A real-time stereo vision system for distance measurement and underwater image restoration

Camilo Sánchez-Ferreira, Jones Y. Mori, Mylène C. Q. Farias & Carlos H. Llanos

Journal of the Brazilian Society of Mechanical Sciences and Engineering

ISSN 1678-5878

J Braz. Soc. Mech. Sci. Eng.
DOI 10.1007/s40430-016-0596-5

ONLINE FIRST

Journal of the Brazilian Society of
Mechanical Sciences and Engineering

Special Issue: Journal's 35th Anniversary

Springer

ABCM
Brazilian Society of Mechanical Sciences and Engineering

J Braz. Soc. Mech. Sci. Eng.
DOI 10.1007/s40430-016-0596-5

TECHNICAL PAPER

A real-time stereo vision system for distance measurement and underwater image restoration

Camilo Sánchez-Ferreira[1] · Jones Y. Mori[1] · Mylène C. Q. Farias[1] · Carlos H. Llanos[1]

Received: 1 December 2015 / Accepted: 28 June 2016

Abstract This paper presents the development of an embedded real-time system that performs distance measurement and restoration of underwater images, using stereo vision techniques. To achieve a high performance low-cost implementation, the overall system has been developed using a hardware/software codesign approach. Several hardware modules have been designed to implement the several pixel intensive tasks, such as background image storage, background subtraction, center of mass calculation, and image restoration. On the other hand, less intensive tasks, such as the estimation of the disparity and the distance tasks (performed just once for each image), are executed using an embedded soft processor (Altera Nios II). The developed platform employs a pair of identical CMOS cameras for the stereo vision system, a low-cost FPGA, and an small screen for visualization of the images. In this paper, we describe both the overall design of the system and the calibration procedure used to determine the stereo vision system parameters. The Altera Quartus II was used as a synthesis tool, which estimates that the system consumes 115.25 mW and achieves an output of 26.56 frames per second for images of 800 × 480 pixels. The synthesis results and the measurement precision show that the developed system is suitable for real-time tasks.

Keywords Underwater vision · Image restoration · Image degradation · FPGA · Real-time

1 Introduction

Underwater robots can be used for understanding marine environments, protecting ocean resources from pollution, performing sea bottom surveys, or monitoring and maintaining offshore structures, among others. However, even with modern technologies, there are several challenges in the unstructured and hazardous undersea environment that make it difficult to work underwater. Sending an autonomous vehicle, with limited online communication, into an unknown and unstructured environment requires adding on-board intelligence and the ability to react in a reliable way to unexpected situations. Techniques such as artificial intelligence, neural networking, and fuzzy logic can be used to perform this kind of critical tasks. The embedded sensory system also needs to be robust and reliable, offering information that allows developing tasks related to localization, mapping, simultaneous localization-mapping (SLAM), and navigation. The demand for advanced underwater robot technologies is increasing in the last years, creating a necessity for more specialized and reliable underwater devices. Most commercial underwater robots are remotely operated, being referred to as remotely operated vehicles (ROVs). There are also robots that do not require an operator and are known as autonomous underwater vehicles (AUV).

Technical Editor: Sadek C. Absi Alfaro.

✉ Carlos H. Llanos
llanos@unb.br

Camilo Sánchez-Ferreira
sanchezfer@unb.br

Jones Y. Mori
jonesyudi@unb.br

Mylène C. Q. Farias
mylene@ene.unb.br

[1] Faculty of Technology, University of Brasilia, Brasília, D.F., Brazil

Published online: 06 July 2016

The sensory system of an underwater vehicle faces noisy and unstructured environments. Moreover, technologies such as GPS are not applicable because electromagnetic transmission cannot be used in underwater environments. In this case, vision-based systems are frequently used, even though they are not fully reliable due to the poor visibility. For, example, some ROVs are equipped with TV cameras as a vision system and with a manipulator as the working system. However, it is very difficult to operate a manipulator remotely, given that it does not provide correct distance measurements of the working environment [1]. Other applications in underwater imaging include 3D reconstruction of scene objects [2–4]. As a consequence, improving the performance of computer vision algorithms targeted at unstructured environments, such as underwater, has become an important problem both in academia and in industry. It is worth pointing out that, because of the physical properties of the environment, underwater images suffer from different types of degradations. One of the most important degradations is attenuation, which produces a decrease of the image contrast. The attenuation strength depends on the distance from the object to the camera system. By knowing this distance, one can determine the degradation parameters and restore the images.

Stereo vision systems are used for performing optical-based distance measurements, but these techniques present more challenges than single camera systems. In stereo vision, pre-processing tasks, such as enhancement, filtering, segmentation, and restoration need to be executed twice for each pair of stereo images, duplicating the computational effort. In this case, parallel computing architectures can be used to increase the performance of these kinds of systems and allow them to operate in real time. In the last years, several efforts have been made to develop embedded systems that implement computer vision techniques for applications that face several performance and power consumption challenges, such as mobile robotics, underwater robots, and more specifically AUVs.

Given that several image processing algorithms are computationally expensive, embedded vision systems are time-critical. In underwater applications, an additional challenge arises from the harmful environment. When compared to a monocular system, a stereo vision system allows collecting more information about the environment. More specifically, stereoscopic vision systems are able to extract depth information from a scene using two images acquired from a pair of cameras. To estimate depth, the algorithm first solves a correspondence problem, which consists of determining the corresponding points in both images. Once the cameras are calibrated and the correspondences are measured, the algorithm executes a 3D reconstruction for generating a 3D map of the scene.

