Nessie III Autonomous Underwater Vehicle for SAUC-E 2008

J. Cartwright, N. Johnson, B. Davis, Z. Qiang, T. L. Bravo, A. Enoch, G. Lemaitre, H. Roth, Y. Petillot

Ocean Systems Laboratory, Heriot-Watt University, Scotland, UK

The Nessie III autonomous underwater vehicle was been designed and built in the Ocean Systems Laboratory at Heriot-Watt University to compete in the 2008 Student Autonomous Underwater Challenge – Europe competition. Employing a robust, modular hardware and software design the vehicle was able to successfully complete all of the tasks set out in the competition,, and will serve as an excellent platform for further development.

***Index Terms*— machine vision, mobile robots, underwater vehicles**

I. Introduction

This paper describes the Nessie III autonomous underwater vehicle (AUV) developed at the Ocean Systems Laboratory at Heriot-Watt University. The vehicle was designed to compete in the 2008 Student Autonomous Underwater Challenge Europe (SAUC-E) competition held at the IFREMER deep wave basin near Brest, France. Section II describes the vehicle hardware, and section III covers the software architecture.

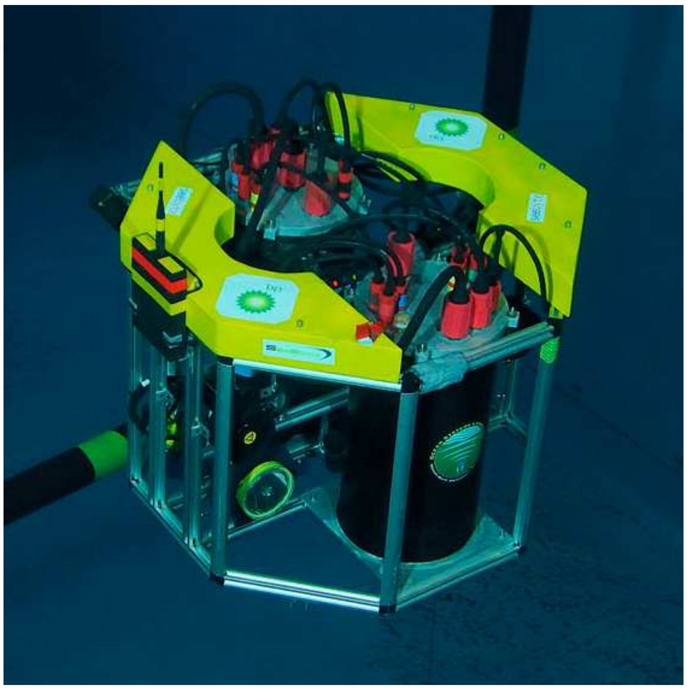

Figure 1: Nessie III AUV at SAUC-E 2008 (image courtesy of Yves Gladu)

II. Hardware

The core of the vehicle is comprised of two 22cm diameter cylindrical aluminium hulls surrounded by a metal frame. This frame serves as a mounting point for sensors, protects the contained devices from impact, and keeps the thrusters safely out of the way of human divers. One hull, dubbed the motor hull, houses batteries and H-bridge controllers to drive the thrusters. The other, the PC hull, contains the embedded computers, interfacing and sensing electronics and the batteries to power these. Separating the power supplies and electronics in this way provides a degree of isolation from noise and supply voltage fluctuation caused by the H-bridges. It also ensures that even if the thrusters drain their batteries to a low level, the computers remain operational and control is still possible. A photograph of the submerged vehicle can be seen in Figure 1.

A schematic of the hardware architecture is given in Figure 2. The various hardware components of the vehicle will now be described in more detail.

