

Article

A Kinect-Based Real-Time Compressive Tracking Prototype System for Amphibious Spherical Robots

Shaowu Pan ^{1,2}, Liwei Shi ^{1,2,*} and Shuxiang Guo ^{1,2,3}

¹ The Institute of Advanced Medical Engineering System, School of Life Science, Beijing Institute of Technology, No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China; E-Mails: panshaowu.hit@gmail.com (S.P.); guoshuxiang@bit.edu.cn (S.G.)

² Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, the Ministry of Industry and Information Technology, Beijing Institute of Technology, No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China

³ Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Kagawa 760-8521, Japan

* Author to whom correspondence should be addressed; E-Mail: shiliwei@bit.edu.cn; Tel./Fax: +86-10-6891-5908.

Academic Editor: Vittorio M.N. Passaro

Received: 31 January 2015 / Accepted: 30 March 2015 / Published: 8 April 2015

Abstract: A visual tracking system is essential as a basis for visual servoing, autonomous navigation, path planning, robot-human interaction and other robotic functions. To execute various tasks in diverse and ever-changing environments, a mobile robot requires high levels of robustness, precision, environmental adaptability and real-time performance of the visual tracking system. In keeping with the application characteristics of our amphibious spherical robot, which was proposed for flexible and economical underwater exploration in 2012, an improved RGB-D visual tracking algorithm is proposed and implemented. Given the limited power source and computational capabilities of mobile robots, compressive tracking (CT), which is the effective and efficient algorithm that was proposed in 2012, was selected as the basis of the proposed algorithm to process colour images. A Kalman filter with a second-order motion model was implemented to predict the state of the target and select candidate patches or samples for the CT tracker. In addition, a variance ratio features shift (VR-V) tracker with a Kalman estimation mechanism was used to process depth images. Using a feedback strategy, the depth tracking results were used to assist the CT tracker in updating classifier parameters at an adaptive rate. In this way, most of the deficiencies of CT, including drift and poor robustness to occlusion and high-speed

target motion, were partly solved. To evaluate the proposed algorithm, a Microsoft Kinect sensor, which combines colour and infrared depth cameras, was adopted for use in a prototype of the robotic tracking system. The experimental results with various image sequences demonstrated the effectiveness, robustness and real-time performance of the tracking system.

Keywords: compressive tracking; Kinect; tracking system; real time; mobile robot

1. Introduction

To execute various tasks autonomously in ever-changing environments, it is critical for a robot to be able to detect its surroundings. As a kind of feasible sensor with advantages of low cost, low power consumption and strong adaptability [1], digital cameras have been widely used in robotics to guide electromechanical devices and realise intelligent methods. In a machine vision-based robot, the visual tracking system plays a very important role in realising diverse robotic functions, such as autonomous navigation [2,3], path planning [4,5], visual servoing [6,7], robot-human interaction [8,9], *etc.*

In general, existing tracking algorithms can be categorised as estimation-based and classification-based algorithms [10]. Estimation-based or generative algorithms model the target based on appearance features and then search for it in each frame of the visual stream [11]. Examples of this type of algorithm, which are mainly driven by innovations in appearance models, include MeanShift [12], particle filter-based algorithms [13], Incremental Visual Tracking (IVT) [14] and optical flow-based algorithms [15]. Classification-based or discriminative algorithms treat tracking as a binary pattern recognition problem and try to separate the target from the background [16]. This type is usually built upon pattern recognition algorithms, such as support vector machine (SVM) [17], Bayes classifier [18], K-means [19], *etc.* Most studies on visual tracking are performed on high-performance computers and are evaluated with standard benchmark sequences, each of which contain controlled disturbances in the target or environment.

Compared with studies of tracking algorithms based on commercial computers in a controlled laboratory environment, designing a reliable visual tracking system for mobile robots is an even more challenging task for two reasons [20,21]. First, the appearance of targets varies due to pose or scale changes, random motion and occlusion, making it difficult to establish a reliable appearance model [22]. Second, the ambient environment of robots can be easily disturbed by variations in illumination, camera vibration and outside interference, which may cause problems of drift or target loss [23]. More importantly, most state-of-the-art studies aimed at improving tracking precision, adaptability and robustness, overlooking the real-time performance and computational requirements of algorithms. This has led to a situation in which most existing algorithms are unsuitable for application to mobile robots equipped with embedded microprocessors and limited power sources.

To realise flexible and economical underwater exploration, in 2012 our team designed an amphibious spherical robot inspired by turtles [24–26]; an improved version conceptualised as “all programmable” was designed in 2014 [27]. Given the special working environments and application functions of the amphibious robot, an improved RGB-D visual tracking algorithm with dual trackers is

proposed and implemented in this paper. Compressive tracking (CT) was selected as the basis of the proposed algorithm to process colour images from a RGB-D camera, and a Kalman filter with a second-order motion model was added to the CT tracker to predict the state of the target, select candidate patches or samples and reinforce the tracker's robustness to high-speed moving targets. In addition, a variance ratio features shift (VR-V) tracker with a Kalman prediction mechanism was adopted to process depth images from a RGB-D camera. A visible and infrared fusion mechanism or feedback strategy is introduced in the proposed algorithm to enhance its adaptability and robustness. Through the feedback strategy, the depth tracking results were used to assist the CT tracker in updating classifier parameters at an adaptive rate. To evaluate the effectiveness of the algorithm, a Microsoft Kinect device, which is a combination of colour and depth cameras, was adopted for use in a prototype of the robotic tracking system. Experimental results in various scenarios demonstrated the effectiveness, robustness and real-time performance of the tracking system. Furthermore, drift, which is an obvious problem of CT, was partly solved in the proposed algorithm.

The rest of this paper is organised as follows: an overview of related work, which covers our amphibious spherical robot and CT, is presented in Section 2. Three major areas of the proposed algorithm are elaborated in Section 3 and the structure of the prototype tracking system for our amphibious spherical robot and experimental results on various benchmark sequences are covered in Section 4. Section 5 presents the conclusion and suggestions for relevant follow-up research.

2. Related Works and Application Requirements

2.1. Amphibious Spherical Robot

As introduced in references [24] and [27], an amphibious spherical robot was proposed for covert missions or tasks in narrow spaces that a typical autonomous underwater vehicle (AUV) could not complete. As shown in Figure 1, the shell of the robot consisted of a hemispheric upper hull (250 mm in diameter) and two quarter-sphere lower hulls (266 mm in diameter) that can open and close. The hard upper hull was waterproof and served to protect the internal electronic devices and batteries from collisions.

