AUV Real-Time Acoustic Vertical Plane Obstacle Detection and Avoidance

Serge Karabchevsky
Department of Electro-Optical Engineering
Ben Gurion University of the Negev
Beer-Sheva, Israel

Boris Braginsky, Hugo Guterman
Department of Electrical and Computer Engineering
Ben Gurion University of the Negev
Beer-Sheva, Israel

***Abstract*— Autonomous Underwater Vehicles (AUVs) operate in unknown underwater environments and take decisions based on sensor readings, without any link with a human operator. It is critical for the AUVs to be able to avoid submerged obstacles and to adapt to changes in the ocean floor terrain. A typical mission for an AUV is to inspect a given area using a side scan. It is crucial to maintain a straight line course during the inspection in order to ensure proper side scan sonar operation. This limits the obstacle avoidance to only the vertical plane. In this study, a real–time obstacle detection and avoidance algorithm that uses vertical plane oriented forward looking sonar is outlined. Experimental results that demonstrate the performance of the proposed method are presented.**

Keywords-AUV; Obstacle avoidance; Sonar

I. INTRODUCTION

Autonomous Underwater Vehicles (AUVs) operate in unknown environments and must take decisions subject to mission-based information provided by their sensors, without feedback or intervention from a human operator. Typical missions for AUVs include sea-floor exploration, mine detection and pipeline and cable inspection [1]. Regardless of the mission, it is critical for the AUVs to be able to avoid submerged obstacles and to deal with changes in the ocean floor terrain while staying on the same course, in order to achieve optimal search resolution. Hence, altitude maneuvering is the only option for obstacle avoidance. Due to the absence of an operator in the control loop, obstacle detection and path planning and avoidance algorithms must be located onboard the AUV. An underwater imaging sonar is typically used by the AUVs for obstacle detection purposes [2] as it is able to obtain data from larger ranges than is possible with conventional optical imaging. Imaging sonar suffers from a low Signal-to-Noise Ratio (SNR) due to the speckle noise that is caused by acoustic wave interference [3], as can be seen in Fig. 1. The low SNR has to be compensated for using onboard algorithms that require a large amount of processing power and cannot be implemented in real-time using standard software techniques. In addition to the obstacle detection problems, the obstacle avoidance system must be adaptive and flexible in order to deal with the nonlinear dynamics of the AUV[4].

In this research, a vertical plane real-time obstacle detection and avoidance method is proposed. The obstacle detection algorithm is able to detect the obstacle height and the distance of the obstacle from the AUV. The detection is based on an adaptive Frost speckle noise filter and local image histogram entropy segmentation, which is not sensitive to the speckle noise. A Radon Transform-based technique was used to detect the ocean floor for obstacle height estimation. After detection, a reactive obstacle avoidance algorithm is initiated. The proposed algorithm responds immediately to the obstacles as they appear. Our study is based on potential field and edge detection methods. Experimental results demonstrate that the method enables the AUV to navigate safely to the destination avoiding obstacles in the vertical plane. FPGA technologies were used in order to maintain real-time performance.

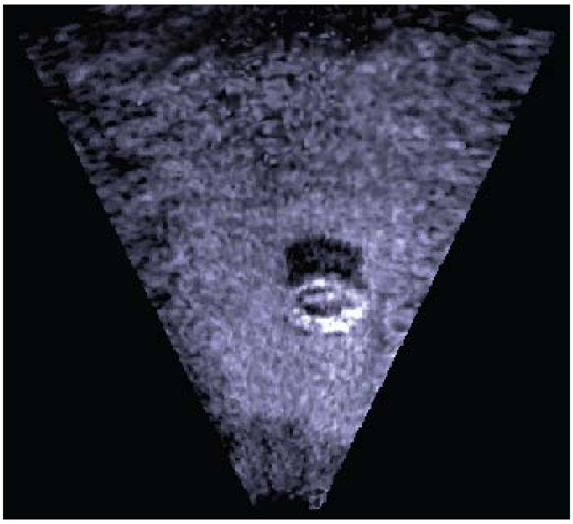

Figure 1. Sonar image of a tire on the sea floor (from BlueView test data); speckle noise and seabed returns can be easily seen.

