Visual Inspection of Hydroelectric Dams Using an Autonomous Underwater Vehicle

Pere Ridao, Marc Carreras, David Ribas, and Rafael Garcia
Computer Vision and Robotics Research Group, Institute of Informatics and Applications, Universitat de Girona, Campus de Montilivi, 17071 Girona, Spain
e-mail: pere@eia.udg.edu, marc.carreras@udg.edu, dribas@eia.udg.edu, rafa@eia.udg.edu

Received 15 January 2010; accepted 30 May 2010

This paper presents an automated solution to the visual inspection problem of hydroelectric dams. A small autonomous underwater vehicle, controllable in four degrees of freedom (surge, sway, heave, and yaw), is used for autonomously exploring the wall of a dam. The robot is easily programmed using a mission control language. Missions are executed by an intelligent control architecture that guides the robot to follow a predefined path surveying the wall. During the mission, the robot gathers onboard navigation data synchronized with optical imagery, sonar, and absolute navigation data, obtained from a moored buoy equipped with an ultra-short baseline system. After the mission, the images of the wall are used to build a photomosaic of the inspected area. First, image features are matched over the image sequence. Then, navigation data and interimage correspondences are optimized together using bundle adjustment techniques. Thus, a georeferenced globally aligned set of images is obtained. Finally, a blending algorithm is used to obtain smooth seam transitions among the different images that constitute the mosaic, compensating for light artifacts and improving the visual perception of the scene. © 2010 Wiley Periodicals, Inc.

1. INTRODUCTION

During the past two decades we have witnessed the application of autonomous underwater vehicles (AUVs) to traditional fields such as marine science (German, Yoerger, Jakuba, Shank, Langmuir, et al., 2008), wreck (Eustice, Singh, Leonard, Walter, & Ballard, 2005) and underwater (Mindell & Bingham, 2001) archaeology, and maritime security and energy sources (Hagen, Storkersen, & Vestgard, 1999). Recently, the maturity of the technology, the cost reduction of the underwater instrumentation, and the latest developments in underwater image processing have opened the door to new applications of scientific (Pizarro & Singh, 2003; Pizarro, Eustice, & Singh, 2004; Singh, Roman, Pizarro, Eustice, & Can, 2007; Williams & Mahon, 2004), industrial, and security (Negahdaripour & Firoozfam, 2006) interest. This article focuses on one of these applications: hydroelectric dam inspection. A preliminary meeting of our research team with civil engineers from a Spanish power generation company allowed us to identify the following tasks of interest:

- **Visual survey of the wall:** assess the state of the concrete for safety purposes.
- **Visual survey of the protecting fence of the water inlet to the penstock gallery:** assess the quantity of vegetal residuals obstructing the water flow and reducing the generated power (see Figure 1).

A multimedia file may be found in the online version of this article.

Currently, these inspections are performed through a careful visualization of a video recorded by a professional diver, often without any localization information. Sometimes a global positioning system (GPS) reading gathered at the surface is overlaid on the image, introducing an approximate location of the underwater camera. A diver-tracking system may also be used, but even in this case the system is not accurate enough to stitch all the images together to set up a global map of the surveyed area.

During the past several years, several companies have claimed to provide underwater robots for dam inspection (Seabotix, Inc., 2010; VideoRay LLC, 2010). Normally they propose the use of small-class, remotely operated vehicles (ROV), working as teleoperated cameras for video recording, to replace the professional diver who traditionally occupied this place. Few research precedents provide an added-value solution. One of the most relevant works is the ROV3 system developed by the researchers of the Institut de Recherche HydroQuébec (Cote & Lavallee, 1995). This is a small ROV, localized through a long base line (LBL) system, which makes use of a multibeam sonar for collision avoidance. The system is able to control the distance to the wall and includes several video cameras as well as a laser system for two-dimensional (2D) and three-dimensional (3D) measurements. The COMEX and Electricité de France companies developed a similar project (Poupart, Benefice, & Plutarque, 2001). In this case, an ROV manufactured by COMEX was localized using five LBL transponders. Again, several video cameras together with a 2D (double-spot)

View this article online at wileyonlinelibrary.com • DOI: 10.1002/rob.20351

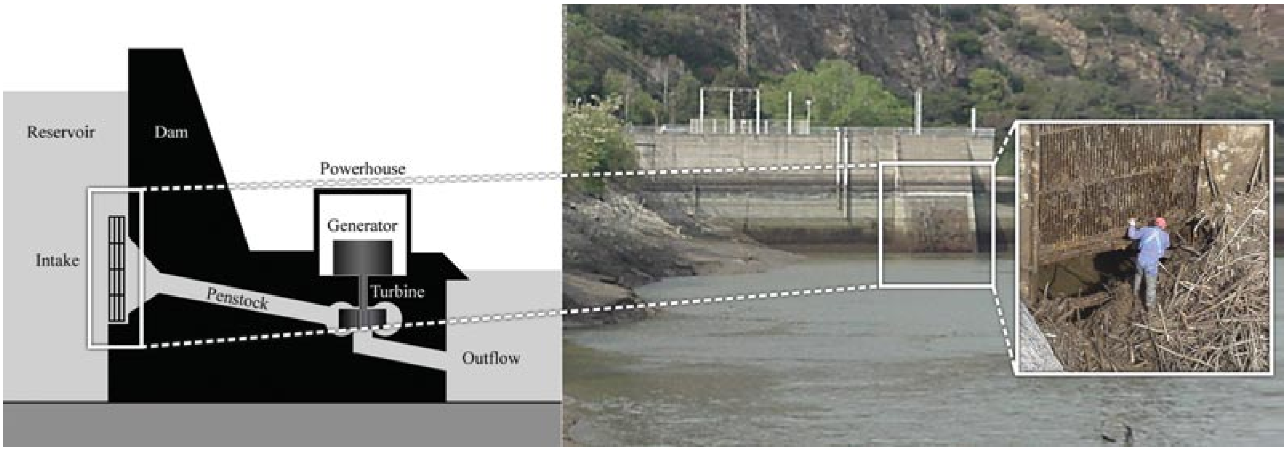


Figure 1. Vegetal residuals accumulated in the protecting fence of the water intake obstruct the inlet to the penstock pipe. Although this fence is commonly submerged, an unusually low level of the waterline made it visible in this picture.

laser system was used to take measurements. During 2002, in collaboration with the Research Development and Technological Transfer Centre (CIFATT) IPA-Cluj, our team used the URIS robot working as an ROV to build an image mosaic (Batlle, Nicosevici, García, & Carreras, 2003) of a small area of the wall of the Tartina Dam in the surroundings of Cluj, Romania. To the best of the authors' knowledge, this is the first time that image mosaicking techniques were applied for dam inspection. This solution gives an important added value because it provides civil engineers with a global view of the inspected area. Unfortunately, the ROV was not localized, and hence the resulting image mosaic was not georeferenced. The lack of this information makes it more difficult to perform periodic inspections on damaged spots and also to localize the areas where repair work must take place.

Although very few works in dam inspection have been reported in the literature, related work is available for inspecting the submerged portions of ship hulls, which is a similar problem, especially when mapping the side hull. In this context, most of the recent work uses acoustic imaging systems (Djapic, 2009; Englot & Hover, 2009), whose principal disadvantage is the cost of acoustic cameras, which is about one order of magnitude greater than that of standard vision-based systems. A few authors have reported the use of systems based on optical cameras. In Negahdaripour and Firoozfam (2006), a stereo-vision system was used for robot positioning, navigation, and mapping of the hull using stereo images. This approach did not enable autonomous operation but allowed the authors to achieve real-time orientation of an ROV with respect to the hull using uniquely stereo data, without any additional sensor. Kim and Eustice (2009) recently presented a calibrated monocular camera system mounted on a tilt actuator to keep a nadir view to the hull. In this work, the authors used a pose-graph simultaneous localization and mapping (SLAM) algorithm with an extended information filter for inference. A characteristic of these works is that the proposed methods focus solely on the construction of local maps and, hence, georeferencing is not effectuated. This makes sense for ships and other mobile structures whose position is not fixed. However, as mentioned before, tasks such as dam inspection will benefit from this information during posterior surveillance or maintenance tasks. In a related line of work, imaging systems have also been used for the inspection of underwater pipelines used in the telecommunication and oil industries (Antich, Ortiz, & Oliver, 2005; Balasuriya & Ura, 2002; Petillot, Reed, & Bell, 2002). However, most of the works focus on the detection and tracking of pipes rather than on the creation of visual maps.

This article proposes the use of a highly maneuverable AUV for automatically surveying a dam's wall while snapping pictures and gathering navigation data in order to build a globally optimized and georeferenced photomosaic to enable systematic inspections. The robot can be programmed using a mission control language (MCL). Missions are executed by the intelligent control architecture of the robot, allowing the vehicle to follow a predefined path while ensuring the vehicle safety. Navigation is achieved by combining dead-reckoning data from the onboard sensors, a Doppler velocity log (DVL), and a motion reference unit (MRU) assisted by a fiber-optic gyro (FOG), with the corrections from a moored buoy equipped with an ultra-short baseline (USBL) compensated by a differential GPS (DGPS) and an MRU. A mechanical scanning imaging sonar is used to detect and track the line feature corresponding to the dam's wall. Combining the navigation data and the estimated line feature, is it possible to georeference the images snapped with the calibrated monocular vision system. For the problem at hand, and because a dam wall can be locally approximated by a planar surface with high accuracy, simple methods for 2D photomosaicking can do the work, it not being necessary to use more sophisticated methods for 3D mapping as in Nicosevici, Gracias, Negahdaripour, and

García (2009), Pizarro et al. (2004), or Williams and Mahon (2004). Then, the visual information is merged with the navigation data to provide a globally registered photomosaic. At the end, a blending process is applied to smooth seam transitions among the different images that constitute the mosaic, compensating for light artifacts and improving the visual perception of the scene.

The article is organized as follows. First, the experimental vehicle (Section 2) and its intelligent control architecture (Section 3) are presented. Then, Sections 4 and 5 introduce, respectively, the navigation system and the module used to detect and track the dam wall. Section 6 explains the method used to build the photomosaics, and the experimental results are reported in Section 7. Finally, conclusions are presented in Section 8.

2. ICTINEU AUV

The research platform used in this project was composed of the Ictineu robot [see Figure 2(a)] and a surface buoy for globally localizing the vehicle [see Figure 2(b)].

2.1. The Vehicle

The Ictineu vehicle (Ribas, Palomer, Ridao, Carreras, & Hernàndez, 2007) was conceived around a typical open frame design [see Figure 2(a)] as a research prototype for validating new technologies. It is a small (0.8 × 0.5 × 0.5 m), light (60 kg in air), and very-shallow-water (depth rating 30 m) vehicle. Although the hydrodynamics of open-frame vehicles is known to be less efficient than that of closed-hull-type vehicles, they are suitable for applications not requiring movements at high velocities or traveling long distances, such as a visual inspection of a dam. The vehicle is practically neutral (approx. 0.6 kg of positive buoyancy), it is stable in roll and pitch due to the weight and volume distribution, and it can be controlled in surge, sway, heave, and yaw with six Seabotix SBT150 thrusters. Each propeller can generate 22 N of force. Four of them are placed horizontally in a rhombus configuration that makes it possible to thrust in any horizontal direction simultaneously (surge and sway) and perform rotation (yaw). The other two thrusters are placed vertically and can actuate the heave degrees of freedom (DOF). The four DOF allow the scanning of the wall of the dam while maintaining the distance and point of view of the camera while moving vertically and horizontally.

