

Article

Visual Detection and Tracking System for a Spherical Amphibious Robot

Shuxiang Guo ^{1,2}, Shaowu Pan ¹, Liwei Shi ^{1,*}, Ping Guo ¹, Yanlin He ¹ and Kun Tang ¹

¹ Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, the Ministry of Industry and Information Technology, School of Life Science, Beijing Institute of Technology, Beijing 100081, China; guo@eng.kagawa-u.ac.jp (S.G.); panshaowu@bit.edu.cn (S.P.); guoping1962@hotmail.com (P.G.); heyanlin@bit.edu.cn (Y.H.); tangkun@bit.edu.cn (K.T.)

² Faculty of Engineering, Kagawa University, 2217-20 Hayashicho, Takamatsu, Kagawa 761-0396, Japan

* Correspondence: shiliwei@bit.edu.cn; Tel.: +86-10-6891-5908

Academic Editor: Simon X. Yang

Received: 20 January 2017; Accepted: 5 April 2017; Published: 15 April 2017

Abstract: With the goal of supporting close-range observation tasks of a spherical amphibious robot, such as ecological observations and intelligent surveillance, a moving target detection and tracking system was designed and implemented in this study. Given the restrictions presented by the amphibious environment and the small-sized spherical amphibious robot, an industrial camera and vision algorithms using adaptive appearance models were adopted to construct the proposed system. To handle the problem of light scattering and absorption in the underwater environment, the multi-scale retinex with color restoration algorithm was used for image enhancement. Given the environmental disturbances in practical amphibious scenarios, the Gaussian mixture model was used to detect moving targets entering the field of view of the robot. A fast compressive tracker with a Kalman prediction mechanism was used to track the specified target. Considering the limited load space and the unique mechanical structure of the robot, the proposed vision system was fabricated with a low power system-on-chip using an asymmetric and heterogeneous computing architecture. Experimental results confirmed the validity and high efficiency of the proposed system. The design presented in this paper is able to meet future demands of spherical amphibious robots in biological monitoring and multi-robot cooperation.

Keywords: spherical amphibious robot; Gaussian mixture model; moving target detection; system-on-chip (SoC); visual tracking

1. Introduction

With the increased interest in ocean exploitation activities, amphibious robots have become essential tools for applications such as ecological observations and military reconnaissance in littoral regions [1,2]. Compared with legged, finned, and snakelike amphibious robots, spherical amphibious robots generate less noise and water disturbance, providing better stealth ability and bio-affinity [3,4]. Driven by pendulums [5], propellers [6] or rollers [7], most spherical amphibious robots are able to move flexibly in mud, snow or water with a zero turning radius. In 2012, our team proposed a novel spherical amphibious robot [8]. With a deformable mechanical structure, it walked on land with legs and swam in water with vectored propellers, which provided better overstepping ability and adaptability in littoral regions.

Moving target detection and tracking is a fundamental function for spherical amphibious robots to complete missions such as multi-robot formation, biological investigations, target localization, and navigation. The detection and tracking system recognizes an approaching target and then successively calculates its motion trajectory. Due to the restrictions presented by the environment,

few sensors are suitable for detection and tracking applications of small-sized amphibious robots which have limited carrying capacities and battery power. As a fundamental telemetry device for most autonomous underwater vehicles (AUVs), acoustic sensors (e.g., side-scan sonar, single-beam sonars, and multi-beam sonars) can be effective over medium-to-long distances, however, most acoustic sensors are heavy and not suitable for observations at short distances (i.e., within 10 m) [9,10]. Optical and sonic tags broadcasting specific codes can be used to mark targets of interest (e.g., fish) for precise tracking over long distances [10,11]. However, this method is limited by the target size and provides few details on the surroundings or the target. Optical sensors, based on scanning light detection and ranging (LiDAR), laser-line scanning, structure light, and photometric stereo, have been used for underwater three-dimensional (3D) reconstruction [12]. However, most optical 3D reconstruction solutions require sophisticated optical structures and data processing devices, making it difficult to integrate such systems into a sub-meter-size spherical amphibious robot. With advantages in terms of weight, flexibility, and environmental adaptability, a visual detection and tracking system has been the key sensing equipment for small-scale spherical amphibious robots in executing close-range observation or inspection tasks in amphibious scenarios.

Although great progress has been achieved in the field of ground robotic vision, it still remains a challenging task to design a robotic detection and tracking system for spherical amphibious robots. First, image degradation is a major problem in underwater environments, which greatly impacts the performance of robotic visual systems. Second, interfering factors such as partial object occlusion, light variance, and pose changes are common in the potential application scenarios of amphibious robots, which may lead to detection or tracking failures. Third, most amphibious robots are small-sized and have relatively weak image processing power. Thus, both the visual algorithms and processing circuits must be carefully designed and optimized. As far as we know, most studies on robotic vision systems were conducted in terrestrial environments. Some visual detection and tracking systems have been designed for underwater robots or underwater surveillance networks, but few studies have involved amphibious robots.

Yahyaa et al. [13] proposed a visual detection and tracking system to guide an AUV towards its docking station. A tracker using color thresholding and morphological operations was designed to track artificial objects, but the robotic vision system could only recognize and track specific red light sources. Zhang et al. [14] presented a multi-target tracking system using multiple underwater cameras. An extended Kalman filter was initialized to track the moving targets. Speed up robust features (SURF) and random sample consensus (RANSAC) algorithms were used to match the target objects across the overlapping fields of view, but the cameras used were static and the visual system had poor real-time performance, which made it unsuitable for robotic applications. Chen et al. [15] proposed a novel 3D underwater tracking method in which haze concentration was used to estimate the distance between the target and the camera. However, this only provides the motion trends of underwater objects rather than precise measurements. Chuang et al. [16] proposed a robust multiple fish tracking system using Viterbi data association and low-frame-rate underwater stereo cameras. But it could only work in dark environments and provide a frame rate as low as 5 frame per second (fps). Chuang et al. [17] proposed a novel tracking algorithm on the basis of the deformable multiple kernels to track live fish in an open aquatic environment. Inspired by the deformable part model technique, the algorithm outperformed the recent tracking-by-detection algorithms for tracking one or multiple live fish in challenging underwater videos. But it could also provide a frame rate lower than 1 fps, which limited its applications in mobile robotic platforms.

In general, most existing robotic detection and tracking systems adopted vision algorithms using static or coarse appearance models [13,18], making them only capable of effectively processing specific targets such as fish and beacons under the specific scenes. There were some detection and tracking systems which adopted state-of-the-art visual algorithms and was capable of processing generic targets [16,17,19]. Given the real-time performance of these sophisticated algorithms, these systems have to be built upon high performance computers, making them only suitable for large-scale

underwater robots or ground robots. Thus, existing solutions cannot be used directly in the small-sized amphibious spherical robot, which has limited load space and computational capabilities.

Focusing on the tasks of ecological observations and intelligent surveillance in littoral regions, a moving target detection and tracking system was proposed for our amphibious spherical robot in this study. Given the potential application scenarios of the robot, an industrial camera and vision algorithms using adaptive appearance models were used to construct the designed system. To handle the problem of light scattering and absorption in the underwater environment, the multi-scale retinex with color restoration (MSRCR) algorithm was used for image enhancement. In the detection stage, the amphibious spherical robot lurked in the surveyed region in hibernation mode and sensed the surroundings by capturing 320×240 color images at 15 fps. The Gaussian mixture model (GMM) was used to sense moving targets entering the robot's view field. Once a moving target had been detected, the robot was woken up to the tracking stage. A fast compressive tracker (FCT) with a Kalman prediction mechanism was launched to locate the target position successively. Considering the limited load space and power resources of the robot, the designed visual detection and tracking system was implemented with a low-power system-on-chip (SoC). A novel asymmetric and heterogeneous computing architecture was used to ensure the real-time performance of the system. Experimental results confirmed that the proposed system was capable of detecting and tracking moving targets in various amphibious scenarios. In comparison with most relevant detection and tracking systems, the proposed system outperformed in terms of processing accuracy, environmental adaptability, real-time performance, and power consumption. It was able to meet future demands of the amphibious spherical robot in biological observation and multi-robot cooperation. The study in this paper provided a reference design to vision systems of small-sized amphibious robots.

The rest of this paper is organized as follows. An overview on our amphibious spherical robot and its vision application requirements is introduced in Section 2. The structure of the proposed vision system is presented in Section 3. Details of the underwater image enhancement subsystem and the detection-then-tracking subsystem are described in Sections 4 and 5. Experimental results under various scenarios are reported in Section 6. Section 7 provides our conclusions and relevant follow-up research work.

2. Previous Work and Application Requirements

2.1. An Amphibious Spherical Robot

Figure 1 shows the amphibious spherical robot, which consisted of an enclosed hemisphere hull (diameter: 250 mm) and two openable quarter-sphere shells (diameter: 266 mm). Electronic devices and batteries were installed inside the hemispherical hull, which was waterproof and provided protection from collisions. Four legs, each of which was equipped with two servo motors and a water-jet motor, were installed symmetrically in the lower hemisphere of the robot. Driven by the two servo motors, the upper joint and lower joint of a leg were able to rotate around a vertical axis and a horizontal axis, respectively. The water-jet motor was fixed in the lower joint and could generate a vectored thrust in a specific direction in water. In underwater mode, the openable shells closed to form a ball shape, and the robot was propelled by vectored thrusts from four water jets. In land mode, the openable shells opened and the robot walked using the four legs.

Restricted by the narrow load space and limited power resources of the small-scale robot, the robotic electronic system was fabricated around an embedded computer (Xilinx XC7Z045 SoC, 512 MB DDR3 memory, Linux 3.12.0), as shown in Figure 2. The robot was powered by a set of Li-Po batteries, with a total capacity of 24,000 mAh. Sensors including a global positioning system (GPS) module, an inertial measurement unit (IMU), and an industrial camera were used to achieve adaptive motion control of the robot [20]. An acoustic modem and a universal serial bus (USB) radio module were used for communication on land and in water, respectively [21].

