HAL
open science

Hardware Implementation and Validation of 3D Underwater Shape Reconstruction Algorithm Using a Stereo-Catadioptric System

Rihab Hmida, Abdessalem Ben Abdelali, Frédéric Comby, Lionel Lapierre, Abdellatif Mtibaa, René Zapata

To cite this version:

Rihab Hmida, Abdessalem Ben Abdelali, Frédéric Comby, Lionel Lapierre, Abdellatif Mtibaa, et al.. Hardware Implementation and Validation of 3D Underwater Shape Reconstruction Algorithm Using a Stereo-Catadioptric System. Applied Sciences, 2016, 6 (9), pp.247-272. 10.3390/app6090247 . lirmm-01387784

HAL Id: lirmm-01387784

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01387784v1

Submitted on 26 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

applied sciences

MDPI

Article

Hardware Implementation and Validation of 3D Underwater Shape Reconstruction Algorithm Using a Stereo-Catadioptric System

Rihab Hmida [1,2,*], Abdessalem Ben Abdelali [1], Frédéric Comby [2], Lionel Lapierre [2], Abdellatif Mtibaa [1] and René Zapata [2]

[1] Laboratory of Electronics and Microelectronics, University of Monastir, Monastir 5000, Tunisia; abdessalem.benabdelali@enim.rnu.tn (A.B.A.); abdellatif.mtibaa@enim.rnu.tn (A.M.)

[2] Laboratory of Informatics, Robotics and Microelectronics, University of Montpellier, Montpellier 39000, France; comby@lirmm.fr (F.C.); lapierre@lirmm.fr (L.L.); zapata@lirmm.fr (R.Z.)

* Correspondence: hmida.rihab@gmail.com or rihab.hmida@lirmm.fr; Tel.: +33-665-743-106

Academic Editor: José Santamaria
Received: 19 May 2016; Accepted: 17 August 2016; Published: 31 August 2016

Abstract: In this paper, we present a new stereo vision-based system and its efficient hardware implementation for real-time underwater environments exploration throughout 3D sparse reconstruction based on a number of feature points. The proposed underwater 3D shape reconstruction algorithm details are presented. The main concepts and advantages are discussed and comparison with existing systems is performed. In order to achieve real-time video constraints, a hardware implementation of the algorithm is performed using Xilinx System Generator. The pipelined stereo vision system has been implemented using Field Programmable Gate Arrays (FPGA) technology. Both timing constraints and mathematical operations precision have been evaluated in order to validate the proposed hardware implementation of our system. Experimental results show that the proposed system presents high accuracy and execution time performances.

Keywords: stereovision; underwater; 3D reconstruction; real-time; FPGA; System Generator

1. Introduction

Exploitation and preservation of groundwater resources are very important with many challenges related to economic development, social constraints and mainly environment protection. In this context, hydro-geologists are, for example, highly attracted by the interest of confined underwater environments. They mainly address the question of dynamic modeling of such complex geological structure (the karst). In this domain, robotic is of great interest, and many different hot topics in terms of control, navigation, sensing and actuation are addressed. On the sensing aspect, robotic exploration of confined environment requires sensors providing pertinent input information to supply the control loop. In this scope, mechanical profiling sonars exist but they provide a too low sampling rate to be used directly in the robot control loop. That is why the use of vision techniques is the most suitable solution even though it presents a more demanding constraint in terms of quantity of information to be processed.

Different visual perception devices are proposed in literature, offering a large view of the environment. However, a trade-off between images resolution, cost, efficiency and real time processing has to be taken into consideration. The use of cameras and computer vision techniques represents a common choice for autonomous scene reconstruction. Nevertheless, the computational complexity and large amount of data make real-time processing of stereo vision challenging because of the inherent instruction cycle delay within conventional computers [1]. In this case, custom architecture implementation on FPGA (Field Programmable Gate Arrays) is one of the most recommended solutions.

It permits to exploit the intrinsic parallelism of the 3D reconstruction algorithms and to optimize memory access patterns in order to ensure real-time 3D image reconstruction [2]. By taking advantage of the resources integrated into modern FPGA circuits, dedicated architectures are designed with more hardware efficiency and time performances. In addition, due to its reconfigurability, FPGA allow the architecture to be modified and adapted to the application requirements. It can be optimized depending on the implemented task, unlike conventional processors or Digital Signal Processors.

In the field of computer vision applications, 3D reconstruction algorithms are nowadays of great interest and could be exploited in different embedded systems and real-time applications such as geographic information systems, satellite-based earth and space exploration, autonomous robots and security systems. In this paper, we propose a 3D shape reconstruction algorithm for underwater confined environment (karst) exploration using stereoscopic images and its pipelined hardware architecture design. The hardware implementation is performed on a Virtex-6 FPGA using the Xilinx System Generator (XSG) development tool. We aim to propose a high efficiency system in terms of level of accuracy and execution time performances.

The rest of the paper is organized as follows. In Section 2 we present and discuss various existing image processing methods that focus on laser spot detection, matching and 3D reconstruction. In Section 3, we present the vision-based sensing device and its calibration. A flowchart of the proposed algorithm and the key processes details are presented in Section 4. Section 5 presents the hardware XSG design of the 3D reconstruction process. In Section 6, we present the experimental and performance evaluation results. Finally, conclusions and future works are presented in the last section.

2. Related Work and the Proposed Algorithm

2.1. Related Work

The reconstruction process first needs a number of feature points to obtain an accurate 3D modeling of the karst. Multiple image processing algorithms, permitting the detection and extraction of region of interest are available in literature. Fast descriptors [3] are commonly used in order to speed-up classification and recognition of a specific set of keypoints as well as SIFT (Scale-Invariant Feature Transform) [4], Harris [5] and SURF (Speeded Up Robust Features) [6] descriptors. All of these feature detectors have invariant characteristics in a spatial domain. However, they are lacking quality when images undergo important modifications like underwater ones [7]. Using active systems with laser devices permits to facilitate the extraction of accurate features based on the laser characteristics. In [8], authors proposed a new laser detection technique that combines several specifications such as brightness, size, aspect ratio, and gradual change of intensity from the center of laser spot. In [9], others use a combination of motion detection, shape recognition and histograms comparison methods in order to detect the laser spot. In the same context, segmentation process appears in most of works such as [10–15]. In fact, thresholding permits to classify the laser spot according to brightness value using a previously calculated threshold value. In [16], a new approach is proposed using a different color system that represents Hue, Saturation and Intensity (HSI) instead of RGB color system. The previous mentioned methods are sometimes not enough accurate and present some false-offs because of lighting conditions or threshold value selection. In [17], authors present an approach that permits to differentiate good and bad detections by analyzing the proportionality between the projection surface diameter and the number of classified laser pixels in a group. In [18], fuzzy rule based system adjusted by a genetic algorithm is proposed to improve the success rate of the laser pointer detection algorithm decreasing as much as possible the false-offs. In [19], authors developed an approach based on motion estimation to extract "good" features. They first use the Kanade-Lucas-Tomasi feature tracking algorithm to extract features at time t which are then tracked at time $t + 1$. Using the disparity map previously extracted for both time steps, tracked points that do not have a corresponding disparity at both time t and $t + 1$ are eliminated. In [20], authors were focused on feature point extraction, the first step of the sparse 3D reconstruction system, since they represent the geometrical characteristics of the

scene. They propose a methodology to extract robust feature points from acoustic images and offer 3D environment reconstruction. This approach is realized in two steps: contours extraction and corners from contours feature points extraction. For the first step, canny edge detection [21] was applied based on intensity discontinuities. For the second step, an algorithm that analyzes each defined curve to mark the first and last points as feature points. These characteristic points, extracted from each image, are matched in order to estimate the camera movement and reconstruct the scene. In [22], authors used also edge detection approach to devise stereo images into blocks. A sparse disparity map is generated based on stereo block matching. They investigate for this aim, the performances and cost calculation results of different methods of matching such as Sum of Absolute Differences (SAD), Herman Weyl's Discrepancy Measure (HWDM) and Adaptive Support Windows (ASW) to compare performances of each method in estimation of disparity and maximum disparity values.

In all 3D systems, depth information of the real scene is essential. Algorithms of reconstruction depend principally on the system of vision (monocular, stereo, passive or active ...), the scene (indoor/outdoor, illumination ...), the input data, the application goals (pose estimation, tracking, mapping ...). Three dimensional shape reconstructions could be achieved using image shading, texture, photometric stereo, structured light, motion sequences, etc. In addition, reconstruction could be performed using image formation models and specific numerical techniques such as those presented in [23]. Early, shapes from stereo techniques in computer vision were tested on stereo image sequences [24] where a dense object reconstruction is obtained using iterative feature matching and shape reconstruction algorithms. The proposed algorithm resolves matching ambiguities by multiple hypothesis testing. Modern techniques use Delaunay triangulation algorithms for point cloud data with only geometric parameters as input since it have been shown to be quite effective both in theory and practice [25], although its long computational time when processing large data. It is also used in [6,7,23,24] in addition with elastic surface-net algorithm [26] or the marching cubes algorithm [27] to extract mesh. Kiana Hajebi presented in her work [28], a sparse disparity map reconstruction were produced based on features-based stereo matching technique from infrared images. A number of constraints (epipolar geometry) and assumptions (image brightness, proximity ...) are employed to retrieve matching ambiguities. A Delaunay triangulation algorithm is applied in order to reconstruct the scene, using disparity map information.

In the case of underwater applications, water-glass-air interfaces between the sensor and the scene modify the intrinsic parameters of the camera and limit the standard image processing algorithms performances. Therefore, underwater image formation are studied and developed in [29–31] in order to incorporate properties of light refraction in water. The light refraction effects should be considered either when an underwater calibration is carried out [32–34] or during the 3D reconstruction algorithm [35]. The Refractive Stereo Ray Tracing (RSRT) process allows accurate scene reconstruction by modeling and tracing the refraction of the light. It is well suited for complex refraction through multiple interfaces; however this requires complete knowledge of the 3D projection process. This approach does not need an underwater calibration of the system of vision but the algorithm of light trajectory tracking is computationally time consuming particularly when there are multiple different environment between the 3D point and its 2D image projection. In contrast, underwater calibration, despite its experimentations ambiguities, provides a direct and simplified approximation between the merged 3D point and its 2D image correspondence as it takes into consideration light refraction effects in the expression of 2D-3D transformation.

Several studies on 3D reconstruction systems are proposed in literature. However, few of them are designed for underwater application and fewer target real-time implementation. In Table 1, we summarize the main treatments and characteristics of some existing systems and those of our algorithm. The algorithms used for each case depend on multiples aspects: the 3D capturing sensor (passive/active, monocular/stereo, calibrated/non-calibrated ...), the points of interest that will be reconstructed (most interest points in all the image, specific key points, checkerboard corners ...), the application requirements (precision, software and hardware resources, timing ...), the system

environment (aerial, underwater), etc. For example, the capturing sensor constitutions and configurations present a great interest to facilitate the image processing algorithms. For underwater applications, with the use of passive sensors [2,36], the identification and recognition of object is tough and sometimes hopeless because of the underground homogenous texture and lack of illumination. Atmospheric conditions that affect the optical images at different acquisition times could be a potential source of error and they should be taken into consideration to avoid missed detection or false-offs. The use of active devices directed toward the target to be investigated, like scanners [37] or lasers [38,39] or sonars [5,40] the use of depth sensor like the Kinect [41], allows overcoming these drawbacks particularly when real-time performance is desired. In the other hand, when the configuration of the different sensors is less complex, the geometric modelisation (calibration) of the whole system becomes possible and more accurate. This allows us to define the properties of the epipolar geometry (calibration parameters) that could be used afterwards to enhance the different image processing, particularly to remove the outliers [5,6] facilitate the matching process using epipolar lines [5,36] and obviously for 3D triangulation [5,6]. Furthermore, methods such as SURF detector [6] or Harris corner detector [5] or Sobel edge detector [2] were used to detect most interesting points in the image. These methods, despite their good detection rate, necessitate big computation time since they use complex calculations (convolution with a Gaussian, gradient vectors, descriptors, etc.). Therefore, some authors propose different alternatives that permit to substitute global methods with a combination of standard processing methods such as those presented in [13,36,40] which permits to ensure good accuracy and run time.