The vehicle has two big cylindrical pressure vessels that house the power and computer modules. The power module contains a pack of sealed lead acid batteries, which supplies 24 Ah at 24 V and can provide the Ictineu with more than 1 h of running time. The computer module has two PCs, one for control (PC104 AMD GEODE-300MHz) and one for image and sonar processing (mini-ITX computer Via C3 1 GHz) connected through a 100-Mbps Ethernet switch. An interesting characteristic of this vehicle is that it can operate either as an ROV (tethered mode) or as an AUV (untethered mode). An optional umbilical cable can be connected to the two modules to supply power and Ethernet communication to the vehicle. This mode of operation is very useful not only to operate the Ictineu as a ROV but also to monitor the software architecture while the vehicle is performing the dam inspection autonomously. When working in full AUV mode, the umbilical cable is removed and the vehicle relies on batteries to power all the systems and therefore has a limited running time but a longer range of operation. Communication can then be established using an acoustic modem, which is integrated in the USBL sensor. The sensors onboard the Ictineu AUV are listed in Table I.

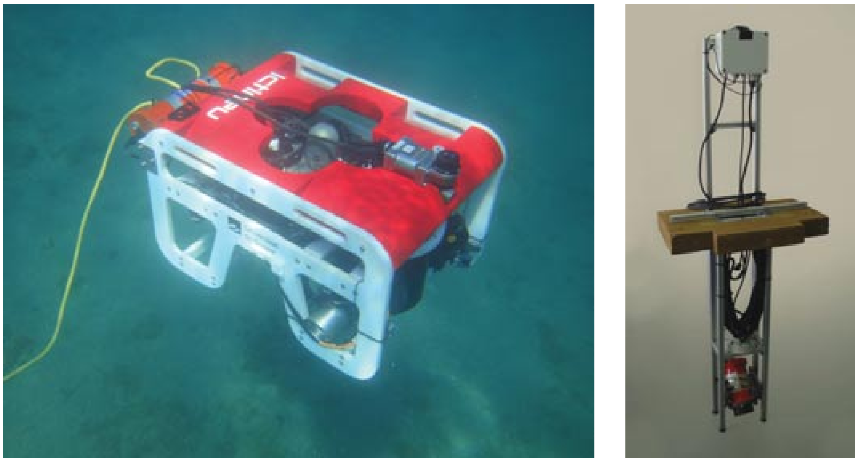

(a) (b)

Figure 2. (a) AUV Ictineu in water. (b) Surface buoy for localizing the vehicle.

Table I. Ictineu AUV sensor suite.

Sensor	Model	Characteristics
MSIS	Tritech Miniking	Maximum range: 100 m Horizontal beam width: 3 deg Scan rate: 5–20 s/360-deg sector Frequency 675 kHz
DVL	Sontek Argonaut	Accuracy: 0.2% of measured velocity Frequency: 1,500 kHz
MRU + FOG	Tritech iGC/iFG	Accuracy: better than 1 deg
Vision system	B&W camera	1/3-in. charge-coupled device 0.01 lux 50 Hz 625 lines
	Halogen lights	
Echo sounder	Airmar Smart Sensor	Frequency: 235 kHz
USBL	Linkquest Tracklink 1500 HA	Angular accuracy: 0.25 deg Range accuracy: 0.5% slant range Frequency: 31–43.2 kHz

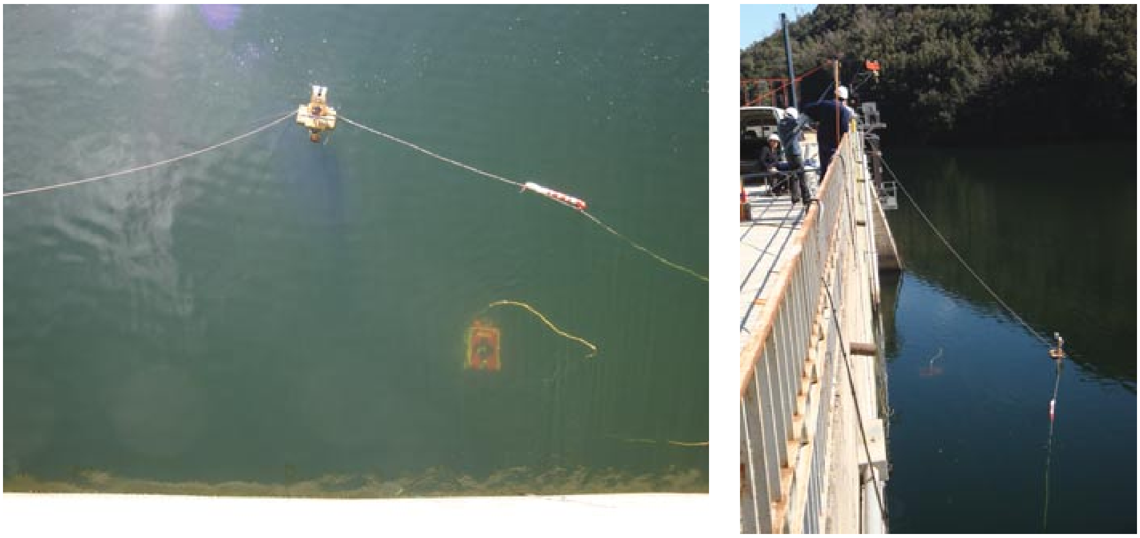
2.2. The Surface Buoy

The purpose of the buoy is to determine the absolute position in world coordinates of the USBL transponder mounted on the vehicle. This information is necessary for georeferencing the sensor data acquired with the vehicle as well as to reduce the drift that inherently affects the dead-reckoning navigation estimate. A USBL consists of a transceiver, which is usually placed on the surface, and a transponder mounted on the AUV. The device determines the position of the vehicle by calculating the range and angles obtained after the transmission and reply of an acoustic pulse between the transceiver and the transponder.

This buoy system is composed of a Linkquest Tracklink 1500 USBL transceiver operating at 31–43.3 kHz and its supporting sensors, a DGPS and an Xsens MTi MRU, whose objective is to correct for the position and attitude changes of the transceiver during the vehicle position estimation process. The different components are attached to a 1.5-m-high aluminum structure [see Figure 2(b)], which depending on the mission needs can be mounted outboard of a small boat or attached to a mooring buoy, as in the case of the experiments presented in this paper (see Figure 3). The data logging is performed on an external computer connected to the sensors through RS232. To integrate the sensor information acquired with the robot with the position estimates from the USBL system, the data should have a common time base. For this reason, the computers in charge of the data logging are synchronized prior to the beginning of the mission. No significant time drift has been observed for the typical mission duration (half an hour for the reported results).

3. INTELLIGENT CONTROL ARCHITECTURE

The intelligent control architecture has the goal of fulfilling the mission that has been predefined by a user. The architecture acts as an hybrid control architecture (Arkin, 1998), combining the continuous-time control laws corresponding to vehicle behaviors with a discrete event system (DES) responsible for enabling and disabling these behaviors according to a particular mission plan. The hybrid control architecture is divided at the same time into two main blocks: the software control architecture (SCA) and a mission control system (MCS) (see Figure 4). The SCA contains the vehicle behaviors but also other systems or modules. A perception module computes the state of the vehicle, and it obtains and processes sensor information from the robot interface module and sends the data to the control module, which finally sets an output on the vehicle's actuators. As can be seen in Figure 4, the control module is composed of a set of behaviors, a coordinator, and a velocity controller. The MCS is responsible for enabling and disabling these behaviors. A Petri net formalism has been chosen as the DES representation to model, program, and execute AUV missions. To simplify the description of Petri net missions, a MCL has been proposed. The user programs the mission with MCL, and the MCL compiler transforms this code into a Petri net (Palomeras, Ridao, Carreras, & Silvestre, 2009). Then, the real-time Petri net player executes this Petri net mission by enabling/disabling the different behaviors of the SCA. An architecture abstraction layer (AAL) keeps the MCS independent from the vehicle SCA. The next subsections offer details about the SCA as well as the MCS.



(a) (b)

Figure 3. Images of the Ictineu vehicle and the surface buoy during the dam inspection experiments from the top of the dam (a) and from the shore (b).

3.1. Software Control Architecture

The vehicle architecture has to guarantee the AUV functionality. From the implementation point of view, the real-time POSIX interface, together with the CORBA-RT (Tao, 2003), has been used to develop the architecture as a set of distributed objects with soft real-time capabilities. The

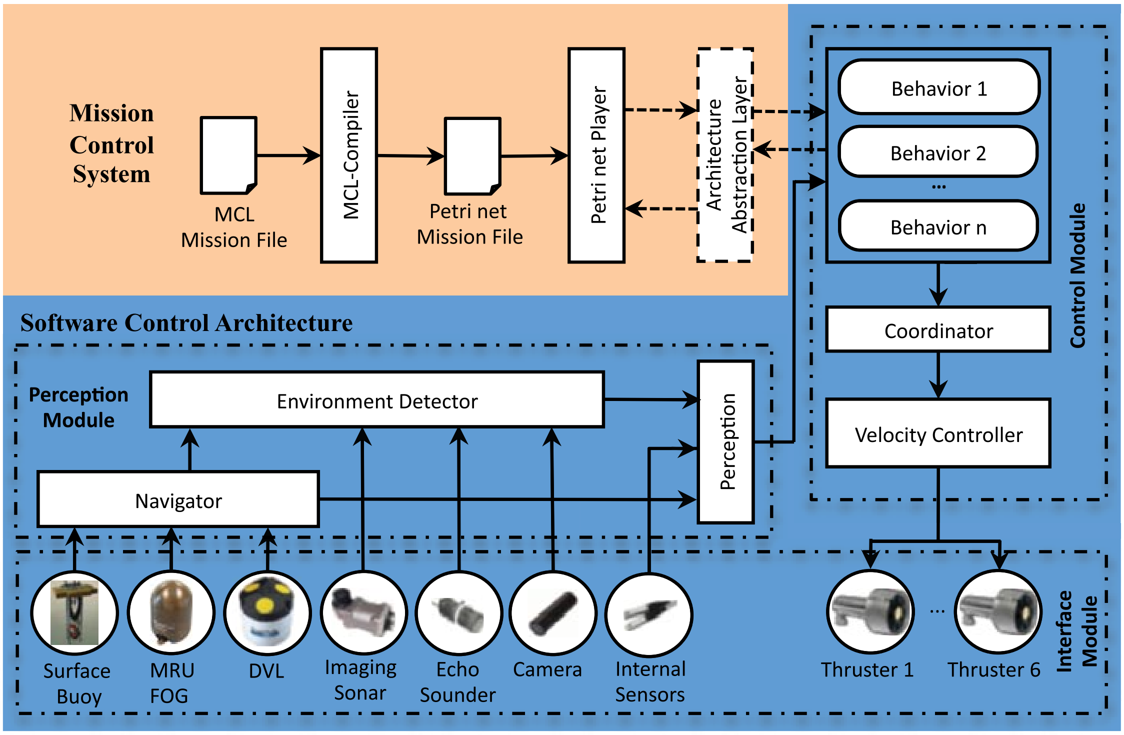


Figure 4. Schematic of the intelligent control architecture used on Ictineu.

architecture is composed of a base system and a set of objects customized for the desired robot. Software objects included in the architecture provide soft real-time capabilities, guaranteeing the execution period of tasks such as the controllers or the sensors. Another important part of the base system is the loggers. A logger object is used to log data from sensors, actuators, or any other object component. Moreover, all the computers in the network are synchronized so that all the data coming from different sensors can be time related. The SCA is divided into three modules: robot interface module, perception module, and control module.

Robot Interface Module

The robot interface module contains software objects that interact with the hardware. Sensor objects are responsible for reading data from sensors, and actuator objects are responsible for sending commands to the actuators. Sensor objects include drivers for the surface buoy, MRU–FOG, DVL, imaging sonar, echo sounder, and camera. There are also objects for the internal sensors, such as the water leakage detectors and internal temperature and pressure sensors that allow for the monitoring of the conditions within the pressure vessels. For the output, one actuator object has been designed for every thruster. A virtual version of every component allows us to transparently connect the robot control architecture to a real-time graphical simulator allowing hardware-in-the-loop simulations (Ridao, Batlle, Ribas, & Carreras, 2004).

Perception Module

This module contains two components: the navigator and the environment detector. The navigator object has the goal of estimating the position and velocity of the robot (Section 4), combining the data obtained by all the navigation sensors. The control module uses the navigation data provided by the navigator, keeping the behaviors independent of the physical sensors being used for the localization. In the context of this application, the environment detector is used to detect the relative position (range and bearing) of the dam's wall with respect to the robot, as explained in Section 5.