A. Target Application

The target of this research is to develop an obstacle detection and avoidance system for an AUV operating close to the seabed in a straight line seabed-following mission. The Bluefin 12 [5] AUV with a vertically positioned Forward Looking Sonar (FLS) (BlueView P450) was employed in this research. The dimensions and shape of the AUV can be seen in Fig. 2.

Fugure 2. Bluefin 12 dimensions: 380cm in length and 32cm in diameter

B. *Target Sonar*

A BlueView P450 FLS was used for collecting data [6]. The sonar was configured to export data in polar coordinates in order to reduce the computational cost. The sonar has a 45° × 15° field of view, 256 beams of width 1° × 15° with 0.18° spacing, maximum range of 137 m, range resolution of 2 in, operating frequency of 450 kHz and update rate of up to 10 Hz.

The rest of the paper is organized as follows. In section II the system concept and design are presented. Results are presented in section III and, finally, conclusions are presented in section IV.

II. System Structure

The system comprises two main components; a computer and an FPGA board that hosts all the critical time-consuming parts of the algorithms, as described in Fig. 3.

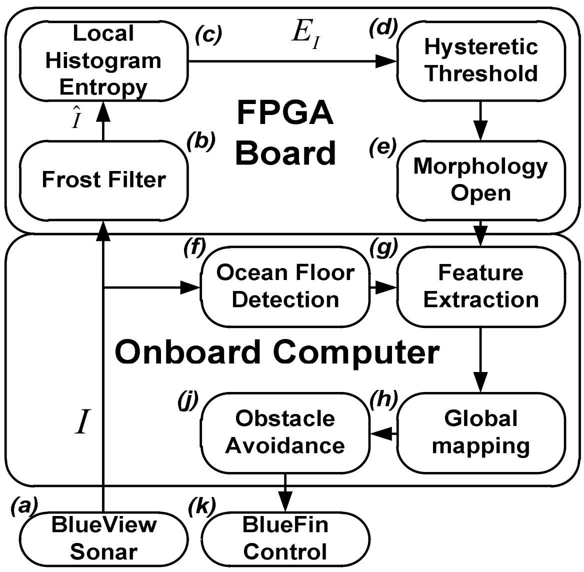

Figure 3. Block diagram of the system architecture. The sonar image is acquired by the on-board computer and forwarded to the FPGA board (a), then the image is filtered using the Frost filter to suppress speckle noise (b). Next, the local histogram entropy of the image is calculated (c) and hysteretic threshold determination takes place (d). Prior to the feature extraction (g), morphological open operation is used (e) and the ocean floor is detected (f). After all the obstacles have been extracted they are mapped to the global coordinates (h) and the obstacle avoidance algorithm is performed (j). If the obstacle is on the AUV path, a trajectory change is requested from Bluefin controller (k).

The FPGA board that was used is a GiDEL PROC104 development board [7] with Altera StratixIII-110E FPGA [8]. The sonar data is received by the onboard computer and forwarded to the FPGA board as a stream of pixels using a PCI Express interface. At the FPGA board, speckle noise suppression is carried out using an adaptive Frost [9] filter. Then, local image histogram entropy is calculated to provide an informational measure. Lastly, the hysteretic threshold operation takes place and the result is returned to the computer. Back at the onboard computer, the ocean floor position and slope are calculated using a Radon Transform-based technique to estimate the height of the obstacles. A feature extraction procedure is performed to label the obstacle regions and calculate the height and range of each obstacle. The detected obstacles are then mapped to the global coordinates. The obstacle avoidance algorithm only takes place when an obstacle is detected in the way of the AUV, when it generates avoidance commands that are sent to the Bluefin control.

The following sections describe in detail the system algorithms.

A. *Frost Filter Speckle Noise Suppression*

Speckle noise is a granular noise that exists in coherent imaging such as laser imaging, radar and ultrasonic imaging. It is a speckle pattern of random intensity produced by the mutual interference of a set of wave fronts. Speckle noise is a multiplicative noise, i.e. it is in direct proportion to the local intensity of the area. The speckle noise has to be removed before calculating local image histogram entropy because both speckle noise and an obstacle result in high entropy.

Two common approaches exist to deal with this type of noise. The first approach is to average several images acquired from the same scene, which is called multi-look processing or super-resolution [10]. In this way, the speckle noise will be reduced, due to its random nature, while the observed scene will not be degraded. The second technique is based on filtering the speckle noise on the basis of a single image using a two-dimensional filter. Standard filters generally degrade the observed scene. In contrast to standard filters, adaptive filters take local image information into consideration, while carrying out the filtration process.