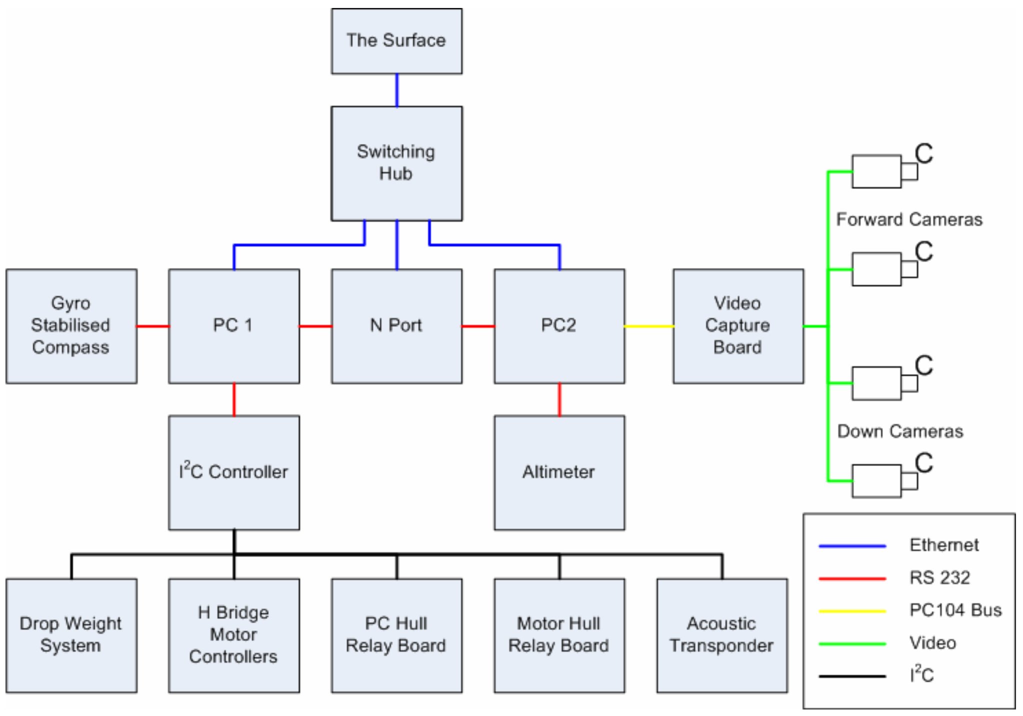

Figure 2: Schematic of the vehicle hardware.

A. PC104 embedded computer

The computers used in the vehicle are industrial MSM800 PC104 embedded PCs. This model was chosen both for its relatively small size and its low power consumption which results in longer battery life and less heat. It has an AMD Geode LX800 500 MHz Processor, 512 MB of RAM, 4 USB ports, 2 serial ports and one 100 Mbit Ethernet port. Two of these embedded PCs are used in the vehicle; one for low level sensing and control, and another for video capture and processing. These are connected via Ethernet. This split ensures that that primary control PC is never starved of resources by the image processing algorithms.

To make the computers more robust, flash based solid-state hard disks are used instead of standard magnetic disks. These further reduce the power requirements of the system and make it more robust to bumps and jerks as are expected in a mobile vehicle. The internals of the assembled PC hull are shown in Figure 3.

In case of boot faults, an Ethernet-accessible dual serial port is connected to the serial console of each PC104, which allows the boot progress to be monitored seconds after the vehicle is powered on. This enables diagnosis and repair of operating system faults over an Ethernet link (wired or wireless), without opening the hull.

B. Power supply

Each of the hulls contains twenty D-type Nickel-Metal-Hydride (NiMh) cells connected in series to provide a nominal 24 volts. The capacity of the cells is 9 amp hours, which allows the vehicle to run continuously for several hours. Each hull is fitted with a charging connector that enables the batteries to be charged in place using external battery chargers. The packs are fused to protect against dangerous current levels.

During testing, a 100 metre power tether may be employed for constant running. The connector used on this tether is wet mateable, with the tether power off. The tether is driven at 30 volts at the top end, using a 20 amp bench supply. Fast action relays are used within the hulls to allow the vehicle to hot-swap between tether and battery power with no PC downtime, which proved to be an incredibly useful facility.

