ARMY RESEARCH LABORATORY

Multimap Procedures for Robot Route Finding in Open Terrain

by Aivars Celmiņš

ARL-TR-1897

February 1999

Abstract

An important task for scouting robots in open terrain is the planning of a route to a prescribed destination with the help of digital maps. In an open environment, available terrain maps usually will be too large for a timely analysis of a whole map and may cover areas that are not relevant for route planning. One needs, therefore, route-planning procedures that use partial terrain maps. This report presents two such procedures that use sets of partial maps. A first method uses a series of vicinity maps that are aligned with the position of the robot; a second method uses sets of telescopic maps that follow the robot. In both procedures, the routes are determined using a navigation function method based on Huygens' principle of wave propagation. Examples of results by the two methods are presented and compared. It is concluded that the vicinity map method is faster, but the telescopic map method is more robust and its results are less sensitive to the choices of sizes and resolutions of partial maps.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

The cybernation of battlefield scouting robots and a process for finding routes in a terrain map have been described in a previous report (Celmiņš 1997). In that report, the feasibility of the route-finding process was demonstrated without considering problems that can arise when the algorithms are used by robots in a real battlefield. In particular, it could be shown that the described process provides a complete solution of the route-finding problem if the terrain maps that are available to the robot precisely and completely describe the environment. (A complete solution in the context of robot navigation is defined as a solution that always produces the desired route if the route exists and signals the nonexistence of a solution.) Such environments are typical for laboratories that test robot motion. Real-life battlefield robots must, however, operate in more general environments.

In this report, we expand the applicability of the route-finding process to open-field environments. We consider the following navigation task. A robot is provided an approximate terrain map, the coordinates of a destination, and the current coordinates of the robot. The robot is asked to determine from the map a route with the shortest travel time and to navigate approximately along that route. As the robot proceeds along the intended route, it receives new information about the terrain either from its sensors or from other sources (e.g., from collaborating robots). The robot is expected to use such new information for real-time adjustments and modifications of the planned route.

The most important feature of the stated open-terrain problem (in contrast to robot motion in controlled, finite environments) is the requirement for real-time reaction to changing terrain information. In practical terms, this requirement means that the route needs to be planned precisely only in a neighborhood of the robot since the environment might change by the time when the robot reaches distant locations. One can exploit this feature by representing the terrain in the neighborhood by a precise map and using low-resolution maps beyond the neighborhood. By using low-resolution maps or simplifying assumptions about the terrain at distant locations, computing time for terrain analysis can be reduced. (Some analysis of the terrain up to the destination is always needed; information about the immediate neighborhood is not sufficient for route planning.)

It should be obvious that any navigation algorithm that solves the described problem can also be used for explorations of completely unknown terrains and for fast estimates of approximate travel directions. By using approximate maps in parts of the terrain, the algorithm loses its "complete solution" property, but this should be of no concern for practical navigation in a battlefield. Also, for navigation in a dynamically changing environment the concept of a "complete solution" is meaningless.

The two solutions presented in this report are variations of the route-finding method described by Celmiņš (1997 and 1998). In that method, one first computes a navigation function for the given terrain map (called Huygens' relief) by making use of Huygens' principle in optics and then determines stepwise the desired route by a local algorithm on the navigation function. In the new methods that are presented in this report, details of the navigation function are computed only in a neighborhood of the robot. Consequently, a precise route is provided only in that neighborhood. The route beyond the neighborhood need not be computed; it suffices to provide a general direction for the robot's path with the help of rough approximations of Huygens' relief. A fast computation of an approximate Huygens' relief outside the neighborhood region can be achieved by using a simplified view of the terrain. In a first approach presented here, the terrain outside the precise vicinity map is assumed to be homogeneous with uniform properties. In the second solution, the terrain at distances from the robot is represented by raster maps whose cells increase in size as the distances from the robot become larger. Both solutions are significantly faster than the finding of a detailed route in a complete terrain map.

In section 2, we give a short overview of the algorithm from Celmiņš (1997). Sections 3 and 4, respectively, contain descriptions of the two proposed solutions, and section 5 is a summary and conclusions section.

2. Navigation With Huygens' Relief

The idea of using a navigation function for route finding was first introduced by Rimon and Koditschek (1988) as an alternative to potential function methods (Khatib and Le Maitre 1979) for navigation in a homogeneous environment with impenetrable obstacles. The function was assumed to have a unique minimum at the destination and constant positive values at the boundaries of the obstacles. In such a navigation function, a route to the destination that avoids all obstacles can be found by steepest descent. In an inhomogeneous open terrain, where different navigable areas can be negotiated with different speeds, the navigation function must be generalized such that the navigation speed is properly taken into account. Also, the constant-value condition at obstacle boundaries is not essential and can be deleted. One function that satisfies these conditions and can be used for navigation in open terrains is a is a function whose value at every point of the space equals the arrival time of a signal from the destination, whereby the signal propagation speed equals the navigation speed. We call such a function a Huygens' relief because it can be efficiently computed with an algorithm that is based on Huygens' principle of wave propagation in optics. In a given Huygens' relief, the path with the shortest travel time can be found by steepest descent.

A possible realization of the Huygens' relief method for battlefield robots is described by Celmiņš (1997 and 1998). For battlefield navigation, the terrain map is presented in a raster form where each cell has assigned to it the corresponding average navigation speed. The robot's location is assumed to be always in the center of a cell. This means that the navigation space is granulated and the calculated route is defined by a list of cell-center points. In a granulated plane, Huygens' relief can be calculated by a very simple algorithm as follows. First, all cells that are not source cells (i.e., are not destination cells) are assigned a sufficiently large value (H_{large}) that exceeds the possible maximal signal arrival time. The source cells are assigned zero signal arrival time. Next, a preliminary relief value is computed for every receiving cell by assigning to the cell the smallest among the eight signal arrival times from the cell's eight neighbor cells. These calculations are repeated by sweeping over all cells until convergence is achieved. By properly arranging the direction and sequence of sweeps, their number can be held down to less than 10 in most practical applications. The resulting relief has, by its calculation method, minima with zero values only at the destination cells and it has no relative minima. To find a route with the shortest travel time in a granulated Huygens' relief, one has to chose as the next position along a route the center of one of those neighbor cells that is a signal source for the current position cell (instead of going the steepest descent route that would be adequate for a continuous relief). Neighboring source cells are found by a simple local algorithm that recalculates the arrival times from each of the eight neighbor cells. If more than one neighbor cell is source, then a tie-braking algorithm is used, which alternatively chooses the leftmost and rightmost source cell, respectively.

An example of route finding by this method is shown in Figures 1 and 2. Figure 1 shows a schematic terrain map in a 300 × 300 grid resolution. The initial position of a robot is indicated by a circle and the destination by a square. Different shades indicate areas with different navigation speeds: the black area has a very low speed, indicating an impenetrable obstacle, and the light lines indicate roads with high navigation speeds. The indicated route between the initial position and the destination is a route with shortest travel time. To determine the route, a Huygens' relief corresponding to the terrain map was calculated first. The relief is shown in Figure 2, where the contour lines correspond to constant signal arrival times. The contour lines in open-field areas of the map are octagons instead of the expected circles because of the granulation of the terrain. (It is assumed that signals can propagate from any cell only in the eight directions to its eight neighbor cells.) Next, the robot's route was determined by the described local algorithm in the relief map. Plotting the computed route in the input terrain map (Figure 1) shows that the route with the shortest travel time indeed makes use of available roads and avoids low-speed areas.