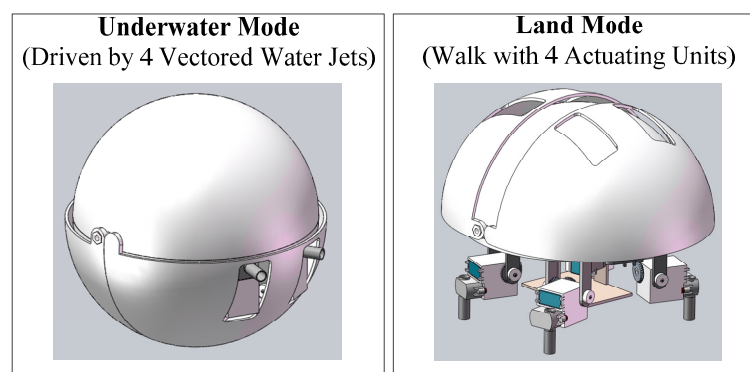


Figure 1. Diagram of our spherical amphibious robot.

As shown in Figure 2, four actuating units, each consisting of two servomotors and a water jet motor, were symmetrically installed under the upper hull. When the robot was in underwater mode, the

lower hulls closed, and four water jet motors in the actuating units provided vectored thrust through strip-type holes in the lower hulls to realise motion with six degrees of freedom (DOF). When the robot was in land mode, the lower hulls opened, and the actuating units stretched out to walk quadrupedally under the driving force of eight servomotors. To facilitate upgrades, three-dimensional (3-D) printing technology, a minimal Xilinx Zynq-7000 SoC system and an embedded computer equipped with an Intel Atom processor were adopted to fabricate the robot [26].

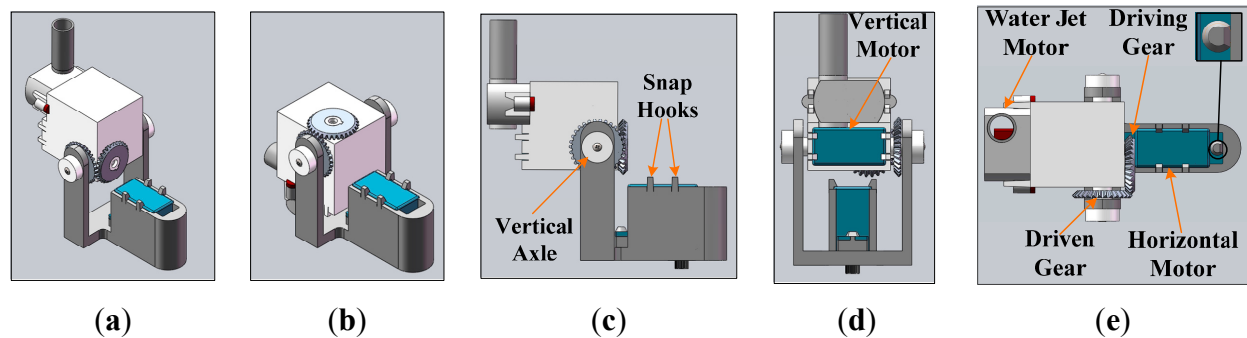


Figure 2. Diagram of actuating units and related driving parts. (a) Stretching mode; (b) Huddling mode; (c) Frontal view; (d) Profile view; and (e) Top view.

Due to the unique mechanical structure and the specialised potential application scenarios, customising the tracking system of the amphibious spherical robot was a challenging task. First, the upper hull of the robot, in which electronic devices, scientific instruments and batteries were installed, provided only a very narrow enclosed space. Hence, high-speed computers, which consume considerable power and generate a great deal of heat, could not be adopted in this mobile robot and consequently, the real-time performance of the adopted tracking algorithm became a critical problem. Second, as the robot may work under diverse land and underwater conditions, robustness to environmental disturbances caused by camera vibration and occlusion was essential to the adopted tracking algorithm. Third, the field of view of the camera was very narrow due to limitations imposed by the installation space, making it crucial to improve the adaptability of the tracking algorithm to small and high-speed moving targets [28]. Moreover, the visual tracking system of the robot was intended for use chiefly in autonomous navigation and in guiding electromechanical devices. Consequently, its tracking precision and effectiveness should be acceptable.

2.2. Real-Time Compressive Tracking Algorithms

In 2012, Zhang *et al.* proposed the CT algorithm, which provided a novel framework for developing real-time tracking algorithms for mobile robots [29]. Figure 3 shows the main components of the CT algorithm. As an effective and efficient discriminative algorithm, CT consists of two stages: Tracking and updating. In the tracking stage, candidate image patches or samples of the target of the $(n + 1)$ -th frame are sampled around I_n , which is the tracking result at the n -th frame. Then, integral vectors of these patches are calculated by accumulation. Based on compressive sensing theory, the high-dimensional integral vectors of samples are compressed to extract Haar-like features using a static measurement matrix. The process of compression or measurement can be denoted as $\mathbf{v} = \mathbf{R}\mathbf{u}$, where

$\mathbf{u} \in \mathbb{R}^n$ indicates the integral vectors and $\mathbf{v} \in \mathbb{R}^m$ indicates the compressed feature vectors with dimensions $m \ll n$. $\mathbf{R}$ is a very sparse random matrix, the entries of which were defined as:

$$r_{i,j} = \sqrt{s} \times \begin{cases} +1, \text{with probability } \frac{1}{2s} \\ 0, \text{with probability } 1 - \frac{1}{s} \\ -1, \text{with probability } \frac{1}{2s} \end{cases} \quad (1)$$

where s is set to $m/4$. For each row of $\mathbf{R}$, fewer than four entries are non-zero, which results in very low computational complexity of the compression process. Then, the low-dimensional feature vectors are entered into an online learning Naïve Bayes classifier. The sample with maximal classifier response is set to the target for determining I_{n+1} . In the updating stage, training samples of the target and the background are oversampled according to the tracking result at the $(n+1)$ -th frame (I_{n+1}), and the compressed feature vectors of the training samples are used to update the parameters of the Naïve Bayes classifier, which will be used in the tracking stage of the $(n+2)$ -th frame.

