The Geometric Path Planner for Navigating Unmanned Vehicles in Dynamic Environments

Juan Pablo Gonzalez, Bryan Nagy and Anthony Stentz
The Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
{jgonzale, bnagy, axs}@ri.cmu.edu

Abstract-- *In this paper, we describe the Geometric Path Planner (GPP) that produces routes for unmanned ground and air vehicles. The GPP generates plans that minimize a weighted, multi-metric cost function considering factors such as mobility risk, traversal time, sensor coverage, and stealth. In addition to cost minimization, the GPP is able to reason about hard constraints imposed on one or more of these cost metrics. The GPP re-plans paths in real time in response to changing conditions, tasks, and new information. The planner has been used to control both ground and air vehicles.*

I. INTRODUCTION

Unmanned Ground Vehicles (UGVs) have improved their capabilities considerably in the last decade. Much attention has been given to development of hazard detection and avoidance, since the success of an unmanned vehicle capable of point-to-point navigation depends on its autonomous mobility capabilities more than anything else. However, as the desire for more intelligent operation arises, so does the need for a higher level planner able to incorporate existing maps, plan paths that optimize multi-metric objective functions and satisfy constraints, and re-plan efficiently.

The planner presented here addresses many of these requirements. It inputs prior geographic data (digital elevation maps, and digital feature maps) and information about known terrain objects and mission-specific parameters (goal points and lines, boundaries, no-go regions, etc.) to produce paths that satisfy mission requirements. Although the planner consists of the aggregation of many previously published algorithms, to the authors' knowledge it is first planner to provide all of the above capability in a single system.

The current implementation models four mission metrics: travel time, mobility risk, coverage cost, and exposure cost. For each metric, the planner maintains a map (cost layer) specifying the cost incurred against the metric for each patch of terrain. The planner finds the path that minimizes a weighted sum of the four metrics. The weight of each layer, ranging from 0 to 1, determines its importance to the mission. A higher weight for a layer indicates that the metric corresponding to that layer should be emphasized, while a lower weight assigns less importance to that metric. For example, a path planned with weights of 1 for each of the layers will weight all cost layers equally, producing a path that is a compromise between travel time, mobility risk, exposure and coverage. A path planned with a weight of 1 for travel time and 0 for all other layers will find the quickest path, without considering mobility risk, coverage or exposure.

Travel time can also be used as a hard constraint, such that the resulting path must be reachable within a specified time.

The planner uses Field D* as its path planning algorithm, allowing for fast replanning and eliminating most of the problems derived from 8-connected world representations. The constrained version uses Constrained D*, allowing for efficient search and replanning of constrained solutions as well. Typical execution times when planning from scratch are of a few seconds for a planning area of 3 km x 3 km at 10 meter resolution. Successive calls to the planner take less time, depending on what percentage of the world model has changed.

II. TECHNICAL APPROACH

The planning algorithm at the core of the planner is Field D* [1 2], a version of D* [3 4] that uses interpolation to create globally smooth paths. Its predecessor, D*, is a dynamic version of A* that provides for fast and efficient replanning. However, on a grid map, D* has the same limitations as A*: paths are planned between the centers of grid cells, thereby limiting the angles in the resulting path to multiples of 45^{o}. Field D* uses linear interpolation and allows paths to pass through any portion of the cells (i.e., not constrained to pass through the centers). This improvement removes the 45^{o} quantization and produces smoother paths, while preserving the capability to replan efficiently

There are two main types of data utilized by the planner: prior data and mission data. Prior data consists of digital elevation maps (DEMs) and feature maps for roads, rivers and terrain cover. Mission data consists of locations of known threats, location of areas of interest, etc. Elevation maps are usually grid maps, at a resolution of 10 to 30 meters per cell, and mission data and feature maps are usually vector maps with an accuracy of 10 to 30 meters. In order to combine grid-based and vector-based data, feature maps are converted to a grid-based representation of the same resolution as the DEM. In this representation, a cell is assigned a feature type if any location within the cell boundaries contains a feature (for example, if there is a river on any part of the cell). Figure 1 shows some of the data that is typically available: elevation data (represented as shaded relief on the top, and as contour lines on the bottom, for easier visualization), paved roads (black lines), dirt roads (gray lines), rivers and lakes (blue lines).