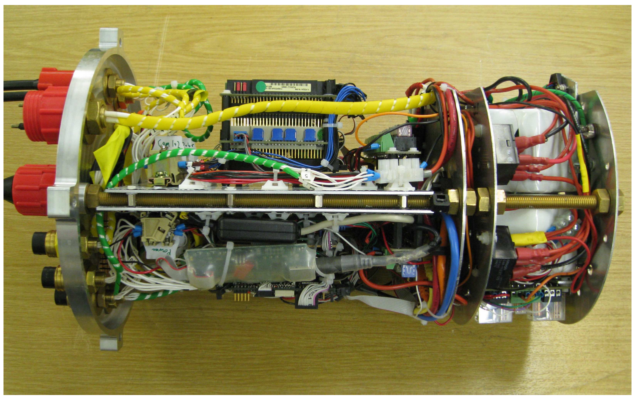
Figure 3: Assembled PC hull internals.

C. I^2C Interface

The primary PC104 employs a USB I^2C interface (Total Phase Aardvark), in order to control multiple hardware devices. These include the H-bridges, the bespoke relay boards, drop weight actuator, acoustic transponders, and battery temperature sensors.

D. Hull Connectors

Wet mateable Subconn connectors are used for all external connections to the hulls. These provide assurance that even if a connector is incorrectly attached or dislodged when submerged, water will not immediately penetrate the hull.

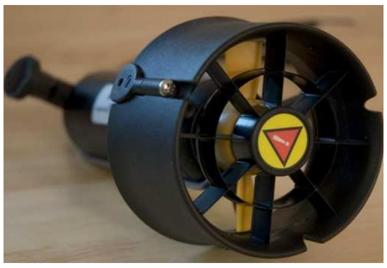
Figure 4: SeaBotix BTD150 thruster.

E. Thrusters and H-Bridges

The vehicle has been equipped with five SeaBotix BTD150 thrusters (Figure 4), which use 80 watts of power at maximum speed to produce 2.2 kilogram-force of thrust. The five thrusters are arranged thus; port, starboard, lateral, and two vertical. This configuration provides 4 degrees of freedom in control: surge, sway, heave and yaw. Two linked vertical thrusters are used for additional power in descent/ascent. The thrusters are driven by four H-Bridges controlled using the I^2C interface.

F. Submersible switch and visual feedback device

A bespoke submersible module combining reed switches and LEDs was constructed for basic system control and visual feedback. One reed switch is activated by a magnetic key placed in a recess, serving as the PC on switch. Another can operate as a generic start/stop signal, e.g. for mission start. Three high brightness LEDs can be controlled via the I^2C interface, giving operator feedback.

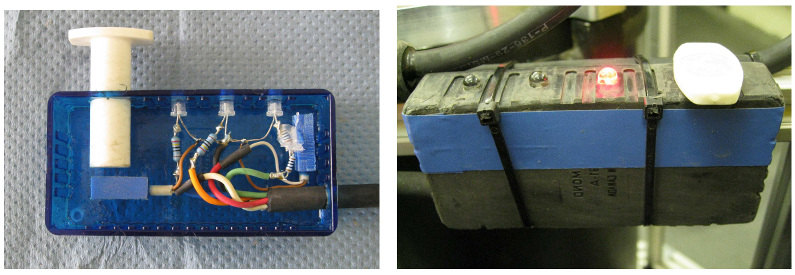

Figure 5: Switch module before and after moulding

The end plate of the motor hull also has a reed switch behind it, which activates power to the motor hull via a relay and serves as a ‘kill switch’, making the vehicle safe for handling by divers.

G. Altimeter

A Tritech PA500 is used to measure the distance between the vehicle and the floor of the tank. This has a maximum operating range of 50 metres, well over the depth of the competition tank. It uses an RS232 connection to the PC104 and is powered by a 12V supply.