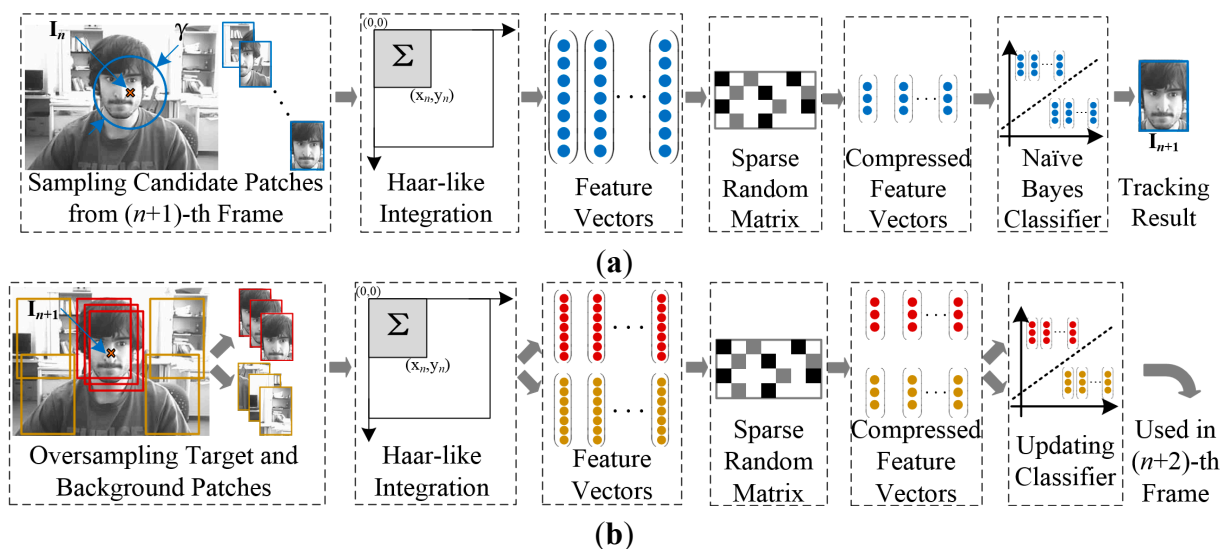


Figure 3. Diagram of the original CT algorithm. (a) Tracking at the $(n+1)$ -th frame; and (b) Updating classifier after Tracking at the $(n+1)$ -th frame.

The CT algorithm succeeded in real-time performance and provided a low-cost and lossless way to compress image information and exact appearance features by using compressive sensing theory. However, CT samples the candidate patches within a predefined radius around the tracking result of the previous frame. Consequently, tracking precision relies heavily on the value of the radius, and CT is unable to track targets in complex or high-speed motion. Moreover, as the parameters of the classifier are updated at a constant rate, the tracking results continue to drift once a deviation occurs. Furthermore, the target will be lost if it is fully occluded for a long time. In addition, scale invariance is not taken into account in CT, which exacerbates the problem of tracking drift.

To compensate for the deficiencies of CT, Xu *et al.* proposed an improved CT algorithm containing two weighted classifiers that processed previous and current samples separately [30]. Chen *et al.* proposed a scalable CT algorithm that used optical flow to estimate the size of the target [18]. Yan *et al.*

combined CT with sparse representation and proposed a kernel sparse tracking algorithm [31]. Zhang *et al.* added a coarse-to-fine search strategy to CT and proposed a fast compressive tracking (FCT) algorithm [32]. However, improvements introduced in the above studies added significant complexity to the algorithm or only solved a specialised problem, limiting their applications in our amphibious spherical robot.

3. Proposed Algorithm

On the other hand, the drift problem of CT is associated with the accumulation of tracking errors caused by the online self-updating process of the classifier. Therefore, adding a semi-supervised or feedback mechanism to the updating process should be an efficient method for addressing the drift problem [10].

On the other hand, as digital images are projections of 3-D objects on a two-dimensional (2-D) plane, visual tracking is a process of estimation based on incomplete information. Consequently, stereovision and multi-sensor fusion, which provide additional information about the target (depth, temperature, velocity, *etc.*), are effective means of improving the precision and robustness of tracking algorithms. For the reason that a depth image directly provides the three-dimensional information of the surface of an object, it can be used in RGB-D tracking algorithms to handle thorny problems such as occlusion, drift and clutter environment. The Microsoft Kinect, which is a low-cost commercial RGB-D camera proposed in 2012, has promoted the development of RGB-D tracking algorithms in recent years [33]. As Kinect is close to a medium-resolution stereo camera in resolution and precision of the depth measurement [34], it has been successfully used to fabricate robust tracking systems by combining colour and depth features [35]. Because most available vision algorithms designed for colour images cannot be directly extended to detect or track targets in depth images, histograms of oriented gradients (HOG), histogram of oriented normal vectors (HONV) and local ternary patterns (LTP) were usually adopted to extract depth features of targets in existing Kinect-based RGB-D tracking algorithms [36,37]. However, these algorithms are unsuitable to embedded microprocessor-based robotic applications in consideration of their real-time performance [38,39].

Making use of the features of Kinect and CT, an improved visual tracking algorithm for application of our amphibious spherical robot is presented based on the concept of multi-feature fusion and feedback presented in this paper. The main components of the proposed algorithm are shown in Figure 4. Microsoft Kinect was adopted as a compound RGB-D image sensor for the prototype tracking system, and dual trackers were implemented to process colour and depth images separately. An improved CT tracker with a Kalman prediction mechanism was deployed as the main tracker to process colour images (C-Frame), and a VR-V tracker was implemented as an assist tracker to process depth images (D-Frame). To realise sensor fusion and improve robustness of the proposed algorithm, a feedback strategy inspired by expert control was designed to fuse the tracking results of the trackers. The major contributions of the proposed algorithm are in three areas: Motion estimation for CT, VR-V-based depth tracking and the feedback strategy.

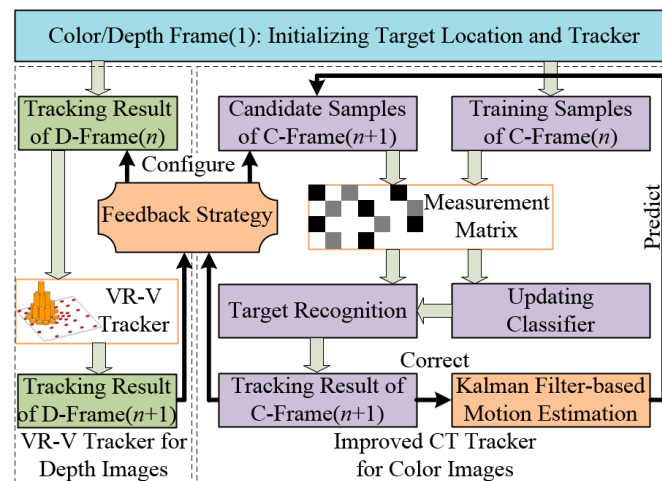


Figure 4. Main components of the proposed algorithm.

3.1. Visual Compressive Tracking with Motion Estimation

As mentioned in Section 2, the original CT algorithm samples candidate patches around the previous tracking result in a static radius set manually; this process can be denoted as:

$$\mathbf{D}^\gamma = \{z \mid \|\mathbf{I}(z) - \mathbf{I}_n\| < \gamma\} \quad (2)$$

where $\mathbf{D}^\gamma$ is a set of candidate samples at the $(n + 1)$ -th frame, $\mathbf{I}_n$ is the centre of the target at the n -th frame, and γ is the sampling radius, which was set to 25 in the sample CT program [29]. This sampling method reduces the robustness of CT to targets in complex or high-speed motion. As shown in the left-hand image of Figure 5, because the radius is static, the desired tracking result $\tilde{\mathbf{I}}_{n+1}$ may be on the edge of or even outside the sampling region. This can lead to visual drift or to missing the target entirely. This problem can be partially solved by setting the sampling radius to a larger value, but this will significantly increase the number of candidate patches to be sampled and processed, which decreases the superiority of CT in real-time performance.

