Straight-Line Walking and Path-Turn Identifying Algorithms for Tracking Devices in Underground Mines

Jingcheng Li, *Senior Service Fellow*
The National Institute for Occupational Safety and Health

BIOGRAPHY

Dr. Jingcheng Li is a senior service fellow at Pittsburgh Research Lab, The National Institute for Occupational Safety and Health. Previously, he was as a senior software engineer at Analog Devices Inc. Earlier, he was an instructor in the Department of Electrical and Automation Engineering at China University of Mining and Technology. He received a B.S. in Electrical Engineering from China University of Mining and Technology in 1982, and an M.S. and Ph.D. from Pennsylvania State University in Mining Engineering specializing in electrical engineering applications in mining engineering in 1992 and 1996.

ABSTRACT

Underground coal mines can be thought of as large coplanar tunnel networks generally laid out in a grid pattern, often extending for many kilometers or miles. A growing number of coal mines are installing miner tracking systems to track miners working underground and provide the location information to a mine office located on the surface. One of the major challenges for these systems is to provide enough accuracy to be able to pinpoint the location of miners within the working areas of the mine to positively impact safety. Many current mine tracking systems use a limited number of sensors placed within key tunnels or intersections of the mine as references to estimate the location of a tracking device (tag) carried by the mine worker. The accuracy of those systems is different in different areas in a mine depending on the density of the sensors. A greater density of sensors can result in a higher system accuracy, but at a higher installation and maintenance cost. Underground mine tracking systems may use the straight-line walking and path-turn algorithms presented here to improve their accuracy in the areas with a reduced density of sensors. The straight-line walking algorithm helps to pinpoint a miner's location within a tunnel path and map his real location to the straight tunnel centerline while the path-turn algorithm helps pinpoint his location in a tunnel intersection and acknowledge that a turn has been made at the intersection. These algorithms are especially useful for inertial tracking devices to refine their location determination in straight tunnels and to recognize when a turn from the straight path has been taken.

INTRODUCTION

Underground coal mines can be thought of as a large tunnel network generally laid out in an orderly coplanar grid pattern, often extending for many kilometers or miles. There are two types of tunnels. One is called an "entry", and another is called a "crosscut". They are generally constructed to intersect each other perpendicularly. For convenience, both "entry" and "crosscut" will be termed "entry" in this paper because the major difference between them is their lengths, the crosscut generally being shorter than the entry. The entry and crosscut have the same width throughout a mine. The entry width will be assumed to be constant in the algorithms. The entries only provide free space for miners to travel and stay.

Many underground coal mines are installing a miner tracking system to track miners working underground and provide the location information to a mine office on the surface. The lives of trapped miners in a mine may be saved during a rescue mission if the tracking system is able to provide the accurate location information of those miners.

It is still one of the major challenges for the current tracking systems to provide the location of a tracking device worn by a miner consistently within an entry path throughout a mine. An entry path covers the space between two adjacent entry intersections. The typical entry path length is 30 m (100 feet) and its corresponding width is 6 m (20 feet). The majority of the current tracking systems use a limited number of external sensors placed in some key entries and entry intersections of the mine as positioning references to estimate the location of a tracking device. The accuracy of those systems ranges from a distance from less than 15 m to more than 300 m, depending on the density of the sensors in the mine. The radio frequency identification (RFID) tracking system is an example of such a system. In that system, readers are placed in strategic locations in a mine to sense the signals from the tags located on the miner. The system accuracy of those types of the tracking systems depends almost entirely on the density of the sensors. A greater density of sensors can result in a higher system accuracy, but at higher purchase, installation and maintenance cost.

The straight-line walking and the path-turn identifying algorithms presented here can be used to pinpoint the location of a tracking device within an entry path in an area with reduced density of external sensors to improve the overall accuracy of the tracking system. The straight-line walking algorithm also maps the device's actual location to the entry's centerline as the device's location to simplify the display and the record of the device's trace. The path-turn identifying algorithm helps a device to identify a turn at an entry intersection.

A miner walking in an entry is generally unable to keep a straight path. A tracking device, especially an inertial tracking device, carried by a miner, will draw a curved trace. Figure 1 shows three examples of curved traces drawn from three inertial tracking devices when they are moving in the same entry. It is difficult to directly use those different traces for the devices' position recording and displaying. The straight-line walking algorithm identifies a tracking device's entry path, and converts its curved trace to a straight path along the entry's centerline. In Figure 1, *W* is the width of the entry, which is typically, as stated before, 6 m (20 feet.)

