An Autonomous Surface-Aerial Marsupial Robotic Team for Riverine Environmental Monitoring: Benefiting from Coordinated Aerial, Underwater, and Surface Level Perception

Eduardo Pinto, Francisco Marques, Ricardo Mendonça, André Lourenço, Pedro Santana, and José Barata

***Abstract*— This paper presents RIVERWATCH, an autonomous surface-aerial marsupial robotic team for riverine environmental monitoring. The robotic system is composed of an Autonomous Surface Vehicle (ASV) piggybacking a multirotor Unmanned Aerial Vehicle (UAV) with vertical takeoff and landing capabilities. The ASV provides the team with long-range transportation in all-weather conditions, whereas the UAV assures an augmented perception of the environment. The coordinated aerial, underwater, and surface level perception allows the team to assess navigation cost from the near field to the far field, which is key for safe navigation and environmental monitoring data gathering. The robotic system is validated on a set of field trials.**

I. Introduction

Environmental monitoring tasks are key for the maintenance of ecosystems. Among these tasks, the monitoring of riverine environments is one of the most demanding, given the need to simultaneously assess biodiversity and pollution levels in both water (river stream) and land (riverbanks). Moreover, riverine environments are most often extensive and remote, which posit considerable difficulties for human-based field work. Environmental sensor networks [1] and satellite remote sensing techniques [2] have been successful in mitigating some of these limitations. Despite all the advances ensured by sensor networks and remote sensing, these also have their own shortcomings. Sensor networks are unable to generate fine spatiotemporal data gathering and require expensive deployment procedures. Remote sensing generates data often outdated and is unable to execute in-situ water analysis protocols. Both approaches are unable to perform sample return for offline laboratorial analysis.

In-situ water analysis, long range operation, and sample return can be attained with Autonomous Surface Vehicles (ASV), in particular when these have the ability to perform energy harvesting (Dunbabin and Marques [3] review key robotic developments in this context). Nevertheless, the low vantage point above the surface line offered by surface vehicles hampers a proper analysis of the far field, namely, of distant shore-lines, shoals, and riverbanks. This impacts negatively the safe navigation capabilities of the vehicle and its ability to generate ecologically relevant products.

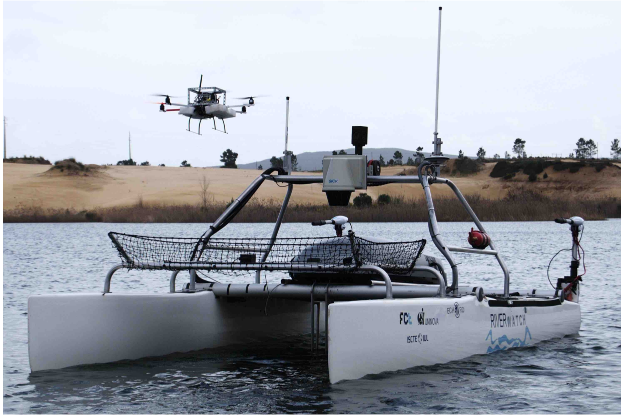


Fig. 1. The ASV-UAV marsupial robotic team. The UAV just took off the ASV in order to engage on an aerial survey for expanding the robotic system's awareness of the environment. As a result of this cooperative perception, the robotic ensemble is capable of setting navigation plans with a lookahead that far exceeds the one available from the ASV's onboard sensors.

An aerial perspective over the environment is known to help ASVs safely navigating in the environment [4], a concept originally applied to ground vehicles [5], [6]. In these cases, the aerial perspective over the environment is obtained prior to the mission with satellites or whatever aerial platform (e.g., airplane). Two limitations of using these data are the cost of its acquisition and their outdating. Up-to-date sensory feedback can be obtained by teaming the surface vehicle with an aerial vehicle [7]. However, attention must be given to the fact that the limitations of each team member may weaken the ensemble. For instance, aerial platforms have considerably lower energetic autonomy and robustness to demanding weather conditions than surface vehicles.

Energetic sufficiency and all-weather operation can be fostered by building a marsupial robotic team. For example, the aerial vehicle can be piggybacked on a surface vehicle [8]. The same concept has been earlier applied to ground vehicles [9]. In this configuration, the surface vehicle ensures energetic autonomy for long-lasting operation and protection for the aerial vehicle in demanding weather conditions. Conversely, the aerial vehicle provides an augmented perception of the environment, which has been exploited for situation awareness of remote human operators [8]. However, the fact that remote environmental monitoring requires fully autonomous robotic operation makes the dependency on a

P. Santana is with ISCTE - Instituto Universitário de Lisboa (ISCTE-IUL), Portugal and Instituto de Telecomunicações (IT), Portugal. E-mail: pedro.santana@iscte.pt

E. Pinto, F. Marques, R. Mendonça, A. Lourenço and J. Barata are with CTS-UNINOVA, Universidade Nova de Lisboa (UNL), Portugal. E-mails: {emp,fam,rmm,afl,jab}@uninova.pt

human operator too restrictive. This constraint has been removed in RIVERWATCH, the autonomous surface-aerial marsupial robotic team presented in this paper. RIVERWATCH is based on a $4.5\,\mathrm{m}$ catamaran-like ASV piggybacking a multi-rotor Unmanned Aerial Vehicle (UAV) with vertical takeoff and landing capabilities (see Fig. 1). Leveraging on a small sized ASV [8] would limit the ability of the system to perform future missions with sample return requirements. This paper extends significantly the system's preliminary overview that can be found elsewhere [10].