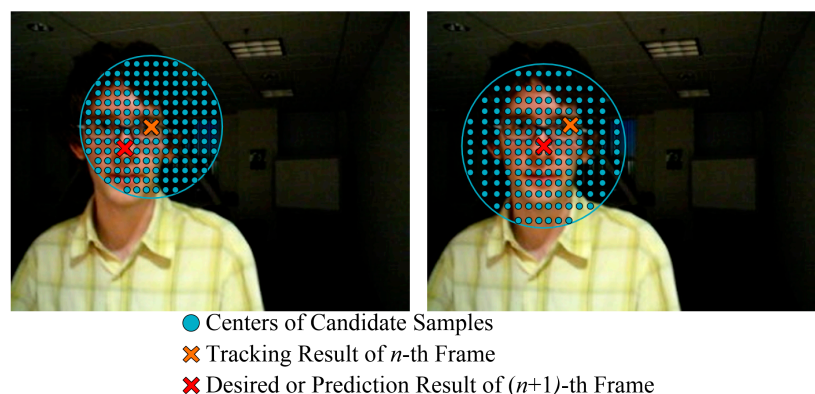


Figure 5. Diagram of candidate patch sampling.

As shown in the right-hand image of Figure 5, to balance the real-time performance and robustness of CT, the centre of the sampling region was modified so that it was located at the predicted tracking result for the $(n + 1)$ -th frame, and a Kalman filter was deployed to estimate the sampling centre.

Considering the prediction and tracking errors of the improved CT tracker, the location and motion tendency of the target can be estimated with the linear dynamic model denoted as:

$$\mathbf{X}_{n+1} = \Phi \mathbf{X}_n + \beta \mathbf{W}_n \quad (3)$$

$$\mathbf{Y}_{n+1} = \mathbf{H} \mathbf{X}_{n+1} + \alpha \mathbf{V}_n \quad (4)$$

$$\Phi = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (6)$$

$$\mathbf{X}_n = (x_n, y_n, x_n - x_{n-1}, y_n - y_{n-1})^T \quad (7)$$

$$\mathbf{Y}_n = (\tilde{x}_n, \tilde{y}_n)^T \quad (8)$$

where Φ is the transfer matrix, $\mathbf{H}$ is the measurement matrix, α and β are adjustable parameters of the filter, $\mathbf{W}_n$ is the noise vector of prediction and $\mathbf{V}_n$ is the noise vector of measurement. The real state vector $\mathbf{X}_n$ contains 2-D position and velocity information on the target, and the measured state vector $\mathbf{Y}_n$ contains 2-D position information on the target. As most objects in smooth motion can be described by a third-order motion model and error in the tracking process is common, a second-order motion model was adopted in this filter to increase reliability.

The prediction process can be described as:

$$\hat{\mathbf{X}}_{n+1|n} = \Phi \hat{\mathbf{X}}_{n|n} \quad (9)$$

$$\mathbf{P}_{n+1|n} = \Phi \mathbf{P}_{n|n} \Phi^T + \beta \mathbf{Q}_{n+1} \quad (10)$$

where $\mathbf{P}_{n+1|n}$ is the covariance matrix of noise. After predicting the motion state at the $(n + 1)$ -th frame, the sampling centre was set to $\hat{\mathbf{I}}_{n+1}$, which was extracted from the prediction vector $\hat{\mathbf{X}}_{n+1|n}$, and the candidate samples centred at $\hat{\mathbf{I}}_{n+1}$ were collected based on a 2-D Gaussian distribution. As shown in the right-hand image of Figure 5, the density of candidate samples decreased with distance from the predicted location, which was closer to the actual position of the target. Consequently, as the number of samples would not increase exponentially with the sampling radius, it was possible to track high-speed moving targets using larger search regions in real-time. In addition, a matrix defining the Gaussian location of samples was calculated in advance to accelerate the sampling process.

After the CT tracking stage, which processed candidate samples and output the tracking result, the correction process can be described as:

$$\mathbf{K}_{n+1} = \mathbf{P}_{n+1|n} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{n+1|n} \mathbf{H}^T + \alpha \mathbf{R}_{n+1})^{-1} \quad (11)$$

$$\hat{\mathbf{X}}_{n+1|n+1} = \hat{\mathbf{X}}_{n+1|n} + \mathbf{K}_{n+1} (\mathbf{Y}_{n+1} - \mathbf{H} \hat{\mathbf{X}}_{n+1|n}) \quad (12)$$

$$\mathbf{P}_{n+1|n+1} = (\mathbf{I} - \mathbf{K}_{n+1} \mathbf{H}) \mathbf{P}_{n+1|n} \quad (13)$$

where $\mathbf{K}_{n+1}$ is the Kalman gain matrix at the $(n + 1)$ -th frame, $\mathbf{I}$ is a unit matrix, $\hat{\mathbf{X}}_{n+1|n}$ is the measured state vector containing the tracking result and $\hat{\mathbf{X}}_{n+1|n+1}$ is the filtered state vector. In this way, the

correction process corrected parameters of the filter with residual error and the tracking result at the $(n + 1)$ -th frame. Visual tracking is an inaccurate estimation process the results of which often incorporate error and drift caused by ever-changing environments and targets. Therefore, the coefficient vectors (α and β) of the noise matrix were adjusted adaptively by the feedback strategy, which will be introduced in Section 3.3.

3.2. Depth Tracking Based on Improved VR-V

As mentioned in Section 2, the original CT algorithm updates parameters of the Naïve Bayes classifier in the tracking process for each frame. As a result, if the target were fully occluded for a long time, the classifier would learn the features of the occluding object rather than the target, and the target would ultimately be lost.

Because the motion of a target is continuous, it is clear that the distance between the camera and targets would not change markedly in most of the time when there is no occlusion occurring. Similarly, the depth value of the target region would decrease significantly when occlusion occurs. Consequently, the occlusion problem of tracking algorithms can be alleviated or even handled by utilising depth images captured by RGB-D cameras. Besides, the adaptability of the entire tracking system was improved by a fusion of colour and depth image features, as the depth features of targets have some robustness to environmental disturbance.

As shown in Figure 4, a VR-V tracker was deployed to assist the CT tracker. As an updated version MeanShift, VR-V selects the best discriminative features of targets online and then locates targets with multiple MeanShift trackers [40]. As shown in Figure 6b,d, the grey histogram of the target region in a depth image describe the contour and surface features of the target roughly, so MeanShift-based VR-V algorithm are able to track targets in depth images. And the selection of tracking features provides an adaptive online learning mechanism to track target, the depth value of which is changing over time. Furthermore, the real-time performance of the entire tracking system can be ensured because VR-V is a concise algorithm with high efficiency.