Adaptive filters reduce the speckle noise in homogeneous areas, while preserving texture and high frequency information in heterogeneous areas. During our previous work [11], it became clear that the adaptive Frost [9] filter in a 17×17 pixel neighborhood with K=1 is the best choice for this type of task. The Frost filter [9] estimates the observed scene by convolving the acquired image with an adaptive kernel, which changes its properties according to the local image statistics. The process can be defined by the following equation:

$$\hat{I}(x,y) = I(x,y) * m(x,y) \tag{1}$$

where *(x,y)* is the image spatial coordinates, I is the acquired image, $\hat{R}$ is the estimated scene, and *m(t)* is the adaptive kernel defined by:

$$m(x,y) = \frac{e^{-KC_I^2(x,y)|d|}}{\sum e^{-KC_I^2(x,y)|d|}}, C_I(x,y) = \frac{\sigma_I(x,y)}{\bar{I}(x,y)} \tag{2}$$

where K is the dumping factor (between 0.1 and 1) and $|d| = |x - x_0| + |y - y_0|$ is the distance of the pixel from the kernel center. $C_I(x,y), \sigma_I(x,y)$ and $\bar{I}(x,y)$ are the image variation, standard deviation and mean in the filter area.

B. Ocean Floor Location

The FLS uses an array of 256 piezoelectric transducers, where each transducer is orientated at a different angle. For each frame, the FLS first produces a wave front using the transducers and then senses the echo returns. The time of each echo return is proportional to the distance to the detected object. The result of each ping is a matrix which contains a scene representation in polar coordinates. This matrix can be transformed to Cartesian coordinates for easier perception. The conversion to Cartesian coordinates requires complex computations and results in lower, non-uniform resolution.

The cross section of a sonar scan while seabed-following can be seen in Fig. 4.

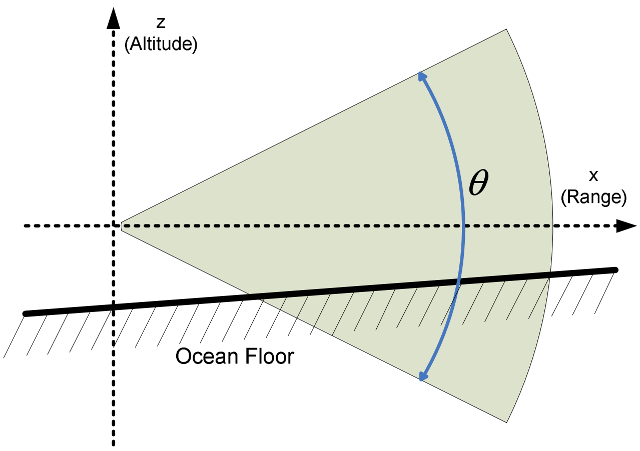

Figure 4. Cross Section of the vertical sonar plane

The ocean floor scatters most of the acoustic energy and therefore will result in a straight line in the Cartesian sonar image because no acoustic energy will pass beyond the ocean floor. An acoustic image of the ocean floor in Cartesian and in polar coordinates can be seen in Fig. 5 and Fig. 6, respectively.

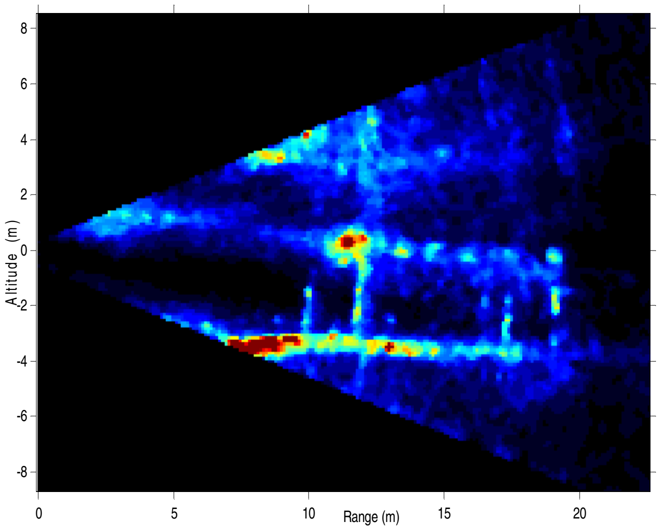

Figure 5. Sonar image in Cartesian coordinates; lower straight horisontal line indicates the ocean floor.

