A GIS-BASED SIMULATION MODEL FOR MILITARY PATH PLANNING OF UNMANNED GROUND ROBOTS

M.-W. KANG[1], M.K. JHA[2] & D. HWONG[3]
[1]Department of Civil Engineering, University of South Alabama, AL, USA.
[2]Office of Information Technology, Prince William County, VA, USA.
[3]Department of Information Technology, Prince William County, VA, USA.

ABSTRACT

A GIS-based simulation model has been developed for military path planning of unmanned ground robots in a hazardous combat environment. Critical factors (such as slope of terrain, surface travel distance, degree of bumpiness in various land-uses, and exposure to enemies) that affect the path planning in a combat environment are carefully reviewed and modeled. Line-of-sight analysis and probabilistic decision making process are also modeled for the effective military path planning. In the model, the path planning is dealt with two sub-problems: (i) global path finding and (ii) local navigation. These two sub-problems are recursively processed during the simulation; a temporary least-cost path is calculated in the first stage, and navigation of the unmanned ground robot along the path is then processed in the second stage. The simulation ends when the robot reaches a target location. The proposed model has been applied in different environments with various GIS data. The result shows that the model performs well for the military path planning, and can be useful for testing and evaluating tactical missions to be performed by the robot before its deployment in a real-world situation.

Keywords: decision making, distance transformation, GIS-based simulation, military path planning.

1 INTRODUCTION

1.1 Background and scope

Due to the uncertainties and higher risks of fatality in combat situations, the role of robots (e.g. unmanned ground robot [UGR]) has become an important part of many applications in real world problems. Many robotic units have been developed and are actively being developed for their usage in an outdoor environment over a wide variety of terrains to carry out critical military operations (such as rescuing friendly agents and searching for hostile obstacles).

This study seeks to develop a military path planning method that effectively positions and routes UGRs in a complex battlefield environment. Critical factors associated with the path planning are reviewed and modeled for the system development. A GIS-based simulation platform, which tests and evaluates the proposed path planning method, is also developed. As such, the sensitivity of the result from the path planning method to various UGR capabilities and environments can be analyzed through the computer simulation before its actual field deployment. A variety of GIS data and known/unknown information about the obstacle are employed in the proposed method (as an input) to find the best paths of the friendly UGR as well as to help it safely navigate in the environment.

The proposed path planning problem is dealt with two sub-problems in this study: (i) global path finding and (ii) local navigation. These two sub-problems are recursively processed during the path planning simulation. In the first stage, a temporary least-cost path between start and target locations of the UGR is calculated. Navigation of the UGR in a simulation environment is then processed in the second stage based on the least-cost path found in the first stage.

ISSN: 2041-9031 (paper format), ISSN: 2041-904X (online), http://journals.witpress.com
DOI: 10.2495/SAFE-V1-N3-248-264

1.2 Brief literature review

The robotic units in combat situations often face multiple hazards and complex decisions while carrying out their tactical missions through unfamiliar and hostile battlefield environments. In such cases, they must evaluate and change their plans and positions by processing all available information directly obtained from the environment as well as through the communications with other units and/or the control center. Many decision support systems that fuse the available information and process it for the military path planning have been developed. However, identification of ideal positions, paths, and rates of advance for the military unit is still a very challenging problem due to the complexity of the battlefield environment. Although existing path planning systems (e.g. RIUGV (Remote imagery for unmanned ground vehicles) and Ecognition WAFE with Pathfinder) have proven to work well to some degree in providing feasible routes for robot operations (Frederick et al. [1]), data used in the systems largely rely on known and static information, such as slope of terrain and distance to target. It should be noted, however, that the information of unknown obstacles (such as enemy locations and dangerous areas that may be stationary or dynamically changed over time) and friendly robot's response to the obstacles are not systematically analyzed and modeled in current path planning systems.

Many mathematical models and technologies have been proposed for intelligent robot path planning, uncertainty of surrounding environment, and human–robot interaction and controls. Among them, the most popular, yet challenging one, is the transportation path planning. It is generally defined as the search for paths (from certain points to destinations) that satisfy specified objectives under various conditions. To date, a variety of approaches have been developed to solve the problem, such as genetic algorithms [2, 3], neural networks [4], distance transformation (DT) [5–8], and analytical methods [9].

Through a careful review of the available literature, although genetic algorithms are common for many path planning applications, DT is one of the most popular, fast, and effective methods for searching shortest paths in outdoor environments under different circumstances. This approach not only incorporates complicated terrain information for the path search, but also can handle uncertainty about the target location and obstacles movements. In addition, DT can be applied for a path finding problem with multiple starting points and multidimensional spaces. As such, this study adopts the concept of a DT algorithm to find least-cost paths of friendly UGRs.

The remainder of this paper begins with investigating factors affecting the robot path planning in a hazardous environment. Mathematical formulation for finding the least-cost paths is also discussed in this section. Next, navigation of the friendly UGR is discussed in Section 3. Radial line-of-sight (RLOS) and probabilistic decision making process are modeled in this section. Section 4 describes the input/output of the proposed model with discussion of the path planning simulation. In Section 5, we demonstrate the performance of the model with various simulation examples. Finally, we conclude with the summary of this study and future work.

2 PATH FINDING METHODOLOGY

Many factors (such as slope of terrain, surface friction of land covers, travel distance, exposure to enemies, and dangerous choke points) affect the path planning of the friendly UGRs in hostile environments, and they may possess, partially known, or unknown information. This section discusses the characteristics of such critical factors and develops a method for finding the best paths of the UGR between any points.

2.1 Factors affecting UGR path finding

2.1.1 Slope impact

Slope is generally defined as the maximum rate of change in *z* values (elevation) from one cell to its neighbors. The maximum change in elevation over the distance between the cell and its neighbors identifies the steepest downhill descent from the cell; the lower the slope value, the flatter the terrain; the higher the slope value, the steeper the terrain. Note that the slope value can be calculated using the average maximum technique reported in Burrough and McDonell [10].