Autonomous behaviour in unstructured environments depends heavily on the ability of the system to assess mobility cost from sensory feedback. This is particularly challenging in unstructured environments, as, in these, the mapping between sensory data and mobility cost changes over time and space. The presence of sensory redundancy and the ability to learn new perceptual categories are known assets to deal with this challenge. For instance, cost maps obtained from underwater sonar range data can be registered on aerial images for automatic water/non-water labelling of portions of the aerial image, which, in turn, are used as a training set for online supervised learning of aerial image classifiers [4]. The resulting classifier is capable of segmenting the aerial image into water/land regions and, from that, a long-range surface-level navigation cost map can be produced.

RIVERWATCH employs a scheme similar in spirit to the one proposed by Heidarsson and Sukhatme [4] and goes a step further by benefiting from coordinated aerial, underwater, and surface level sensory feedback, and also by addressing the aerial image acquisition process from a piggybacked aerial vehicle. In a nutshell, the system's main workflow is as follows: (1) the ASV computes local navigation cost maps from its onboard sensors; (2) by exploiting the overlap between both robots' field of views (FOV), the UAV takes off and learns an image classifier by associating its visual input with the ASV's local navigation cost maps; (3) the learnt image classifier is applied to generate, from the UAV's visual input, long-range ASV's navigation cost maps; (4) the UAV docks in the ASV; (5) the ASV proceeds using its now extended perception from the far field (see Fig. 2).

This paper is organised as follows. The robotic system is presented in Section II. Then, the aspects related to safe navigation are described in Section III. Afterwards, the system's environment mapping and understanding processes are presented in Section IV. Finally, the experimental results are analysed in Section V and some conclusions and future work directions are drawn in Section VI.

II. The Robotic System

The ASV is based on a $4.5\,\mathrm{m}$ Nacra catamaran, which has received special carbon fibre reinforcements for the roll bars and motor supports. The hulls have been filled with special PVC closed cells foam, making them virtually unsinkable. High processing capacity is assured by three i7-3770 Ivy Bridge systems, which are water cooled to assure the watertightness of the system. The power distribution, management and fail-safe mechanisms are implemented using one board designed specifically for this application based on a DSPIC33EP256MU810 microcontroler. Power is assured by two banks of 100Ah, 8S LifeYpo4 batteries ($25.6\,\mathrm{V}$) with a BMS from JSC Elektromotus, which interfaces with the control board. Differential propulsion is assured by two fixed Haswing Protuar $2\,\mathrm{hp}$ electric motors, one per hull. This configuration permits a linear velocity of $3\,\mathrm{ms}^{-1}$ and a maximum in-place rotation rate of 0.2 radians per second.

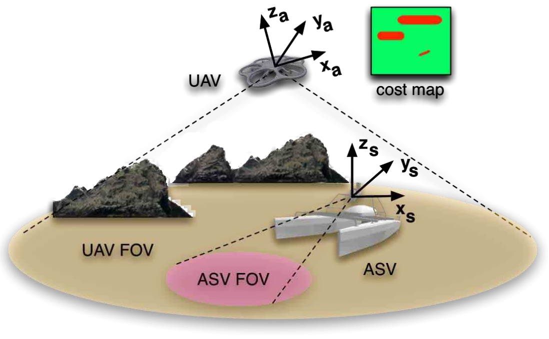


Fig. 2. RIVERWATCH's cooperative perception principle. The UAV provides an high vantage point, as can be seen by the area of the UAV's FOV, in order to increase the ASV's perceptual capabilities. The perceptual maps with different perspectives, from both robots, are then integrated into a final cost map for the ASV's safe navigation.

The UAV uses the VR brain from Network Team as the low level control board (STM32F407 at 192Mhz-ARM Cortex-M4) and an Odroid-U2 from Harkernell (Quad core 1.7Ghz ARM Cortex-A9) for high level processing. The communications are assured using wireless radios from the Ubiquiti Networks airMAX line of products and Xbee Pro modules for the telemetry.

ASV's environment perception is ensured by a long range tilting laser scanner LD-LRS2100, from Sick, a tilted underwater sonar DeltaT 837B, from Imagenex, a multi-camera vision system Ladybug3, from Point Grey, and a Genius WideCam F100 HD upwards looking camera with 120 degrees of field of view fit to the ASVs deck. This camera is used in the cooperative docking procedure as described in Section III-C. For localisation purposes, the ASV is equipped with a $2\,\mathrm{cm}$ horizontal accuracy GPS-RTK Proflex 800 from Ashtec SAS, and an Inertial Measurement Unit (IMU) PhidgetSpatial from Phidgets while the UAV possesses an integrated GPS and IMU in its low-level control board. Both UAV and ASV run their control and navigation systems on the top of the Robotics Operating System (ROS) [11]. Low-level image and point clouds processing is handled by the OpenCV [12] and Point Cloud Library (PCL) [13] libraries, respectively.

In order to piggyback the UAV, a docking station with a 1.0 m x 1.30 m platform and a lateral safety net was fitted to the ASV deck as shown in Fig. 1. While docked the UAV is held in place by the rubberized texture of the docking station. This may be not enough in more adverse weather conditions that lead to prominent ripple in water waves. Therefore, the height and slope of the safety net was designed to prevent the UAV from slipping from the ASV into the water. The safety net also helps to center and secure the UAV into its final docking position, as the nearer the UAV gets to the ASV's deck the more chaotic becomes the airflow created by the UAV's rotors, leading to a drift in its position.

