AUV Planning, An application for AUV mission planning

António Coelho Ricardo Gonçalves Rui Rodrigues
FEUP/DEI/INESCTEC
Rua Dr. Roberto Frias s/n 4200-465, Porto, Portugal
{acoelho@fe.up.pt, refg@inescporto.pt, rui.rodrigues@fe.up.pt}

Resumo

Veículos autónomos submarinos (Autonomous underwater vehicles - AUV), são veículos controlados por computador que operam de forma autónoma em ambientes submarinos. A inacessibilidade que resulta de operar a grandes profundidades complica o planeamento e controlo devido à dificuldade em monitorizar os veículos em ambientes subaquáticos. Neste artigo é apresentada uma aplicação 3D que simula a execução de missões por AUVs em ambientes subaquáticos, permitindo tanto o planeamento como a visualização de missões. A possibilidade de ter uma referência visual dos AUV's enquanto desempenham as suas tarefas pode assim ajudar a evitar problemas e a diminuir o risco de fracasso das missões.

Abstract

Autonomous underwater vehicles (AUV) are computer-controlled vehicles that operate autonomously in underwater environments. The inaccessibility that results of operating at such depths further complicates the mission planning and control because of the difficulty to observe end monitor the vehicles in underwater environments. This paper presents a 3D application that simulates mission execution by AUVs in underwater environments, allowing both the planning and visualization of missions. This possibility of having a visual reference of the AUV while these are performing their tasks may help avoid problems and diminish the risk of mission failure.

Keywords

AUV, Mission Planning, Virtual, 3D, Environment, Unity3D

1. INTRODUCTION

The ocean's floor is full of resources like oil, gas and minerals. Nevertheless, deep sea exploration is dangerous and despite of the great advances made in this area, most of this territory remains unexplored. One possible way to lower the costs and raise the chances of mission success is through the use of autonomous underwater vehicles (AUV). However, AUVs pose their own challenges, such as the inaccessibility to the vehicle [Kuroda96] and the fully autonomous and unattended navigation.

In this paper we present a software application created to help planing and visualizing missions in a 3D environment for the Mares and Trimares AUVs [CN08].

Running the missions virtually as a way of testing them may help reduce potential failures in at-sea trials [Song03, Brutzman95] by anticipating or identifying problems that could otherwise pass unnoticed until the actual live mission. This application is able to receive any amount of tasks and assign them to AUVs to perform, concurrently or sequentially, allowing a higher degree of mission customization. Finally, by providing a visual representation of missions, people who are not familiar with the subject will be able to better grasp what missions entail, how AUV's perform and the benefits of this approach. Giving them a better understanding of the whole process, this application is being used both for marketing and educational purposes.

2. RELATED WORK

Computer simulation and planning of real life objects (or potential objects) has always been one of the best ways to evaluate scenarios and anticipate problems and flaws, before trying them out on the field. This approach not only allows the decreasing of resource costs, but also provides a platform for safer and faster testing. This is why simulation is present in most fields of engineering. Robotics is one of such fields, related with what we are presenting in this paper and therefore we will look at some simulation applications that currently exist.

The Webots [Michel04] is a commercial development tool that allows simulating mobile robots with custom attributes and shapes in a ucommon environment, as well as the use of predefined robots such as Sony Aibo[1]. These robots can

[1] http://www.sonyaibo.net/

also be programmed through the use of a C/C++ API that uses TCP/IP to communicate with the simulator. Physics simulation is handled by ODE[2] (Open Dynamics Engine) library.

USARSim [Carpin07] is an open source and free to use multi-robot and environment simulator created with Unreal Engine to model arbitrary application scenarios. This simulator allows the customization of robots through its capabilities of modeling and adding sensors. It also uses Mobility Open Architecture Simulation and Tools framework (MOAST)[3] which allows the use of modular modeling and design to the robots, making every module semi-independent and reprogrammable.

Another open source and free simulator is the Player Project[4]. This simulator was designed for research of robotic and sensor systems. Because of its extension capabilities, it allows simulation of both 2D and 3D environments, through the use of Stage and Gazebo backends respectively, and the use of third party plugins that can customize and add more features as third party utilities.

Unity3D[5] is a full featured engine which can be used to create high quality 3D interactive real time applications. It combines authoring tools for creating content, as well as flexible development facilities for implementing complex behaviors, in an integrated manner. Another strong point of Unity3D is its capability to deploy to several systems such as Windows, Mac, Android, IOS and also as a browser application.