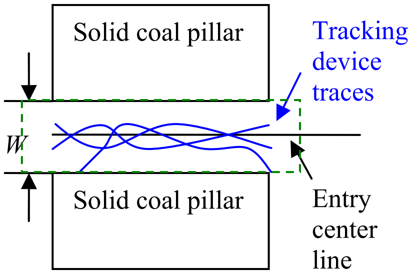


Figure 1: Tracking device traces in the inside of an entry path

Figure 2 shows some examples of the path turns at an intersection by an inertial tracking device at different time. The turning traces differ from each other. One of them passes over the center point of the intersection; one follows the perimeter of the intersection; one runs near a corner of the pillar. The path-turn identifying algorithm is able to detect all of those turns regardless of the turning traces.

Instead of using the external sensors as the positioning references to locate the miners, these algorithms use the mine's inherent landmarks as positioning references to locate the miners. The algorithms take advantage of the uniform nature of mine entries. The entries, constructed as straight segments, have almost constant widths throughout the mine. An entry path space generally appears in a rectangular shape with its length equal to the path length and its width equal to the width of the entry as shown in the dotted lines in Figure 1. To the straight-line walking algorithm, the rectangular area is the landmark for the entry path. The straight-line walking algorithm locates the tracking device's entry path by locating that entry's rectangular area that contains the device. Similarly, the landmark for an entry intersection is the intersection's square area with each of its sides equal to the entry width as shown in the dotted square in Figure 2. The path-turn identifying algorithm identifies a path turn by locating the square area that contains the tracking device.

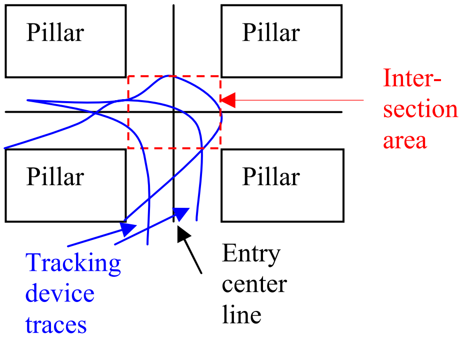


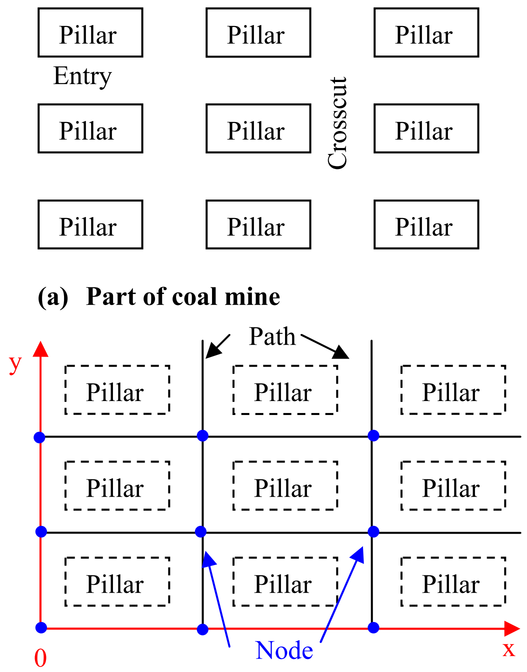
Figure 2: Tracking device turns from one entry path to another

The tracking device worn by a miner is commonly called a "tag." We will use the terms "tag" "tracking device" interchangeably in the latter sections of this paper.

The operating foundation of both algorithms is the mine's coplanar node-path network model described in [1]. The paper starts with a brief introduction of the mine's node-path network followed by the introduction of the algorithms.

MINE MODEL

The mine's coplanar node-path network model can be thought of as a mathematical representation of the mine's underground entry network. The detailed description of it can be found in [1]. The node-path network is formed from all of the straight centerline segments of the mine entries. Those segments are called network paths representing the entries. The intersections of those paths are called network nodes representing the entry intersections. A part of a mine shown in Figure 3 is used to illustrate the concept of this node-path network representation of the mine's entry network. Figure 3 (a) shows a portion of a mine; and Figure 3 (b) shows its corresponding node-path network with the coplanar coordinate system superimposed on it. With the use of the mine's node-path network to model a mine, the tags are only allowed to travel along the network paths and make a turn at a network node.



(a) Part of coal mine

(b) The node-path network

Figure 3: Mine and its node-path network

STRAIGHT-LINE WALKING ALGORITHM

The algorithm locates an entry path for a tag by locating the path's rectangular area, and then maps the actual location of the device to the nearest point on the entry's centerline. The tag thus appears to be moving along the entry's centerline.