III. Navigation

This section describes the mechanisms present in both ASV and UAV that allow them to move safely in the environment.

A. Motion Control and Pose Estimation

Riverine settings pose a significant challenge to vessel motion control mainly because of water currents and wind. To cope with these disturbances, the ASV controls its differential propulsion with two distinct velocity PID controllers, one for the linear speed and the other for the angular speed. These two speeds are summed to generate control signals for the two motors responsible for the differential thrust. The PID values where obtained by a series of field trials to achieve a stable and agile response. Linear and angular speed sensory feedback are obtained directly from pose estimates obtained from an Extended Kalman Filter fed by the GPS-RTK and IMU devices. These pose estimates are also used by the navigation system described in the next section. The UAV's stabilised motion control is provided directly by its VR brain low-level control system.

B. Motion and Path Planning

The ASV's navigation system aims at determining the best way through the environment to reach a set of GPS waypoints, which can be defined offline by recurring to satellite imagery or online with any exploration strategy. To prevent pursuing unreachable waypoints, e.g., over obstacles, it was empirically established that the current waypoint is assumed as reached within a 10 m radius.

Local motion planning is assured by determining at 10 Hz the best kino-dynamically feasible arc of trajectory and linear speed given an objective function that accounts for clearance to obstacles, closeness to an intermediate goal, and stability in decision making [14]. The intermediate goal is computed at 1 Hz by a path planner that assumes robot linear trajectories [15] (see Fig. 3).

To cope with narrow environments, the motion and path planners need to sample more densely and more faithfully the ASV's decision space. To take this into account, the smaller is the distance to the nearest obstacle the higher are the number of possible trajectories analysed by the motion planner. Similarly, when this distance becomes too short the navigation system switches to a path planner capable of considering both linear and non-linear motion primitives [16]. This context aware adaptation is an extension of a solution originally developed for ground vehicles [17]. The higher computational demands generated by the use of more faithful motion and path planning are compensated with a proportional reduction in navigation speed.

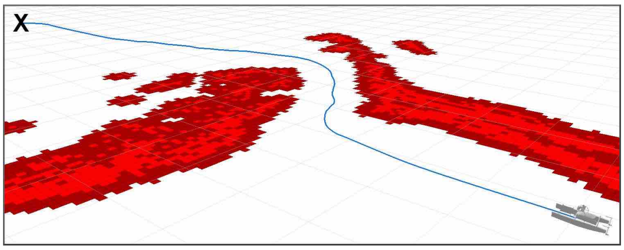


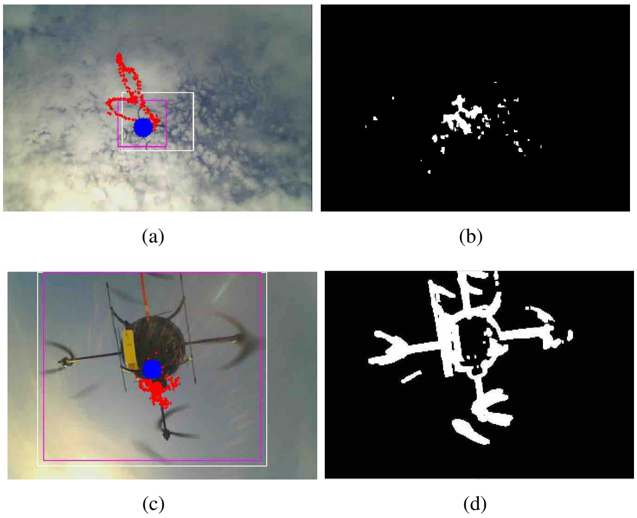
Fig. 3. The safe navigation system planning a path (blue line) that takes the ASV around an unexpected obstacle through a narrow passage so as to reach the waypoint represented by the **X**. The red squares represent the high navigation cost cells, i.e., the obstacles. The cost map is inflated according to the robot's footprint.

The UAV navigation system is limited to simple GPS waypoint following. The absence of airborne obstacles above the river stream renders unneeded sensory-driven obstacle avoidance strategies. Given that the aerial images are acquired with the purpose of extending the ASV's perception to the far field of interest, the UAV flies over the next GPS waypoints that have been set to the ASV. The actual travelled distance depends on the desired ASV's path planning lookahead. Advanced sensor planning strategies can be applied to adapt this behaviour.

C. Cooperative Takeoff and Landing

Safe takeoff from the ASV's docking platform is attained by simply climbing rapidly up to a safe altitude. Conversely, docking is a somewhat more difficult task, which is usually solved by detecting the docking platform from a downwards looking camera on board the UAV (e.g., [18]). However, demanding lighting conditions may render this solution brittle. Moreover, it requires the UAV to be equipped with enough computational power for the purpose. To circumvent these challenges, RIVERWATCH approaches the problem from a multi-robot cooperative perspective. Namely, the ASV is equipped with an upwards looking camera located at the centre of the docking platform whose purpose is to detect and track the UAV throughout its descend. As the background in the images taken by this camera, i.e., the sky, is rather stable, the detection and tracking procedure is fast and robust. The detection and tracking process is only initiated after the UAV roughly aligns with the ASV according to both pose estimates, which are shared via the communications framework.

