GEMMP - A Google Maps Enabled Mobile Mission Planning Tool for Autonomous Underwater Vehicles

Steven D. Seeley Jr., Ramprasad Balasubramanian
Computer and Information Science, University of Massachusetts Dartmouth, North Dartmouth, MA, 02747

Article Info

Article history:

Received Apr 10, 2012
Revised May 28, 2012
Accepted Jun 5, 2012

Keyword:

Android
Autonomous underwater vehicle
Google Maps
Mission planning
Mobile devices
Tablet

ABSTRACT

Many applications for mobile robotics involve operations in remote, outdoor environments. In these environments, it can be difficult to plan missions dynamically due to the lack of portability of existing mission planning software. Mobile platforms allow access to the Web from nearly anywhere while other features, like touch interfaces, simplify user interaction, and GPS integration allows developers and users to take advantage to location-based services. In this paper, we describe a prototype AUV missionplanner developed on the Android platform, created to aid and enhance the capability of an existing AUV mission planner, VectorMap, developed and maintained by OceanServer Technology, by taking advantage of the capabilities of existing mobile computing technology.

Corresponding Author:

Ramprasad Balasubramanian
Computer and Information Science,
University of Massachusetts Dartmouth
North Dartmouth, MA, 02747
Email: r.bala@umassd.edu

1. INTRODUCTION

Being able to rapidly deploy any sort of autonomous agent significantly amplifies its capability and usefulness. When it comes to autonomous vehicles, be it ground, surface, underwater or aerial, their performance is enhanced by the amount of domain information that is provided to it a priori. This a priori information allows them to interact with their environment effectively and provides the operator opportunities to design the mission as carefully as possible. Most autonomous vehicles are equipped withsensors and controllers (front-seat and/or back-seat) that allow them to deal with unexpected situations in the environment. While domain information is remarkably helpful for improving the odds of a successful mission, they also take a long time to pre-process and integrate into the mission plan for the vehicle. In many cases the information is simply not available. In other cases the operator must be willing to take advantage of opportunistic conditions to deploy the vehicle rapidly in the field. Most current mission planning software sufficiently provides the capability to design and execute detailed and complex missions. The downside is the amount of time it takes and the a priori information that is required. While the ensuing discussion and description can be applied to any modality we confine our attention to Autonomous Underwater Vehicles (AUVs).

One of the key issues we wanted to address with this software was the lack of portability associated with many existing mission planning tools. This is an issue because it allows no likelihood of on-site mission planning. Instead, a mission must be planned in some off-site location. Issues can arise where the mission plan needs to be modified after arriving at the mission location due to some unexpected environmental conditions. For example, unexpected debris might have fallen into the body of water, or water depth could have significantly dropped since the last satellite image was taken. In these cases, the path of the vehicle will have to be altered to avoid running aground or a mission-fatal collision.

Since our nominal mission involved having an AUV travel through a river, another limitation revealed itself. Existing AUV mission planning software is often designed exclusively for missions in deep open water bodies such as the ocean or lakes, but more unusual environments, like rivers and streams, are less common, and therefore, support for them is limited. The standard input devices, such as a keyboard and mouse work for point-by-point waypoint placement, but planning a mission through a winding river is difficult without the ability to generate waypoints automatically through some means, like freehand drawing. Today’s mobile devices, especially tablet devices, with their large screens and haptic touch screen interfaces, allow the operator to draw directly on the map which the AUV will operate.

2. BACKGROUND

In this section, we will present a brief background on some of the technologies used to implement our prototype application.

2.1. Android Platform

The Android SDK for smartphones and tablet devices is open source, platform independent, and supports integration with Google Maps through the Android Google Maps API. It provides abstracted and simplified access to mobile device’s hardware components, such as the accelerometers, GPS, and touch screen.

The Android SDK provides an easy to use interface for interacting with the GPS in a number of ways, including retrieving instantaneous GPS fixes as well as scheduling periodic updates. It also provides an simple interface for handling touch input from a user interacting with the touch screen.

2.2. Google Maps

Google Maps is a free, non-commercial mapping API service provided by Google for displaying geo-referenced satellite maps on web pages and mobile devices. This service is provided to mobile devices running on the Android platform through the Android Maps API. This API comes packed with the Android SDK, and provides several useful tools for creating map-based applications for Android devices.

The Google Maps API for Android provides an extension to the Android SDK which allows for the use of Google’s map-based services on mobile devices. The main feature of the Android Google Maps API that we will focus on is the map. The map interface provided by the Google Maps API has the capability to display and interact with a dynamic Google-based satellite map. The map images are downloaded automatically and dynamically from Google’s map database, which requires an active Internet connection. To look at a different location, the user simply needs to interact with the map, by touching and dragging to a location, or they can provide a GPS location through which the map service can display a map centered about that location.

Additionally, each map has a projection, which determines how the latitude and longitude coordinates map to screen coordinates within the view. This mapping is based on the zoom level as well as the positioning of the map. The projection is updated automatically any time the position or zoom level of the map is changed. It can be used to draw objects onto the map by converting geo-referenced points into screen points. Alternatively, the projection can be used to convert screen points into geodetic points, which is particularly useful for detecting where screen touch events coincide on the map.