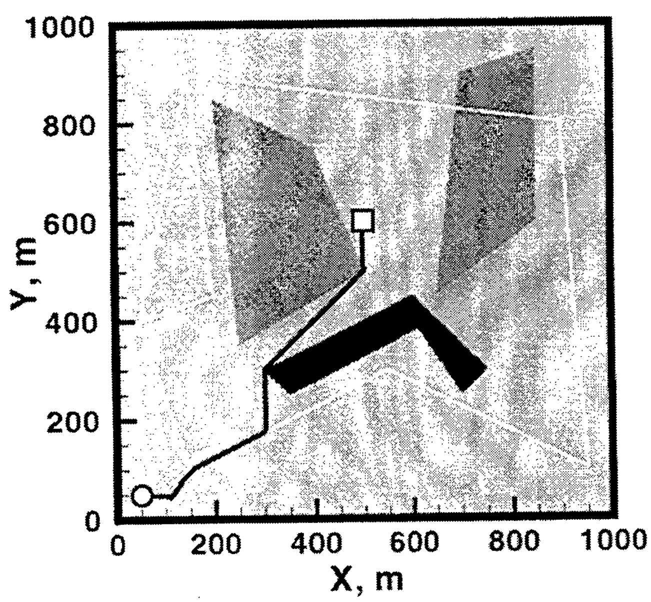


Figure 1. Terrain Map.

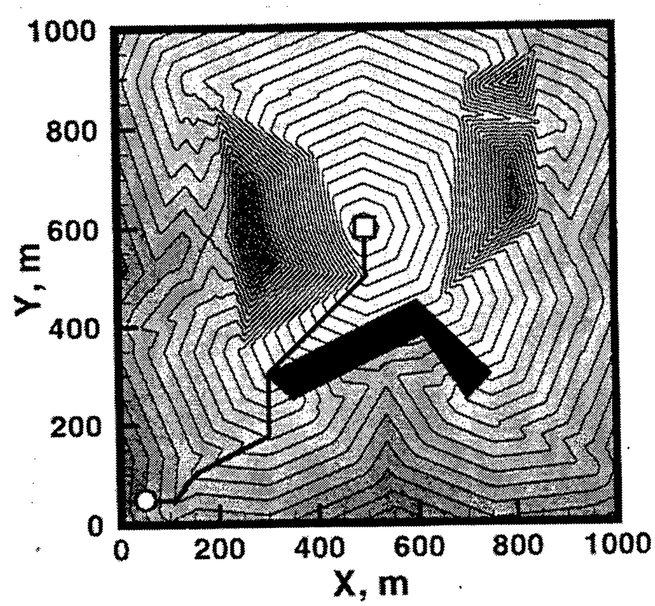


Figure 2. Huygens' Relief.

The described route-finding method is useful for moderate-size input terrain maps and static terrains, but it can be too slow otherwise. First, the computation time for Huygens' reliefs increases in proportion to the number of cells and might not be negligible for maps that contain more than about 200 × 200 cells. Some computing time could be saved by clever data storage and processing, but, in principle, large maps with high resolution present a problem. Second, whenever the terrain information is updated (for instance, when the destination moves or when new terrain information is received), Huygens' relief must be recalculated for the whole map. Even for moderate map sizes, this can cause computing times that are too long for real-time decisions. In the next two sections, variations of the Huygens' relief method that help to economize the computing time for route planning are presented.

The computation of the relief in Figure 2 takes about 27 s on a workstation computer. However, the computing time is proportional to the number of cells in the map and solving the same problem, if a map with 600 × 600 cells is used, requires about 74 s of computing time. For more complicated environments that contain maze-type obstacles, the computation of Huygens' relief can take even more time. On the other hand, the finding of a route in a given Huygens' relief with the same high resolution requires less than 1 s of computing time. This example shows that, in dynamically changing environments where the relief must be repeatedly recalculated, the computing times can become too long for real-time decisions unless smaller maps or coarser grids are used. It also indicates that the overall computing times can be most efficiently reduced by reducing the computing times for Huygens' reliefs.

The simplest way to reduce computing times is to use maps with coarser resolutions. We illustrate the effects of coarser resolution in the next example for which we use the same terrain as in the first example but reduce the number of cells from 300 × 300 to 50 × 50. Figures 3 and 4 show the results. The computing time for the coarse grid example was less than a fraction of a second. On the other hand, because we now have a coarser granulation of the terrain, only the general shape of the route is the same as in the first example. However, in applications with dynamically changing environment, the general shape suffices to start the robot's travel in the right direction (along the road in this case). A precise definition of the route is needed only in the immediate vicinity,

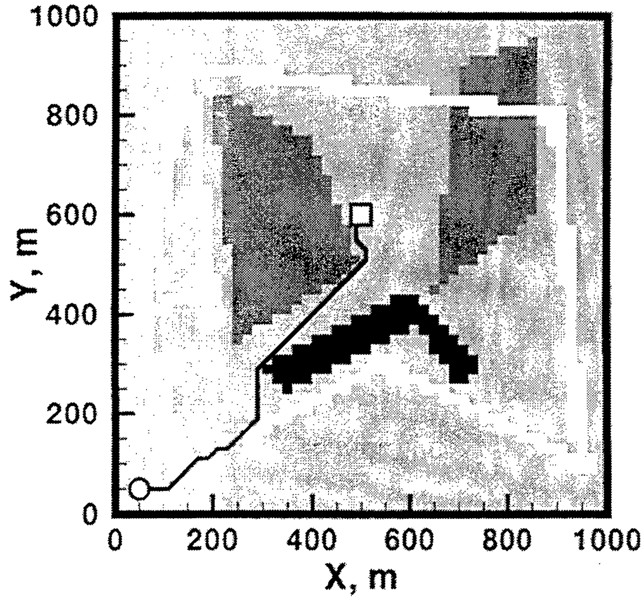


Figure 3. Coarse Terrain Map.

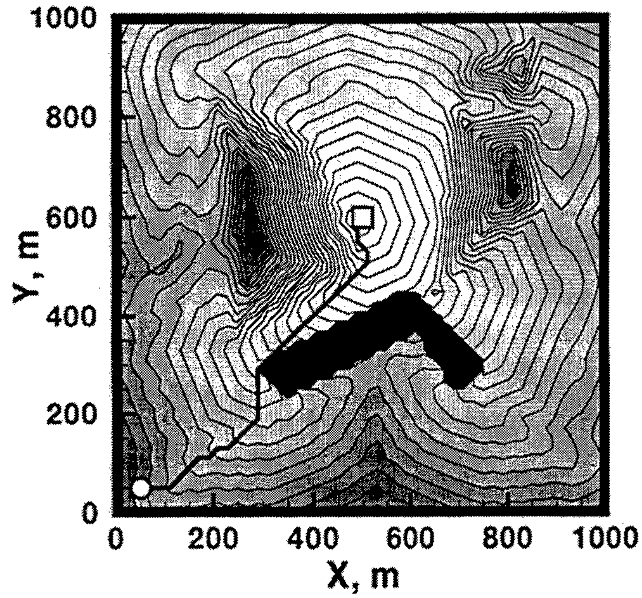


Figure 4. Coarse Huygens' Relief.

defining, for instance, the width of the road, ditches along the road, and possible crossings of a ditch to reach the road. Therefore, computing time can be reduced by providing the robot with a general indication of travel direction and a precise definition of the route in a neighborhood of the robot. The general direction of travel can be obtained either by using a coarse map or by some other simplifying assumptions.

3. Vicinity Map Method

In the "vicinity map method" that is presented in this section, the general direction toward the destination is obtained by assuming that the terrain outside the vicinity map is uniform and has a constant navigation speed. The corresponding Huygens' relief values are computed in closed form for the boundaries of the vicinity map. Inside the vicinity map, Huygens' relief is determined as described in section 2, except that the boundary cells now have predetermined initial values that are less than H_{large}. Once Huygens' relief has been calculated, the robot's route is determined within the vicinity map up to a point in a boundary zone of the map. (The boundary zone for this method is defined by distances to the map boundary that are less than the sensing radius of the robot; if the robot has no sensing devices, then the boundary zone is assumed to be two cells wide.) When the robot reaches the boundary zone, a new vicinity map that is centered around the present position of the robot is prepared and the process repeated.