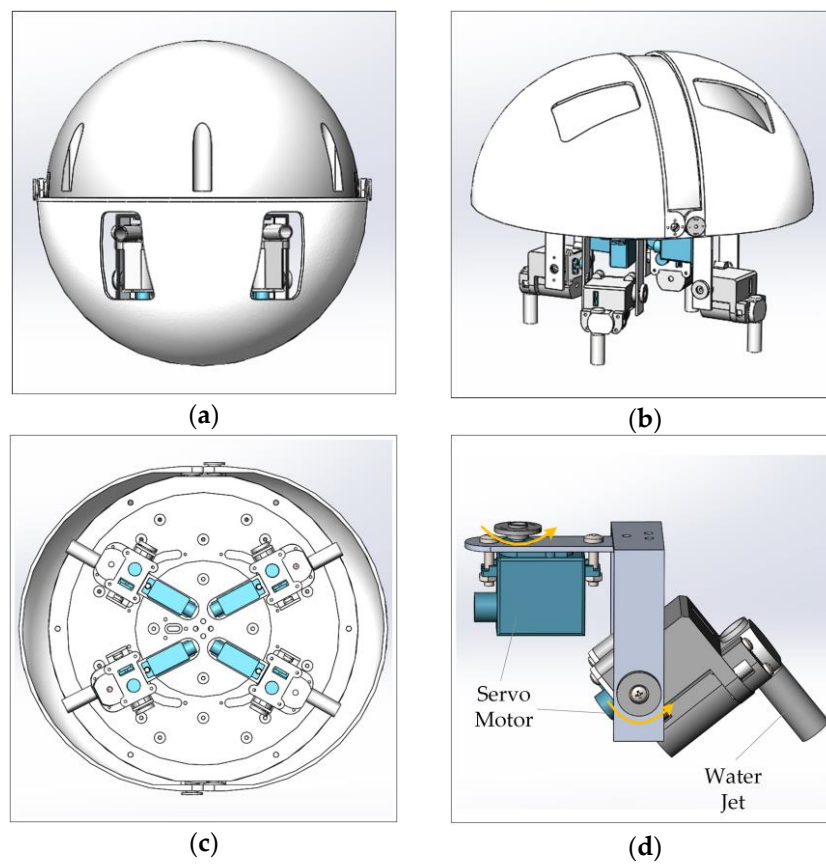


Figure 1. Mechanical structure of the amphibious spherical robot. (a) The amphibious spherical robot in the underwater mode; (b) The amphibious spherical robot in the land mode; (c) Bottom view of the amphibious spherical robot; and (d) The mechanical structure of a leg.

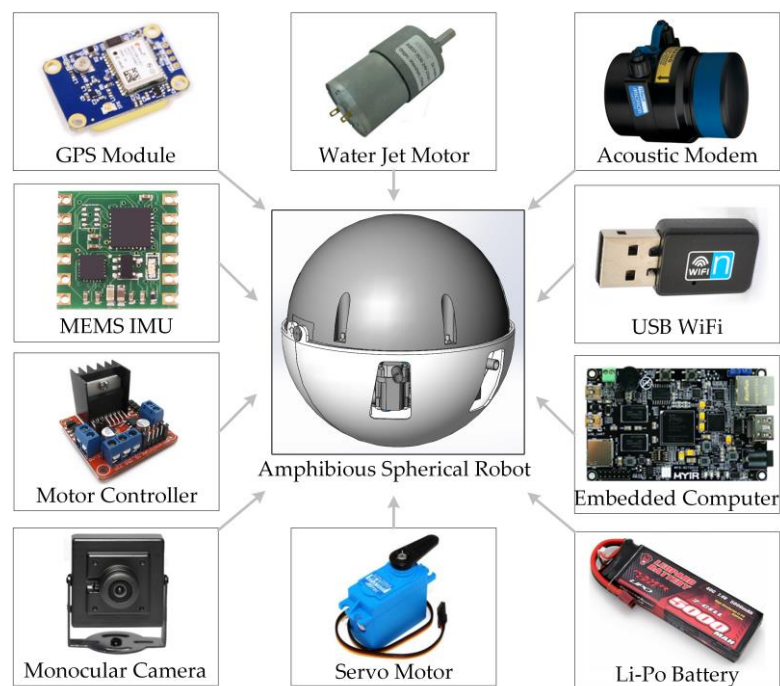


Figure 2. Major functional components of the amphibious spherical robot.

2.2. Vision Application Requirements

Due to the special working environment, a vision method is the preferred solution to realize intelligent functions of the spherical robot in amphibious scenarios. Compared with intelligent automobiles and large-scale AUVs, the amphibious spherical robot has higher requirements for the robotic vision system in terms of robustness and efficiency.

First, due to the uneven illumination and the optical properties of water, the captured image may be blurred and dusky. Thus, image pre-processing is essential to enhance visibility before implementing vision algorithms. Second, many interfering factors in amphibious environments, including swaying aquatic plants, suspended organic particles, illumination changes, and a cluttered background, may mislead the detector and the tracker. Thus, the robustness and precision of the adopted computer vision algorithms should be acceptable to meet the requirements of robotic applications. Third, the robot has a limited velocity and cruising ability. Thus, the captured image should be processed in real time to avoid missing a target or omitting information. Moreover, considering the narrow enclosed load space of the spherical robot, the hardware platform of the robotic vision system should be highly efficient to reduce power consumption and heat dissipation issues. Furthermore, the implementation of the adopted vision algorithms should be optimized carefully in accordance with characteristics of the hardware platform.

In 2015, a prototype moving target detection system was designed and constructed for the amphibious spherical robot [22]. A single Gaussian background model (GBM) was adopted to sense moving objects getting close to the robot and the heterogeneous computing technology was used to enhance the real-time performance. However, the prototype system did not perform well in practicality experiments because the illumination problem and the interfering factors in amphibious environments were not taken into consideration yet. Moreover, due to the principle of the adopted algorithm, the detection system could work normally only when the robot was static, which was not the case. Besides, the coarse system architecture led to a slower respond speed of the control system, which affected the performance of the robot.

3. Visual Detection and Tracking System

3.1. Workflow of the System

Benefiting from its ball shape, the amphibious spherical robot generated fewer disturbances to the surroundings, which is meaningful in ecological and military applications [23]. Moreover, the symmetric mechanical structure contributed to the stable and flexible motion characteristics of the robot, making it a platform suitable for amphibious data acquisition. However, the compact size and the spherical shape of the robot also resulted in a limited cruising speed and a short continuous operating time. Thus, it was unable to complete search or investigation tasks over a large region. Indeed, it was more appropriate to use the robot as an intelligent and movable monitoring node for close-range ecological observations or security surveillance.

A potential application scenario is shown in Figure 3. The working process was divided into the moving target detection stage and the visual tracking stage. In the moving target detection stage, the small-scale robot lurked in the survey region in hibernation mode. Most of its functional units, including motors, the data recording subsystem, and the acoustic modem, were shut down to avoid exhausting the batteries and storage resources too early. Enhanced images of surroundings were entered into the visual detection subsystem to search for moving targets, such as fish and swimmers. Once a moving object entered the view field of the robot and was marked as a target, the robot would be activated and switched to visual tracking mode. In the visual tracking stage, a visual tracker was launched to track the specified target. The tracking results were then used to guide the movement and follow-up operations of the robot.

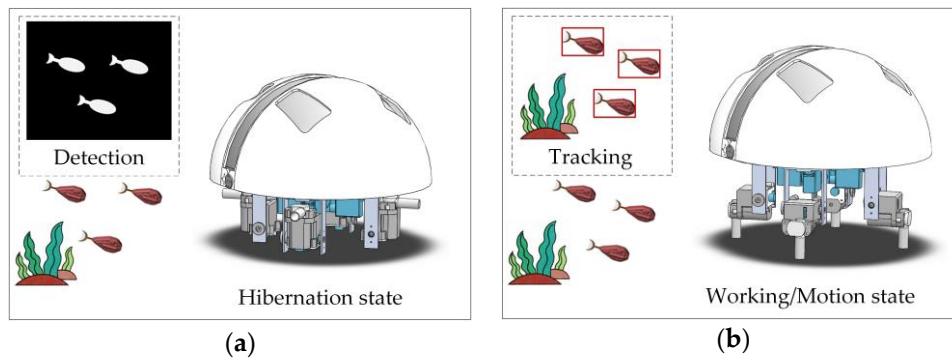


Figure 3. Application scenario of the visual detection and tracking system. (a) The moving target detection stage; and (b) The visual tracking stage.

3.2. Structure of the System

The entire visual detection and tracking systems were integrated on a Xilinx XC7Z045 SoC, as shown in Figure 4. As the center of the robotic electronic system, the SoC consists of the processing system (PS), which is centered on a dual-core ARM processor, and the programmable logic (PL), which is equivalent to a field-programmable gate array (FPGA) [24]. The PL served as a customized peripheral of the PS and communicated with programs running on the PS through advanced extendable interface (AXI) ports.

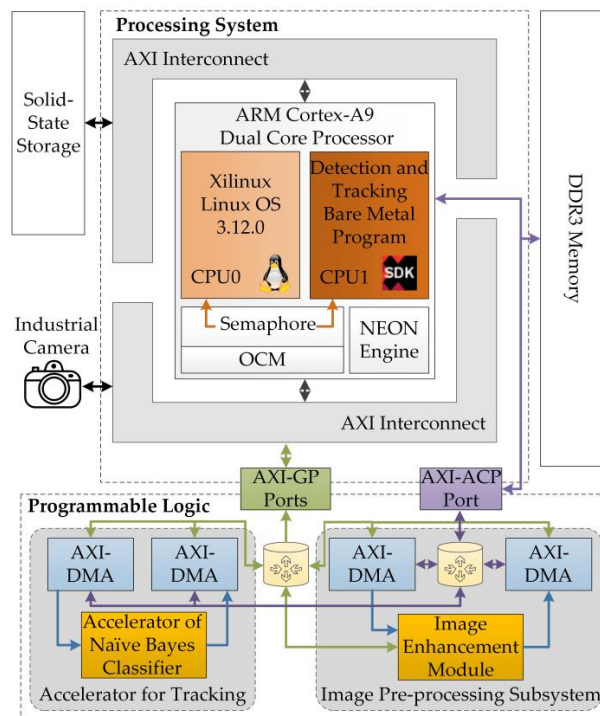


Figure 4. Hardware structure of the visual detection and tracking system.