Control Module

The control module acts as the reactive layer of our hybrid control architecture. It receives sensor inputs from the perception module, keeping the behaviors independent of the physical sensors being used, and sends command outputs to the actuators residing in the robot interface module. Several behaviors are defined to perform the desired mission. Behaviors can be enabled and disabled and their parameters can be changed by means of actions sent by the MCS through the AAL. Also, events can be produced by the behaviors to announce that a goal has been reached or a failure has been detected within the vehicle architecture.

In particular, the following behaviors were developed to perform the inspection of the dam:

- **GoTo:** Performs a trajectory from the current position to the specified X, Y, and Z positions and yaw orientation. Takes the position and orientation data from the navigator object.
- **WallInspection:** Used to follow a 2D path in Y and Z DOF in front of the wall of the dam following a sequence of waypoints (*path*).
- **AchieveHeadDist:** Allows the robot to keep a preprogrammed distance and heading with respect to the wall, which are estimated by the environment detector object.
- **Surface:** Performs a vertical trajectory from the current position to the surface.
- **StartCamera:** Sensing behavior used to enable/disable the vehicle frontal camera to gather images of the wall.
- **Alarms:** Sensing behavior responsible for issuing an event if water leakage or a dangerous temperature or pressure is detected inside a pressure vessel.

3.2. Mission Control System

The MCS acts as a deliberative and control execution layer of our hybrid control architecture. However, the MCS does not automatically plan the set of active behaviors at each moment. A user describes the mission using a high-level language called MCL, and then the mission is automatically translated into a DES represented as a Petri net that is able to decide which behaviors will be executed depending on the observed events.

The MCS has been designed to be as generic as possible. To achieve this goal, the proposed MCS presents a clear interface with any particular vehicle control architecture based on *actions* and *events*. Actions are specific orders given by the MCS to the control architecture to enable one behavior or to configure parameters. Events are some facts that the control architecture detects or measures, which are sent to the MCS. Between the MCS and the vehicle control architecture, there is an AAL that adapts these *actions* and *events* for every particular architecture (see Figure 4). The AAL depends on the control architecture being used, allowing the MCS to remain architecture independent. With the AAL, it is possible to use this MCS approach in different vehicles with different control architectures.

A mission is programmed using a set of parallel execution flows, each one involving an iterative/sequential execution of tasks. In our system, a task is the basic execution block at the event-driven domain. When a task begins, it enables (by means of *actions* sent through the AAL) the set of behaviors (running in the time domain) needed to achieve its associated goals. When certain conditions are detected (by means of *events* or *timeouts*), the task disables the set of behaviors. Tasks have two possible ending status, *correct* or *fail*. Internally, tasks are modeled using the Petri net formalism (Murata, 1989) and can be joined using control structures (if-then-else, try-catch...) to set up mission programs. The control structures themselves are encoded using Petri nets, and so it is the full mission program. The adoption of the Petri net formalism allows us to do formal verifications such as asserting deadlock avoidance and ensuring the mission progress from the initial state to one of the possible ending estates. To avoid the burden of programming the mission directly as a Petri net, a MCL has been defined (Palomeras, Ridao, Carreras, & Silvestre, 2008). MCL mission programs are automatically translated into a Petri net representing the mission. A complete description of our MCS and MCL can be found in Palomeras et al. (2008) and Palomeras et al. (2009), respectively.

The following structures are supported by the MCL language:

Sequence: Used to execute one block of tasks after another. In an MCL code (an example can be seen in Algorithm 1), the sequence command is represented with a semicolon. It is worth noting that the last task in a block finishes without semicolon.

Try-Catch-Do: Executes the try block in parallel with the catch block. If the former finishes before the latter, the catch block is canceled and the execution continues after the try-catch-do structure. However, if the catch block finishes first, the try block is aborted and the do block is executed.

Parallel-And: Executes two blocks in parallel. If both blocks finish correctly, the whole control structure finishes correctly. Otherwise, the parallel-and finishes with a *fail* status.

Parallel-Or: Executes two blocks in parallel. The first structure to finish aborts the other. The parallel-or finishes with the final state of the first block to end.

If-Then-Else: Executes the block inside the if statement and, depending on whether the block ends with an *ok* or *fail*, the block inside the then statement or the else statement is executed, respectively.

While-do: Executes the block inside the while statement. If this block finishes with an ok, it executes the do statement; otherwise it ends with an ok. If the do statement finishes with an ok, it executes again the while statement; otherwise the whole structure ends with a fail.

The control structures together with the tasks constitute the MCL code that will be finally compiled to create the Petri net that represents the mission.

For the mission presented in this paper, the tasks that were needed are as follows:

- **Alarms()** Checks the events related with the alarms behavior.
- **StartLogs()** Starts or stops all the log objects to save the data.
- **GoTo()** Sets the goal position of the GoTo behavior and waits until the position is achieved.
- **AchieveHeadDist()** Sets the parameters of the AchieveHeadDist behavior and waits until the heading and distance are achieved.
- **KeepHeadDist()** Sets the parameters of the AchieveHeadDist behavior and starts or stops it.
- **KeepHeadDistException()** Evaluates whether the heading and distance cannot be achieved by checking its corresponding event.
- **StartCamera()** Starts or stops the camera acquisition.
- **WallInspect()** Sets the parameters of the WallInspection behavior and waits until the trajectory is achieved.
- **Surface()** Starts the Surface behavior and waits until it finishes.

Using these tasks, the dam inspection mission is described as follows. After checking all the alarms, the sensor logs are enabled and, in parallel with the rest of the mission, two monitors are used to send an *event* in case the pressure/temperature exceeds a threshold or if a water leakage is detected inside a pressure vessel. Using the *GoTo* behavior, the vehicle goes to the initial waypoint of the survey and, after achieving the desired orientation and distance with respect to the dam wall, the survey starts. During the survey, imagery is recorded using the black-and-white camera looking forward. When the survey finalizes, the camera stops, the vehicle goes to the recovery position, and the logs are disabled. If during the mission any of the monitors generates an alarm *event*, the mission is aborted and the vehicle surfaces.

The mission program (see Algorithm 1) was programmed and compiled into a Petri net representing the mission (Palomeras et al., 2008). The try-catch-do control structure was used to describe all possibilities and to execute the blocks of actions accordingly. Figure 5 shows a simplified version of the resulting Petri net. In the figure, tasks have been simplified to single places (gray vertexes) and transitions are shown as squares with a T inside. The places labeled as begin, abort, ok, and fail are used to start, stop, and indicate the finalization state of each task. The Petri net starts with begin/abort places and finalizes with ok/fail

Algorithm 1 Dam Inspection Mission Code in MCL

```
MISSION {
    StartLogs(enable) ;
    TRY {
        GoTo(initialPos) ;
        AchieveHeadDist(h, d) ;
        KeepHeadDist(enable, h, d) ;
        TRY {
            StartCamera(enable) ;
            WallInspect([path]) ;
            StartCamera(disable)
        }
        CATCH {
            KeepHeadDistException()
        } DO{
            Surface()
        } ;
        KeepHeadDist(disable) ;
        GoTo(recoveryPos)
    }
    CATCH {
        Alarms()
    } DO{
        Surface()
    } ;
    StartLogs(disable)
}
```

Note that semicolons represent sequential execution between tasks.

places. It can be proved, by construction and without further verification, that the obtained Petri net will progress from the initial begin place to one of these two possible final places without any deadlock.

4. NAVIGATOR MODULE

The navigator module is one of the key elements of this system, both to estimate the vehicle position during the execution of a mission and to provide an input trajectory for the mosaicking framework. The navigator executes an extended Kalman filter (EKF) (Kalman, 1960; Thrun, Burgard, & Fox, 2005), a stochastic sensor fusion algorithm that estimates the vehicle state (position, heading, and velocity) and its corresponding uncertainty following a two-step recursive process that alternates the predictions from a model with the updates from sensor measurements. Next, a detailed description of this filter is provided.

4.1. State Vector

All the information to be estimated by the filter is stored in a state vector. In our particular implementation, the state vector contains information regarding the position

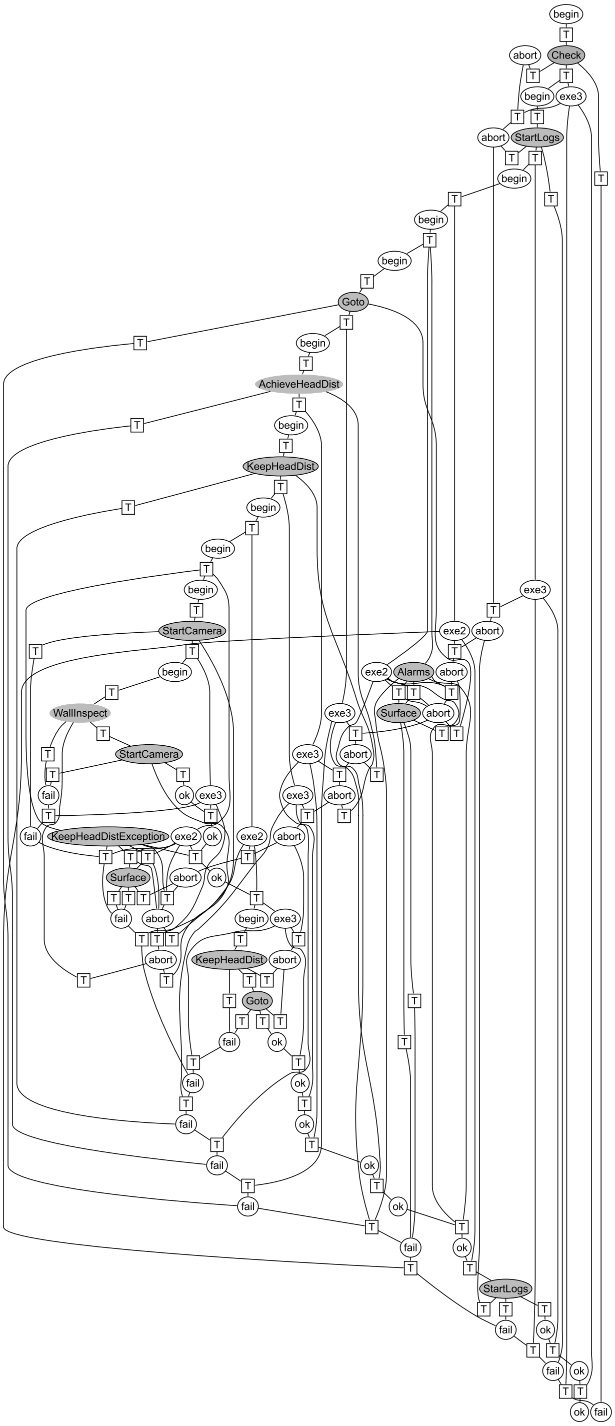


Figure 5. Automatically generated Petri net for Algorithm 1. The reported net is a simplification in which every *task* has been simplified to a single place (gray vertexes).

