Algorithms to Identify Its Own and Surrounding Tunnels for an Underground Mine Tracking Device

Jingcheng Li, *Senior Service Fellow*
Richard Unger, *Lead Research Engineer*
The National Institute for Occupational Safety and Health

ABSTRACT

Underground coal mines can be thought of as a large, intersecting tunnel network generally laid out in a grid pattern, often extending for many kilometers or miles. A growing number of underground coal mines are installing miner tracking systems to monitor miners working underground. 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 the degree that safety is positively impacted. Many current mine tracking systems use a limited number of sensors placed within key tunnels or intersections of the mine as location references to estimate the location of a tracking device carried by the mine worker. The accuracy of those systems could be less than 15 m in one area or greater then 300 m in another area of the same 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. In addition, more sensors imply the need for additional power components and battery backup units, along with the risks those systems introduce in the underground environment. The algorithms presented here can be used for a tracking device to locate the tunnel it is in or nearby tunnels in an area with a lesser density of sensors to improve and maintain the system accuracy. The algorithms operate on a tunnel intersection matrix of a mine. The tunnel intersection matrix is a collection of the locations of tunnel intersections in a global coplanar coordinate system of the mine's tunnel network, and serves as a mine-wide tunnel geometrical layout information source. The algorithms use the information to locate a tracking device's own or tunnels nearby if it cannot locate its own tunnel. The accuracy of the tracking system is hence less dependent on the density of the external sensors.

INTRODUCTION

An underground coal mine can be thought of as a large tunnel network laid out in an orderly grid pattern. The main tunnels are called entries, which may extend for many kilometers (miles,) and are intersected by the tunnels called crosscuts generally in a perpendicular or near perpendicular manner. The distance between two adjacent intersections varies from mine to mine, and is typically about 30 meters (100 feet.) In this paper, all of the mine's tunnels are called "entries" regardless of whether they are entries and crosscuts, because the types of tunnels have no influence on the operation of the algorithms.

Many coal mines have installed a tracking system to monitor the location of miners in the underground entry network and provide the location information to a mine office on the surface. This is to help protect lives by being able to provide an accurate location of the miners during a rescue operation. Presently, it is still a challenge to construct a tracking system that is able to provide a consistently precise location of a miner within an entry path.

Based on the current technologies available for the underground mining industry, the Mine Safety and Health Administration has set the required accuracy of tracking systems from 60 m (200 feet) in the main working sections to 600 m (2,000 feet) in haulage or escapeway entries. Nearly all of the current tracking systems use external sensors as references to help locate the tracking device worn by the miners. A typical example is a radio frequency identification (RFID) tracking system. In an RFID tracking system, key entries or intersections of the mine contain RFID tags or readers to provide a tracking location reference. The accuracy of such a tracking system is almost entirely dependent on the density of the RFID tags or readers. A greater density results in higher system accuracy, but at higher purchase, installation and maintenance cost. In addition, more reference units imply the need for additional power components and battery backup units, along with the risk those components and units introduce in the underground environment.

In general, a desirable tracking system will be the one in which its tracking devices are self-directed and can independently locate themselves, as inertial tracking devices might in an underground mine entry network. In the case of a mine accident, when some of the external location references are disabled, the remaining tracking devices should still autonomously function and report their positions to rescuers. The self-directed tracking devices have to heavily rely on the software instead of external hardware to run the positioning algorithms to locate themselves.

This paper introduces algorithms to locate miners with no need for external reference sensors. The algorithms can be used by tracking devices which return their locations in a mine global coordinates. An inertial tracking device needs to convert its location values from its local coordinate system to the mine global coordinate system, which will be defined in the next section, before using the algorithms to find the entry in which it is located or the nearby entries.

The algorithms locate a tracking device in its own or nearby entries if the device is reportedly outside an entry by referring to the mine's entry network data stored in the tracking device's internal computer. Tracking devices can use the algorithms to identify its own or nearby entry paths (between any two adjacent entry intersections) in an area with a lower density of reference sensors to improve and maintain the system accuracy.

The algorithms operate on a tunnel or entry intersection matrix. The matrix is a collection of the locations of all entry intersections from a node-path network model of an underground coal mine described in [1]. The node-path network is a representation of the plane geometrical layout of the entry network of the mine. The algorithms make the use of two techniques, linear coplanar coordinate system transformation and linear distance calculation and comparison among lines to the origin of the coordinate system, to locate the nodes near a tracking device on the mine node-path network.