2.2. The Proposed Algorithm

For the adopted 3D underwater shape reconstruction system, we tried to integrate the most key approaches applied in literature while ensuring a good accuracy and a satisfactory processing time. The following points summarize the main involved techniques and advantages of the proposed system:

(1) Stereo active system: Because of underwater restrictions, we equipped our system with a laser pointing devices to facilitate the identification of region of interest. The stereo omnidirectional configuration is used in order to provide a large and common field of view of the scene, particularly the lasers projections. In addition, the vertical configuration of both cameras let us profit from the epipolar image superposition to facilitate the matching process which is, in general, the most time consuming treatment.

(2) Lighting conditions consideration: The use of a single threshold value for the binarisation may lead to two consequences: partially degraded information when using a high threshold value for poor-lighted images or a lot of outliers when using a low threshold value for well-lit ones. Several experiments were done on different lighting conditions in order to define an adequate thresholding expression, not a value, suitable for different conditions and that permits to efficiently highlight the laser spot projection with respect to the background.

(3) Good detection rate: Generally, there is no generic method which solves all of feature detection problems. In fact, some false-offs may appear. That is why some authors propose to use other geometric constraints (proximity criteria, region criteria, epipolar constraint, etc.). In our case, we profit from the configuration of the laser belt with respect to the stereo cameras to define a region-based approach that permits to eliminate outliers.

Table 1. Taxonomy of 3D reconstruction systems.

Ref.	[6]	[5]	[2]	[13]	[42]	[43]	Our algorithm
3D capturing sensor	uncalibrated underwater images	stereo camera images using multi-beam sonar	perspective stereo image pairs	perspective camera + Laser pointers devices	CMOS camera+laser structured system	perspective camera + single laser pointer	Omnidirectional stereo images+structured laser system
Calibration	use given calibration information	underwater checkerboard corners detection	assume that cameras are calibrated and rectified	Homography on checkerboard measurement	checkerboard corners detection	manual distance measurements as reference	underwater checkerboard corners detection
Points of Interest detection	SURF detector	Harris corner detector	Sobel edge detector	Green segmentation	Median filter Erosion Region labeling	Segmentation Smoothing Dilating Center finding	Adaptive segmentation
Remove outliers	Epipolar geometry	Epipolar geometry	×	×	Median filter		Epipolar geometry
3D points	All stereo corresponding SURF points	All stereo corresponding Harris points	All edge points	Intensity Weighted Sum of laser clouds	Laser clouds center calculation	Laser position in axis center	Laser clouds gravity centers calculation
Matching	Euclidean distance between features vectors	Lucas–Kanade Tracker (epipolar lines)	Sum Absolute Difference	Grid Mapping Heuristic	Epipolar geometry (lines)	No matching needed	Epipolar geometry (search along a line)
3D reconstruction	Triangulation (calibration informations)	Triangulation (midpoint method)	Triangulation (epipolar geometry)	Triangulation (system geometry)	Triangulation using off-line depth model	using reference measurement	Omnidirectional projection using system geometry
Underwater restitution	×	underwater calibration parameters	×	×	×	underwater reference measurement	underwater calibration parameters
Software implementation	✓	✓	×	✓	×	✓	×
Hardware implementation	×	×	FPGA	×	FPGA	×	FPGA

(4) Accurate 3D reconstruction: A good system calibration leads to a better 3D reconstruction result. In [43], an important reconstruction error is obtained when using non-calibrated images. In our work, we perform with an underwater calibration process using a checkerboard pattern of known geometry. This process permits us to define an accurate relation between 2D coordinates and corresponding 3D points.

(5) Underwater refraction effects consideration: In order to decrease underwater reconstruction error, light refraction effects should be considered either during the calibration [2,5] or in the 2D/3D transformation algorithm [29,30,37,38,44]. Different theories and geometries, based on the ray-tracing concept, were proposed in [37,38,44]. However, the main drawback of these ones is their complexity and cost in terms of execution time. In our work, calibration was directly performed underwater. The estimated parameters describe the physical functioning of our system of vision when merged in water. They define the relation between a pixel projection on the image and its corresponding 3D underwater position. In this case, there is no need to track the light trajectory in each environment (air, waterproof, and water) since refraction effects are already taken into consideration through the underwater estimated parameters.

(6) Use of XSG tool for hardware implementation: During the algorithm development, we attempt to propose an architecture considering maximum use of parallelism. For the hardware implementation validation of the pipelined architecture, we use System Generator interface provided by Xilinx. The main advantage of XSG use is that the same hardware architecture will be used first for the software validation and then for performances evaluation.

3. The Stereo System and Its Underwater Calibration

3.1. Description of the Stereo-Catadioptric System

Many techniques and devices developed for the aim of 3D mapping applications were evaluated in robotics and computer vision research works. Omnidirectional cameras are frequently used in many new technological applications such as underwater robotics, marine science and underwater archaeology since they offer a wide field of view (FOV) and thus a comprehensive view of the scene.

In our work, we go towards an observation of the entire robot's environment with the use of minimum cameras to avoid occlusion and dead angles. We propose a stereo catadioptric system (Figure 1) mounted with a belt of twelve equidistant laser pointing devices (Apinex.Com Inc., Montreal, Canada). The catadioptric system consists on a Point Grey's Flea2 camera (Point Grey Research Inc., Richmond, BC, Canada) pointing at a hyperbolic mirror (V-Stone Co., Ltd., Osaka, Japan). The laser pointer is a typical pen-shape of 532 nm wavelength laser pointer. In order to ensure the sealing of the cameras, we insert the catadioptric systems and the lasers into a tight waterproof system.

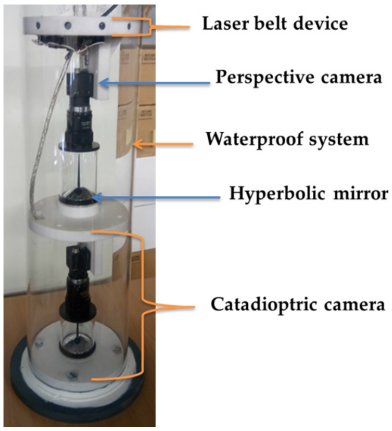


Figure 1. Architecture of the omnidirectional system composed of stereo catadioptric system and the laser belt.

3.2. Off-Line Stereo Catadioptric System Calibration

To reproduce a 3D scene from stereoscopic capturing images taken in different angle of view, the estimation of each model of the camera is essential. This process, called calibration, offers the intrinsic and extrinsic parameters that describe the physical functioning of the camera. In fact, it allows finding relationships between the pixel-wise dimensions on the image and the actual metric measurements. It can be achieved by having information either about the scene geometry or on the movement of the camera or simply by using calibration patterns. In this paper, we use the third possibility: the system of cameras and a checkerboard pattern of known geometry were placed underwater and a set of stereo images were taken for different orientations of the test pattern. The process of calibration consists on the definition of 3D corners of the pattern test and the extraction of the corresponding 2D projections in the image. The 3D and 2D corresponding coordinates constitute the data that will be processed in the calibration algorithm to determine an accurate model of each camera and the extrinsic measurements related to the stereo system and the laser pointing devices.

We use the generalized parametric modelisation described and detailed by Scaramuzza et al. in [45]. They propose a generalized approximation that relates 3D coordinates of a world vector and its 2D image projection. It consists on a polynomial approximation function denoted "f" which is defined through a set of coefficients $\{a_0; a_1; a_2; a_3; \ldots .\}$ as given in expression (Equation (1)). The calibration process is done once-time in off-line mode and the resulting coefficients will be used as variable inputs for the 3D reconstruction algorithm. The 2D coordinates (u, v) represent the pixel coordinates of a feature point extracted from an image with respect to the principal point (u_0, v_0). This latter is considered, at first, as the image center to generate a first parameter estimation. Then, by minimizing the re-projection error using the iterative RANdom SAmple Consensus (RANSAC) algorithm [46], the exact coordinates of the principal point is determined. This permits to highlight the misalignment between the optical axis of the camera and the axis of the mirror which allow refining the estimation results (Figure 2).

$$f(\rho) = a_0 + a_1\rho + a_2\rho^2 + a_3\rho^3 + a_4\rho^4,\ \rho = \sqrt{u^2 + v^2} \tag{1}$$

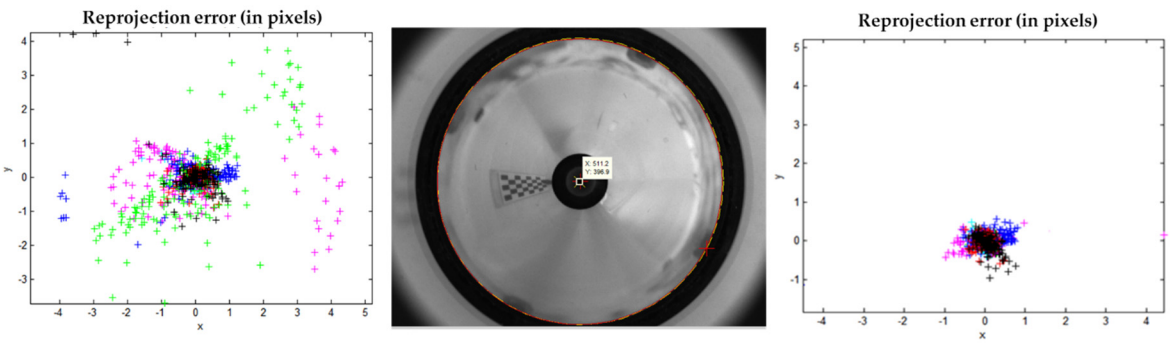


Figure 2. Reprojection error before and after principal point estimation.

Additional metric parameters of the stereoscopic vision system are measured such as the Baseline (the distance between the centers of both cameras) and the distance "Z_{laser}" of the laser belt with respect to each camera center. For underwater application, we use a plexi waterproof system of thickness "e" and radius "d". The estimated calibration parameters consider that all the system (stereo cameras + lasers + waterproof device) is a compact system, which means that the polynomial function "f" takes into consideration light refraction through the plexi. In fact, it permits to reconstruct a world point outside the waterproof system from its 2D image projection.

4. Description of the Proposed 3D Underwater Shape Reconstruction System

4.1. The Flowchart of the Algorithm

Underwater 3D reconstruction algorithms are, generally, divided into two principle parts: the first step aims to detect and extract 2D coordinates of the most interesting points and the second one consists of the 2D/3D transformation and then the 3D restitution of the underwater shape points coordinates. Figure 3 illustrates the flowchart of the processed methods used in our algorithm.