2.3. OceanServer’s Vector Map and UVC

VectorMap is a proprietary mission planning application developed for Windows, by OceanServer Technologies, specifically for creating mission files designed for use with their UVC control system. UVC is software designed by OceanServer Technologies which runs on the AUV computer and is used to translate the mission file into actions the AUV can perform which allows the AUV to execute the mission. VectorMap, in conjunction with UVC, provides a wealth of features which are relevant to typical missions performed by their AUV vehicles, including: downloadable geo-referenced map imagery, waypoint placement with various control settings, and safety settings to prevent damage or loss of the vehicle. The goal of our application, GEMMP, is to allow the capability of VectorMap to be used on a mobile device, to enhance the mission planning experience, by making it easier and more convenient to use.

2.4. Other Technology Used

The Java Topology Suite (JTS) API © is a computational geometry API which we use to implement our no fly zone feature. It has support for validation, cleaning, integration, and querying of spacial data sets, and allows us to create and manipulate geometries within our application.

2.5. Other work

We looked at a project by Ian Katz at the MIT Sea Grant AUV lab using Google Earth in a web-based mission planner using AJAX. There was very little documentation on this project, but there was a video, linked here [4], which provided some of the initial inspiration and guidance for using Google Maps for a Mission Planner.

3. APPLICATION FEATURES

Our mission planning application builds on the existing technologies outlined in the previous section to provide similar functionality to that of OceanServer's VectorMap mission planning software. In addition, our goal is to provide a portable and quickly accessible mission planning experience by adding features which take advantage of existing mobile device capabilities. Two of these main capabilities of interest in this application were the integration of GPS and touchscreen support. For the purpose of mission planning, this meant we were to integrate automatic map-retrieval from GPS position and automatic generation of waypoints from hand-drawn curves using the touchscreen, in addition to supporting many of the features provided by VectorMap. A demo showing a majority of the features can be found in this link.

3.1. VectorMap Features

We defined several classes representing different planning elements in VectorMap.

3.1.1. VectorMap Waypoints

For compatibility with OceanServer's proprietary VectorMap mission file format, GEMMP can produce waypoints which are identical to waypoints produced by VectorMap. Each VectorMap waypoint has a number of settings which tell the vehicle how to get to that waypoint and what to do when that waypoint is reached, including:

- The latitude and longitude of the waypoint in degrees.
- The radius, in meters, of the circle surrounding waypoint which the AUV must come within for the AUV to consider that waypoint reached.
- The speed at which the AUV will travel while approaching this waypoint in meters per second.
- The length of time, in seconds, that the AUV should park at that waypoint.
- The heading from the last waypoint to the current waypoint in degrees, ranging from 180 to -180 with 0 being due north.
- The distance from the last waypoint to the current waypoint in meters.
- The depth of the next waypoint, and whether this depth should be measured as height from bottom or depth from surface.
- Various sidescan sonar, and mulibeam sonar configurations which are currently supported by VectorMap, including device name, range, frequency, gain, and channel settings.
- A geodetic point referring to the point displays on the Google map.

3.1.2. Lawnmower Patterns

A lawnmower pattern is a standard sweeping scanning pattern which is created for the purpose of having the AUV survey of some rectangular area. Lawnmower patterns in VectorMap and GEMMP are defined by the following settings:

- The latitude and longitude of the corners of the survey area.
- The distance between each leg of the scan, which can be determined by setting a static distance, in meters, or automatically based on the range of the sonar.
- The depth of each leg of the sweep, and the surfacing depth for entering the next leg.
- The distance to surface and descend each sweep and the distance needed to adjust the position entering the next leg.
- The sensors to activate during each leg, such as the sidescan sonar or multibeam.
- The rotation of the rectangular area to be covered by the Lawnmower pattern. An example of a

lawnmower pattern is shown in Figure 1.