The energy consumption of the UGR may vary depending on the slope of terrain on which it travels. If it is moving downhill, its energy consumption will decrease; if it is going uphill, the energy consumption will increase. To represent such slope impacts in estimating the travel cost of the UGR, a vertical factor is introduced in this study. The vertical factor (denoted as F^V) is computed based on the slope between two adjacent points, and can be expressed as ESRI [12]

$$F^V_{i-1,i} = \begin{cases} a_1 \times \theta_{i-1,i} + a_2 & \text{if } \theta_{min} \le \theta_{i-1,i} \le \theta_{max} \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

$$\theta_{i-1,i} = \tan^{-1}\left(\frac{z_i - z_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \right) \tag{2}$$

where

$F^V_{i-1,i}$ = Vertical factor between two adjacent points (λ_{i-1} and λ_i)
λ_i = i^{th} point in an input elevation map (i.e. search space); $\lambda_i = (x_i, y_i, z_i)$
a_1, a_2 = Parameters used for calculating the vertical factor
$\theta_{i-1,i}$ = Slope between two adjacent points
θ_{min}, θ_{max} = Minimum and maximum slopes of terrain that UGR can climb

UGR's hill climbing ability is considered in the proposed model, by introducing minimum and maximum slopes that it can overcome for moving on a hilly terrain. If a slope between two adjacent points falls in between θ_{min} and θ_{max}, the UGR can move forward with the vertical factor value estimated from eqn (1). It cannot travel to locations where slopes are outside θ_{min} and θ_{max} limits, and vertical factor 1 represents a flat terrain. Slope would also be an important factor for finding suitable drop locations of the UGR (if it is air-dropped). Areas of relatively flat terrain are preferred for a safe landing; on the contrary, areas beyond the limit should be prohibited to avoid possible risk of accidents from an unstable landing (Kang et al. [8]).

2.1.2 Surface distance

The surface distance is the actual length of a path along which the ground robot travels in a complex terrain to reach a target. The surface distance is the 3D length of the line segment of the path projected on a surface. It is either more than or equal to the planimetric distance because the surface distance takes into account the variation in *z* values. These variations increase the measured length of all line segments that are not flat. An elevation map (typically in a raster format; e.g. DEM) may be required as a model input to calculate the surface distance. The line segments of the path are sampled for height, using bilinear interpolation, at their vertices plus at a distance no more than the cell size of the input raster. It should be noted that the raster cell size (resolution) affects the calculated surface distance. Decreasing the raster cell size between sampling points allows the line segments to more closely conform to the actual surface.

$$D^{S}_{i-1,i} = \sqrt{\left(x_i - x_{i-1}\right)^2 + \left(y_i - y_{i-1}\right)^2 + \left(z_i - z_{i-1}\right)^2} \tag{3}$$

where
$D^{S}_{i-1,i}$ = Surface distance between two adjacent points.

2.1.3 Bumpiness of surface

The surface of terrain in a hostile environment may consist of various land uses, such as desert, forest, water, built-up area with paved and unpaved roads, or combination of those complexities. Different land uses have different degrees of bumpiness on their surfaces. More fuel consumption may be required for the UGR to travel on a rough and/or bumpy surface. In the proposed model, travel distance over the friction (which compensates the degree of bumpiness on the surface) is considered to realistically represent the fuel consumption of the UGR. A GIS map, which contains various land-use features, is employed in the model to compute the friction factor (denoted as F^F), and it is calculated with an equation expressed as:

$$F^{F}_{i} = \beta_1 + \beta_2 \times \exp\left(L^{U}_{i}\right) \tag{4}$$

where
F^{F}_{i} = Value of friction factor at i^{th} point,
β_1, β_2 = Friction parameters
L^{U}_{i} = Land-use index at i^{th} point (0: open space with flat terrain; 1: agricultural or farmland; 2: desert; 3: forest; 4: built-up; 5: water or swamp; 6: rocky mountain)

Among various land uses in the environment, there may be areas to which the UGR should not travel or at which its landing (if air-dropped) is prohibited. Such land use types as river and enemy facilities should be regarded as ‘No-Go’ areas and removed from the feasible areas of its operation.

2.1.4 Exposure to enemy

Many obstacles (such as mobile and stationary enemies) may also exist in a hostile environment. These obstacles significantly affect the path planning of the friendly UGR, and make its surrounding environment more complex. The locations of some enemies may be identified by UGR in advance; however, many of those are unknown to it. It is important to note that the UGR should avoid its exposure to the enemies if at all possible for a safe and efficient mission completion. We thus consider the UGR exposure to the enemies in the proposed model as one of the important factors for finding the least-cost path, by evaluating enemy locations and their ranges of surveillance in the environment. Some assumptions made for this are:

- The friendly UGR prefers to avoid enemies rather than fighting against them.
- Among a set of enemies distributed in the environment, only the information of known enemies is employed to generate an enemy exposure map with which friendly UGR’s degree of exposure to the enemies is calculated.
- The friendly UGR can detect unknown enemies during the navigation. If any unknown enemy is newly detected, this information is used to reconstruct the enemy exposure map.

$$F_i^E = \sum_{k=1}^{n\left(\mathbf{E}^{\text{known}}\right)} p_i^{e^k} \tag{5}$$

$$p_i^{e^k} = \left[\begin{array}{ll} 1 & \text{if } \; d_i^{e^k} \le R_e^{e^k} \\ 1 - \dfrac{d_i^{e^k}}{R_{max}^{e^k}} & \text{if } \; R_e^{e^k} < d_i^{e^k} \le R_{max}^{e^k} \\ 0 & \text{if } \; R_{max}^{e^k} < d_i^{e^k} \end{array}\right. \tag{6}$$

where

F_i^E = Friendly UGR’s degree of exposure to all known enemies at i^{th} point (λ_i),

$p_i^{e^k}$ = Friendly UGR’s exposure probability to k^{th} enemy at λ_i,

$d_i^{e^k}$ = Visible distance from the UGR to k^{th} enemy at λ_i,

$R_e^{e^k}$ = k^{th} enemy’s effective search limit (measured in distance),

$R_{max}^{e^k}$ = k^{th} enemy’s maximum search limit,

$n(\mathbf{E}^{\text{known}})$ = Total number of known enemies whoes locations are known.

Taking all these considerations into account, friendly UGR’s degree of exposure to the enemies (called the enemy factor) at i^{th} point can be expressed as eqn (5). It is the sum of the UGR exposure probabilities to all known enemies, and can be used for calculating the weighted travel cost with the other factors defined earlier.

We would also consider radar systems of enemy bases designed for the surveillance of friendly air-carriers (if the UGR is air-dropped). The drop location should be outside the surveillance limit, and the areas within the limit should be regarded as ‘No-Drop’ areas, and thus removed from the feasible drop locations.