If the destinations are outside the vicinity map, then the computations are done as follows. To start the calculations, initial values in all cells of the vicinity map are set equal to a large number H_{large}. Next, signal arrival times are computed for all boundary cells assuming a constant navigation speed v_{out} along straight lines between the boundary cells and the destination. The outside navigation speed v_{out} is determined as the average speed within the vicinity map. This procedure provides for the Huygens' relief calculation in the vicinity map initial boundary values with a gradient toward the destination. This simple calculation of boundary values is adequate as long as the destination is not very close to the vicinity map (in terms of the map size). To have proper initial boundary values also for destinations that are very close to the map boundary, the following modified boundary value formula is used. Let D_{ij} be the distance between the cell center of a boundary cell (i, j) and the destination, d_{ij} be the cell size, and v_{ij} be the navigation speed in the cell. Then the initial boundary value of Huygens' relief in the boundary cell is computed by

$$H_{ij} = (D_{ij} - d_{ij}/2)/ v_{out} + d_{ij}/(2\ v_{ij}). \qquad (1)$$

In cases with several destinations outside the vicinity map, the signal arrival times from all such destinations are computed with equation (1) and the smallest arrival time among the calculated values is chosen as the final value for the boundary cell.

If a destination is inside the vicinity map, then there will usually be no need for Huygens' relief outside the map and, therefore, initial boundary values need not be computed. Exceptions might be cases where the selected size of the map is too small for the problem so that the destination is separated from the robot by impenetrable obstacles that extend over the whole width of the map and all possible routes are partially outside the vicinity map. In such cases, the proper approach is to increase the size of the vicinity map or to use the telescopic map method described in section 3.

The initial relief value of source cells in the vicinity map is zero. The initial relief values of other interior cells (that are not source or boundary cells) is set equal to a high value H_{large} that exceeds all possible signal arrival times for the problem at hand.

Next, the final relief values are calculated iteratively by sweeping over the vicinity map. In this iteration, the value at the center point of each interior nonsource cell is determined as the lowest signal arrival time among the arrival times from all eight neighbor cells. The formula for signal arrival time is as follows. Let x_{ij} and y_{ij} be the coordinates of a cell center, v_{ij} be the navigation speed in the cell, and H_{ij} be the signal arrival time. The arrival time of a signal that arrives in the cell (i, j) from a neighbor cell (k, l) is

$$H_{ij}^{kl} = H_{kl} + 0.5\ (1/v_{kl} + 1/v_{ij}) \left((x_{kl} - x_{ij})^2 + (y_{kl} - y_{ij})^2 \right)^{1/2}. \qquad (2)$$

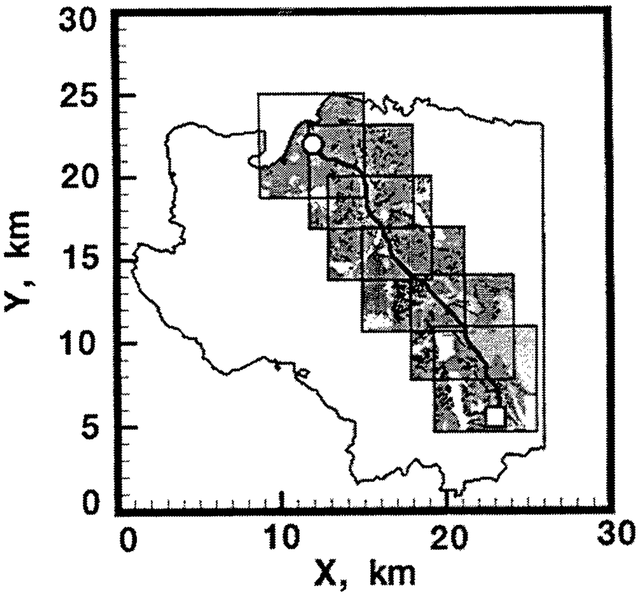
The updated arrival time H_{ij} in the cell (i, j) is set equal to the minimum of the values H_{ij}^{kl} over the index sets $i - 1 \le k \le i + 1$ and $j - 1 \le l \le j + 1$ (that is, over all eight neighbor cells of cell $[i, j]$) and H_{large}. For boundary cells, equation (2) is applied only for neighbors that are within the vicinity map, the minimum is selected among those calculations, and the initial boundary value computed with equation (1). The iteration is terminated when all values H_{ij} have converged. The theoretical maximum number of iterations is equal to the number of cells in the map. However, in practical applications, the number of iterations can be held down to about 10 or less by a proper arrangement of sweep directions and sequences. It is easy to see that the resulting relief has no relative minima inside the map (each inside cell has at least one neighbor that is a source cell with a lower relief value) and that cells that have the value H_{large} after convergence cannot be reached by a signal from any source. (That is, no destination can be reached from those cells.)

The robot's route within the vicinity map is determined stepwise by proceeding from the robot's position to that neighbor cell, which is a source for the signal arrival according to Huygens' relief. The neighboring source cells are found by recalculating the arrival times with equation (2). If there is more than one source, then a tie-breaking algorithm alternatively chooses the rightmost and leftmost source cell as the next position of the robot. Because there are no relative minima within the map, the robot cannot become stuck and the route ends either in a destination or in a boundary cell.

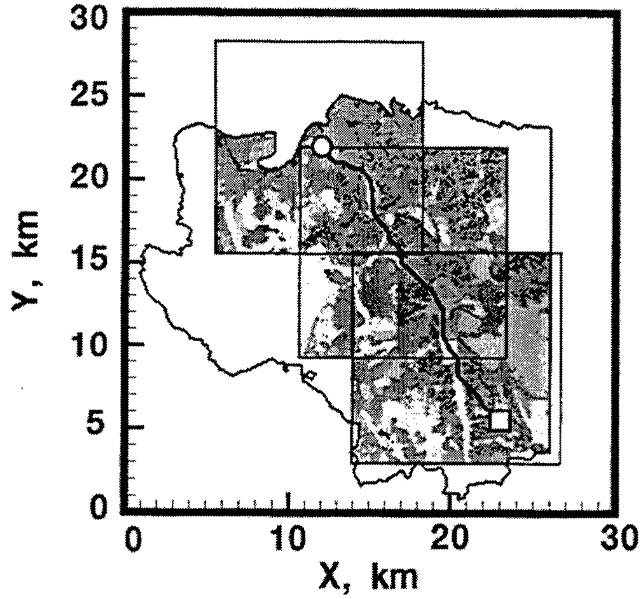
To apply the described vicinity map method, one must also specify, in addition to the robot's location and the coordinates of the destinations, the size and resolution of the vicinity map. The choice of the size and resolution depends on the robot's task and on the resolution of the input map. (It makes little sense to use vicinity maps with finer resolution than the available input map unless the robot is making itself a new map.) One can expect to arrive at different routes for different vicinity map specifications. We illustrate this with some examples. Figures 5a and b show a series of vicinity maps that are used by a robot as it proceeds along a route. The different shades in the maps indicate different navigation speeds. The outer contour in the figures indicates the area that is covered by the given input map. (The speed map* in this and the following examples is for an actual terrain, see Bullock 1998). Of that input map, only those parts have been used for route determination, which are shown in the series of vicinity maps. The resolutions in these examples are the same as for the input map (cell sizes of 100×100 m^2). The vicinity maps in Figure 5a have 64×64 cells and in Figure 5b, 128×128 cells. Figures 6a and b show the corresponding Huygens' relief maps. Because the larger maps have four times more cells than the smaller maps, the computing time for the relief in Figure 6b was about four times longer than that for the relief in Figure 5a. In spite of equal resolutions, the robot's route is slightly different in these two examples. This is so because a small vicinity map does not contain information about the terrain outside the map, but such information can be taken into account if a larger map is used. As a rule, a vicinity map should be commensurate with the sizes of salient terrain features, such as road connections, patches of open fields, and boundaries of wooded areas. In a maze, the vicinity map must be large enough so that openings and dead-end traps are properly covered. (It might be worthwhile to design specific algorithms for navigation through mazes, but this is not the subject of this report.) By close inspection, one can observe that the Huygens' relief contour lines in Figures 6a and b are not continuous across adjacent maps because the relief in each map has been calculated from different boundary values. These discontinuities do not affect the route-finding algorithm because the algorithm is always applied only within a single Huygens' relief map.