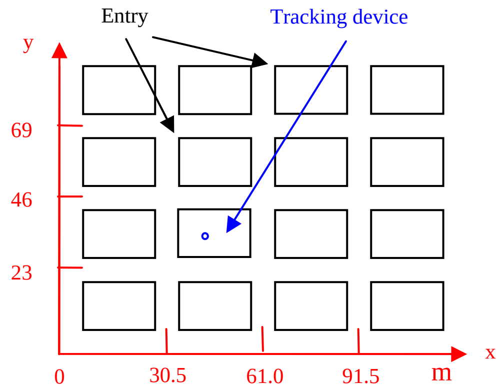
The algorithms take advantage of the homogeneity of the size and shape of underground entries and coal pillars. The coal pillars generally have a rectangular quadrilateral shape as viewed from above. The entry paths are straight and have similar lengths as described in the next section.

This paper begins with a brief introduction of the mine model and then defines useful mathematical operations that can be performed on the node coordinates, and concludes with a description of the algorithms.

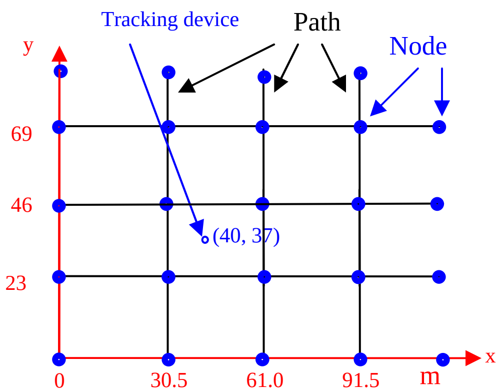
MINE MODEL

The purpose of establishing a mine model is to help to establish a mine global coordinate system for the algorithms. The algorithms operate on a coplanar node-path network defined by the entries and intersections in the mine. The entry centerlines are the network paths which are all straight. The entry intersections are the network nodes. A segment between any two adjacent network nodes represents an entry path. The positions of the nodes at the two ends of a line segment determine the length and orientation of a path. Every entry path and every point in the mine become mathematically locatable by using the mine model.

A tracking device is assumed to be confined to move along the network paths and able to make a turn only at a node. Figure 1 gives an example of mine's network. A portion of a mine is shown in Figure 1 (a) with a coplanar coordinate system superimposed on it, and the mine's node-path network is shown in Figure 1 (b). In the algorithms, the endpoints of the dead-end entry paths are all regarded as network nodes. It is assumed that each of the network nodes has the associated attributes to keep the list of its connected nearest-neighbor nodes. The typical dimensions of the network paths are also given in Figure 1 (b). In addition, as shown in Figure 1, a tracking device reports its location in the inside of a solid pillar, and Figure 1 (b) shows its corresponding location on the network. This scenario is not uncommon for an inertial tracking device.



(a) Coordinate system superimposed in part of a mine



(b) The mine's coplanar network

Figure 1: A mine and its coplanar node-path network

DEFINITIONS OF SPECIAL OPERATIONS

The algorithms require some mathematical special operations and expressions that will be defined in this section. These operations facilitate transformation of the coordinate system and provide for nearby network node identification. Once a coplanar coordinate system has been established over a mine's entire node-path system (The choices for the orientation of the orthogonal axes and their origin for the coplanar network are arbitrary,) any point in the mine can be uniquely located with its coordinate values. In the algorithms, *P*(x, y) is used to express a point on the coplanar plane, where x denotes the x coordinate value of the point, and y the y coordinate value of the point. A point can represent a network node, or a location of a tracking device, or an endpoint of an entry path.

The subtraction operation between two points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is defined as the difference of the corresponding x and y values of those two points as shown in (1). This operation effectively transfers the point $P_1(x_1, y_1)$ from its original coordinate system to a new coordinate system with its new origin at (x_2, y_2).

$$P_1(x_1, y_1) - P_2(x_2, y_2) = P_1(x_1 - x_2, y_1 - y_2). \qquad (1)$$

The operation of the absolute value of a point is defined in (2) which computes the square root of the sum of squares of x and y values of a point. The absolute value of a point indicates the straight line distance of the point *P*(x, y) from the origin. The algorithms will use the absolute value to determine the closeness of a point to the origin.

$$|P(x, y)| = \sqrt{x^2 + y^2}. \qquad (2)$$

Point comparison operations are defined in (3). There are three of them. The first is the equality of two points, and it defines that point 1 is considered equal to point 2 only if their absolute values to the same origin are equal, as shown in (3*a*). Two points have the same distance to the origin if they are equal to each other.