H. Inertial Measurement Unit (IMU)

The Xsens MTi is a miniature, gyro-enhanced Attitude and Heading Reference System (AHRS). It is used to measure the vehicle’s 3D orientation and its angular velocity by using a combination of gyroscopes, magnetic field sensors and accelerometers. The INS weighs 50g and is supplied by 5V.

I. Doppler Velocity Log (DVL)

The vehicle is fitted with an RD Instruments Explorer DVL, which can provide very accurate information regarding velocity of the vehicle over the ground. The DVL was disabled for the SAUC-E competition as it was not permitted by the rules, but it will be used in future research applications.

J. Cameras

The vehicle is fitted with four 50 metre rated underwater colour cameras, with optics specifically designed for minimal distortion underwater. They interface to the second PC104 with a PC104 form-factor video capture board. This can capture up to 25 frames per second, at 24 bit colour depth.

The cameras are arranged on mounts in pairs, for potential stereoscopic vision in both the forward and downwards directions. For the competition where the floor of the tank was flat, only a single downward facing camera was used. However, in future research both downwards facing cameras will be used to provide 3D reconstruction of surfaces underneath the vehicle.

K. Wireless/wired Communication

A D-Link 802.11G Wireless Access Point is used for wireless communication with the vehicle when on the surface. This was stripped of its case and unnecessary connectors and moulded with a high gain aerial in rubber compound to make it submersible. The access point is connected to an Ethernet hub and 5V power source in the PC hull via a wet mateable connector. This same hull connector may also be used to attach an Ethernet tether instead of the wireless access point, if communication is desired during submerged testing.

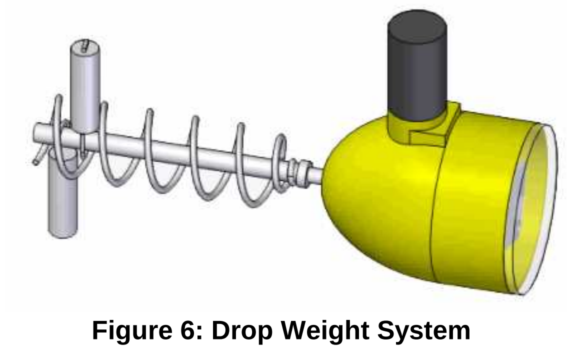

Figure 6: Drop Weight System

L. Drop Weight System

The drop weight system utilises a custom made helical actuator built around a recycled thruster housing. A simulation of the system is shown in Figure 6. The unique design allows both markers and buoyancy to be suspended from a central bar and expelled from the system by rotating the actuator. A stepper motor connected to the drive shaft provides precise control over the rotation.

The stepper motor is driven by a custom microcontroller board which takes care of low level control of the system and communicates with the vehicle's high level control systems through the I^2C bus. Settings held in EEPROM memory on the controller allow for variable speed, soft start and variable torque.

M. Acoustic Positioning Transponders

In order to aid in the navigation of the vehicle, experimental active transponders were developed. These can be deployed at known locations and will then use hydrophones to communicate with the vehicle. When a specific signal is sent from the vehicle a transponder will reply, and the delay in the vehicle receiving this reply gives the two way time of flight for the signal. From this a simple speed/distance/time calculation gives the distance to the transponder.

The transponders use a single hydrophone as both a receiver and transmitter, and operate by sending chirps in the frequency range 60kHz-90kHz. An analogue band pass filter amplifies the received signal before it is sampled and processed by a dsPIC microcontroller running at 29.4 MIPS. This microcontroller implements a real time matched filter in order to detect the desired signals, and also generates the chirps which act as a reply. A DC-DC step up converter is used to produce a higher voltage when transmitting a signal, in order to maximise range.

On board the vehicle an almost identical transponder interfaces with the vehicle's control systems using the I^2C bus. Update rates of one reading per second are used from each deployed transponder, although faster rates are easily possible if required. The system can be used with two deployed transponders to give precise positioning when combined with the reading from the vehicle's altimeter. The system could be expanded in future to minimise errors by utilising more than two transponders.