2.2 Weighted travel cost

Due to the complexity of the environment, finding best paths of the friendly UGR based only on the minimum distance to the target is not sufficient and lacks reality. The best path should guide the UGR to reach the target as quickly as possible and to avoid not only no-go areas but also highly risky regions exposed to enemies. As such, a weighted travel cost function which comprehensively evaluates friendly UGR’s energy consumption (with consideration of terrain slope, surface distance, and friction) and degree of exposure to the enemies is developed here to find the best path. A mathematical formulation that minimizes total travel cost of the UGR under specified constraints can be expressed as eqn (7).

$$\begin{aligned} \underset{\lambda_1,\cdots,\lambda_i,\cdots,\lambda_{n^{IP}}}{\text{Minimize}} \quad C_{path}^T &= \sum_{i=1}^{n^{IP}+1} C_{i-1,i}^T \\ &= \sum_{i=1}^{n^{IP}+1} \left[\left(\omega^S D_{i-1,i}^S\right)\left(\omega^V F_{i-1,i}^V\right)\left(\frac{\left(\omega^F F_{i-1}^F + \omega^E F_{i-1}^E\right) + \left(\omega^F F_i^F + \omega^E F_i^E\right)}{2} \right) \right] \end{aligned} \tag{7}$$

Subject to: $\lambda_{\text{low}} \le \lambda_{\text{i}} \le \lambda_{\text{upper}} \; \forall \; \lambda_{\text{i}} \in \Lambda$

$\lambda_i \notin \Lambda_{NG} \; \forall \; \lambda_{\text{i}} \in \Lambda$

$\theta_{min} \le \theta_{\text{i}-1,\text{i}} \le \theta_{max} \; \forall \; \lambda_{\text{i}} \in \Lambda$

$R_e^{e^k} < d_i^{e^k} \; \forall \; \lambda_i \in \Lambda; \; \forall \; e^k \in \mathbf{E}^{\text{known}}$

where

C^T_{Path} = Total travel cost of the friendly UGR from start point to target;
$C^T_{i-1,i}$ = Weighted travel cost between two successive points of Λ;
ω^S, ω^V, ω^F, ω^E = Weights for D^S, F^V, F^F, and F^E, respectively;
Λ = A set of successive points between the start and target locations;
$\lambda_i \in \Lambda = [\lambda_0,...,\lambda_i,...,\lambda_{nIP+1}]$; Λ is a subset of the search space ($\boldsymbol{O}$);
n^{IP} = The total number of points in Λ;
λ_S, λ_T = Start and target locations of friendly UGR, respectively; $\lambda_S = \lambda_0$; $\lambda_T = \lambda_{\,nIP+1}$;
λ_{upper}, λ_{low} = Upper/lower limits of the search space; $\lambda_{upp} = (x_{upp}, y_{upp}, z_{upp})$, $\lambda_{low} = (x_{low}, y_{low}, z_{low})$
Λ_{NG} = No-go areas in the search space

Four types of constraints are used for the formulation, and these are (i) upper and lower limits of the search space, (ii) No-go areas, (iii) slope constraints, and (iv) enemies' sensor search range. The path finding problem can now reduce to finding a set of successive points, and distance transformation (DT) algorithm is employed to find the best set of points.

2.3 Distance transformation

DT is a simple algorithm that determines a transformed distance between a particular point and target point. It is very popular for solution of path planning with both initially known and unknown field topology, and has been widely used for robot path planning in various environments [5–8]. In DT, the path is generated by following the steepest gradient of the distance transform values from start to target. It can provide efficient path in the presence of obstacles, and the minimum cost obtained by DT is the summation of incremental distances from one cell to another on a grid map. The weighted travel cost function shown in eqn (7) is used to find the least-cost path of the friendly UGR from any point to the target in the search space. DT propagates from the source cell (i.e. target), marking all free cells with an incrementing value. Once all cells that are not prohibited have been marked, a search from a selected start point can be made. If the start point has been marked with a DT value, a path is possible; otherwise a path does not exist.

Various GIS layers are employed to generate a weighted travel cost map, and it is then used to find the least-cost path (i.e. Λ*) of the UGR through the DT process. Each cell of the output map resulting from the DT process stores the least accumulative cost to the target. Thus, a least-cost path from any point to the target can be found by backtracking all least-cost cells from that point to the target. The input and output of the DT process is shown in Fig. 1, and DT process is described as follows:

DT process:

Step1: Target cell is identified.
Step2: Weighted travel cost to each neighboring cell that adjoins target cell is determined.
Step3: Each of the neighboring cells is listed from least costly to most costly.
Step4: The cell with the least cost is removed from the list.
Step5: The least accumulative cost to each of the neighboring cells removed from the list is decided.

The process is repeated until all cells on the output raster have been assigned an accumulative weighted travel cost.