To ensure balance between the power consumption of the electronic system and the real-time performance of the robotic vision system, an asymmetric and heterogeneous computing architecture was used to develop the visual detection and tracking system. The CPU0 ran the Linux operating system (OS), which provided a multi-task platform for basic robotic functions, such as motion control and battery management. The CPU1 ran bare-metal programs for real-time detection and tracking. Customized accelerators deployed on the PL assisted the bare-metal programs to ensure real-time

performance. The application programs running in the Linux OS communicated with the bare-metal programs through a shared on-chip memory. The 320×240 color images to be processed were captured by an industrial camera mounted on a USB port of the SoC. To address the problem of image degradation, an image-enhancement module using the MSRCR algorithm was implemented on the PL for real-time image pre-processing. A customized accelerator for the naïve Bayes classifier was deployed on the PL to speed up the bare-metal visual tracking program. Two pairs of direct memory access (DMA) channels were used to read unprocessed data from the PS and then transmit the processed results back to the bare-metal programs.

4. Image Pre-Processing Subsystem

4.1. Principle of the Image Pre-Processing Algorithm

Due to the short sensing range and the image degradation problem, cameras have not been at the center of attention as underwater robotic sensors. The degradation of underwater images is caused primarily by multiple factors including light attenuation, light scattering, and suspended organic particles [18]. Existing underwater enhancement algorithms can be divided loosely into four classes. The time domain algorithms and the frequency domain algorithms enhance image quality using ‘classical’ digital image processing techniques such as histogram equalization and homomorphic filtering. The physics-based algorithms build an optical model of underwater imaging devices and recover the image visibility using optical components [25,26]. The algorithms based on the theory of color constancy were inspired by the human vision system and seek to depress image degradation caused by illumination factors [27,28]. Among these algorithms, the MSRCR algorithm provides a good processing effect by taking advantage of multi-scale analysis and color recovery.

The MSRCR algorithm was inspired by the model of lightness and color perception of human vision. The retinex theory holds that the image projected onto the retina $I(x, y)$ is determined by the illumination component $L(x, y)$ and the relative reflectance component $R(x, y)$:

$$I(x, y) = L(x, y) \cdot R(x, y), \quad (1)$$

where x and y represent the coordinate of an image. Thus, the negative influence of light scattering and absorption can be excluded by estimating $L(x, y)$. As shown in Figure 5c, an estimate of $L(x, y)$ can be acquired using a Gaussian low-pass filter:

$$\hat{L}(x, y) = I(x, y) * k \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right), \quad (2)$$

where σ represents the scale of the Gaussian filter. Then, the relative reflectance component $R(x, y)$ can be represented as

$$R(x, y) = \log(I(x, y)) - \log(F(x, y) * I(x, y)), \quad (3)$$

where $F(x, y)$ represents the Gaussian filter.



Figure 5. Cont.

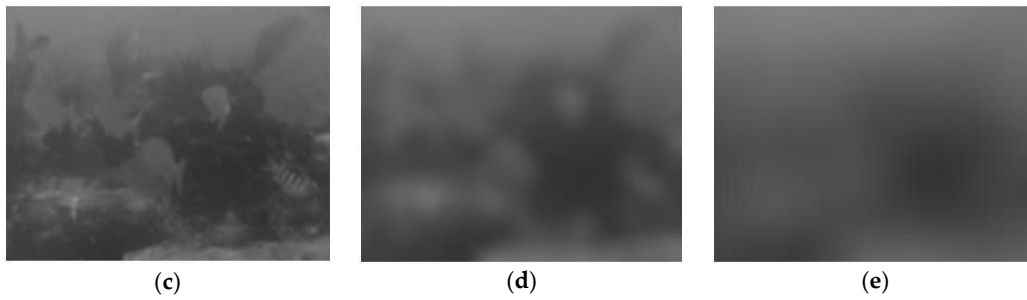


Figure 5. Diagram of the image pre-processing algorithm. (a) The original 320×240 image; (b) The enhanced image; (c) Estimate of $L(x, y)$ using the 5×5 Gaussian filter; (d) Estimate of $L(x, y)$ using the 24×24 Gaussian filter; and (e) Estimate of $L(x, y)$ using the 48×48 Gaussian filter.

The value of σ is important for the retinex algorithm, especially for an image with non-uniform background illumination. A small σ works better on dark regions of the image, and a large σ leads to better color constancy, as shown in Figure 6b–e. To make use of the strength of multi-scale synthesis, the MSRCR algorithm combining multiple scales is commonly used with linear weighting:

$$R(x, y) = \log(I(x, y)) - \sum_{i=1}^{n_S} w_i \log(F_i(x, y) * I(x, y)), \quad (4)$$

where n_S represents the number of adopted filter scales. Both the details and the color constancy of the processed image can be ensured using a $n_S > 3$, as shown in Figure 6f–h. A larger n_S may lead to better algorithm performance and higher computational needs. Given the characteristics of the robot and the size of images to be processed, the proposed system adopted three scales ($\sigma_1 = 5$, $\sigma_2 = 24$, and $\sigma_3 = 48$), which balanced the image contrast, color constancy, and computational efficiency.

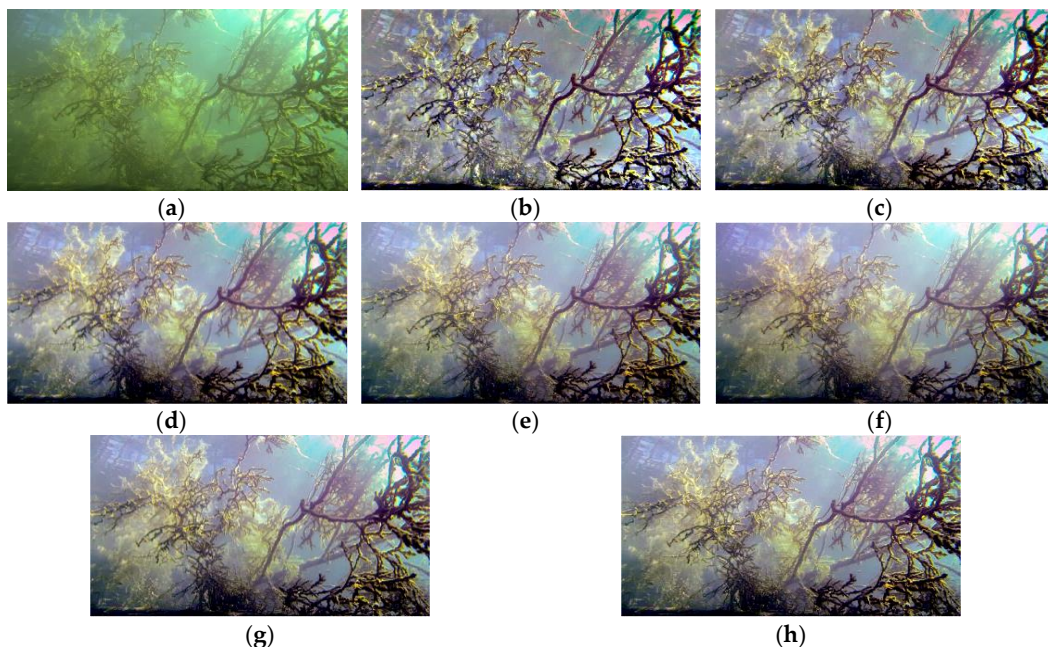


Figure 6. Comparison of the multi-scale retinex with color restoration (MSRCR) algorithm with different parameters. (a) The original 852×480 image; (b) Enhanced image ($n_S = 1$, $\sigma_{\max} = 50$); (c) Enhanced image ($n_S = 1$, $\sigma_{\max} = 100$); (d) Enhanced image ($n_S = 1$, $\sigma_{\max} = 200$); (e) Enhanced image ($n_S = 1$, $\sigma_{\max} = 300$); (f) Enhanced image ($n_S = 2$, $\sigma_{\max} = 300$); (g) Enhanced image ($n_S = 3$, $\sigma_{\max} = 300$); and (h) Enhanced image ($n_S = 4$, $\sigma_{\max} = 300$).

4.2. Image Pre-Processing Subsystem

The MSRCR algorithm involves large amounts of multiplication operations, which are time-consuming. To ensure real-time performance of the proposed robotic vision system, a customized IP core was designed to implement the MSRCR algorithm using high-level synthesis (HLS) tools. A 320×240 24-bit color image was read serially from the DMA channel into the IP core through an AXI-Stream port, as shown in Figure 7. The color image was converted to an 8-bit gray image and then buffered into a slice of block RAM (BRAM). Next, three convolution operations were executed in parallel. Then, logarithmic transformations were carried out serially over the calculated $\hat{L}(x, y)$. Finally, the enhanced image $R(x, y)$ was sent out through an AXI-Stream port after a linear color correction operation.