The transponder system was not ultimately used in the SAUC-E competition runs, but will play a part in future research with the vehicle.

III. SOFTWARE

A. Architecture

The software architecture for Nessie III is illustrated in figure Figure 7. All modules are implemented as separate processes, and communicate using the OceanSHELL system developed in the Ocean Systems Laboratory. This is a lightweight UDP based communications protocol for distributed, modular systems. Each module process is monitored by a 'watcher' process, which will restart the module if it fails. Adjustable parameters for the modules are stored in configuration files, to simplify modification.

A version of the Ubuntu Server Linux distribution is used on the embedded PCs. The software modules used in the vehicle will be described in more detail in the following sections.

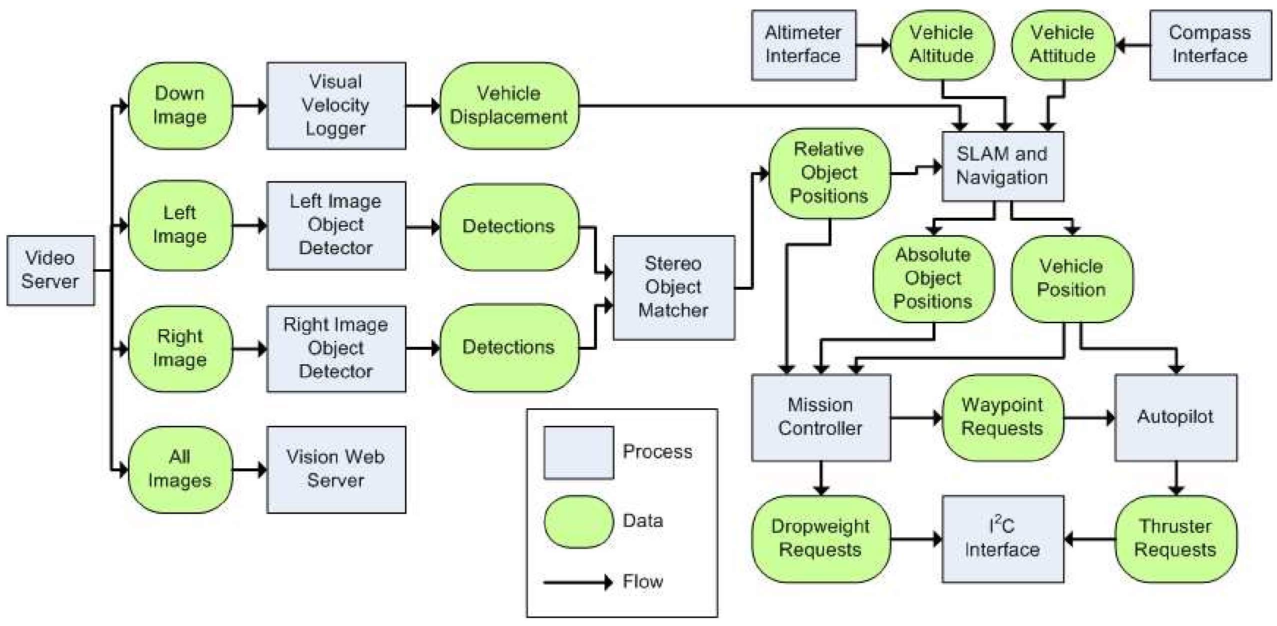

Figure 7: Software architecture

B. *Mission Control*

The mission control system (see Figure 8) consists of several layers of abstraction, working up from a vehicle systems interface to a parameterised behaviour based finite state machine system.

At the highest level, a finite state machine based controller oversees the mission. Each state in this FSM represents a particular behaviour with a particular set of parameters. Exactly one behaviour has control of the vehicle at any one time, and the controller monitors its condition, switching to different states as required by the mission specification.