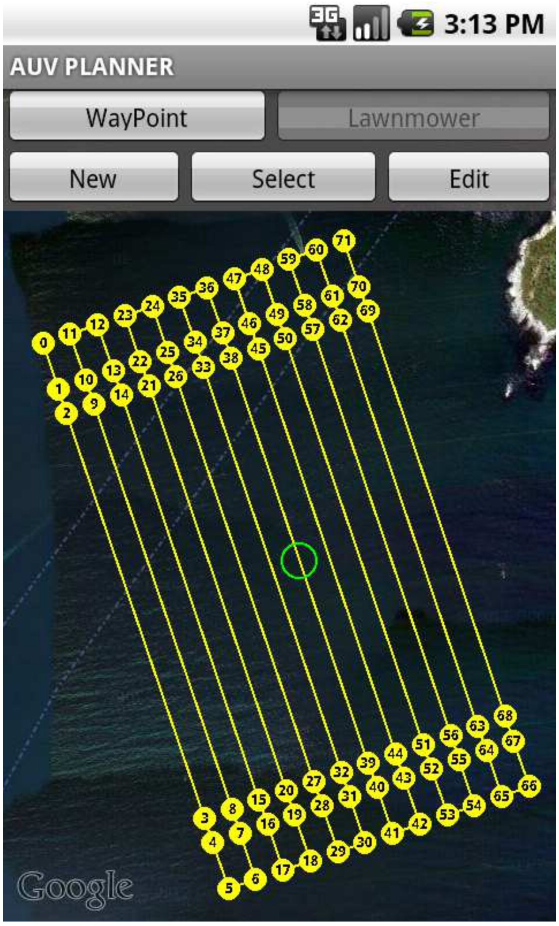


Figure 1. A Lawnmower pattern in GEMMP

3.2. GEMMP Additional Features

In addition to the features mentioned above, we have added two other features to aid in the planning process. How these features work in the context of the application will be described later, but they will be introduced below.

3.2.1. Waypoint Templates

A waypoint template is a set of waypoints which can be saved to a file and are defined relative to reference point. In other words, the waypoints in a template are not defined by absolute latitude and longitude coordinates, but rather by their relative position in relation to a reference point which is set upon the creation of the template. This is particularly useful for special types of mission patterns, such as a star pattern, which has a specific and complex path the AUV must take, but whose position is not specific to one absolute location, and should be able to be performed easily in any location. An example of using a waypoint template to create a repetitive pattern is shown in Figure 2.

3.2.2. No-Fly Zones

Due to tidal changes, the depth at certain areas on the map can change dramatically. The depth changes may cause some areas to be dangerous where damage or loss of the AUV could result if they are entered. No-Fly Zones represent these areas where the AUV should not go, and which should be avoided. No-fly zones consist of a set of geodetic points on the map which define the vertices of the polygonal hazard area.

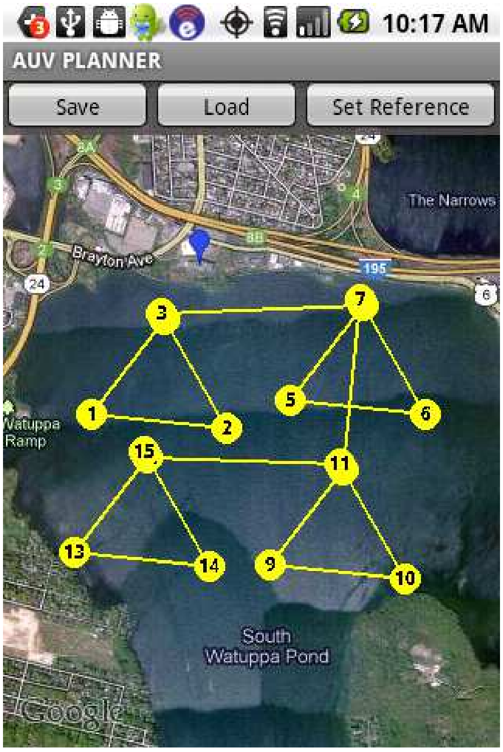


Figure 2. Using a waypoint template to insert 4 identical triangular waypoint patterns.

4. MISSION PLANNING OPERATION

In this section, we will address some of the design decisions and the various capabilities of the application, how they are performed, and the various algorithms used to accomplish these actions.