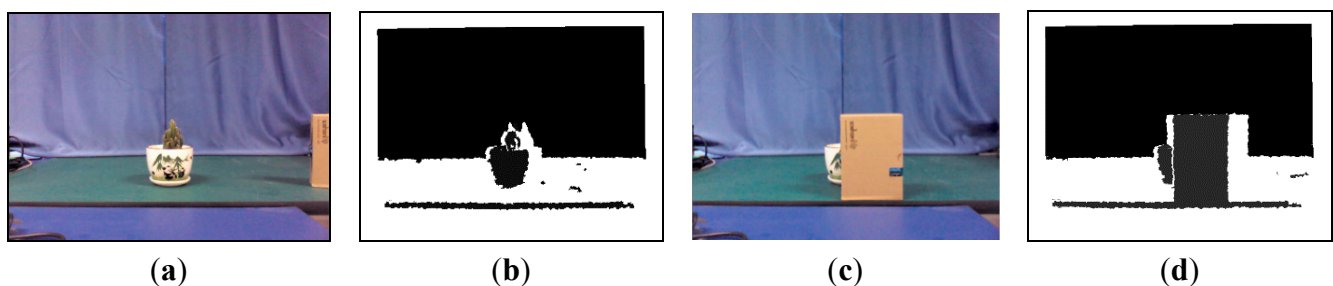


Figure 6. Colour and depth images captured by the Kinect. (a) Colour image; (b) Depth image; (c) Colour image after occlusion; and (d) Depth image after occlusion.

As shown in Figure 6a,c, the improved CT tracker cannot be aware that the target in colour images is partly or fully occluded, but the histogram-based VR-V tracker for depth images is able to detect the exception and instruct the CT tracker to temporarily stop the classifier update. To utilise this indication, a VR-V tracker with a Kalman prediction mechanism was adopted to process depth images. The prediction and correction processes of the Kalman filter were similar to the improved CT

algorithm, and the depth tracking position provided by the VR-V tracker could be used as reference information for the improved CT tracker, which will be introduced in Section 3.2.

In actual experiments, the grey pixel value of depth images ranged from 0 to 255, but the effective working distance of the Kinect was only 1.0–1.5 m, which made the image mostly dark, with most of the pixels concentrated in the 200–255 range, as shown in the left image of Figure 7. To improve the imaging quality and thereby emphasise features of the target, an enhancement process was designed for use after image acquisition. As shown in the middle and right-hand images of Figure 7, the values of pixels were flipped, and a histogram equalisation operation was performed. As the depth values of smooth surfaces and remote objects may be inaccurate, the VR-V tracker may fail in tracking if the histogram of the target in depth images is unstable. To cope with this situation, the VR-V tracker will be reset when the tracking error is too large, which will be introduced in Section 3.3.

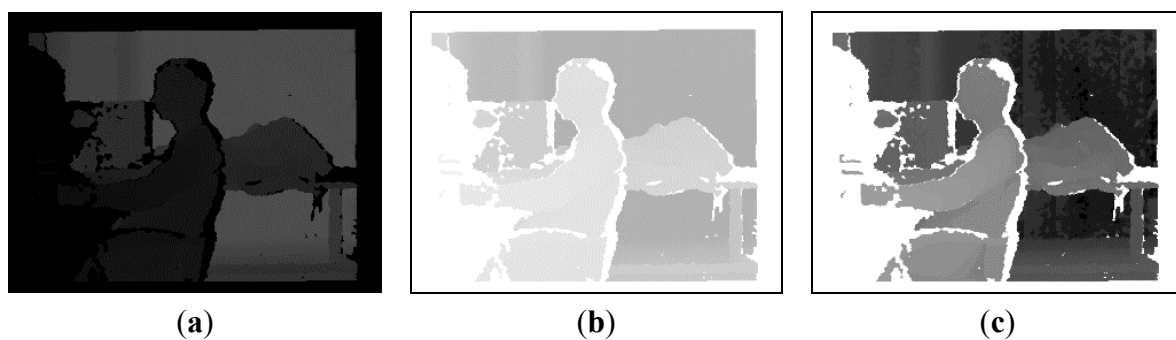


Figure 7. Diagram of depth image enhancement. (a) Before enhancement; (b) After invert; and (c) After equalization.

3.3. Feedback Strategy for Adaptive Online Learning

As mentioned in Section 2.2, a fatal defect of CT is that the parameters of the classifier are updated at a constant rate, which leads to tracking failure when the target is fully occluded. A dual tracker framework was designed to improve CT algorithm in Sections 3.1 and 3.2. And a feedback strategy inspired by expert control was introduced in this section to fuse tracking results of both trackers and realise the adaptive updating of the parameters of the CT tracker.

Table 1 shows the status transition of the feedback strategy. The strategy tried to evaluate working status of the tracking system by analysing the quality of tracking results, detect occlusion by check histogram similarity of depth images, and improve robustness of the proposed tracking algorithm with an adaptive parameter updating mechanism. In Table 1, $I_{C(n)}$ ($I_{D(n)}$) represents the tracking results at the n -th colour (depth) frame, S_{hist} represents the histogram similarity measured with Bhattacharyya coefficient between n -th and $(n + 1)$ -th depth frame, γ represents the sampling radius of the CT tracker, α_{CT}, β_{CT} represents the parameters of the Kalman filter adopted in the CT tracker and $\lambda_0, \varepsilon_0, \delta_0$ are boundary constants that control the adjustment of parameters.

Table 1. Status transition table of the feedback strategy.