In this work, we develop a real-time system for both distance measurement and image restoration, using a hardware/software co-design approach implemented in an FPGA-based (field programmable gate array) platform. The hardware module performs the most demanding tasks, whereas the soft processor is responsible for the non-critical tasks. A high flexibility in the design can be achieved using this co-design methodology, leading to better performance results, both in terms of surface area and execution time. Therefore, the main contributions of this paper are the following: (a) implementation of an efficient image restoration architecture, using a low complex restoration algorithm that does not incur in loss of image quality; (b) implementation of an underwater image restoration system that uses an integration of a stereo vision system and a distance estimation technique; (c) proposal of a low cost and efficient embedded FPGA implementation; (d) validation of the restoration process using the Structural Similarity (SSIM) index; and (e) validation of the hardware architecture, obtained by comparing the experimental results with results of previous works.

The paper is organized as follows. Section 2 gives a short theoretical background of stereo 3D reconstruction and underwater image degradation, while Sect. 3 gives a review of the techniques available in the literature. Section 4 explains how the vision system is calibrated to estimate distances and how the hardware/software project was designed. Section 5 describes both the design methodology and the architectures developed. Synthesis results, restoration and measurement's validation are presented in Sect. 6. Finally, Sect. 7 presents the conclusions of the work.

2 Background

2.1 Stereo vision

The 3D-Reconstruction process, which employs stereo vision, consists of determining the disparities between a pair of images, generated by two cameras placed in different locations in space. Knowing these locations, we can compute a 3D map of the scene by calculating disparities and generating a disparity map. This section presents a method for computing the disparity map and the geometric analysis that supports 3D-Reconstruction.

The most usual stereo vision systems are built using two cameras, as presented in Fig. 1. To easily determine the optical characteristics of the stereo system pair, identical cameras are generally placed at the same plane, with parallel optical axes and the same optical parameters. The mounting setup shown in Fig. 1 is known as epipolar

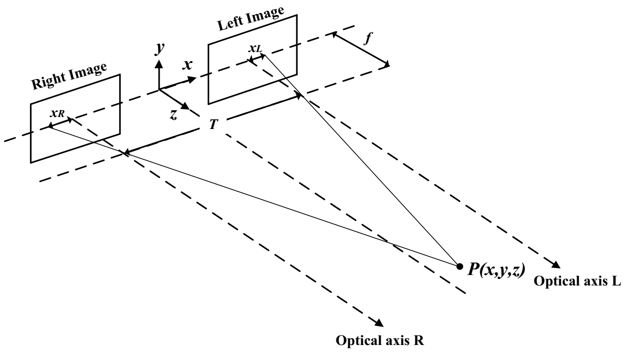

Fig. 1 Epipolar geometry

geometry. Calibration of stereo vision setup is considered completed when the focal axis of each camera is at the same vertical coordinate *y*, which simplifies the computation of the disparity map [5, 6].

Considering a calibrated stereo vision system as shown in Fig. 1, the variables and parameters of the stereo vision setup are defined as the following:

- *f*: focal length of the cameras (the focal length must be equal for both cameras);
- $P(x, y, z)$: real scene point;
- x_L: *x* coordinate at left image;
- x_R: *x* coordinate at right image;
- *T*: distance between optical axis *L* and *R*;
- It must be considered that *y* coordinate is the same for both images.

The value $(x_L - x_R)$ is the disparity between two images in an specific point $P(x, y)$. In the literature, there are two classes of algorithms for computing the disparity:

1. Correlation-based: Algorithms in this class are based on measures of differences between images, like for example SAD (sum of absolute differences), SSD (sum of squared differences), and CC (cross-correlation).
2. Feature-based: Algorithms in this class uses differences between determined features of objects in the image, such as corners, lines, countours, skeletons, object's area, among others.

In general, correlation-based and feature-based algorithms are computationally demanding. Correlation-based algorithms are executed pixel-to-pixel. So, even for small images, this kind of algorithm demands a considerable processing time. On the other hand, feature-based algorithms search for pre-determined features or characteristics in each image, which also demands a good amount of time. To decrease the time required to compute disparities, several hardware implementations have been proposed, as detailed in Sect. 3.

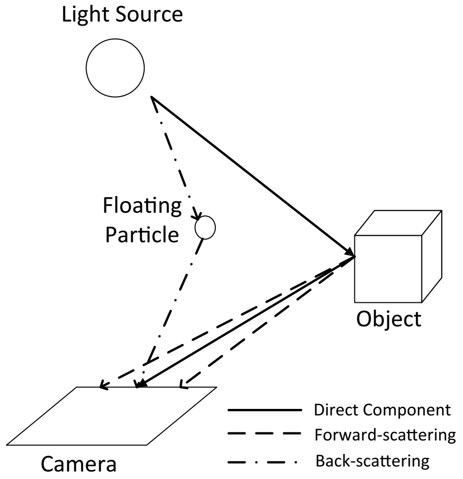

Fig. 2 Light propagation in a real environment

2.2 Underwater image degradation

Light interacts with the environment through two physical processes, namely: (1) absorption and (2) scattering. Absorption is characterized by the attenuation of power that happens as light travels in the medium. Scattering refers to the deflection of light caused by particles in the medium [7]. According to the Lambert–Beer law, we can model the power loss caused by these two processes as an exponential, as described by the following equation:

$$I = I_o.e^{-(a+b).r}, \tag{1}$$

where *I* is the irradiance at the position *r*, *a*, and *b* are the absorption and scattering coefficients of the medium, respectively. Assuming an isotropic and homogeneous medium, the total attenuation coefficient *c* is composed by the sum of *a* and *b*, as shown by the following equation:

$$I = I_o.e^{-c.r}. \tag{2}$$

Typical attenuation coefficients for deep ocean water, coastal water, and bay water are 0.05 m^1, 0.2 m^1, and 0.33 m^1, respectively [8].