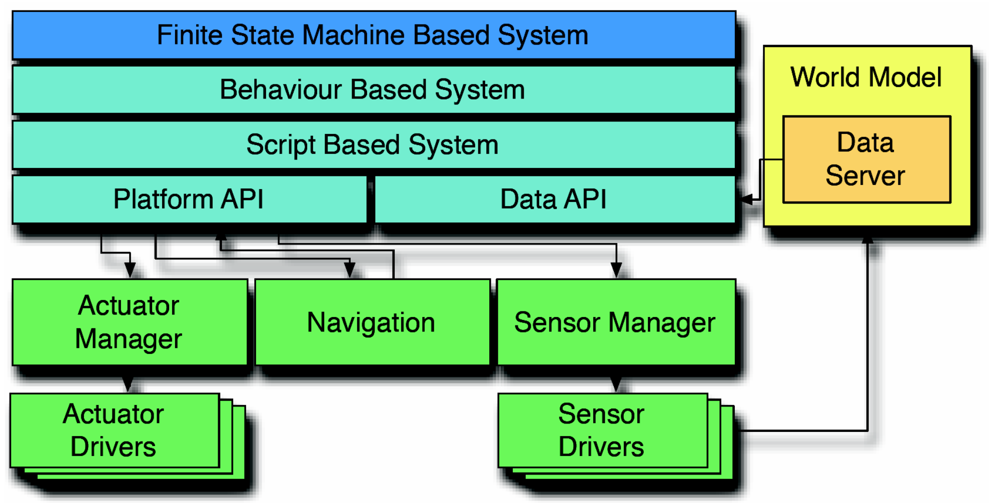

Figure 8: Mission control system

Generic behaviours are defined in a dynamic scripting language known as ‘Groovy’ [5]. These are able to pass instructions to the low level vehicle systems through a ‘Platform’ API. These instructions could be setting the mode of a sensor or actuator, or requesting movement to a relative or absolute position. This API also provides the script with basic information about the vehicle, such as its current position and attitude.

Information received from the sensor processing systems is entered into a ‘data server’, access to which is provided by the ‘Data’ API. This allows the script to both read and write to the data server, allowing access to information about the current position of objects in the world and also for different behaviours to pass information to each other. Times, scalar values, relative and absolute positions and areas may be stored in the data server. The parameters of a behaviour instruct it as to which elements it should retrieve from the data server in order to carry out its particular task. Example tasks which a behaviour may carry out are passing through the validation gate, searching the mission area and colliding with a mid water target.

For complex tasks in dynamic environments, the finite state machine based layer may be replaced with a full mission replanner. This system is fully implemented but was not deployed on the vehicle for the competition as the simpler system was sufficient.

C. Autopilot

As with most hover capable underwater vehicles, only the surge, sway, heave and yaw of the platform are controlled, leaving the vehicle's centre of mass and buoyancy to maintain the pitch and roll approximately zero. The autopilot is comprised of a bank of cascading Proportional, Integral, Derrivative (PID) controllers, and an axis force to thrust controller. It is the job of the autopilot to accept waypoints from the mission controller and manoeuvre the vehicle appropriately.

D. Visual Object Detection

One of the goals during the SAUC-E competition was to detect and classify different objects within the tank. The vehicle employs video cameras for this task, as visibility was very good and the colours of the objects provided good features for classification. The video capture board provides images with a resolution of 360 x 288 pixels. For image processing, the Intel Open Computer Vision Library 1.0 (OpenCV) is used.