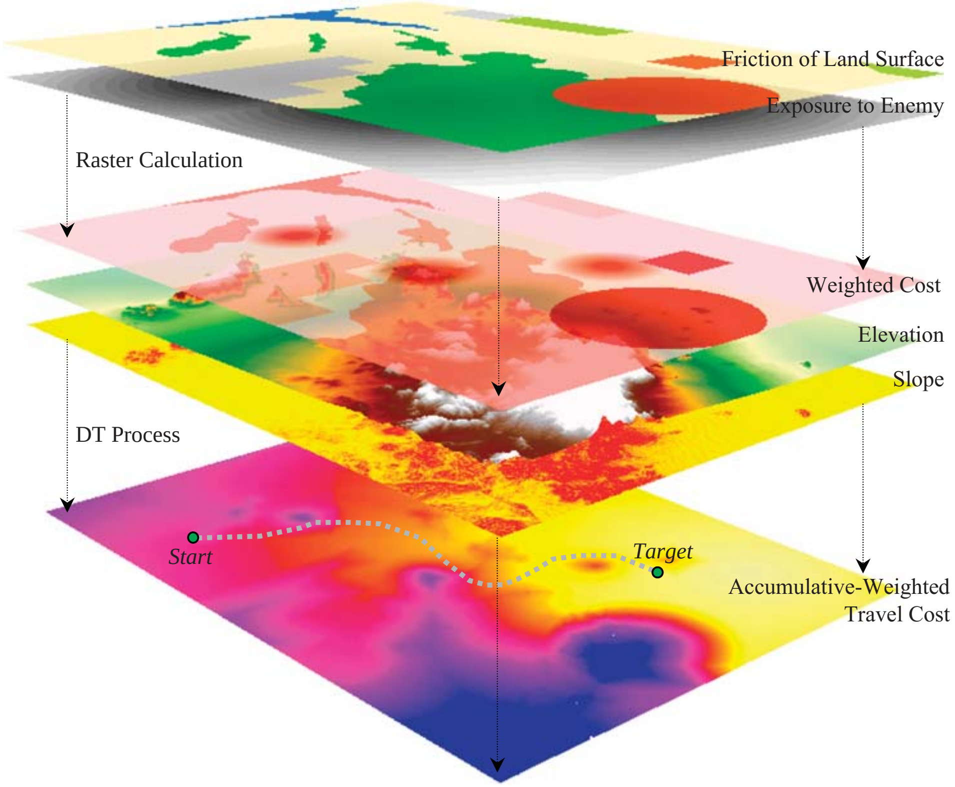


Figure 1: Screen capture of input and output of DT process.

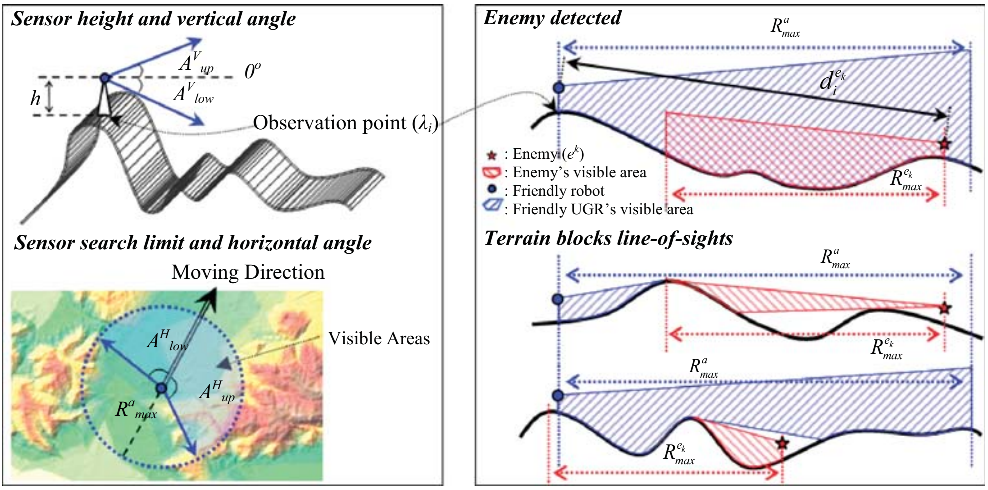
3 NAVIGATING UNMANNED GROUND VEHICLE

Information used for finding the initial least-cost paths of the friendly UGR (e.g. start, target, and known obstacles locations) is limited and may change over time. Thus, the UGR may encounter new situations if it keeps traveling along the path. Navigation of the UGR in the proposed model is decided based on the result of (i) *Radial line-of-sight* (RLOS) analysis and (ii) *Survival probability* to enemies. Enemy information is periodically updated during the navigation with newly detected ones through the line-of-sight analysis. Accordingly, the least-cost path (Λ^*) is recalculated repeatedly with response to the change of its surrounding environment. Note that the start point of the UGR is initially given, and any intermediate point in the least-cost path ($\lambda_i \in \Lambda^*$) can be a start point of a new path during the navigation.

3.1 Radial line-of-sight

During the UGR navigation, the RLOS analysis is repeatedly processed, and through which not only visible areas where it can view from its current location but also distance to the obstacles identified in the visible areas are measured. Key parameters required for the RLOS analysis are as follows:

- *Sensor Height* (h) of the optical sensor equipped on the friendly UGR, and is measured from the ground surface (see Fig. 2(a)).



(a) (b)

Figure 2: (a) UGR sensor parameters and (b) Possible outcomes from radial line-of-sight analysis.

- *Maximum Search Limit* ($R^a{}_{max}$) of the optical sensor from the observer point, for which the visibility is calculated; areas beyond the limit are excluded from the RLOS analysis.
- *Vertical Angle* ($A^V{}_{low}$, $A^V{}_{up}$) defines the sensor search angle above and below the horizon.
- *Horizontal Angle* ($A^H{}_{low}$, $A^H{}_{up}$) defines the azimuth range of the sensor.

Figure 2(b) shows possible outcomes from the RLOS analysis. It shows that depending on terrain complexity, enemies may not be detected although it is within friendly UGR' sensor limit.

3.2 Probabilistic decision making

The friendly UGR should be able to evaluate any perceived information during its navigation, and make a proper decision based on the evaluation result. For example, a decision whether to change its current position, recalculate a new path, and/or fight against the enemy should be effectively made when it encounters any enemy threats during the navigation. For this purpose, a probabilistic decision making process is incorporated in the model. Few assumptions made for this are:

- If the friendly UGR detects an enemy at its current location through the RLOS analysis, the following two events are evaluated, and its survival probability is calculated for each event:
- Event A_i^e: defeat the enemy
- Event B_i^e: avoid the enemy
- Enemies are independent of each other
- If the UGR decides to avoid the enemy, the least-cost path is immediately recalculated; the time lag required for information update and for least-cost path calculation is negligible.

Note that p(A_i^{ek}) and p(B_i^{ek}) can be expressed as a function of (i) killing success rates of friendly UGR and enemies, respectively, (ii) visible distance between them, and (iii) UGR's maximum search limit.

$$p\left(A_i^{e^k}\right)=\left[\begin{array}{ll}\left[(m_a)\times(1-m_a)^{d_i^{e^k}}\right] & \text{if } d_i^{e^k}\le R_{max}^{a}\\ 0 & \text{if } R_{max}^{a}< d_i^{e^k}\end{array}\right. \tag{8}$$

$$p\left(B_i^{e^k}\right)=\left[\begin{array}{ll}1-\left[(m_{e^k})\times(1-m_{e^k})^{d_i^{e^k}}\right] & \text{if } d_i^{e^k}\le R_{max}^{e^k}\\ 1 & \text{if } R_{max}^{e^k}< d_i^{e^k}\end{array}\right. \tag{9}$$

where

$p(A_i^{ek})$ = Friendly UGR's survival probability by defeating k^{th} enemy at λ_i,
m_a = Friendly UGR's killing success rate,
$p(B_i^{ek})$ = Friendly UGR's urvival probability by avoiding k^{th} enemy at λ_i,
= 1 – enemy's probability of killing the friendly UGR,
m_e^k = k^{th} enemy's killing success rate.