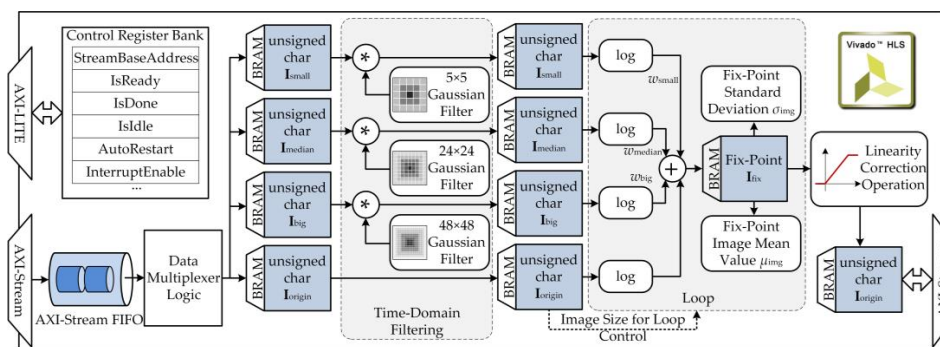


Figure 7. Diagram of the image pre-processing subsystem.

The low-pass filtering and standard deviation calculation functions were designed with C++, referring to their counterparts in the OpenCV library. In the convolution operation, the input image was extended at the boundary with the edge pixel not duplicated. Because the quality of synthesis results provided by HLS tools are less than ideal, it was essential to conduct design optimization. To reduce resource consumption of the PL, the filter parameters were represented in the accelerator by fixed-point approximations. The synthesis report showed that the operation time of the designed IP core was ~ 48.0 ms, which was 3.7 times faster than the software implementation on the PS.

5. Detection and Tracking Subsystem

5.1. Moving Target Detection Subsystem

As mentioned in Section 3, the robot sensed moving objects entering its observation field and then specified an eligible one as a target to be tracked. A common method for moving target detection is using background subtraction or motion detection algorithms that have been used successfully in intelligent surveillance systems. State-of-the-art background subtraction algorithms, such as most reliable background mode (MRBM) and effect components description (ECD) demand large amounts of memory and/or computing time [29], making them unsuitable for use in the amphibious spherical robot. Moreover, ‘classical’ algorithms, such as the frame difference algorithm and the weighted moving mean algorithm, may easily be misled by interfering factors, including swaying aquatic plants and suspended organic particles in practical applications [30].

Thus, the adaptive Gaussian mixture model for foreground detection proposed by Kaewtrakulpong et al. [31,32], which has good detection precision and is able to neglect noises caused by background jitter, was adopted in the proposed robotic vision system. An overview on the principles of the adopted moving target detection algorithm is shown in Algorithm 1. Each pixel of the input image was modeled with a mixture of K Gaussian distributions:

$$R(x, y) \sim w_{x,y,k} N(\mu_{x,y,k}, \sigma_{x,y,k}), \quad (5)$$

where $\mu_{x,y,k}$, $\sigma_{x,y,k}$, and $\omega_{x,y,k}$ are parameters of the k th Gaussian component. The K Gaussian distributions are ordered based on the fitness value $\omega_{x,y,k}/\sigma_{x,y,k}$. The top B distributions constituted the background model where B was defined as:

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_{x,y,k} > T \right). \quad (6)$$

If an input pixel was less than d standard deviations away from any of the distribution of the background model, it was regarded as belonging to the background scene. Otherwise, it was regarded as part of the potential moving target. Algorithm parameters were updated with the learning rate α to adapt to environmental changes. The detected foreground image was processed with erode and dilate operations to filter noise. A moving object larger than $Area_{Thresh}$ would be specified as the target to be tracked in the following processes.

Algorithm 1. Gaussian mixture model-based moving target detection

input: the enhanced image $R_{x,y}$ and parameters of Gaussian mixture model $\mu_{x,y,k}$, $\sigma_{x,y,k}$ and $\omega_{x,y,k}$, where $x \in [1, Width]$, $y \in [1, Height]$, $k \in [1, K]$

output: the foreground image $F_{x,y}$, where $x \in [1, Width]$, $y \in [1, Height]$

procedure GaussianMixtureModelDetection(\mathbf{R} , $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, \mathbf{w})

Step #1 Initialize the parameters of Gaussian mixture model

$\mu_{x,y,k} \leftarrow \text{rand}()$, $\sigma_{x,y,k} \leftarrow \sigma_0$, $\omega_{x,y,k} \leftarrow 1/K$

Step #2 Try to match the Gaussian mixture model with the n -th image

for $k = 1$ to K **do**

if $R_{x,y,n} - \mu_{x,y,k} < d \cdot \sigma_{x,y,k}$ **then**

$match_k = 1$

$\omega_{x,y,k} = (1 - \alpha) \cdot \omega_{x,y,k} + \alpha$

$\mu_{x,y,k} = (1 - \alpha / \omega_{x,y,k}) \cdot \mu_{x,y,k} + \alpha / \omega_{x,y,k} \cdot R_{x,y,n}$

$\sigma_{x,y,k} = \sqrt{(1 - \alpha / \omega_{x,y,k}) \cdot \sigma_{x,y,k}^2 + \alpha / \omega_{x,y,k} \cdot (R_{x,y,n} - \mu_{x,y,k})^2}$

else

$\omega_{x,y,k} = (1 - \alpha) \cdot \omega_{x,y,k}$

end if

end

Step #3 Normalize the weight $\omega_{x,y,k}$ and sort the model with $\omega_{x,y,k}/\sigma_{x,y,k}$

Step #4 Reinitialize the model with minimum weight if there is no matched model,

if $\sum_{k=1}^K match_k = 0$ **then**

$\mu_{x,y,0} = pixel_{x,y,n}$

$\sigma_{x,y,k} = \sigma_0$

end if

Step #5

for $k = 1$ to K **do**

if $\omega_{x,y,k} > T$ **and** $R_{x,y,n} - \mu_{x,y,k} < d \cdot \sigma_{x,y,k}$ **then**

$F_{x,y} = 0$

break

else

$F_{x,y} = 255$

end if

end

Step #6 Execute 3×3 erode and dilate operations over $R(x,y)$

Step #7 Execute connected region analysis and list potential moving target

Step #8 Specify the object larger than $Area_{Thresh}$ as the target

end procedure

The designed moving target detection subsystem was implemented as a bare-metal program running on CPU1. The number of Gaussian distributions was set to 4. The NEON engine was used to

optimize the floating-point arithmetic of the bare-metal program, which increased the detection rate from 7.4 fps (135.16 ms/f) to 19.7 fps (50.8 ms/f). If a moving object was detected, the detector would inform CPU0 by writing the coordinate of the target to a specific memory location. Then, the tracking subsystem would be launched to handle the target.

5.2. Visual Tracking Subsystem

The major task of the visual tracking subsystem was successively marking the position of the specified target for robotic applications. As an active research field in the area of computer vision, visual tracking is the basis of high-level robotic functions, such as automatic navigation, visual servoing, and human-machine interactions [33,34]. Many state-of-the-art tracking algorithms, built on tracking-by-detection [35], correlation filtering [36], and convolutional neural networks [37], have been proposed in recent years. However, it is still challenging to ensure both high tracking precision and real-time performance, limiting their use in small-scale mobile robot platforms.

The fast compressive tracking (FCT) algorithm was selected in the visual tracking subsystem as it offers the advantages of effectiveness and efficiency [38]. As a tracking-by-detection algorithm with online learning mechanisms, the FCT algorithm contains a training stage and a detection stage. In the training stage at the n th frame, the tracker densely crops positive samples S_{pos} and negative samples S_{neg} around the current target position I_n , as shown in Figure 8a:

$$S_{\text{pos}} = \{s \mid \|s - I_n\| < \alpha\}, \quad (7)$$

$$S_{\text{neg}} = \{s \mid \zeta < \|s - I_n\| < \beta\}, \quad (8)$$

where $\alpha < \zeta < \beta$. Then, random Haar-like features of samples were extracted using a static sparse matrix. Affected by the optical properties of water, it is not easy to extract local invariant image features in underwater vision applications. Thus, a global feature like the random Haar-like features was more effective in the designed system. After that, a naïve Bayes classifier was trained using feature vectors of the samples. In the detection stage at the $(n + 1)$ th frame, candidate samples S_{can} were densely cropped around I_n using a coarse-to-fine mechanism:

$$S_{\text{can}} = \{s \mid \|s - I_n\| < \gamma\} \quad (9)$$

The candidate with the maximum classifier response was selected as the current target I_{n+1} . Regarding the vision application of the amphibious spherical robot, there are two potential problems that may affect the performance of the FCT algorithm. One is that the FCT algorithm is not good at maneuvering target tracking due to its sampling mechanism. Moreover, the tracker may lose the target in this robotic vision system with its relatively low frame rate. To address this, a second-order Kalman filter was used to predict the target position at the $(n + 1)$ th frame in the detection stage.

$$\left\{ \begin{array}{l} X_{n+1} = \Phi X_n + \beta W_n \\ \hat{I}_{n+1} = H X_{n+1} + \alpha V_n \\ \Phi = \begin{pmatrix} 1 & 0 & \Delta t & 0 & \Delta t^2/2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \Delta t^2/2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\ H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \\ X_n = (x_n, y_n, v_{x,n}, v_{y,n}, a_{x,n}, a_{y,n})^T \\ \hat{I}_n = (\hat{x}_n, \hat{y}_n)^T \end{array} \right. , \quad (10)$$

where (x_n, y_n) , $(v_{x,n}, v_{y,n})$, and $(a_{x,n}, a_{y,n})$ are the position, the velocity, and the acceleration of the target in the n th frame, respectively. The candidate samples S_{can} would be sampled around the estimate position $\hat{\mathbf{I}}_{n+1}$ rather than \mathbf{I}_n , as shown in Figure 8b:

$$S_{\text{can}} = \{\mathbf{s} \mid \|\mathbf{s} - \hat{\mathbf{I}}_{n+1}\| < \gamma\} \quad (11)$$

Because most moving objects in water have a stable trajectory, the improved tracker was able to adapt to the motion by predefining appropriate parameters.