The images from both the front and down cameras are passed through a collection of detection algorithms to identify all of the possible targets in the arena. The algorithms are designed to be as efficient as possible given the tight processing limitations of the platform. Distinguishing features of the targets were identified on an individual basis and incorporated into the detection algorithms, such as the yellow colour of the cones and the concentric circles present for the tyres. Transforms applied to the images include Canny Edge Detection, and Hough Circles and Hough Lines. Different colours spaces are also used as appropriate to best highlight differences between the targets

E. Stereoscopic Vision

By combining object detections from two images taken from differing viewpoints, it is possible to obtain estimates of relative 3D position. Firstly, an object is tracked in both images using the detection algorithms mentioned above, which give the centre of the object. Epipoloar geometry constraints are used to reject outliers, and application of the intrinsic matrixes for each camera converts from pixels to metric values. Finally, the triangulation method is used to calculate the estimated 3D positions.

F. Video Velocity Log (VVL)

Given the ban on Doppler Velocity Log (DVL) systems for the SAUC-E competition, a visual alternative was developed using optical flow estimation with the downward facing camera. This outputs estimated surge and sway velocities, and served as the primary input to the vehicle's navigation system in the competition.

1) Estimation of Optical Flow

The algorithm for estimating the optical flow between two frames of the video stream used for the VVL is the Lucas-Kandade iterative image registration technique. This method uses the spatial intensity gradient and a Newton-Raphson iteration to find matches between feature points. The function is computationally cheap and can deal with rotation and other distortions of the images. The main advantage of the method is that it assumes that the shift between two images is small. Therefore, it can be considered as approximately registered [2]. This assumption is certainly true for the available frame rate and the speed of the AUV. We are using the pyramidal implementation of the Lucas-Kanade feature tracker proposed in [3] from the OpenCV-library. Figure 2 shows the motion vectors of the optical flow between two frames estimated with Lucas-Kandade-method.

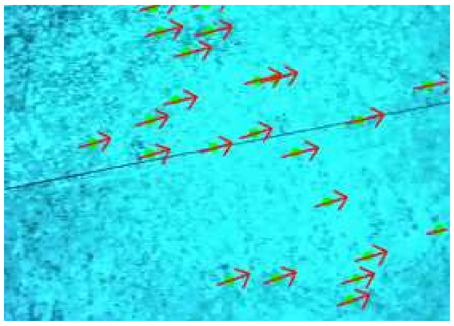

Figure 9: Optical flow estimation with Lucas-Kandade-method

2) Camera Motion Estimation

The optical flow between two frames is used to estimate the motion of the AUV. The algorithm has four stages:

1. Convert motion vectors from pixel coordinates to metrics
2. Remove rotational component from motion vectors
3. Compute translational velocities for each vector
4. Use the median velocity as final estimation

The conversion from pixel coordinates to metrics uses the intrinsic parameters of the camera estimated from the camera calibration. The rotational component of motion is computed based on the output of the IMU sensor, which is then subtracted from the metric motion vectors, in theory leaving just the translational components. The distance from the camera to the observed plane (the bottom of the tank) is given by the altimeter sensor, and this is used to calculate the translational velocities for each vector. A final motion vector estimate is taken to be the median of these translational velocities.

G. Navigation System

Given the available incoming data from the vehicle's sensors, the navigation system aims to produce the best possible estimate of the vehicle's current absolute position and pose, and the absolute location of detected targets within the world. A Kalman Filter is used for this purpose. The inputs to the navigation system are:

1. Yaw (heading) from the IMU
2. Altitude from the altimeter
3. VV L velocity estimates (if enabled)
4. DVL velocity estimates (if enabled)
5. Visual target detection positions

In the simple case, only the first four of these inputs are used to refine the Kalman Filter's estimate of the vehicle position. In this mode the target detection inputs are only used to refine the position estimates of the targets themselves. The navigation system has a more complex mode of operation where a form of Simultaneous Localisation and Mapping (SLAM) is used to update both vehicle and target positions based on the sensor observations. However, in practice this was not used in the competition as false target detections led to inaccuracies.

H. Augmented Reality Framework

The Augmented Reality Framework (ARF) allowed for testing of mission planning and autopilot control in advance of the real AUV hardware being ready. Using simulated video camera detections and a simulated altimeter and compass sensor combined with a hydrodynamic model, the autopilot was initially tested using simulation alone. Once the autopilot module was proved, it was possible to test mission planning in simulation, to evaluate how the vehicle behaved upon detection of the object types used in the SAUC-E competition.