Because of all the positive aspects of Unity3D and also due to the fact that the research group already has experience working with this tool and therefore having a considerably lower learning curve in comparison to the other solutions, the AUV Planner was developed using this game engine.

3. AUV MISSION PLANNER

The AUV mission planner application aims at producing a 3D simulation of a mission, described as a set of tasks assigned to one or more AUV's.

As shown in figure 1, the task manager starts by receiving a XML file (at this point of development, this file is still being created by hand), containing all tasks to be executed, which is parsed by the task manager. The task manager is an object within the virtual world that simulates a console. At this moment the user cannot yet interact with it, but in the future it will allow the user to change simulation parameters of the active mission.

The XML File is divided in three parts, the Console tasks, the tasks to be assigned to the AUVs and finally, camera information. The first part is a list of tasks for the console to execute, and this happens as soon as the file has been parsed. These tasks may include the deployment of AUVs, or the management of the vertical AUV transport rail.

[2]http://www.ode.org
[3]http://moast.sourceforge.net
[4]http://playerstage.sourceforge.net
[5]http://unity3d.com/

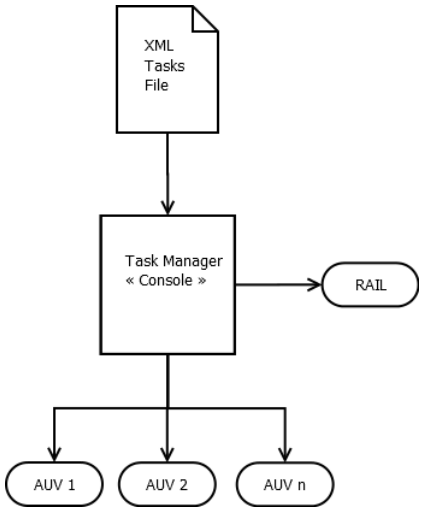

Figure 1. Mission Process

```
<TaskManager>
- <camera>
    <task type="LookAT" position="12.8;-8.0;2.9" direction="11.9;-7.6;2.7"
      id="0" />
    <task type="LookAUV" targetname="SURVEYOR1" position="8;27;2"
      id="1" />
  </camera>
- <console>
    <task type="LowerRail" changecamid="2" raildisp="20" />
    <task type="OpenRail" />
    <task type="DeployAUV" targetname="SURVEYOR1" />
  </console>
- <auv name="SURVEYOR1">
    <task type="Wait" duration="5" />
  - <task type="FollowWP" description="Preparing to dive">
      <waypoint type="followmode" position="15.5;-15.5;-1"
        velocity="0.5;0.4;2" />
      <waypoint type="divemode" position="15.5;-2995;-1"
        velocity="2;0.1;0" description="Diving to damaged pipeline" />
    </task>
    <task type="Search" targetname="damagedpipe"
      description="Searching for pipeline" />
  </auv>
</TaskManager>
```

Figure 2. XML file example

The AUV task part is divided in sections, one for each AUV. Each section contains the list of tasks to be associated to the corresponding AUV by the console. These tasks can be set in five modes:

1. Wait Mode - The AUV waits for a defined number of seconds until it goes to the next task;
2. Follow waypoints Mode - This task defines one or more waypoints to be followed by the AUV one by one until all waypoints have been visited;
3. Search Mode - The AUV begins a outwards spiral search, with its scanner active, searching for a matching object with the one defined in the task;
4. Pause tasks mode - The AUV enters an indefinite wait state until it receives a signal from another AUV or from the console to resume its tasks;
5. Resume tasks mode - Sends a resume task signal to the console or to an AUV.

3.1. Waypoint following

Each task can also change a camera mode and define a description for the current task. The waypoint following mode is possibly one of the most used and important task mode because it is through this mode that the AUV knows where to move and how to move. Each waypoint has two main fields of information that are always required: the *target coordinate*s to where the AUV must move and a *velocity* field that is composed by *top speed*, *slowdown speed* and *minimum distance* to the target coordinates to start slowing down. Besides these mandatory fields, the waypoint can also define a *jump to* field with coordinates to instantly move the AUV to that place, useful for skipping long dives, and also a *heading* field to be used in pair with the *jump to* so that when the AUV is moved, it will be facing the defined direction. As with tasks, waypoints can also change the camera mode and define or redefine the current task description. Camera functionality will be addressed on the next section.