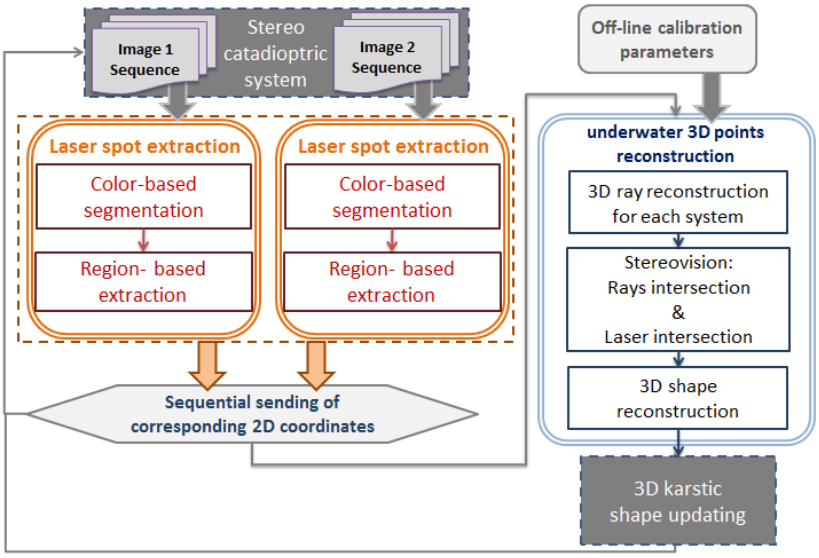


Figure 3. Flowchart of the proposed 3D reconstruction algorithm.

The system inputs are the omnidirectional images acquired both in the same time at each position of the exploration robot and the calibration parameters which were estimated off-line and used as input variable data. The first processing block includes initially morphological operations based on color pixel information in order to highlight the laser spot projection with respect to the rest of the image and then a specific based-region process applied to identify and then extract the laser projection 2D coordinates in each image. This block is designed twice for each camera image sequence. The second processing module (3D underwater shape reconstruction) contains first the 3D ray reconstruction based on 2D coordinates and the calibration parameters. Then, the 3D shape of the karst is determined using the 3D coordinates obtained by the intersection of the stereo rays and the 3D coordinates obtained by the intersection of generated ray vectors corresponding to each system and the laser beams directions. After each 3D underwater shape generation, new stereoscopic images are acquired and sent to be processed. Since the first and second processing blocks are totally independent, we can execute them in a parallel way: either on a fully hardware processing or through a Sw/Hw design. In the following sections, we present in more details the key processing steps: laser spot extraction and underwater restitution of the 3D shape coordinates. Then, we present the complete workflow of the processing chain.

4.2. Laser Spot Detection and Extraction Algorithm

The RGB images, acquired from each catadioptric system, present the inputs of the laser spot extraction block. The proposed algorithm aims to detect regions of green-laser projections in order to extract their image positions. In order to achieve good detection accuracy, different methods were tested on our images in order to define the adequate algorithm. The green-colored laser spots, used in our vision system, are analyzed through their projection on the image. Experiments show that the color of the laser spot detected by camera is not only green; its center is very bright ($200 \leq (R, G, B) \leq 255$).

Therefore, multiple tests were done in different lighting conditions in order to analyze the laser projection and define the expression that permits to highlight the laser spots in the image. For the extraction of the 2D coordinates of the laser projection, we proceed first with a color-based binarisation of the RGB image. Expression (2) permits to highlight pixels of which the green component value "G" is more important than the red "R" and the blue "B" ones.

$$\left\{\begin{array}{l} if((G > 00) \text{ and } (2xG - R - B > 70)) \text{ or } ((G > 110) \text{ and } (2xG - R - B > 120)) \\ pixel = 1; \\ else \\ pixel = 0; \end{array}\right\} \quad (2)$$

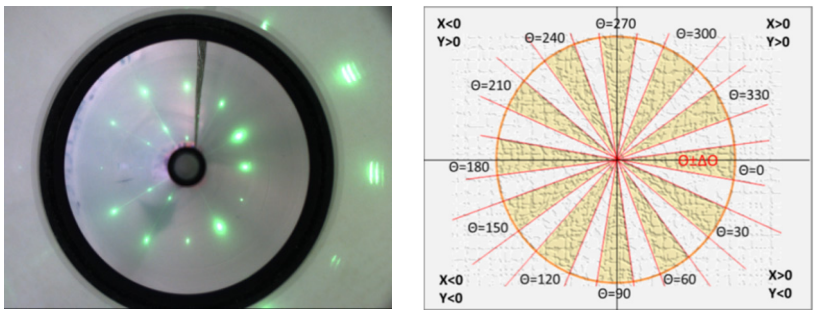
Because of water physical properties, a laser beam can undergo non-useful reflections before reaching the object. The RGB image contains green-colored pixels that represent correct and useful laser spot projections and some wrong projections caused by reflections of laser beams when encountering reflecting surfaces such as the plexi waterproof tank (Figure 4a). In order to eliminate these projections, that could mistake the shape reconstruction, we propose to filter them using geometric considerations. Based on multiple experiments, we noted that:

(1) laser projections that are not located inside the mirror projection are considered false ones;

(2) laser projections that do not belong to the predefined regions of interest are considered as wrong projections that should be detected and transformed into black-colored pixels;

(3) if there is multiple projection of the same laser beam on the same direction, the farthest cloud points is the real projection since it represents the extremity of the laser beam when encountering an object. In contrast, all along the laser trajectory, existing small cloud of points are those of unnecessary reflections which will be eliminated later.

For that, Cartesian coordinates of each white pixel are determined and corresponding polar coordinates (r, θ) are calculated. Eliminated pixels are those that have a ray "r" > "Rmirror" and of whose angle of inclination "θ_r" do not belong to predefined regions "$\theta \pm \Delta\theta$" where "θ" belongs to the predefined set (Figure 4b). If there are multiple cloud points in the same region, the one who is characterized by the bigger ray is considered.



(**a**) (**b**)

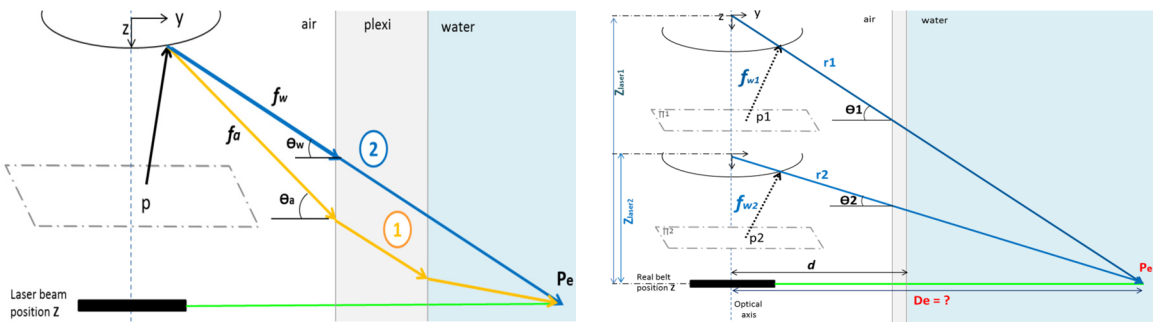
Figure 4. Result of laser spot extraction process: (**a**) RGB image; (**b**) Region-based image classification.

For the extraction of the gravity center of a cloud of points, we determine the average value of all pixels belonging to the line crossing the cloud of points and defined through its inclination with respect to the horizontal (Equation (3)) in which C represents the number of the image column, $p(x_i, y_i)$ represents the pixel coordinates of each point belonging to the line and $c(x_c, y_c)$ is its center and n represents the number of points.

$$\left\{\begin{array}{l} x_c = \frac{\max\{p(x_i,y_i)_{i=1:n}\} - \min\{p(x_i,y_i)_{i=1:n}\}}{2} / C \\ y_c = \frac{\max\{p(x_i,y_i)_{i=1:n}\} - \min\{p(x_i,y_i)_{i=1:n}\}}{2} - (x_c \times C) \end{array}\right. \quad (3)$$

4.3. 3D Underwater Shape Reconstruction Algorithm

The proposed 3D reconstruction algorithm depends on the configuration of the stereo cameras and the laser belt device. In Figure 5a, we illustrate the difference between the use of air-calibration parameters or underwater-calibration ones. In the first case, light refraction effects are considered in the algorithm of ray tracing which consists on following the trajectory of a light when crossing different environment (of different refraction indexes). In the second case, light refraction effects such as those due to water properties (liquid type, refraction index, temperature, pressure . . .), and those due to environment variability (air/waterproof tank/water) are taken into consideration in the estimated polynomial function 'f_w'. In fact, the stereo system calibration is performed in the environment of test with all actual conditions. This equivalence was proved, experimentally, by applying both cases on the same pairs of images by using underwater-calibration and using air-calibration with ray tracing relations.



(**a**) (**b**)

Figure 5. 2D/3D underwater geometric restitution: (**a**) Illustration of air and underwater reconstruction; (**b**) illustration of underwater stereo reconstruction.

The system of vision was calibrated using underwater images of the checkerboard pattern test (Figure 2). The relationship is established between 3D points and the 2D corresponding ones extracted from acquired images. The obtained system of parameters corresponds to a polynomial function noted "f_w" (case of underwater calibration). This relationship (Equation (4)) permits to correspond to each pixel point $p(x, y)$ a 3D vector r (X_r, Y_r, Z_r). In case of stereo system (Figure 5b), each laser is represented by a projection on image^1 and on image^2. For each pixel point respectively (p_1 and p_2), a ray vector is determined using the corresponding sensor parameters respectively (f_1 and f_2). In case of using the laser position information "Z_{laser}", we determinate the distance "D_e" between the real point "P_e" and the system optical axis using refraction angle "θ" corresponding to each projection as expressed in (Equation (5)). This later is determined using the ray vector coordinates. The resulting 3D coordinates of the real point are given by (Equation (6)). In case of using only the stereo information, the coordinates of the real point "P_e" is determined using the intersection of the stereo ray vectors "r^1" and "r^2" using the relations of (Equation (7)).

In our work, we exploit the redundancy between the passive stereo restitution (camera^1+camera^2) and the active monocular restitution (camera+lasers) to refine the result of reconstruction. In fact, 3D coordinates of the world point "Pe" are determined using three following approximations:

(1) Intersecting the ray vector "r^1"corresponding to the laser projection on image plane ($\prod^1$) to the laser beam position ($Z_{laser}{}^1$) with respect to camera^1

(2) Intersecting the ray vector "r^2"corresponding to the laser projection on image plane ($\prod^2$) to the laser beam position ($Z_{laser}{}^2$) with respect to camera^2

(3) Intersecting the ray vector "r^1"corresponding to the laser projection on image plane ($\prod^1$) to the ray vector "r^2"corresponding to the laser projection on image plane ($\prod^2$).

$$p\begin{bmatrix} u \\ v \end{bmatrix} \leftrightarrow \vec{r}\begin{bmatrix} X_r = u - u_0 \\ Y_r = v - v_0 \\ Z_r = f(\rho) \end{bmatrix}, \ \rho = \sqrt{X_r^2 + Y_r^2} \tag{4}$$

$$D_e = \left(\frac{Z_{\text{laser}}}{\tan(\theta)}\right), \ \theta = \arctan\left(\frac{Z_r}{\sqrt{X_r^2 + Y_r^2}}\right) \tag{5}$$

$$P_e\begin{bmatrix} X_e \\ Y_e \\ Z_e \end{bmatrix} = \begin{bmatrix} Z_r \\ sign(Y_r)\cdot\sqrt{{D_e}^2 - {X_e}^2} \\ Z_{\text{laser}} \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_{r1} \\ Y_{r1} \\ Z_{r1} \end{bmatrix} + \left[\frac{(Y_{r2} - Y_{r1})\cdot\sin(\theta_2) - (Z_{r2} - Z_{r1})\cdot\cos(\theta_2)}{\cos(\theta_1)\cdot\sin(\theta_2) - \cos(\theta_1)\cdot\sin(\theta_2)}\right]\cdot\begin{bmatrix} 0 \\ \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} \tag{7}$$

4.4. Complete Workflow of the Proposed System

In Figure 6, we propose the workflow of the complete processing methods used in our algorithm and showing all the steps with images.