To detect the UAV in the camera's field of view, the background pixels are labelled using an adaptive Gaussian mixture model [19]. Next, a bounding box is fit to the remaining foreground pixels (see Fig. 4). Then, a Kalman filtered estimation of the UAV's tridimensional position with respect to the camera is obtained from the bounding box's position and dimensions, given the camera's intrinsics, learned from an offline calibration procedure. The estimated tridimensional position of the UAV and the ASV's heading are sent to the UAV so that it is able to determine which corrective measures it must apply in order to centre itself on the upwards facing camera's view. These corrective measures



(a) (b) (c) (d)

Fig. 4. Cooperative docking procedure. The UAV is detected and tracked in different weather conditions and docking stages. First column depicts the upwards looking camera input images depicting the UAV during its descent in two different configurations. The quadcopter configuration can be seen at (a) and (b), as well as the foreground mask in (c) and (d), respectively. Here, the white and purple rectangle mark the instantaneous and filtered UAV's bounding box position, respectively. The blue circle epics the estimated centre of the UAV. The red circles depicts the UAV's path since it was tracked. (c) High altitude approach detection, where despite the background subtraction noise caused by heavy winds and rapidly moving clouds the centre of the UAV is successfully detected. (d) The UAV in the intermediate stage of the descent in a low wind situation as can be perceived by the low dispersion of the UAV's path.

are the output of simple PID controllers linked directly to the UAV's motion control system at 100 Hz.

These corrective measures are managed by a set of decoupled PID controllers, one per controlled degree of freedom - height, latitude, longitude, and heading. These controllers feed the low-level motion control system, i.e., VR Brain, at 100 Hz.

IV. Environment Mapping and Understanding

This section describes the process by which all sensory data is accumulated and processed by the ASV in order to build a bidimensional robot-centric navigation cost map to be used by the navigation system (see Section III). The cost map represents how difficult it is for the ASV to traverse each cell of the environment through the probabilistic integration of a set of partial bidimensional cost maps obtained from different sensory modalities: (1) the ASV's onboard tilting laser scanner; (2) the ASV's onboard tilted underwater sonar; (3) monocular cues present in the ASV's onboard multi-view camera (see [20]); and (4) monocular cues from the UAV's downwards looking camera. The superposition of the several cost maps is possible given the rigid transformation between all sensors and between ASV and UAV. The former is estimated via calibration and the latter by sharing the robots' GPS positions. Details on these maps is provided in the following sections.

A. *Navigation Cost From Range Data*

Range data from laser and sonar are accumulated on two allocentric on-growing octree-like tridimensional structures [21], which allow probabilistic occupancy upkeep of their volumetric elements' states, i.e., unknown, occupied, and free. These structures are gravity-aligned and their occupancy is attained through ray casting every range measurement's beam both the laser and the sonar produce, determining which of the structure's voxels belong to the beam's path and which refer to it's hit point, hence, free and occupied space, respectively. The probabilistic octrees' leaf sizes, given the vehicle's size and the distance covered, were empirically set to $0.3\,\mathrm{m}$. Moreover, different (*free*, *occupied*) insertion log odd tuples were assigned to the octree for the laser scanner, $(0.2, -0.44)$, and to the octree for the sonar, $(0.85, -0.41)$, abstracting a stronger confidence credited to laser readings comparatively to sonar's.

Given the a priori knowledge that deep water rarely reflects in the laser wavelength [22], columns of fully unknown space in the laser's probabilistic octree are assumed to be representative of the water surface. These columns are said to have zero cost when projected to the laser-based bidimensional cost map. Conversely, partially or fully occupied columns are representative of non-water surfaces and are, consequently, said to have unit cost, i.e., impossible to traverse by the ASV. Spurious reflections are naturally filtered out by the probabilistic structure of the octrees. Columns in the sonar's probabilistic octree showing occupied voxels close enough to the water's surface are said to have unit cost in the sonar-based cost map and, otherwise, a decreasing finite cost as the distance to the water surface grows.

B. *On-Land Obstacle Detection*

The ability to find a docking area for the ASV may be vital in severe weather conditions or considerably low-power availability. It is also important that the ASV is capable of delivering any collected samples to a human or a robotic agent on the shore. Such an area is defined as a set of contiguous cells with unit cost in the laser-based cost map (i.e., non-water regions) that are not occupied with obstacles that would hamper the ASV from docking. To detect these obstacles, in a column-wise manner, the lowest occupied cell is taken as the surface bearing in the respective column. Then, all occupied cells from the bearing on, up to the ASV's height, are said to be obstacles. The search is limited to the ASV's height in order to reject overhanging objects, such as tree canopy. Then, the maximum slope between every obstacle cell's centroid in a given column and the surface bearing cell's centroid of every adjacent column is computed. The slopes of all columns are normalised to cover the interval $[0, 1]$ and projected to the surface plane in order to generate a cost map that can be used to plan docking motor plans. That is, the higher the slope the higher the cost.

