. The output of the filter is converted from a distance measure into cartesian coordinates, then displayed on a computer map of the mine. The screen shot shown in Figure 11 is an example of this interface. This gives the human operators a visual representation of the state of the mine.

The current state of the vehicles and percentage of trip completed can also be expressed in tabular form (see Table I). The table is stored on a data base, allowing access to this data by external fleet management software.

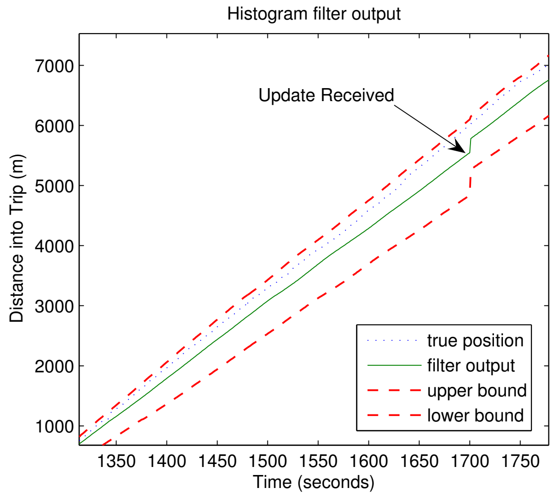

Fig. 12. Graph showing an update to the Histogram filter where the truck was driving faster than the average

VII. CONCLUSION

This paper presents a new method of providing real-time and continuous estimated location information for a fleet of mining vehicles. Data is collected and processed when vehicles return from a haul trip. This data is used to build a computer map of the mine and update the localisation filters on the base station computer.

Two non-parametric filters were used to implement the vehicle localisation, and results of experiments are provided. The data for the experiments was collected from the original haul truck computer system that is currently operating in two mines in Western Australia.

The outcome of these experiments was that the two filters produced similar output, and both provide very good estimates of vehicle location. The histogram filter was selected as the better filter since it was considerably faster to run and did not have the variability of the particle filter. The next stage of this project is to implement the histogram filter approach in a full mine experiment for further testing.

The next stage of this project is to implement a truck queuing algorithm, for predicting the length of time taken for a truck to be loaded with ore. The design of the queuing algorithm, and the methodology for combining this with the localisation filter, is a future area of research.

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] F. W. Cathey and D. J. Dailey, “Transit vehicles as traffic probe sensors,” in *IEEE Intelligent Transportation Systems Proceedings*, Aug. 2001, pp. 579–584.

[2] ——, “A prescription for transit arrival/departure prediction using automatic vehicle location data,” *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3-4, pp. 261–264, June 2003.

[3] E. Nebot, J. Guivant, and S. Worrall, “Haul truck alignment monitoring and operator warning system,” *Journal of Field Robotics*, vol. 23, no. 2, pp. 141–161, March 2006.

[4] P. Jaquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, “Optimized link state routing protocol for ad hoc networks,” in *IEEE INMIC Proceedings: Technology for the 21st Century*, August 2001, pp. 62–68.

[5] E. Kreyszig, *Advanced Engineering Mathematics*. New York: John Wiley and Sons, 1999.

[6] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall, 2003.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Mass: MIT Press, 2005.

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” in *Radar and Signal Processing, IEE Proceedings*, vol. 140, no. 2, Apr. 1993, pp. 107–113.