

Autonomous Vehicle Guidance and Control using OpenStreetMap and Advanced Integration Techniques

Fahad Ullah, Qaiser Habib, Muhammed Irfan, and Khawaja M. Yahya

Abstract—Autonomous Guided Vehicles (AGVs) or Autonomous Vehicles (AVs) are in widespread use in many applications such as bomb disposal, underwater exploration, industrial transport, etc. Control and guidance are vital aspects of AV research and many techniques have been proposed in literature which range from fully autonomous and intelligent systems to laser and radar guided systems and line followers. This paper presents a new technique for guiding and controlling autonomous vehicles by using the OpenStreetMap (OSM) map library. An AV with a built-in GPS receiver can be controlled effectively from a central computer which has the OSM powered control mechanism installed. The system proposed in this paper can control the AV in an online-mode or offline-mode depending upon the quality of the link between the AV and the central computer. A sample implementation of the system is also given.

Index Terms—Open streetmap, autonomous vehicles, autonomous guided vehicle, intelligent vehicles.

I. INTRODUCTION

Automated guided vehicle (AGV) or Autonomous Vehicle (AV) is a driverless mobile robot that can be controlled and navigated across a surface (terrestrial AGV) or through some medium (air or water) remotely. Automated vehicles and machinery are an essential part of today's industry. In any industrial setup, most of the machinery is automated, particularly that related to production. Unmanned Aerial Vehicles (UAVs) are another common class of automated vehicles used for reconnaissance and attack by military forces. Terrestrial AGVs are still in evolution, although most of these vehicles are used by military these days for instance for bomb disposal, yet the notion that AGVs can replace many current transportation means is highly rated.

Since the advent of Automation, the guidance and control of an AGV became the primary focus for research groups. Various algorithms have been developed for the guidance mechanism and different techniques have been implemented for the control of the autonomous vehicles. Some of the techniques use guide tape, laser [1] and even gyroscopic guidance. For instance, in a vision based lane detection of an autonomous vehicle [2], different algorithms were used to get the optimum guidance and navigation results. In another method [3], a path generation algorithm was developed using

a sensor platform and two electric motors which make the panning and tilting motions. Similarly, to avoid collisions and deadlocking of AGVs, various schemes are devised [4].

In this paper, a low-cost and very simple autonomous vehicle guidance system has been presented. OSM can serve as a graphical and easy to use way to handle the control and navigation of an AGV. The OpenStreetMap system is a collaborative project i.e. it can be edited or modified by anyone, just like Wikipedia [5]. A desktop-based implementation has been proposed in this work to ensure reliability, speed and security. An advanced integration technique has been proposed for coupling the web-based OSM API with the desktop based front-end application.

Section III of the paper gives an introduction to OSM and briefly outlines its advantages over other popular mapping systems. Section IV presents the system architecture of this project, section V describes the working principle with the help of a flow chart, section VI details the implementation of the system, section VII explain the output of the sample implementation and section VIII and IX describe the future work possibilities and conclusions respectively.

II. RELATED WORK

OpenStreetMap has been widely used in numerous scientific and research endeavors. Oliver O'Brien [6] proposed an accurate way to create street Orienteering maps of areas with complete OSM coverage. He predicted that it would make both the orienteering and OSM communities benefit from field-checking observation and although the process itself wouldn't be as easy as it must be for widespread adoption all over the community, yet the automation capabilities of systems like Quantum Geographical Information System (GIS) and further development of open-source GIS and source data would enhance the process.

Ricky Jacob et al.[7] demonstrated a web-based, multilingual campus guidance system with emphasis on pedestrian navigation to provide support for delegates attending International conferences at the National University of Ireland Maynooth campus. They used OSM for creating the interface and the system would generate the shortest pedestrian paths. Visual assistance was also integrated in the system using geo-tagged images.

Michael J. Casey [8] in his paper provided details on the state-of-the-art citizen mapping all over the globe. He presented the associated advantages and disadvantages and the work done by mapping agencies to counteract the issues. He speculated about Citizen Charting and analyzed various associated questions.