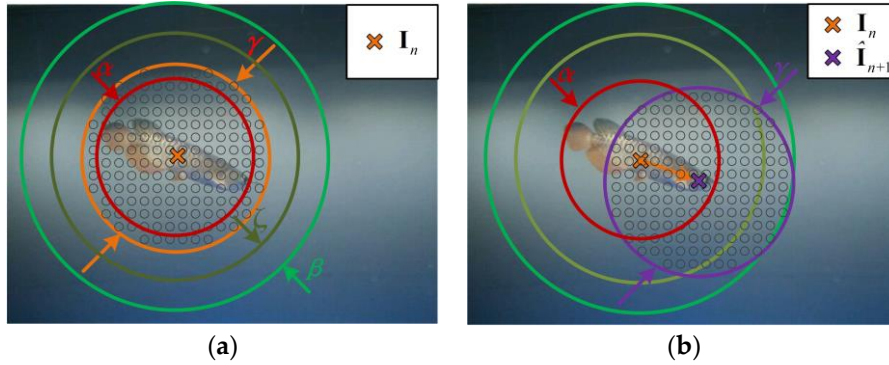


Figure 8. Principle of the visual tracking algorithm. (a) Original fast compressive tracker (FCT) algorithm; and (b) Improved FCT algorithm.

Another problem was that the floating-point arithmetic processes of the FCT algorithm, especially the naïve Bayes classification process, could not be processed efficiently by CPU1:

$$H_{\text{pos},i}(\mathbf{v}) = \frac{\exp(-(v_i - \mu_{\text{pos},i})^2 / (2\sigma_{\text{pos},i}^2 + 10^{-30}))}{\mu_{\text{pos},i} + 10^{-30}}, \quad (12)$$

$$H_{\text{neg},i}(\mathbf{v}) = \frac{\exp(-(v_i - \mu_{\text{neg},i})^2 / (2\sigma_{\text{neg},i}^2 + 10^{-30}))}{\sigma_{\text{neg},i} + 10^{-30}}, \quad (13)$$

$$H(\mathbf{v}) = \sum_{i=1}^m (\log(H_{\text{pos},i}(\mathbf{v}) + 10^{-30}) - \log(H_{\text{neg},i}(\mathbf{v}) + 10^{-30})), \quad (14)$$

where $\mathbf{v} \in \mathbb{R}^m$ represents the feature vector of a candidate sample and μ_{pos} , σ_{pos} , and σ_{neg} represent classifier parameters. As described by Equations (12)–(14), the naïve Bayes classification process primarily concerns the exponent and logarithm, which are equivalent to iterative multiplication operations. Thus, a customized accelerator may perform better than a general-purpose accelerator in speeding up these calculations.

A customized accelerator of the naïve Bayes classifier was designed and implemented on the PL section using HLS tools, as shown in Figure 9. The sampled feature vectors and classifier parameters were read from the bare-metal program running on the PS through a DMA channel and then buffered into BRAM slices. Then, a three-stage pipeline was designed to complete the classifier response calculation loop in parallel. Finally, the maximum response and the number of the candidate sample were found and sent back to the PS through a DMA channel. Using the customized accelerator, the average processing rate of the heterogeneous tracking subsystem was 56.3 fps (17.8 ms/f), which was 4.1 times faster than the software implementation on the PS (13.6 fps or 73.5 ms/f).

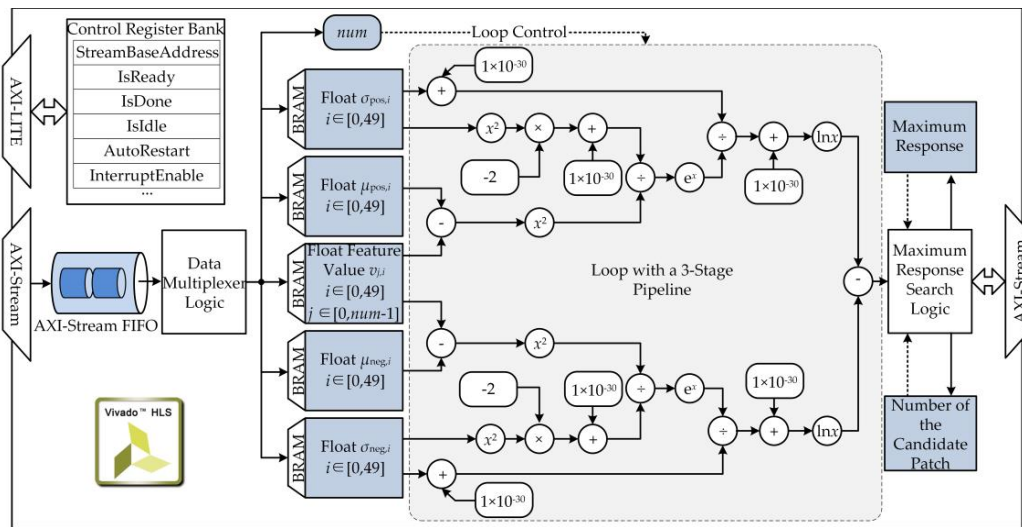


Figure 9. Diagram of the accelerator of naïve Bayes classifier for the tracking subsystem.

6. Experimental Results

The improved version of the amphibious spherical robot is shown in Figure 10. A core board carrying the Xilinx SoC (XC7Z045) and an industrial CMOS camera were used to assemble the proposed detection and tracking system. Then, the embedded robotic vision system was sealed in the upper hemisphere of the spherical robot using a transparent shell. To confirm the validation of the proposed robotic vision system, two phases of experiments were conducted to test its detection and tracking precision, real-time performance, and power consumption.

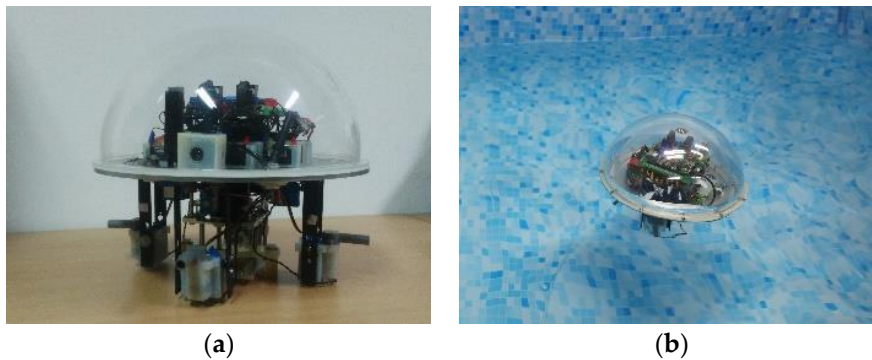


Figure 10. Picture of the proposed robotic vision system. (a) Installation of the vision system; and (b) Picture of the robot in working state.

(1) In the parametric test phase, an Agilent 34410A multimeter, controlled by C# programs, was used to evaluate the average power consumption of the proposed system by continuously measuring the current and voltage values. The power consumption of the robot in idle mode was regarded as the baseline. Test results showed that the dynamic power consumption of the proposed system was as low as 4.57 W, which was able to provide a continuous working time more than 2.5 h. To test the process rate of the proposed system, eight 320×240 image sequences (*CarDark*, *Trellis*, *David*, *Couple*, *Fish*, *Dog1*, *Sylvester*, and *ClifBar*) of Visual Tracker Benchmark [39,40] were entered into the system using debugging tools. The run time of each subsystem were measured using a hardware counter deployed on the PL section, respectively. Test results indicated that the system was able to provide an average pre-processing rate of 20.8 fps, an average detection rate of 19.7 fps, and an average tracking rate of 56.3 fps. Thus, it was able to process images captured by the industrial camera in amphibious scenarios

in real time. As shown in Table 1, the proposed system had advantages in real-time performance, power consumption, size and weight, which could fully meet the application requirements of the amphibious spherical robot.

Table 1. Comparison of detection and tracking systems for underwater or amphibious applications.

Vision System	Hardware Platform	Image Size	Maximum Frame Rate	Working Scenarios
Proposed System	SoC	320 × 240	56.3 fps	Static and dynamic background
Shiau [19] et al.	PC	640 × 480	20.0 fps	Static background
Chuang [15] et al.	PC	2048 × 2048	5.0 fps	Dark environment
Lei [41] et al.	PC	352 × 288	3.3 fps	Swimming pool
Walther [42] et al.	PC	720 × 480	30.0 fps	Dark environment

(2) In the detection and tracking test phase, four images sequences captured in various amphibious scenarios were used to evaluate functional performance of the system. The ground truth of detection and tracking were annotated manually. The proposed detection subsystem was compared with the GBM-based detection algorithm. The proposed tracking subsystem was compared with three state-of-the-art discriminative tracking algorithms (CT [38], WMIL [35], and HOG-SVM [43]) and five classical tracking algorithms (TemplateMatch, MeanShift, VarianceRatio, PeakDifference and RatioShift [44]) which were widely used in robotics. Four metrics were used to evaluate the functional performance of the detection and tracking system. The first metric is the percentage of wrong classifications (*PWC*) of the detection process, defined as:

$$PWC = (FP + FN) / (TP + TN + FP + FN) \times 100, \quad (15)$$

where *TP*, *TN*, *FP*, and *FN* represent the number of true positive, true negative, false positive, and false negative pixels, respectively. The second metric is the precision (*Pr*) of the detection process, defined as:

$$Pr = TP / (TP + FP). \quad (16)$$

The third metric is the success rate (*SR*) of the tracking process, defined as:

$$SR = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)}, \quad (17)$$

where *ROI_T* is the tracked bounding box, *ROI_G* is the ground truth bounding box, and *area*(·) denotes the number of pixels in the region. If the *score* is larger than the given threshold (0.5 in this study) in a frame, it counts as a success. The fourth metric is the center location error (*CLE*), which is the Euclidean distance between the central points of the tracked bounding box and the ground truth bounding box.