No.	$I_{C(n+1)} - I_{D(n+1)}$	$I_{D(n+1)} - I_{D(n)}$	s_{hist}	Adjust Parameter	Classifier Update
1	$ I_{C(n+1)} - I_{D(n+1)} < \lambda_0$	$ I_{D(n+1)} - I_{D(n)} < \varepsilon_0$	$s_{hist} < \delta_0$	$\gamma = \gamma_0$	True
2	$\lambda_0 < I_{C(n+1)} - I_{D(n+1)} < \lambda_1$	$ I_{D(n+1)} - I_{D(n)} < \varepsilon_0$	$s_{hist} < \delta_0$	$\alpha_{CT}, \beta_{CT}, \gamma = 1.2\gamma_0$	True
3	$ I_{C(n+1)} - I_{D(n+1)} > \lambda_1$	$ I_{D(n+1)} - I_{D(n)} < \varepsilon_0$	$s_{hist} < \delta_0$	$\gamma = 1.5\gamma_0$	False
4	$ I_{C(n+1)} - I_{D(n+1)} < \lambda_0$	$ I_{D(n+1)} - I_{D(n)} > \varepsilon_0$	$s_{hist} < \delta_0$	$\gamma = 1.2\gamma_0$, Reset VR-V	True
5	$\lambda_0 < I_{C(n+1)} - I_{D(n+1)} < \lambda_1$	$ I_{D(n+1)} - I_{D(n)} > \varepsilon_0$	$s_{hist} < \delta_0$	$\alpha_{CT}, \beta_{CT}, \gamma = 1.5\gamma_0$ Reset VR-V	True
6	$ I_{C(n+1)} - I_{D(n+1)} > \lambda_1$	$ I_{D(n+1)} - I_{D(n)} > \varepsilon_0$	$s_{hist} < \delta_0$	Reinitialization	False
7	$ I_{C(n+1)} - I_{D(n+1)} < \lambda_0$	$ I_{D(n+1)} - I_{D(n)} < \varepsilon_0$	$\delta_0 < s_{hist} < \delta_1$	$\gamma = 1.2\gamma_0$	True
8	$ I_{C(n+1)} - I_{D(n+1)} < \lambda_0$	$ I_{D(n+1)} - I_{D(n)} < \varepsilon_0$	$s_{hist} > \delta_1$	$\gamma = 1.5\gamma_0$	False
9	$ I_{C(n+1)} - I_{D(n+1)} > \lambda_1$	$ I_{D(n+1)} - I_{D(n)} > \varepsilon_0$	$s_{hist} > \delta_1$	Reinitialization	False

Under condition 1, if the tracking results of both trackers are almost consistent, indicating that there is little interference factors affecting the tracking system, the CT tracker will update the classifier parameters at a constant rate. Under conditions 2 and 7, if the tracking results of the CT tracker are not so consistent or the depth histogram of the target changes in certain limits, indicating that the target may move at a fast speed or the scale of the target may be varying, the search region of the candidate patches will be enlarged and parameters of the Kalman filter will be selectively adjusted. Under conditions 3 and 8, if the tracking results of the CT tracker are inconsistent or the depth histogram of the target changes significantly, indicating a high likelihood that the environmental conditions or target model has deteriorated (partly occlusion, illumination variance or pose change), the CT tracker will stop updating and try to search the target in a much larger region. Under conditions 4 and 5, if the tracking results of the CT tracker are consistent while the tracking results of the VR-V tracker are inconsistent, which may be caused by complex motion of the target or instability of the infrared tracking subsystem, the parameters of the Kalman filter will be adjusted, the search region of the CT tracker will be enlarged and the target position of VR-V tracker will be reset to $I_{C(n+1)}$. Under conditions 6 and 9, if the tracking results of the two trackers differ considerably, the target may have left the field of view or may have been lost. Therefore, classifier updating will terminate for 10 frames and then attempt to reinitialise the valid tracking process after recovery of the target. In these cases, the feedback extends the stable margin of the CT tracker by limiting the self-updating, enlarging the search region and adaptively adjust parameters.

4. Experiment and Discussion

4.1. Experimental Environment and Tracking System Prototype

As the study was performed to design a real-time, robust and effective algorithm for our amphibious spherical robot, both robot-based and computer-based experiments were designed to test its performance in virtual and actual scenarios with occlusion, irregular motion, illumination variation and other forms of interference. Experimental tests on the proposed algorithm consisted of three parts:

- (1) In the benchmark test, the improved CT algorithm, which contained a motion estimation mechanism and ran on a computer (Intel Core i7-4712MQ, 8 GB RAM), was tested on seven standard benchmark sequences compared with the original CT algorithm and the online multiple-instance learning (MIL) algorithm [41].
- (2) In the actual test, the proposed algorithm, running on the same computer, was deployed to process colour and depth image sequences captured by a Kinect compared with MIL algorithm, RGBD baseline algorithm and RGBDOcc algorithm [38].
- (3) In the robotic test, the proposed algorithm was implemented to guide the motion of our amphibious spherical robot. As shown in Figure 8, due to limitations imposed by the size of the current version of our robot, a preliminary prototype of the robotic tracking system for the algorithm evaluation was fabricated, in which a Kinect was installed in the region of activity of the robot, where it monitored the motion of the robot while the proposed algorithm, running on the same computer, tracked a specialised guiding object and controlled the robot via the wireless local network as it followed the object.



Figure 8. Prototype of the robotic tracking system.

4.2. Experimental Results

In the benchmark and actual tests, two metrics were used to evaluate the improved CT and proposed algorithm with CT and MIL. The first metric was the success rate (SR) of the benchmark sequences. The success rate of a frame was defined as:

$$score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)} \quad (14)$$

where ROI_T is the tracked bounding box, ROI_G is the ground truth bounding box and $area(\cdot)$ denotes the number of pixels in the region. If the score is larger than the given threshold (0.5 in this paper) in a frame, it is counted as a success [32]. The second metric is the centre location error (CLE), which is the Euclidean distance between the central points of the tracked bounding box and the ground truth bounding box. Given that a tracker may be sensitive to the initial conditions or parameters of the target in the first frame, various test results of the same algorithm can be obtained with different initial bounding boxes of the target. For a fair performance evaluation, the initial position and scale size of the target were specified according to the data of ground truth bounding boxes provided by [38]. Therefore, although there was no modification of the CT and MIL programs, the experimental results in this paper may be slightly poorer than the counterparts reported previously [32,41].

Table 2 lists the major challenging factors of the test image sequences for the benchmark and actual tests. The sequences “Coke”, “Couple”, “Liquid”, “Suv”, “Tiger1”, “Tiger2” and “Fish”, taken from [42], were used to test the improved CT algorithm, which was a single combination of CT and motion estimation based on the Kalman filter (elaborated in Section 3.1). The sequences “Note”, “Book”, “Glove1” and “Glove 2”, which consisted of colour images and depth images, were recorded with a Kinect for this study and then used to evaluate the proposed algorithm (elaborated in Sections 3.1–3.3). The total frame number of all image sequences was 6512.

Table 2. Challenges of the benchmark sequences.

Sequences		Major Challenge
Standard	Coke	Fast motion, Occlusion, Background Clutters
	Couple	Fast Motion, Deformation, Background Clutters
	Liquid	Occlusion, Fast Motion, Background Clutters
	Suv	Occlusion, In-Plane Rotation, Out-of-View
	Tiger1	Illumination Variation, Fast Motion, Occlusion
	Tiger2	Illumination Variation, Fast Motion, Occlusion
	Fish	Out-of-View
Ours	Note	Fast Motion, Occlusion
	Book	Random Motion
	Glove 1	Random Motion, Occlusion
	Glove 2	Random Motion, Leave Field of View

Table 3 shows the quantitative evaluation results of the benchmark test. In the test of the improved CT algorithm, the sampling radius was set to $\gamma = 50$, the dimensionality of compressed feature vectors was set to $n = 50$ and the learning parameter was set to $\lambda = 0.85$.