The following assumptions are made for this algorithm. 1) It is able to save and retrieve the previously mapped position of the tag on the entry centerline from the last round operation. 2) The algorithm requires as an input, the width of the entries, *W*, and regards the value as a constant. 3) A mapping tool which is not considered part of this algorithm has already mapped an actual three-dimensional (3-D) location of the tag. If the tag outputs its location readings in 3-D in the real world (e.g., inertial tracking device), then they are mapped to a location (x_i, y_i), called initial device location (IDL), on the mine's coplanar node-path network as shown in Figure 4. In the figure, (x_0, y_0) and (x_1, y_1) are the endpoints of the entry centerline segment; (x_p, y_p) is the previously mapped position (PMP) on the entry centerline; (x_c, y_c) is the nearest point on the entry's centerline mapped from the device's IDL (x_i, y_i). The rectangular territory of the entry path is also marked in the figure.

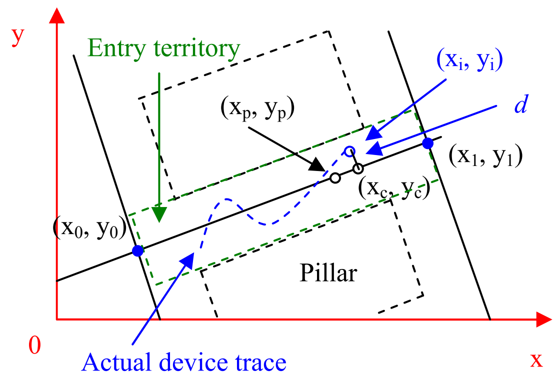


Figure 4: A curved path of a tag

The algorithm takes several steps. The first step is to fetch the PMP (x_p, y_p), and to retrieve the location data of the two endpoints of the entry's centerline segment that the PMP is on. They are (x_0, y_0) and (x_1, y_1) as shown in Figure 4. Calculate the perpendicular distance, *d*, from the IDL (x_i, y_i) to the centerline using (1).

$$d = \frac{\left|(x_1 - x_0)(y_0 - y_i) - (x_0 - x_i)(y_1 - y_0)\right|}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}}. \qquad (1)$$

The second step is to examine whether the tag is still located in the entry's rectangular territory and not in a coal pillar by using (2) where *W* is the width of the entry. The tag will be inside of the entry territory as long as inequality (2) holds true as shown in Figure 5. Otherwise, it will be out of the entry territory, or in a coal pillar, and a method of finding the closest entry should be considered to correct the tag's location before continuing with this algorithm.

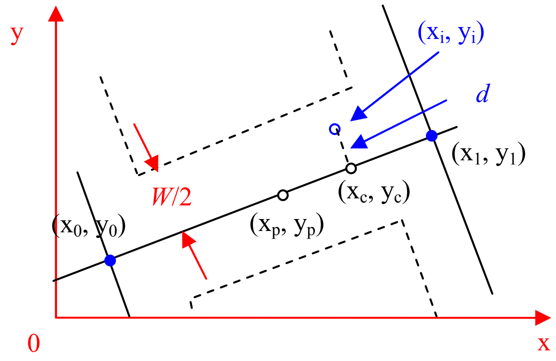


Figure 5: Check if the tag is in the entry territory and find the nearest point on the entry centerline from the tag's IDL

$$d \le W / 2. \qquad (2)$$

The third step is to map the tag's location (x_i, y_i) to the nearest point (x_c, y_c) on the entry's centerline as shown in Figure 5. Point (x_c, y_c) is actually the intersection of the entry's centerline and the perpendicular line A through (x_i, y_i). Many formulas can be used to find (x_c, y_c), and equation (3) is the one of them given in [1].

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{cases} \begin{bmatrix} x_i \\ y_0 \end{bmatrix}, & \textit{for } a = 0 \\ \begin{bmatrix} \dfrac{a^2 x_0 - a y_0 + x_i + a y_i}{1 + a^2} \\ \dfrac{a x_i + a^2 y_i - a x_0 + y_0}{1 + a^2} \end{bmatrix}, & \textit{for } 0 < a < \infty \\ \begin{bmatrix} x_0 \\ y_i \end{bmatrix}, & \textit{for } a = \infty \end{cases} \qquad (3)$$

$$\textit{where } a = {(y_1 - y_0)}\big/{(x_1 - x_0)} \textit{ slope of the entry centerline}$$

The final step is to use the tests in (4) to check if the tag is indeed located within the territory of the current entry path and not on its extension. The tests in (4) examine if the mapped point (x_c, y_c) is in the valid range of the entry centerline segment from (x_0, y_0) to (x_1, y_1), and should be satisfied if the point (x_c, y_c) is located within the entry's solid centerline segment. The algorithm will update the device's PMP with its current position on the centerline if all the tests have been passed.