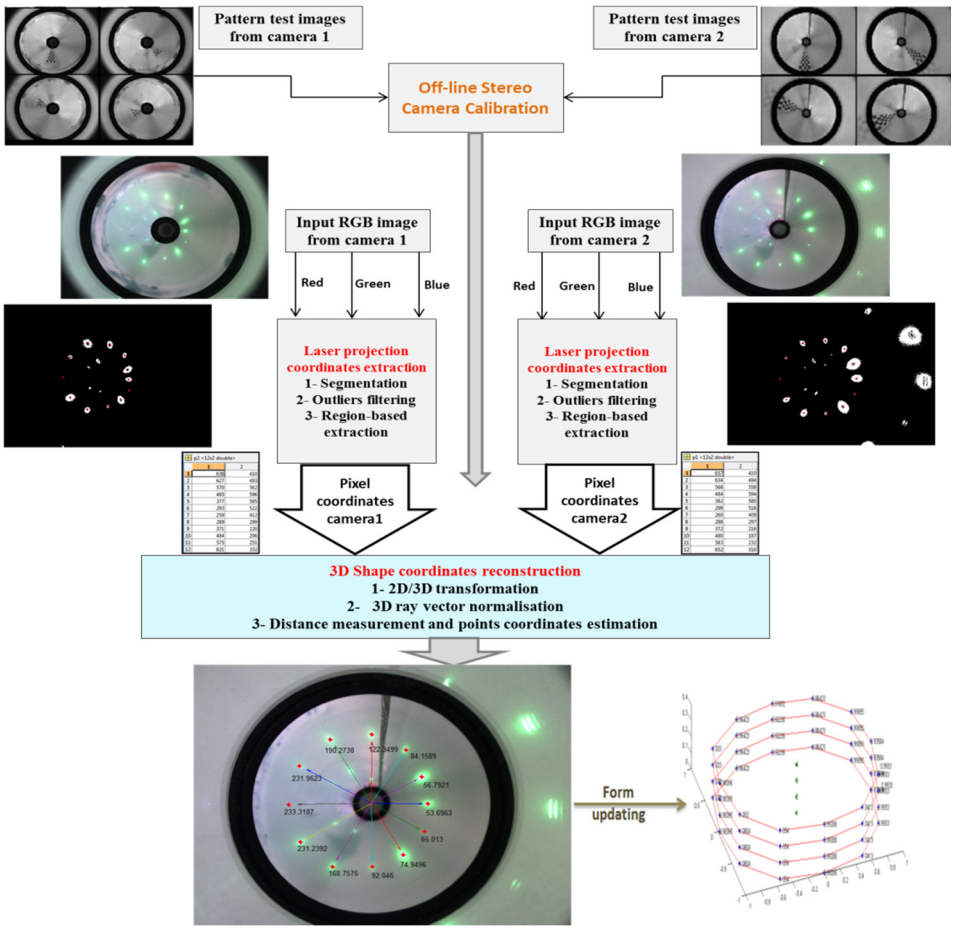


Figure 6. Full workflow of the processing algorithms of the 3D shape reconstruction system.

The workflow gives the processing chain of the following steps: off-line calibration applied on a set of stereo images of pattern test placed on both cameras common field of view, image segmentation that highlights the laser projections, outliers elimination based on region-constraints, laser spot extraction along the laser beam direction, and the underwater 3D shape reconstruction algorithm based on distance measurements between the optical center and each laser projection.

5. Hardware Design of the Proposed Algorithm

5.1. XSG Design Flow Diagram

For rapid prototyping, a development environment in which the hardware and software modules can be designed, tested, and validated is required. For this purpose, the Xilinx System Generator tool for DSP (Digital Signal Processor) development was used. It permits a high level implementation of DSP algorithms on FPGA circuit. It consists on a graphical interface under Matlab/Simulink environment used for modeling and simulating dynamic systems. A library of logic cores, optimized for FPGA implementation, is provided to be configured and used according to the user's specifications. When implementing System Generator blocks, the Xilinx Integrated Software Environment (ISE) is used to create "netlist" files, which serve to convert the logic design into a physical file and generate the bitstream that will be downloaded on the target device through a standard JTAG (Joint Test Action Group) connection. Simulation under Matlab Xilinx System Generator allows us testing the FPGA application with the same conditions with the physical device and with the same data width of variables and operators. The simulation result is "bit-to-bit" and "cycle" accurate, which means that each single bit is accurately processed and each operation will take the exact same number of cycles to process. Thus, there will be almost no surprises when the application is finally running in the FPGA.

Figure 7 presents the processing flow of System Generator tool from the hardware architecture design to the generation of the reconfiguration file used for the RTL (Register Transfer Level) design generation and the co_simulation process. The architecture designed using XSG blocks will be simulated under Simulink to compare calculation results obtained by the Matlab code algorithm and those obtained by the designed hardware module.

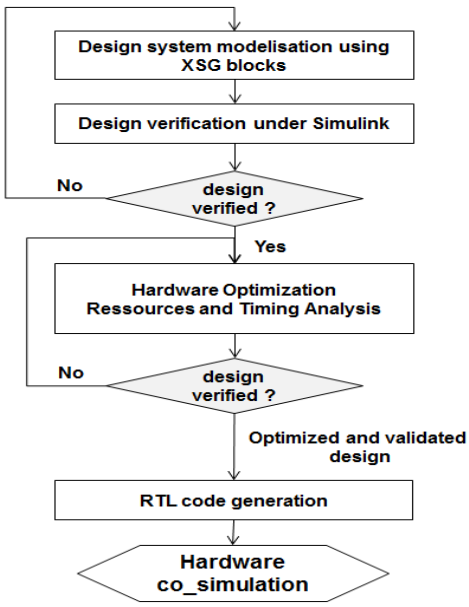


Figure 7. XSG design flow diagram.

5.2. Hardware Optimization

During hardware design, the principal aim is to trade between speed, accuracy and used resources. The algorithm of 3D reconstruction uses multiple division operations and square root approximations. The COordinate Rotation DIgital Computer (CORDIC) cores offered by Xilinx to realize the mentioned functions were tested under different configuration. However, this block provides only a rough approximation of the required result especially when a large fractional precision is needed [47]. The square root results were not enough accurate and although the satisfying CORDIC division result, its large size and low execution speed make it less advantageous. Therefore, custom hardware architectures for the cited functions are proposed.

Using XSG development tool, maximum precision could be obtained by configuring all blocks, between the gateways, to run with a full output type. However, this could lead to an impossible synthesis result. Word-length optimization is a key parameter that permits to reduce the use of FPGA hardware resources while maintaining a satisfactory level of accuracy. In the case of square root approximation, word-length is hard to compute since output format increase by the looped algorithm feedbacks. For that reason, successive simulation reducing iteratively outputs word length is made until achieving the minimum world length while ensuring the same maximum precision.

5.3. Hardware Architecture of the proposed 3D Shape Reconstruction System

The hardware architecture of the proposed algorithm, presented in Figure 8, follows the same processing chain as the workflow algorithm. It consists of two main stages: spot detection and 3D reconstruction.

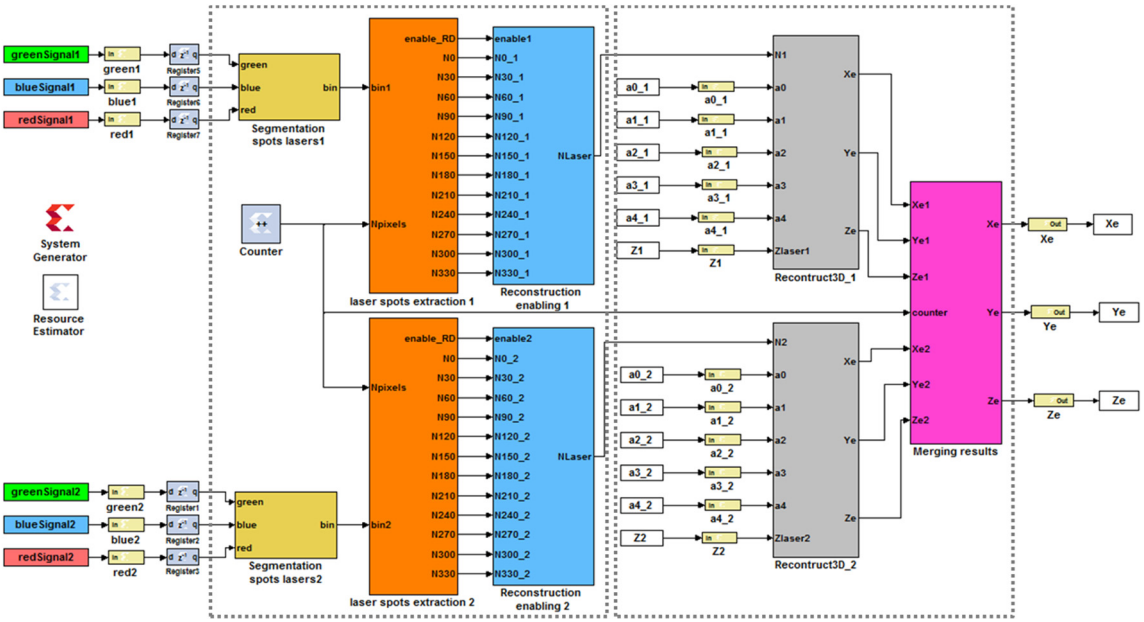


Figure 8. Hardware architecture of the proposed algorithm.

The first subsystem presents two pipelined sequential treatment blocs of the stereo images. Each bloc provides twelve 2D coordinates of the laser projections of the corresponding image. These coordinates are sent to the next processing bloc when all image pixels are covered and checked. The second stage takes as inputs the corresponding stereo coordinates and the calibration parameters. It permits to generate the 3D coordinates corresponding to the world points identified through the laser spots projections. An additional bloc is inserted to synchronize the reading of corresponding 2D coordinates and their sequential sending to the reconstruction process. We take profit from stereovision to optimize the reconstruction result. Two pipelined architectures are designed to simultaneously treat

the stereo RGB images and reconstruct the 3D laser spots world positions for each system. The system output is the average of the two generated results. A description of the functioning of each subsystem will be given in the following sections.

5.3.1. Hardware Architecture of the Laser Spot Detection Algorithm

The laser spots detection algorithm consists on highlighting the laser projection on the stereo RGB images in order to easy the extraction of the corresponding 2D coordinates. The first step is the binarisation of the RGB images based on pixel color information. White pixels will represent the laser projections and black ones the background. In Figure 9, we propose the hardware architecture of the segmentation step, which correspond in fact to the expression (Equation (2)) given in Section 4.2.

