Proceedings of the 2011 IEEE
International Conference on Mechatronics
April 13-15, 2011, Istanbul, Turkey

Reactive Path Planning for Autonomous Sailboat using an Omni-Directional Camera for Obstacle Detection

Yan Guo, Miguel Romero, Sio-Hoi Ieng, Frédéric Plumet, Ryad Benosman, Bruno Gas
Institut des Systèmes Intelligents et Robotique, UPMC Univ. Paris 06,
UMR 7222, F-75005, Paris France
firstname.lastname@upmc.fr

Abstract— **Unmanned autonomous vehicles have the potential to operate for extended periods of time in different environments like in the air or the sea. These environments are often complex and arise many challenges for the unmanned vehicles research. The ASAROME project**[1] **(Autonomous SAiling Robot for Oceanographic MEasurements) is focused on an autonomous sailboat to make measurements and observations in the marine environment for long term.**

This paper describes a routing strategy for obstacle avoidance using an omnidirectional camera based obstacle detection. Experiments with a panoramic vision system and sailboat simulation results have shown the expected performances for obstacle detection and avoidance.

I. INTRODUCTION

Operating unmanned autonomous vehicle is a challenging task which is far more difficult than classic mobile autonomous robots because of the outdoor context of application. Outdoor environments are hazardous for their unpredictable characters and any physical object is a potential threat or an obstacle to the vehicle that can prevent it to reach its goal. Most of the autonomous outdoor navigation works have been done for ground and air vehicles but unmanned surface vehicules (USV) have also gathered attention of robotics researchers like [1], [2], [3], [4], [5], [6], [7] for autonomous sailboat. It is usual to equip ground or aerials vehicle with inertial and vision sensors but in the case of surface ones it is more marginal to our knowledge [8]. Active sensors like sonar, radar or even laser are more likely used in surface vehicule for obstacle detection [9], [10], [11].

The ASAROME (Autonomous SAiling Robot for Oceanographic MEasurements) project fits into this context of autonomous USV, aimed to the design and construction of an autonomous vessel able to carry out long term oceanographic measurement campaign. Because of the long term constraint, the vessel cannot be maintained or repaired before a long time, thus an accurate and efficient strategy of obstacle detection and navigation is fundamental to preserve the integrity of the ship. Multisensory approaches are usually the solution to the problem as it is shown in several past works [12], [13], [14].

In this project, several on-board sensors (panoramic camera, sonar, hydrophones) will be used for obstacle detection (boats, drifting bodies). Multisensory systems are meant to integrate data provided by each sensor to build a complete representation of the boat environment, but because of its complexity of use, we will first focus on the omnidirectional camera system to build a vision based obstacle detector. Future work will encompass other embedded sensors. The route determination method that we use is conceived to optimize the trajectory based on the absolute wind, looking for a favorable wind propulsion through the entire route between waypoints to avoid unnecessary tack maneuvers and to avoid obstacles detected by the panoramic camera.

The paper is organized in several sections. Section II describes the system's architectures. In section III, we present the calibration method for a catadioptric camera and its use for obstacle detection. In section IV, we introduce a simple but efficient method for local routing and obstacle avoidance. First experimental and simulation results are presented in the last section of this paper.

II. SYSTEM ARCHITECTURE

The USV used for the ASAROME project is a 3.5m long sailboat. It's command system is made of sensing and decision blocks as shown in Fig. 1. We have three blocks of sensors that deliver data to the heading and sail trimming calculator and to the data fusion layer.

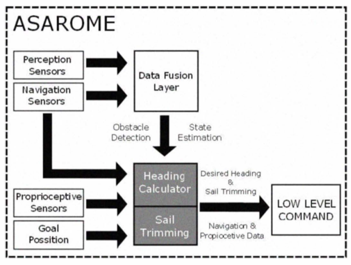

Fig. 1: ASAROME's Command Layer Architecture.

The Navigation Sensors Block is a set of IMU and GPS (Xsens MTi-G), anemometer and wind vane (Furuno CV3F) sensors. The Perception block includes sonar (Tritech Micron DST), hydrophones (Aquarian H2A) and a panoramic camera (Prosilica IP CameraIS GE1350C). Finally, we can find the boom's angle (Hermetic 24bit Baumer IVO Absolute Encoder), and rudder angle (LINAK LA12 300N IP 65) sensors in the Proprioceptive set.