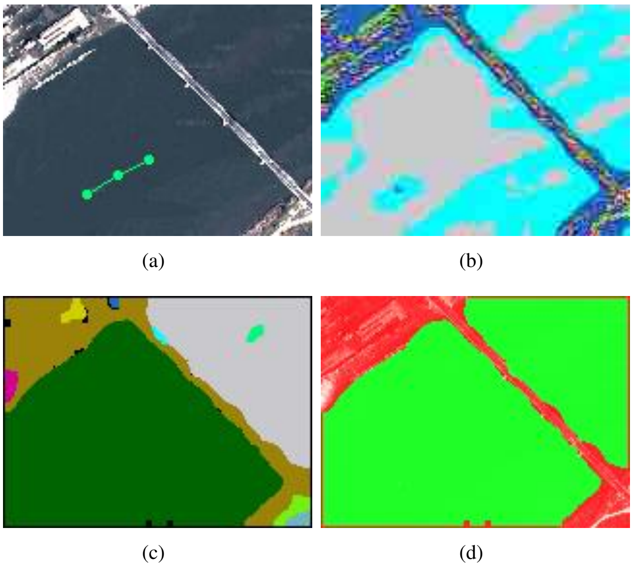
C. *Cooperative Perception*

This section describes the mechanisms required for the UAV to generate a navigation cost map to support ASV's path planning from the near to the far field based on the principles depicted in Fig. 2. Here it is assumed that the UAV performs a meaningful flight in order to maximise the gathering of sensory information bearing the task in mind (see Section III). Aerial images are gathered and sent back to the ASV, which executes a water region segmentation process on them (see below). The resulting segmented images are registered on a common binary bidimensional navigation cost map, given the UAV pose estimates. This cost map signals which cells of the environment are navigable, i.e., are occupied by water, or otherwise. As a result, we get a bidimensional navigation cost map that goes beyond the ASV's field of view and, as a result, pushes farther the planning lookahead.

The water region segmentation of the aerial images starts with an image-wise unsupervised segmentation procedure in order to find super pixels in the images in which the ASV is present. The portion of path taken by the ASV not nearby high cost navigation regions (as determined by the ASV's onboard sensors) is registered in these images, given both ASV's and UAV's pose estimates. The image segments overlapped by the registered path are labelled as water segments, whereas the others are labelled as non-water. The water and non-water segments are then used to create a training set for the supervised learning of a water/non-water image classifier. This classifier is then applied to all images, including those in which the ASV is not present, i.e., which image the far field. The following details these steps.

1) Unsupervised Water Segmentation: The unsupervised water segmentation procedure proceeds as follows. First, feature descriptors are built for all image pixels. These features encompass texture information encoded in 3×3 Law's masks, BGR values acquired from the image's color channels and intensity entropy information. The descriptors are then grouped into 16 feature clusters using K-means. Then, each pixel in the input image is assigned to the representative feature cluster. Subsequently, a set of histograms accounting for the feature cluster assignment frequencies is computed over a sliding window of 7×7 pixels [23]. These histograms are then clustered into 8 histogram clusters using K-means. Again, each pixel in the input image is re-assigned to the representative histogram cluster. This way, local contextual information is used to bring together similar regions. In order to further merge similar segments, a connected components operation is carried out on the re-classified image. Small sized components are removed. Similar components are merged together, according to a threshold of 0.5 in the Bhattacharyya distance between their representative histogram clusters. The product of this process is an over-segmentation of the input image (see Fig. 5(b)).

2) Supervised Water Segmentation: The ASV's trajectory is known to overlap water regions, and, consequently, to overlap segments in the over-segmented input image that represent water. These segments are merged together into a single segment, which is then labelled as the *original* water segment. The next step is to determine which other segments from the over-segmented input image are similar to the original water segment. This matching is done using the Bhattacharyya distance on the segments' descriptors. Then, the matched segments are merged together with the original water segment, resulting in an *expanded* water segment. This process is basically propagating the water label from the segments overlapping the ASV's trajectory to remote unexplored segments (see Fig. 5(d)). As the UAV moves in the environment in order to classify the far field, the supervising signal provided by the ASV's trajectory becomes unavailable. That is, the current ASV's trajectory does not overlap the UAV's visual field. To classify these new images, the system recurs to water/land image classifier learned from images capturing the ASV's executed path. The classifier is implemented as a Support Vector Machine (SVM) with a RBF kernel. The SVM learns the mapping between the image descriptors used in the over-segmentation process and the obtained water/land segmentation. Then, the trained SVM-based classifier is used to classify the input images that the UAV will capture from the far field.


(a)

(b)

(c)

(d)

Fig. 5. Water segmentation pipeline. (a) Input image (satellite imagery) with ASV's trajectory overlaid (green connected dots). (b) Intermediate result from the unsupervised segmentation. (c) Result from the unsupervised segmentation step, in which the green segment corresponds to the *original* water segment. (d) Intermediate result from the supervised segmentation result, in which the green and red overlays represent the *expanded* water segment and land, respectively. This segmentation is used to train a SVM, which is then used to classify the input images that the UAV will capture from the far field.

V. Experimental Results

This section presents the experimental results obtained on a set of field trials. Videos with these and other results can be viewed on the RIVERWATCH project's website[1]. The field trials were carried out in a private lake nearby Sesimbra in Portugal, with an area of roughly $1.5\,\mathrm{km}^2$. This site offered, in a single place, most of the environmental traits that can be found in riverine environments, such as narrow

[1] `http://riverwatchws.cloudapp.net`

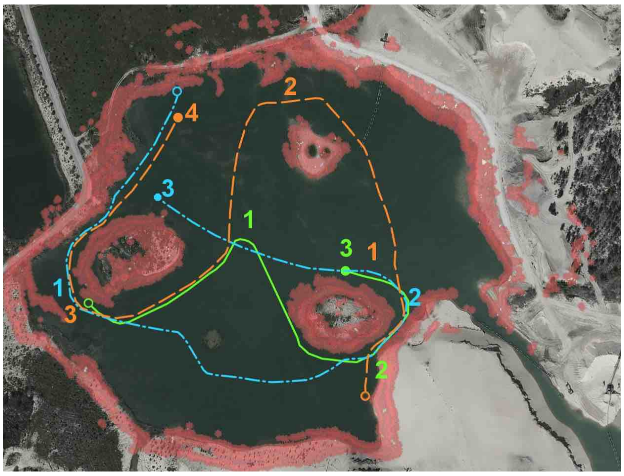
passages, open space areas, deep and shallow waters, shores with disparate kinds of vegetation ranging from sander dunes to large trees passing by zones of extreme vegetation density.