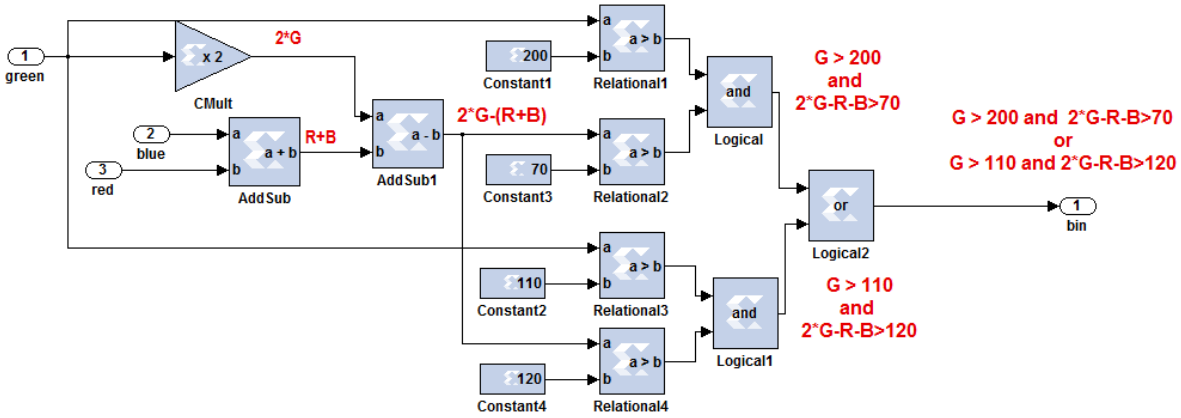


Figure 9. Hardware architecture of the segmentation block.

The segmentation results demonstrate that the binarisation presents some outliers. These errors should be eliminated to avoid its effect on the final restitution of the karst 3D shape. Figure 10 illustrates the subsystems of the laser spots extraction module, which consists of respectively Cartesian2Polar transformation, region-based process to eliminate outliers and the extraction of laser positions. The image center (x_0, y_0) is estimated during the calibration process.

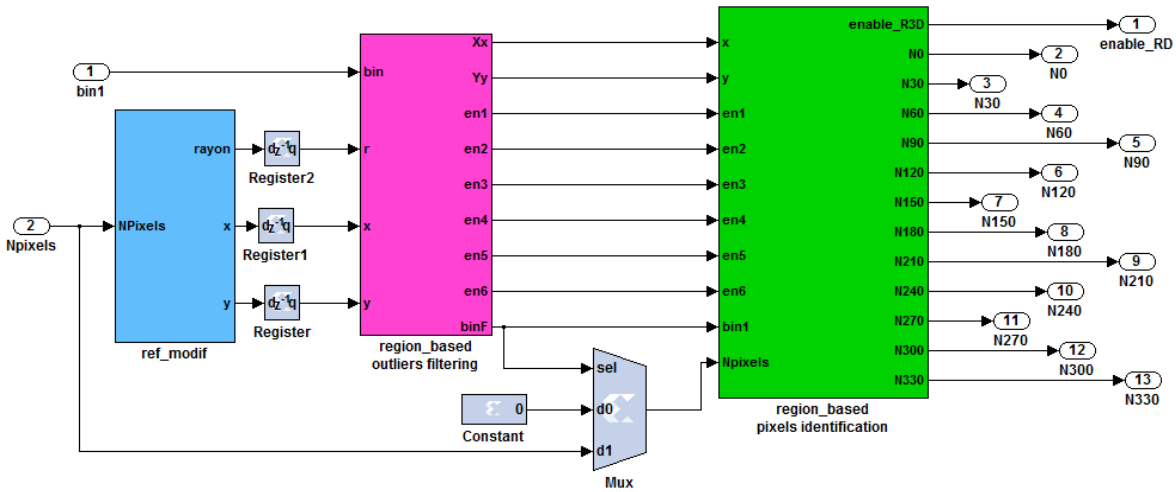


Figure 10. Hardware architecture of the region-based laser spot extraction module.

In order to eliminate wrong laser projections, we use an approach based on region constraints. All pixels that are white ('bin' = '1'), that are inside the mirror projection {'*r*' < 'Rmiroir'}, and that the relation between x and y coordinates is close to one of the predefined set, or belonging to the region

where the x coordinates is negative ($x < 0$) are considered to form a cloud points. A selection is applied to choose which laser projection will be considered as an object and which will be merged with the image background. This is done by analyzing the values of the polar and Cartesian coordinates of each laser pixel. For a pixel belonging to a direction whose inclination angle is "θ", its (x, y) coordinates are proportional to the "arctangent (θ)" value. In our work, we use twelve laser pointers, which let us define twelve directions corresponding to θ Є {0, 30, 60, . . . , 330} corresponding to the arctangent value Є {±0.577, ±1.732} and pixels belonging to the vertical axis {$-3 < y < 3$} and horizontal axis {$-3 < x < 3$}. Because of the symmetry of the arctangent function, two faced region is described by the same arctangent value. To differentiate between the two regions, we analyze the signs of the (x, y) coordinates (Figure 4b). Twelve selection blocks are designed to specify each region of the image (Figure 11). Each block contains the region characteristics defined to sort pixels which are identified as laser projection. For the gravity center calculation, we use a "MCode" blockset inserted to determine the median position value of all entering pixel values of a cloud points for each region.

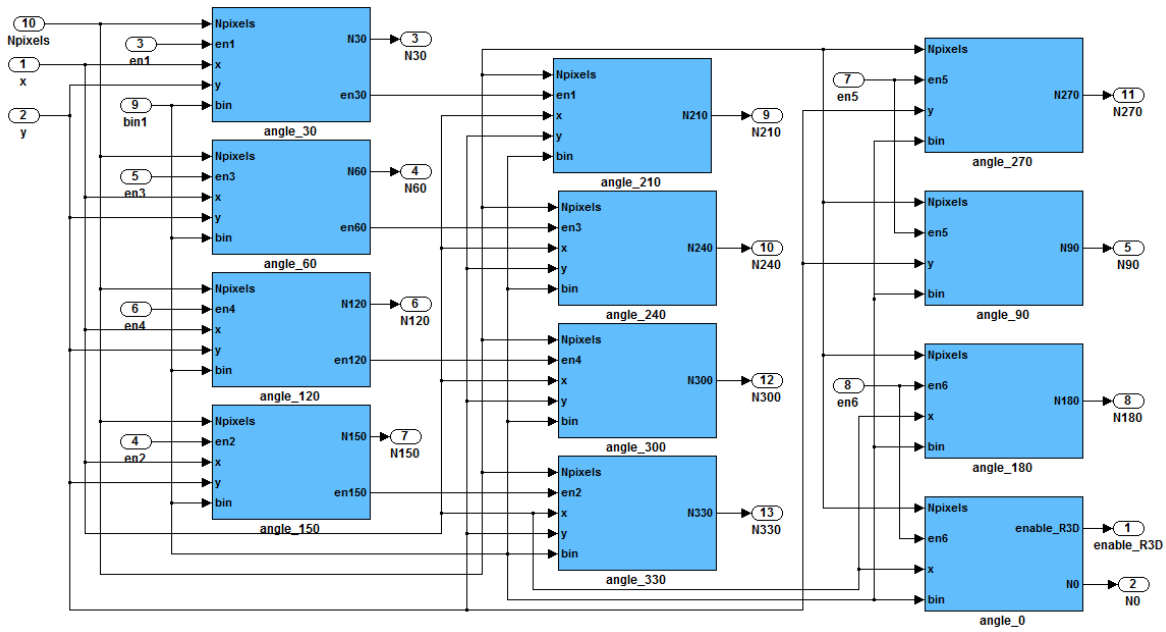


Figure 11. Hardware architecture of laser spots extraction modules.

5.3.2. Hardware Architecture of the 3D Reconstruction Algorithm

As mentioned in the workflow, the algorithm of 3D shape reconstruction takes as inputs the 2D coordinates of laser spot projections from both images and the calibration parameters of both catadioptric systems. In order to take profit of pipeline, two hardware systems are designed to be executed in a parallel way for each image. Each system represents the hardware module of the transformation from air 2D projections to the 3D underwater coordinates of the world point. In Figure 12, we present the successive under sub-blocks of the 3D shape reconstruction module. The 2D/3D transformation algorithm consists on four steps, which are respectively:

(1) Scaramuzza calibration function block;
(2) Normalization block;
(3) Refraction angle estimation block;
(4) Distance measurement and real point coordinates estimation block.

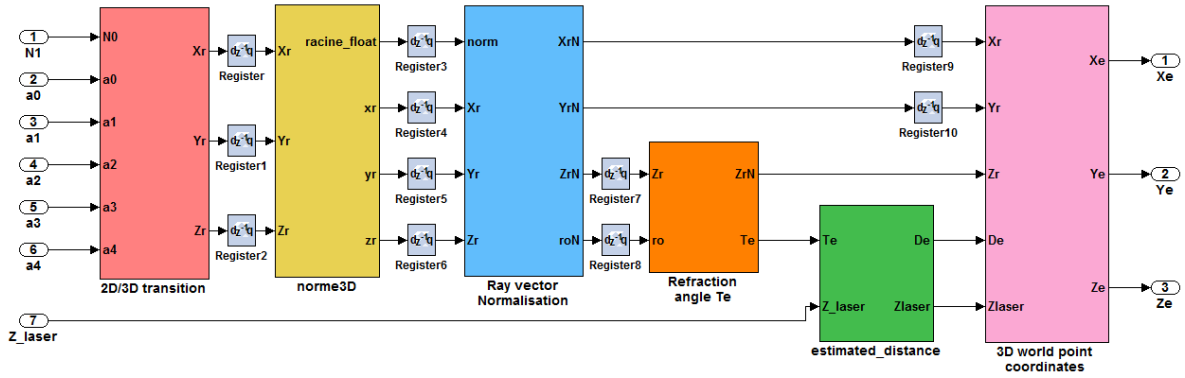


Figure 12. Hardware sub-systems of the 2D/3D transformation.

(a) Scaramuzza calibration function block

The first bloc takes as input the 2D extracted coordinates and the system calibration parameters. The estimated polynomial function is applied on the pixel coordinates to generate the coordinates of the ray vector crossing the water as given in the expression (Equation (4)). Figure 13 gives the hardware blocks arranged to perform this task.

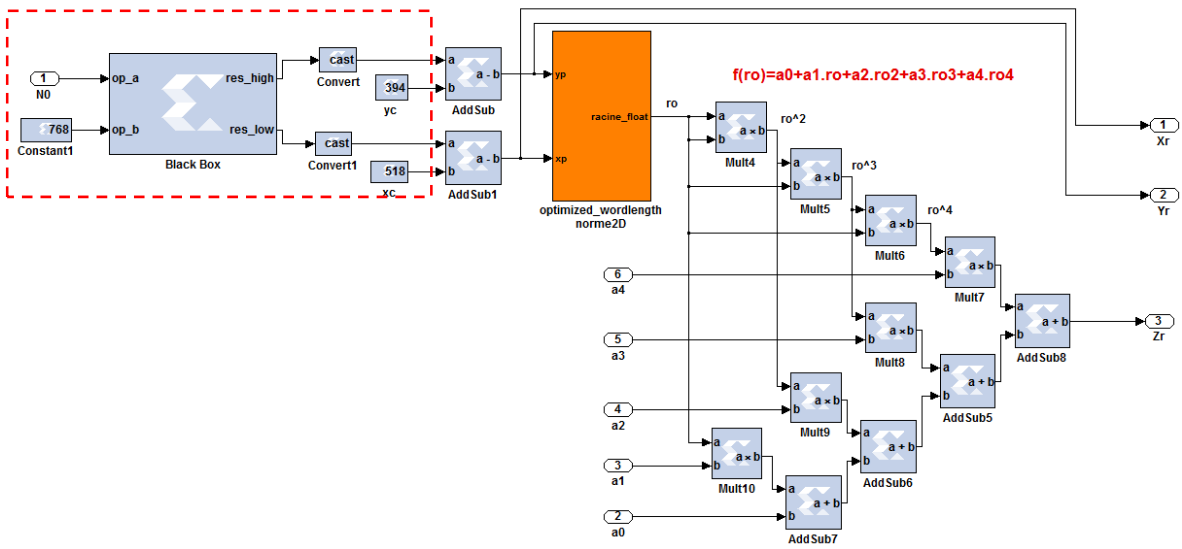


Figure 13. Hardware architecture of the calibration application function bloc.