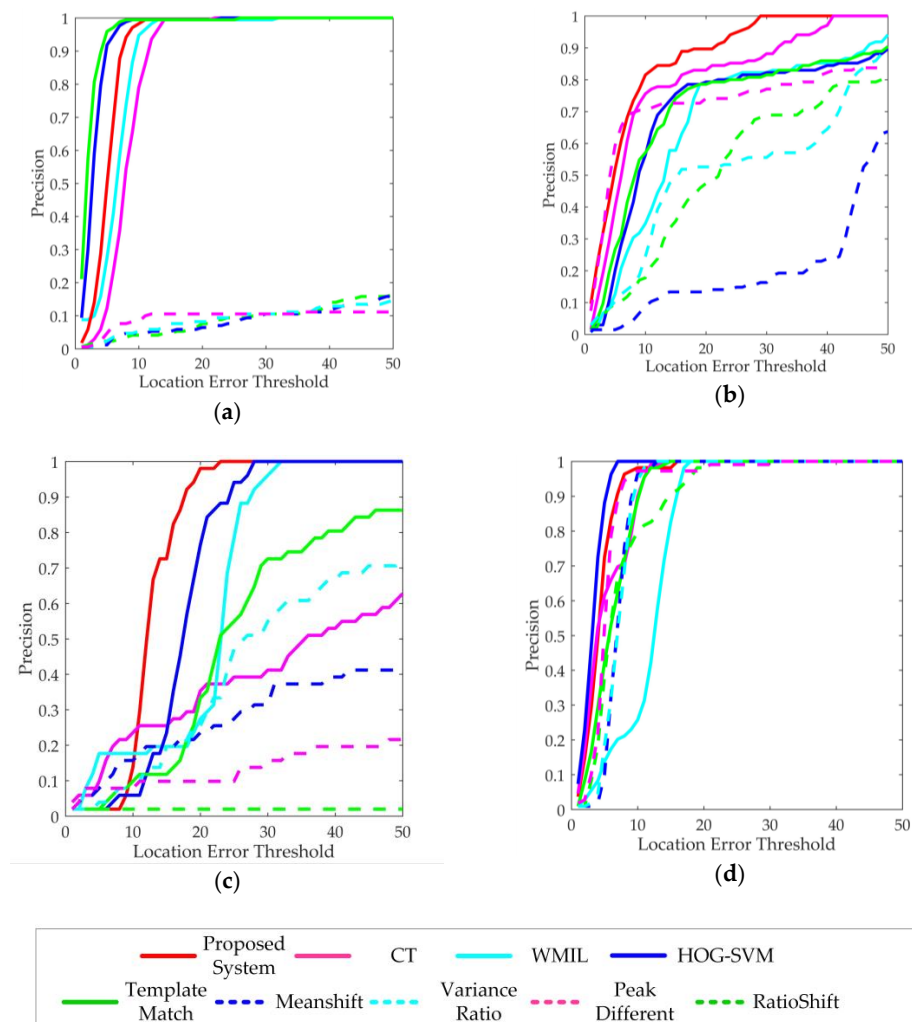
As shown in Table 2, the proposed detection subsystem provided the 47.7% lower *PWC* and the 17.2% higher *Pr* on average than the GBM-based detection algorithm. Thus, the proposed detection subsystem was more robust to environmental disturbances. As shown in Table 3 and Figure 11, the proposed tracking subsystem outperformed other tracking algorithms in terms of the *SR* and the *CLE*. The five classical tracking algorithms were not able to steadily track underwater targets in the tests of Sequence 1, Sequence 2, and Sequence 3 because they adopted static or coarse appearance models. The CT and WMIL algorithms adopted adaptive appearance models, but they were lack of effective motion prediction or dynamic update mechanisms. Thus, they did not perform well in some scenarios due to the drift problem. The HOG-SVM algorithm adopted an effective feature extractor and a strong classifier. Thus, it performed better than the proposed tracking subsystem in the tests of Sequence 1 and Sequence 4. But it was a non-real-time tracking algorithm and could only provide a processing rate as low as 2.7 fps. Thus, it was not suitable for the applications of the amphibious spherical robot. In general, the discriminative tracking algorithms using adaptive appearance models performed better than the classical tracking algorithms, especially in the underwater environments.

Table 2. Experimental results of the visual detection subsystem.

Sequences	PWC (Proposed)	Pr (Proposed)	PWC (GBM)	Pr (GBM)
Sequences 1	0.018	0.821	0.092	0.733
Sequences 2	0.069	0.675	0.183	0.484
Sequences 3	0.030	0.985	0.052	0.924
Sequences 4	0.254	0.784	0.382	0.564

Table 3. Experimental results of the visual tracking subsystem.

Algorithm	Criteria	Sequence 1	Sequence 2	Sequence 3	Sequence 4
Proposed	SR (CLE)	100 (11.7)	91.8 (21.2)	100 (17.8)	100 (6.6)
CT	SR (CLE)	98.8 (13.8)	87.1 (27.1)	71.3 (27.1)	100 (8.6)
WMIL	SR (CLE)	100 (12.2)	77.3 (29.3)	98.7 (20.1)	92.1 (12.8)
HOG-SVM	SR (CLE)	100 (6.5)	85.1 (27.4)	100 (18.7)	100 (3.7)
TemplateMatch	SR (CLE)	100 (6.1)	84.7 (28.1)	80.3 (23.1)	100 (9.8)
MeanShift	SR (CLE)	10.3 (58.1)	14.3 (53.2)	35.4 (62.3)	100 (9.8)
VarianceRatio	SR (CLE)	12.2 (59.2)	50.6 (33.2)	56.3 (36.7)	100 (9.7)
PeakDifference	SR (CLE)	12.0 (58.9)	72.1 (31.7)	16.3 (67.2)	100 (7.2)
RatioShift	SR (CLE)	11.9 (45.6)	67.4 (28.2)	3.2 (87.3)	100 (8.4)

**Figure 11.** Precision plot of tracking for test image sequences. (a) Precision plot for Sequence 1; (b) Precision plot for Sequence 2; (c) Precision plot for Sequence 3; and (d) Precision plot for Sequence 4.

As shown in Figures 12 and 13, two underwater videos of fishes provided by the Fish4Knowledge project [19] were used to evaluate the performance of the proposed system towards underwater targets.

Sequence 1 was collected from the underwater observatory at Orchid Island, Taiwan. A distant moving fish with a stable motion trajectory was selected as the target. Due to the light scattering and absorption effect of ocean water, the captured underwater images were blurry, and the appearance characteristics of the target were not significant. In the test of Sequence 1, the proposed system was able to detect and then track the small moving fish with high accuracy. Most trackers without image pre-processing successively lost the target because the appearance characteristics were not so significant. Test results of Sequence 1 demonstrated that the proposed system was capable of detecting and tracking practical target in the undersea environment.

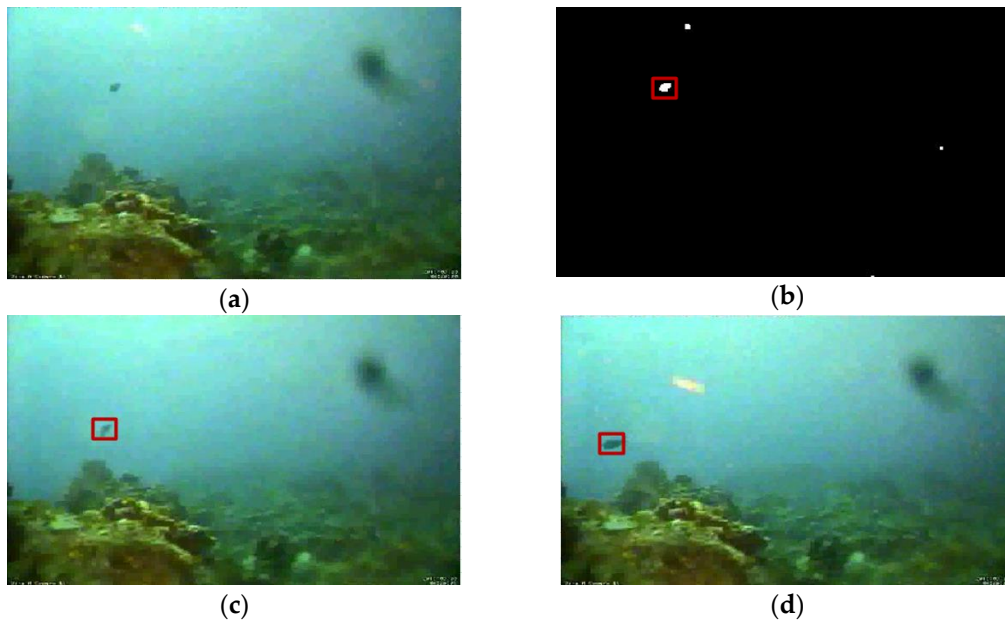


Figure 12. Experimental results of Sequence 1. (a) Image collected from the underwater observatory; (b) Detection result; (c) Tracking result; and (d) Tracking result.

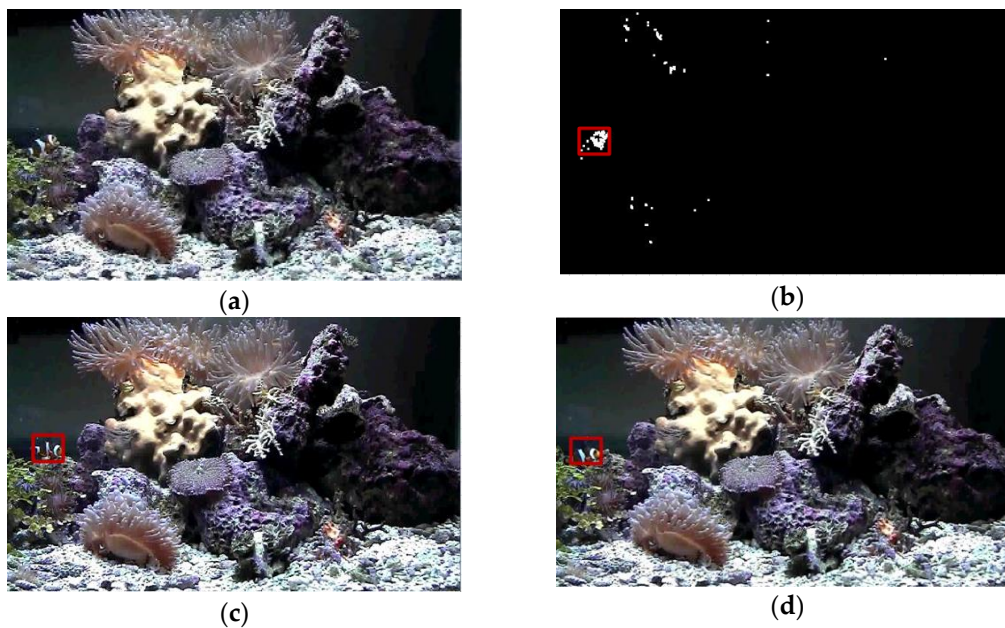


Figure 13. Experimental results of Sequence 2. (a) Image collected from the underwater observatory; (b) Detection result; (c) Tracking result; and (d) Tracking result.