A. Autonomous Navigation

In a first set of experiments, the ASV's navigation system was tested on a windy day. The goal was to verify whether the system is capable of taking the ASV across a set of predefined waypoints in three different runs using the cost map generated online from laser and sonar range data. The missions were defined using a web-based tool developed for the purpose. Here the remote operator establishes the robotic system's waypoints aided by satellite imagery. Although satellite imagery outdating is sufficient reason to force the ASV to avoid unexpected obstacles, this behaviour was purposely exaggerated by selecting waypoints whose connecting line segments cross the centre of the obstacles present in the environment, such as small islands. Fig. 6 depicts the test site's satellite imagery, selected waypoints, the non-traversable labelled regions of the environment by the perceptual mechanisms, and the path taken by the robot in autonomous mode. These paths crossed open waters, near shoals and most shores, which allowed the ASV to build online a thorough and quite accurate range-based representation of the lake (see Fig. 7), essential to safe autonomous navigation. Minor mis-registrations between this environment's representation and the satellite imagery can be explained by vegetation growth, terrain erosion, water level variations, and range data registration errors. Furthermore, overhead imagery suffers from perspective distortion. From Fig. 6, it becomes clear the ASV's ability to properly segment water from land, as well as to produce safe optimised paths. Overall, the ASV travelled 2.1 km at an average speed over ground of $0.8\,\mathrm{ms}^{-1}$, without any collisions, reaching a top speed of $1.4\,\mathrm{ms}^{-1}$. The system was able to robustly track the planned paths, even facing considerable environmental disturbances, like wind.

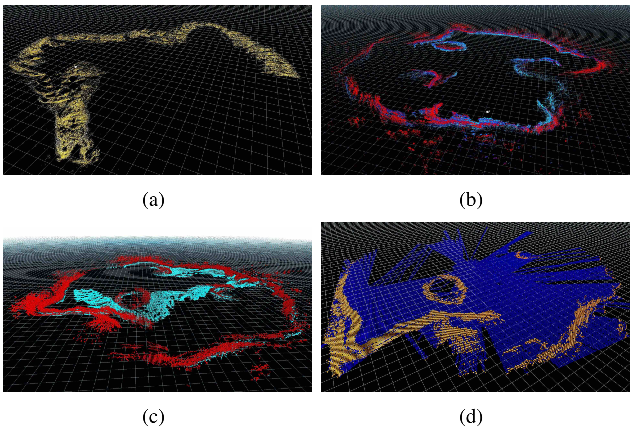
B. Cooperative Perception

To assess the effectiveness of the cooperative perception principle, a final mission was handed to the robotic system. In this mission, the UAV took off the ASV and performed a flight over the desired ASV's path so as to gather enough sensory information to build a navigation cost map including the far field (see Fig. 8). To this end, an obstacle-free (according to the ASV's onboard range sensors) small portion of the ASV's path immediately before the UAV taking off is used to supervise the water/land segmentation process in the first aerial image acquired (see Section IV-C.2). In turn, this segmentation is used by the system to learn an image classifier (see Section IV-C.2) that is applied to a set of 7 subsequently acquired images, evenly spaced across the flight. All these classified images are then composed on a single cost map used by the ASV to plan a path whose lookahead far exceeds its onboard sensors' field of view.



(a)

Fig. 6. Three autonomous navigation runs. Obstacles are represented by the red overlay. The initial and final points of the ASV's path (represented by each line) in each run are represented by an unfilled and a filled circle, respectively. The waypoints followed in each run are represented by the depicted numbers.



(a) (b)

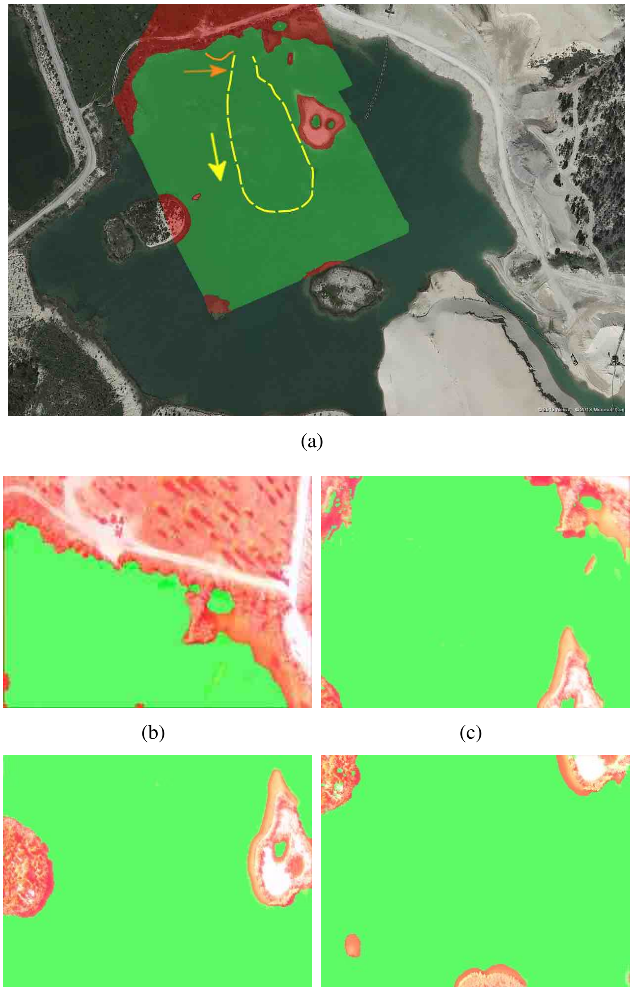
(c) (d)