On the other hand, several image formation models have been proposed. For instance, McGlamery's work [9] laid out the theoretical aspects of the optical image formation model, whereas Jaffe [10] extended the model and applied it in the design different subsea image acquisition systems [7]. Currently, the Jaffe–McGlamery model is the most complete model available. According to this model, the underwater-acquired image can be represented as a linear superposition of three light components (see Fig. 2): (1) the direct light component (light reflected by the object that travels directly to the imaging system); (2) the forward-scattering light component (light reflected by the object that has been scattered at a small angle); and (3) the back-scattering light component (light reflected by objects that are not on the target scene, for example due to floating particles).

The total irradiance ET is given by the equation [9, 10]:

$$E_T = E_d + E_f + E_b, \tag{3}$$

where E_f and E_b represent forward-scattering and back-scattering components, respectively. E_d represents the direct component, also represented by Eq. 2. In this work, we only consider the direct component as the main cause of image degradation.

3 Related works in stereo vision and image restoration implemented using embedded systems

The works discussed in this section are related to the following subjects: (a) stereo vision algorithms implemented in hardware, (b) applications of stereo vision, and (c) underwater vision systems implemented with embedded systems based on FPGA platforms. This section describes works that propose hardware implementations to these problems, using for example FPGAs. In the literature, there are several works that use image restoration techniques for underwater environments, but there are only a few embedded implementations that are based on FPGA.

With regard to the first subject (*the stereo vision systems implemented in hardware*), a hardware implementation of a stereo vision technique was proposed by Lazaros et al. [11]. Jin et al. [12] implemented a full pipelined stereo vision system that provides a dense disparity image with real-time sub-pixel accuracy. The entire stereo vision process, which includes rectification, stereo matching, and post-processing, is performed using a single FPGA-based platform without any external devices. In this work, the hardware implementation runs 230 times faster than a software implementation operating on a conventional PC.

Kalomiros and Lygouras [13, 14] proposed a system with two stereo accelerators implemented using reconfigurable hardware. The authors presented a comparison between the SAD (Sum of Absolute Differences) algorithm and a dynamic programming technique (both running on 640 × 480 pixels images), obtaining outputs of 162 and 81 frames per second, respectively. Georgoulas and Andreadis [15] presented the design and implementation of a real-time hardware-based stereo vision system. In their work, a new algorithm that mixes SAD with fuzzy logic was used, obtaining outputs with 1713 frames per second when considering images with 320 × 240 pixels. Kalomiros and Lygouras [16] used a co-processor approach to compute the stereo disparity with FPGAs. Ambrosch et al. implemented a system using an Altera's FPGA that generates a dense disparity map [17]. The system is based on a SAD algorithm and is able to provide a throughput of 136 frames per second when working with 330 × 375 pixels images.

In a more recent work from Gardel et al. [18], SAD and SSD algorithms implemented in FPGA were used to obtain a disparity map, producing a throughput of 50 frames per second for images with 2 Mega-pixels (Mpixels). In their work, a feature-based algorithm was used to determine the disparities. On the other hand, a previous work [19] used a background subtraction technique to detect the target object and generate segmented images. Afterwards, the coordinates of the center of mass (x, y) of this object were determined. After performing these two steps (for both right and left images), the disparity ($x_L - x_R$) is computed. Once the disparity is computed, the distance between the scene and image planes can be estimated.

Murphy at al. have used the Census Transform [20] to calculate the disparity, producing outputs of 150 fps for images with 320 × 240 pixels. Hadjitheophanous at al. [21] developed a system for computing disparity maps in real time using a low-cost FPGA. At the same time, Banz et al. [22] implemented a technique known as SGM (semi-global matching), achieving a throughput of 30 frames per second for a VGA resolution (640 × 480 pixels). Perri et al. [6] used a SAD technique to compute disparity maps of images with 512 × 512, obtaining a throughput rate of 25.6 frames per second.

As can be observed, in stereo vision the disparity calculation (for stereo vision) is complex and, generally, obtained using Census Transform, SAD, SSD, or dynamic programming techniques, among others [13, 15, 17]. Mathematical operations need to be performed efficiently to be used in real-time stereo vision systems. Otherwise, our work uses a simple and fast architecture for disparity calculation, which is based on a simple background subtraction technique. The proposed technique detects an object in the scene using a pipelined approach that subtracts an image from its background (stored in memory) pixel by pixel. Then, the center of mass of the object is computed and this information is used to estimate the disparity.

With respect to the second subject (*the applications of stereo vision*), there are several applications in which an embedded implementation in FPGA-based platforms was used for mobile robotics. Mori et al. [23] achieved real-time performance using an FPGA for developing a vision-based sensor for robotics applications. Kalomiros and Lygouras [13] implemented a SLAM (simultaneous localization and mapping) algorithm using a hardware-based stereo vision system. Jia et al. [24] implemented a miniature intelligent mobile robot system, with a trinocular stereo vision system for autonomous navigation, using a single FPGA chip. On the other hand, Villalpando et al. [25] proposed a stereo computation co-processing system for mobility applications that was optimized for fast throughput, being implemented on a single Virtex 4 LX160 FPGA. Iwata and Saneyoshi [26] implemented a forward obstacle detection system using a stereo vision technique with good detectability rate and distance accuracy. Their system produces a dense disparity image and is based on a region-based matching algorithm implemented on an FPGA-based real-time stereo processing system. In general, the available applications (using FPGAs) achieve good results, with respect to time and accuracy. However, with respect to previous works, our approach has the advantage of using a co-design methodology, which was not explored by other approaches, allowing for a high flexibility design and leading to a better performance in terms of area and time. Additionally, the calibration process (used for distance measurement) provides an easy way to adapt to several underwater environments.