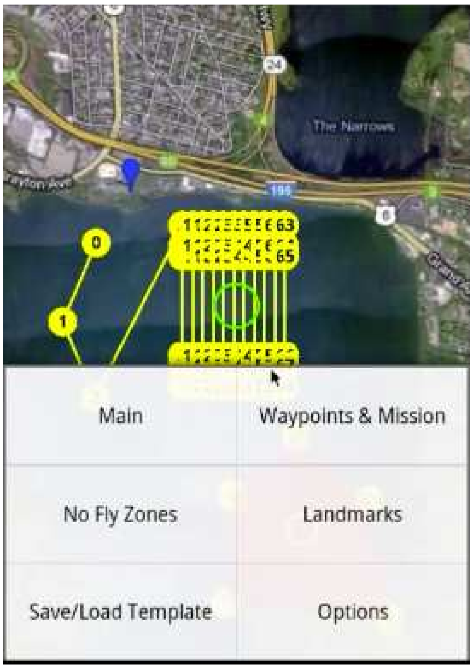


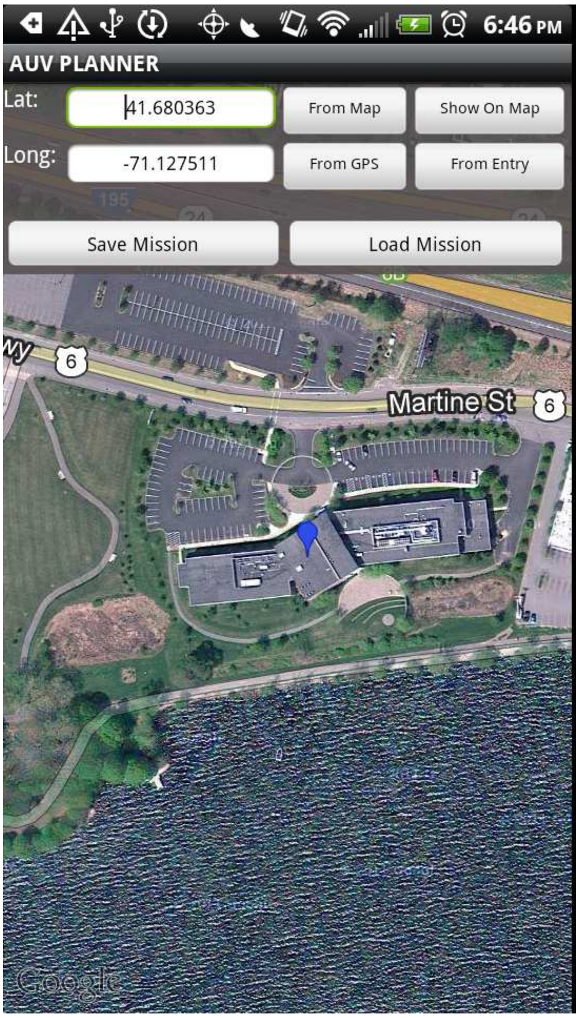
Figure 3. The options menu.

Since the screen real-estate and processing power of mobile devices are often limited, instead of having a single GUI menu to serve all of our needs, we had to divide the application into several operating modes. These interfaces are selected from the option menu shown in Figure 3 which is accessed using the hardware specific option button. A menu with the different interfaces available is displayed, which include: the main map interface, waypoint entry and mission interface, the template load/save interface, no-fly zone interface, and defaults and safety settings interface. Across all of these interfaces, a single map is used, but how the user interacts with this map changes depending on the selected feature. The selected feature reflects the user's desired actions. For each interface, a different heads-up display (HUD) menu is shown to allow the user to select from the actions taken in this interface.

To provide these different options for the map, we introduced an extension of the map interface provided in Google Maps which provides different modes through which all interactions involving the map are facilitated. Each mode provides a different set of actions which can be done to update the map or modify the mission. A description of each of these modes is given in this section.

4.1. Main Map Settings

This is the interface, shown in Figure 4, which the application displays initially, and provides the user with several options for overriding how the map image is retrieved, as well as options for saving and loading mission files. In this mode, the user can choose to enter a GPS location manually via two text fields for the latitude and longitude, which will update the area viewed on the map to be centered about that location. Alternatively, the user can select this location from the map by tapping an area in the map on the touchscreen.



Lat: 41.680363

Long: -71.127511

Figure 4. Main Map Settings interface.

The location is also marked with a blue marker on the map. This location is treated as a return point. In case the user pans away and loses this location, they can click a button to re-center the map on that location. By default, this location is retrieved automatically through a GPS device, which upon receiving the location update, the map is automatically marked and centered at that location. At any time, the user can swipe their finger to pan around the map, and as long as there is an active Wi-Fi or 3G connection, the screen will automatically download and refresh the map.

From this same interface, the user can load a mission file on the mobile device's SD card, and that mission file will be parsed for the waypoints which will automatically be placed in the correct location on the Google map with the appropriate settings. At which time, the map will also be re-centered to show these waypoints on the map. The user can also save the mission file from this menu. When the user clicks the save button, the waypoints will be parsed for all their settings and will be encoded in the.mis file format. The .mis format is defined and supported by VectorMap and the UVC control architecture. The AUV will use this .mis file to define its path and take intelligent actions during operation.

4.2. Waypoints and Mission

This mode is entered by selecting *Waypoint & Mission* from the options menu, and handles any and all functions related to the mission specification, including waypoint placement and changing settings for those waypoints.

Geo-referenced waypoints can be added to the Google map in one of several ways. A user must first click a button on the HUD to enter waypoint placement mode for the map. While in waypoint placement mode, the map is frozen in place, and the user can interact with the touch screen to add waypoints.

4.2.1. Individual Waypoints

A single tap on the map will place a waypoint at that location on the map with default settings. The waypoint is added to the map by retrieving the screen position of the touch event and using the map projection to convert itinto a geodetic point which is added to the map as a waypoint.

4.2.2. Smooth Curve Waypoint Generation

An important feature and unique feature that we included in order take full advantage of the convenience of the touchscreen, is the ability to draw a curve, and from that, have waypoints generated automatically. By touching and dragging the user's finger or stylus across the map, a curve is drawn on the map. When the user lifts their finger or stylus, the waypoint generation process begins. The list of points sampled while drawing the curve is processed to generate a smaller set of discrete waypoints fitting the curve.