If one or both tests in (4) fail, the tag's mapped point will be on the extension of the centerline of the entry path indicating that the tag is out of the current entry territory. When this happens, a large change in position of the IDL (x_i, y_i) between two consecutive readings is assumed because the tag must normally pass through the shared area of the current entry path and its adjacent entry path in their intersection before moving out from the current entry path. A large change in position of the tag's IDL can be prevented by having a location update time short enough that the tag can not miss the entry's shared area when it is transiting from one entry path to another. The details on a normal transition of the tag from one entry path to another will be given in the section on the path-turn identifying algorithms.

$$\begin{aligned} &\min(x_0, x_1) \le x_c \le \max(x_0, x_1) \\ &\min(y_0, y_1) \le y_c \le \max(y_0, y_1) \end{aligned} \qquad (4)$$

At each tag update, the curved trace of the tag is mapped to the entry's centerline. As a result, the tag appears as if it is always on the centerline of the entry.

Because the tag's true trace is mapped to the entry's centerline, a position difference between its true location and its marked location will result. The maximum position difference in a given reading will be a half of the entry width or *W*/2, which is the distance between the centerline and one side of the entry pillar walls. For most mines with an entry width *W* = 6 m, the maximum position difference produced from the algorithm is then 3 m.

PATH-TURN IDENTIFYING ALGORITHM

The path-turn identifying algorithm gives a tag an ability to recognize a turn it makes at an entry intersection. As shown in Figure 2, tags make many turning traces; the algorithm must be able to detect all of them, and mark them as a simple turn at the node on the mine's node-path network. The turn-algorithm acknowledges a path turn only after the tag has completed a turning sequence. The turning sequence includes two events in series; the tag enters into an intersection area first and then enters the territory of another connected entry path. An entry intersection area is considered as a shared territory of all of the connected entries at the intersection as shown in Figure 2 in which four entry paths share the intersection; each of them duplicates half of the intersection area. The shared intersection area of all the entries is clearer when looking at Figures 2 and 4 together. It is obvious that two perpendicular entry paths overlap by a quarter of the intersection area.

The following assumptions are made for the turn-algorithm. 1) The algorithm is able to save and retrieve the location information of the network nodes. 2) The width, *W*, of the entry is known to the algorithm as a constant. 3) The shape of the intersection area is a square of dimensions *W* by *W*, this is typical of most underground coal mines. In some rare cases in which the intersections have an irregular shape, the square area can be extended to cover the furthest point of the intersection. 4) A mapping tool, which is not included as part of the algorithm, has mapped a 3-D location of the tag. If the tag happens to produce its location readings in 3-D in the real world, then it will be mapped to a location called the initial device location on the mine's coplanar node-path network so that the path-turn identifying process can be performed on the mine's 2-D network. Figure 6 shows a general example when a tag enters into an intersection, where (x_i, y_i) is the initial device location (IDL) of the device; (x_p, y_p) is its previously mapped position (PMP); (x_0, y_0) and (x_1, y_1) are the two endpoints of the entry centerline segment A. B is the centerline segment of the entry path that intersects line A.

The straight-line walking algorithm will always be running concurrently with the path-turn identifying algorithm. Among other things, successfully running the straight-line walking algorithm can ensure that the tag is located within the valid entry territory and not in a coal pillar. After each execution of the straight-line walking algorithm, the path-turn identifying algorithm will perform a simple test to determine if the tag has moved into the intersection area. If the tag is determined to be in the intersection area, then the turn-algorithm will continue with its full course of execution. Otherwise the turn-algorithm will stop and hand the execution back to the straight-line algorithm. Figure 6 shows an example when a tag enters the intersection area.