3.2.1 Joint survival probability

If multiple enemies are detected through the RLOS analysis, friendly UGR's path planning decision should be determined based on its joint survival probability to them. Assuming that each enemy is independent of each other, the joint probability can be expressed as eqn (10). Note that the total number of possible events (N_i^s) to be evaluated for the joint survival probability is $2^{n(E_i)}$ where $\mathbf{E}_i$ is a set of enemies detected at λ_i. For example, if two enemies are detected at λ_i, N_i^s becomes 4.

$$P_i^{s*} = \prod_{k=1}^{n(\mathbf{E}_i)} Max\left[p\left(A_i^{e^k}\right),\, p\left(B_i^{e^k}\right)\right] \tag{10}$$

where

P_i^{s*} = Friendly UGR's joint probability of survival to all enemies detected at λ_i,
$n(\mathbf{E}_i)$ = Total number of enemies detected at λ_i; $e^k \in \mathbf{E}_i$.

3.2.2 Cumulative survival probability

If the friendly UGR keeps moving in a hostile environment without prior knowledge ahead, its chance of survival will be getting reduced. Assuming that its joint survival probability is independent, for all the intermediate points, of the least-cost path ($\forall\ \lambda_i \in \Lambda^*$), its cumulative survival probability can be expressed as eqn (11). The cumulative survival probability ($\Pr_j^{cs}$) is used as a key measure of effectiveness (MOE) during the navigation for making a proper decision.

$$\Pr_j^{cs} = \prod_{i=0}^{j} P_i^{s*} \quad \forall\ \lambda_j \in \Lambda^* = [\lambda_0, ..., \lambda_j, ...\lambda_{nIP+1}]^* \tag{11}$$

where

$\Pr_j^{cs}$ = UGR's cumulative survival probability until it reaches j^{th} intermediate point.

4 SIMULATION MODEL STRUCTURE

Five GIS input layers are used in the proposed model to realistically represent a hostile environment where the friendly UGR operates for its critical mission. These are (i) initial start points for UGR operation, (ii) targets, (iii) obstacles, (iv) ground elevation, and (v) land-use information. The GIS input can be classified into a static and/or a dynamic entity as shown in Table 1 based on its mobility characteristic.

It is important to note that each GIS input layer requires its own attribute field to represent roles and characteristics during the UGR navigation simulation. For example, the enemy layer (EnemyP) needs five attribute fields (named *ID, Known, Moving, MovePtn,* and *Alive*), which are interactively referenced during the simulation. *Moving* field is used to define the mobility of elements; known and unknown enemies are distinguished with *Known* field; *MovePtn* field is for representing enemy's moving pattern and direction; and finally a matter of enemy's life or death is determined with *Alive* field. Note that only the live enemy of which *Alive* field value = 1 is activated during the simulation, and it is changed to 0 if destroyed by the friendly UGR. Details of all attribute fields that characterize the input GIS layers are presented in Table 2.

The simulation model provides a control loop driven by an external timer to handle the navigation of the friendly UGR and to give the other dynamic entities an opportunity in performing their own processing. The mobile enemy uses this time slice to make a random patrol or scheduled patrol, while the friendly UGR travels from its start point to target. The land use, fixed enemy, and ground elevation layers are not active entities, and therefore they do not require the processing time.

A simulation process for friendly UGR's path planning in a hostile environment is described in Fig. 3. At the beginning of the simulation, parameters which specify the capability of the friendly UGR are initialized. Next, a database storing only known information about the environment is retrieved to find an initial least-cost path from its start point to target. The UGR then begins to move from one point to another along the least-cost path while performing its radial line-of-sight (RLOS) analysis. If no enemy is detected or if enemies are detected but they do not detect the UGR, it keeps traveling along the current path. Otherwise, its cumulative survival probability is compared with a threshold value for making a right decision. If the decision is to avoid the enemy (with a higher survival

Table 1: Property of input GIS layers.

GIS layer	Classification	Type	Fields used for the attribute table	Description
StartP	Static/Dynamic	Point	*ID, Known*	Initial start location of robot
TargetP	Static/Dynamic	Point	*ID, Known*	Initial target location
EnemyP	Static/Dynamic	Point	*ID, Known, Moving, MovePtn, Alive*	Fixed and mobile enemies
SurfaceG	Static	Polygon	*ID, Known, Value, LandUse*	Land-use of the environment
ElevR	Static	Raster	*Value*	Ground elevation

Table 2: Attribute fields describing the input GIS layers.

Field name	Description	Data type
ID	Element ID number	Integer
Known	0 = Unknown; 1 = Known element	Boolean
Moving	0 = Fixed; 1 = Moving element	Boolean
MovePtn	1 = Move in x direction 2 = Move in y direction 3 = Move in xy direction 4 = Random movement	Integer
Alive	0 = Dead; 1 = Live element	Boolean
Value	Raster value that represents Ground elevation (for ElevR) Land-use index (for SurfaceG)	Double
Land use	Land-use type	Text