Considering the third subject (*the underwater vision systems and restoration task issues*), Bonin and Burguera [27] presented an extensive survey of components, techniques, and methods used to build underwater vision systems. In their paper, they described the phenomena that affects image formation in underwater environments. In addition, the authors described illumination techniques and light sources, including lasers. Finally, the paper introduced techniques for improving the quality of underwater images. The authors described a technique to overcome undesired scatter in images using polarized light, which is similar to the work presented by Schechner et al. [28]. Additionally, Yoshida [29] presented a review of the available embedded system used in underwater robots, which includes their hardware specifications.

In underwater vision systems, degradation processes affect the captured image and a restoration has to be performed afterwards. Hardware architectures are a good alternative for performing this restoration process. For instance, Iwata and Saneyoshi [30] have investigated problems in FPGA-based systems arising from resource constraints. In this case, using an iterative image restoration algorithm, they showed how to manipulate the original algorithm to adequate it to an FPGA implementation. The consequences of these manipulations, such as loss of image quality, have been studied. The authors presented performance results obtained from an actual implementation on a Xilinx FPGA.

Seda et al. [31] have proposed a high performance digital architecture design that implemented a non-linear image enhancement technique. In the proposed FPGA-based architectural design, systolic, pipelined, and parallel design techniques are effectively used to achieve a real-time processing. In this context, estimation and folding techniques are used in the hardware design to achieve a faster, simpler, and more efficient architecture. The video enhancement technique is implemented using a Xilinx's multimedia development board that contains a VirtexII-X2000 FPGA and is capable of processing approximately 66 Mpixels per second.

For the implementation of underwater vision systems and underwater image restoration, previous works [30, 31] presented both an image restoration system and an image enhancement system. However, these works are not targeted to underwater environments. The restoration algorithm proposed by Seda et al. [30] had to be modified to be used in underwater environments, causing a significant loss of the output image quality. In contrast, our work also presents an efficient image restoration architecture that uses the distance information to restore the contrast lost because of underwater attenuation. The proposed approach uses a low complex restoration algorithm that does not cause losses in the image quality.

Finally, it can be observed that in the context of (a) stereo vision problems, (b) applications of stereo vision approaches, (c) underwater vision, and (d) restoration of underwater images, several works target the complexity of the implementations, without providing a good trade-off in terms of complexity, power consumption, image quality, and throughput. Our approach offers a complete underwater image restoration system, which is based on stereo vision, to calculate the object/camera distance. It uses a hardware/software coding approach based on an FPGA-based embedded system. Therefore, when compared to other techniques, the proposed approach has several advantages, achieving a good trade-off with respect to simplicity, efficiency, power consumption, throughput, and cost.

4 System design and implementation

4.1 Calibration process

The calibration procedure is performed to determine the relationship among distance and disparity for a specific point in the scene. Some of the stereo vision system parameters, such as distortions of lens and the focal length (f), can be estimated from calibration. However, the main objective of calibration procedure used in this work is not to estimate these parameters.

Figure 3a shows the stereo system mount setup with two identical cameras, whereas Fig. 3b shows the overall system. Each point in the calibration table is separated 2.54 cm. To calibrate the system, three steps must be followed:

1. Place an object at a known position;

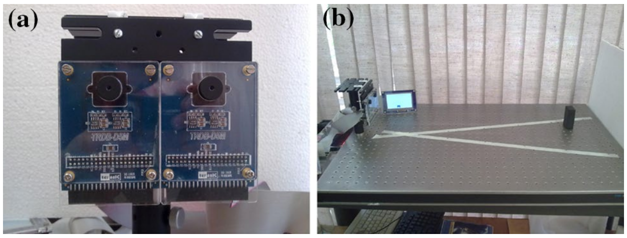

Fig. 3 **a** Parallel stereo system mount setup with two identical cameras. **b** Overall stereo system placed over an optical table for calibration

2. Use the background subtraction method to estimate the center of mass of the target object for each image;
3. Repeat the previous two steps for all calibration points.

Figure 4a presents the common field of view (fov), where the calibration points were located. The system is able to compute the distance efficiently only for the objects placed inside the common fov. Figure 4b shows the setup used for the stereo vision system calibration, for which a set of 146 points was used like a reference. Three different positions in y axis (vertical coordinate) were used for each calibration point. Therefore, a total of 438 points were used in the calibration process. In this case, the main objective of the calibration process consists of finding a relation between the computed disparity and the real distance between the object and the cameras. This relation can be calculated easily by estimating the best curve for the experimental data.

Figure 5 presents the function for distance versus disparity. The curve fitted to the experimental data returns the following function:

$$z = 13170 * (x_L - x_R)^{-1.035}. \tag{4}$$

This fit presented a root mean-square error (RMSE) of 0.667, what shows the good quality of the curve fit. Also, Fig. 5 shows that larger distances present lower dispersion of the calibration data if compared with smaller distances. This is because the object is smaller in the image for larger distances and this produces smaller errors in the calculation of the center of mass.