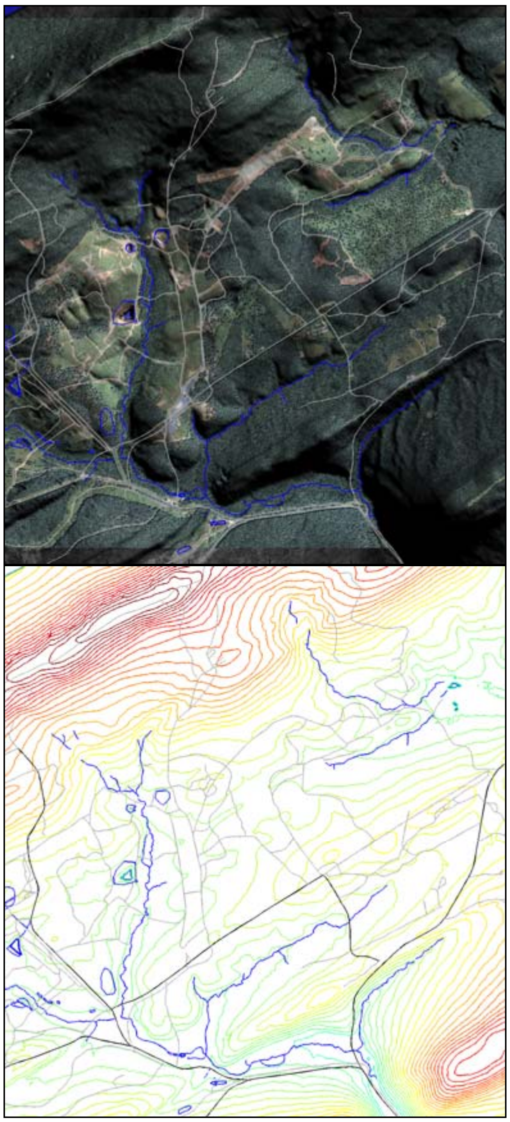


Figure 1. Top: Aerial photography of Ft Indiantown Gap. Bottom: Contour lines, roads and rivers for test area. Contour lines are approximately 8 meters apart.

The planner uses four different layers or cost maps to represent cost data for each of the above metrics.

II.A. Mobility Risk Layer

This layer represents the difficulty of traversing a particular area in the map. The mobility risk is a number between 1 and 100, where 1 is the lowest possible cost, and 100 is a non-traversable cell. Each feature type has an associated mobility risk and a slope coefficient.

The mobility risk part represents how difficult it is on average to traverse terrain of current feature type (for example, the mobility risk of paved roads is 1, the mobility risk of dirt roads is 10, and the mobility risk of tree-covered regions is 70). The slope coefficient represents how much the terrain impacts the mobility of each feature type (for example, paved roads have a slope coefficient of 0.1, dirt roads have a slope coefficient of 0.2, and tree-covered areas have a slope coefficient of 0.5). The total mobility is therefore given by:

$$C_M(x,y) = C_{MF}(x,y) + k_{SF}(x,y) \cdot slope(x,y) \qquad (1)$$

where $0 \le k_{SF} \le 1$ is the slope coefficient for the feature type of the cell *(x,y)*, *slope* is its slope normalized to 0 – 100 (for slopes between 0 and *max_slope*), and C_{MF} is its mobility risk. Cells that contain no features use a default value for C_{MF} and k_{SF}. Figure 2 shows the resulting mobility risk map using rivers and roads as features. Figure 3 shows the resulting cost map after adding tree-covered regions as well.