The pixel coordinates are obtained by dividing the extracted position "N0" by the number of lines in the image. A hardware block called "Black Box" is inserted to perform the division. This primitive permits the designer to reuse an existing VHDL code that divides two integers of 20 bits and gives as outputs: the quotient "y" representing the y-axis position and the rest "x" representing the x-axis position. The calibration function is applied on "ρ" value which is, in fact, the radius value corresponding to the 2D coordinates (x_p, y_p): such that $x_p = u - u_0$ and $y_p = v - v_0$ (red box). The center image coordinates values (u_0, v_0) were obtained in the calibration process as described in Section 3.2.

A sub-block named "norme2D" is used to approximate the square root function. The algorithm processes as follows: we convert the floating number into an integer and a rest. We use a "Black-box" block that calls a VHDL code to provide the square root of the integer part "r". We assume then that the square root "ρ of the floating number is between "r" and "$r + 1$". At first, we compare "$\rho^2 - r^2$" and "$(r + 1)^2 - \rho^2$" to take the closer value as the first boundary and its half the second one. Successive

iterations are applied on each chosen boundaries in order to be as close as possible to the exact square root "ρ" of the input floating number. The number of iterations is limited depending on a defined accuracy. The last sub-system is designed to calculate the 3D norm of the ray vector coordinates to be sent to the further processing using the same approach.

(b) Normalization block

The normalization function, given by the expression (Equation (8)), applies the Euclidian 3D norm of the ray vector "r" using as inputs its 3D coordinates. A 32-bit divider is called by the "Black-box" element to perform the division and a 32-bit approximation subsystem to perform the square root function.

$$\begin{bmatrix} X_r^N \\ Y_r^N \\ Y_r^N \end{bmatrix} = \frac{1}{\sqrt{X_r{}^2 + Y_r{}^2 + Z_r{}^2}} \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix} \tag{8}$$

(c) Refraction angle estimation block

Having the normalized ray vector coordinates crossing the water ($X_r{}^N$, $Y_r{}^N$, $Z_r{}^N$), we calculate the refraction angle "θ_e" since it represents its inclination with respect to the normal to the waterproof surface (Figure 5) in which the variable "r_o" represents the square root of ($X_r{}^N + Y_r{}^N$). A 32-bit VHDL code was programmed and tested under ISE development tool and called through a "Black-Box" block.

(d) Distance measurement and real point coordinates estimation block

The "P_e" coordinates are proportional to the 3D ray vector r (X_r, Y_r, Z_r), to the distance between the laser projection and the optical axis of the stereo cameras "D_e" and to the laser beam position. We re-use two 32-bit division subsystems of real numbers division and a 32-bit HDL code for the square root approximation function. The outputs ports are changed from std_logic_vector into Fix_32_14 data type.

6. Experimentation and Performance Evaluation

The main functionality of to the developed system is to make 3D reconstructions of a confined dynamic underwater environment while giving accurate metrics based only on acquired images (1024 × 768 pixels). In the validation and evaluation of the developed system accuracy, we use examples of simple geometric shapes with known dimensions. The evaluation tests were made in a specific water pool built in our laboratory of a width of 3 meters. Tests could be made on the pool itself or throughout different tubes (forms) introduced in the pool. In each experimental test, dozens of camera positions are used for evaluation. The obtained pairs of images are processed and the estimated distances are compared to the real ones corresponding to these known camera positions. In Figure 14, we present an example of simulation results performed in the hexagonal pool. The lasers pointing at the cross section of the object are projected in the stereo images (Figure 14a). The binary stereo images, obtained using the thresholding expression, highlights the laser projections with respect to the background (Figure 14b). Only the defined image-regions are covered in order to extract unique 2D coordinates of the laser spot on each region. Laser projections belonging to the same image region are matched (Figure 14c) and sent to the triangulation process to determine the 3D distances corresponding to each laser (Figure 15). The mean error scale of distance measurements, for the different objects used in experimentation, is of 2.5%. For the hexagonal object, the mean square error is about 5.45 mm^2 which is quite satisfactory regarding bad resolution of underwater images and light refraction effects.

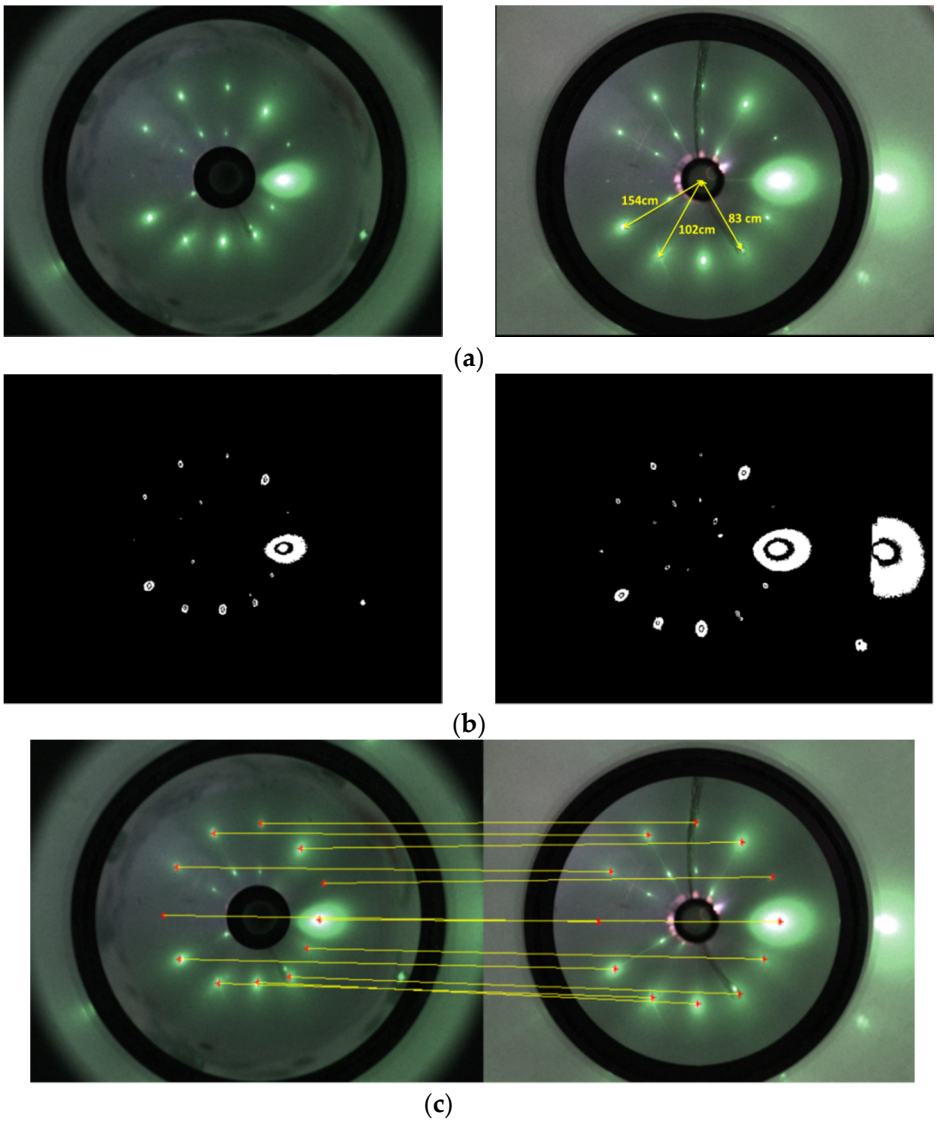


Figure 14. Simulation results (**a**) stereo RGB images; (**b**) Segmented images; (**c**) stereo matching.

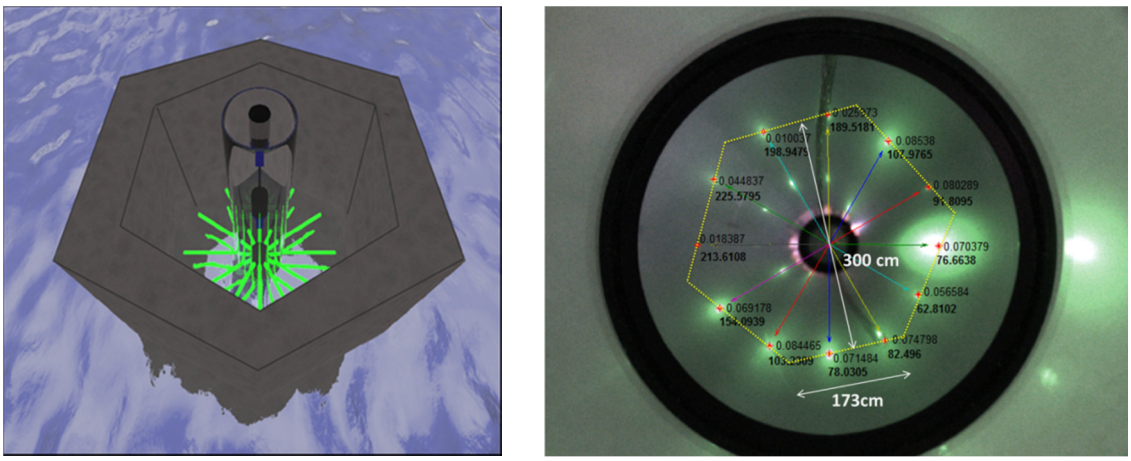


Figure 15. Simulation results: 3D distances measurements and distance error.

Error between estimated distances and real measurements could be caused either because of calibration error, XSG blocks optimization or because of laser spot center detection errors. Analyses of the spot center accuracy detection influence on 3D distance results were also performed. Figure 16 illustrates distance measurements with respect to the gravity-center positions for different real measurements (up to 480 cm). To be able to evaluate the effect of the error on the spot center detection (in terms of pixel position) on the resulting distance, we can use the curve of Figure 16. This curve gives the distance function of spot center position (polar coordinate $r = \sqrt{x^2 + y^2}$). For a reference spot distance, we can determine the distance error by varying the pixel coordinates. Table 2 gives an example of error evaluation for three points with a variation of Δx = 1 pixel and Δx = 10 pixels. For distances measures up to 480 cm, the maximum relative distance error is about 1.3%.