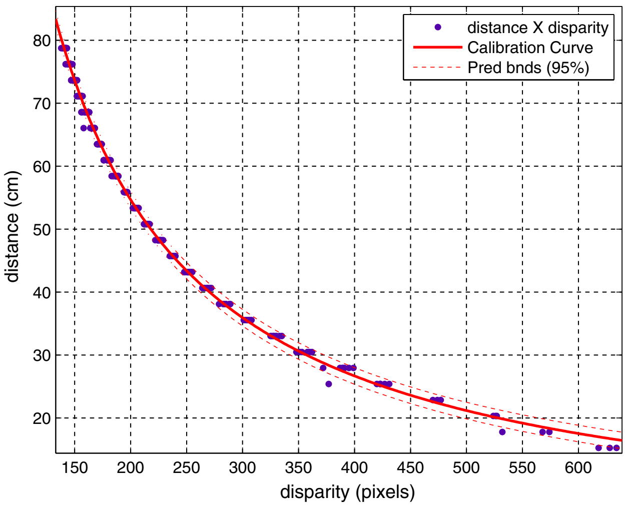

Fig. 5 Calibration curve. It shows the real distance vs. the measured disparity by the stereo system

4.2 Hardware/software partition

The proposed system must be able to: (a) locate the desired object using a background subtraction algorithm in both cameras; (b) determine its position (center of mass) in both cameras; (c) estimate the distance from the target object to the robot; (d) restore the images. Table 1 describes each of these tasks (a to d) describing them by steps and sub-steps. Also, in the last column, the table shows how many times per frame each step/sub-step must be performed (for a resolution of 800 × 480 pixels).

The hardware/software partition is performed based on the data shown in Table 1. Pixel intensive tasks must be performed in hardware to achieve real-time execution. Less intensive computations, performed only a few times per frame, can be executed by software using the embedded processor. In our system, steps/sub-steps 7 and 8 were

Fig. 4 **a** The field-of-view for each camera and the common field-of-view. **b** Calibration points inside the common field-of-view

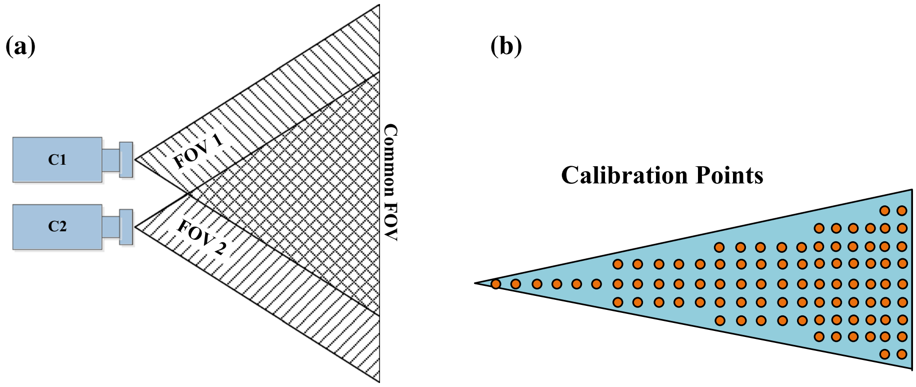

Table 1 Algorithms description

	Step	Sub-step	Description	Occurrences (per frame)
1	Background subtraction	Acquisition control	Control the CMOS camera, synchronizing the acquisition for each pixel	(800 × 480) = 380,000
2		SRAM storage	Write all pixels, one by one, to the SRAM memory	380,000
3		SDRAM/SRAM readout	Synchronize pixels, reading from background (SRAM) and current (SDRAM) images	2 × 380,000
4		Pixels subtraction	Subtract each corresponding pair of pixels	380,000
5		Threshold	Set values 0 or 1 depending on difference values	380,000
6	Center of mass	–	Compute the center of mass of the detected object	1
7	Distance estimation	Disparity	Compute the disparity between the right and left images	1
8		Distance	Estimate the distance using Eq. 4	1
9	Restoration	–	Restore images, pixel by pixel	2 × 380,000

selected to be executed in software, while the other remaining steps/sub-steps were directly mapped in hardware modules.

5 The system architecture

Figure 6 shows the global system architecture. Both the LCD Control and the CMOS Control blocks were provided by the board manufacturer. All other blocks were developed for this architecture, with variations from previous works presented in [19, 32]. One pixel per clock cycle is sent by each camera, and each processing block is synchronized using the pixel clock signal. The background subtraction block acquires an image to be used as a background and saves it in the external SRAM. After that, the background subtraction block performs a subtraction between the background image and the new (current) image. Then, after receiving the pixel stream and the pixel positions of the target object, the center of mass block determines the coordinates of the center of mass. The soft-processor (NIOS II) also receives the pixel clock signals using them to determine when to initiate the disparity and distance computations. Finally, the (x, y) position of the center of mass is sent back to the Nios II processor, which is responsible for computing the disparity and estimating the distance using the function presented in Fig. 5 and Eq. 4. Therefore, the restoration block receives the pixels and performs the mathematical operation shown in Eq. 5. To implement the stereo vision approach, the CMOS control, background subtraction, and center of mass operations must be replicated for each camera.

Again, in the proposed approach, the hardware module executes the more critical calculations, such as the background subtraction and the center of mass tasks that are performed for the whole image. On the other hand, tasks that are executed just once for each image, such as disparity and distance estimation, are considered non-critical (with respect to time performance) and are executed in the software.

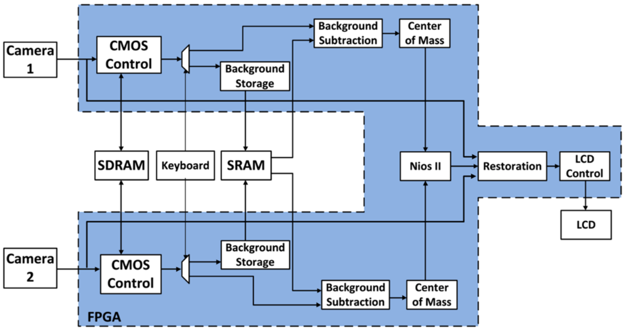

Fig. 6 Block diagram of the system hardware architecture

6 Results

6.1 Distance measurement

To evaluate the precision of the proposed system, we tested it with eleven objects. Table 2 shows the actual distances of objects and the distances estimated by the system. These results show errors of between 0.6 and 2.8 % for most cases. But, for a set 4 points (marked with *), the object was not inside the common fov, which generated additional measurement inaccuracies and errors between 10.6 and 16.5 %. These results show that the calibration process used in this approach has a good performance.