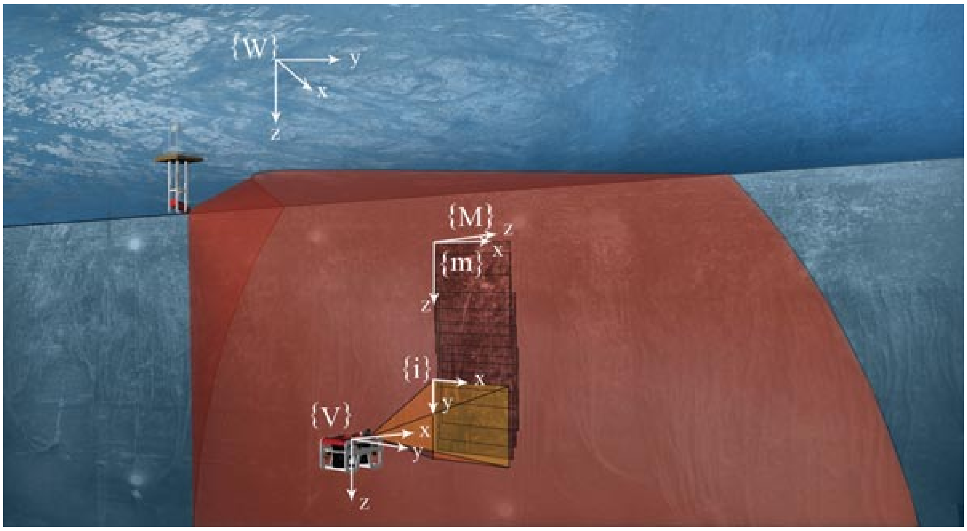


Figure 6. Reference frames involved in the proposed system.

and velocity of the vehicle at time k:

$$\mathbf{x}(k) = [x\ y\ z\ \psi\ u\ v\ w\ r]^T, \tag{1}$$

where, following the nomenclature proposed in Fossen (1994), the vector $[x\ y\ z\ \psi]$ represents the position and heading of the vehicle in the global reference $\{W\}$ (see Figure 6) and $[u\ v\ w\ r]$ are the linear and angular velocities, which are represented on the vehicle's coordinate frame $\{V\}$. Because the Ictineu vehicle is passively stable in *roll* and *pitch*, their corresponding angles and velocities have not been included in the state vector.

4.2. Initializing the Filter

At the beginning of the mission and before starting the Kalman filter, the initial value of the state vector $\mathbf{x}(0)$ should be determined. With measurements from the surface buoy available, it is possible to determine the vehicle's initial position with respect to the global reference $\{W\}$. Therefore, the first measurement from the buoy system, and its associated uncertainty, will be used to define the initial position of the vehicle. Because $\{W\}$ is aligned with the north, the same strategy can be used to initialize the heading with the compass-referenced FOG. During the initialization phase, the vehicle is kept almost static. Therefore, it is not unrealistic to initialize the velocities with a zero mean but including some uncertainty to account for possible perturbations. The resulting estimate for the state vector at time 0 is

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} x_U \\ y_U \\ z_U \\ \psi_C \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{P}(0) = \begin{bmatrix} \sigma^2_{Ux} & \sigma_{Uxy} & \sigma_{Uxz} & 0 & 0 & 0 & 0 & 0 \\ \sigma_{Uyx} & \sigma^2_{Uy} & \sigma_{Uyz} & 0 & 0 & 0 & 0 & 0 \\ \sigma_{Uzx} & \sigma_{Uzy} & \sigma^2_{Uz} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_C & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_u & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_w & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_r \end{bmatrix}, \tag{2}$$

where the subindex U stands for the USBL and C for the compass. Note that the position covariance submatrix is not diagonal. The data from the buoy system are the result of combining the information from the GPS and the MRU with the vehicle position measured from the USBL transceiver; hence the estimated global position is correlated. These correlations can be determined by setting each sensor's covariance according to the manufacturer specifications and then defining the necessary transformations to produce the vehicle position in world coordinates; that is, combining a GPS reading with the angular measurements from the MRU to obtain the position and attitude of the buoy and then composing it with the USBL measurement (in spherical coordinates) to produce the vehicle position in Cartesian coordinates.

4.3. System Model

A simple four-DOF constant velocity kinematics model is used to predict how the state will evolve from time $k-1$ to time k:

$$\mathbf{x}(k) = f(\mathbf{x}(k-1), \mathbf{n}(k-1)),$$

$$\begin{bmatrix} x \\ y \\ z \\ \psi \\ u \\ v \\ w \\ r \end{bmatrix}_{(k)} = \begin{bmatrix} x + \left(ut + n_u \dfrac{t^2}{2}\right)\cos(\psi) - \left(vt + n_v \dfrac{t^2}{2}\right)\sin(\psi) \\ y + \left(ut + n_u \dfrac{t^2}{2}\right)\sin(\psi) + \left(vt + n_v \dfrac{t^2}{2}\right)\cos(\psi) \\ z + wt + n_w \dfrac{t^2}{2} \\ \psi + rt + n_r \dfrac{t^2}{2} \\ u + n_u t \\ v + n_v t \\ w + n_w t \\ r + n_r t \end{bmatrix}_{(k-1)}, \quad (3)$$

where t is the time period and $\mathbf{n} = [n_u \;\; n_v \;\; n_w \;\; n_r]^T$ represents a vector of white Gaussian acceleration noises with zero mean whose covariance values have been set empirically according to the observed performance of the constant velocity model. They are additive in the velocity terms and propagate through integration to the position. The covariance of the $\mathbf{n}$ vector is represented by the system noise matrix $\mathbf{Q}$:

$$E\,[\mathbf{n}(k)] = \mathbf{0}, \qquad E[\mathbf{n}(k)\mathbf{n}(j)^T] = \delta_{kj}\mathbf{Q}(k), \qquad (4)$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_{n_v}^2 & 0 & 0 & 0 \\ 0 & \sigma_{n_u}^2 & 0 & 0 \\ 0 & 0 & \sigma_{n_w}^2 & 0 \\ 0 & 0 & 0 & \sigma_{n_r}^2 \end{bmatrix}. \qquad (5)$$

The model described in Eqs. (3) is nonlinear and therefore the prediction should be performed with the EKF equations (Thrun et al., 2005).

4.4. Measurement Model

The vehicle is equipped with a number of sensors providing direct observations of particular elements of the state vector. The general linear model for such measurements is written in the form

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k|k-1) + \mathbf{m}(k),$$

$$\begin{bmatrix} z_{u_D} \\ z_{v_D} \\ z_{w_D} \\ z_{z_P} \\ z_{\psi_C} \\ z_{x_U} \\ z_{y_U} \\ z_{z_U} \end{bmatrix}_{(k)} = \begin{bmatrix} 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \psi \\ u \\ v \\ w \\ r \end{bmatrix}_{(k)} + \begin{bmatrix} m_{u_D} \\ m_{v_D} \\ m_{w_D} \\ m_{z_P} \\ m_{\psi_C} \\ m_{x_U} \\ m_{y_U} \\ m_{z_U} \end{bmatrix}_{(k)}, \qquad (6)$$

where the subindex U stands for the USBL, C for the compass, D for the DVL, and P for the pressure sensor; $\mathbf{z}$ is the measurement vector, and $\mathbf{m}$ represents a vector of white Gaussian noises with zero mean, affecting the observation process. The covariance matrix of the measurement noise $\mathbf{R}$ is given by

$$E[\mathbf{m}(k)] = \mathbf{0}, \;\; E[\mathbf{m}(k)\mathbf{m}(j)^T] = \delta_{kj}\mathbf{R}(k), \qquad (7)$$

$$\mathbf{R}(k) = \times \begin{bmatrix} \sigma_{Du}^2 & \sigma_{Duv} & \sigma_{Duw} & 0 & 0 & 0 & 0 & 0 \\ \sigma_{Dvu} & \sigma_{Dv}^2 & \sigma_{Dvw} & 0 & 0 & 0 & 0 & 0 \\ \sigma_{Dwu} & \sigma_{Dwv} & \sigma_{Dw}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_P^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\psi^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{Ux}^2 & \sigma_{Uxy} & \sigma_{Uxz} \\ 0 & 0 & 0 & 0 & 0 & \sigma_{Uyx} & \sigma_{Uy}^2 & \sigma_{Uyz} \\ 0 & 0 & 0 & 0 & 0 & \sigma_{Uzx} & \sigma_{Uzy} & \sigma_{Uz}^2 \end{bmatrix}_{(k)}. \qquad (8)$$

The covariance values for the $\mathbf{R}$ matrix have been assigned according to the specifications from the manufacturers of each particular sensor. The form of the observation matrix $\mathbf{H}$ changes according to the measurements available from the sensors at each time step k.

Note that the DVL covariance submatrix is not diagonal. The reason behind these correlations is that the measurements provided by the DVL are not directly observed but are calculated from the projection of the vehicle's velocity onto the multiple-beam axes of the sensor. The correlation of the measurements depends on the beam geometry and, hence, should be determined for each particular device (Brokloff, 1994). In our particular implementation, these correlations have been determined for the three-beam configuration of our DVL (Ribas, Ridao, & Neira, 2010). The same happens for the USBL covariance submatrix, which is full because the USBL fix is represented in Cartesian coordinates instead of cylindrical as usual.

4.5. Trajectory Smoothing

To produce a better trajectory estimate, the state vector can be augmented with a history of past vehicle positions (Leonard & Rikoski, 2001; Smith, Self, & Cheeseman, 1990). Executing an augmented state Kalman filter allows the propagation of sensor information to past states through their correlation with the current estimate. As a result of this process, their values are refined and a smooth estimated trajectory is obtained. To limit the computational burden of a growing augmented state vector, it is possible to remove the older clones from the state after some time. In an estimator such as the one proposed here, the most recent clones are the ones more strongly correlated with the current vehicle state. Therefore, the information from a sensor measurement will propagate strongly to them but have almost no effect to the older ones.

Performing this trajectory smoothing is not necessary during the mission. However, it has been employed during the offline postprocessing of navigation data to produce a better trajectory estimate for the photomosaic-building framework.

5. WALL DETECTION AND TRACKING

This section presents the algorithm developed for the environment detector (see Section 3), a key element of the system whose function is to detect and track the dam wall during the autonomous execution of the survey using the mechanically scanned imaging sonar (MSIS) onboard the vehicle. The objective of this algorithm is to provide the necessary information to ensure that the vehicle is facing the dam wall perpendicularly and at a particular distance to avoid distortions on the acquired images.

Although dam walls are generally curved to improve the structural resistance to the water pressure, they can be locally approximated by a line feature (see Figure 7). These vertical and planar structures can be efficiently detected in acoustic images (Kazmi, Ridao, Ribas, & Hernàndez, 2009). The process begins with the estimation of the initial position of the wall with respect to the vehicle. First, the image gathered with the MSIS is binarized. After discarding the echo intensity readings in the neighborhood of the sonar head, the highest-intensity echo return is located and used to compute the range for each beam corresponding to a particular bearing. A modified procedure is used to compute the Hough transform (HT) in which the set of admissible lines containing a point is restricted to those that also satisfy a tangency criterion resulting from the finite beam width of the sonar. Once the initial line candidate has been computed, it is further refined through a tracking process involving a Kalman filter, in which the information from the heading sensor and the points extracted from the sonar data are used to update the line estimate and hence track the position of the dam wall with respect to the vehicle. It is worth mentioning that although this tracking filter could be easily integrated into the navigation filter presented in the preceding section, it has been implemented as an independent module to allow for alternative tracking methods in those situations in which the planar wall assumption does not hold. Next, we describe the sonar model used for the HT voting.