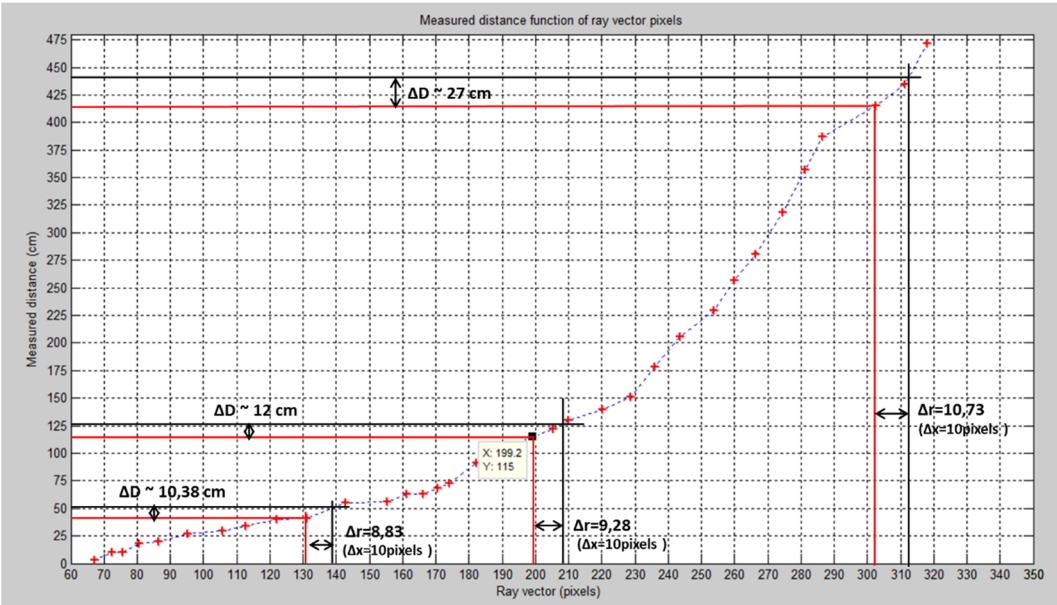


Figure 16. Effect of spot center accuracy on distance measurements.

Table 2. Example of error evaluation.

Relative Measurements		Point 1 D = 40 cm	Point 2 D = 115 cm	Point 3 D = 415 cm
	Δr	0.88	0.93	0.97
Δx = 1 pixel	ΔD	0.81	0.96	2
	%	1.02%	1.01%	1%
	Δr	8.83	9.28	10.73
Δx = 10 pixels	ΔD	10.38	12	27
	%	1.26%	1.10%	1.07%

6.1. Hardware Co_Simulation Results Comparison

In hardware co-simulation concept, the main treatment (processing module) is really implemented in the FPGA circuit and data acquisition and visualization are performed using Matlab. The system reads the image from MATLAB workspace; pass it to Xilinx Blockset where the different operations are performed and the result is sent over back to MATLAB workspace. The data used to evaluate the final system accuracy corresponds to the hardware co-simulation results which are really obtained from the FPGA circuit and which represent the final data that will be generated by the system.

To implement the design on FPGA, ISE tool and core generator automatically generate a programming bit file using the System Generator blockset. A system block is created to point into the configuration file (bit) for the co_simulation. The generated co_simulation block (Figure 17) replaces

the XSG model (Figure 8) and will be loaded on the FPGA device through the JTAG cable by closing the loop after connecting the board (Xilinx ML605).

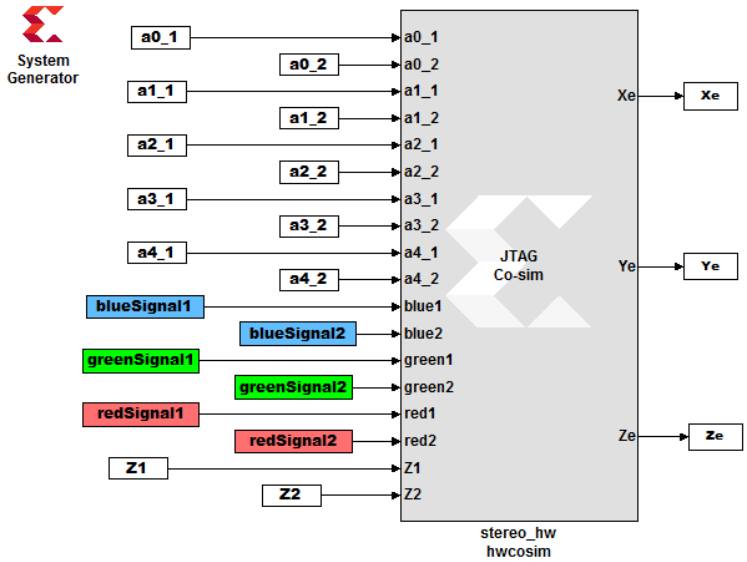
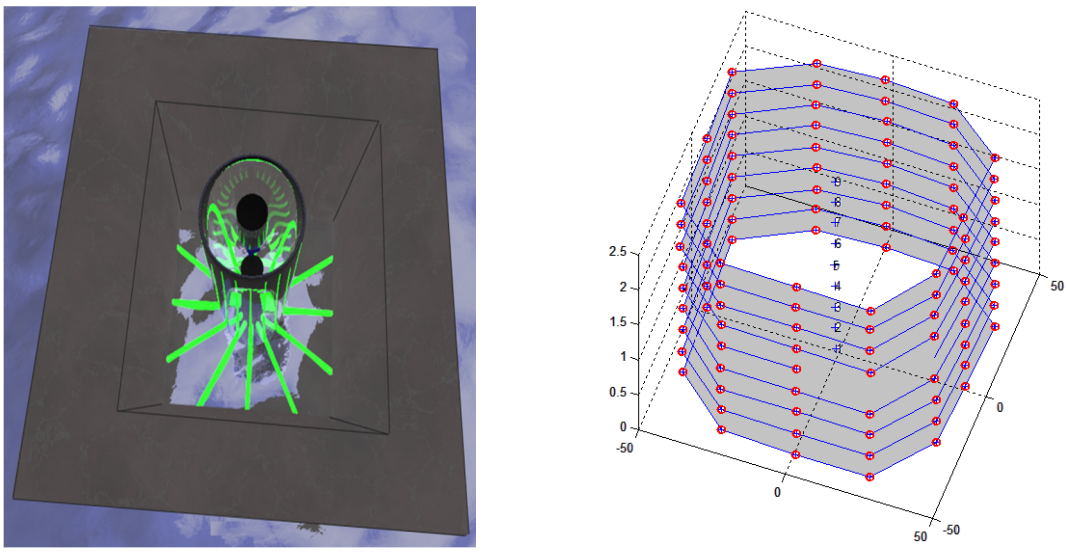


Figure 17. Hardware co_simulation bloc.

All along the robot movement, a stereo image is acquired and analyzed in order to generate 3D shape of the confined structure using the generated 3D world points. In order to verify if the behavior of the XSG model corresponds to the theoretical Matlab code, results of both models are compared noting that it could be relative errors according to the number of bits used. Figure 18 illustrates an example of square shape sparse reconstruction with both Matlab code (blue markers) and the proposed FPGA implemented system (red markers). An average error of about 0.46 cm is obtained between the software algorithm and the XSG architecture for the object of Figure 18.



(**a**) (**b**)

Figure 18. Hw/Sw accuracy comparison: (**a**) Environment of test illustration; (**b**) 3D shapes reconstruction by simulation.

6.2. System Performances Analysis and Discussion

FPGA implementation is proposed in the aim to be able to respect the real-time application performances requirements. Analysis of the hardware architecture must be done to determine the latency of the complete processing chain. In one hand, we proceed with a timing analysis using the Xilinx "Timing analysis tool" in order to identify and then decrease the latency of the critical paths. In addition, in the other hand, the processing algorithm should perform at least at the frame rate frequency of the camera to satisfy timing constraints.

In our work, by taking profit from pipeline to decrease the latency, the operating frequency has been increased to 275.18 MHz. We used two standard IEEE-1394b FireWire cameras (Point Grey Research Inc., Richmond, BC, Canada). The maximum pixel clock frequency could reach 66.67 MHz, which corresponds to a pixel period of 15 ns. The maximum FPS (frames per second) of the camera is 30 FPS at a maximum resolution of 1024 $\times$ 768. This means that between two frames, it exists a period of T = 33 ms. The system should respond during this period in order to satisfy the application constraints. The proposed XSG architecture for the pre-processing and laser spots extraction presents a total latency of 1024 $\times$ 768 clock cycles (the resolution of the image). The 3D reconstruction module, which is designed to be looped twelve times to generate the 3D coordinates of the 12 laser pointers presents a total latency of 1200 clock cycles.

For a maximum clock frequency our system can attain the following performances:

(1) Maximum throughput of 275 Mega-pixel per second
(2) Pre-processing and laser spots extraction execution time = 2.86 ms
(3) 3D reconstruction execution time = 0.00436 ms
(4) Total execution time considering the pipeline = 2.86 ms

The complete execution time of the proposed architecture can reach the value of 2.86 ms which is inferior to the 33 ms constraint related to the used camera characteristics. This permits to use more sophisticated camera with higher frame rate and resolution and to go further in the reconstruction speed and precision. In addition, the remaining time can be exploited for more treatments for the robot control. In our work, we use a prototype of twelve lasers devices inserted with the stereo cameras in order to validate the proposed 3D shape reconstruction system. More lasers could be introduced in the aim to increase the number of points representing the shape to be reconstructed. This permits to obtain a more accurate structure of the karst without any effect on the execution time, given that the detection blocks relative to the laser projection zones work in a parallel way. For example, when using a distance angle of 0.5° (instead of 30°) between neighboring lasers, the hardware architecture of the reconstruction module will take 0.26 ms (720 $\times$ 100 $\times$ 3.634) and the total latency will note change. With a robot navigation speed of 2 ms^{-1}, for example, the proposed system is able to generate a 3D shape every 5.72 mm. Examples of some FPGA implementations of real-time 3D reconstruction algorithms and their main characteristics are presented in Table 3.

Table 3. Stereo vision FPGA implementation performances.

Ref.	System Application	Sensing Device Performances	Implementation Target	Time Performances
[41]	Depth map 3D representation	Asus Xtion (30 Hz) Kinect camera (30 Hz)	Intel Core i7 3.4 GHz CPU	21.8 ms/frame
[48]	Real-time 3D surface model reconstruction	Stereoscopic CCD cameras	Xilinx Virtex-5 FPGA (LX110T)	170 MHz 15 fps
[36]	Video-based shape-from-shading reconstruction	Stereo digital cameras	Xilinx Virtex-4 FPGA (VLX 100)	60 MHz 42 s/frame
[2]	Real-time 3D reconstruction of stereo images	Ten calibrated stereo image pairs	Virtex-2 Pro FPGA (XC2VP30)	100 MHz 75 fps
[49]	Real-time 3D depth maps creation	Typical cameras 30 fps	Xilinx Virtex-2 (XC2V80)	30 fps
[50]	Stereo matching for 3D image generation	3 CCD cameras 320 × 320 pixels 33 MHz video rate	ALTERA FPGA (APEX20KE)	86 MHz 0.19 ms/frame
Our system	Real-time 3D shape reconstruction from stereo images	Stereo catadioptric system 1024 × 768 pixels 66 MHz video rate	Xilinx Virtex-6 (VLX240T)	275 MHz 30 fps 2.86 ms/frame

7. Conclusions

The use of rapid prototyping tools such as MATLAB-Simulink and Xilinx System Generator becomes increasingly important because of time-to-market constraints. This paper presents a methodology for implementing real-time DSP applications on a reconfigurable logic platform using Xilinx System Generator (XSG) for Matlab.