Table 2 Object distance measurement for 11 objects

Actual distance (cm)	Estimated distance (cm)	%Error
27.9	31.2	10.6[a]
38.1	44.6	14.6[a]
40.6	39.9	1.8
43.2	42.4	1.9
53.3	63.8	16.5[a]
53.3	53.6	0.6
68.6	70.8	3.1
76.7	77.7	1.29
78.7	93.3	15.7[a]
81.3	82.9	1.9
81.3	83.6	2.8

[a] In these points, the object is not inside the common fov, causing disparity measurement errors

6.2 Restoration algorithm

We performed some tests to evaluate the restoration performance of images degraded by light attenuation. The restoration is computed using the following equation:

$$I_r = I_i.e^{c.r}, \tag{5}$$

where I_r, I_i, c, r are the restored image, the input image, the attenuation coefficient, and the estimated distance, respectively. The first line of Fig. 7 shows three degraded images, for which an attenuation coefficient characteristic of deep ocean water ($c = 0.05$/m) is used. The second line of Fig. 7 shows the result of applying the restoration formula (Eq. 5) to each of the degraded images.

To assess the quality of the restored images, we use the structural similarity (SSIM) index that is a very simple full-reference metric. For the SSIM index, a higher value corresponds to a better image quality.

The graphics depicted in Fig. 8a, b show the SSIM values for both degraded and restored images. These graphs show that the restored images have higher quality than the distorted images, showing that the proposed restoration technique has a good performance. More specifically, the SSIM values for degraded images (see Fig. 8a) are very low when compared with SSIM values for restored images (see Fig. 8a).

6.3 Synthesis and performance results

In this work, Verilog HDL has been used for developing all the architectures and Altera Quartus II EDA tool was used for synthesis. Synthesis results for all architectures in Fig. 6 are shown in Table 3. On the other hand, Table 4 shows the overall performance information, such as power consumption, maximum frequency, and throughput achieved by the system.

The throughput shown in Table 4 indicates that the system operates in a real-time manner. Notice that the system is capable of locating an object and determining its distance 26 times in one second, i.e., with a real throughput performance of 26 frames per second. It is worth pointing out that the throughput is limited by the maximum frequency achieved by the center-of-mass block, since the architectures were developed to give one output pixel per clock cycle. The hardware/software partition scheme, explained in Sect. 4.2, is efficient enough to achieve the throughput goals. The use of FPGAs resources make the system very flexible, allowing designers to further enhance it by implementing other kinds of algorithms.

For comparison purposes, Table 5 shows the device, frame size, and throughput used by similar works. Due to the differences in the image size used by the different works, the results had to be converted from fps to Mpixels to make possible to compare them with the

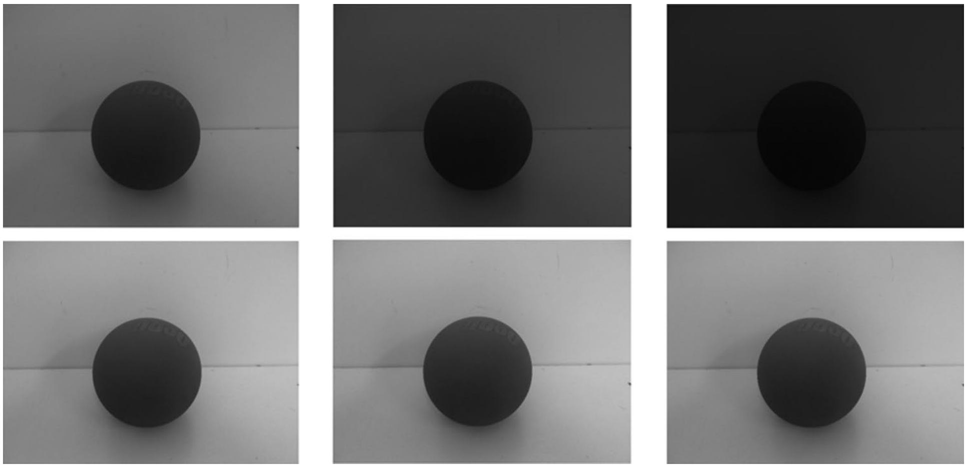

Fig. 7 In the first row, we show images degraded by attenuation in a simulated deep ocean water for three different distances (1 m [SSIM = 0.9471], 3 m [SSIM = 0.7150], 5 m [SSIM = 0.4951]). In the second row, the corresponding restored images are shown (1 m [SSIM = 0.0.9992], 3 m [SSIM = 0.9993], 5 m [SSIM = 0.9947])