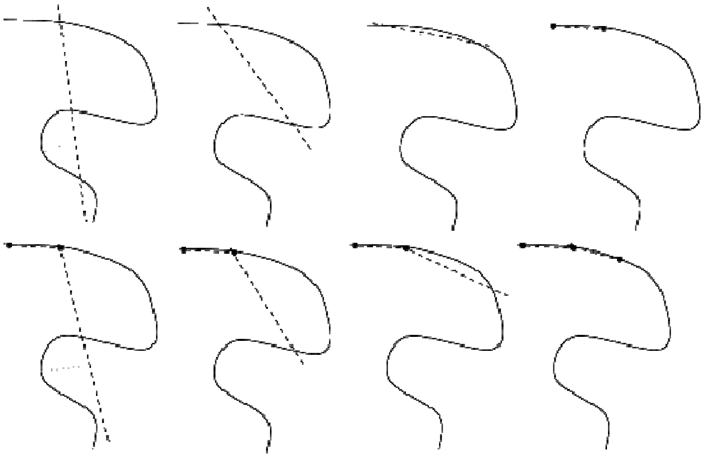


Figure 5. Line fitting of a curved path

To break this non-linear curve into a set of waypoints with linear paths, a piece-wise linear sampling is done on the curve by testing how each point fits to a line, and shortening that line until the fit value is below some threshold. The algorithm worksas follows: a line is generated using a linear regression on a sampling of points in the curve. The following equations obtained from [2] are used to calculate the fit line:

$$slope = \frac{n(\sum x\, y) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$intercept = \frac{\sum y - n(\sum x)}{n}$$

In these equations, x and y are the x and y screen coordinates of each point in the sampled portion of the curve, and n is the total number of sampled points. After the line is generated, it is then compared to the curve points using the sum of least squares error in the y direction. If the fit value falls below a threshold, the curve is divided in half, and the fit value is recalculated for the smaller curve. This is repeated, with the line getting shorter and shorter, until the curve fits the line within some reasonable threshold. A step-by-step diagram of this is shown in Figure 5. This process is repeated for the rest of the curve until the curve is fit to a set of lines. The endpoints of these lines are then converted into geodetic points and are added to the mission as waypoints. An example of how this is done in the application is shown in Figure 6.

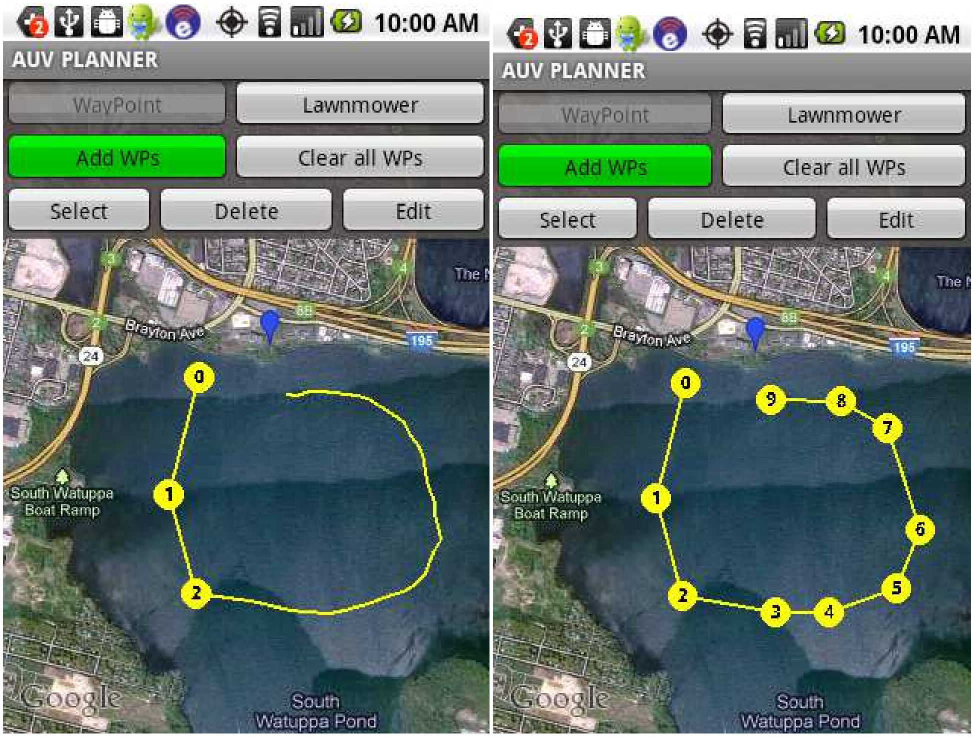


Figure 6. A drawn curve (left) and the resulting waypoints after applying the line fitting algorithm (right) in GEMMP.

It can be problematic for the mission if waypoints are too far apart from each other, especially if the vehicle is traveling a substantial distance underwater, where GPS fixes are not available. To prevent this, a minimum and maximum waypoint distance can be defined in the application, to ensure that, in the curve-fitting process, waypoints are not generated too close or too far away from each other. This distance is defined in meters and is set in the default options menu, which is described later. While the curve is being fit with straight lines, the algorithm uses the minimum and maximum distances to make sure it does not cut the line to short or too long. When the fit line is too long, it will simply be cut once again, even if it sufficiently matches the curve. If waypoints are generated in such a way that a turn would be done underwater, a set of three waypoints will be generated consisting of a surface point, a turn alignment point, and a dive point so the vehicle does not perform a turn underwater. This is a problem because while the vehicle is underwater, it uses pure dead-reckoning for navigation, so it has a tendency to accumulate position error especially when turning [probably want to cite this].