Manuscript received December 11, 2010 ; revised June, 22, 2011

Fahad Ullah, Muhammad Irfan and Khawaja M. Yahya, University of Engineering and Technology, Peshawar, Pakistan (email: fahadullah@nwfpuet.edu.pk; irfan697@gmail.com; yahyakm@yahoo.com).

Qaiser Habib, Concordia University, Montreal, Quebec, Canada (email: qaiser.habib.khan@gmail.com).

III. OPENSTREETMAP

OpenStreetMap(OSM) is a collaborative project to create free and editable maps of places in the world. Unlike popular map sources like Google Maps, OSM relies mostly on data from portable GPS devices, aerial photography or simply local knowledge. This implies that there is no guarantee to the accuracy of the map data. Nevertheless, since every user is allowed to edit the map content, there is a high chance of the rectification of errors with the passage of time.

Currently, the OSM API provides almost all the functionality needed for AGV guidance and control. The advantage of using OSM is that it does not have stringent license issues and hence a wide variety of applications can be created using OSM at the expense of MAP accuracy. The Google maps API's terms of service [9] explicitly prohibit the application of the service for automobile guidance and tracking applications. Map accuracy keeps improving with the passage of time as more and more sources are added to the OSM map data and keeping in view the progress OSM has made in the last few years, we can safely predict that in the future, OSM maps will be considered as much authentic as Google or any other mapping service for that matter.

OSM maps rely on Javascript for displaying their maps and all the map functions are derived from the OpenLayers mapping library. A map is created by instantiating the OpenLayers.Map object which takes the ID of a DIV tag in which the map is to be displayed as its parameter. OSM maps offer numerous features like a collection of controls, markers (also called permalinks), event handlers, Asynchronous Javascript and XML (AJAX), etc. [10].

IV. SYSTEM MODEL

The system proposed by this paper is a desktop application since desktop applications have the required speed, power and security for AV control. OSM maps, however, work in a web-based environment. Therefore, this paper presents a model which comprises of three components: The web based component, intermediary storage component and the desktop based component as shown in Figure 1.

The web-based component fetches the OSM maps and provides them to the desktop component for display. It consists of PHP and Javascript components such that the internal processing, which includes but is not limited to, storing the longitude and latitude information in XML files, is provided by PHP while the interfacing with desktop and map rendering functions are handled by Javascript. The AJAX XMLHttpRequest object is used to send data from Javascript (client side) to PHP (server side). In the simplest case, the 'data' passed is essentially the longitude and latitude information which is the most basic information needed for navigation.

The desktop component provides a platform for interacting with the user. In this case, the user can provide a path for the AV by clicking various locations on the map. Which is read into the desktop component via an intermediary storage node and is then relayed to the AV via any available interface to the AV which can be wired interfaces like RS232, USB, etc or wireless interfaces like Bluetooth, Zigbee, 802.11, RF, etc.

The intermediary storage component can be any storage mechanism like a database system such as MySQL/MSSQL, or an XML file or even a plain text file. In this paper an XML file has been used for the sake of simplicity and speed.

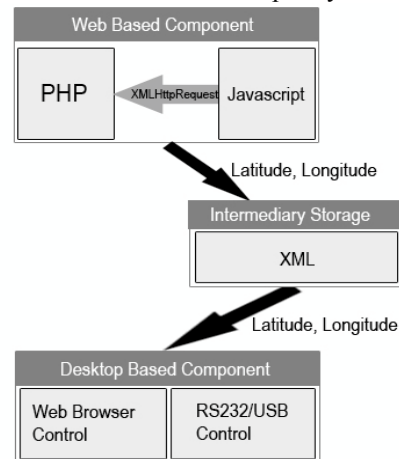


Fig. 1. System model for the simplest case

V. PRINCIPLE

At startup, the system displays the OSM map centered on a predefined location. The current position of the AV, taken from the AV's GPS receiver, is pinned on the map using a marker. When the user clicks on another position, the coordinates (latitude and longitude) of that point are passed to the server-side PHP script using XMLHttpRequest and the PHP script writes these coordinates in an XML file along with any other attributes that we may need. The desktop component then retrieves this XML file and reads it and fetches the longitude and latitude, etc. These longitude and latitude are then relayed to the Autonomous Vehicle which takes this location as its destination and navigates there. This mechanism is illustrated in Fig. 2.