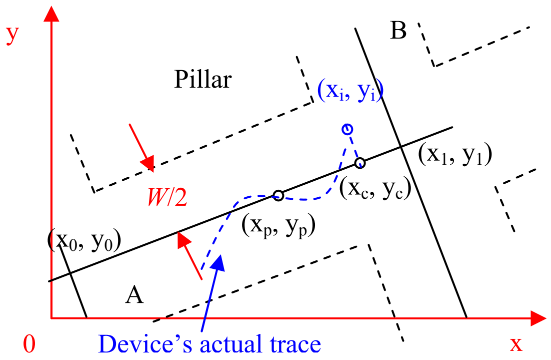


Figure 6: A tag enters an intersection area

The turn-algorithm must first determine if the tag is in the intersection area. A tag is considered to be in the intersection area if a circle centered at the IDL (x_i, y_i) with a radius of $W/2^{1/2}$, where W is again the width of the entry as shown in Figure 7, includes the intersection node (x_1, y_1). The purpose of selecting $W/2^{1/2}$ is to ensure that the circle is big enough to include the intersection center point (x_1, y_1) no matter where the point (x_i, y_i) is within the intersection area. $W/2^{1/2}$ is also the distance from the network node (x_1, y_1) to the corner of a pillar. The function for the circle is given in (5). Understandably, if the tag is anywhere in the intersection area, its IDL (x_i, y_i) must satisfy the inequality (6), or, in other words, point (x_i, y_i) and the node (x_1, y_1) must be in the same circle. The inequality (6) can then be selected to serve as the sole test to check if the tag is in the intersection area or not; if it is, the turn-algorithm will proceed with the rest of its operations.

$$(x-x_i)^2+(y-y_i)^2=W^2/2. \qquad (5)$$

The entry intersection is the shared area of the connected entry paths over it. The next step is to determine which entry path shares its territory with the current entry path. An example is illustrated in Figure 8 where the tag is within the shared area of both entry paths A and B. In addition to entry path A, the straight-line walking algorithm can now also be applied to the entry path B to obtain the mapped point (x_{c1}, y_{c1}) on the centerline of the entry path B and the distance, d_B, between the IDL (x_i, y_i) and (x_{c1}, y_{c1}). Because the tag is also in the territory of the entry path B, it must have $d_B \le W/2$. As such, two mapped points (x_c, y_c) and (x_{c1}, y_{c1}) on two entry paths A and B have been obtained from the IDL (x_i, y_i). (A test similar to (4) can be used to determine if the tag is in entry path C or D.) These mapped points all need to be saved for later recall.

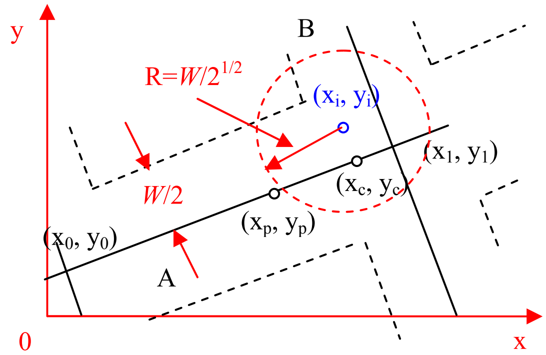


Figure 7: Tests if the tag enters in the intersection area

$$(x_1-x_i)^2+(y_1-y_i)^2\le W^2/2. \qquad (6)$$

Normally a tag's IDL (x_i, y_i) should be mapped onto two perpendicular entry paths in the intersection area as shown in Figure 8 because every two of those perpendicular paths are overlapped over a quarter of the whole intersection area. However, a test may also show that the an IDL (x_i, y_i) can be mapped onto three entry paths A, B and D when the tag's IDL (x_i, y_i) happens to be on the centerline of the entry path A causing $d_A = 0$, or four entry paths, A, B, C and D as the IDL is on the intersection node. If they happen, all of those mapped points need to be saved in order to track the movements of the tag in the intersection area. Obviously as long as the tag remains within the intersection area, its IDL can be successfully mapped onto two or more entry centerlines, and inequality (6) continuously holds true. As long as the tag remains in the intersection area, the turn-algorithm marks the tag's location at the node (x_1, y_1) regardless of its actual location indicating that the tag has not left the intersection yet.