4.2.3. Lawnmower Patterns

The third way to add waypoints is to create a lawnmower pattern, which is effectively a sweeping raster scan type pattern used for surveying some rectangular area of the location. This is done by clicking the button to create a new Lawnmower pattern, and then dragging a rectangular box on the map. After releasing the touchscreen a dialog box pops up with various fields for the user to adjust the spacing, depth, dive distance, and adjustment distance. When the user clicks the accept button in this dialog box, the application generates the waypoints with the modified parameters in a sweeping raster scan pattern to cover the rectangular area. The user can change these parameters at any time by clicking the button to select a lawnmower and then clicking the button to edit the lawnmower. Additionally, the user can move and rotate the Lawnmower pattern by interacting with the touch screen. While the lawnmower is selected, the user can touch the center of the lawnmower and drag their finger to reposition the lawnmower in a different location on the map. Also, the user can click outside the lawnmower and drag around the center to rotate the lawnmower about its center point.

4.2.4. Modifying Existing Waypoints

After waypoints have been generated, the user will often want to change some of the settings for those waypoints. In the Waypoint and Mission menu, the user can click the button to select waypoints, which switches the application into waypoint selection mode. Now, if the user touches the screen area over a waypoint, that waypoint will turn green indicating it has been selected. The user may also touch and drag a rectangle to select multiple waypoints, which will remain on screen until the user drags another box or leaves waypoint selection mode. With some waypoints selected, the user can click the button to edit waypoints, which brings up a dialog window with fields and drop-down boxes for changing all of the settings which were mentioned previously. In addition, when a waypoint has been selected, the user can again touch the area over that waypoint and drag their finger which will drag the waypoint from its current position. When the user lifts their finger, the position of the waypoint will be updated to the new position. When there are multiple waypoints selected, the user may touch anywhere within the selection box and drag to reposition all the selected waypoints.

4.2.5. No-Fly Zones

One of the biggest limitations of most AUV mission planning applications is their inability to rapidly change their paths. A problem frequently faced by those who deploy AUVs is the change in water depth due to tidal changes.

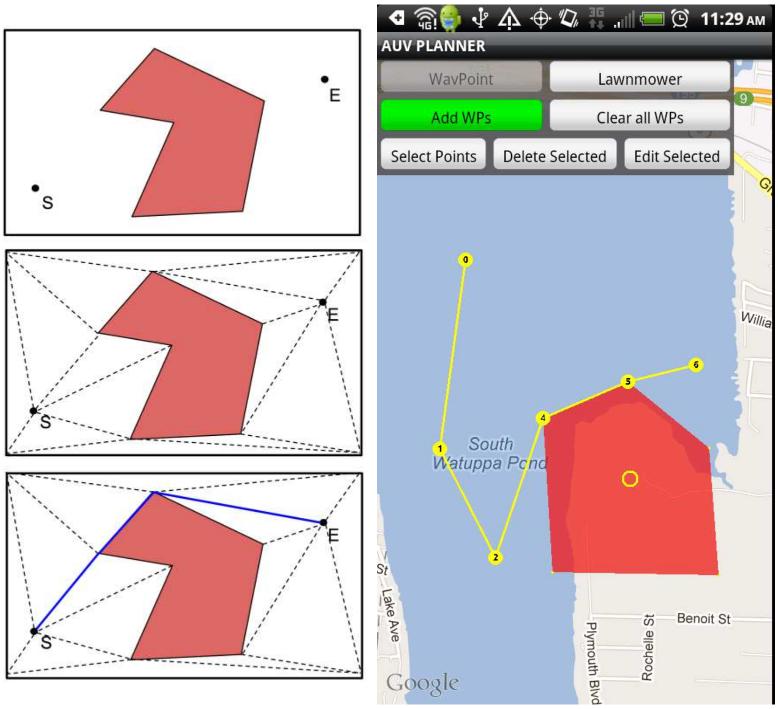


Figure 7. Algorithm to find a path around a no fly zone (left) and an example path generated around a no fly zone (right)

Often this will require the operator to either wait until high tide or change the mission in order to prevent the vehicle from running aground. GEMMP provides the ability to create no-fly zones on the fly that will modify the path of the AUV to avoid areas that could be potentially hazardousto the AUV. Even after a mission has been created and waypoints generated, GEMMP allows the operator to create no-fly zones, which are closed polygonal regions where the AUV will not be allowed to travel over. After such a region or regions are defined, GEMMP automatically adjusts the waypoints around these regions. Details of this step are given below. The algorithm we use consists of 3 major steps derived from algorithms discussed in [3] and is shown in Figure 7.

1. Given the rectangular area visible on the screen and the polygonal areas given by the no-fly zones, create a rectangular polygon with holes, where the polygonal shell is the rectangular area surrounding the area of the map visible on the screen, andthe holes are the polygonal areas defined by the no-fly zones.
2. From this polygon, create a mesh connecting all of the vertices of the polygon which do not pass through no-fly zone. The edges in this mesh represent all the possible paths the AUV can take to get from its start to destination point without passingthrough the no-fly zone.
3. Given this mesh, construct a graph of nodes with connections between nodes representing edges in the mesh. Using Dijkstra's algorithm [4], calculate the shortest path through this graph, and then create waypoints from the vertices of this path and add them to the mission plan.