VI. IMPLEMENTATION

The aforementioned system was implemented on a Windows based computer while keeping the web-based components on a local server machine (Apache server) and PHP. The desktop component was created using Microsoft Visual C# and it consists of a web browser, a serial port control and XML reading capabilities.

The web-based component consists of a Javascript source file for creating and displaying the map. Such that, the JS script OpenLayers.js, hosted on the OSM site, must be imported as the 'src' attribute of a script tag. Secondly, a DIV tag must be created, the ID of which shall be passed to the OpenLayers.Map function for creating the map. The map must be centered on a predefined longitude, latitude and zoom by using the SetCenter function. There might be certain transformations involved, too. For instance, the coordinates might be in World Geodetic System (WGS) 84 format. The OpenLayers library provides the "LonLat.transform" function for this purpose.

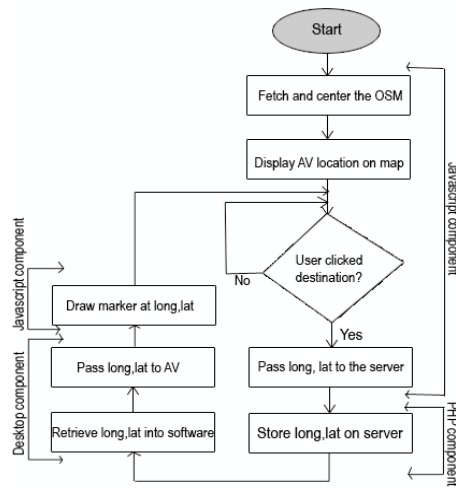


Fig. 2. The system's flowchart

For this specific application, we need to create a listener to detect user events like “click”, such that, when the user clicks anywhere on the map, the longitude and latitude of that point need to be fetched and passed to the server which, in turn, passes it to the desktop component. The initialization process is shown in Fig 3.

The server-part, scripted in PHP, reads longitude and latitude values via the GET method and then stores them in an XML file on the server, while providing the JavaScript part with a valid response i.e. the status of the operation which can be success or error. The Javascript part, depending on the status received from the server part, plots a marker on the map at the point where the user clicked, or displays an error message.

The desktop component's web browser control displays the OSM map since it can only be rendered on web-based interfaces. Map control functionality like panning, zooming and moving the map is prebuilt in OSM maps, all that is needed to enable it is, to call the function, `map.addControl` where `map` is the variable holding the OSM Map. The map also has the event listener functionality from the web-based component and when the user clicks on a specific location, all that is needed to be done at the desktop component is to read an XML file which can be retrieved from the server using normal HTTP downloading methods or even using file explorer techniques if the server is hosted on the local machine.

- 1 Import OpenLayers.js
 - 2 Create a map object [`OpenLayers.Map(div)`]
 - 3 Define center point and zoom / Transformations
 - 4 Set map center [`Map.SetCenter(Longlat, zoom)`]
 - 5* Listen for user interaction events e.g. Click
 - 6* Pass long, lat to server via `XMLHttpRequest`
- * Optional or application specific

Fig. 3. The OSM map initialization process

Once the longitude and latitude information has been read, it needs to be passed on to the AV. This paper uses

the RS232 interface for connecting to the AV. The RS232 interface relies on serial data transfer and many wireless modules are available which work on this principle so we can plug additional communication channels in, later on. MS Visual C# provides the serial port control which can easily be deployed in any application and provides a serial data transmission and reception channel.

The system may also have a waypoint flush mechanism such that, a path is first created on the map and then flushed onto the AV for navigation. This mechanism is more efficient since it decreases the dependency on the link between the navigator and the AV. If, however, a reliable link exists between the AV and the navigation system, online control is a better option wherein, we command the AV on-the-go and monitor its current position at any time.

VII. SAMPLE IMPLEMENTATION

The system, in a sample implementation, is shown in Fig. 4. The present location of the AV is shown by the car icon and the system is ready to take click input from the user. The status of the serial port, XML file and server connectivity is shown in the status bar. The current location of the AV is given by a car icon (custom marker) and upon clicking anywhere, a new waypoint will be created and stored.