II.B. Time Layer

This layer represents the expected time to travel each cell on the map. The input data to calculate the time cost are the features labeled as paved roads and dirt roads. In the current implementation, there are three possible values for each cell: 1/*paved_road_speed*, 1/*dirt_road_speed*, and 1/*off_road_speed*. The specific value for each type of terrain depends on the characteristics of the vehicle for which the planning is being done. With this information it is possible to estimate the total time required to complete a path. Figure 4 shows a contour map of the test area with paved and unpaved roads, and the time cost derived from these features.

II.C. Exposure Layer

The exposure layer represents the cost of exposure to known threats. Low costs in this layer correspond to areas that are not exposed to threats, and high costs correspond to areas that are exposed. The inputs required to create this layer are the elevation map and the location of known (or suspected) threats. In the current implementation there are two possible values for cells in this layer: 1 (not exposed) and 100 (exposed). However, areas with exposure cost of 100 are not treated in the same way as areas with mobility cost of 100. The former are expensive cells to traverse, while the latter are non-traversable cells. In order to determine whether a cell is exposed, we perform a visibility calculation for all cells in the map assuming a threat at a given location with a maximum detection range. If a cell can be seen from the location of the threat the cell is labeled as exposed, otherwise it is labeled as not-exposed.

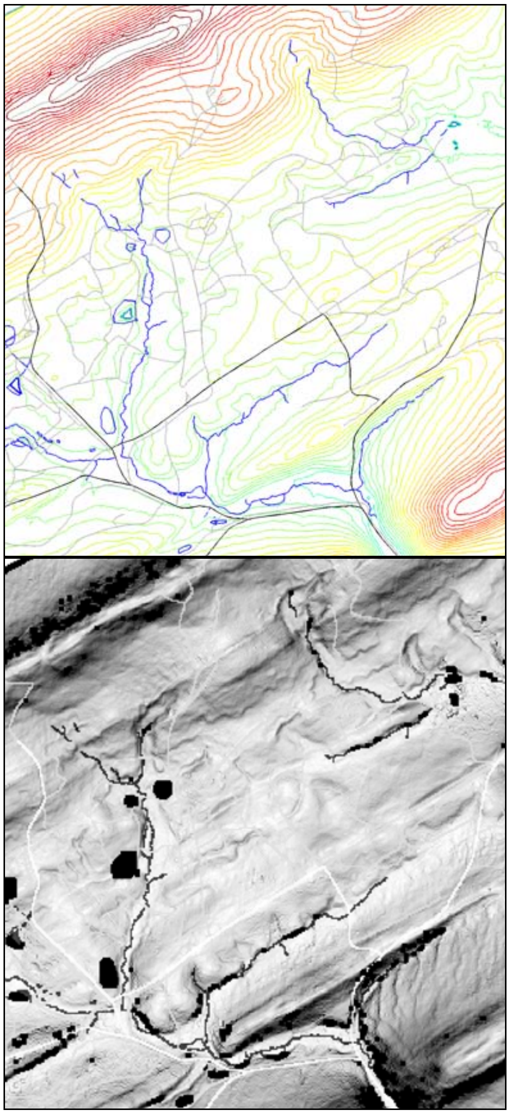


Figure 2. Top: contour map of test area showing rivers and lakes (blue), paved roads (black) and dirt roads (gray). Bottom: mobility risk considering these features plus slope. Lighter areas are easier to traverse, while darker areas are more difficult. Black areas are non-traversable.