In this paper, we have proposed an FPGA-based stereo vision system for underwater 3D shape reconstruction. First, an efficient Matlab-code algorithm is proposed for laser spot extraction from stereo underwater images and 3D shape reconstruction. Algorithm evaluation shows that a mean error scale of distance measurements accuracy is of 2.5%, between real and estimated measurements, which is quite satisfactory regarding bad resolution of underwater images and light refraction effects. Nevertheless, the software algorithm takes few seconds to be executed which is not permitted by the real-time application performances. In order to satisfy required performances, we have developed a hardware architecture, focused on the fully use of pipeline, aimed to be implemented on a Virtex-6 FPGA. During the design process, optimizations are applied in order to optimize the used hardware resources and to decrease the execution time while maintaining a high level of accuracy. Hardware co_simulation results show that our architecture presents a good level of accuracy. An average error of about 0.0046 m is obtained between the software algorithm and the XSG architecture.

Furthermore, timing analysis and evaluation of the complete design show that a high performance in terms of execution time can be reached. Indeed, the proposed architecture largely respects timing constraints of the used vision system and the application requirements. The developed system supports performance increase with respect to image resolution and frame rate. A further modification and amelioration could be applied in order to fit the 3D structure of the karst. In the end of the paper, experimental results in terms of timing performances and hardware resources utilization are compared to those of other existing systems.

Acknowledgments: The authors would like to acknowledge the support of Olivier Tempier, an engineer in charge of the mechatronics hall of the University, for the experimental setup.

Author Contributions: R.H. developed the algorithm, conceived and performed experiments and wrote the paper; A.B.A. contributed on the hardware design, experimentations and the paper writing; F.C. conceived the prototype and contributed on the algorithm development and experimentations; L.L. contributed on algorithm development and experiments performing; A.M. and R.Z. contributed with general supervision of the project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jin, S.; Cho, J.; Pham, X.D.; Lee, K.M.; Park, S.-K.; Kim, M. FPGA design and implementation of a real-time stereo vision system. *IEEE Trans. Circuits Sys. Video Technol.* **2010**, *20*, 15–26.
2. Hadjitheophanous, S.; Ttofis, C.; Georghiades, A.S.; Theocharides, T. Towards hardware stereoscopic 3D reconstruction a real-time FPGA computation of the disparity map. In Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Leuven, Belgium, 8–12 March 2010; pp. 1743–1748.
3. Calonder, M.; Lepetit, V.; Fua, P. Keypoint signatures for fast learning and recognition. In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 58–71.
4. Zerr, B.; Stage, B. Three-dimensional reconstruction of underwater objects from a sequence of sonar images. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 16–19 September 2010; pp. 927–930.
5. Johnson-Roberson, M.; Pizarro, O.; Williams, S.B.; Mahon, I. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *J. Field Rob.* **2010**, *27*, 21–51.
6. Gokhale, R.; Kumar, R. Analysis of three dimensional object reconstruction method for underwater images. *Inter. J. Sci.Technol. Res.* **2013**, *2*, 85–88.
7. Meline, A.; Triboulet, J.; Jouvencel, B. Comparative study of two 3D reconstruction methods for underwater archaeology. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 740–745.
8. Ladha, S.N.; Smith-Miles, K.; Chandran, S. Multi-user natural interaction with sensor on activity. In Proceedings of the 1st IEEE Workshop on User-Centered Computer Vision, Tampa, FL, USA, 16–18 October 2013; pp. 25–30.
9. Kirstein, C.; Muller, H.; Heinrich, M. Interaction with a projection screen using a camera-tracked laser pointer. In Proceedings of the 1998 Conference on MultiMedia Modeling, Washington, DC, USA, 12–15 October 1998; pp. 191–192.
10. Ahlborn, B.; Thompson, D.; Kreylos, O.; Hamann, B.; Staadt, O.G. A practical system for laser pointer interaction on large displays. In Proceedings of the International Symposium on Virtual Reality Software and Technology, New York, NY, USA, 7–9 November 2005; pp. 106–109.
11. Brown, M.S.; Wong, W.K.H. Laser pointer interaction for camera-registered multi projector displays. In Proceedings of the IEEE International Conference on Image Processing, Barcelona, Spain, 14–17 September 2003; pp. 913–916.
12. Davis, J.; Chen, X. Lumipoint: Multi-user laser-based interaction on large tiled displays. *Displays* **2002**, *23*. [CrossRef]
13. Latoschik, M.E.; Bomberg, E. Augmenting a laser pointer with a diffraction grating for monoscopic 6DOF detection. *Virtual Reality* **2006**, *4*. [CrossRef]
14. Oh, J.Y.; Stuerzlinger, W. Laser pointers as collaborative pointing devices. In Proceedings of the Graphics Interface Conference (GI'02), Calgary, Canada, 17–24 May 2002; pp. 141–149.
15. Olsen, D.R.; Nielsen, T. Laser pointer interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, Washington, USA, 31 March–5 April 2001; pp. 17–22.
16. Kim, N.W.; Lee, S.J.; Lee, B.G.; Lee, J.J. Vision based laser pointer interaction for flexible screens. In Proceedings of the 12th International Conference on Human-Computer Interaction: Interaction Platforms and Techniques (HCII'07), Beijing, China, 22–27 July 2007; pp. 845–853.
17. Meško, M.; Toth, Š. Laser spot detection. *J. Inf. Control Manage. Syst.* **2013**, *11*, 35–42.
18. Chavez, F.; Olague, F.; Fernández, J.; Alcalá-Fdez, R.; Alcalá, F.; Herrera, G. Genetic tuning of a laser pointer environment control device system for handicapped people with fuzzy systems. In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ), Barcelona, Spain, 18–23 July, 2010; pp. 1–8.
19. Hogue, A.; German, A.; Jenk, M. Underwater environment reconstruction using stereo and inertial data. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Montréal, Canada, 7–10 October 2007; pp. 2372–2377.
20. Brahim, N.; Gueriot, D.; Daniely, S.; Solaiman, B. 3D reconstruction of underwater scenes using image sequences from acoustic camera. In Proceedings of the Oceans IEEE Sydney, Sydney, NSW, Australia, 24–27 May 2010; pp. 1–8.

21. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698.
22. Gurol, O.C.; Oztürk, S.; Acar, B.; Sankur, B.; Guney, M. Sparse disparity map estimation on stereo images. In Proceedings of the 20th Signal Processing and Communications Applications Conference (SIU), Mugla, Turkey, 18–20 Apirl 2012; pp. 1–4.
23. Deguchi, K.; Noami, J.; Hontani, H. 3D fundus shape reconstruction and display from stereo fundus images. In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000; pp. 94–97.
24. Moyung, T.J.; Fieguth, P.W. Incremental shape reconstruction using stereo image sequences. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; pp. 752–755.
25. Dey, T.K.; Giesen, J.; Hudson, J. Delaunay based shape reconstruction from large data. In Proceedings of the IEEE Symposium on Parallel and Large-Data Visualization and Graphics, San Diego, CA, USA, 22–23 October 2001; pp. 19–146.
26. Gibson, S.F. Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Heidelberg, Germany, 11–13 October 1998; pp. 888–898.
27. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *In ACM Siggraph Comput. Graphics* **1987**, *21*, 163–169.
28. Hajebi, K.; Zelek, J.S. Structure from infrared stereo images. In Proceedings of the Canadian Conference on Computer and Robot Vision, Windsor, UK, 28–30 May 2008; pp. 105–112.
29. McGlamery, B. *Computer analysis and simulation of underwater camera system performance*; Visibility Laboratory, University of California: San Diego, CA, USA, 1975.
30. Jaffe, J.S. Computer modeling and the design of optimal underwater imaging systems. *IEEE J. Oceanic Eng.* **1990**, *15*, 101–111.
31. Schechner, Y.Y.; Karpel, N. Clear underwater vision. In Proceedings of the IEEE Computer Society Conference in Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 29 June–1 July 2004; pp. 536–543.
32. Jordt-Sedlazeck, A.; Koch, R. Refractive calibration of underwater cameras. In Proceedings of the Computer Vision–ECCV 2012, Florence, Italy, 7–13 October 2012; pp. 846–859.
33. Treibitz, T.; Schechner, Y.Y.; Kunz, C.; Singh, H. Flat refractive geometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 51–65.
34. Kunz, C.; Singh, H. Hemispherical refraction and camera calibration in underwater vision. In Proceedings of the IEEE OCEANS, Quebec City, Canada, 15–18 September 2008; pp. 1–7.
35. Sorensen, S.; Kolagunda, A.; Saponaro, P.; Kambhamettu, C. Refractive stereo ray tracing for reconstructing underwater structures. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Quebec City, Canada, 27–30 September 2015; pp. 1712–1716.
36. Carlo, S.Di.; Prinetto, P.; Rolfo, D.; Sansonne, N.; Trotta, P. A novel algorithm and hardware architecture for fast video-based shape reconstruction of space debris. *J. Adv. Sign. Proces.* **2014**, *147*, 1–19.
37. Bräuer-Burchardt, C.; Heinze, M.; Schmidt, I.; Kühmstedt, P.; Notni, G. Underwater 3D surface measurement using fringe projection based scanning devices. *Sensors* **2015**, *16*. [CrossRef]
38. Moore, K.D.; Jaffe, J.S.; Ochoa, B.L. Development of a new underwater bathymetric laser imaging system: L-Bath. *J. Atmos. Oceanic Technol.* **2000**, *17*, 1106–1117. [CrossRef]
39. Newman, P.; Sibley, G.; Smith, M.; Cummins, M.; Harrison, A.; Mei, C.; et al. Navigating, recognizing and describing urban spaces with vision and lasers. *Inter. J. Rob. Res.* **2009**, *28*, 1406–1433. [CrossRef]
40. Creuze, V.; Jouvencel, B. Avoidance of underwater cliffs for autonomous underwater vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 30 September–4 October 2002; pp. 793–798.
41. Niessner, Z.; Zollhöfer, M.; Izadi, S.; Stamminger, M. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graphics* **2013**, *32*. [CrossRef]
42. Ayoub, J.; Romain, O.; Granado, B.; Mhanna, Y. Accuracy amelioration of an integrated real-time 3d image sensor. In Proceedings of the Design & Architectures for Signal and Image Processing, Bruxelles, Belgium, 24–26 November 2008.
43. Muljowidodo, K.; Rasyid, M.A.; SaptoAdi, N.; Budiyono, A. Vision based distance measurement system using single laser pointer design for underwater vehicle. *Indian J. Mar. Sci.* **2009**, *38*, 324–331.

44. Massot-Campos, M.; Oliver-Codina, G.; Kemal, H.; Petillot, Y.; Bonin-Font, F. Structured light and stereo vision for underwater 3D reconstruction. In Proceedings of the OCEANS 2015-Genova, Genova, Italy, 18–21 May 2015; pp. 1–6.
45. Scaramuzza, D.; Martinelli, A.; Siegwart, R.A. Toolbox for easily calibrating omnidirectional cameras. In Proceedings of the Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 5695–5701.
46. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applicatlons to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
47. Mailloux, J.-G. Prototypage rapide de la commande vectorielle sur FPGA à l'aide des outils SIMULINK - SYSTEM GENERATOR. Thesis report, Quebec University, Montreal, QC, Canada, 2008.
48. Chaikalis, D.; Sgouros, N.P.; Maroulis, D. A real-time FPGA architecture for 3D reconstruction from integral images. *J. Visual Commun.Image Represent.* **2010**, *21*, 9–16. [CrossRef]
49. Lee, S.; Yi, J.; Kim, J. Real-Time stereo vision on a reconfigurable system. *Embedded Comput. Sys. Architectures Model. Simul.* **2005**, *3553*, 299–307.
50. Hariyama, M. FPGA implementation of a stereo matching processor based on window-parallel-and-pixel-parallel architecture. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2005**, *E88-A*, 3516–3522. [CrossRef]

© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (http://creativecommons.org/licenses/by/4.0/).