5.1. Sonar Model

Owing to the horizontal beam width, a sonar reading cannot be related to a single point in space. In Leonard and Durrant-Whyte (1992), a sonar model is described in which a polar measurement $[\rho_S \;\; \theta_S]$ is not associated with a unique point but with an arc of points placed at a range ρ_S and within an aperture α being centered in the direction θ_S of the actual sonar measurement. Therefore the points $[\rho_{\alpha,i} \;\; \theta_{\alpha,i}]$ associated with the measurement are those satisfying

$$\theta_S - \alpha/2 \leq \theta_{\alpha,i} \leq \theta_S + \alpha/2; \rho_{\alpha,i} = \rho_S, \tag{9}$$

which extends the sonar measurement with a range of possible bearings within the horizontal aperture α of the beam (where $\alpha = 3$ deg in our case). Hence, the set of candidate lines (corresponding to planar objects) that can explain the sonar measurement are those tangent to the arc. Although this model is suitable for mobile robots using low-resolution, wide-angle sonar beams, it still needs to be improved to explain the behavior experimentally observed with the MSIS. As can be seen in Figure 7, the visibility of the dam wall is not limited to those objects tangent to the narrow beam width of the sonar. In fact, high echo intensity values are clearly obtained 2–3 m around the nadir point. It can also be observed that although the wall is still visible at both ends of the line, the intensity and the precision of the measurement decreases. In Ribas, Neira, Ridao, and Tardós (2006), the previous sonar model was further extended for beams with a narrow horizontal aperture and working underwater. In this case, the measurements are associated not only to the arc tangent surfaces within the horizontal aperture but also to the surfaces incident at certain angles β (for the MiniKing imaging sonar, we have observed that a $\beta = 60$ deg works fine for most of the tested wall surfaces). Now, a line $[\rho_{\beta,k}, \theta_{\beta,k}]$ candidate to correspond with a sonar measurement is described by relation (10) and Eq. (11) (see Figure 8):

$$\theta_{\alpha,i} - \beta \leq \theta_{\beta,k} \leq \theta_{\alpha,i} + \beta, \tag{10}$$

$$\rho_{\beta,k} = \rho_{\alpha,i} \cos(\theta_{\alpha,i} - \theta_{\beta,k}). \tag{11}$$

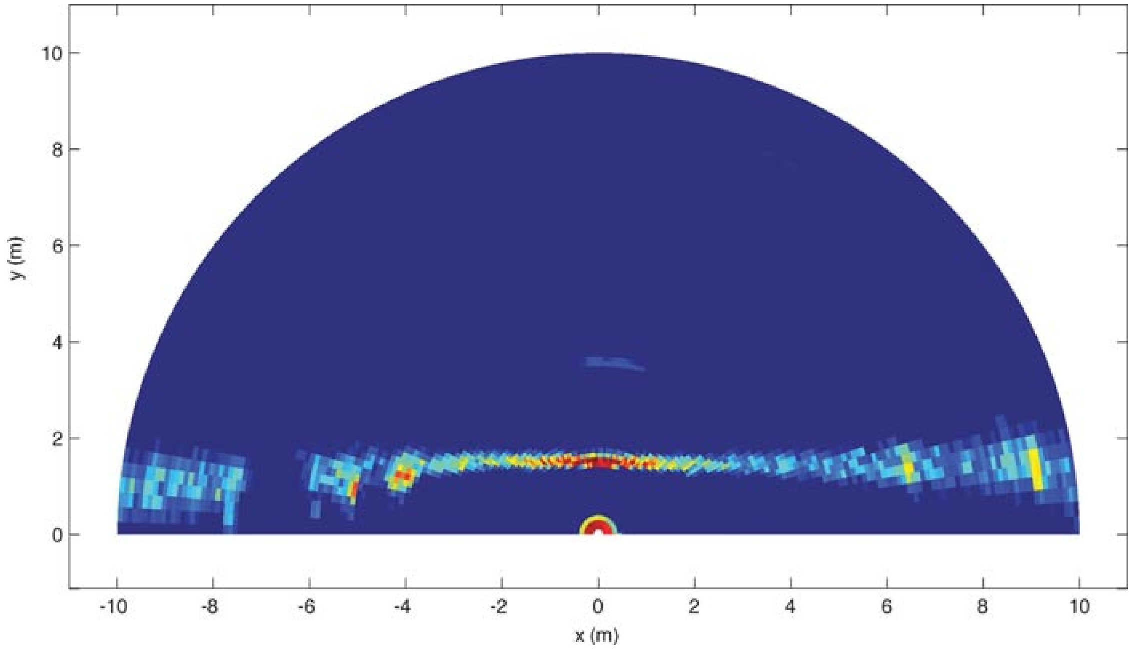


Figure 7. 180-deg sonar scan sector of the dam's wall.

5.2. Filter Initialization

As mentioned above, the HT is used to produce an initial estimate for the position of the dam wall given the measurements contained in the sonar scan. After selecting the bins with the highest echo return, the set of compatible line candidates are determined according to the model described in the preceding section. The voting process begins by initializing a Hough space consisting of a discretized representation of a $[\rho \;\; \theta]$ space in which all candidate lines are represented. Then, the subset of candidate lines that are compatible with a particular sonar measurement determine which cells of the Hough space should receive a vote. The process of assigning votes is repeated for all the measurements contained in the scan. After the voting, the cell in the Hough space with the largest number of votes is the one representing the line candidate that more likely corresponds with the real position of the wall. Therefore, this line feature $[{}^V\rho_L \, {}^V\theta_L]$ is chosen to initialize the state vector of the Kalman filter:

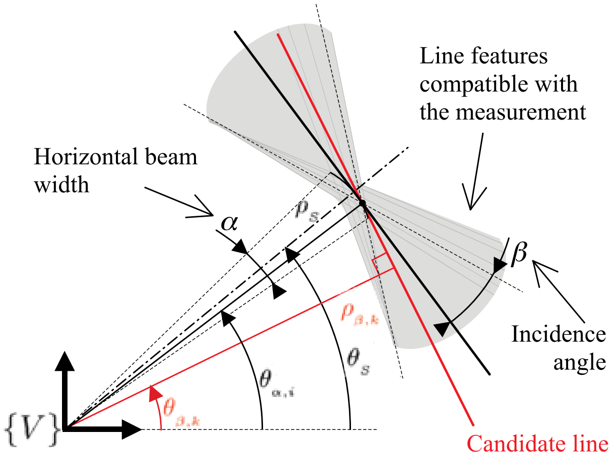


Figure 8. Modified sonar model.

$$\hat{\mathbf{x}}(0) = \begin{bmatrix} {}^V\rho_L \\ {}^V\theta_L \\ {}^W\theta_L \end{bmatrix}, \tag{12}$$

where the term ${}^W\theta_L$ corresponds to the parameterization of the line orientation with respect to the global frame $\{W\}$ defined as (see Figure 9)

$${}^W\theta_L = \psi + {}^V\theta_L. \tag{13}$$

The uncertainty of this initial state is defined as

$$\mathbf{P}(0) = \begin{bmatrix} \sigma^2_{{}^V\rho_L} & 0 & 0 \\ 0 & \sigma^2_{{}^V\theta_L} & \sigma_{{}^V\theta_L,{}^W\theta_L} \\ 0 & \sigma_{{}^W\theta_L,{}^V\theta_L} & \sigma^2_{{}^W\theta_L} \end{bmatrix}, \tag{14}$$

where it has been assumed that the robot was static enough during the first scan to consider ${}^W\theta_L$ independent from ψ, so

$$\sigma^2_{{}^W\theta_L} = \sigma^2_{{}^V\theta_L} + \sigma^2_{\psi}; \qquad \sigma_{{}^V\theta_L,{}^W\theta_L} = \sigma^2_{{}^V\theta_L}. \tag{15}$$

Although, as stated in Ribas, Ridao, Tardós, and Neira (2008), it is possible to obtain an estimate of the covariance

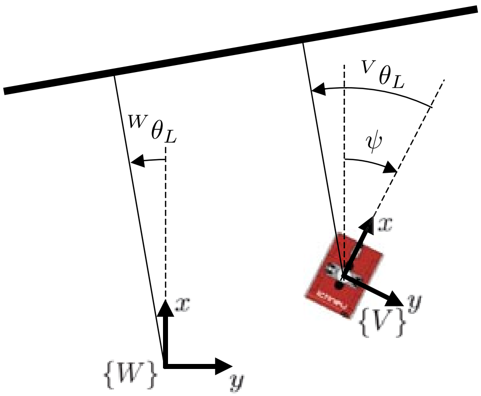


Figure 9. Description of the different angles and frames involved in the line feature estimation.

of the line feature through a careful processing of the acoustic imprint left by the line in the segmented image, in this work, for the sake of simplicity, the covariance has been initialized according to the resolution of the Hough space. The uncertainty σ_ψ^2 can be obtained at any time from the state of the navigation filter described in Section 4. Also, note that in order to simplify the spatial transformations, the sensor reference frame has been taken as coincident with the vehicle's reference frame {V}.

5.3. System Model

Because the robot is controlled to keep a relative distance and orientation with respect to the wall while surveying, the line feature representing the wall will remain almost static in the robot frame except for minor oscillations due to an imprecise control as well as some possible perturbations. Therefore, a static motion model including Gaussian noise to account for these perturbations is used in the motion prediction step of the EKF:

$$\mathbf{x}(k) = \mathbf{x}(k-1) + \mathbf{n}(k-1)), \tag{16}$$

where $\mathbf{n}$ represents a vector of additive white Gaussian noises with zero mean and covariance $\mathbf{Q}$:

$$E[\mathbf{n}(k)] = \mathbf{0}, \qquad E[\mathbf{n}(k)\mathbf{n}(j)^T] = \delta_{kj}\mathbf{Q}(k), \tag{17}$$

5.4. Heading Updates

The MRU–FOG provides direct observation of the vehicle heading ψ. Therefore, according to Eq. (13), the following linear observation model can be defined:

$$\mathbf{z}_C(k) = \mathbf{H}_C\mathbf{x}(k|k-1) + \mathbf{m}(k), \tag{18}$$

$$\mathbf{H}_C = [0 \quad -1 \quad 1], \tag{19}$$

with $\mathbf{m}$ defined as an additive white Gaussian noise with zero mean and covariance $\mathbf{R}_C = \sigma_C^2$. Again, the standard Kalman filter equations are used to perform the update.

5.5. Sonar Updates

Each time the MSIS produces a beam, it may contain a high echo intensity bin corresponding with the wall. Therefore, it can provide information to update the current estimate of the line feature. Here, the measurement is defined by the polar coordinates of the point corresponding with this high-intensity bin:

$$\hat{\mathbf{z}}_M(k) = \begin{bmatrix}\rho_p & \theta_p\end{bmatrix}^T, \tag{20}$$

$$\mathbf{z}_M(k) = \hat{\mathbf{z}}_M(k) + \mathbf{v}(k), \tag{21}$$

$$E[\mathbf{v}(k)] = \mathbf{0}, \tag{22}$$

$$E[\mathbf{v}(k)\mathbf{v}(j)^T] = \delta_{kj}\mathbf{R}_M(k), \tag{23}$$

where $\mathbf{R}_M$ corresponds to the uncertainty of the MSIS sonar point measurement. To relate the points to the line, an implicit measurement equation is used (Castellanos & Tardós, 1999), which states that the point belongs to the line, or equivalently, that the point to line distance is zero:

$$h_M(\mathbf{z}_M(k), \mathbf{x}(k)) = {}^V\rho_L - x_p\cos({}^V\theta_L) - y_p\sin({}^V\theta_L), \tag{24}$$

$$= 0, \tag{25}$$

with x_p and y_p being the Cartesian representation p of the polar measurement $[\rho_p, \theta_p]$ obtained with the nonlinear transformation g:

$$\hat{\mathbf{p}}(k) = g(\hat{\mathbf{z}}_M(k)), \tag{26}$$

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \rho_p\cos(\theta_p) \\ \rho_p\sin(\theta_p) \end{bmatrix}. \tag{27}$$

Only the points that are statistically compatible with the line are used for the update. Individual compatibility for every point is checked using the compatibility test based on Mahalanobis distance:

$$\begin{aligned} D^2 = {} & h_M(\hat{\mathbf{z}}_M(k), \hat{\mathbf{x}}(k))^T\big[\mathbf{H}_{M,1}\mathbf{P}(k|k-1)\mathbf{H}_{M,1}^T \\ & + \mathbf{H}_{M,2}\mathbf{G}\mathbf{R}_M\mathbf{G}^T\mathbf{H}_{M,2}^T\big]^{-1} h_M(\hat{\mathbf{z}}_M(k), \hat{\mathbf{x}}(k)) < \chi^2_{d,\alpha}, \end{aligned} \tag{28}$$

with

$$\mathbf{H}_{M,1} = \frac{\partial h_M}{\partial \mathbf{x}}(\hat{\mathbf{x}}(k), \hat{\mathbf{p}}(k)); \qquad \mathbf{H}_{M,2} = \frac{\partial h_M}{\partial \mathbf{p}}(\hat{\mathbf{x}}(k), \hat{\mathbf{p}}(k));$$

$$\mathbf{G} = \frac{\partial g}{\partial \mathbf{p}}(\hat{\mathbf{p}}(k)), \tag{29}$$

where D^2 is the squared Mahalanobis distance and $\chi^2_{d,\alpha}$ is the chi square value of confidence; α being the confidence level and d the dimension of h (1 in this case). If the point is compatible, the update is carried out using the EKF equations for an implicit measurement function:

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}_{M,1}^{T}\big[\mathbf{H}_{M,1}\mathbf{P}(k|k-1)\mathbf{H}_{M,1}^{T} + \mathbf{H}_{M,2}\mathbf{G}\mathbf{R}_{M}\mathbf{G}^{T}\mathbf{H}_{M,2}^{T}\big]^{-1}, \tag{30}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k-1) - \mathbf{K}(k)h_{M}(\hat{\mathbf{z}}_{M}(k), \hat{\mathbf{x}}(k)), \tag{31}$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}_{M,1}]\mathbf{P}(k|k-1). \tag{32}$$

6. BUILDING PHOTOMOSAICS OF THE DAM

6.1. Camera Modeling

To carry out precise and reliable measurements from images, camera calibration is a necessary prerequisite. Images are first corrected for lens distortion, and the camera is modeled as the classical projective pinhole camera (Hartley & Zisserman, 2004) using Bouget's toolbox (Bouguet, 2010), which defines the projection of the 3D point ${}^{W}\mathbf{X}_j$ in the world onto the image plane $\{i\}$ as

$${}^{i}\mathbf{x}_j = \mathbf{K} \cdot {}^{C}_{W}\mathbf{R} \cdot [I \quad {}^{W}\mathbf{T}] \cdot {}^{W}\mathbf{X}_j, \tag{33}$$

where $\mathbf{K}$ encodes the intrinsic camera parameters and ${}^{C}_{W}\mathbf{R}$ is the 3×3 rotation matrix that transforms vector coordinates defined in the world coordinate frame $\{W\}$ to coordinates in the camera frame $\{C\}$; ${}^{W}\mathbf{T}$ is the translation of the camera center with respect to the frame $\{W\}$, and ${}^{i}\mathbf{x}_j$ is the jth 2D point in homogeneous coordinates that corresponds to the projection onto the image plane of the scene 3D point ${}^{W}\mathbf{X}_j$.

6.2. Robust Matching

A crucial part in the mosaicking workflow is the registration (or matching) of two or more images of the same scene taken at different times and from different viewpoints. For this reason, special effort is devoted in this work to ensure robust image registration through a three-step mechanism.

First, SURF (Bay, Tuytelaars, & Gool, 2006) features are extracted from both images. Then, a putative set of correspondences is obtained (Bay et al., 2006), and outliers are rejected using RANSAC (Fischler & Bolles, 1981) under a planar projective motion model (using a homography with eight DOF) (Brown & Lowe, 2003; Vincent & Laganière, 2001). The remaining inliers are then used to compute the homography that registers both images (Hartley & Zisserman, 2004). As a third step, a number of verifications are performed to ensure that this homography corresponds to a valid camera motion:

1. The obtained homography does not include improper rotations.
2. The 3D motion Euler angles associated to this planar transformation are within a certain range (Triggs, 1998).
3. The number of SURF correspondences between the two images must be larger than a certain threshold.
4. The overlapping area between the images must be higher than a threshold.

As a last step, further point correspondences are searched for in the image pairs that passed successfully the above tests. The two images are aligned according to the previously estimated homography. Then, we extract Harris corners (Harris & Stephens, 1988) in one of the images, and their correspondences are detected through correlation (Garcia, Batlle, Cufí, & Amat, 2001) in the other image. If enough correspondences are not found, the pair of images is rejected.

The process described above is used first to obtain correspondences between time-consecutive images. We assume that each pair of consecutive images of the sequence is a potential candidate to be a sequential pair of overlapping frames (although they may not overlap in some cases).

The matching process is run offline on a dedicated server that requires slightly more than a second to process every image.

6.3. Navigation Data

Although image processing produces locally coherent image alignment, it requires position estimates from vehicle navigation to avoid significant drifts in the localization of images because of the cumulative error of successive matchings. The available sensor suite (USBL, DGPS, MRU in the moored buoy, and the DVL and MRU–FOG in the vehicle) provides navigation data corresponding to the position and the heading of the vehicle with respect to the world coordinate frame $\{W\}$. Therefore, this information is also taken into account to build the photomosaic, by registering the mosaic in the world frame.

6.4. Loop Detection and Registration of Nonconsecutive Images

To obtain a globally coherent mosaic, the next step is the detection of nonconsecutive overlapping pairs of images. When the vehicle revisits a previously mapped area, new constraints for the global alignment of the photomosaic are provided. Possible crossings are identified using motion estimation from consecutive images and from vehicle navigation data. Therefore, a method to include all the possible overlaps has been devised. The center of each image is computed offline in the 2D mosaic frame according to the planar transformations obtained from the registration of consecutive images and the previously estimated navigation data. These two data sources (image cues and navigation data) are merged through a nonlinear optimization (bundle adjustment) that minimizes the cost function described in the next section. A set of images that are mapped closer than a certain threshold from the crossover point are analyzed to identify candidate image pairs that are then robustly matched using the procedure described in Section 6.2. The pairs retained with the associated correspondence points are used as input for the global alignment.

6.5. Global Alignment

Although our vehicle is roll and pitch stable, to keep the image processing system generic enough to be applied to other vehicles, we parameterize the camera trajectory in the most general terms with six DOF (3D position and orientation) using unitary quaternions (Salamin, 1979) to prevent singularities in the representation of the camera rotation. Therefore, pose rotation matrices are converted into unit quaternions ${}^{C}_{M}\mathbf{q}_j$. This gives three parameters for the position and four for the orientation, giving rise to seven parameters to be estimated for each image frame j.

An offline bundle adjustment (Triggs, McLauchlan, Hartley, & Fitzgibbon, 2000) procedure is used to minimize the residuals from image registration and estimated navigation data (see Section 4).

The cost function to perform global alignment is defined as the weighted sum of three terms: (1) the squared residual of the robot position, (2) the squared residual of the robot attitude, and (3) the squared residual of the point matchings in the image plane. The position and attitude residuals are computed from the navigation data. Given a certain point feature p, observed as p_1 and p_2 in the image planes corresponding to two different views I_1 and I_2, respectively, the point matching residual is computed as the difference between p_1 and the reprojection of p_2 into the image plane corresponding to I_1. The minimization of the cost function and the estimation of the trajectory parameters are carried out using Matlab's large-scale methods for nonlinear least squares (Escartín, Garcia, Delaunoy, Ferrer, Gracias, et al., 2008). The optimization algorithm requires the computation of the Jacobian matrix containing the derivatives of all residuals with respect to all trajectory parameters. Fortunately, this Jacobian matrix is very sparse because each residual depends on only a very small number of parameters (Triggs et al., 2000). Furthermore, it has a clearly defined block structure, and the sparsity pattern is constant (Capel, 2004). These conditions allow for the efficient computation of the Jacobian. In our implementation, analytic expressions were derived and used for computing the blocks of the Jacobian matrix.

6.6. Loop Detection and Optimization Iteration

Mosaic alignment is improved through several iterations of loop detection and optimization. After each iteration, the resulting optimized trajectory of the camera is used as a starting point for a new iteration of detection of new potential overlapping image pairs and bundle adjustment to find new constraints (i.e., more overlapping image pairs near crossings). Iterations are repeated until no new crossovers are detected. The number of iterations depends on the quality of the data, but normally three to four iterations is enough.

6.7. Blending of the Photomosaic

The globally aligned mosaic must be blended to compensate for nonuniform illumination, caustics (from reflections when close to the surface), blurring, and suspended particles and scattering, especially in turbid waters, which is the case in most lakes and dams. For this reason, we render the optimized mosaic by projecting every individual image onto the mosaic plane. The rendering of every pixel is decided as a function that penalizes the distance from that pixel to the center of the image, because central pixels are normally better exposed than those on the border of the image. Finally, this composite mosaic is transferred to the gradient domain, where we compensate for the gradients at the image seams, and a least-squares solution is found by solving a discrete Poisson equation, going back to the spatial domain to obtain the seamless blended mosaic. More details about an algorithm specifically adapted to blend large mosaics can be found in Prados (2007).

7. EXPERIMENTS

The validation experiments for the proposed system were performed during February 2009 in the Pasteral hydroelectric dam (Ter River) in Girona (Spain). Figure 10 shows the experimental setup. The surface buoy, described in Section 2.2, was moored approximately 8 m away from the wall with the transceiver tilted at an angle of 45 deg. A supervisory computer was located on top of the dam wall, connected to the buoy through an umbilical cable including three serial lines (one for the DGPS, one for the USBL, and another for the MRU). Although the robot was connected to the supervisory computer through an umbilical cable providing power and an Ethernet link, it was operated in a totally autonomous mode. The umbilical allowed for long experiments without worry about battery time. The robot internal variables including the video images were monitored through an Ethernet connection. Owing to operational constraints accorded with the dam owners, and because the plant was generating power, during the field experiments the surveyed area was limited to a small area located at the left of the penstock pipe–protecting fence, far from the stream generated by its water inlet. Nevertheless, the area was considered wide enough (about 15 m) for these proof-of-concept experiments. The robot was programmed using MCL (see Algorithm 1) to follow a survey trajectory in the face of the dam wall. During the experiment, the robot used only the onboard navigation system based on a DVL and a MRU–FOG and the data gathered from the moored buoy (USBL, DGPS, and MRU) were processed and stored on the supervisory computer and later used offline to reestimate the trajectory for the global alignment and georeferencing of the image sequence.

Figure 11 represents the trajectory executed during the survey mission. The trajectory consisted of a series of vertical movements alternated with lateral displacements while

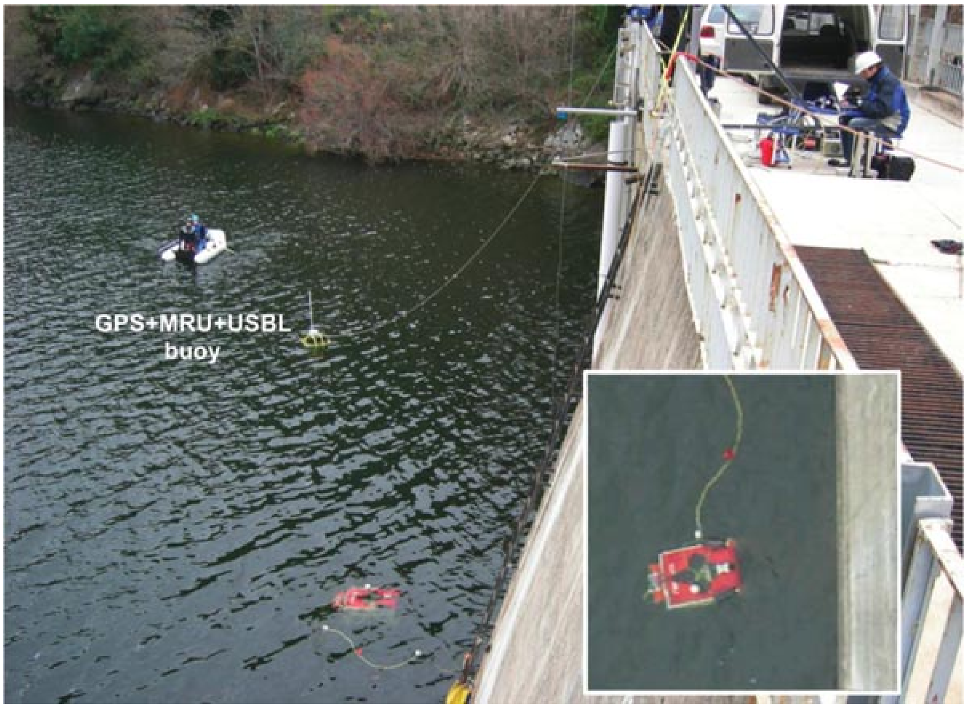


Figure 10. Experimental setup for the wall dam inspection task.