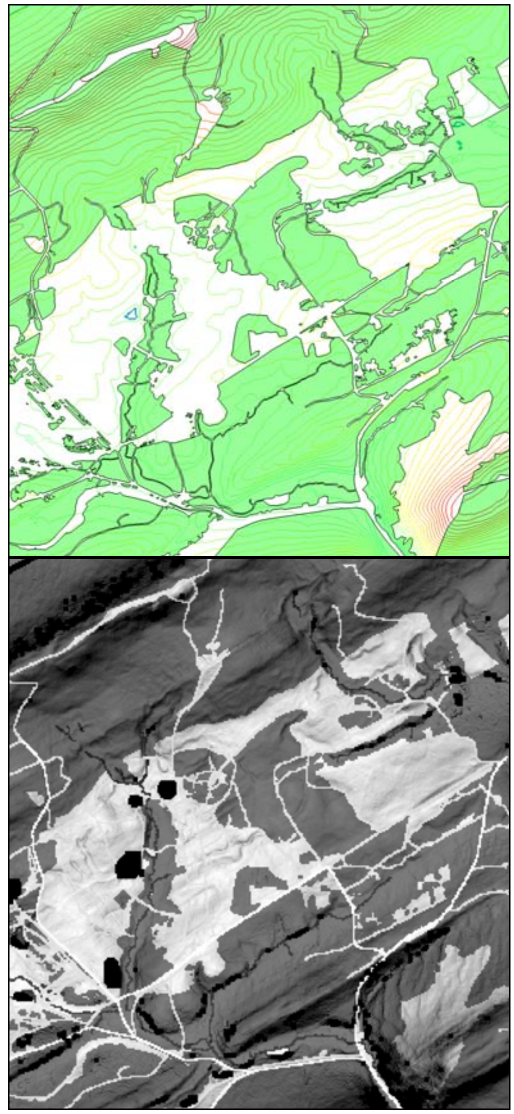


Figure 3. Top: contour map of test area showing tree-covered regions (light green). Bottom: mobility risk considering these features plus slope, rivers and roads. Lighter areas are easier to traverse, while darker areas are more difficult. Black areas are non-traversable

II.D. Coverage Layer

The coverage layer represents the cost of not covering a desired object of interest. Low costs in this layer correspond to areas that allow line of sight to a desired target (within a given observation range), and high costs correspond to areas that do not allow observation of the desired object. The inputs required to create this layer are the elevation map and the location of desired objects. In the current implementation there are two possible values for cells in this layer: 1 (location with coverage of any object) and 100 (location without coverage of any object). As with exposure, areas with a coverage cost of 100 are expensive to traverse, as opposed to areas with mobility cost of 100, which are non-traversable. In order to determine whether a cell allows coverage of an object, we perform a visibility calculation for all cells in the map assuming an object at a given location, and assuming a maximum detection range on the robot. If a cell can see the object location, the cell is labeled as with coverage, otherwise it is labeled as without coverage.

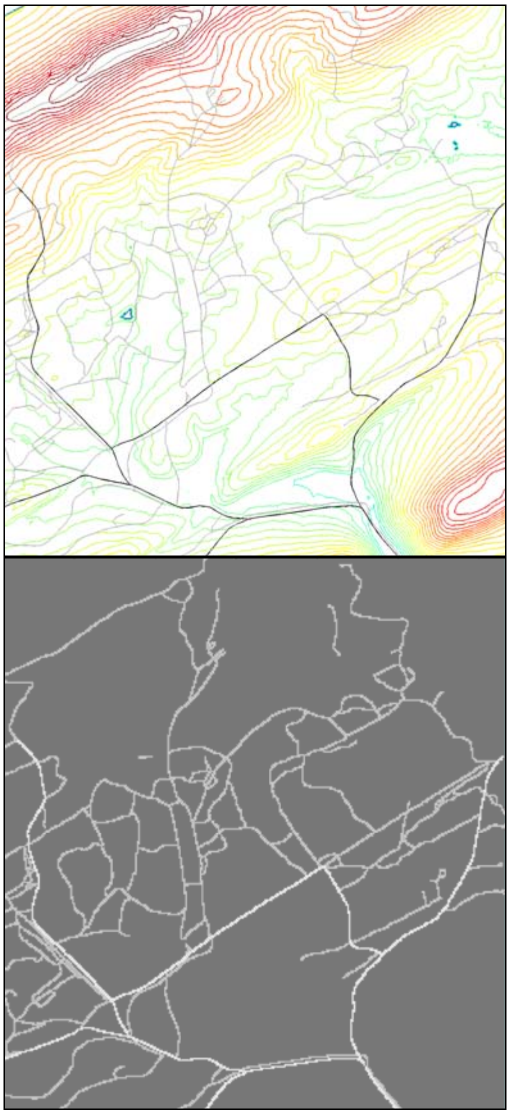


Figure 4. Top: contour map of test area showing paved roads (black) and dirt roads (gray). Bottom: time cost calculated based on the travel speed for each type of road. Lighter areas are faster to traverse, while darker areas are slower.