A 3D virtual environment was constructed to mimic the conditions at the competition. This includes tank dimensions, vehicle dimensions, and representations of the objects and obstacles present in the SAUC-E arena. The environment is also used for monitoring the various systems on the AUV. This can be useful when doing real world tests as the output of the running systems can be visualised and any problems quickly highlighted. For example the navigation system output is visualised in the virtual environment allowing decreases in navigation accuracy to be seen immediately. Several images produced by the ARF are shown in Figure 10.

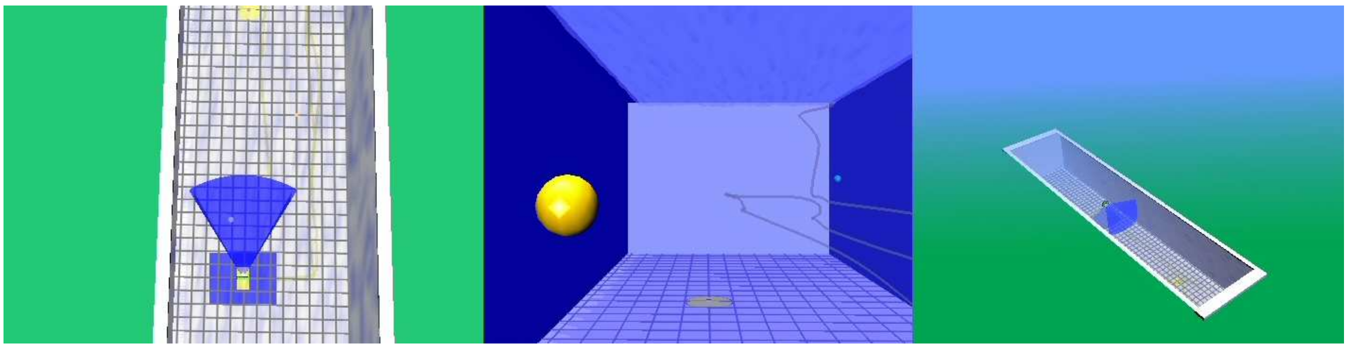
Figure 10: Example visualisations from ARF

IV. Conclusion

The Nessie III AUV successfully completed all of the tasks set out for SAUC-E 2008, earning the first place position and the Thales Special Award for Innovation in Decision Making Autonomy. The vehicle's modular design will allow it to grow as new technology is added, serving as a valuable new research platform for the Ocean Systems Laboratory. Since the competition, a WHOI acoustic modem has been integrated with the vehicle, and the addition of a Tritech Micron sonar is planned in the near future.

Acknowledgment

Team Nessie would like to thank its sponsors BP Limited and Seebyte Limited, all of the people in the Ocean Systems Laboratory, and Daniel Serrano, David Taddei, and José Vasquez of Seebyte Limited. The generous support of these groups was pivotal to the success of the project and the team is extremely grateful. We would also like to thank Edinburgh Council and the staff of the Edinburgh Royal Commonwealth Pool for allowing us to use their diving facility for testing.

References

[1] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, New Jersey, 1998.
[2] Bruce D. Lucas and Takeo Kanade, *An Iterative Image Registration Technique with an Application to Stereo Vision, Proceedings of Imaging Understanding Workshop*, pp. 121-130, 1981.
[3] Jean-Yves Bouguet. *Pyramidal implementation of the lucas kanade feature tracker - description of the algorithm*. Intel Corporation, Microprocessor Research Labs.
[4] Y. Ma, S. Soatto, J. Kosecka, S. Sastry. *An introduction to 3-D Vision*, Springer-Verlag New York Inc., Ch. 5, 2003.
[5] *Groovy Dynamic Scripting Language,* http:// groovy.codehaus.org