1. This work is funded under the project ANR ASAROME (Num. ANR-07-ROBO-0009)

The data fusion layer takes the information from the Navigation and Perception blocks to localize the obstacles and compute their position relative to the sailboat.

The Heading Calculator and Sail Trimming block includes the decisional algorithms which take into account intrinsic sailboat's characteristics (i.e. its polar diagram) to compute the best heading and sail trimming. Those values are sent to the low level command block. This block computes the corresponding boom and rudder desired angles and passes them to the low level servo control loop. The rudder is actuated by a LINAK (LAR 300N, IP65) linear actuator and the boom by an electric fishing reels (Kristal fishing XL630).

The sailboat has three on-board computers in order to optimize the data acquisition and processing load.

Since the perception process implies intensive use of the processors, we use a dedicated ARBOR mini ITX PC Box with an embedded ATOM N270 Processor. This processor board is directly linked with the IP Camera using a GbEth port and with the sonar and hydrophones. It is also used for the data fusion computation. A LIPPERT PC-104 with ATOM N270 is used for routing and high level command computation.

An EMTRION embedded case module with Renesas SH7781@400mhz processor, fast ethernet port and a can bus controller is used to centralize all the information from the navigation and proprioceptive sensors. The data transmission for the IMU-GPS is made over a serial bus and the rest of the sensors over a CAN-bus. The actuator command are also sent by this computer through the CAN-Bus. Then, the communication between computers, to send and retrieve the navigation data as well as the desired heading and sail trimming, is made by UDP over an ethernet network.

Finally, the power supply is based on a battery pack (12V, 60A), which is charged by a windmill (ATMB 1000 50W-12V) and a $0.5m^2$ solar panel (Sunset AS60, 60W-17.9V).

III. Omnidirectional vision sensor

The obstacle detection is one of the most important tasks to fulfill to ensure the safety of the boat and the possibility to define the most adequate manoeuver commands. These tasks are possible only if the handling of the sensors information are highly accurate. ASAROME embedded several sensors to provide this information and for this paper we are focusing on the vision sensor to address the problems of obstacles detection and avoidance. Sensors data are used to feed the navigation command loops but they need firstly to be calibrated correctly, i.e., the mappings between measurements and the real world metric must be estimated and all sensors must be related to each other.

A. Catadioptric sensor calibration

An omnidirectional vision sensor is especially adapted for navigation and obstacle detection purposes. The omnidirectional sensor we use for these tasks is a catadioptric system, combining a perspective camera with a reflective surface. The system is non-central to avoid any physical placement constraint but to be able to use it properly, a geometric calibration is mandatory. This operation provides the mapping between a pixel to some scene geometric entity (3D points,lines,etc.). Roughly, we have to estimate the relative poses of each components of the sensor : the perspective camera, the mirror and the transparent acrylic tube used as the casing of the whole structure. The perspective camera must be calibrated intrinsically using standard techniques described in literature; typically, we used [15] to achieve this task. With the knowledge of the intrinsic the matrix K, we estimate the relative pose of the camera and the mirror with a variation of the method described in [16], based on an homography between two judiciously chosen planes according to [17]. These planes are the image plane and the one defined by the upper edge of the mirror. That method makes the calibration always possible because of the permanent visibility of the mirror edge (see Fig. 2).

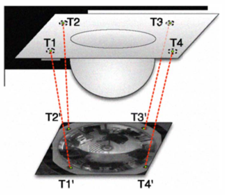

Fig. 2: The upper edge of the mirror is used as calibration pattern to localize the intrinsically-calibrated camera.

To complete the calibration, the acrylic tube, coaxially aligned with the mirror, is taken into account for inducing the refraction of the light when rays pass through the successive interfaces air-tube-air. With that step, the calibration is complete and one is able to map any pixel $\mathbf{p}_i$ to a 3D line going through the point $\mathbf{m}_i$ on the mirror, with the direction $\mathbf{u}_i$ (see Fig. 3).