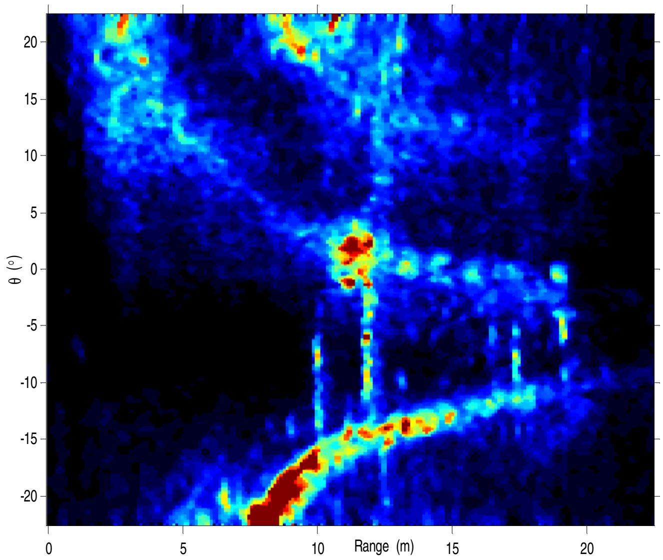

Figure 6. Sonar image in polar coordinates; lower curve indicates the ocean floor.

Due to the resolution problems in the Cartesian coordinate system, the sonar images will be processed in their native polar coordinates. Therefore the ocean floor line must be mapped to the polar coordinate system.

An ocean floor in Cartesian coordinates can be defined as

$$y = ax + b \tag{3}$$

where *(x,y)* is the coordinate in the sonar image *a* is a slope of the ocean floor and *b* is the AUV altitude. A transform equation has to be applied to map the above line to sonar polar coordinates (r, θ)

$$x = r\cos\theta, y = r\sin\theta \tag{4}$$

resulting in the following representation of the ocean floor in polar coordinates:

$$r = \frac{b}{\sin\theta - a\cos\theta} \tag{5}$$

Under the assumptions that the angular range of the sonar and the ocean floor slope are bounded by $\theta \in [-22.5°,\ 22.5°]$, $a \in [-1, 1]$ the denominator of Eq. (5) can be approximated by a second order polynomial, using Taylor series to speed up the calculations:

$$r = \frac{b}{\sin\theta - a\cos\theta} = \frac{b}{\theta - a} \tag{6}$$

For the detection of the ocean floor in polar coordinate we propose a Radon Transform-based method [12]. The Radon Transform is defined as a line integral over a continuous two dimensional function $f(X) = f(x, y)$ by:

$$\Re f(L) = \int_L f(x)\,dx \tag{7}$$

For our particular case, the Radon Transform is defined as:

$$\Re f(a,b)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(r,\theta)\delta\left(\frac{b}{\theta-a}-r\right)drd\theta = \\ =\int_{-\infty}^{\infty} f\left(\frac{b}{\theta-a},\theta\right)d\theta \quad (8)$$

where $f(r,\theta)$ is a two dimensional function that represents the sonar image in polar coordinates, b is the AUV altitude and a is the ocean floor slope. The Dirac δ-function converts the two-dimensional integral to a line integral along the line $\frac{b}{\theta-a} - r = 0$ and (r,θ) are the pixel coordinates in the polar sonar image. Then, the result of the Radon Transform is searched for the maximal value that will indicate *(a,b)* of the best fitted ocean bottom curve. The result of the Radon Transform on an image in polar coordinates can be seen in Fig. 7.