A sample intermediary XML file is shown in Fig. 5. As is visible, there are four waypoints such that `waypoint0` is the initial position of the AV. It should be noted that the longitude/latitude values are sample data taken just for experimentation. The waypoint names are created using a counter that increments with each click.



Fig. 4. Sample Project Output

```

<markers>
<marker lat="7.4495019944114" lng="44.23425575432" name="waypoint0" />
<marker lat="11.559742145644" lng="42.11599424556" name="waypoint1" />
<marker lat="9.0153023334206" lng="47.19726562545" name="waypoint2" />
<marker lat="10.401377554544" lng="43.30810546875" name="waypoint3" />
</markers>
    
```

Fig. 5. Sample XML file

VIII. FUTURE WORK

The idea presented in this paper has a lot of scope for improvement and further research. The paths made from waypoints can be further corrected based on terrain information such that the surface types like roads, water, forestry etc can be taken into account for more accurate path planning. As OSM improves its mapping library and the exact AV parameters become known, a fully fledged guidance and control system can be made from this proposal

with a very user friendly interface and more powerful communication links with a very low actual cost compared to contemporary AVs.

IX. CONCLUSIONS

This paper presented a low-cost, simple and easy to implement Autonomous Vehicle guidance system using the freeware OpenStreetMap (OSM) map collection. An advanced integration technique was proposed for integrating OSM maps in desktop applications which involved a server component and a browser component for accessing the server component via HTTP requests. AJAX was used for writing a communal XML file on the server which can be accessed by the desktop front-end and hence the destination coordinates be read which are in turn transferred to the AV for navigation. It was supposed that the AV has an inbuilt GPS receiver, sensors and coordinate resolution algorithms for finding its paths from latitude, longitude waypoints and avoiding any obstacles in the way.

REFERENCES

- [1] Sert, B., Maddox, J., and Veatch, P., "Laser Assisted Intelligent Guidance For Automated Guided Vehicles", Intelligent Vehicles '93 Symposium, pp. 201-206, Jul 1993.
- [2] Bush, F.N. and Esposito, J.M., "Vision-based lane detection for an autonomous ground vehicle: A comparative field test", 42nd Southeastern Symposium on System Theory (SSST), pp. 35-39, Mar 2010.
- [3] Tong-Jin Park and Chang-Soo Han, "A path generation algorithm of autonomous robot vehicle through scanning of a sensor platform", IEEE International Conference on Robotics and Automation, Vol. 1, pp. 151-156, 2001.
- [4] Fanti, M.P. and Turchiano, B., "Deadlock avoidance in automated guided vehicle systems", International Conference on Advanced Intelligent Mechatronics, Vol. 2, pp. 1017-1022, Jul 2001.
- [5] Haklay, M. and Weber, P., "OpenStreetMap: User-Generated Street Maps", Pervasive Computing, IEEE, Vol. 7, Issue: 4, pp. 12-18, 2008.
- [6] Oliver O'Brien, "Creating and Maintaining Street Orienteering Maps using OpenStreetMap", Department of Geography, University College London, 2009. Available: http://oliverobrien.co.uk/files/2009/04/streeto_extabr3_gisruk09.pdf
- [7] Ricky Jacob, Jianghua Zheng, Blazej Ciepluch, Peter Mooney and Adam C. Winstanley, "Campus Guidance System for International Conferences Based on OpenStreetMap", 9th International Symposium on Web and Wireless Geographical Information Systems, LNCS, Vol. 5886, pp. 187-198, 2009.
- [8] Michael J. Casey, "Citezen Mapping And Carting: How Crowdsourcing is Helping to Revolutionize Mapping & Charting", Hydro 2009 Conference, Norfolk, Va., 2009.
- [9] Goolge Maps/Earth Terms of Services. Available: http://www.google.com/help/terms_maps.html.
- [10] OpenLayers Javascript Mapping Library Developer Docs. Available: <http://www.dev.openlayers.org/docs/>