---

* The map is from an area of Fort Knox, KY, and the speeds were computed using the computer program "NATO Reference Mobility Model - Version 11" for vehicle cross-country speeds.

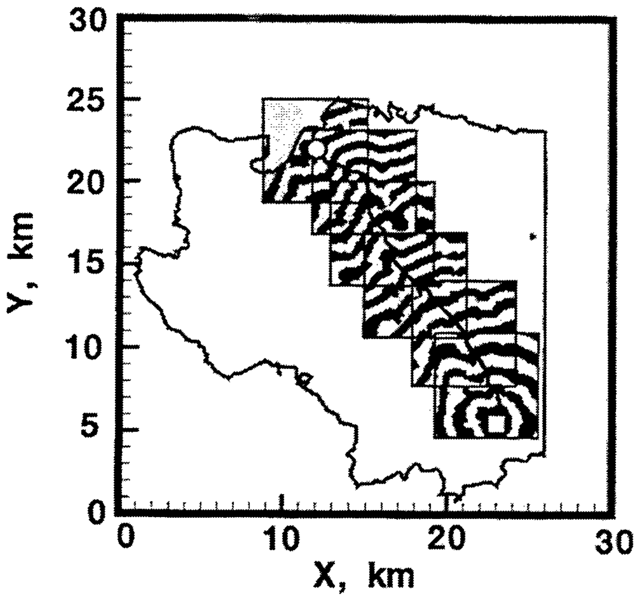


(a)

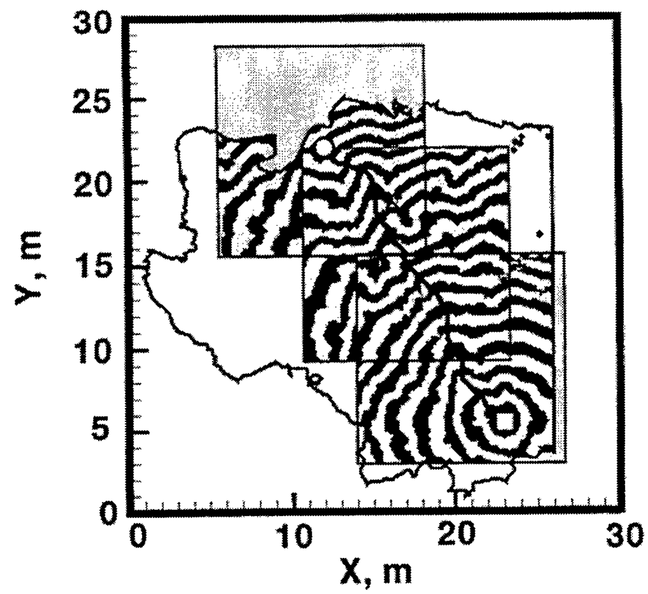


(b)

Figure 5. Series of Vicinity Terrain Maps.



(a)



(b)

Figure 6. Series of Vicinity Huygens' Relief Maps.

The effects of the two most important parameters of the vicinity maps — their sizes and resolutions — are illustrated in Figures 7a and b. Figure 7a displays three routes that are found with the resolution of the input map (cell size 100×100 m^2) but with different sizes of vicinity maps. The leftmost route is found if the vicinity map covers the whole mapped area, the dashed route is obtained by using vicinity maps with 128×128 cells (as in Figure 5b), and the rightmost solid curve is obtained by using small vicinity maps with only 32×32 cells. The computing time for the smaller map series was about 20% of the computing time for the whole region, and the computing time for the larger map was about 50% of that for the whole region. This roughly corresponds to the relative areas covered in the three examples. The loss of information that is caused by using partial maps of the terrain affects the details of the route. However, such effects are unavoidable in principle because a terrain area that can be represented by maps is always finite and, for a big map, there is never a guarantee that an even bigger map would not contain a better route. The map size must be chosen in accordance with the task of the robot.

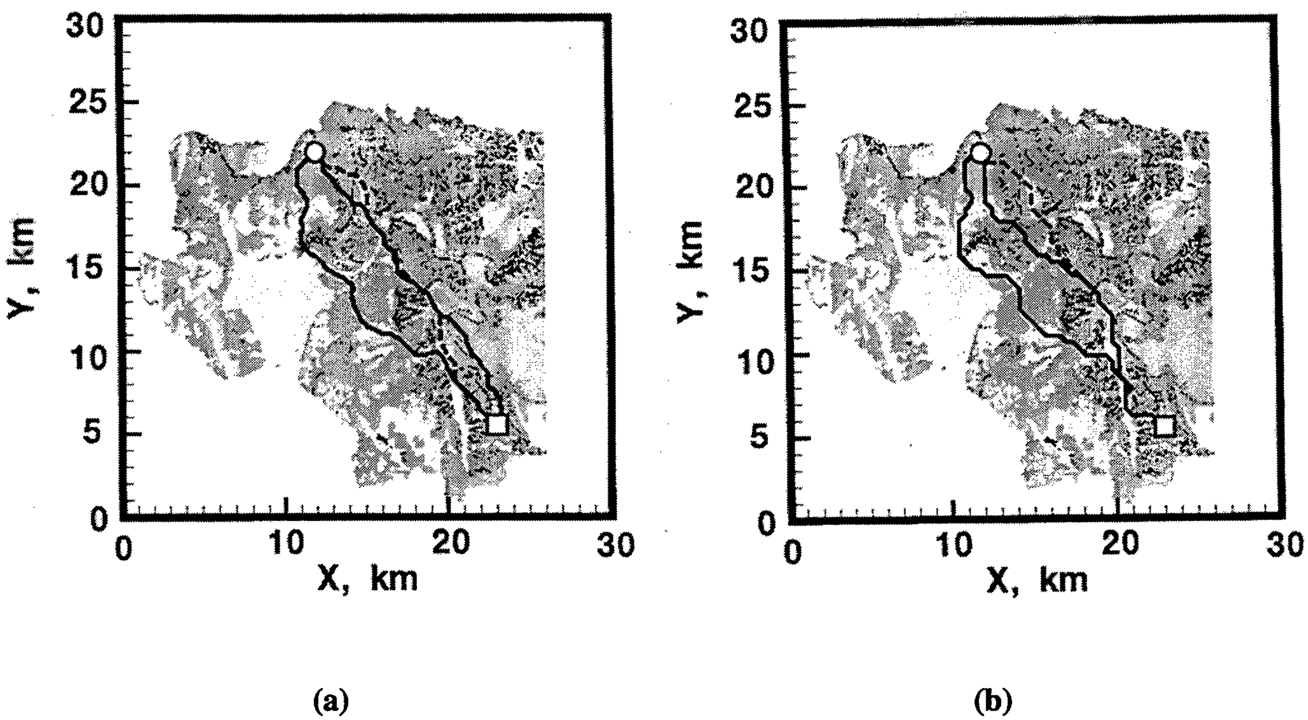


Figure 7. Routes Obtained With the Vicinity Map Method.