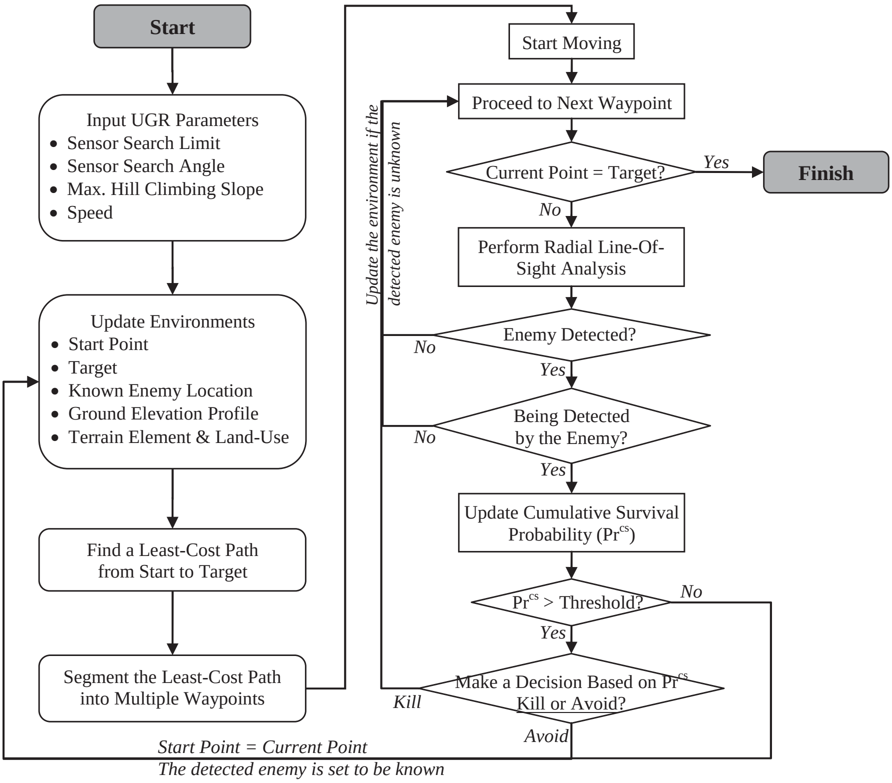

Start
Input UGR Parameters
Sensor Search Limit
Sensor Search Angle
Max. Hill Climbing Slope
Speed
Update Environments
Start Point
Target
Known Enemy Location
Ground Elevation Profile
Terrain Element & Land-Use
Find a Least-Cost Path from Start to Target
Segment the Least-Cost Path into Multiple Waypoints
Start Moving
Proceed to Next Waypoint
Current Point = Target?
Yes
Finish
No
Perform Radial Line-Of-Sight Analysis
Update the environment if the detected enemy is unknown
Enemy Detected?
No
Yes
Being Detected by the Enemy?
No
Yes
Update Cumulative Survival Probability (Pr^cs)
Pr^cs > Threshold?
No
Yes
Make a Decision Based on Pr^cs Kill or Avoid?
Kill
Avoid
Start Point = Current Point
The detected enemy is set to be known

Figure 3: Simulation flow chart for military path planning.

probability than the threshold value), the least-cost path is recalculated. When recalculating the path, the current location of the friendly UGR is set to be the start point of the new path, and any new information obtained until this point is used for updating the UGR database. For instance, if the detected enemy is unknown, then its property is changed to be known, and thus the UGR can keep track of its movement during the navigation. If the decision is to defeat the enemy, the enemy operation is deactivated as if it is killed, and the UGR keeps traveling along the current path without path recalculation. The simulation program ends when it reaches the target.

5 EXAMPLES

Several case studies have been conducted to demonstrate how the proposed simulation model works in different environments. Real GIS data for various terrain and land-use maps (obtained from USGS Seamless database for several mountainous areas in Afghanistan) are employed to construct different examples, and some hypothetical obstacles are added to make the example more complex. UGR parameters used for the test are the hill climbing ability and sensor height.

5.1 Hill climbing ability

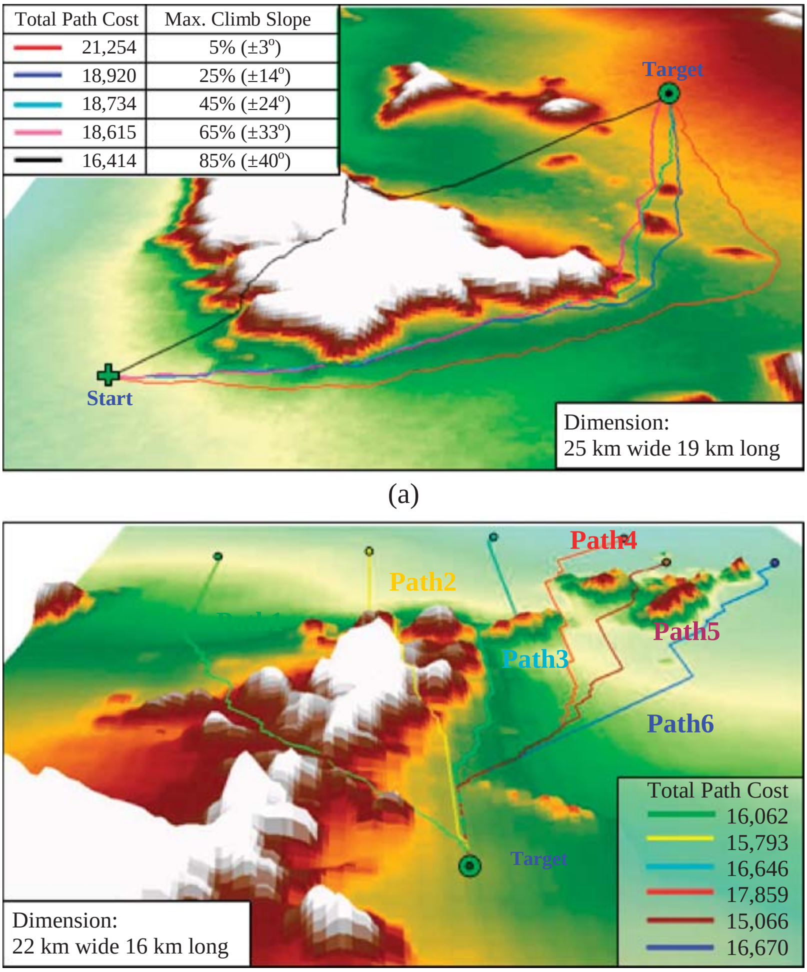
Recall that the least-cost path of the friendly UGR is determined based on the comparison of elevation difference between two neighboring points and its ability to move on those points. Thus, if the hill climbing ability (i.e. maximum slope that the UGR can overcome) is higher than the slope between its current and next points, it then can move to the next point. Figure 4(a) shows least-cost paths of the friendly UGR in a hilly terrain with different maximum climbing slopes. As shown in the figure, the UGR can cut across the hilly terrain with ±40° of the climbing slope to reach the target location, and thus it is the shortest among the alternatives described in the figure. However, with ±3° of the climbing slope it is observed that the UGR travels only through the plain area, producing the longest travel distance.

5.2 Least-cost paths from multiple start points

Assuming that there is no enemy in the simulation environment and that the mission of the friendly UGR is to reach a known target, Fig. 4(b) shows least-cost paths to the target from multiple starting points. Note that the maximum climbing slope that the friendly UGR can overcome is set to ±14° (25%) in this example. As shown in the figure, all the six paths completely avoid the high slope regions in mountainous areas and are relatively straight to the target. The result indicates that the surface distance and slope factors are simultaneously evaluated to find the least-cost path.