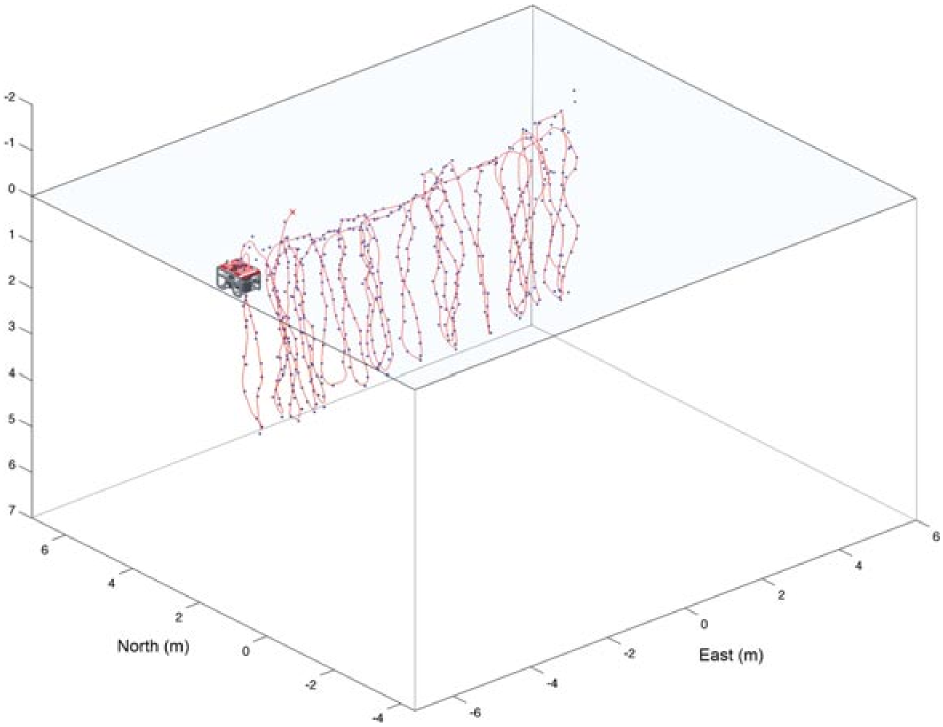


Figure 11. First trajectory realized by the AUV corresponding to a 191-m-length path covering an area of 40 m^2 in approximately 29 min.

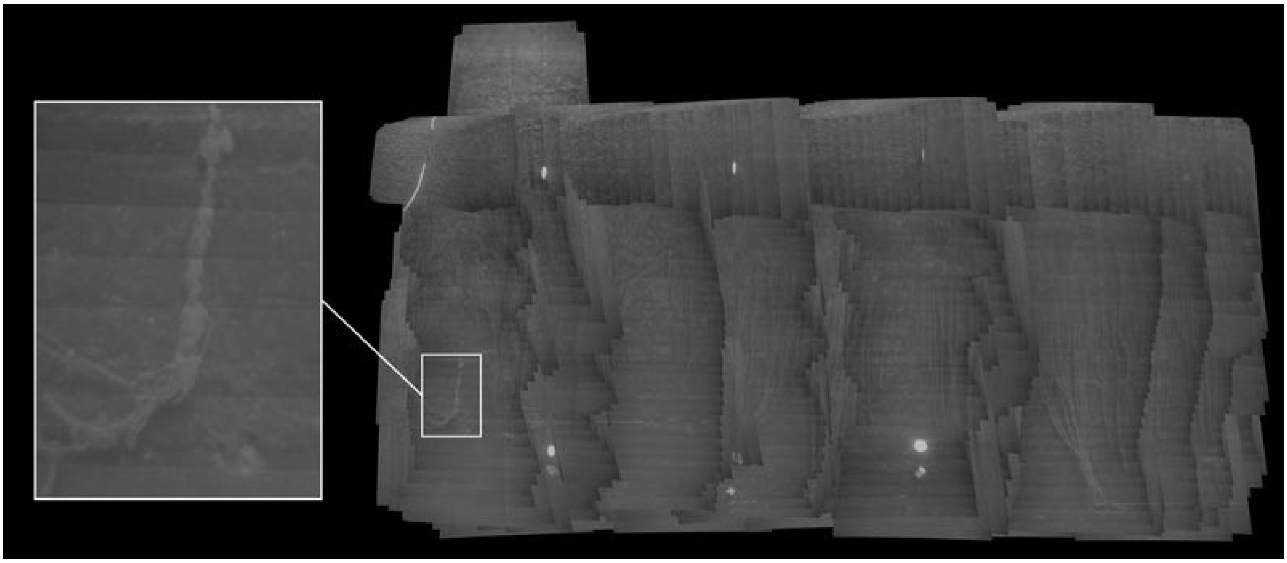


Figure 12. Globally aligned image mosaic corresponding to the trajectory shown in Figure 11.

maintaining the vehicle perpendicular to the dam's wall and at a constant distance. To avoid holes in the final mosaic and to ensure covering 100% of the surveyed area with redundant visual data, the vehicle scanned the area from left to right and then again from right to left back to the starting position. Revisiting previously surveyed areas and ensuring a good overlapping between adjacent images (always greater than 50%) improves the precision and quality of the final mosaic. The total mission time to execute a 40-m^2 survey was 29 min. The trajectory (solid line) in the figure was obtained with the filter described in Section 4, incorporating the surface buoy position measurements (dots) as explained in Section 4.4 and smoothed as reported in Section 4.5. It is worth mentioning that although the represented data have their origin placed over the surface buoy, the resulting trajectory was obtained in world coordinates. During the execution of the filter, a few surface buoy measurements were discarded after being considered as outliers by the individual compatibility test. Some of them can be easily spotted on the upper-right corner of the trajectory as isolated points.

The position data shown in Figure 11 were used together with all the captured images to build the mosaic presented in Figure 12. The mosaic is a high-resolution image, more than 67 Mpx (approx. 1 pixel/ml), including a total of 1,998 images. In Figure 12, the seam transitions among the images as well as strong differences in the illumination can be perceived, making visual inspection difficult. After the blending process described in Section 6.7, the resulting image mosaic (Figure 13) becomes a much clearer image in which plenty of algae can be seen on the wall, as well as circular markers that were added to verify the result. Additional results can be observed in Figure 14, which shows the final result corresponding to a second trajectory where, in the bottom-left corner, rocks can be seen on the floor of the reservoir.

The distance between each pair of circular markers attached to a rope was known prior to the experiment. Although we had no means to precisely survey the exact final position of the markers hanging on the wall, we can approximate that the local accuracy of the final mosaic is in the order of a few centimeters. With respect to the global position estimate, the error is largely dependent on the precision of the GPS system mounted on the buoy. With a standard DGPS such as the one used during the trials, a precision of about 2 or 3 m can be expected. The precision of the USBL while operating in such short range (its maximum operative range is 1,500 m) should be of a few centimeters and, therefore, insignificant in comparison with the errors coming from the GPS.

8. CONCLUSIONS

This paper has presented the results of a research project that proposes an automated solution to the visual inspection problem of hydroelectric dams. The solution consists of using a small AUV together with a localization buoy to acquire a set of images, navigation data, and other information while the robot is autonomously performing a trajectory in front of the wall of a dam. After this experimental phase, all the information is processed in the lab to generate a high-quality georeferenced photomosaic of the inspected wall.

Results obtained from an experiment conducted in February 2009 at the Pasteral hydroelectric dam demonstrate the feasibility of the system. The results include two mosaics of the dam with sufficient resolution (approx. 1 pixel/mm) to make possible the detection of damage

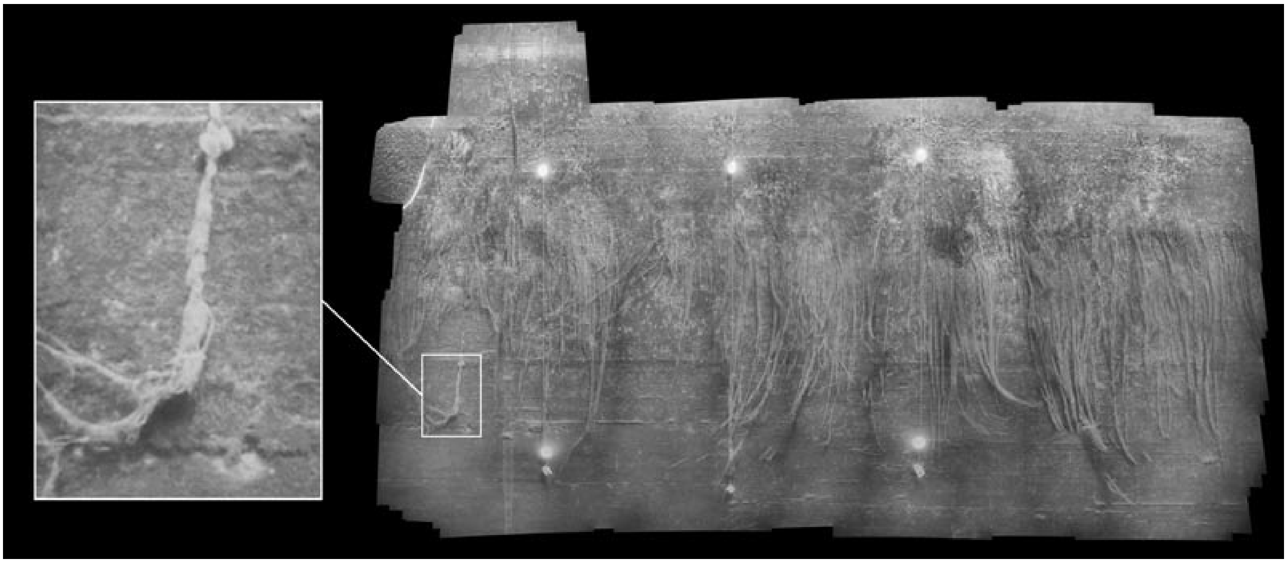


Figure 13. Blended image mosaic corresponding to the trajectory shown in Figure 11.

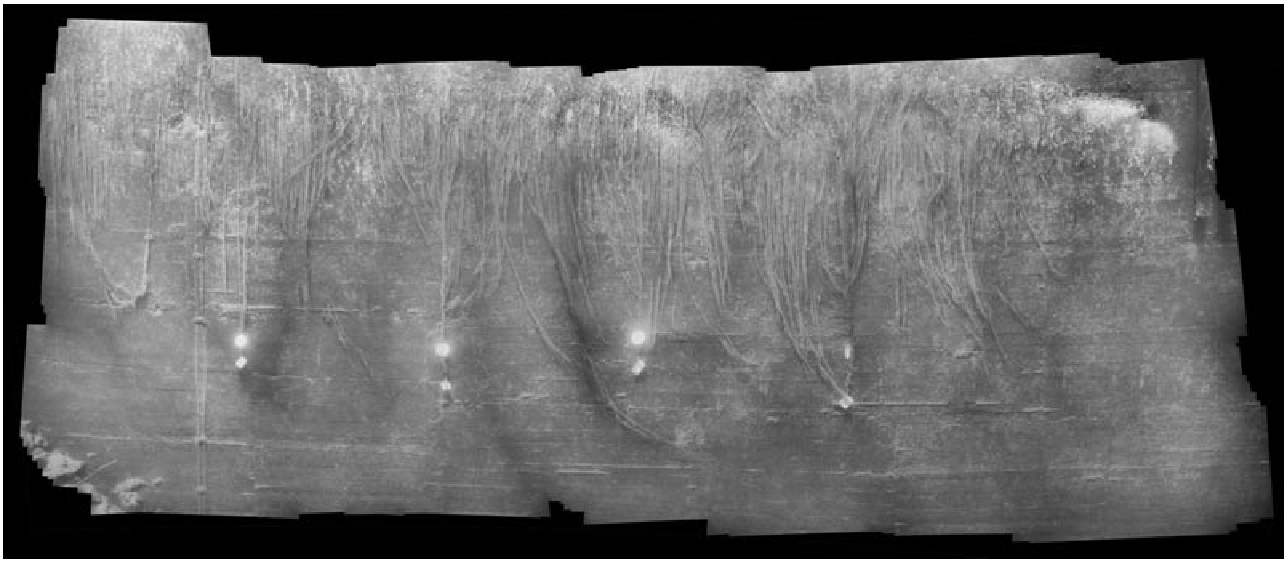


Figure 14. Blended mosaic corresponding to a second trajectory.

in the concrete wall. Unfortunately, because the dam was generating power, it was not possible to perform a survey of the protecting fence at the water inlet to the penstock gallery. However, this scenario has enough similarities with the demonstrated mission that we are confident in the adequacy of the proposed framework as long as the fence is massive enough to be detected with the sonar.