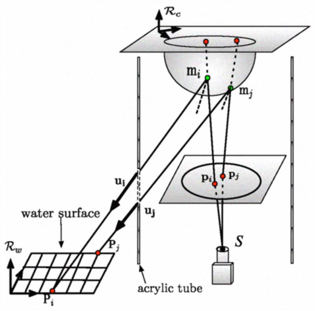

Fig. 3: A complete calibration of the catadioptric sensor provides the mapping of any image pixels $\mathbf{p}_i$ to a 3D line $(\mathbf{m}_i, \mathbf{u}_i)$ and its intersection with the surface defined by the water. $\mathbf{P}_i$ can be computed if the relative pose is known.

B. Boat pose estimation

A global coordinate frame has to be set at first to localize all sensors in relationship with the sailboat, the chosen frame is the one defined by the axes of the ship, taken as an ellipse. In the case of the camera, the ship position is determined from the images. The boat is first segmented from the background and the covariance matrix of the pixels distribution is computed. Principals axes directions of the ellipse are then given by the matrix eigenvectors (see Fig. 4). Any structure or scene object will now be referenced within this frame for the rest of the paper.

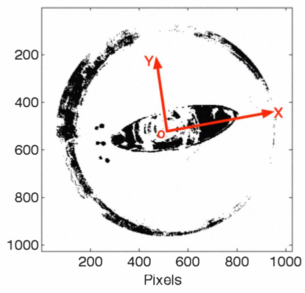

Fig. 4: The ship's axes are used to define a global coordinate frame assuming the camera is placed perpendicular to it.

With the sensor calibrated as described and placed on the the top of the mast, it is possible to compute each pixel's projection on the plane defined by the water under the assumption of a calm sea state, if the relative pose of this plane and the vision sensor (thus the boat) can be estimated. This operation is required if we want to extract metric information from the images without the need of stereovision. In the similar way, as we did for the camera pose with respect to the mirror estimation, we can estimate the omnidirectional sensor pose by detecting projections of the sea plane on the catadioptric image according to [18]. To achieve this task we place into the water several easy-to-detect buoys that are acting as seamarks to underline the sea plane (see Fig. 5).

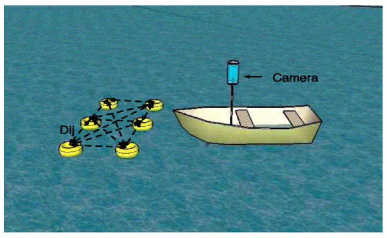

Fig. 5: Seamark used for the pose estimation.

If the structure defined by the buoys is known, i.e., their relative distances of the buoys are known, then enough geometric constraints can be provided in addition to the coplanarity one for the plane estimation. A standard parametrization of a plane we can use is :

$$ax + by + cz + d = 0 \tag{1}$$

Given n buoys used to built the seamark, the distance between the i^{th} and the j^{th} buoys, D_{ij} is manually measured. There are $m = \binom{n}{2} = \frac{n!}{2!(n-2)!}$ of such distances and m is higher than the number of parameters for $n \geq 4$ (in our case $n = 6$). The buoys projected on the camera are also reduced to their centroids $\mathbf{p}_i$ in the image plane and, with the sensor calibration, n 3D lines $(\mathbf{m}_i, \mathbf{u}_i)$ are computed for the n centroids.

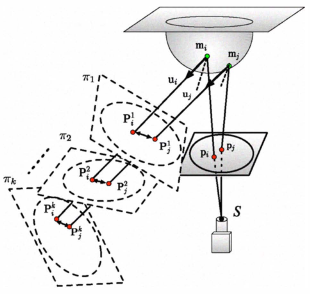

Fig. 6: Fitting plane on the calibrated sensor.

Without proving it directly, with a non-central sensor, there is likely only one plane under the distances constraints that intersects a given bundle of rays which in our case is the set of all rays associated to the buoys. We use a Hough-based method for the plane decision. Where a, b, c are precomputed from -1 to 1 with a constant step of 0.002 and d is calculed from 1 to 500 with a constant step of 1. We define π_k, the k^{th} plane that intersects the ray bundle. Each ray intersects the plane and the distances between every two intersections are computed :

$$D_{ij}^k = |\mathbf{P}_i^k - \mathbf{P}_j^k| \tag{2}$$

Where $\mathbf{P}_i^k$ and $\mathbf{P}_j^k$ are the positions of intersections for the point i and j. If π_k is the plane that we are looking for, it will satisfy the condition $D_{ij} \approx D_{ij}^k$ (check Fig. 6 for a general illustration). With the possibility to determine the camera pose with respect to the sea surface, we can map any pixel $\mathbf{p}_i$ to a 3D point $\mathbf{P}_i$ on the plane (see Fig. 3).