The second definition is that point 1 is greater than point 2 only if the absolute value of point 1 is greater than that of point 2 as shown in (3*b*). Apparently, point 1 is farther away from the origin than point 2 if point 1 is greater than point 2.

The third definition is that point 1 is less than point 2 only if the absolute value of point 1 is less than that of point 2 as shown in (3*c*). Point 1 is closer to the origin than point 2 if point 1 is less than point 2.

These comparison operations are the tools which allow us to quantitatively compare the distances of any two points from the origin.

$$P_1(x, y) = P_2(x, y), \qquad \textit{only if } |P_1(x, y)| = |P_2(x, y)|. \qquad (3a)$$
$$P_1(x, y) > P_2(x, y), \qquad \textit{only if } |P_1(x, y)| > |P_2(x, y)|. \qquad (3b)$$
$$P_1(x, y) < P_2(x, y), \qquad \textit{only if } |P_1(x, y)| < |P_2(x, y)|. \qquad (3c)$$

Next, a point matrix is defined. The point matrix is the collection of the points from a mine node-path network, including all of the network nodes or the intersections of the entries. The nodes are all placed in the matrix in their relative positions on the plane. Its general form is shown in (4).

$$\begin{bmatrix} P_{0,n-1}(x, y) & \cdots & P_{m-1,n-1}(x, y) \\ P_{0,n-2}(x, y), & \cdots & P_{m-2,n-2}(x, y) \\ \vdots & & \\ P_{0,1}(x, y), & \cdots & P_{m-1,1}(x, y) \\ P_{0,0}(x, y), P_{1,0}(x, y), \cdots, & P_{m-2,0}(x, y), & P_{m-1,0}(x, y) \end{bmatrix}. \qquad (4)$$

Here, m and n are dimensions of the matrix; m is the first index for column, which corresponds to the x coordinate;

n is the second index for row, which corresponds to the y coordinate. The indexes can be better thought of being the identities of the individual nodes rather than the orderly numbers, and even the actual intersection names can be used to identify the nodes. With the coordinate system chosen to be that as shown in Figure 1(b), for example, the row index (n) starts from the bottom increasing upwards while the column index (m) starts from the left and increases to the right. In this example, the point in the lower left corner of the matrix represents the node which happens to be at the origin of the coordinate system. The matrix has its element points in the same configuration order of the nodes as it has on the mine node-path network. To simplify the expression, a point matrix is sometimes written as $P_{m,n}[\ ]$. For example, all of the network nodes of the mine entries shown in Figure 1 can be expressed in its matrix form shown in (5). As shown in (5), the point at the lower left corner of the matrix is the origin of the network's coplanar coordinate system. The origin, however, can be chosen at any location on the mine path-node network. Some nodes will be in quadrants other than the first quadrant if the origin and the coordinate system are selected differently from that shown (5).

$$P_{5.5}[\ ] = \begin{bmatrix} (0,92), & (30.5,92), & (61.0,92), & (91.5,92) & \\ (0,69), & (30.5,69), & (61.0,69), & (91.5,69), & (122,69) \\ (0,46), & (30.5,46), & (61.0,46), & (91.5,46), & (122,46) \\ (0,23), & (30.5,23), & (61.0,23), & (91.5,23), & (122,23) \\ (0,0), & (30.5,0), & (61.0,0), & (91.5,0), & (122,0) \end{bmatrix}. \quad (5)$$

A least-valued point is defined as the point having the least absolute value among all of the points in a given point matrix. The point comparison operations defined in (3) can be used to compare the points to each other and obtain the point having the lowest absolute value in the matrix. The expression (6) is used for the operation of least-valued point.

$$P(x, y) = \min(P_{m,n}[\ ]). \quad (6)$$

The P(x, y) has no indices with it because it is a non-dimensional point. For example, the least-valued point of the matrix in (5) can be obtained as shown in (7), and it is zero in value, which is the lowest absolute value comparing to all of the other points in the matrix.

$$P(0,0) = \min(P_{5,5}[\ ]) = |P_{0.0}(0,0)| = \sqrt{x^2 + y^2} = \sqrt{0^2 + 0^2} = 0. \quad (7)$$

The above example unveils an important property of a least-valued point. The least-valued point is the origin of the coordinate system if the origin happens to be at the intersection of two entries, or else it is a point closest to the origin if the origin is not located at an intersection.