Table 3. Success rate (SR) and centre location error (CLE) of the improved CT.

Sequences	Improved CT		Original CT		MILTrack		Number of Frames	
	SR	CLE	SR	CLE	SR	CLE		
Standard	Coke	19.2	31.2	15.8	36.1	9.2	28.8	291
	Couple	83.6	8.6	68.6	35.4	65.7	35.9	140
	Liquor	40.1	141.9	28.1	168.9	21.0	131.8	1741
	Suv	58.5	39.1	17.9	79.2	46.2	59.8	945
	Tiger1	49.3	23.6	46.8	28.0	8.5	104.6	354
	Tiger2	51.7	24.5	45.2	28.6	45.7	28.3	365
	Fish	9.5	37.1	3.8	38.8	81.3	10.2	476
Frames per Second		61.4		63.1		32.5		

The results indicated that the robustness to fast motion, occlusion and background clutter of the improved CT algorithm was reinforced by the motion estimation mechanism, with only a slight increase in the computational complexity of the improved CT algorithm, which did not influence its real-time performance. As shown in Figure 9a, the bounding box of CT and MIL obviously drifted when the Coke can in the image moved erratically at #200 and #225, whereas the improved CT tracker was able to track the target precisely.

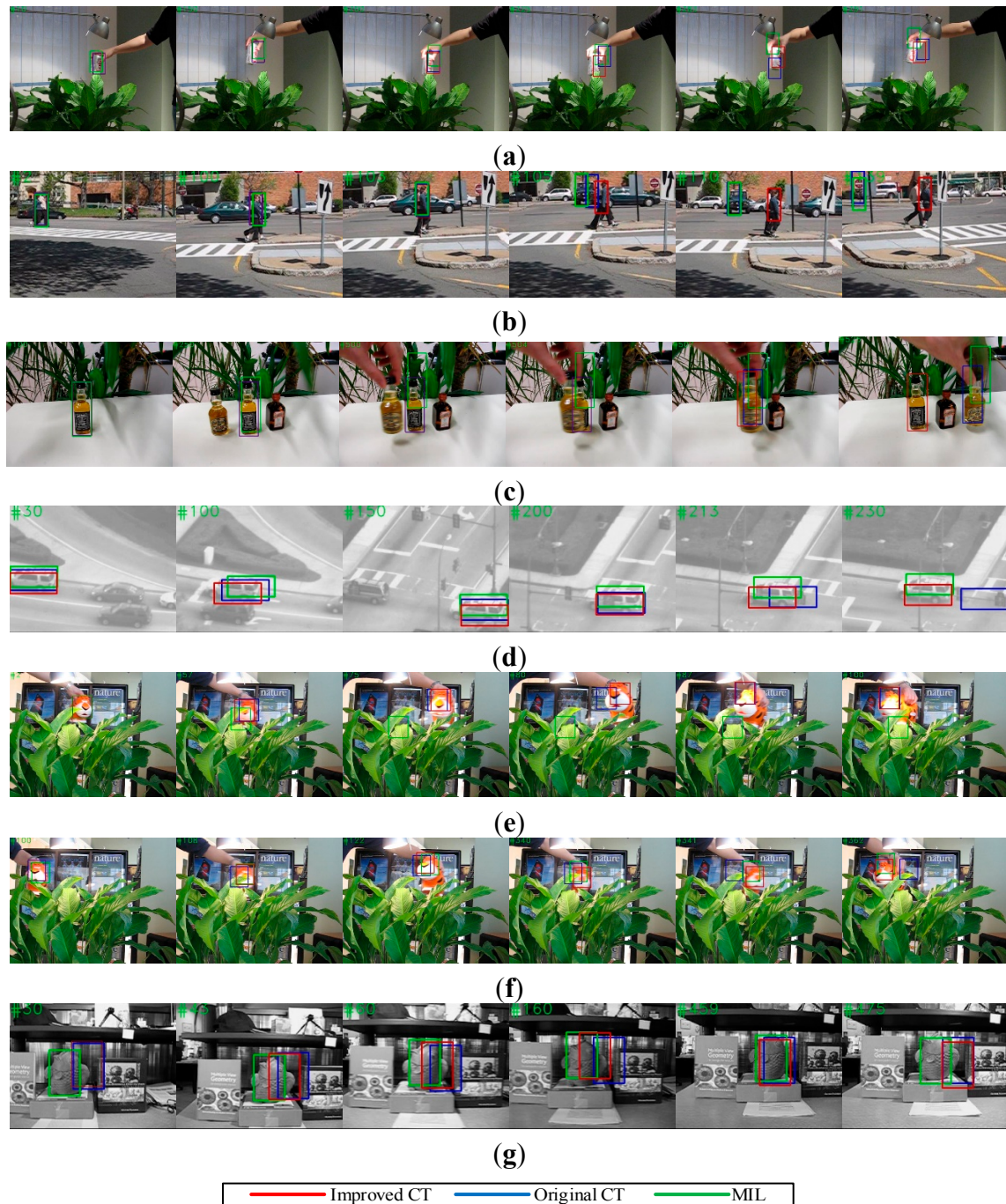


Figure 9. Tracking result on classical image sequences. (a) Tracking results of “Coke”; (b) Tracking results of “Couple”; (c) Tracking results of “Liquor”; (d) Tracking results of “Suv”; (e) Tracking results of “Tiger1”; (f) Tracking results of “Tiger2”; and (g) Tracking results of “Fish”.

As shown in Figure 9f, the CT tracker completely lost the target when occlusion occurred frequently at #341 and #362, whereas the improved CT tracker successfully detected the target when it reappeared. As shown in Figure 9b, the CT tracker and MIL tracker both missed the couple in the image when a car with similar Haar-like features appeared, but the improved CT tracker avoided this failure by predicting the motion trend of the target. Although the improved CT algorithm had a higher success rate in the benchmark test, it made little contribution to the performance improvement when

illumination variation or full occlusion occurred, as shown in Figure 9c,g. Furthermore, as shown in Figure 9e, if the target moved randomly or too rapidly, motion estimation based on a Kalman filter may not have improved and may even have diminished the effectiveness of the CT algorithm, as the Kalman filter only works well for Gaussian linear systems. This result demonstrated that a backup detection mechanism to supervise the improved CT tracker is essential.