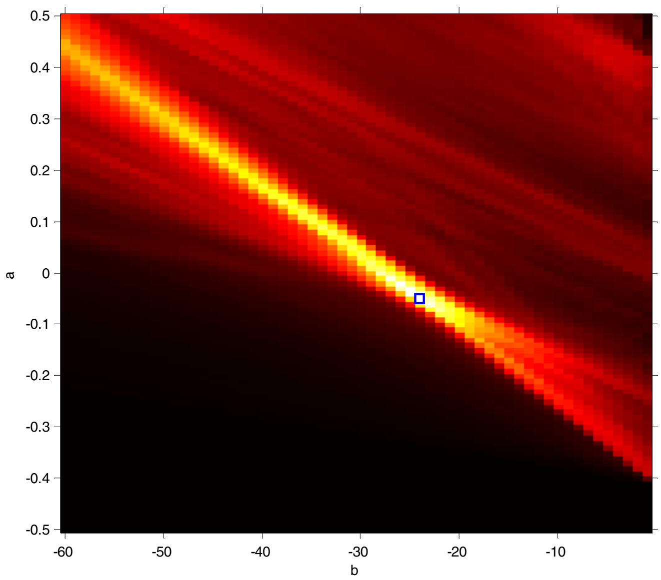

Figure 7. Radon Transform of sonar Image. Blue rectangle indicates the best matching (a,b) position

C. Segmentation

Segmentation of sonar images can be very complex. Different obstacles with different echo return levels can exist in a single image. Therefore, standard fixed threshold algorithms [13, 14] and even adaptive algorithms may miss obstacles with a low echo return. During our research it became obvious that local image histogram entropy-based [15] segmentation provides outstanding results. The local histogram entropy method is based on the Shannon Theorem [16] and has been previously used for the segmentation of optical images [17]. Entropy is a measure of the information content in a probability distribution and can be defined by (9):

$$E_I(x,y)=\sum_{i=0}^{N_{bins}}\frac{p(i)}{N_{bins}}\cdot\log\left(\frac{p(i)}{N_{bins}}\right) \quad (9)$$

where *p(i)* is the value of the particular bin of the N_{bins} local histogram of the image in the $N \times N$ pixel neighborhood at *(x,y)* coordinates. As our experiments show, obstacles will result in high entropy, which is not dependent on the image magnitude. A local histogram of 256 bins and a 9×9 window has shown good quality results.

A hysteretic threshold was then applied on the entropy result. A hysteretic threshold uses two different threshold levels. The first, high-level threshold selects the parts of the image that are to be treated as obstacles, while the second, low-level threshold selects the areas that will be treated as an obstacle only if they are connected by four-connectivity to the pixels that are above a high-level threshold. 70% and 60% of the maximum entropy level were used. The main objectives of this algorithm are to discard middle value peaks that are not connected to high entropy regions and to ignore the noise. A threshold is followed by a 3×3 pixel, morphologically open [18] noise removal operator.

D. Feature Extraction and Global Mapping

Once segmented, regions representing the obstacles in the image are labeled using a standard labeling algorithm that defines a bounding box for each obstacle. Then, the obstacles are mapped onto global coordinates according to (10):

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = A(\varphi,\theta)\begin{bmatrix} d \\ 0 \\ h \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ 0 \end{bmatrix} \quad (10)$$

where x_g, y_g, z_g are the obstacle representation in global coordinates, d is the distance to the obstacle, h is the obstacle altitude, x_c, y_c, z_c are vehicle global coordinates, φ is yaw angle, θ is pitch angle and $A(\varphi,\theta)$ is a transform matrix defined by (11):

$$A=\begin{bmatrix} \cos(\varphi)\cdot\cos(\theta) & -\cos(\theta)\cdot\sin(\varphi) & 0 \\ \cos(\theta)\cdot\sin(\varphi) & \cos(\varphi)\cdot\cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

E. Vertical Obstacle Avoidance Algorithm.

A reactive Obstacle-Avoidance Algorithm (OAA) gives an immediate response to the problem as it occurs. It works only when an obstacle is near to an AUV. This method does not take into account the obstacles that are not in the path of the AUV. The main reactive OAAs are based on edge detection, certainty grids and potential field methods [19].

1) Edge detection

One popular OAA is based on edge detection. This algorithm tries to determine the position of the vertical edge of the obstacle and then tries to find the optimal path around either one of the edges. A disadvantage of this approach is that not all of the edges may be visible.

2) Potential field methods

The basic principle of the potential field method lies in creating imaginary force actions on the robot [20]. In this method, obstacles exert repulsive forces, while the goal applies an attractive force on the robot.

The OAA receives bounding box parameters of the detected obstacles in local coordinates, as can be seen in the simulation sonar image in Fig. 8.