The second parameter of vicinity maps — the map resolution — has a larger effect on computing times. Some results with coarse maps are shown in Figure 7b. The sizes of the vicinity maps in these examples were the same as in the examples in Figure 7a, but the resolutions were four times lower; that is, the area of each cell was 400×400 m^2. (Each cell of the low-resolution maps was four times larger than the original map; that is, each low-resolution cell contained 16 high-resolution cells.) The leftmost route in Figure 7b is obtained by using the whole input terrain map, the rightmost solid curve was obtained with the smallest vicinity maps (8×8 cells), and the dashed curve, with the larger vicinity maps (32×32 cells.) The routes are not much different from those of Figure 7a, but computing times for any of the examples in Figure 7b were only about 8% of the computing time for the whole input region with fine resolution.

The examples show that the vicinity map method for long-range route planning can be used even for dynamically changing battlefield environments when computing speed is important. The sizes and resolutions of the maps should be chosen commensurate with the features of the environment, and the appropriateness of vicinity map sizes is crucial for route detection. The detrimental effects of increased coarseness on route quality are not very great (see also Figures 1 and 3), but coarser maps reduce the computing time significantly.

The "quality" of a planned route can be measured by the anticipated travel time along the route. However, a comparison based on travel times is fair only among routes that are based on the same terrain granulation because a change of granulation also changes the information about the terrain that is available to the robot for route planning. Table 1 lists the travel times for the routes that are shown in Figures 7a and b. Comparing the travel times within each column, one observes a slight increase as the size of the vicinity map decreases. A comparison of travel times across columns is not meaningful, as previously explained.

4. Telescopic Map Method

The examples in Figures 7a and b indicate that a finite-size vicinity map can miss an optimal route because the route planning cannot take into account terrain features that are outside the vicinity

Table 1. Travel Times by Vicinity Map Method

Vicinity Map Size	Cell Size	
	100×100 m^2	400×400 m^2
Entire Terrain	2 hr 9 min	2 hr 3 min
12.8×12.8 km^2	2 hr 22 min	2 hr 17 min
3.2×3.2 km^2	2 hr 23 min	2 hr 27 min

map. (In the vicinity map method, the outside terrain is assumed to be homogeneous.) To accommodate outside features and, at the same time, keep the vicinity map small (to save computing time), one can use a general representation of salient terrain features outside the vicinity map instead of assuming a uniform outside field. This can be achieved by embedding the fine-resolution vicinity map in maps with coarser resolutions. Such maps permit one to take larger terrain features into account without calculating a detailed Huygens' relief up to the destination. In this section, we propose such an embedding in form of a telescopic series of maps, whereby each outside map has cells that are twice as large as the cells of the next inside map. At the beginning of its journey, the robot establishes such a series of maps and calculates a route in the innermost map. It then proceeds along the route until it reaches a boundary zone of the innermost (i.e., vicinity) map. At that time, a new series of maps is established and the process repeated.

Figure 8 illustrates the process. In Figure 8a, the robot is in its initial position and has established four telescopic maps, whereby the destination is in the fourth map. The map-generating algorithm is programmed to generate one more telescopic map than is necessary to cover the destination (see Figures 8b, c, and d), but, in this case, the fourth map already completely covers the input terrain map. The innermost map in this example has 32×32 cells and covers an area of 6.4×6.4 km^2. Figure 8b shows the robot at an advanced position and the corresponding telescopic map series. The destination is now located in the third map, and the fourth map again covers the whole input map. Figure 8c shows that the area covered by the map series shrinks as the robot approaches the destination. (There is no fourth map in Figure 8c.) Finally, in Figure 8d, the

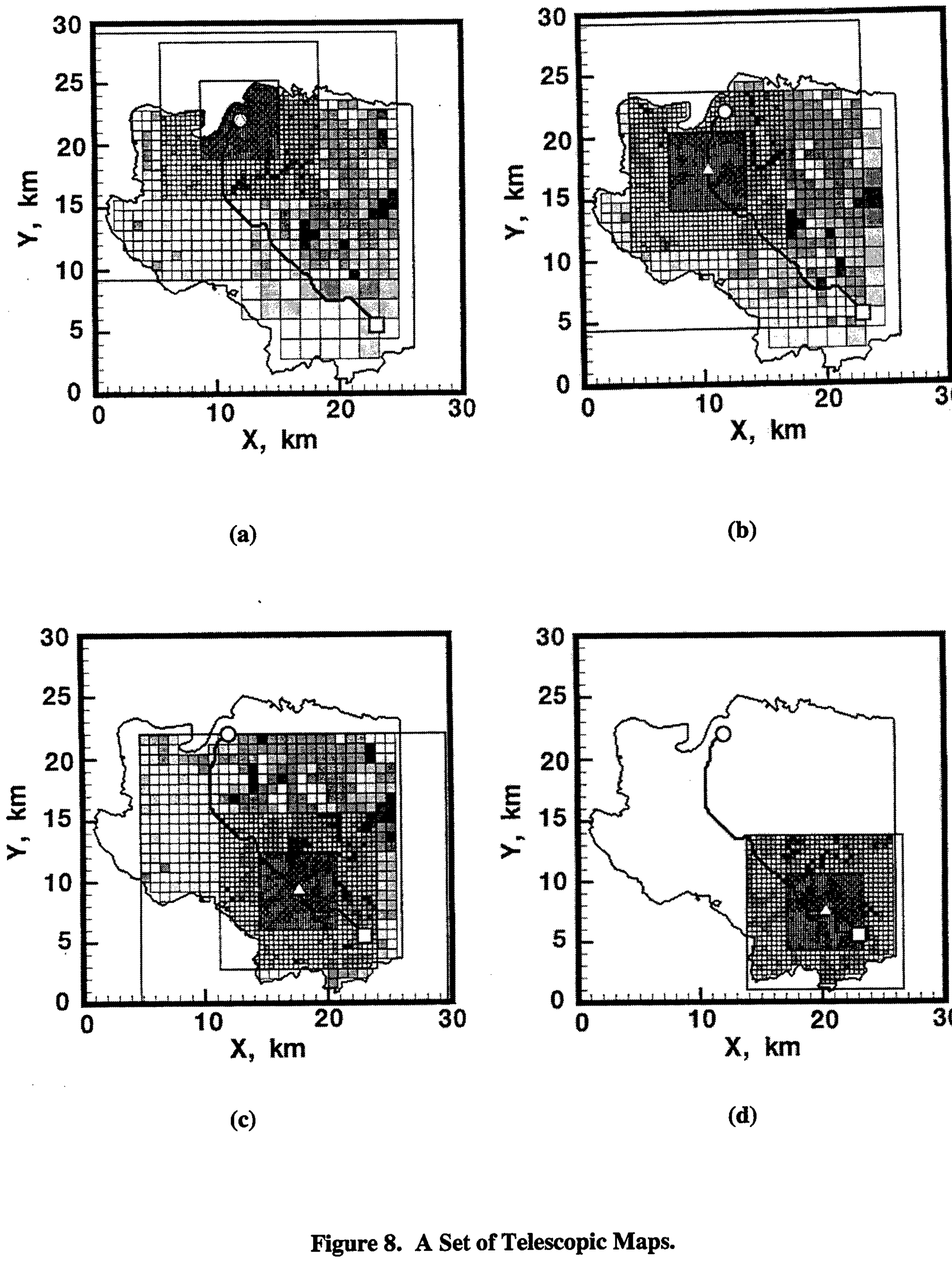


Figure 8. A Set of Telescopic Maps.

destination is within the innermost map. The next larger map is also computed so that possible routes that are partially outside the vicinity map are not excluded from the analysis.