Sequence 2 was collected from the underwater observatory at National Museum of Marine Biology and Aquarium, Taiwan. A tropical fish moving randomly was selected as the target. The captured images were clear, and the target had obvious texture features. However, the ever-changing motion trajectory of the target and the swaying corals in the background would mislead the robotic vision system. The disturbance of swaying corals in Sequence 2 was neglected in the detection process by using the GMM-based method, ensuring the correct detection of the tropical fish. The GBM-based method got disturbed and provided higher error rate. However, because the proposed tracking subsystem as well as the three discriminative tracking algorithms do not have scale invariant and affine invariance properties, the visual trackers finally lost the target when the fish changed poses. The five classical tracking algorithms lost the target soon because the disturbances caused by similar objects in the background. Test results of Sequence 2 verified that the proposed system could provide relatively accurate detection and tracking results when working in the complex and cluttered underwater environment.

Two videos captured by the amphibious spherical robot in underwater and terrestrial environments were used for evaluation, as shown in Figures 14 and 15. Sequence 3 was collected from the amphibious spherical robot in a tank. A small toy fish swimming fast was adopted as the moving target. The image quality of Sequence 3 was better than that of Sequence 1. But the robotic platform rocked slowly with the water in a practical underwater scenario, which would present difficulties for the detection and tracking process. In the test of Sequence 3, the detection precision of the proposed system was acceptable even though the robotic platform was not so steady. Because the robot had a small view field and a low frame rate, the fish model swam at a relatively fast speed in the video. By predefining appropriate parameters for the Kalman filter, the proposed system could stably track the fish after detecting it. And the drift problem occurred when using the original CT algorithm, which resulted in worse tracking performance. The five classical trackers failed in the tracking process because they cannot adapt to the ever-changing target. Test results of Sequence 3 verified that the proposed system was capable of detecting and tracking the target object moving at fast speed and it could meet the application requirements of the amphibious target tracking in underwater environments.

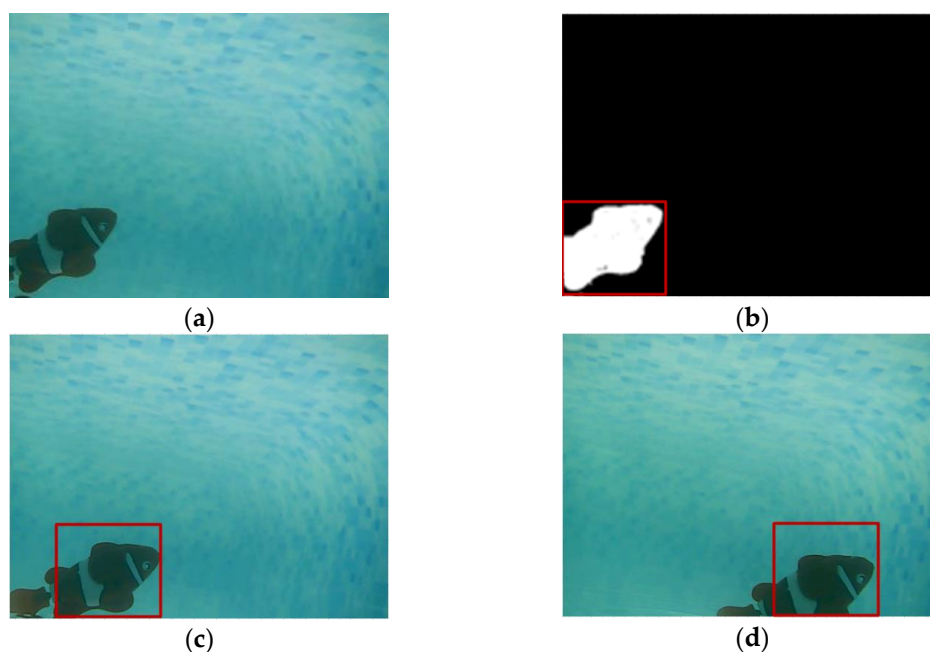


Figure 14. Experimental results of Sequence 3. (a) Image captured by the robot; (b) Detection result; (c) Tracking result; and (d) Tracking result.

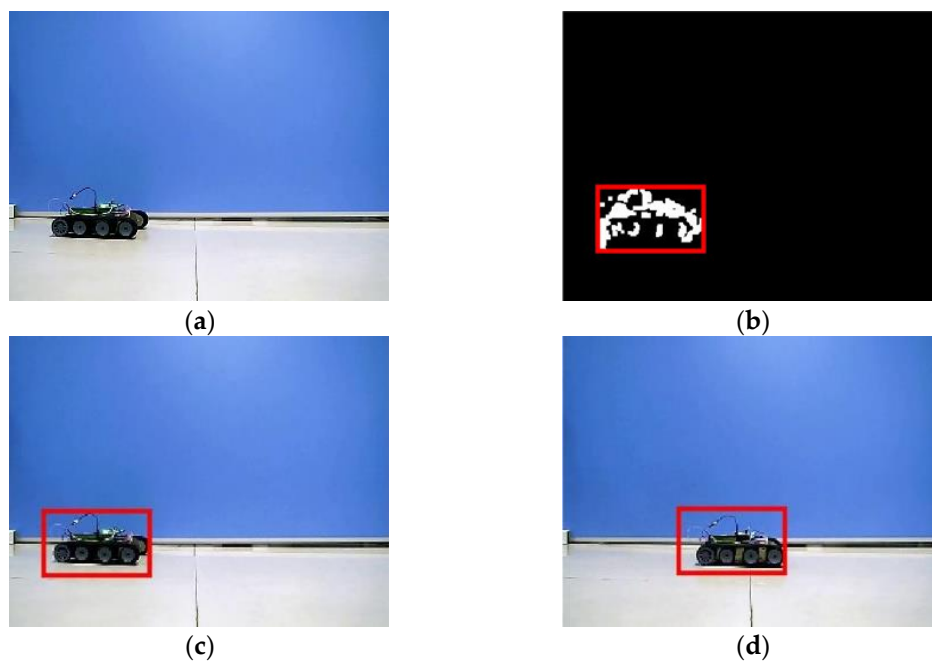


Figure 15. Experimental results of Sequence 4. (a) Image captured by the robot; (b) Detection result; (c) Tracking result; and (d) Tracking result.

Sequence 4 was collected from the amphibious spherical robot in the laboratory environment. A small tracked robot moving at a low speed was adopted as the moving target. The robotic platform was relatively stable, and the image quality was good. But the appearance characteristics of the target slowly changed, which might lead to the drift problem in the tracking process. In the test of Sequence 4, the detection results were not so good because the motion speed of the small car was slow. However, the target region of the small car was recognized correctly, and the GMM-based detector provided better PWC and Pr than the GBM-based detector. Except that the WMIL tracker encountered the drift problem, all the trackers were capable of successively tracking the specified target which remained nearly unchanged. That demonstrated that visual tracking in underwater environments is a much more challenge work than that in terrestrial environments from another side. Because most studies on robotic vision were conducted on land, which might be not suitable for underwater applications, optimizations were essential in the design of visual detection and tracking system for amphibious spherical robots. Test results of Sequence 4 verified that the proposed system was able to steadily detect and track the target object on land. And it could meet the application requirements of the amphibious in terrestrial environments.

7. Conclusions and Future Work

To meet the practical application requirements of the spherical amphibious robot in ecological observations and intelligent surveillance tasks, an embedded detection and tracking system was designed and implemented. To address the image degradation problem in underwater scenarios, captured images were pre-processed with the MSRCR algorithm to reduce the effects of light absorption and scattering. Then, the Gaussian mixture model was used to detect moving targets entering the robot's view field. The marked target was tracked successively using a FCT tracker with a Kalman prediction mechanism. Using these algorithms with online learning mechanisms, the designed detection and tracking subsystems were able to resist disturbances, such as the swaying aquatic plants in the detection stage and the fast motion of a target in the tracking stage. Considering the unique mechanical structure and limited load space of the robot, the whole vision system was integrated into a low-power SoC using an asymmetric and heterogeneous computing architecture. Evaluation

experiments confirmed the validation and efficiency of the proposed system. The proposed system was capable of precisely detecting and tracking various target objects in both underwater and terrestrial environments in real time. With the features of low power consumption, high-real-time performance, and good environmental adaptability, it was able to meet the potential demands of the small-sized spherical amphibious robot in multi-robot cooperation and multi-target tracking tasks. As far as we know, it was the first practical visual detection and tracking system towards generic targets for small-sized amphibious robots. In comparison with most relevant studies, the proposed system provided higher detection and tracking accuracy by implementing adaptive visual algorithms and introducing improvement methods. Built upon a heterogeneous embedded system, it could fit in well with the characteristics of small-sized amphibious and underwater robots.