The AUV navigation from waypoint to waypoint starts by slowly rotating towards the destination waypoint and then shifting its position vector, respecting the modeled AUVs top speed of 2m/s, towards its forward vector.

3.2. Camera control

Being able to add camera control information to tasks and waypoints is useful to create demos of the mission with a cinematic feel to it. Cameras can be set in one of three modes that focus on an AUV:

1. Fixed position mode, rotating to follow an AUV
2. AUV chase mode, moving behind an AUV at a defined distance
3. Onboard camera mode, displaying the area in front of the AUV

These camera modes can be triggered to change when an AUV starts a new task, but can only be activated if the camera is set on cinematic mode. However, if the camera is set in free mode, the user can jump between AUVs by pressing a keyboard key, or roam free through the scene.

4. SCENE ELEMENTS

Another of our aims is to show the simulation as close to reality as possible in order to create a compelling and immersive environment. That is, the creation of an underwater look. At first we started looking at real underwater videos in order to identify the main elements in such a scene. Some of these elements were: poor visibility at medium range; light reduction with increasing depth; water particles that flow with underwater currents; water transparency and water disturbance from the propellers.

The next step was mixing all these elements and tweaking them until we got the desired underwater effect. Another aspect of the underwater effect was that the simulation could start above water and therefore, the scene would have to work accordingly. To solve this problem we created two effects, one for above the surface of the water and another for bellow. When the camera is above the ocean surface, the scene is composed by a skybox, the sun, and of course the ocean surface. On the other hand, when the camera moves bellow the ocean surface, the skybox is disabled and uses the camera default background color instead. To address the poor visibility a fog effect was added to occlude the vision. In order to create a seamless color effect in the scene the fog color and the camera's default background color are always the same. As the depth increases this color (cyan blue) darkens, however instead of going totally black as is expected in a real world deep underwater area, one of the goals of this application would be compromised because the whole scene would be darkened. Additionally the sun flares are also disabled under the water and the light source intensity is also decreased as the depth increases, due to the same reason stated above, the default scene ambient light is set to a dark grey that still allows a good visibility at very high depths.

To make the scene look more dynamic, and also enhance the underwater effect, a particle generator was added to the camera that simulates water particles being affected by currents. This particle generator produces particles with random energy in a cubic volume with a side length of 15 meters. Additionally to each particle a random velocity is added to create the illusion of different currents and movement.

Finally to add a sense of movement to the AUV a trail was added to its rear. This trail is a Unity3D specific component called Trail Renderer. Initially, the trail seemed to be dragged by the AUV. To solve this issue a script was added to the trail in order to animate the texture by shifting the texture UVs in the opposite direction of the AUV movement.

For the moment, the AUV Planner is using a fixed 3D scene. However we have plans to create two procedural approaches to scene creation. One and the simplest of the approaches is the use of a XML file with scene specifications, such as the start and end coordinates of a pipeline and with those values the scene generator would add as many pipes as needed. The other approach involves an algorithm that is fed with mission parameters and generates a random scene.

5. CASE STUDY

An initial version of this framework was tested with a mission created for a commercial demonstration of the capabilities of a specific set of AUVs of two types, Mares and Trimares. Because the actual mission planning and supervision mode are still being developed, these were not part of this test.

The created scene takes place in some part of an ocean where at the center of the scene and on the surface there is an offshore platform that has a vertical rail that can transport two Mares AUVs at once from a control room at the top of the platform, down to 15 meters below the surface of the water, as well as bring them back up.