Additionally, the no fly zones also shorten the legs of a lawnmower pattern such that it remains outside of a no fly zone. This is done by detecting if a leg intersects with the boundary of the no fly zone, and finding the intersect point and adjusting the leg such that it ends at that intersection point. An example of an adjusted lawnmower pattern is shown in Figure 8.

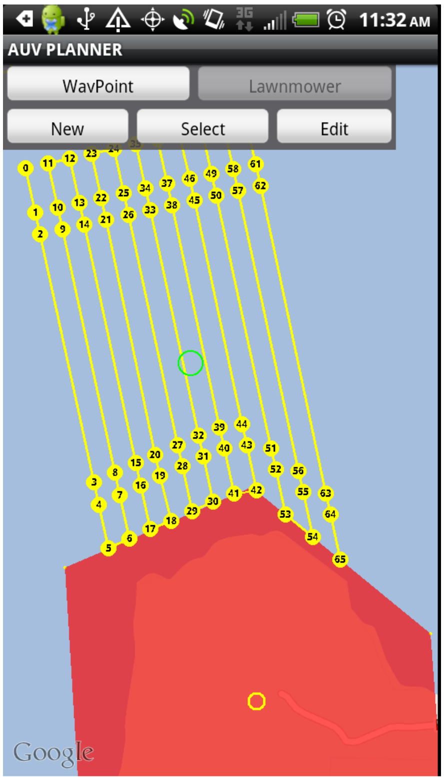


Figure 8. Lawnmower pattern adjusted to remain outside of a no fly zone.

4.3. Waypoint Templates

It is very common to have the same pattern used in several different missions. GEMMP provides the ability to create pattern templates on the fly. The template interface provides the user with the ability to take the waypoints which are currently added to the mission, and create a template from them, or to load an existing template into the current mission. In either case, the user selects a reference point by clicking the Set Reference Point button and then tapping the screen, which is then marked with a small circle indicating that location is the reference point. If the user clicks the Save Template button, the template will be saved such that the relative distance of the waypoints to that reference point is saved. When the user selects Load Template, they are prompted to select a saved waypoint template file, and upon doing so, the waypoints specified in the template file will be added to the map with the same relative distance from the reference point provided, even if this reference point is different from the point chosen when saved.

4.4. Defaults and Safety Settings

The Defaults and Safety Settings interface is a separate activity window which handles modifications to the default values of the waypoints, such as speed to the waypoint, as well as application specific values, such as maximum and minimum waypoint generation distance for curves. From this interface the user can also modify OceanServer's proprietary VectorMap safety settings which handle conditions where the safety of the AUV could be compromised, which if triggered, can allow the AUV to automatically abort the mission.

5. DEPLOYMENT

In this section we'll briefly describe the overall process of planning the mission and deploying it on an AUV. The process begins by opening the mission planner application and setting or retrieving the mission location to display the map. When the mission area is visible on the map, the user may then enter waypoints individually, by drawing curves, or creating a box for a lawnmower pattern. The user can also create no fly zones which can be used to avoid certain areas on the map. The user can modify the waypoints at any time by selecting them and pushing the button to edit their parameters or position. When the waypoints have been placed, the user can then set the safety settings for the given mission, and save the mission file. After the user saves the mission file, they may send the file wirelessly to the AUV's computer over a Wi-Fi local area network connection, and the AUV can be commanded to start the mission using remote desktop protocol.

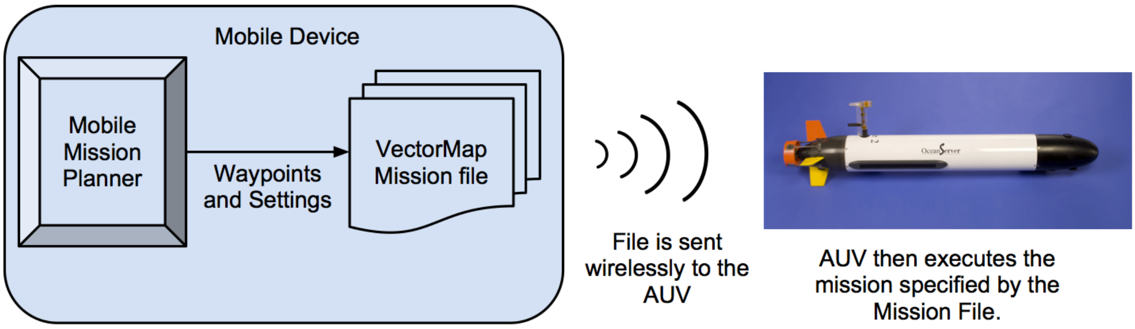

Mobile Device
Mobile Mission Planner
Waypoints and Settings
VectorMap Mission file
File is sent wirelessly to the AUV
AUV then executes the mission specified by the Mission File.