The proposed system has several drawbacks. First, the MSRCR algorithm does not have adaptability towards different environments. Consequently, the algorithm parameters had to be adjusted carefully before use. This may limit the applications of the robot in ever-changing environments. Second, the detection and tracking algorithms used in the system are not sufficiently robust or precise for long-term robotic vision applications. Our future work will focus on high-level vision applications, including automatic navigation and object grabs. Additionally, advanced visual algorithms and tools including convolutional neural networks will be used to improve the designed robotic vision system.

Acknowledgments: This work was supported by National Natural Science Foundation of China (61503028, 61375094), Excellent Young Scholars Research Fund of Beijing Institute of Technology (2014YG1611), and the Basic Research Fund of the Beijing Institute of Technology (20151642002). This research project was also partly supported by National High Tech. Research and Development Program of China (No.2015AA043202).

Author Contributions: Shaowu Pan conceived the robotic vision system and wrote the paper. Shuxiang Guo guide the system design and revised the manuscript. Liwei Shi and Ping Guo analyzed the data. Yanlin He designed key mechanical parts of the improved version of the amphibious spherical robot. Kun Tang performed the experiments in amphibious environments.

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

References

1. Thompson, D.; Caress, D.; Thomas, H.; Conlin, D. MBARI mapping AUV operations in the gulf of California 2015. In Proceedings of the OCEANS 2015 - MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015.
2. Tran, N.-H.; Choi, H.-S.; Bae, J.-H.; Oh, J.-Y.; Cho, J.-R. Design, control, and implementation of a new AUV platform with a mass shifter mechanism. *Int. J. Precis. Eng. Manuf.* **2015**, *16*, 1599–1608. [[CrossRef](#)]
3. Ribas, D.; Palomeras, N.; Ridao, P.; Carreras, M.; Mallios, A. Girona 500 auv: From survey to intervention. *IEEE ASME Trans. Mechatron.* **2012**, *17*, 46–53. [[CrossRef](#)]
4. Shi, L.; Guo, S.; Mao, S.; Yue, C.; Li, M.; Asaka, K. Development of an amphibious turtle-inspired spherical mother robot. *J. Bionic Eng.* **2013**, *10*, 446–455. [[CrossRef](#)]
5. Kaznov, V.; Seeman, M. Outdoor navigation with a spherical amphibious robot. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010.
6. Jia, L.; Hu, Z.; Geng, L.; Yang, Y.; Wang, C. The concept design of a mobile amphibious spherical robot for underwater operation. In Proceedings of the 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Chengdu, China, 19–22 June 2016.
7. Chen, W.-H.; Chen, C.-P.; Tsai, J.-S.; Yang, J.; Lin, P.-C. Design and implementation of a ball-driven omnidirectional spherical robot. *Mech. Mach. Theory* **2013**, *68*, 35–48. [[CrossRef](#)]
8. Guo, S.; He, Y.; Shi, L.; Pan, S.; Tang, K.; Xiao, R.; Guo, P. Modal and fatigue analysis of critical components of an amphibious spherical robot. *Microsyst. Technol.* **2016**, 1–15. [[CrossRef](#)]
9. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: A review. *IEEE J. Ocean. Eng.* **2014**, *39*, 131–149. [[CrossRef](#)]

10. Grothues, T.M.; Dobarro, J.; Eiler, J. Collecting, interpreting, and merging fish telemetry data from an AUV: Remote sensing from an already remote platform. In Proceedings of the 2010 IEEE/OES Autonomous Underwater Vehicles, Monterey, CA, USA, 1–3 September 2010.
11. Bosch Alay, J.; Grácias, N.R.E.; Ridao Rodríguez, P.; Istenič, K.; Ribas Romagós, D. Close-range tracking of underwater vehicles using light beacons. *Sensors* **2016**, *16*, 429. [[CrossRef](#)] [[PubMed](#)]
12. Massot-Campos, M.; Oliver-Codina, G. Optical sensors and methods for underwater 3D reconstruction. *Sensors* **2015**, *15*, 31525–31557. [[CrossRef](#)] [[PubMed](#)]
13. Yahya, M.; Arshad, M. Tracking of multiple light sources using computer vision for underwater docking. *Procedia Comput. Sci.* **2015**, *76*, 192–197. [[CrossRef](#)]
14. Zhang, L.; He, B.; Song, Y.; Yan, T. Consistent target tracking via multiple underwater cameras. In Proceedings of the OCEANS 2016 - Shanghai, Shanghai, China, 10–13 April 2016.
15. Chen, Z.; Shen, J.; Fan, T.; Sun, Z.; Xu, L. Single-camera three-dimensional tracking of underwater objects. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2015**, *8*, 89–104. [[CrossRef](#)]
16. Chuang, M.C.; Hwang, J.N.; Williams, K.; Towler, R. Multiple fish tracking via Viterbi data association for low-frame-rate underwater camera systems. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013), Beijing, China, 19–23 May 2013.
17. Chuang, M.-C.; Hwang, J.-N.; Ye, J.-H.; Huang, S.-C.; Williams, K. Underwater Fish Tracking for Moving Cameras Based on Deformable Multiple Kernels. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *PP*, 1–11. [[CrossRef](#)]
18. Lee, D.; Kim, G.; Kim, D.; Myung, H.; Choi, H.-T. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Eng.* **2012**, *48*, 59–68. [[CrossRef](#)]
19. Shiau, Y.-H.; Chen, C.-C.; Lin, S.-I. Using bounding-surrounding boxes method for fish tracking in real world underwater observation. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 261–270. [[CrossRef](#)]
20. Li, M.; Guo, S.; Hirata, H.; Ishihara, H. A roller-skating/walking mode-based amphibious robot. *Robot. Comput. Integr. Manuf.* **2017**, *44*, 17–29. [[CrossRef](#)]
21. Li, Y.; Guo, S. Communication between spherical underwater robots based on the acoustic communication methods. In Proceedings of the 2016 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, China, 7–10 August 2016.
22. Pan, S.; Shi, L.; Guo, S.; Guo, P.; He, Y.; Xiao, R. A low-power SoC-based moving target detection system for amphibious spherical robots. In Proceedings of the 2015 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 2–5 August 2015.
23. Pan, S.; Shi, L.; Guo, S. A Kinect-based real-time compressive tracking prototype system for amphibious spherical robots. *Sensors* **2015**, *15*, 8232–8252. [[CrossRef](#)] [[PubMed](#)]
24. Crockett, L.H.; Elliot, R.A.; Enderwitz, M.A.; Stewart, R.W. *The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*; Strathclyde Academic Media: Glasgow, UK, 2014; pp. 15–21.
25. Schechner, Y.Y.; Karpel, N. Clear underwater vision. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004.
26. Roser, M.; Dunbabin, M.; Geiger, A. Simultaneous underwater visibility assessment, enhancement and improved stereo. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014.
27. Jobson, D.J.; Rahman, Z.U.; Woodell, G.A. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Trans. Image Process.* **1997**, *6*, 965–976. [[CrossRef](#)] [[PubMed](#)]
28. Xiao, S.; Li, Y. Fast multiscale Retinex algorithm of image haze removal with color fidelity. *Comput. Eng. Appl.* **2015**, *51*, 176–179.
29. Liu, Y.; Yao, H.; Gao, W.; Chen, X.; Zhao, D. Nonparametric background generation. *J. Vis. Commun. Image Represent.* **2007**, *18*, 253–263. [[CrossRef](#)]
30. Negrea, C.; Thompson, D.E.; Juhnke, S.D.; Fryer, D.S.; Loge, F.J. Automated detection and tracking of adult pacific lampreys in underwater video collected at snake and Columbia River fishways. *North Am. J. Fish. Manag.* **2014**, *34*, 111–118. [[CrossRef](#)]
31. KaewTraKulPong, P.; Bowden, R. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*; Springer: Berlin, German, 2002; pp. 135–144.

32. Mukherjee, D.; Wu, Q.J.; Nguyen, T.M. Gaussian mixture model with advanced distance measure based on support weights and histogram of gradients for background suppression. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1086–1096. [CrossRef]
33. Ibarburen, A.; Martínez-Otzeta, J.M.; Maurtua, I. Particle filtering for industrial 6DOF visual servoing. *J. Intell. Robot. Syst.* **2014**, *74*, 689–696. [CrossRef]
34. Yang, S.; Scherer, S.A.; Schauwecker, K.; Zell, A. Autonomous landing of MAVs on an arbitrarily textured landing site using onboard monocular vision. *J. Intell. Robot. Syst.* **2014**, *74*, 27–43. [CrossRef]
35. Zhang, K.; Song, H. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognit.* **2013**, *46*, 397–411. [CrossRef]
36. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
37. Zhang, K.; Liu, Q.; Wu, Y.; Yang, M.-H. Robust visual tracking via convolutional networks without training. *IEEE Trans. Image Process.* **2016**, *25*, 1779–1792. [CrossRef] [PubMed]
38. Zhang, K.; Zhang, L.; Yang, M.-H. Fast compressive tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2002–2015. [CrossRef] [PubMed]
39. Wu, Y.; Lim, J.; Yang, M. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
40. Visual Tracker Benchmark. Available online: <http://www.visual-tracking.net> (accessed on 26 April 2017).
41. Lei, F.; Zhang, X. Underwater target tracking based on particle filter. In Proceedings of the 2012 7th International Conference on Computer Science & Education (ICCSE), Melbourne, Australia, 14–17 July 2012.
42. Walther, D.; Edgington, D.R.; Koch, C. Detection and tracking of objects in underwater video. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004.
43. Wang, N.; Shi, J.; Yeung, D.-Y.; Jia, J. Understanding and Diagnosing Visual Tracking Systems. In Proceedings of the 2015 IEEE International Conference on Computer Vision (CVPR 2015), Santiago, CA, USA, 7–13 December 2015.
44. An Open Source Tracking Testbed and Evaluation Web Site. Available online: <http://vision.cse.psu.edu/publications/pdfs/opensourceweb.pdf> (accessed on 15 December 2016).



© 2017 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/>).