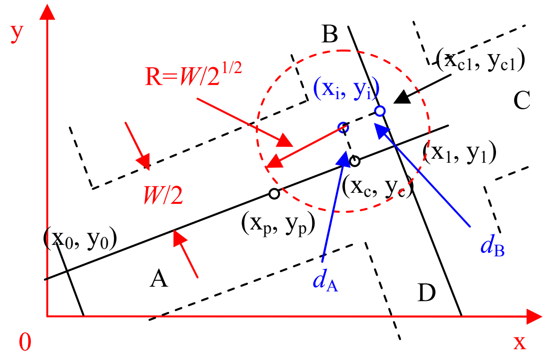


Figure 8: The tag is on two entries

When inequality (6) first fails and the straight-line walking algorithm finds that it can no longer map the tag's current position onto two or more entry centerlines, but only one, the turn is completed. Figure 9 shows an example of an up or left turn of the tag from the entry path A to B. It can be seen that the node (x_1, y_1) falls outside the circle centered at (x_i, y_i) with $R = W/2^{1/2}$, indicating that the tag has just left the shared intersection area. It is also true that $d_A > W/2$ and $d_B \leq W/2$. It can also be seen in Figure 9 that the straight line walking algorithm can no longer apply to the entry path A but only entry path B. The tag starts moving along entry path B alone. The turning sequence is complete. Similar turns from the entry path A to the entry paths C and D can also be identified in the same way by the turn-algorithm.

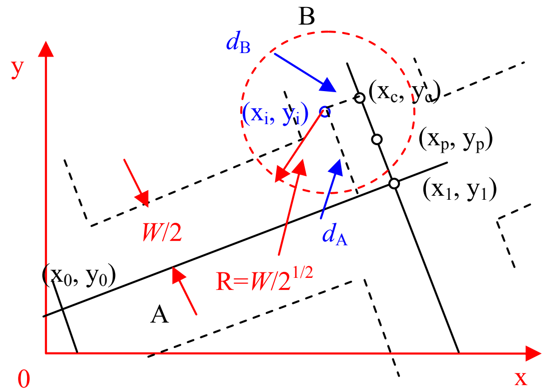


Figure 9: The algorithm acknowledges the turn from entry path A to entry path B

The maximum position difference between a tag's IDL and the marked location by the algorithm in a given reading produced by the path-turn identifying algorithm is $W/2^{1/2}$ which is the radius of the circle. The maximum position difference occurs when a tracking device makes a turn right at the corner of a pillar.

DISCUSSION

The straight-line walking and path-turn identifying algorithms can still be used for entry paths and intersections that have irregular widths even though those entries and intersections are less common in underground coal mines. By simply considering the value of the widest portion of the entry path and intersection as the input constant, *W*, to the algorithms, the algorithms should locate the entry path for a tag and identify a turn it makes as they normally do with uniform entry paths and intersections.

SUMMARY

The straight-line walking and the path-turn identifying algorithms are presented for underground mine tracking systems. The straight-line walking algorithm gives a tracking device the ability to locate its own entry path, and map its curved trace onto the straight centerline of the entry. The path-turn identifying algorithm gives a tracking device the ability to recognize a turn at an entry intersection. These algorithms can be supplements for electronic tracking systems which use external reference sensors to locate miners, to improve the overall system accuracy of the underground tracking systems. These algorithms use the mine's inherent markers as positioning references to locate tracking devices. In this sense, the algorithms turn the obstacles to tracking created by the underground entry (tunnel) network into an assistant to improve the accuracy of the tracking systems.

REFERENCES

J. Li, D. Snyder, N. Damiano, "Exploration of Two position Adjustment Methods For Underground Mine Tracking Systems." ION ITM Anaheim CA, Jan. 24-26, 2009, Proceedings of The Institute of Navigation, d3-6.

[2] D. Bernstein, A. Kornhauser "An Introduction to Map Matching for Personal Navigation Assistants." Proceedings of the 77th Annual Meeting of the Transport Research Board, 11-15 January 1998, Washington.D.C, USA, pp. 561-569.

[3] G. Taylor, G. Blewitt. "Virtual Differential GPS & Road Reduction Filtering by Map Matching." ION GPS '99, 14-17 September 1999. pp. 1675-1684.

[4] J. Geenfield. 2002. "Matching GPS Observations to Locations on a Digital Map," www.njtide.org/reports/TRB2002-greenfeld.pdf

[5] M. Yu, Z. Li, Y. Chen, and W. Chen, "Improving Integrity and Reliability of Map Matching Techniques." Journal of Global Positioning Systems 2006 Vol. 5, No. 1-2:40-46

[6] M. Bittle, P. Brandt, A. Facemire, M. Hunt, M. Sweetman, 2007 "Team 2: Mine Tracking System: Detailed Design Review," Senior Design Seminar, West Virginia University, https://eidr.wvu.edu/files/5444/MineTrackingSystem.pdf

[7] L. Miller, 2006 "Indoor Navigation for First Responders: A Feasibility Study," National Institute of Standards and Technologies, www.antd.nist.gov/wctg/RFID/Report_indoornav_060210.pdf