The distance of each pixel to the boat in the image can be computed as : $d = |\mathbf{p}_i|$ in the global coordinate frame introduced earlier. This latter result allows to built a resolution map of the sensor on the sea surface since we can project all the pixels on it (see Fig. 7).

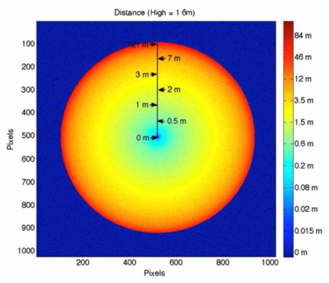

Fig. 7: Distance estimation for each pixel of the image.

The plane pose estimation method is tested with synthetic data by randomly generating 3 planes defined by their four parameters. For each plane, 6 points are placed randomly on it, assuming that their relative distances with respect to each other point are known. For each of this setting, we execute the plane detection algorithm. This operation is repeated 100 times to provide enough results to produce meaningful statistics. The table I summarizes the results obtained.

TABLE I: Result of estimated parameters

	Parameters estimation error rates(%)			
Parameters	a	b	c	d
Case 1	4	1	1	3.3
Case 2	1	0	0	0
Case 3	1	2	0	6.7

According to the result, the plane estimation algorithm accuracy is reasonable with estimates errors below 7% and a mean value of 1.67%.

The resolution map (Fig. 7) is built with the mast normal to the water plane (calm sea state), however an oscillating sea is more likely expected, implying to recomputed the resolution map for different orientations of the ship with respect to the sea surface (see Fig. 8). To avoid to run constantly the costly construction, a set of maps is precomputed for several values of orientations one time. Each time the ship detect a significant change of orientation, the correct map is loaded into the computer. The maps are precomputed for angle α spanning from -90 to $90°$ with a constant step of $1.80°$.

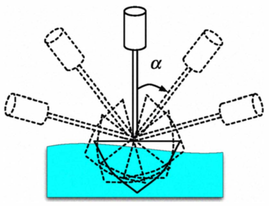

Fig. 8: Oscillation of the boat in a tempestuous sea.

C. *Obstacle detection*

The obstacle detection is done in two step : the first is to perform an image difference between two consecutive images to have a coarse position of moving objects conside-red by default as obstacles. This will underlined areas in the image to process at the second step.

A colorimetric criterion is then used to segment objects in the regions given by the image difference. To be able to detect obstacle, a color signature of the sea is first computed by selecting samples of regions representing the water. Each pixel of these regions are reprojected to the three planes of the RGB coordinate frame according to their color components. This operation produces a cluster in each plane and is representative of the color distribution of the background (i.e sea).

To segment an object pixel from the background, we pro-ject it into the RGB components planes and it is considered being an obstacle if at least one of its RGB component does not belong to the clusters defined earlier. The Fig. 9 show an example of segmentation of a seamark formed by six yellow buoys using the combination of image difference and colorimetric signature of the water.

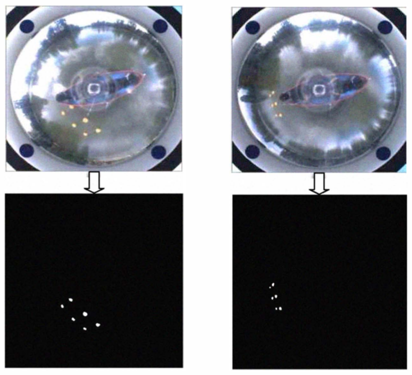

Fig. 9: Seamark detections using colorimetric signature.