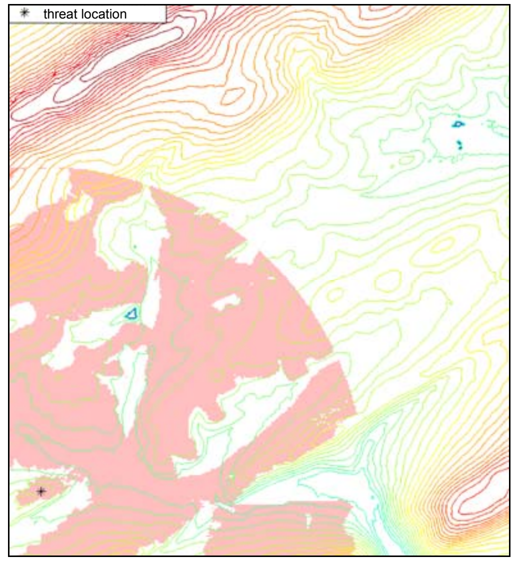


Figure 5. Left: contour map of test area showing threat location and area exposed to it (light red), assuming a detection range of 2000 m

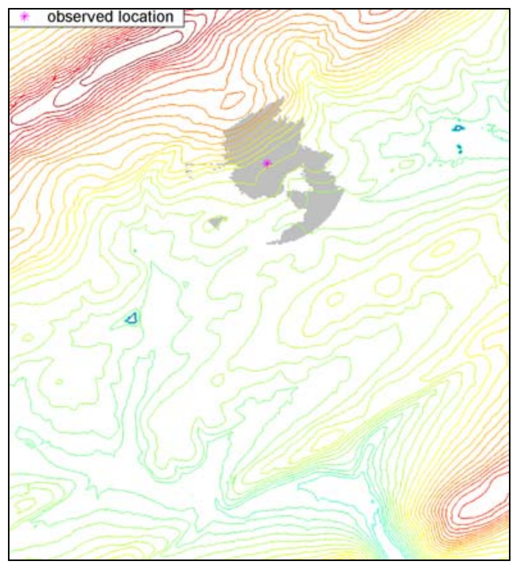


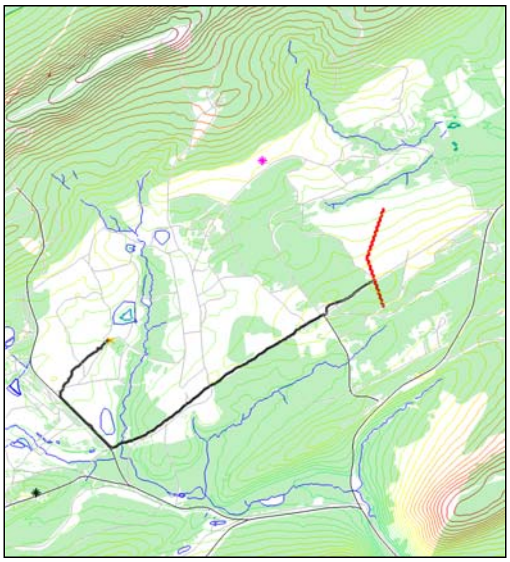
Figure 6. Contour map of test area showing target of interest, and the area from which it can be observed (gray), assuming a detection range of 500 m.