In the presented experiments, the surveyed area was limited to a small portion of the wall (40 m^2). Previous works (Ferrer, Elibol, Delaunoy, Gracias, & García, 2007) demonstrate the capacity of the proposed mosaicking system to deal with very large visual maps (about 1 km^2). Therefore, it should be reasonable to assume that the current system will be capable of generating the mosaic of a complete dam wall.

The resulting visual map offers several advantages with respect to the traditional diver/ROV inspection approach. First, the inspection process is systematic and ensures 100% coverage of the surveyed area. Second, the use of an AUV to perform tasks in such a dangerous environment avoids putting human lives at risk. Also, the resulting map offers a global view of the area, which simplifies the interpretation of the scene. Finally, the georeferencing of the map simplifies the referencing of future surveys and makes possible determining the exact location where repair work should take place.

Future work should focus on testing the performance of the overall system with the objective of determining, and then improving, the precision of the resulting mosaics. Improvements in time required to carry out the experiments, in the robustness of the systems, and in the automation of the offline processes also point to the economical feasibility of the method for high-quality visual dam inspection.

APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

The video is available as Supporting Information in the online version of this article.

Extension	Media type	Description
1	Video	February 2009 experiments in the Pasteral hydroelectric dam

ACKNOWLEDGMENTS

The authors wish to thank the Spanish Ministry for funding project DPI2005-09001-C03-01 and also the workers of Endesa from the Pasteral hydroelectric plant for their collaboration in the experiments. The project was carried out by members (staff and students) of the Vicorob research group of the University of Girona.

REFERENCES

Antich, J., Ortiz, A., & Oliver, G. (2005). A PFM-based control architecture for a visually guided underwater cable tracker to achieve navigation in troublesome scenarios. Journal of Maritime Research, 2(1), 33–50.

Arkin, R. C. (1998). Behavior-based robotics. Cambridge, MA: MIT Press.

Balasuriya, A., & Ura, T. (2002, October). Vision based underwater cable detection and following using AUVs. In Proceedings of the Oceans MTS/IEEE, Biloxi, MS (pp. 1582–1587).

Batlle, J., Nicosevici, T., García, R., & Carreras, M. (2003, September). ROV-aided dam inspection: Practical results. In Proceedings of the 6th IFAC Conference on Manoeuvring and Control of Marine Crafts, Girona, Spain (pp. 309–312).

Bay, H., Tuytelaars, T., & Gool, L. V. (2006, May). SURF: Speeded up robust features. In European Conference on Computer Vision, Graz, Austria (pp. 404–417).

Bouguet, J. Y. (2010). Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html. Accessed 20 May 2010.

Brokloff, N. A. (1994, September). Matrix algorithm for Doppler sonar navigation. In Proceedings of the Oceans MTS/IEEE, Brest, France (vol. 3, pp. 378–383).

Brown, M., & Lowe, D. G. (2003, October). Recognising panoramas. In International Conference on Computer Vision, Nice, France (pp. 1218–1225).

Capel, D. P. (2004). Image mosaicing and super-resolution. Heidelberg, Germany: Springer Verlag.

Castellanos, J. A., & Tardós, J. D. (1999). Mobile robot localization and map building: A multisensor fusion approach. Boston, MA: Kluwer Academic Publishers.

Cote, J., & Lavallee, J. (1995, December). Augmented reality graphic interface for upstream dam inspection. In Proceedings of SPIE Telemanipulator and Telepresence Technologies II, Philadelphia, PA (vol. 2590, pp. 33–39).

Djapic, V. (2009). Unifying behavior based control design and hybrid stability theory for AUV application. Ph.D. thesis, UC Riverside.

Englot, B., & Hover, F. (2009, October). Stability and robustness analysis tools for marine robot localization and SLAM applications. In IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO (pp. 4426–4432).

Escartín, J., Garcia, R., Delaunoy, O., Ferrer, J., Gracias, N., Elibol, A., Cufí, X., Neumann, L., Fornari, D. J., Humphris, S., & Renard, J. (2008). Globally aligned photomosaic of the Lucky Strike hydrothermal vent field (Mid-Atlantic Ridge, 37°18.50N): Release of georeferenced data, mosaic construction, and viewing software. Geochemistry Geophysics Geosystems, 9(12), 1–17.

Eustice, R., Singh, H., Leonard, J., Walter, M., & Ballard, R. (2005, June). Visually navigating the RMS Titanic with SLAM information filters. In Proceedings of Robotics Science and Systems, Cambridge, MA.

Ferrer, J., Elibol, A., Delaunoy, O., Gracias, N., & García, R. (2007, September–October). Large-area photo-mosaics using global alignment and navigation data. In Proceedings of the Oceans MTS/IEEE, Vancouver, Canada (pp. 1–9).

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395.

Fossen, T. I. (1994). Guidance and control of ocean vehicles. Chichester, UK: John Wiley & Sons, Ltd.

Garcia, R., Batlle, J., Cufí, X., & Amat, J. (2001, May). Positioning an underwater vehicle through image mosaicking. In IEEE International Conference on Robotics and Automation, Seoul, Korea (vol. 3, pp. 2779–2784).

German, C., Yoerger, D., Jakuba, M., Shank, T., Langmuir, C., & Nakamura, K. (2008). Hydrothermal exploration with the autonomous benthic explorer. Deep Sea Research Part I: Oceanographic Research Papers, 55(2), 203–219.

Hagen, P. E., Storkersen, N. J., & Vestgard, K. (1999, December). HUGIN—Use of UUV technology in marine applications. In Proceedings of the Oceans MTS/IEEE, Seattle, WA.

Harris, C. G., & Stephens, M. J. (1988, August–September). A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, UK (pp. 147–151).

Hartley, R., & Zisserman, A. (2004). Multiple view geometry in computer vision. Cambridge, UK: Cambridge University Press.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. Transactions of the ASME, Journal of Basic Engineering, 82(Series D), 35–45.

Kazmi, W., Ridao, P., Ribas, D., & Hernàndez, E. (2009, May). Dam wall detection and tracking using a mechanically scanned imaging sonar. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan (pp. 3595–3600).

Kim, A., & Eustice, R. (2009, October). Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009, St. Louis, MO (pp. 1559–1565).

Leonard, J. J., & Durrant-Whyte, H. F. (1992). Directed sonar sensing for mobile robot navigation. Norwell, MA: Kluwer Academic Publishers.

Leonard, J. J., & Rikoski, R. J. (2001). Incorporation of delayed decision making into stochastic mapping (vol. 271, pp. 533–542). Heidelberg, Germany: Springer Verlag.

Mindell, D., & Bingham, B. (2001, November). New archaeological uses of autonomous underwater vehicles. In Proceedings of the Oceans MTS/IEEE, Honolulu, HI.

Murata, T. (1989). Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 77(4), 541–580.

Negahdaripour, S., & Firoozfam, P. (2006). An ROV stereovision system for ship-hull inspection. IEEE Journal of Oceanic Engineering, 31(3), 551–564.

Nicosevici, T., Gracias, N., Negahdaripour, S., & García, R. (2009). Efficient three-dimensional scene modeling and mosaicing. Journal of Field Robotics, 26(10), 759–788.

Palomeras, N., Ridao, P., Carreras, M., & Silvestre, C. (2008, July). Towards a mission control language for AUVs. In 17th IFAC World Congress, Seoul, Korea (pp. 15028–15033). The International Federation of Automatic Control.

Palomeras, N., Ridao, P., Carreras, M., & Silvestre, C. (2009, October). Using Petri nets to specify and execute missions for autonomous underwater vehicles. In International Conference on Intelligent Robots and Systems, St. Louis, MO (pp. 4439–4444).

Petillot, Y., Reed, S., & Bell, J. (2002, October). Real time AUV pipeline detection and tracking using side scan sonar and multi-beam echo-sounder. In Proceedings of the Oceans MTS/IEEE, Biloxi, MS (vol. 1, pp. 217–222).

Pizarro, O., Eustice, R., & Singh, H. (2004, November). Large area 3D reconstructions from underwater surveys. In MTS/IEEE OCEANS Conference, Kobe, Japan (vol. 2, pp. 678–687).

Pizarro, O., & Singh, H. (2003). Toward large-area mosaicing for underwater scientific applications. IEEE Journal of Oceanic Engineering, 28(4), 651–672.

Poupart, M., Benefice, P., & Plutarque, M. (2001, September). Subacuatic inspections of EDF (Electricite de France) dams. In OCEANS, 2000. MTS/IEEE Conference and Exhibition, Providence, RI (vol. 2, pp. 939–942).

Prados, R. (2007). Image blending techniques and their application in underwater mosaicing. Master's thesis, University of Girona.

Ribas, D., Neira, J., Ridao, P., & Tardós, J. (2006, September). AUV localization in structured underwater environments using an a priori map. In Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Crafts, Lisbon, Portugal.

Ribas, D., Palomer, N., Ridao, P., Carreras, M., & Hernàndez, E. (2007, April). Ictineu AUV wins the first SAUC-E competition. In Proceedings of the IEEE International Conference on Robotics and Automation Rome, Italy (pp. 151–156).

Ribas, D., Ridao, P., & Neira, J. (2010). Underwater SLAM for structured environments using an imaging sonar. Number 65 in Springer Tracts in Advanced Robotics. Heidelberg, Germany: Springer Verlag.

Ribas, D., Ridao, P., Tardós, J., & Neira, J. (2008). Underwater SLAM in man-made structured environments. Journal of Field Robotics, 25(11–12), 898–921.

Ridao, P., Batlle, E., Ribas, D., & Carreras, M. (2004, November). NEPTUNE: A HIL simulator for multiple UUVs. In Proceedings of the Oceans MTS/IEEE, Kobe, Japan (vol. 1, pp. 524–531).

Salamin, E. (1979). Application of quaternions to computation with rotations (Tech. Rep.). Stanford University, Stanford, CA.

Seabotix, Inc. (2010). Seabotix, Inc.—Hydro applications. http://www.seabotix.com/applications/hydro.htm. Accessed 3 April 2010.

Singh, H., Roman, C., Pizarro, O., Eustice, R., & Can, A. (2007). Towards high-resolution imaging from underwater vehicles. International Journal of Robotics Research, 26(1), 55–74.

Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In Autonomous robot vehicles (pp. 167–193). New York: Springer-Verlag New York, Inc.

Tao. (2003). TAO developer's guide version 1.3a, volume 2. St. Louis, MO: Object Computing, Inc.

Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. Cambridge, MA: MIT Press.

Triggs, B. (1998, June). Autocalibration from planar scenes. In Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany (vol. 1, pp. 89–105).

Triggs, B., McLauchlan, P., Hartley, R., & Fitzgibbon, A. (2000). Bundle adjustment—A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski (Eds.), Vision algorithms: Theory and practice, Lecture Notes in Computer Science (pp. 298–375). London: Springer Verlag.

VideoRay LLC. (2010). VideoRay LLC—Hydro inspection. http://www.videoray.com/missions/23. Accessed 3 April 2010.

Vincent, É., & Laganière, R. (2001, June). Detecting planar homographies in an image pair. In IEEE Symposium on Image and Signal Processing and Analysis, Pula, Croatia (pp. 182–187).

Williams, S., & Mahon, I. (2004, April–May). Simultaneous localisation and mapping on the Great Barrier Reef. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA (vol. 2, pp. 1771–1776).