Furthermore, the operation of multiple least-valued points is defined as a subset of points, from a given point matrix, all having the absolute values less than any of the other points in the matrix. The point comparison operations can be used to compare the points to each other and obtain multiple least-valued points. The expression (8) will be used for multiple least-valued point operation.

$$P_k[\ ] = \min\_\ k(P_{m,n}[\ ]). \quad (8)$$

In equation (8), $P_k[]$ is a subset matrix that contains all of the least-valued points in the number specified with $k = 2, 3, 4 \ldots$ from the original point matrix $P_{m.n}[]$. Clearly, the subset matrix is the collection of the points nearest to the origin. These k points may not have the same distance to the origin, but they are all the shortest distances comparing to the rest of the points in the matrix. For example, for the point matrix $P_{5,5}[]$ in (5), the multiple least-valued matrix $P_4[]$ is shown in (9).

$$P_4[] = \min\_\ 4(P_{5,5}[]) = \begin{bmatrix} (0,23), & (30.5,23) \\ (0,0), & (30.5,0) \end{bmatrix}. \quad (9)$$

Finally, the operation of subtraction of a point from a point matrix is defined. The operation is to subtract the subtrahend point from each of the elements in the matrix. The resulting matrix is equivalent to a linear transformation of every point in the input matrix to a new coordinate system with the new origin at the subtrahend point. According to the point subtraction defined in (1), the subtraction of a point from a point matrix can be written in the form shown in (10).

$$P_{m,n}[\ ] - P(x_t, y_t) = \begin{bmatrix} P_{0,n-1}(x - x_t, y - y_t), \cdots, P_{m-1,n-1}(x - x_t, y - y_t) \\ \vdots \\ P_{0,0}(x - x_t, y - y_t), \cdots, P_{m-1,0}(x - x_t, y - y_t) \end{bmatrix}. \quad (10)$$

ENTRY AND NEARBY ENTRY PATH IDENTIFYING ALGORITHMS

The algorithms locate the entry path for a tracking device if the device happens to fall on that entry path, or all of the nearest entry paths around the device if it shows its location off any entry. In general, an off-entry device in the mine environment may find itself surrounded by entry paths in different numbers at the different areas of a mine. The device may also find that it is completely enclosed by entry paths around it in one area, and partially enclosed in another of the same mine. The algorithms cover only two scenarios. The first one is that a tracking device is completely enclosed by four entry paths as shown in Figure 1. And the second one is that a device has three entry paths on its three sides with one side open as shown in Figure 2. Those two scenarios should cover the majority of cases in underground coal mines. The algorithms can be extended to other scenarios with some additional considerations which will be briefly discussed in the DISCUSSION section.

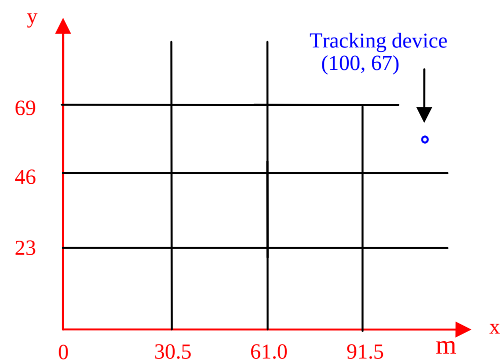


Figure 2: Tracking device has an open side

A. ALGORITHM FOR FOUR ENCLOSED ENTRY PATHS

With four entry paths completely surrounding a tracking device, the algorithm starts by locating the four intersection nodes of those four entry paths. The segments between any two of these adjacent nodes are the entry path centerlines. Figure 3 shows a general case where $P_t(x_t, y_t)$ is the initial position of the tracking device, which is closely surrounded by four nodes, $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$. The following are the steps for identifying those nodes.