Fig. 7. Environment range-based representation of environment built throughout one autonomous run. (a) Raw sonar range data (611230 hit points). Depth represented with an yellowish scale. (b) Raw laser range data (867811 hit points). Depth represented with a blue-to-red colour scale. (c) Superposed laser-based (red) and sonar-based (blue) probabilistic octrees. (d) Final bidimensional cost map built from the octrees depicted in (c). Note that the cost map is limited in range given its robot-centric local nature. Blue and orange represent low-cost and high-cost navigation cells, respectively.

TABLE I

THE DOCKING PROCEDURE CHARACTERISTICS OF FIVE RUNS DIVIDED IN SLICES OF 0.5 METERS. WHERE THE MEAN ERROR AND STANDARD DEVIATION OF THE DISTANCE OF THE UAV IN RELATION TO THE CENTRE OF THE DOCKING STATION LANDING PAD IS DEPICTED.

Altitude [m]	Mean Error[m]	Standard Deviation [m]
2 to 1.5	0.4545	0.0319
1.5 to 1.0	0.3114	0.1209
1.0 to 0.5	0.2626	0.1403
0.5 to 0.0	0.0849	0.1101


(a)

(b)

(c)

(d)

(e)

Fig. 8. Cooperative perception results. (a) Composed water/land segmentation (binary navigation cost map) overlaid on satellite imagery. The ASV and UAV navigation paths are represented by solid-orange and dashed-yellow lines, respectively. The arrows indicate the travelling direction. The red overlay denotes land regions whreas the green overlay labels the water regions. (b)-(e) Some of the water/land segmented aerial images acquired by the UAV that have been used to build the composed segmentation depicted in (a).

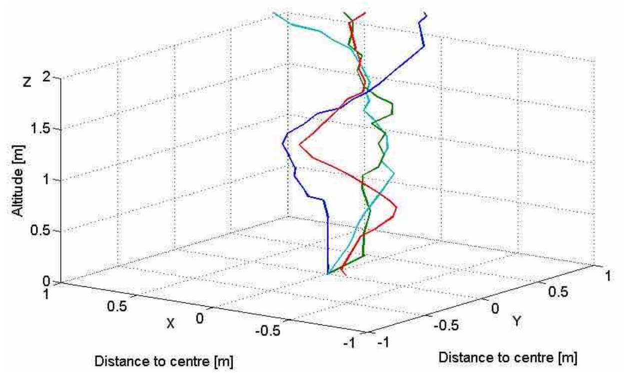


Fig. 9. The UAV's trajectory in relation to the centre of the docking station on the ASV's deck during the different runs, each depicted by a different colour.

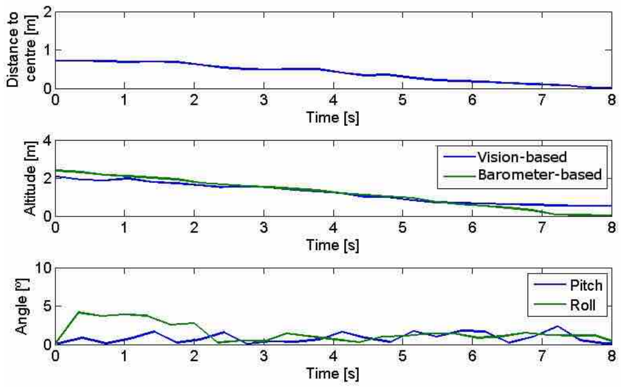


Fig. 10. Field test results position error, altitude and attitude profile observed during a docking procedure.

C. Cooperative Landing

The field experiments designed to assess the robustness of the cooperative landing consisted of four independent runs (see Fig. 9), where the UAV was initially left hovering outside the upwards looking camera field of view. In each experiment the barometer was calibrated to serve as a guideline to assert the accuracy of the estimated height calculated using the bounding box and the focal length. The first step was to autonomously initiate a climb until entering the field of view of the ASV's camera. Subsequently, when the ASV spots the UAV it begins guiding it into the helipad. A statistical analysis of the distance of the UAV to the centre of the helipad during its descent is depicted in Table I. The analysis of the docking procedure was done in 0.5 meters slices starting from 2 meters above the docking station. The mean error displays an expected trend by lowering in proportion to the UAV's altitude. However, the lowest standard deviation is at the highest altitude analysed. This is explained because at that height the movements of the UAV are smaller in relation to the image size. Conversely, the chaotic airflow created by the proximity of the docking station at lower altitudes leads to large displacements in the image.

Fig. 10 shows the last stage of one of the runs. In particular, it shows that landing is successful. However, when in the final stage the closeness to the landing pad entails that the camera's field of view does not encompass the entire UAV. Consequently, the estimated UAV's pose loses reliability. Nonetheless, the information obtained is sufficient for keeping the UAV controlled in this last stage of the landing procedure.