5.3 Sensor capability

The sensor capability of the friendly UGR is also very important for its safe and effective navigation in a hazardous environment. This is because the UGR depends most likely on the sensor for monitoring and detecting hostile obstacles in its surrounding environment. With a low sensing capability, the friendly UGR would be able to detect the obstacles only close to it. On the contrary, with a high sensing capability it can detect the objects that are much far from it.



(a)

(b)

Figure 4: (a) Least-cost paths with different hill climbing abilities and (b) Least-cost paths from different start locations to the target.

Figure 5(a) shows points (along the initial least-cost path) where an unknown enemy is identified for the first time by the friendly UGR with different sensor heights. As shown in the figure, the UGR is able to detect the unknown enemy much earlier with a higher sensor height and change its route to a recalculated least-cost path. With a low sensor height, however, it would not be able to detect the enemy although the enemy is already within the sensor search limit. Such a case is encountered when a terrain blocks its line-of-sight. For example, at the *72th* intermediate point (λ_{72}) of the initial least-cost path, the friendly UGR cannot detect enemy's existence with the sensor height of less than 50 meter due to the blockage of a terrain in between them. The UGR with 10 meter sensor height cannot detect the enemy until it reaches λ_{109} where it would also be exposed to the enemy whose sensor search limit is 600 m. Figure 5(b) shows line-of-sight diagrams at the corresponding enemy detection points.

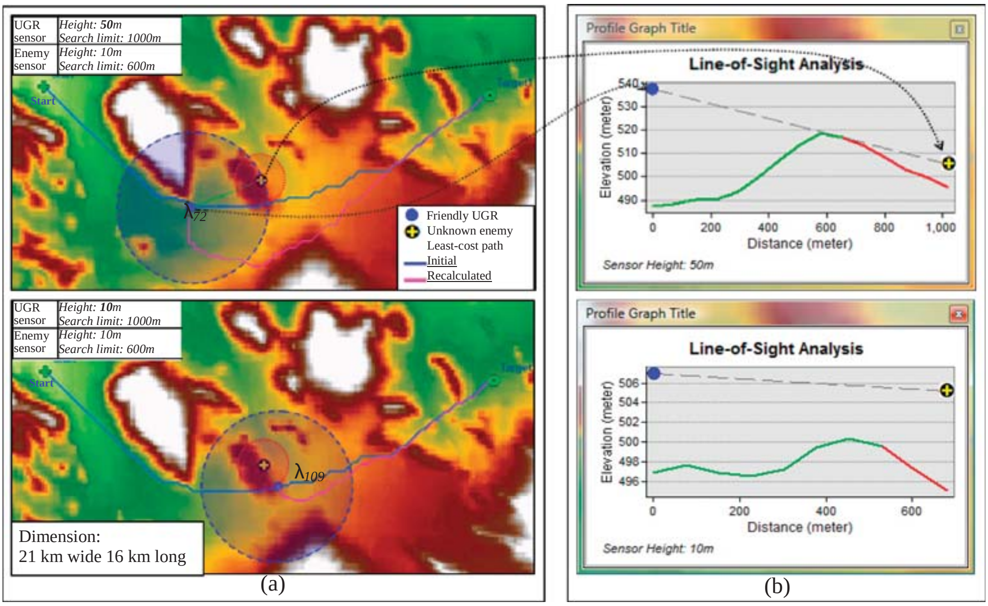


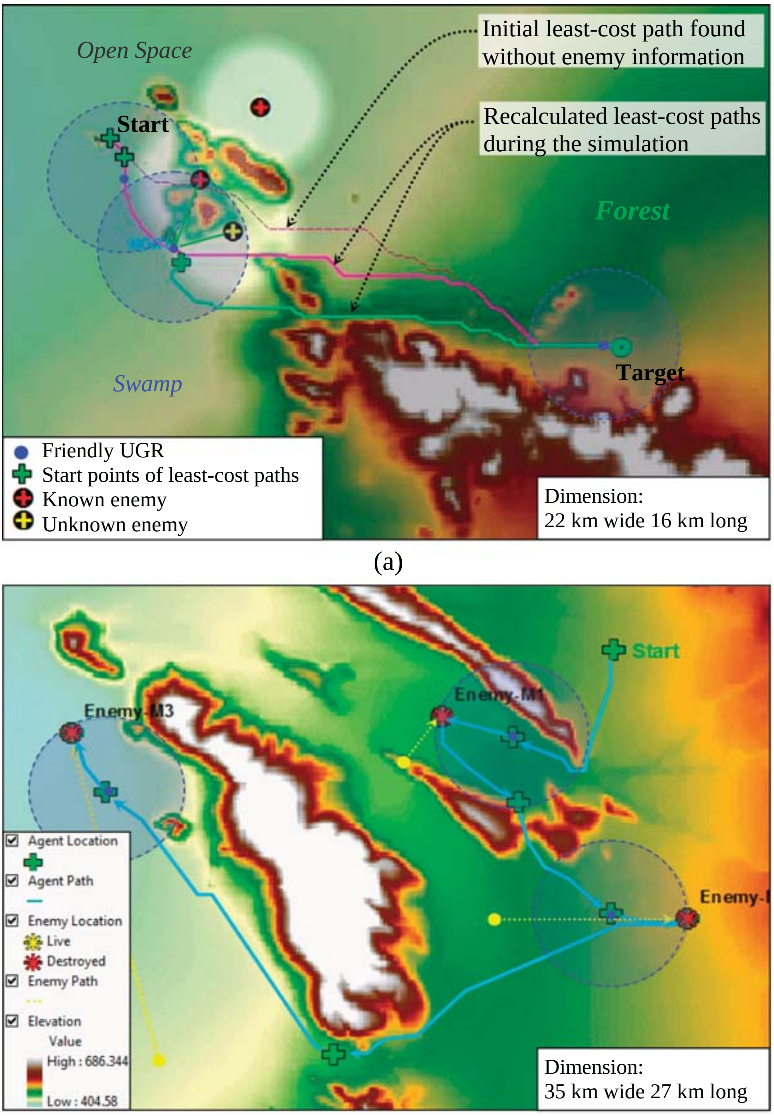
Figure 5: (a) Location where the friendly UGR first detects unknown enemy with different sensor heights and (b) Corresponding line-of-sight at the enemy detection point.