Detected obstacles are usually sets of pixels $\mathbf{P}^{i}_{obs}(x_i, y_i)$ from which the centroids $\mathbf{P}_c(x_c, y_c)$ are computed. We can also estimate roughly the size of a detected obstacle as being the maximal distance between two pixels of the set :

$$l_{obs} = max(|\mathbf{P}^{i}_{obs} - \mathbf{P}^{j}_{obs}|) \quad \text{for } i \neq j. \qquad (3)$$

With the obstacles properly detected and segmented, their positions and their respective size are parsed to the heading calculation.

IV. Local Routing Strategy and Obstacle Avoidance

Before presenting the method, let us recall some basics on sailing. For a sailboat, the propulsive force comes from the aerodynamics effects of the wind on the sails. This leads to the conventional points of sail diagram which describes a sailing boat's course with respect to the wind direction (Fig. 10). The white sectors correspond to normal sailing zones and the two shaded zones are not sailable zones or no-go zones (in fact, the downwind zone could be sailed in theory but is not very efficient and rather unstable).

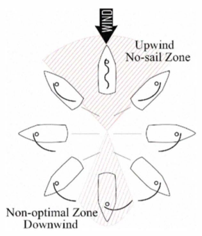

Fig. 10: Standard points of sail diagram

So, unlike conventional motorized robotic vehicles, where a straight line to the goal leads to a shortest path (in time and distance), in this case there is no such easy solution for a sailboat if the target is located directly upwind or downwind. In these cases, the sailboat has to beat (i.e., take a zig-zag course) to reach the goal.

The speed vector of a sailboat depends on many factors like the wind angle, wind speed, sail trimming, currents and waves. This behavior is usually represented for a given boat by a specific polar diagram (Fig. 11). This polar diagram shows the maximum boat speed along a given heading with respect to the wind. Each curve on the polar diagram corresponds to a given wind speed. As we can see on Fig. 11, the wind speed mainly modifies the amplitude without modifying the global shape of the polar diagram.

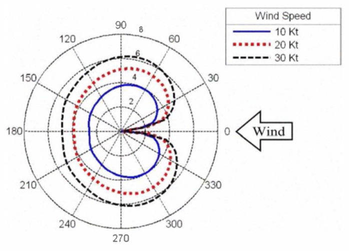

Fig. 11: Boat-specific polar diagram.

In the following, we suppose that this boat-specific polar diagram is known for a set of wind speeds.

The proposed method for local routing and obstacle avoidance is based on the calculation of an optimal heading that minimizes a cost function. This cost function is a combination of two terms : the first one will tend to minimize the time to reach the goal ; the second one will tend to maximize the distance to the obstacles. This method will react to the changing of the environmental conditions (wind speed, wind angles,...) by periodically re-computing the optimal heading.

Without loss of generality, we consider in the following that the behavior of the sailboat is represented by an ideal polar diagram as shown in Fig. 12.

The main objective of our routing algorithm is to minimize the time to go from the current position to the target. As a measure of the boat efficiency to reach the goal, we take the value of the boat speed vector $\mathbf{V}(h)$ for a given heading angle h, projected in the direction of the waypoint WP, that is :

$$V_G = \mathbf{V}(h)^T \cdot \mathbf{T_g} \tag{4}$$

where $\mathbf{T_g}$ is a unit vector pointing to the way point. In order to conveniently use this measure in our routing algorithm, the boat velocity is supposed to be normalized, i.e, the maximum value of the boat speed on a given polar diagram is supposed to be equal to 1, regardless of the wind speed value.

Due to the symmetry of the polar diagram, it may exists two different values of the heading, one to each side of the wind, that lead to the same value for V_G as shown on Fig. 12. This situation occurs when the waypoint is set directly upwind or downwind. Moreover, steering through the eye of the wind, that is, into and across the flow of the wind, is usually a maneuver that must be avoided due to the speed decreasing and, in certain situations, the risks of rollover due to a wind shift. To take these two facts into consideration, we use a penalty factor η_w for the computation of the cost function to minimize the time to reach the goal :

$$C_W = \eta_w (1 - V_G) \tag{5}$$

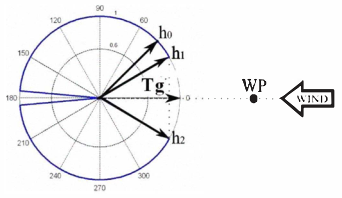

Fig. 12: Two proposed headings situation. Where h_0 is the actual heading, h_1 is the privileged heading and h_2 is the rejected heading.

with η_w = 1 if the actual heading and the new computed heading are in the same side of the wind and η_w = 0.8 otherwise. Such a penalty factor leads to prioritize a new heading that keeps the course in the same side of the wind, rather than crossing the eye of the wind.