Deep into the ocean, 3000 meters below the surface, a net-

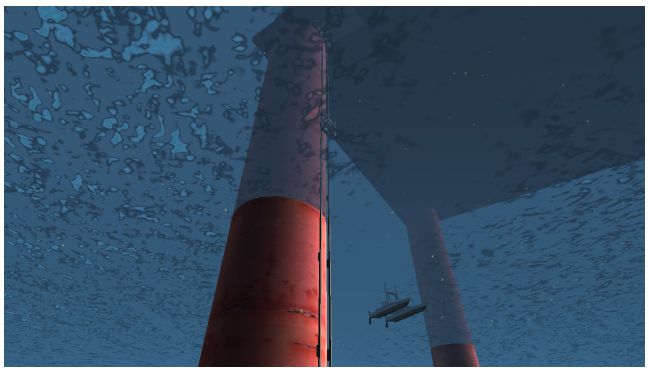

Figure 3. Platform and Rail

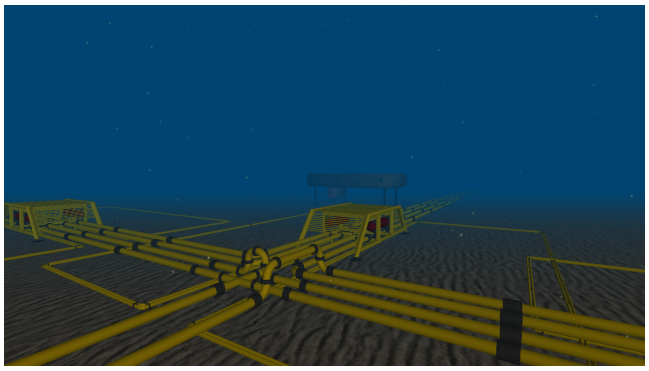

Figure 4. Pipeline network and submarine base

work of pipelines exists, and one of the pipes has suffered damage. Also, near the pipelines there is a small submarine base with a deployment area that can accommodate two Trimares AUVs. The test mission had as objectives the observation and analysis of the damaged area in the pipeline.

The test mission begins with the lowering of the rail, carrying a Mares AUV. When the AUV is finally released at 15 meters below the surface, it begins its descent path until it reaches the depth of 2990 meters. Once there, it begins a spiral search pattern, searching for the pipeline. As soon as it finds the correct pipeline it follows it, with its scanner activated until he finds his target. Once the target has been found it does some preliminary scanning and signals the console it has finished performing the scan. At this point, a Trimares AUV is deployed from the submarine base for a better analysis. Once all tasks are finished, both AUVs return to their respective bases.

6. CONCLUSIONS AND FUTURE WORK

Even though the AUV Planner is still under development, we have been able to visualize some missions for marketing purposes and the results were very satisfactory. The visual impact was tested at the trade fair "Fórum do Mar"[1] and the overall impression was extremely positive. We feel that the environment, especially in 3D, can provide a feeling of deep water exploration. On the other hand, the AUVs were able to perform the mission with success and the XML-based mission process gives us a strong basis from which to work and evolve this system.

Nevertheless there is still much room for improvement and possibly branch into an educational game. The application is being upgraded to be used as a serious game for learning activities in introductory programming courses. In this new application each AUV will be controlled by agents through a client/server architecture and be able to communicate and cooperate with each other. Furthermore, this agent control plugin will also allow the use of the AI operating in the real AUV's.

Possible features for future development include an overhead map with details of all the objects in the scene and the mission, missions that allow the creation of multiple options for specific scenarios that can be chosen in real time through the console, also a mission designer for an easy way of creating the missions and finally transform AUVs into real AI agents, each with its own control script being run remotely in a client/server architecture.

Furthermore, the AUVs are being developed with dynamic models to accurately simulate their behavior. This simulation tools will be integrated in the application to provide accurate simulations of the AUV behavior under specific underwater conditions.

[1] http://www.forumdomar.exponor.pt/

7. ACKNOWLEDGEMENTS

This work is partially supported by the Portuguese government, through the National Foundation for Science and Technology - FCT (Fundação para a Ciência e a Tecnologia) and the European Union (COMPETE, QREN and FEDER) through the project PTDC/EIA-EIA/108982/2008 entitled "3DWikiU - 3D Wiki for Urban Environments".

8. REFERENCES

[Brutzman95] Don Brutzman. Virtual world visualization for an autonomous underwater vehicle. In *Proceedings of the IEEE Oceanic Engineering Society Conference OCEANS 95*, pages 1592–1600, 1995.

[Carpin07] S. Carpin, M. Lewis, Jijun Wang, S. Balakirsky, and C. Scrapper. USARSim: a robot simulator for research and education. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1400–1405, 2007.

[CN08] Matos Aníbal Cruz Nuno. The mares auv, a modular autonomous robot for environment sampling. In *Proceedings of the MTS-IEEE Conference Oceans'2008*, pages 1 –6, 2008.

[Kuroda96] Yoji Kuroda, Koji Aramaki, and Tamaki Ura. Auv test using real/virtual synthetic world. pages 365–372, 1996.

[Michel04] Olivier Michel. WebotsTM: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems, Volume 1, Number 1*, pages 39–42, December 2004.

[Song03] Feijun Song, P.E. An, and A. Folleco. Modeling and simulation of autonomous underwater vehicles: design and implementation. *Oceanic Engineering, IEEE Journal of*, 28(2):283 – 296, april 2003.