The array of maps is constructed such that all maps are centered around a fine-granulated innermost vicinity map, and each larger map is twice as large as its predecessor. The number of cells is assumed to be equal in all maps. Therefore, the cell size of the next larger map in the telescopic sequence is always twice the cell size of the previous map (a cell of the next larger map contains four cells of the previous map). Input for the map-generating algorithm consists of the desired number of cells along a border of the innermost map, the length (in meters) of that border, and the coordinates of the robot and its destinations. To simplify the computing logic, the number of cells is restricted to powers of two. (For practical purposes, the numbers 32, 64, or 128 are most useful.) Using this input, the map-generating algorithm computes a series of telescopic maps such that the robot is located at the center of the innermost map and the most distant destination is located in the penultimate outside map. (Or, as in Figure 8a, as many maps are computed as are needed to cover the whole given terrain.) Next the terrain properties are calculated in each cell of every map by averaging over the input terrain map. The result is a set of granulated terrain maps as shown in Figures 8a–d.

The next step in route calculation is the computation of Huygens' relief. The calculation starts with the largest map and proceeds inward to the smaller maps. The computation in the largest map starts with setting all Huygens' relief values equal to an appropriate upper bound H_{large}. Next, Huygens' relief values in source cells (i.e., in cells that contain at least one destination) are set equal to zero if the largest map is also the innermost map and equal to

$$H_S = d_{ij} / (2 \ v_{max}) \tag{3}$$

otherwise. In equation (3), d_{ij} is the size of the source cell and v_{max} is the maximum speed for the present problem (usually corresponding to road speed). A finite relief value H_S for source cells in the larger maps is necessary because cells in different maps have different sizes. If the relief value would be set equal to zero in a very large cell of an outer map, then this would model an infinite signal propagation across the cell and distort the final relief. After assigning the proper relief values to the source cells, the values in the remaining cells are computed by iterative sweeping as described in section 3; that is, by repeated application of equation (2). This calculation is done for the whole map, including the central area that is covered by the smaller inner maps.

At the end of the iteration, Huygens' relief values in those cells that border the next smaller map are used to calculate initial boundary cell values for the smaller map. By the construction of the map series, each boundary cell in a smaller map borders either two or three cells of the next larger map, see Figure 9. To obtain initial values for the boundary cells, we calculate the signal arrival times from these border cells of the larger map and chose the smallest of the two (or three) arrivals. The calculation is explained with the help of Figure 9. Let d be the size of the smaller cells. Then the distance A (the length of the arrow from cell no. 2 to cell no. 6) in the figure is

$$A = \sqrt{(d\,3/2)^2 + (d/2)^2} = (d/2)\sqrt{10}. \tag{4}$$

The three signal arrival times in the corner cell no. 5 are

$$\left.\begin{aligned} H_5^2 &= H_2 + A\,2/(3\,V_2) + A/(3\,V_5) \\ H_5^3 &= H_3 + d\sqrt{2}/V_3 + d\sqrt{2}/(2\,V_5) \\ H_5^4 &= H_4 + A\,2/(3\,V_4) + A/(3\,V_5) \end{aligned}\right\} . \tag{5}$$

where V_2, V_3, V_4, and V_5 denote the navigation speeds in the corresponding cells. The smallest among these three arrival times is chosen as initial boundary value for cell no. 5. The two signal arrival times in the cell no. 6 of the inner map are

$$\left.\begin{aligned} H_6^1 &= H_1 + d\sqrt{2}/V_1 + d\sqrt{2}/(2\,V_6) \\ H_6^2 &= H_2 + A\,2/(3\,V_2) + A/(3\,V_6) \end{aligned}\right\} . \tag{6}$$

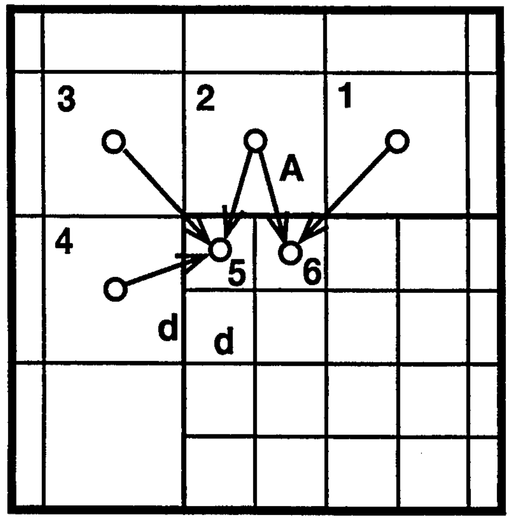


Figure 9. Calculation of Boundary Values.

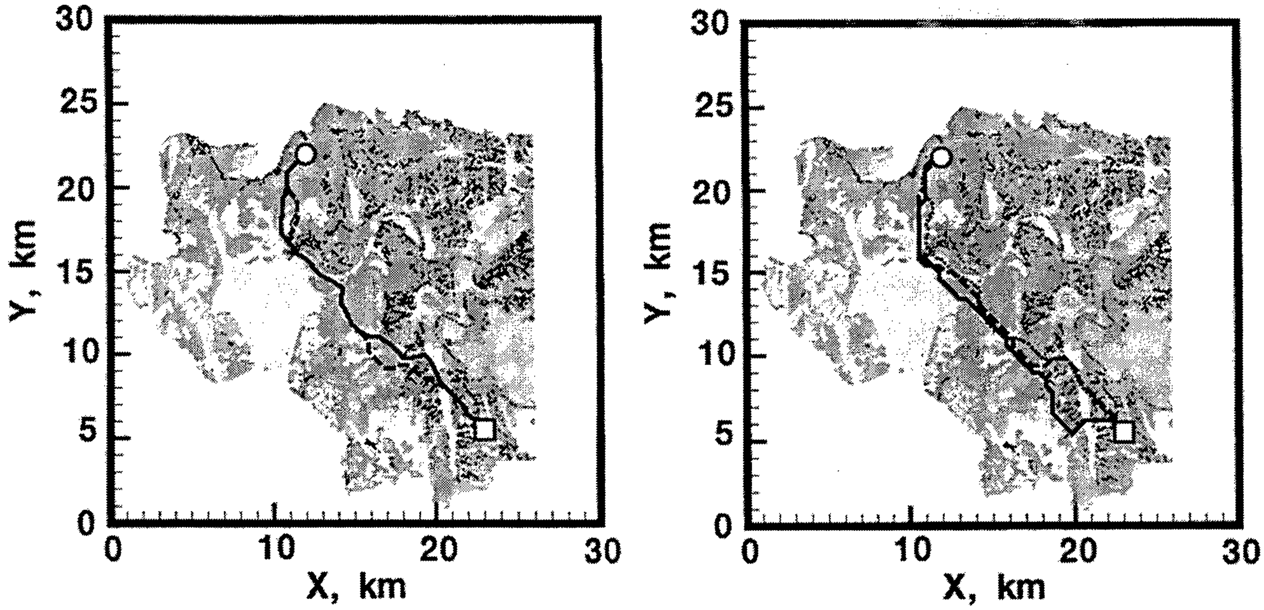
The receiving cell no. 6 is assigned the smaller of these two values. This process is repeated for all boundary cells of the inner map.

The calculation of Huygens' relief for cells in the smaller maps proceeds in principle in the same manner as for the outer map, except that, for the smaller maps, one has initial values for their boundary cells. All other cells initially receive a large relief value, cells that contain sources receive a value computed by equation (3), and an iteration with equation (2) is started. A more detailed description of the iteration and the handling of boundary cells with initial values is given in section 3. This process is repeated until the relief has been calculated for the innermost map (i.e., for the vicinity map).