To take into account the obstacles detected by the perception layer, we use a cost function based on the measured distance between the boat and an obstacle :

$$C_O = \begin{cases} \eta_o \left(\frac{1}{d_{obs}} - \frac{1}{d_0} \right) & if \;\; d_{obs} \leq d_0 \\ 0 & if \;\; d_{obs} > d_0 \end{cases}$$

with η_o as positive scaling factor, d_{obs} as the Euclidean distance from the center of the sailboat to the center of the obstacle and d_0 as the obstacle influence distance (50 m in our case, which is a trade-off between the range of our sensors and the maneuverability of the sailboat).

The final cost function is :

$$C = C_W + \sum_{obs} C_0 \tag{6}$$

This cost function is computed periodically (0.5 sec) and the optimal heading angle minimizing this cost function is sent to the low level layer control. The minimization of this cost function allows the sailboat to be able to navigate while keeping its course outside of the polar's forbidden area (Fig.10), taking the shortest navigable path and avoiding obstacles. Using this rather simple method, there is in fact no guarantee that the system will not be caught in a local minimum. However, since our autonomous sailboat is intended to move in open sea, the probability to have more than one obstacle at a time in the sensors range is actually very low.

V. EXPERIMENTATION

We demonstrate the performance of our algorithm with two different tests : obstacle detection and simulation for the trajectory planning. The first experimentation have been done on a lake, with all the data being recorded during the test. The second one is a simulation, which shows the resulting motion of the sailboat's with and without obstacle.

A. *Real data analysis for obstacle's detection*

We conducted this experimentation on a calm lake with its surface defining a plane. A special seamark with 6 buoys (1m long x 2m wide) is used for this test. The camera is fixed on a 1.9m high tripod.

The seamark position is changed between each record and the distance between the camera and the center of the seamark was manually measured. The distance tested ranges from 2 meter to 14 meter.

TABLE II: Real data error estimation

	Distance mesured (m)	Estimated Distance (m)	Estimated Angle (°)	Dis. estimation error rates (%)
Case 1	2	1.70	259	3.5
Case 2	4	3.72	201	3.3
Case 3	6	5.60	207	4.7
Case 4	8	7.81	199	2.2
Case 5	10	9.61	186	4.6
Case 6	12	11.44	208	6.5
Case 7	14	13.37	187	7.4

From table II, our algorithm seemed to have excellent performance with error rates within 8%. The distances estimated are smaller than the measured distances. The maximum value of estimated error is 0.63m for a distance of 14m, and the minimum is 0.19m for a distance of 8m.

B. SIMULATION RESULTS

Extensive simulations were conducted of sailboat traveling in different wind directions, with different goal positions and with or without obstacles.

A representative simulation result is presented in this paper. Fig. 13 shows the sailboat's motion for an upwind navigation case. Each test was conducted two times with the same conditions, except for the obstacle. For both simulation, the wind angle (45°) and the the wind speed ($15\ Kt$) remain constant during the simulation time. The initial heading is 0°, the start position is at $(0,0)$ and the goal position at $(100,100)$. Those conditions were chosen to force the sailboat to take an upwind trajectory and, in the second case, to put the obstacle over the preceding path.

We can see on Fig. 13 that, in both case, the sailboat reaches the waypoint. In the first case (without obstacle), the sailboat will tack (cross the eye of the wind) one time and in the second case, it have to tack two more times to avoid the obstacle and reach the goal.

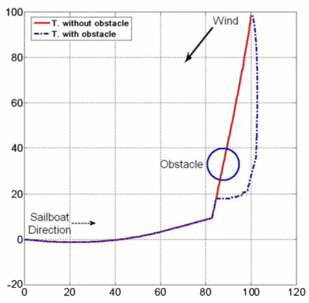

Fig. 13: Simulated motion with and without obstacles.