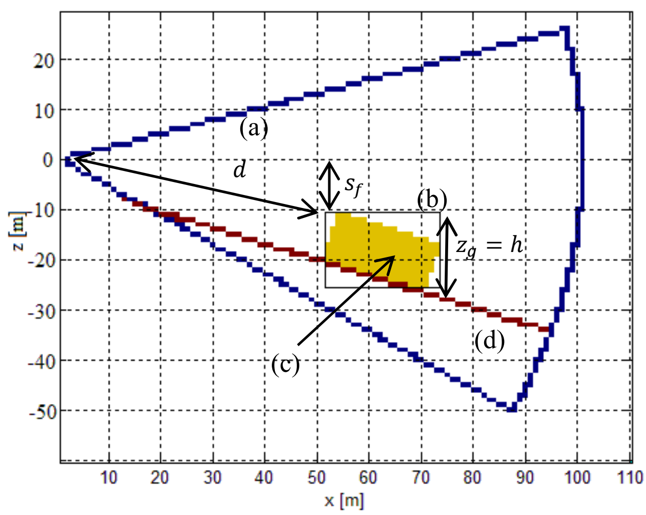

Figure 8. Vertical sonar image from simulation, showing (a) sonar view limits, (b) boundary box of obstacle, (c) obstacle and (d) ocean floor.

Using equation(10), an algorithm transforms the local coordinates of the obstacle into global coordinates z_g and x_g. Then, in the global coordinate space, the OAA can take a decision according to equation(12).

$$z_c < z_g + s_f \qquad (12)$$

where s_f is the safe distance between the vehicle and the obstacle. If inequality (12) is satisfied, the algorithm transfers a new altitude $(z_g + s_f)$ into the Bluefin control and keeps this altitude till the AUV passes above the obstacle or above several close obstacles.

III. RESULTS

The algorithm was tested on several sonar images acquired under different conditions. In all cases the algorithm showed outstanding results. Two of the cases can be seen in Fig. 9, demonstrating detection of a wharf at large echo contrast conditions and in Fig. 10, demonstrating detection of a synthetic obstacle generated from a simulator with speckle noise.

The maximal operation frequency of the FPGA design is 333 MHz. The power consumption of the system is 2.5 Watts and the maximal frame rate is 556 fps. The details of the FPGA resource utilization are given in Table 1.

TABLE I. FPGA RESOURCE USAGE

Parameter	LUTs	Registers	Memory (Kbits)	DSP Blocks
Logic Usage	22,840	21,550	166	176
Usage Percentage	27%	25%	2%	20%

In order to establish a baseline for the performance test, the algorithm was executed on an Intel i7-720QM Processor (6M Cache, 1.60 GHz, 4 Cores). The power consumption of the Processor was 45W (as reported by Intel). This CPU was chosen for its low power consumption and high performance. The software implementation frame rate was 0.263 fps. The hardware architecture provides a speedup of 2,114 and power reduction by 94% compared to the software implementation.

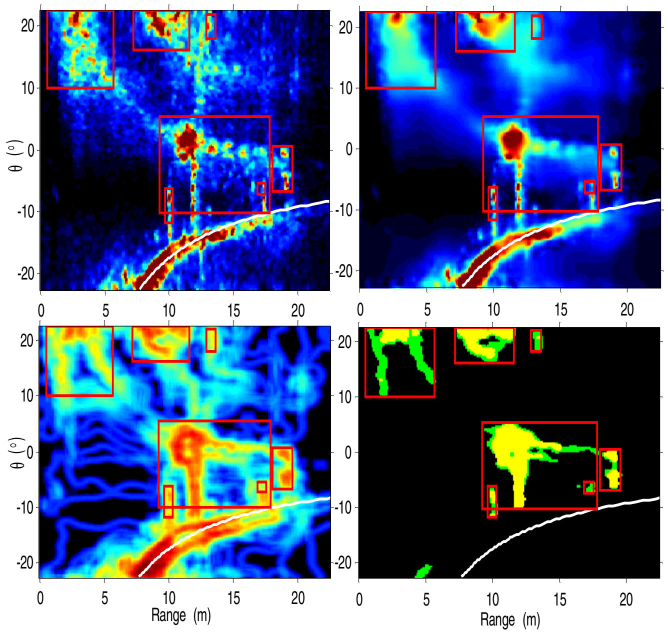

Figure 9. Detection of a wharf. Top Left; an image acquired from the sonar. Top Right; image after speckle suppression. Bottom Left; local image histogram entropy. Bottom Right; yellow color for pixels above high-level threshold, green color for pixels above low-level threshold. The white line indicates the detected ocean floor, while rhe red rectangles mark the detected obstacles