The robot's route is calculated only in the innermost map and up to a point that is close to the boundary of the map. The algorithm for the route calculation is the same as described in sections 2 and 3 — starting from the robot's position cell, one determines those neighbor cells that are sources for the signal arrival in the position cell. Usually, there will only be one source cell, and that cell is chosen as the next position cell. If there is more than one source cell, then a tie-breaking algorithm is activated that alternatively selects the leftmost and rightmost source cell, respectively. When the route reaches a border zone of the innermost map, a new set of telescopic maps is established with the present position of the robot at its center. The border zone of the map is defined as a zone where the distance to the border is less than one fourth of the map size. (If the robot has a positive sensing radius, then the size of the innermost map is made at least 1.5 times larger than the sensing radius.) The process is then repeated in the new set of maps, starting with the gathering of terrain information and computation of the corresponding Huygens' relief maps.

The parameters of the telescopic array of maps are the same as in section 3 — the whole telescopic array of maps is determined by specifying the size of the innermost map and the number of cells within the innermost map. (The number of cells along a side are restricted to powers of two.) Figures 10a and b illustrate the effects of these parameters. In Figure 10a, we have plotted routes that are obtained with innermost maps that have the same resolutions as the input map (cell size 100×100 m^2) but differ by their sizes. The solid curve is obtained using an innermost map with 32×32 cells, and the dashed curve is obtained using a larger innermost map with 128×128 cells. One can compare these results with those in Figure 7a, where the same size and resolution maps were used as vicinity maps, and with Figure 7b, where coarser vicinity maps of same size were used. To make these comparisons, we take, as standard, the route that was obtained by using the whole input terrain information (i.e., the leftmost route in Figure 7a). In Figure 10a, with the telescopic map results, the standard curve coincides with the dashed curve (large innermost map) at the beginning of the route and with the solid curve (small innermost map) for the rest of the route. This is in stark contrast to Figure 7a, where even the 128×128 cell vicinity map was not sufficiently large for finding the standard route. The difference between the two figures illustrates an advantage of the telescopic map method; in the vicinity map method, terrain features that are outside the vicinity map are completely ignored, while in the telescopic map method, such features are taken into account albeit not to the full extent of a complete map analysis.

Figure 10b shows effects of map resolution. Here, the innermost maps had the same sizes as in Figure 10a, but their resolutions were four times lower than in the input terrain map. That is, the innermost maps had the same resolutions and sizes as the vicinity maps in the examples shown in Figure 7b. Two results are presented. The solid curve is obtained using an innermost map with 8×8 cells, and the dashed curve is obtained using a larger innermost map with 32×32 cells. The



(a)

(b)

Figure 10. Routes Computed With the Telescopic Map Method.

standard route is shown as a thin dotted line, and it coincides along most of its length with the dashed curve. The closeness of these results to the standard route again demonstrates the advantage of the telescopic map method that arises from taking into account distant terrain features.

Travel times for the routes shown in Figures 10a and b are listed in Table 2. The travel times in the first column (for a cell size of 100 × 100 m^2) can be compared with corresponding times in Table 1. The comparison reveals a slight advantage of the telescopic map method. The travel times in the second column are also shorter than the corresponding times in Table 1, but this comparison is not meaningful because the route-finding programs for the vicinity and telescopic map methods, respectively, use different algorithms for cell averaging. Therefore, the terrain maps with coarse granulation are different for the two methods and comparisons of route geometry, rather than the travel times, are more relevant.

In the here-presented example of navigation in open terrain, there is apparently a dichotomy for choosing the initial direction of travel, whereby the difference in travel times between an eastern and western route depends on features at larger distances from the starting point. Therefore, a vicinity

Table 2. Travel Times by Telescopic Map Method

Inner Map Size	Cell Size	
	100×100 m^2	400×400 m^2
25.6×25.6 km^2	—	1 hr 55 min
12.8×12.8 km^2	2 hr 12 min	1 hr 55 min
3.2×3.2 km^2	2 hr 11 min	2 hr 5 min

map method with a moderate map size that cannot take those features into account does not capture the optimal initial direction of the route. It is, however, surprising that even the very coarse maps in the examples shown in Figure 10b properly take these features into account. To illustrate this remark, we show in Figure 11 the terrain maps for the robot's initial position in the dashed-curve example of Figure 10b. The innermost map in that example had only 8×8 relatively large cells. The terrain representation in Figure 11 is obtained by averaging over cells with sizes between 400×400 m^2 in the innermost map and 6.4×6.4 km^2 in the outermost map. The result is indeed very coarse but nevertheless sufficient for finding the correct initial direction that leads to the standard route.

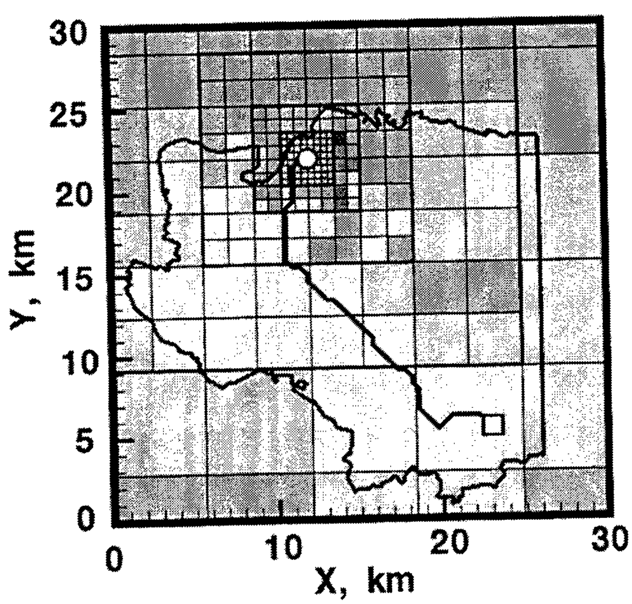


Figure 11. Telescopic Terrain Maps With Low Resolution.

The advantages of the telescopic map method must be paid for by longer computing in comparison to the vicinity map method. In general, they are twice as long as for the vicinity map method with a vicinity map that is equal to the innermost map. (This estimate is only approximate and the ratio of the computing times can vary widely depending on the particular geometry of available input maps.)

5. Summary and Conclusions

The finding of optimal routes in an open terrain is closely related to the availability of terrain maps. Only in exceptional cases will such maps exactly cover the area where the optimal route is located. If the maps are too small, then the optimal route cannot be found and a suboptimal route within the given area must be chosen. (But without the complete map, there is no way to determine whether the route is optimal.) If the map is too large, excessive computing time might be wasted to investigate map areas that are irrelevant for the planned route. In a battlefield environment, the available terrain maps will most likely be larger than necessary for the assigned tasks of a scouting robot. Therefore, a route-finding process is needed that permits handling large maps by selecting from such maps only those areas and terrain features that are important for the navigation task of a robot.

In this report, we have presented two such methods. Both use a navigation function (Huygens' relief) approach that provides a complete solution in a closed environment (e.g., in a laboratory with a complete map of the environment). That approach (Celmiņš 1997 and 1998) is not adequate for route finding in an open terrain because it would require an excessive computing time to analyze the total area represented by terrain maps. In a first modification of the original approach (see section 3), a small vicinity map is used around the location of the robot, while the rest of the environment is treated as a uniform field. In a second modification (see section 4), the small vicinity map is embedded in a set of outer maps in telescopic fashion with a coarse representation of the outer environment. In both methods, the vicinity map (or the set of maps in the telescopic map method) is moved along with the robot's position so that the immediate neighborhood of the robot is represented by a high-resolution map at all times.

Examples presented in sections 3 and 4 show that the vicinity map method has the shortest computing times. However, the size and resolution of the vicinity map must be chosen judiciously and commensurate with salient features of the environment. The telescopic map method requires longer computing times for route finding under similar conditions but has the advantage that the choice of the innermost map that represents the immediate vicinity of the robot is not as critical as in the vicinity map method. This is so because the terrain representation by telescopic maps makes the consideration of distant terrain features possible. Both methods are well suited for onboard calculations because of the speed and extreme simplicity of the algorithms.