VI. CONCLUSION

The experimental results demonstrate the ability of panoramic vision system to obtain and process the data to give a precise position and size of the detected obstacles. The conducted simulations validate the routing algorithm because, with the data from omnidirectional camera system, it is possible to modify the motion of the sailboat to avoid the collisions, keep it on the correct wind's angle and reach the fixed waypoint. The current limitation of the presented routing method is the risk to fall in a local minimum. But, as outlined before, in open sea : the probability to have more than one obstacle at a time in the sensors range is actually very low.

The future work in a short term is aimed to perform several tests implying all the sensor for data fusion to validate the whole system in open sea.

REFERENCES

[1] R. Stelzer and T. Pröll, "Autonomous sailboat navigation for short course racing," *Robotics and Autonomous Systems*, vol. 56, pp. 604–614, 2008.

[2] R. Stelzer, K. Jafarmadar, H. Hassler, and R. Charwot, "A reactive approach to obstacle avoidance in autonomous sailing." in *3rd International Robotic Sailing Conference (IRSC)*, 2010.

[3] Y. Briere, "Between the drifting buoy and the autonomous sailing boat : the microtransat concept," in *ATMA - ASSS 2008 international autonomous surface ship symposium*, France, 2008.

[4] C. Sauze and M. Neal, "A raycast approach to collision avoidance in sailing robots," in *Proceedings of the 3rd International Robotic Sailing Conference*, 2010.

[5] G. H. Elkaim, "The atlantis project : A gps-guided wing-sailed autonomous catamaran," *Journal of the Institute of Navigation*, vol. 53, No. 4, pp. 237–247, 2006.

[6] J. Sliwka and L. Jaulin, "Autonomous robotic boat of ensieta." International robotic sailing conference, july 2009, pp. 1–7.

[7] "Avalon, navigation strategy and trajectory following controller for an autonomous sailing vessel," *IEEE Robotics and Automation Magazine*, vol. 17, No 1, pp. 45–54, 2010.

[8] A. Subramanian, X. Gong, J. Riggins, D. Stilwell, and C. Wyatt, "Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications," in *OCEANS 2006*, 2006, pp. 1–6.

[9] T. Bandyophadyay, L. Sarcione, and F. Hover, "A simple reactive obstacle avoidance algorithm and its application in singapore harbor." in *International Conference on Field and Service Robotics*, 2009.

[10] M. Dunbabin, A. Grinham, and J. Udy, "An autonomous surface vehicle for water quality monitoring," in *Australasian Conference on Robotics and Automation (ACRA)*, 2009.

[11] C. Almeida, T. Franco, H. Ferreira, A. Martins, R. Santos, J. Almeida, J. Carvalho, and E. Silva, "Radar based collision detection developments on usv roaz ii," in *OCEANS*, 2009.

[12] A. Bonci, G. Ippoliti, L. Jetto, T. Leo, and S. Longhi, "Methods and algorithms for sensor data fusion aimed at improving the autonomy of a mobile robot," *in Advances in Control of Articulated and Mobile Robots*, vol. 10, pp. 192–222, 2004.

[13] G. Ippoliti, L. Jetto, A. la Manna, and S. Longhi, "Improving the robustness properties of robot localization procedures with respect to environment features uncertainties," *IEEE International Conference on Robotics and Automation*, pp. 1451 – 1458, 2005.

[14] S. Giompapa, F. Gini, A. Farina, A. Graziano, R. Croci, and R. Distefano, "Maritime border control multisensor system," in *Aerospace and Electronic Systems Magazine*, 2009, pp. 9–15.

[15] J.-Y. Bouguet, *Camera Calibration Toolbox for Matlab*, http ://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[16] J. Fabrizio, P. Tarel, and R. Benosman, "Calibration of panoramic catadioptric sensors made easier," in *Proceedings of the IEEE Workshop on Omnidirectionnal Vision*, june 2002.

[17] P. Sturm, "Algorithms for plane based pose estimation," in *Proceedings of the IEEE International conference on Computer Vision and Pattern Recognition*, 2000, pp. 706–711.

[18] R. Hicks and R. Bajcsy, "Reflective surfaces as computational sensor," *Image and Vision Computing*, 2001.