Figure 9. Mission Planning and Deployment process

6. EXPERIMENTAL SETUP AND RESULTS

To test the basic capabilities of this application, we installed the application on a Nexus One phone running Android 2.2.1 as well as an Archos 10.1 Tablet computer running Android 2.1, and planned a simple mission in South Wuttappa Pond in Fall River, MA. After the mission was created, it was verified in VectorMap for the proper safety settings as well as the correct depth and speed values. It was then transferred to the AUV by a remote file transfer over a Wi-Fi network, and loaded remotely into UVC as per OceanServer's usual procedure. After confirming that UVC accepted our mission file, the AUV was then deployed.

Figure 10 shows the mission we created in GEMMP. The specified mission path is shown by the solid line and the overlaid experimental log file results is shown by the dotted line. The mission consisted of a typical lawnmower pattern with 3 legs, and for safety reasons only going to a maximum depth of 1 meter from the surface. The lawnmower has two turns. The first is a standard turn where the AUV surfaces to make the turn. The second turn is done entirely underwater. We tested these two kinds of turns to match our expectations for the AUV behavior with the log file results, which helps verify that the AUV is doing what we expect it to as specified in GEMMP.

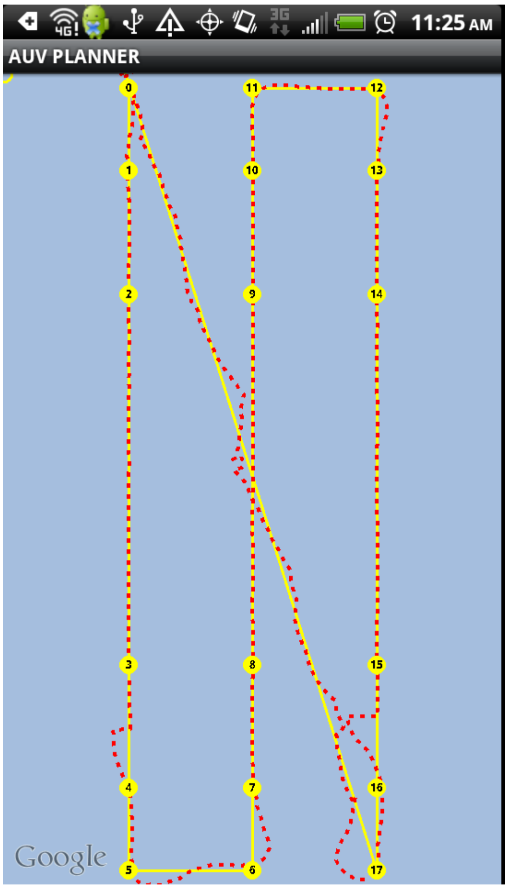


Figure 10. Log File Results of Experimental Testing

The first turn is marked by the waypoints labeled 3 through 8, and as we expected, there is a jump in the logged position between 3 and 4, because the AUV surfaces and retrieves a new GPS fix, correcting its position estimate. The second turn is given bythe waypoints 9 through 14, and since the AUV does not surface as it would normally between waypoints 9 and 10, there is no jump in the logged position. However, when the AUV returns to the surface between waypoints 15 and 16, there is a significantly large jump in the position, due to the accumulated drift from the AUV turning underwater and remaining underwater for a substantial time without surfacing for a GPS fix. These test results verify that our application does indeed allow us to design missions which are compatible with OceanServer's UVC control system, and we were successfully able to execute a mission using a mission file generated by GEMMP.

7. FUTURE WORK

What we have developed is an application with support for few features taking advantage of this mobile technology. Such an application has many uses in different modalities where AUV missions need to be created on the fly. There are numerous applications of mobile technology in this area, many of which are not addressed in the development of this application. The next step might be having the mobile device communicate directly to the vehicle using long range wireless communications. Using a mobile device, we should be able to visualize points of interest detected by the AUV's sensors on the mobile device, and based on this, be able to change the mission plan on the fly, so the AUV could then dynamically receive the new mission begin executing it while abandoning the old mission.

ACKNOWLEDGMENTS

We would like to acknowledge the Coastal Systems Technology team at the University of Massachusetts Dartmouth for their help and support and OceanServer Technologies for allowing us to analyze and use their VectorMap software and their AUVs.

REFERENCES

[1] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). ”Section 24.3: Dijkstra’s algorithm”. Introduction to Algorithms(Second ed.). MIT Press and McGraw-Hill. pp.595601. ISBN0-262-03293-7.

[2] Stefan Waner and Steven R. Costenoble (1998). Fitting Functions to Data: Linear and Exponential Regression [Online]. Available: http://people.hofstra.edu/stefan waner/realworld/calctopic1/regression.html.

[3] Subhash Suri (2002). Polygon Triangulation [Online]. Available FTP:http://www.cs.ucsb.edu/Directory:_suri/cs235/ File: Triangulation.pdf

[4] Ian Katz. ”Interactive AUV Mission Planning with Google Earth.” Internet: http://www.youtube.com/watch?v=vK0wWH9Ijnk,Apr. 5, 2010 [March, 2012].

[5] OceanServer Technology, Inc. ”OceanServer: Battery Systems, Sensors, and Robotics.” Internet: http://www.ocean-server.com [March, 2012].