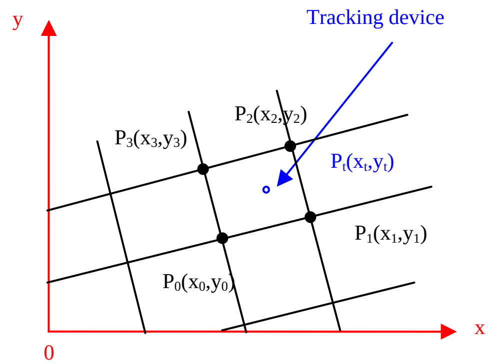


Figure 3: Tracking device reports itself in a coal pillar

1) Put all of the network nodes into a point matrix as shown in (11) to form a tunnel or entry intersection matrix. Four nodes are only explicitly displayed in the matrix for simplicity; the rest of the nodes in the matrix are omitted.

2) Compute the difference between the point matrix $P_{m,n}[]$ and the tracking device location point $P_t(x_t, y_t)$ using the point-matrix-to-point subtraction operation given in (10). The resulting matrix is given in (12).

$$P_{m,n}[\,] = \begin{bmatrix} \vdots \\ \cdots\cdots\ P_2(x_2, y_2) \ldots \\ \cdots\ P_3(x_3, y_3) \cdots\cdots \\ \cdots\cdots\ P_1(x_1, y_1) \cdots \\ \cdots\ P_0(x_0, y_0) \cdots\cdots \\ \vdots \end{bmatrix}. \quad (11)$$

$$P'_{m,n}[\,] = P_{m,n}[\,] - P_t(x_t, y_t) =$$
$$\begin{bmatrix} \vdots \\ \cdots\cdots, P_2(x_2 - x_t, y_2 - y_t), \cdots \\ \cdots, P_3(x_3 - x_t, y_3 - y_t), \cdots\cdots \\ \cdots\cdots, P_1(x_1 - x_t, y_1 - y_t), \cdots \\ \cdots, P_0(x_0 - x_t, y_0 - y_t), \cdots\cdots \\ \vdots \end{bmatrix}. \quad (12)$$

As stated before, this operation effectively transfers all of the nodes from the original coordinate system linearly to a new one with the new origin at $P_t(x_t, y_t)$. Figure 4 shows both the original and the transferred x-y coordinate systems, where the dotted lines x' and y' represent the transferred axes.

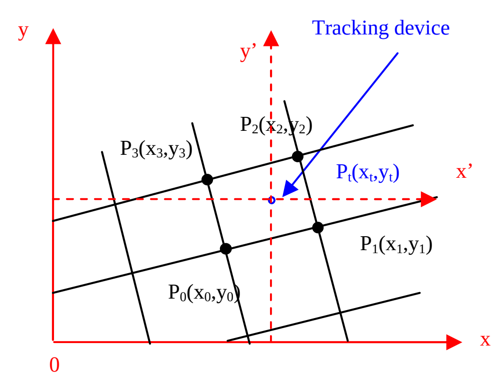


Figure 4: Both original and transferred coordinate systems

3) Use (8) to obtain the four least-valued points around $P_t(x_t, y_t)$ in the new matrix as shown in the expression (13).

$$P_4[\,] = \min\_4(P'_{m,n}[\,]). \quad (13)$$

Clearly, the four nodes, as shown in Figure 4, $P_0(x_0, y_0)$, $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$, will be obtained from the operation because they have the shortest distances to the new origin $P_t(x_t, y_t)$ on the x'-y' coordinate system.

4) Check if the tracking device is actually on one of these four entry paths or inside of the quadrilateral formed by them. The four slope-checking inequalities in (14) can be used for the test, and should all pass if the tracking device is located inside of the quadrilateral.

$$\frac{y_t - y_0}{x_t - x_0} \neq \frac{y_1 - y_0}{x_1 - x_0}. \tag{14a}$$

$$\frac{y_t - y_1}{x_t - x_1} \neq \frac{y_2 - y_1}{x_2 - x_1}. \tag{14b}$$

$$\frac{y_t - y_2}{x_t - x_2} \neq \frac{y_3 - y_2}{x_3 - x_2}. \tag{14c}$$

$$\frac{y_t - y_3}{x_t - x_3} \neq \frac{y_1 - y_3}{x_1 - x_3}. \tag{14d}$$

The inequality (14*a*) examines whether the tracking device is not on the path (line segment) between $P_0(x_0, y_0)$ and $P_1(x_1, y_1)$ while (14*b*), (14*c*) and (14*d*) examine whether it is not on the paths between $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, between $P_2(x_2, y_2)$ and $P_3(x_3, y_3)$, and between $P_3(x_3, y_3)$ and $P_1(x_1, y_1)$ respectively.