6. References

Bullock, C. D. Personal communication. Waterways Experiment Station, Vicksburg, AL, 1998.

Celmiņš, A. "Battlefield Robot Cybernation and Navigation." ARL-TR-1571, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, December 1997.

Celmiņš, A. "Route Finding by Huygens' Principle." *Robotics 98, Proceedings of the Third ASCE Specialty Conference on Robotics for Challenging Environments*, pp. 43–49, Albuquerque, NM, American Society of Civil Engineers, Reston, VA, April 1998.

Khatib, O., and J.-F. Le Maitre. "Dynamic Control of Manipulators Operating in a Complex Environment." *Proceedings of the Third International CISM-IFTOMM Symposium,* Udine, Italy, September 1978, *Theory and Practice of Robots Manipulators,* pp. 267–282, Elsevier 1979.

Koditschek, C. D. "Robot Planning and Control Via Potential Functions." *The Robotics Review 1*, pp. 349–367, O. Khatib, J. J. Craig, and T. Lozano Pérez (editors), MIT Press, 1989.

Rimon, E., and D. E. Koditschek. "Exact Robot Navigation Using Cost Functions: The Case of Distinct Sperical Boundaries in E^n." *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1791–1796, Philadelphia, PA,1988.

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDQ D SCHMIDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	OSD OUSD(A&T)/ODDDR&E(R) R J TREW THE PENTAGON WASHINGTON DC 20301-7100
1	DPTY CG FOR RDE HQ US ARMY MATERIEL CMD AMCRD MG CALDWELL 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN PO BOX 202797 AUSTIN TX 78720-2797
1	DARPA B KASPAR 3701 N FAIRFAX DR ARLINGTON VA 22203-1714
1	NAVAL SURFACE WARFARE CTR CODE B07 J PENNELLA 17320 DAHLGREN RD BLDG 1470 RM 1101 DAHLGREN VA 22448-5100
1	US MILITARY ACADEMY MATH SCI CTR OF EXCELLENCE DEPT OF MATHEMATICAL SCI MAJ M D PHILLIPS THAYER HALL WEST POINT NY 10996-1786

NO. OF COPIES	ORGANIZATION
1	DIRECTOR US ARMY RESEARCH LAB AMSRL D R W WHALIN 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB AMSRL DD J J ROCCHIO 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB AMSRL CS AS (RECORDS MGMT) 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1145
	<u>ABERDEEN PROVING GROUND</u>
4	DIR USARL AMSRL CI LP (305)

NO. OF COPIES	ORGANIZATION
1	OUSD AT STTS/LW M TOSCANO PENTAGON RM 3B1060 WASHINGTON DC 20301
2	US ARMY TACOM AMSTA ZR B BRENDLE AMSTA TR V P LESCOE WARREN MI 48397-5000
1	US ARMY MICOM UGVS/JPO AMCPM UG COL J KOTORA REDSTONE ARSENAL AL 35802
6	NIST ROBOT SYS DIV BLDG 220 RM B 124 K MURPHY S LEGOWIK M JUBERTS S SZABO H SCOTT DR J ALBUS GAITHERSBURG MD 20899
2	US ARMY BELVOIR AMSEL RD CC CCD S LAMBERT AMSEL RD NV MN LTC H MCCLELLAN JR 10221 BURBECK RD STE 430 FT BELVOIR VA 22060-5806
1	TOOELE ARMY DEPOT SDSTE AE F ELDREDGE BLDG 1005 TOOELE UT 84074-5004
1	SUPT US MILITARY ACADEMY MADN A WEST POINT NY 10996-1786
1	US MILITARY ACADEMY CPT R STURDIVANT DEPT OF MATH SCI WEST POINT NY 10996-1786

NO. OF COPIES	ORGANIZATION
1	US ARMY WAR COLLEGE STUDENT OFCR DET LTC R LYNCH CARLYLE BARRACKS PA 17013-5050
1	NCCOSC RDTE DIV 531 B EVERETT 53406 WOODWARD RD SAN DIEGO CA 92152-7383
1	UNIV OF TEXAS ARLINGTON COLLEGE OF ENG DR E G METALLIA PO BOX 19019 ARLINGTON VA 76019-0019
1	STANFORD UNIV COMPUTER SCI DEPT PROF J C LATOMBE STANFORD CA 94305-9010
4	OAK RIDGE NATIONAL LABS J HERNDON DR W HAMEL D HALEY B RICHARDSON PO BOX 2008 OAK RIDGE TN 37831-6304
2	UNIV OF PENNSYLVANIA GRASP LABORATORY PROF R BAJCSY DR E LARGE 34TH AND SPRUCE ST PHILADELPHIA PA 19104
1	USAE WTRWYS EXPRMNT STATION INFORMATION TCHNLGY LAB C D BULLOCK VICKSBURG MI 38180-6199
1	LOCKHEED MARTIN ASTRONAUTICS M ROSENBLUM M S H8380 PO BOX 179 DENVER CO 80201

NO. OF COPIES	ORGANIZATION
1	THE ASSOC FOR UNMANNED VEHICLE SYSTEMS INT B LINDAUER 1735 N LYNN ST STE 950 ARLINGTON VA 22209-2022
1	DARPA T M STRAT 3701 N FAIRFAX DR ARLINGTON VA 22203
2	RAND CORP M S DTP 2 R STEEB J M MATSAMURA 1700 MAIN ST PO BOX 2138 SANTA MONICA CA 90407-2138
18	DIRECTOR US ARMY RESEARCH LAB AMSRL IS CI T MILLS P DAVID P FISHER D HILLIS P GAYLORD J VAN ETTEN S YOUNG AMSRL IS J GANTT P EMMERMAN AMSRL IS C J CHANDRA COL M KINDL AMSRL IS CB L TOKARCIK S HO J BOWDEN S BRAGONIER R GREGORY J LOCKER T ROSE 2800 POWDER MILL RD ADELPHI MD 20783-1145

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
29	DIR USARL AMSRL WM DR I MAY AMSRL WM BB C M SHOEMAKER DR J BORNSTEIN G HAAS R VON WAHLDE C L HENRY R M PHELPS T T VONG N M WOLFE B T HAUG J J SPANGLER M TRAN I T MCLAUGHLIN T L BORSSEAU M D KREGEL AMSRL HR SC T W HADUCH AMSRL IS CI B BROOME A CELMINS (3 CPS) A BRODEEN J DUMER LTC R HAMMELL T HANRATTY R HELFMAN R KASTE E BAUR H INGHAM AMSRL CI HC C ZOLTANI

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1999	Final, Jun 97 - Jun 98

4. TITLE AND SUBTITLE

Multimap Procedures for Robot Route Finding in Open Terrain

5. FUNDING NUMBERS

9TED20

6. AUTHOR(S)

Aivars Celmiņš

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-IS-CI
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-1897

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An important task for scouting robots in open terrain is the planning of a route to a prescribed destination with the help of digital maps. In an open environment, available terrain maps usually will be too large for a timely analysis of a whole map and may cover areas that are not relevant for route planning. One needs, therefore, route-planning procedures that use partial terrain maps. This report presents two such procedures that use sets of partial maps. A first method uses a series of vicinity maps that are aligned with the position of the robot; a second method uses sets of telescopic maps that follow the robot. In both procedures, the routes are determined using a navigation function method based on Huygens' principle of wave propagation. Examples of results by the two methods are presented and compared. It is concluded that the vicinity map method is faster, but the telescopic map method is more robust and its results are less sensitive to the choices of sizes and resolutions of partial maps.

14. SUBJECT TERMS

robot navigation, navigation function, route finding, Huygen's principle, open terrain

15. NUMBER OF PAGES

31

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1897 (Celmiņš) Date of Report February 1999

2. Date Report Received ______

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ______

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ______

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ______

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ______

CURRENT ADDRESS

Organization

Name E-mail Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)