II.E. Planning paths

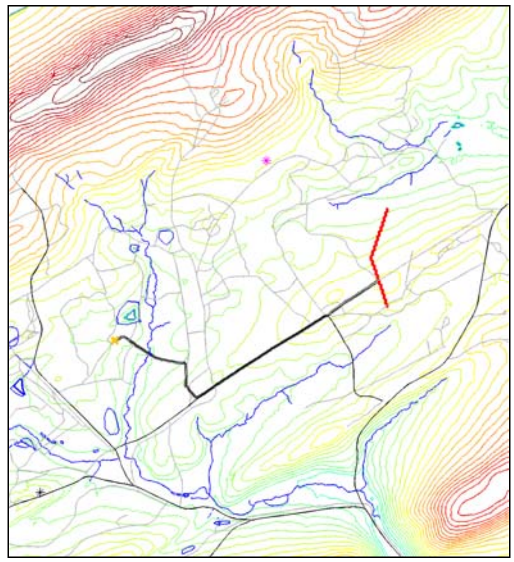
The individual cost layers mentioned above are combined into an aggregate objective function by multiplying the cost data by weights (w_i) between 0 and 1. This allows us to achieve different mobile robot behaviors through different combinations of weights: stealthy behavior can be achieved by assigning a large weight to the exposure layer; keeping an object covered can be achieved by assigning a large weight to the coverage layer; fast travel can be achieved by assigning a large weight to the time layer, etc. Likewise, compromise paths emphasizing some of the behaviors can be achieved by setting several weights to values close to 1. Figure 7 shows different behaviors achieved by setting each one of the layers to 1, and the remaining ones to 0. Figure 8 shows the resulting path when all the weights are set to one. In this case, the resulting path is a compromise solution that tries to achieve low mobility risk, short travel time, low exposure and high coverage.

II.F. Travel Time as a Hard Constraint

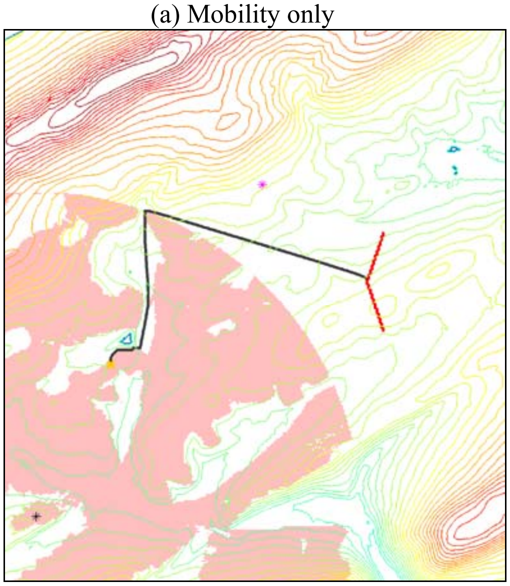
The GPP can also use one of its cost layers as an additive hard constraint along the route. For example, if we select the travel time layer as the hard constraint, the resulting route must have a total travel time shorter than the value *max_time* specified as a hard constraint. In order to meet this requirement, the planner uses Constrained D* (CD*) [5], a variant of D* that automatically selects the best combination of weights in order to satisfy the hard constraint while minimizing the remaining costs.



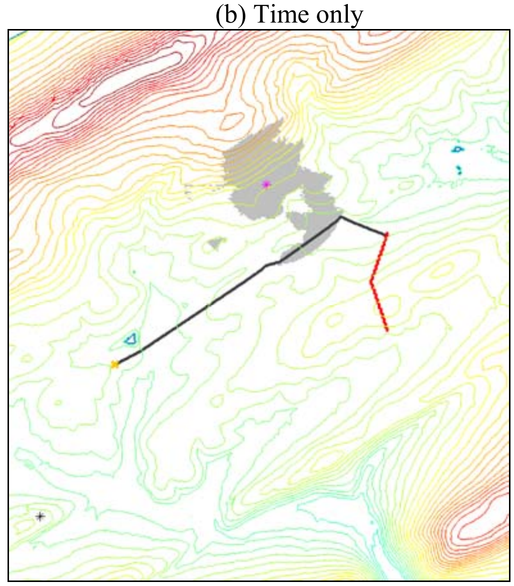
(a) Mobility only



(b) Time only



(c) Exposure only



(d) Coverage Only

Figure 7. Behaviors achieved by selecting different cost layers in the objective function: (a) produces a path with the lowest mobility risk, (b) produces a path with the shortest time of arrival, (c) produces a path with the least exposure to a known threat, and (d) produces a path with the most coverage of a desired target. The start location is a yellow “x” on the left of the map, and the goal region is the red v-shaped line on the right.