5.4 Path planning in a static environment

Suppose that a mission assigned to the friendly UGR is to reach a known target as quickly as possible, and the simulation environment where it operates is static (see Fig. 6(a)). At the beginning of the simulation since no enemy information is given, UGR's initial least-cost path is relatively straight and running through low elevation and land-use friction areas. However, once the locations of known enemies (⊕) are identified, its initial path is recalculated to have a new one that guides the UGR in a much safer way without exposure to the known enemies. While traveling along the recalculated path, the UGR detects an unknown enemy (⊕) at a certain point through its RLOS analysis, and thus the path is recalculated again to avoid engaging with the enemy. If additional enemies are not further detected, the UGR keeps moving along the new path until it reaches the target.

5.5 Path planning in a dynamic environment

In this example the target of the friendly UGR is a moving object, and its surrounding environment is time-varying where the elements of the environment change over time. Some large and small hilly mountains are located in the middle of the simulation environment, and three mobile enemies are initially placed in a southwest region of the friendly UGR. The mission assigned to the UGR is to chase and destroy the enemies distributed in the environment. Thus, calculation of least-cost paths, enemy movement, and UGR's exploration to catch the enemy are iteratively processed during the simulation. Figure 6(b) shows trajectories of the moving objects during the simulation. When the simulation begins, the friendly UGR first chases Enemy-M1 (i.e. target 1) because it is the closest from the UGR. After the first target


Open Space
Initial least-cost path found without enemy information
Recalculated least-cost paths during the simulation
Start
Forest
Swamp
Target
● Friendly UGR
✚ Start points of least-cost paths
✚ Known enemy
✚ Unknown enemy
Dimension:
22 km wide 16 km long

(a)

(b)

Figure 6: Path Planning for (a) Reaching a stationary target in a static environment and (b) Chasing multiple mobile enemies in a dynamic environment.

is destroyed (i.e. deactivated), the UGR moves for the next closest targets (Enemy-M2 and Enemy-M3 sequentially). The simulation runs until all the enemies are destroyed.

In most of the simulation runs, it has been observed that the friendly UGR catches all the enemies since the UGR is assumed to be faster than the enemies. However, if the UGR speed is less than or equal to that of the enemies, it may not be able to catch all the enemies unless it has intelligence to predict their movements in advance. The path planning method proposed in this study will be further improved with consideration of variable speeds and intelligent behaviors of moving objects in a future study.

6 CONCLUSIONS AND FUTURE STUDIES

This study developed a GIS-based path planning model that finds best paths of mobile robots (i.e. UGR) and simulates its mission under various conditions. A weighted travel cost function with distance transformation (DT) algorithm is developed to find the best paths, and a probabilistic decision making process based on a radial line-of-sight (RLOS) analysis is modeled for a safe robot navigation in a hostile environment. The path finding and navigation are iteratively processed during the simulation of robot path planning. The simulation model has been applied in different examples. Various GIS data that describe terrain, land uses, targets, and obstacles of the examples are employed in the model as an input. The result shows that the model performs well in the transportation path planning under various conditions, and that it has great potential for testing and evaluating various missions to be performed by the robot before its deployment in a real-world situation. Despite demonstrated capabilities, the model can benefit from many technical and methodological improvements. Some future works to be considered are:

- Model speeds of mobile robots based on time-varying information and mission characteristics.
- Model intelligence of mobile robots to predict obstacles movement.
- Model cooperation between friendly robots against threats from hostile obstacles.
- Develop an experience-based decision making process in which military units faced with new decisions can act on the basis of experiences gathered from previous operations.
- Develop a cognitive hierarchy model through which meaningful knowledge can be obtained from available information.

ACKNOWLEDGMENT

This work is funded by the Test and Evaluation/Science and Technology (T&E/S&T) Program, Test Resource Management Center (TRMC) of the Department of Defense through the Scientific Research Corporation (SRC). The authors thank Mr. Shannon Arnold of Office of the Secretary of Defense (OSD) TRMC S&T/T&E Program for the support.

REFERENCES

[1] Frederick, P., Kania, R., Rose, M.D., Ward, D., Benz, U., Baylot, A., Willis, J. & Yamauchi, H., Spaceborne path planning for unmanned ground vehicles (UGVs), *Proc. of Military Communications Conference (MILCOM 2005)*, IEEE, Atlantic City, NJ, 2005; http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01606139 (accessed 2010).

[2] Geisler, T. & Manikas, T.W., Autonomous Robot Navigation System Using a Novel Value Encoded Genetic Algorithm, *Proc., 45th Midwest Symposium on Circuits and Systems*, **3**, pp. 45–48, 2002.

[3] Sedighi, K.H., Ashenayi, K., Manikas, T.W., Wainwright, R.L. & Heng-Ming, T., Autonomous local path planning for a mobile robot using a genetic algorithm, *Proc., Congress on Evolutionary Computation*, **2**, pp. 1338–1345, 2004.

[4] Yang, S.X. & Luo, C., A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **34(1)**, pp. 718–724, 2004.

[5] Wang, H., Chin, Y.T., Tay, L.P., Wang, H. & William, Y.C., Vision guided AGV using distance transform, *Proc., 32nd International Symposium on Robotics*, pp. 19–21, 2001.

[6] Marzouqi, M. & Jarvis, R., Covert Path Planning in Unknown Environments with Known or Suspected Sentry Location, *Proc., IEEE/RSJ International Conference on*

Intelligent Robots and Systems, pp. 1772–1778, 2005. doi:http://dx.doi.org/10.1109/IROS.2005.1545483

[7] Taylor, T., Geva, S., & Boles, W., Directed Exploration Using a Modified Distance Transform, *Proc., Digital Imaging Computing: Techniques and Applications*, pp. 365–370, 2005.

[8] Kang, M.W., Jha, M.K. & Karri, G.A., Determination of Robot Drop Location for Military Path Planning using GIS Application, *Proc., 4th WSEAS International Conference on Computer Engineering and Applications*, pp. 194–200, 2010.

[9] Hwang, J.Y., Kim, J.S., Lim, S.S. & Park, K.H., A Fast Path Planning by Path Graph Optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, **33(1)**, pp. 121–129, 2003.

[10] Burrough, P. & McDonell, R., *Principles of Geographical Information Systems*, Oxford University Press: New York, 1998.