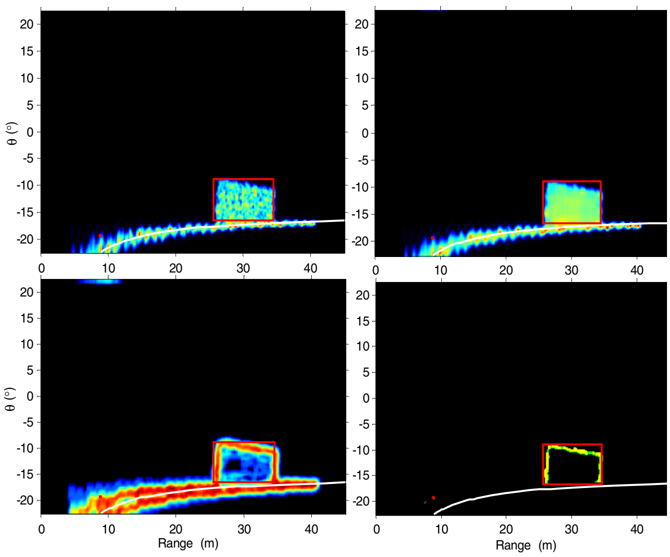

Figure 10. Detection of an obstacle from synthetic data. Top Left; an image acquired from the sonar.Top Right; image after speckle suppression. Bottom Left; local image histogram entropy. Bottom Right; yellow color for pixels above high-level threshold, green color for pixels above low-level threshold. The white line indicates the detected ocean floor, while rhe red rectangles mark the detected obstacles.

In order to test the performance of the sonar image processing with the obstacle avoidance algorithm an AUV simulator based on Bluefin 12 vehicle dynamics was developed. The simulator takes into account the dynamic model of the AUV and the behavior of the sensors. The simulator structure can be seen in Fig. 11.

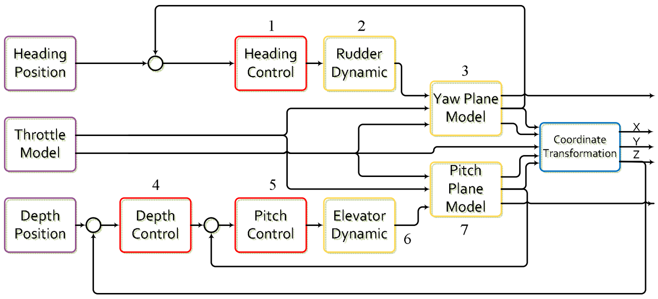

Figure 11. Block diagram of the AUV simulator. (1) Heading controller. (2) Rudder dynamic. (3) Linear yaw plane model. (4) Depth controller. (5) Pitch controller. (6) Elevator second order dynamic. (7) Linear pitch plane model.

The obstacle avoidance was simulated in the above environment with the conditions of 3 knots velocity and sonar detection range of 60m. During the simulated scenario, the AUV encounters obstacles while performing an ocean floor survey mission. The side view can be seen in Fig. 12. The AUV trajectory in the simulated scenario shows that our technique focuses strongly on safety as the AUVs start avoidance maneuvering immediately when the obstacles are detected. It can be seen that after the AUV has overcome the obstacles it returns back to the mission's initial planned altitude.

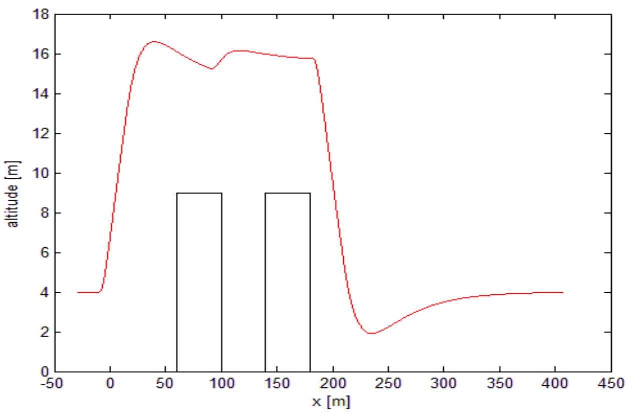

Figure 12. Simulation result of obstacle avoidance algorithm. Red line indicates the AUV path while black rectanlels indicate the obstacles.