Despite the good results, the proposed system for the cooperative landing alone is insufficient for a robust operation in harsh weather conditions (see Section III-C). Winds and the chaotic airflow nearby the ASV may hamper a precisely centred arrival and departure at the docking station. As a result, the UAV frequently comes close to the safety net, which, due to the limited free space, hampers a safe subsequent takeoff. This is a limitation that can hardly be fully solved with additional control. We envision that the solution for this final step may come from the use of smart mechanics for capturing and centring the UAV in the docking station, as those reported by Mullens et al. [9].

VI. Conclusions

An autonomous surface-aerial marsupial robotic team for riverine environments, RIVERWATCH, was presented. The innovative marsupial solution exploits the high endurance of surface robotic platforms to attain long-lasting operation and the wide field of view of aerial robotic platforms to foster surface-level safe navigation with far field lookahead capabilities. A set of field trials have shown promising results of the proposed system. We are currently working together with experts on environmental monitoring to design a protocol capable of exploiting the developed robotic system on real environmental monitoring missions. Although developed for environmental monitoring, the presented robotic solution can also be useful for other domains, such as infrastructure inspection, search & rescue, and surveillance. As future work we also intend to adapt our previous work on attention-based perception [24], [25] to the RIVERWATCH platform in order to reduce the computational load and, as a result, expand the robotic team's energetic autonomy. Finally, we are working on a smart fixture mechanism for safe docking and takeoff of the aerial platform.

ACKNOWLEDGMENT

This work was co-funded by EU FP7 ECHORD project (grant number 231143) and by the CTS multi-annual funding, through the PIDDAC Program funds. We want to thank Pedro Deusdado from IntRoSys S.A. and Paulo Rodrigues for helping us in the hardware integration phase.

References

[1] P. W. Rundel, E. A. Graham, M. F. Allen, J. C. Fisher, and T. C. Harmon, "Environmental sensor networks in ecological research," *New Phytologist*, vol. 182, no. 3, pp. 589–607, 2009.

[2] L. C. Smith, "Satellite remote sensing of river inundation area, stage, and discharge: A review," *Hydrological processes*, vol. 11, no. 10, pp. 1427–1439, 1997.

[3] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 1, pp. 24–39, 2012.

[4] H. K. Heidarsson and G. S. Sukhatme, "Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3160–3165.

[5] N. Vandapel, R. R. Donamukkala, and M. Hebert, "Unmanned ground vehicle navigation using aerial ladar data," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 31–51, 2006.

[6] D. Silver, B. Sofman, N. Vandapel, J. A. Bagnell, and A. Stentz, "Experimental analysis of overhead data processing to support long range navigation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2006, pp. 2443–2450.

[7] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, *et al.*, "Toward reliable off road autonomous vehicles operating in challenging environments," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449–483, 2006.

[8] M. Lindemuth, R. Murphy, E. Steimle, W. Armitage, K. Dreger, T. Elliot, M. Hall, D. Kalyadin, J. Kramer, M. Palankar, *et al.*, "Sea robot-assisted inspection," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 2, pp. 96–107, 2011.

[9] K. D. Mullens, E. B. Pacis, S. B. Stancliff, A. B. Burmeister, and T. A. Denewiler, "An automated uav mission system," DTIC Document, Tech. Rep., 2003.

[10] E. Pinto, P. Santana, F. Marques, R. Mendonça, A. Lourenço, and J. Barata, "On the design of a robotic system composed of an unmanned surface vehicle and a piggybacked vtol," in *Technological Innovation for Collective Awareness Systems*. Springer, 2014, pp. 193–200.

[11] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the ICRA Open-Source Software Workshop*, 2009.

[12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., Sebastopol, CA, 2008.

[13] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 1–4.

[14] B. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *Proceedings of 2008 Workshop on Path Planning on Costmaps (ICRA)*, 2008.

[15] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[16] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[17] F. Marques, P. Santana, M. Guedes, E. Pinto, A. Lourenço, and J. Barata, "Online self-reconfigurable robot navigation in heterogeneous environments," in *Proc. of the IEEE International Symposium on Industrial Electronics (ISIE)*. IEEE, 2013, pp. 1–6.

[18] F. Kendoul and K. Nonami, "A visual navigation system for autonomous flight of micro air vehicles," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2009, pp. 3888–3893.

[19] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Proc. of the Intl. Conf. on Pattern Recognition (ICPR)*, vol. 2. IEEE, 2004, pp. 28–31.

[20] P. Santana, R. Mendonça, and J. Barata, "Water detection with segmentation guided dynamic texture recognition," in *Proc. of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2012, pp. 1836–1841.

[21] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.

[22] L. Elkins, D. Sellers, and W. Monach, "The autonomous maritime navigation (amn) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles," *Journal of Field Robotics*, vol. 27, no. 6, pp. 790–818, 2010.

[23] M. Blas, M. Agrawal, K. Konolige, and A. Sundaresan, "Fast color/texture segmentation for outdoor robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, Piscataway, 2008, pp. 4078–4085.

[24] P. Santana, M. Guedes, L. Correia, and J. Barata, "Stereo-based all-terrain obstacle detection using visual saliency," *Journal of Field Robotics*, vol. 28, no. 2, pp. 241–263, 2011.

[25] P. Santana, L. Correia, R. Mendonça, N. Alves, and J. Barata, "Tracking natural trails with swarm-based visual saliency," *Journal of Field Robotics*, vol. 30, no. 1, pp. 64–86, 2013.