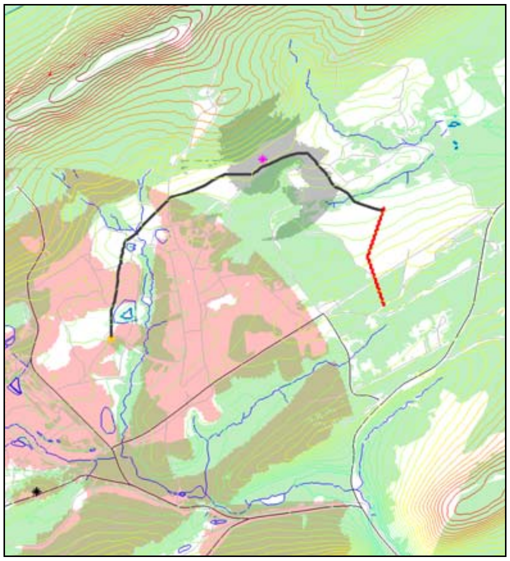


Figure 8. Path calculated with all weights set to 1. This path is a compromise solution that tries to achieve low mobility risk, short time, low exposure and high coverage.

When the planner has a hard constraint, it uses CD* to automatically set the weight assigned to the constraint layer, and the remaining cost layers are combined into a single cost plane using the pre-assigned weights. The algorithm then performs a binary search on the weight of the constraint layer to find the lowest value that allows the constraint to be satisfied. This weight minimizes the objective function of the combined layers, while satisfying the specified constraint. Like D*, CD* can also re-plan efficiently.

Figure 9 shows two examples of the planner running with different time constraints. On the top, the planner is given a time constraint of 6 minutes and weights of 1 for the other layers. On the bottom, the time constraint is changed to 3 minutes. Notice how the path no longer avoids the exposed areas, and it no longer goes to the coverage region either. This is because the constraint given is very close to the absolute minimum time required to travel from the start to the goal (157 seconds), therefore causing the planner to focus almost exclusively on the time layer.

II.G. Planning paths for Unmanned Air Vehicles (UAVs)

The GPP can also be used to plan paths for UAVs. In order to do so, we create a 2½-D surface located at an altitude h_{min} from the ground. The elevation values resulting from adding h_{min} to the elevation of the terrain define the minimum safe altitude for the UAV at each location (x,y). With this elevation, we can calculate visibility, coverage, exposure and time, and produce a path for the UAV that minimizes the composite objective function, while guaranteeing safe navigation.