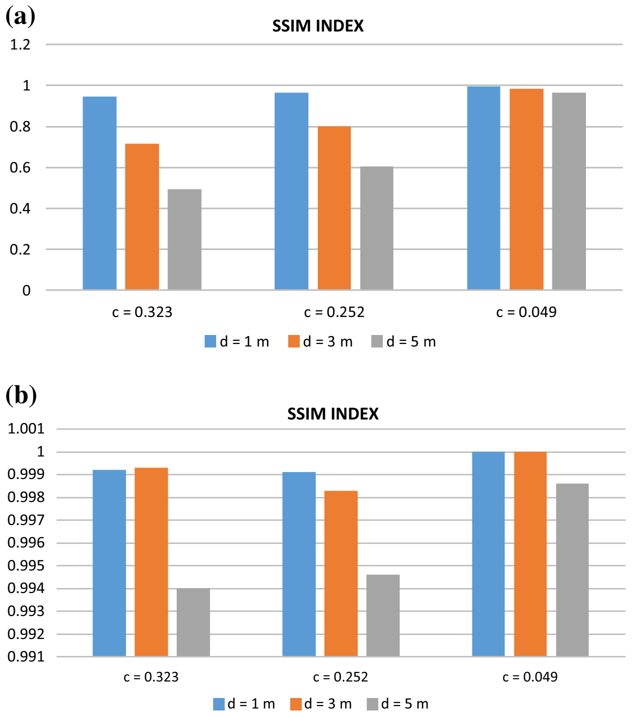

Fig. 8 SSIM for **a** degraded and **b** restored images

Table 4 Performance results of the proposed system

Power consumption	132.48 mW
Maximum frequency	10.2 MHz
Throughput	26.56 frames per second (800 × 480)

other methods. It is important to point out that, in our approach, the throughput is determined from the final restored image, considering the whole processing chain (from image acquisition, passing through image subtraction, center of mass and distance calculation and, finally, image restoration). Since, up to our knowledge, there is no work in the literature in which all these stages are performed, a direct comparison with others works is difficult. Also, it is important to take into account that the FPGA families (and devices) used by other authors are not the same as the ones used in our work. In summary, results in Table 5 must be analyzed carefully. Taking into account these difficulties, it can be observed from Table 5 that the works by Murphy et al. [20] and Ambroch et al. [17] have achieved slightly better performances than our method; however, Spartan-3 and Stratix II devices are slightly faster than Cyclone II. Also, the work by Gardel et al. [18] achieves the best throughput performance among all methods, but it uses a higher performance FPGA (Spartan 6).

7 Conclusions and future works

This work presented the development of a real-time image processing system, based on stereo vision, which measures distance and restores images. With a hardware-software co-design approach, the stereo vision system was able to (1) detect an object using a background subtraction algorithm, (2) calculate its center of mass and (3) restore the images degraded by an underwater environment, which is a fundamental task for underwater robots. Given that there are no previous works that execute the same tasks, a direct comparison with others works is difficult to be done. Nevertheless, the achieved throughput performance of 10.20 MPixels/second is expressive, if we take into consideration our image size (800 × 480) and our low cost FPGA device (Cyclone II).

In the proposed approach, the NIOS II soft-processor was used to execute the non-critical tasks, such as the disparity and distance calculations. Synthesis results show an usage of less than 70 % of the FPGA logical elements and of 90 % of the memory bits. A throughput of 26 frames per second has been achieved, showing that the system is able to detect an object and estimate its distance at a rate of 26 times per second. Also, results show a good precision performance, with an accuracy error lower than 10 %. As future work, other disparity check algorithms will be developed to improve the system capabilities. Other image restoration techniques will

Table 3 Synthesis results for all architectures shown in Fig. 6

Block name	LEs	RAM bits	Embedded multipliers	Max.Freq (MHz)
CycloneII 2C35	33,216	483,840	35	250
CMOS control	2161	57,400	0	45.31
LCD control	2275	49,592	0	123
Background subtraction	753	0	0	250
Center of mass	2521	0	0	10.2
Restoration	1188	260	13	108.15
Nios II processor	3615	285,696	4	250
Complete system	18,513	401,796	17	10.2

Table 5 Comparison with related works

Work	Device	Frame size	Throughput (MPixels/second)
Peri et al. [6]	Virtex 4	512 × 512	6.71
Murphy et al. [20]	Spartan 3	320 × 240	11.52
Hadjitheophanous et al. [21]	Virtex 2	320 × 240	2.73
Ambroch et al. [17]	Stratix II	330 × 375	16.83
Gardel et al. [18]	Spartan 6	2000 × 1000	100.00
Banz at al. [22]	Virtex V	640 × 480	9.22
Our approach	Cyclone II	800 × 480	10.20

be implemented to enhance system usability in underwater environments. The developed system will be adapted to underwater robots for monitoring and exploring tasks.