IV. Conclusions

An obstacle detection algorithm that provides outstanding detection for AUVs operating close to the seabed, in spite of high speckle noise conditions has been presented. Conventional software-based implementations cannot deal with the high computation complexity of the algorithm and FPGA hardware architecture has been proposed for the algorithm in order to maintain the required real-time performance. The reactive OAA manages to navigate the AUV to the target safely, while avoiding the obstacles and maintaining a straight course line.

References

[1] E. Desa, R. Madhan, and P. Maurya, "Potential of autonomous underwater vehicles as new generation ocean data platforms," *Current Science,* vol. 90, May, 2006 2006.

[2] E. O. Belcher, W. L. J. Fox, and W. H. Hanot, "Dual-frequency acoustic camera: a candidate for an obstacle avoidance, gap-filler, and identification sensor for untethered underwater vehicles," in *OCEANS '02 MTS/IEEE*, 2002, pp. 2124-2128 vol.4.

[3] J. G. Abbott and F. L. Thurstone, "Acoustic speckle: Theory and experimental analysis," *Ultrasonic Imaging,* vol. 1, pp. 303-324, 1979.

[4] L. Lapierre and B. Jouvencel, "Robust Nonlinear Path-Following Control of an AUV," *Oceanic Engineering, IEEE Journal of,* vol. 33, pp. 89-102, 2008.

[5] Bluefin. *Bluefin 12 Datasheet*. Available: http://www.bluefinrobotics.com/products/bluefin-12s/

[6] BlueView. *P450-45 Miniature Multibeam Imaging Sonar*. Available: www.blueview.com

[7] GiDEL. *PROCeIII Product Brief*. Available: www.gidel.com

[8] Altera. (2009). *Stratix III Device Handbook*. Available: www.altera.com

[9] V. S. Frost, J. A. Stiles, K. S. Shanmugan, and J. C. Holtzman, "A Model for Radar Images and Its Application to Adaptive Digital Filtering of Multiplicative Noise," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. PAMI-4, pp. 157-166, 1982.

[10] K. Kim, N. Neretti, and N. Intrator, "Mosaicing of acoustic camera images," *Radar, Sonar and Navigation, IEE Proceedings -,* vol. 152, pp. 263-270, 2005.

[11] S. Karabchevsky, D. Kahana, O. Ben-Harush, and H. Guterman, "FPGA-Based Adaptive Speckle Suppression Filter for Underwater Imaging Sonar," *Oceanic Engineering, IEEE Journal of,* vol. 36, pp. 646-657, 2011.

[12] S. Helgason, *The Radon Transform*, 2 ed. vol. 5. Boston, 1999.

[13] L. Henriksen, "Real-time underwater object detection based on an electrically scanned high-resolution sonar," in *Autonomous Underwater Vehicle Technology, 1994. AUV '94., Proceedings of the 1994 Symposium on*, 1994, pp. 99-104.

[14] Y. Petillot, I. Tena Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *Oceanic Engineering, IEEE Journal of,* vol. 26, pp. 240-251, 2001.

[15] C. I. Chang, Y. Du, J. Wang, S. M. Guo, and P. D. Thouin, "Survey and comparative analysis of entropy and relative entropy thresholding techniques," *Vision, Image and Signal Processing, IEE Proceedings -,* vol. 153, pp. 837-850, 2006.

[16] T. Cover and J. Thomas, "Elements of information theory," *(John Wiley & Sons, Inc., 1991)*.

[17] A. S. Abutaleb, "Automatic thresholding of gray-level pictures using two-dimensional entropy," *Computer Vision, Graphics, and Image Processing,* vol. 47, pp. 22-32, 1989.

[18] R. Haralick and L. Shapiro, "Computer and Robot Vision," *Addison-Wesley Publishing Company,* vol. 1, pp. 174 - 185, 1992.

[19] J. Borenstein, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *Transactions on Robotics and Automation,* vol. 7, 1991.

[20] P. M. Stingu and F. L. Lewis, "Motion Path Planning for Mobile Robots."