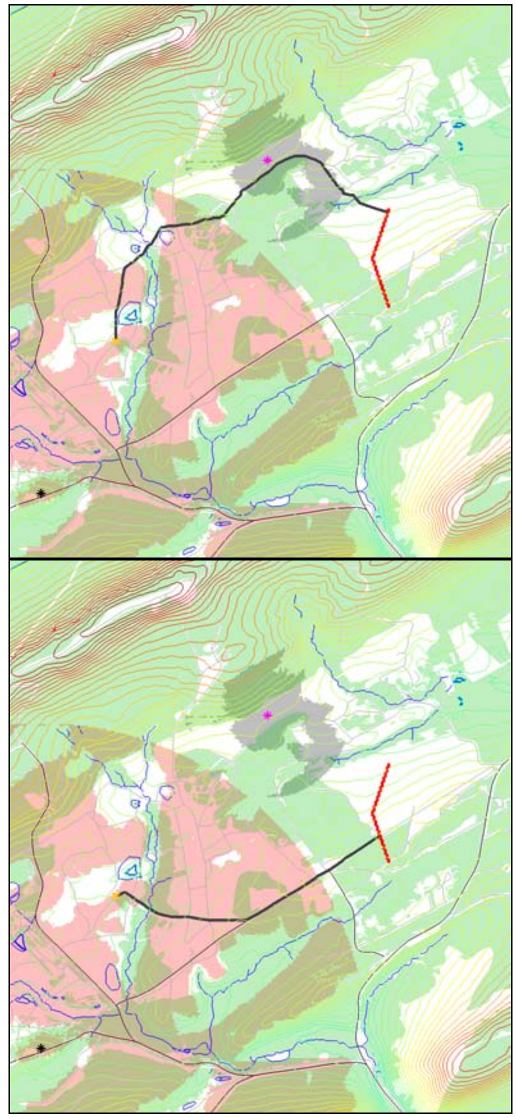


Figure 9. GPP using time as a hard constraint: Top: path generated when the total travel time constraint is set to 6 minutes, and the other layers are weighted evenly. Bottom: path generated when the total travel time constraint is set to 3 minutes.

III. RESULTS AND CONCLUSIONS

The Geometric Path Planner is a high level planner able to incorporate existing maps and to plan paths that optimize multi-metric objective functions and satisfy constraints. It uses prior geographic data (digital elevation maps, and digital feature maps), information about known terrain objects and mission-specific parameters (etc., goal points and lines, boundaries, no-go regions) and combines them to create a global planner that is aware of mission requirements.

The GPP provides efficient replanning, making it very well suited for dynamic environments, such as those encountered in unmanned vehicles. Even though the GPP was originally designed for unmanned ground vehicles, it has been successfully extended to support unmanned air vehicle operations as well.

The GPP is being used as a part of General Dynamics Robotic Systems (GDRS) Decision Support System (DSS) to plan scout missions. The DSS runs on the Operator Control Unit and produces plans for the GDRS XUV vehicles. These plans were driven by an XUV vehicle during experiments conducted at Ft. Indiantown Gap in May and June of 2004. The GPP is also being used in a Stryker vehicle to support a FCS-like combat crew station, also for mission planning.

Besides actual field testing, the planner has been thoroughly tested in simulation. 2.5 million routes have been planned for a total of 3.5 million miles. Of these, 2000 miles were also driven in a UGV vehicle simulator.

IV. ACKNOWLEDGMENTS

This work was sponsored by the U.S. Army Research Laboratory under contract "Robotics Collaborative Technology Alliance" (contract number DAAD19-01-2-0012). The views and conclusions contained in this document do not represent the official policies or endorsements of the U.S. Government.

V. REFERENCES

1. D. FERGUSON AND A. STENTZ. "Field D*: An Interpolation-based Path Planner and Replanner". In Proceedings of the International Symposium on Robotics Research (ISRR), October, 2005.
2. D. FERGUSON AND A. STENTZ. "The Field D* Algorithm for Improved Path Planning and Replanning in Uniform and Non-Uniform Cost Environments". Carnegie Mellon Robotics Institute Technical Report CMU-RI-TR-05-19, June 2005.
3. A. STENTZ. "Optimal and Efficient Path Planning for Unknown and Partially-Known Environments". In Proceedings of the International Conference on Robotics and Automation, May 1994.
4. A. STENTZ. "The Focussed D* Algorithm for Real-Time Replanning". In Proceedings of the International Joint Conference for Artificial Intelligence, August 1995.
5. A. STENTZ. "CD*: a Real-Time Resolution Optimal Re-planner for Globally Constrained Problems". In Eighteenth National Conference on Artificial intelligence (Edmonton, Alberta, Canada, July 28 - August 01, 2002). R. Dechter, M. Kearns, and R. Sutton, Eds. American Association for Artificial Intelligence, Menlo Park, CA, 605-611.