If one of these four inequalities fails, the tracking device will be on that entry path. The entry path on which the device is located is, hence, identified. Otherwise, these four entry paths are the device's closely surrounding entries. (In practice, a tolerance may be required for the tests (14) when considering the fact that an entry always has a width.) Consequently, an optimal method can be selected to find the best location of the tracking device on one of these entry paths.

B. ALGORITHM FOR THREE ENTRY PATHS CLOSE TO TRACKING DEVICE

A tracking device may sometimes find itself having only three entry paths on its sides, as shown in Figure 2. This usually happens in a mine's developing zone. This could also happen when some entry paths are sealed and become inaccessible. A network node can find its nearest neighbor nodes and connectivity with these nodes from its associated attributes. If a node finds that it has only one connected neighbor, it is an end node.

Similar to the algorithm to locate the four nearby entry paths around a tracking device introduced earlier, the algorithm to locate the three entry paths starts with locating the four "intersection" nodes of the entry paths around the tracking device. Two of them are the actual intersection nodes of these three entry paths, and the other two, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$, are the endpoints of the entries on the open side as shown, as an example, in Figure 5. In addition to locating the four closest nodes, the algorithms also need to examine whether the device is closer to one of the two entry's endpoints than any of the other entry paths. If this is the case, the device may be assigned that endpoint as its current location.

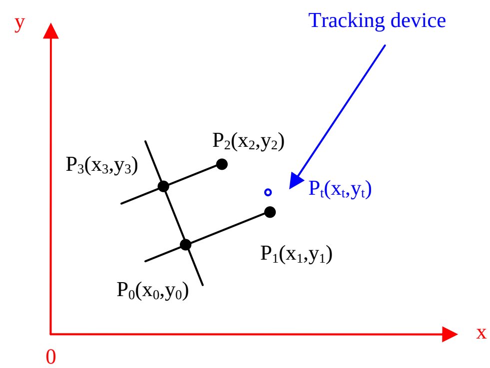


Figure 5: Tracking device is closer to the endpoint of a dead end entry path

The algorithm for identifying three entry paths is slightly different from the algorithm for four entry paths just described.

1) Put all the network nodes and endpoints of the entries into a point matrix to form an entry intersection matrix as shown in (15), where $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ are the entry's endpoints. For simplicity, only the four points are displayed in the matrix in (15) and the rest of the points are omitted.

$$P_{m,n}[\,] = \begin{bmatrix} \vdots \\ \cdots\cdots P_2(x_2, y_2)\ldots \\ \cdots P_3(x_3, y_3)\cdots\cdots \\ \cdots\cdots P_1(x_1, y_1)\cdots \\ \cdots P_0(x_0, y_0)\cdots\cdots \\ \vdots \end{bmatrix}. \tag{15}$$

2) This is the same as step 2 in identifying the four surrounding entry paths introduced early. This step transforms the origin of the coordinate system to the location of the tracking device.

3) This is the same as step 3 in identifying the four surrounding entry paths introduced earlier.

4) Check whether the tracking device is on one of the three surrounding entry paths using the tests given in (16).

If one of the equations in (16) passes within the given ranges of that network path, the tracking device is on that entry path and the device's own entry is, therefore, identified and the process is complete. Otherwise, proceed to the next step.

$$\frac{y_t - y_0}{x_t - x_0} = \frac{y_1 - y_0}{x_1 - x_0}, \quad within$$
$$\min(x_0, x_1) \le x_t \le \max(x_0, x_1) \; and$$
$$\min(y_0, y_1) \le y_t \le \max(y_0, y_1) \qquad (16a)$$
$$\frac{y_t - y_3}{x_t - x_3} = \frac{y_2 - y_3}{x_2 - x_3}, \quad within$$
$$\min(x_2, x_3) \le x_t \le \max(x_2, x_3) \; and$$
$$\min(y_2, y_3) \le y_t \le \max(y_2, y_3) \qquad (16b)$$
$$\frac{y_t - y_1}{x_t - x_1} = \frac{y_0 - y_3}{x_0 - x_3}, \quad within$$
$$\min(x_0, x_3) \le x_t \le \max(x_0, x_3) \; and$$
$$\min(y_0, y_3) \le y_t \le \max(y_0, y_3) \qquad (16c)$$