References

1. Ishibashi S (2009) The stereo vision system for an underwater vehicle. In: OCEANS 2009—EUROPE, pp 1–6. doi:10.1109/OCEANSE.2009.5278314
2. Satish Kumar N, Kumar R (2011) Design & development of autonomous system to build 3D model for underwater objects using stereo vision technique. In: Annual IEEE India Conference (INDICON), pp. 1–4. doi:10.1109/INDCON.2011.6139621
3. Zheng B, Zheng H, Zhao L, Gu Y, Sun L, Sun Y (2012) Underwater 3D target positioning by Inhomogeneous Illumination based on binocular stereo vision. In: OCEANS, Yeosu 2012, pp 1–4. doi:10.1109/OCEANS-Yeosu.6263373
4. Wu Y, Nian R, He B (2013) 3D reconstruction model of underwater environment in stereo vision system. In: Oceans—San Diego 2013, pp 1–4
5. Trucco E, Verri A (1998) Introductory techniques for 3-D computer vision. Prentice Hall PTR, Upper Saddle River
6. Perri S, Colonna D, Zicari P, Corsonello P (2006) Sad-based stereo matching circuit for FPGAs. In: 13th IEEE international conference on electronics, circuits and systems ICECS '06. pp 846–849. doi:10.1109/ICECS.2006.379921
7. Raimondo Schettini SC (2010) Underwater image processing: state of the art of restoration and image enhancement methods. EURASIP J Adv Signal Process, p 14. doi:10.1155/2010/746052
8. Trucco E (2006) Self-tuning underwater image restoration. IEEE J Ocean Eng, pp 511–519. doi:10.1109/JOE.2004.836395
9. McGlamery BL (1979) A computer model for underwater camera systems. SPIE Ocean Opt 208:221–231. doi:10.1117/12.958279
10. Jaffe JS (1990) Computer modeling and the design of optimal underwater imaging systems. IEEE J Ocean Eng, pp 101–111. doi:10.1109/48.50695
11. Lazaros N, Sirakoulis GC, Gasteratos A (2008) Review of stereo vision algorithms: from software to hardware. Int J Optomechatron 2(4):435–462. doi:10.1080/15599610802438680
12. Seunghun J, Cho J, Xuan DP et al (2010) FPGA design and implementation of a real-time stereo vision system. IEEE Trans Circ Syst Video Technol 20:12. doi:10.1109/TCSVT.2009.2026831
13. Kalomiros J, Lygouras J (2010) Robotic mapping and localization with real-time dense stereo on reconfigurable hardware. Int J Reconfigurable Comput, vol 2010. doi:10.1155/2010/480208
14. Kalomiros J, Lygouras J (2009) Comparative study of local sad and dynamic programming for stereo processing using dedicated hardware. EURASIP J Adv Signal Process. doi:10.1155/2009/914186
15. Georgoulas I, Andreadis I (2008) Real-time stereo vision techniques. In: Proceedings of the 16th IFIP/IEEE international conference on very large scale integration (VLSI-SoC 2008)
16. Kalomiros J, Lygouras J (2008) Hardware implementation of a stereo co-processor in a medium-scale field programmable gate array. Comput Dig Tech IET 2(5):336–346. doi:10.1049/iet-cdt:20070147
17. Ambrosch K, Kubinger W, Humenberger M, Steininger A (2008) Flexible hardware-based stereo matching. EURASIP J Embed Syst 2(1–2):12. doi:10.1155/2008/386059
18. Gardel A, Montejo P, Garca J, Bravo I, Lzaro JL (2012) Parametric dense stereovision implementation on a system-on chip (soc). Sensors 12(2):1863–1884. doi:10.3390/s120201863
19. Sánchez-Ferreira C, Mori J, Llanos C (2012) Background subtraction algorithm for moving object detection in FPGA. In: VIII southern conference on programmable logic (SPL), pp 1–6. doi:10.1109/SPL.2012.6211792
20. Murphy C, Lindquist D, Rynning A, Cecil T, Leavitt S, Chang M (2007) Low-cost stereo vision on an FPGA. In: 15th annual IEEE symposium on field-programmable custom computing machines FCCM. pp 333–334. doi:10.1109/FCCM.2007.44
21. Hadjitheophanous S, Ttofis C, Georghiades A, Theocharides T (2010) Towards hardware stereoscopic 3d reconstruction a real-time FPGA computation of the disparity map. In: Design, automation test in Europe conference exhibition (DATE), pp 1743–1748. doi:10.1109/DATE.2010.5457096
22. Banz C, Hesselbarth S, Flatt H, Blume H, Pirsch P (2010) Real-time stereo vision system using semi-global matching disparity estimation: architecture and FPGA implementation. In: International conference on embedded computer systems (SAMOS), pp 93–101. doi:10.1109/ICSAMOS.2010.5642077
23. Botella G, Rodríguez M, García A, Ros E (2008) Neuromorphic configurable architecture for robust motion estimation. Int J Reconfigurable Comput, vol 2008. doi:10.1155/2008/428265
24. Jia Y, Li M, An L, Zhang X (2003) Autonomous navigation of a miniature mobile robot using real-time trinocular stereo machine. In: Proceedings of the 2003 IEEE international conference on robotics, intelligent systems and signal processing. pp 417–421. doi:10.1109/RISSP.2003.1285610
25. Villalpando CY, Morfopolous A, Matthies L (2011) FPGA implementation of stereo disparity with high throughput for mobility applications. In: IEEE aerospace conference. pp 1–10. doi:10.1109/AERO.2011.5747269
26. Iwata H, Saneyoshi K (2012) Forward obstacle detection system by stereo vision. In: Proceedings of the IEEE international conference on robotics and biomimetics, pp 1842–1847. doi:10.1109/ROBIO.2012.6491236
27. Bonin GF, Burguera A (2011) Imaging systems for advanced underwater vehicles. J Marit Res VIII:65–86
28. Schechner NY (2004) Clear underwater vision. Proc Comput Vis Pattern Recognit. doi:10.1109/CVPR.2004.1315078
29. Yoshida H (2009) Fundamentals of underwater vehicle hardware and their applications. InTech 29:557–582. doi:10.5772/6721
30. Memik SO, Katsaggelos AK, Sarrafzadeh M (2003) Analysis and FPGA implementation of image restoration under resource constraints. IEEE Trans Comput 52:390–399. doi:10.1109/TC.2003.1183952

31. Ngo HT, Zhang MZ, Tao L, Asari VK (2006) Design of a digital architecture for real-time video, enhancement based on illuminance-reflectance model. In: 49th IEEE International midwest symposium on circuits and systems, MWSCAS '06, pp 286–290. doi:10.1109/MWSCAS.2006.382053

32. Mori J, Sánchez-Ferreira C, Llanos C (2012) Real-time image processing based on neighborhood operations Using. In: Proceedings of the XVIII International IBERCHIP Workshop, pp 97–102