5) Calculate the real distances from the tracking device at $P_t(x_t, y_t)$ to the three entry paths in the given ranges and the two entry endpoints, $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$; then identify the shortest one among them. If the shortest one is that from the tracking device's initial position to one of the two entry endpoints, select that endpoint as the location of the tracking device. The formulas to calculate the distances from a point to an entry path and from a point to an entry endpoint can be found in many references including [1].

At this point, the three entry paths close to the tracking device have been all identified. A desirable optimal method can be selected to bring the device to a proper entry path among them.

DISCUSSION

To identify the valid network nodes near the location of a tracking device in an area where the entries do not have a uniform layout, some additional steps to the algorithms are needed. Figure 6 gives an example of such a scenario in which the tracking device at $P_t(x, y)$ is in an area where pillars have irregular shapes and the entries have irregular connectivity. The network nodes near the tracking device are apparently $P_0(x, y)$, $P_1(x, y)$, $P_2(x, y)$, $P_3(x, y)$, $P_4(x, y)$, and $P_5(x, y)$.

By choosing k = 4 in the step (3) in the both algorithms, the algorithms will select $P_1(x, y)$, $P_2(x, y)$, $P_3(x, y)$, and $P_4(x, y)$ as the nearest nodes to $P_t(x, y)$ by distance. The nearest nodes that enclose the tracking device, however, should be $P_0(x, y)$, $P_1(x, y)$, $P_4(x, y)$, and $P_5(x, y)$ based on physical connectivity. One of the ways to avoid missing any valid selections is to choose a value of k in the step (3) greater than 4 resulting in selection of more than 4 nearby network nodes by distance, and then down-select among those by physical connectivity. The scenario shown in Figure 6 will be used as an example to describe this additional step. By choosing k = 6, all of these six network nodes should be selected as the "nearest" nodes by distance checking in the step (3) in the algorithms. By checking the physical connectivity, $P_2(x, y)$ and $P_3(x, y)$ will then be eliminated because they are not on the lists of the nearest connected neighbors in the attributes of $P_0(x, y)$, $P_1(x, y)$, $P_4(x, y)$ and $P_5(x, y)$. As a result, $P_0(x, y)$, $P_1(x, y)$, $P_4(x, y)$ and $P_5(x, y)$ will be the only remaining nearby nodes.

Similarly, by selecting k > 4, all of the network nodes having the same distance to the location of a tracking device can be captured, and the invalid network nodes among them can be identified and removed by using the step introduced above.

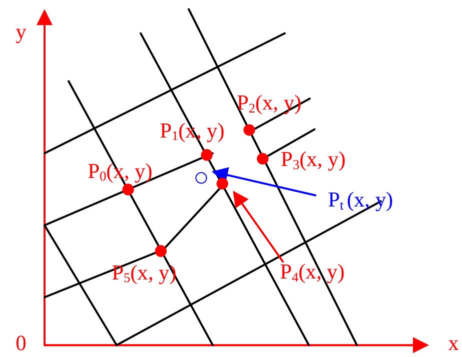


Figure 6: Tracking device is in an area in which the network has irregular connectivity pattern

SUMMARY

The algorithms introduced in the paper can be used to systematically locate a tracking device's own entry path or the entry paths closely around it if the device can provide its coordinates as an inertial tracking device might. It would not be unusual for such a device to report its location as in a coal pillar. An autonomous tracking device, such as an inertial tracking device, can use the algorithms to locate the entry path it is on or nearby if it reports its position in a coal pillar.

A tracking device can use the algorithms to estimate its location in an area with a fewer external location references in underground mines, improving the overall system accuracy of the tracking system. A computer simulation with entries having the orderly grid patterns shows that the algorithms are able to effectively locate a tracking device's own and its closely surrounding entry paths.

REFERENCES

[1] J. Li, D. Snyder, N. Damiano, "Exploration of Two Position Adjustment Method 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