Table 4 shows the quantitative evaluation results of the actual test. In the test of the proposed algorithm, the sampling radius was set to $\gamma = 50$, the dimensionality of compressed feature vectors was set to $n = 50$ and the learning parameter was set to $\lambda = 0.85$. The results indicated that the effectiveness and robustness of the proposed algorithm were further improved with the multi-feature fusion mechanism and the feedback strategy. Although the tracking precision and robustness of RGBDOcc were better, it seemed that the proposed algorithm is more suitable to robotic applications in consideration of the real-time performance. As shown in Figure 10a, the MIL tracker first missed the target when the target moved at a fast speed for a long time at #215, and the improved CT tracker later failed after full occlusion. However, the tracker using the proposed algorithm, which stopped updating the classifier when full occlusion occurred, was subsequently able to detect the target and reset the depth tracker successfully at #430. As shown in Figure 10b,c, the improved CT tracker and the MIL tracker failed in tracking targets that moved randomly or swerved erratically, but the tracker based on the proposed algorithm succeeded by extending the search region and adjusting parameters of trackers.

Table 4. Success rate (SR) and centre location error (CLE) of the proposed algorithm.

Sequences		Proposed Algorithm		Improved CT		MILTrack		RGBD Baseline		RGBDOcc		Number of Frames
		SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	
Ours	Note	87.2	18.3	62.1	22.6	57.2	27.2	72.1	21.7	91.1	13.9	600
	Book	88.5	9.7	36.6	55.3	46.3	47.9	76.8	17.9	86.3	9.1	600
	Glove1	85.5	12.3	43.5	29.1	40.9	63.6	69.3	41.8	88.3	8.3	600
	Glove2	79.2	17.5	47.2	23.5	41.6	42.7	59.4	49.3	75.1	16.2	400
Frames per Second		39.7		57.3		31.7		<1.0		<1.0		

As shown in Figure 10d, the tracker using the proposed algorithm was able to handle the scenario in which the target left the field of view and reappeared later by temporarily stopping online learning. The test results verified the effectiveness and robustness of the proposed algorithm and the feasibility of designing a robust tracking system with compound sensors, such as the Kinect.

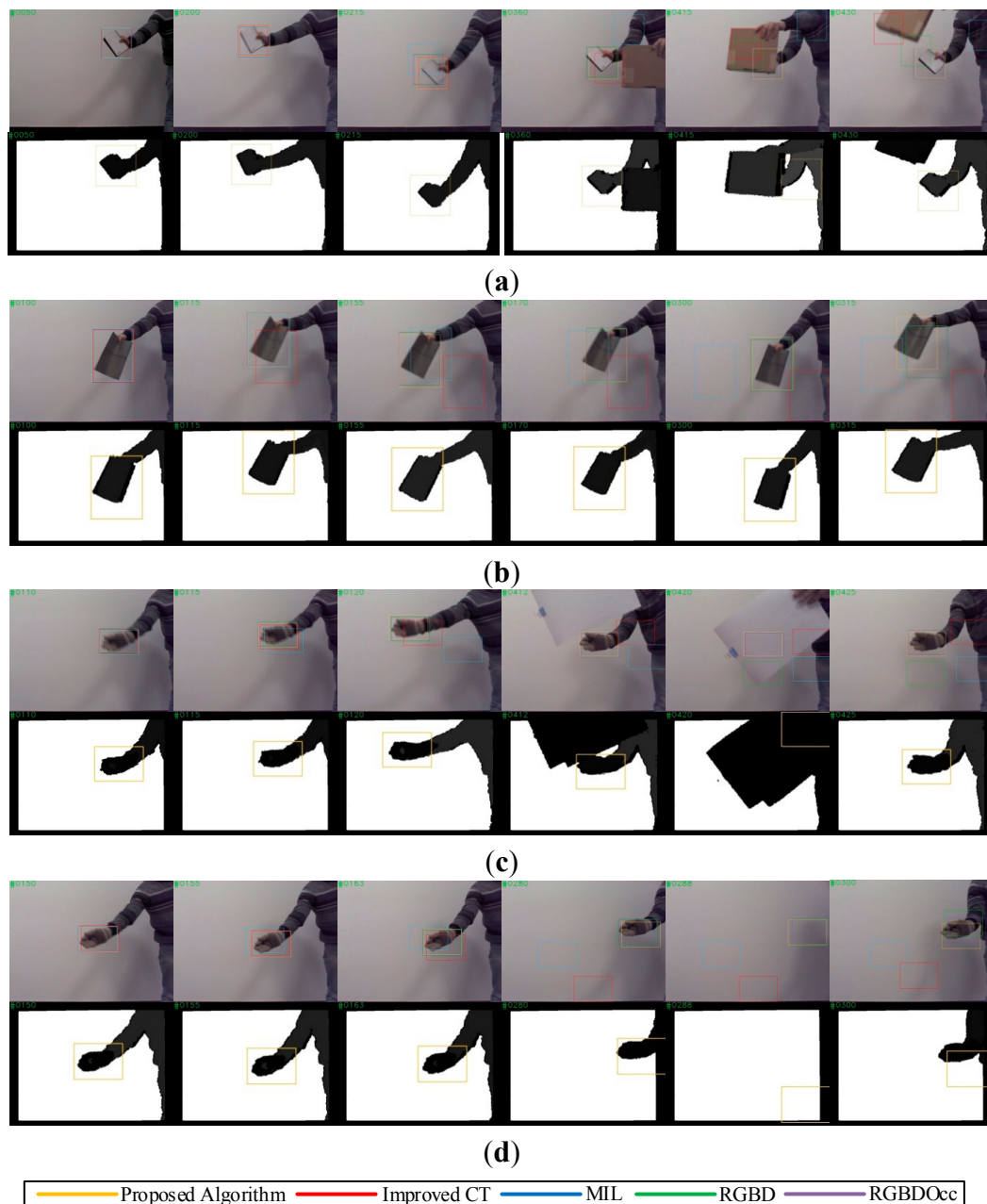


Figure 10. Tracking results of our image sequences in real scenarios. (a) Tracking results of “Note”; (b) Tracking results of “Book”; (c) Tracking results of “Glove1”; and (d) Tracking results of “Glove2”.

In the robotic test, a prototype of the robotic tracking system was built for algorithm evaluation. As it is too large to be installed in our robot, the Kinect was fixed on a frame, and monitored the motion of our spherical robot from overhead, as shown in Figure 11a, while a small orange hemisphere that moved along a black box measuring $0.8 \text{ m} \times 0.6 \text{ m}$ acted as a guiding object. As shown in Figure 8 and Figure 11b, the proposed algorithm running on the computer tracked the guiding object, calculated its trajectory and controlled the motion of our spherical robot via the wireless local area network. When the guiding object arrived at the position $(0.6, 0.4)$, it was fully occluded by a book for several seconds. Figure 11c shows the tracking results (red lines) and the trajectory of the robot (green lines). There was only slight deviation of less than 5 cm or 3 pixels between the tracking result and the

designated trajectory, and the controller module of the robot was set to omit small deviations or noise. As a result, except for drift occurring at bends in the designated trajectory, the robot was able to follow the motion of the guiding object almost exactly with the guidance of the tracking system. Even the target was fully occluded, the tracking system detected the target after slight drift when the target reappeared.

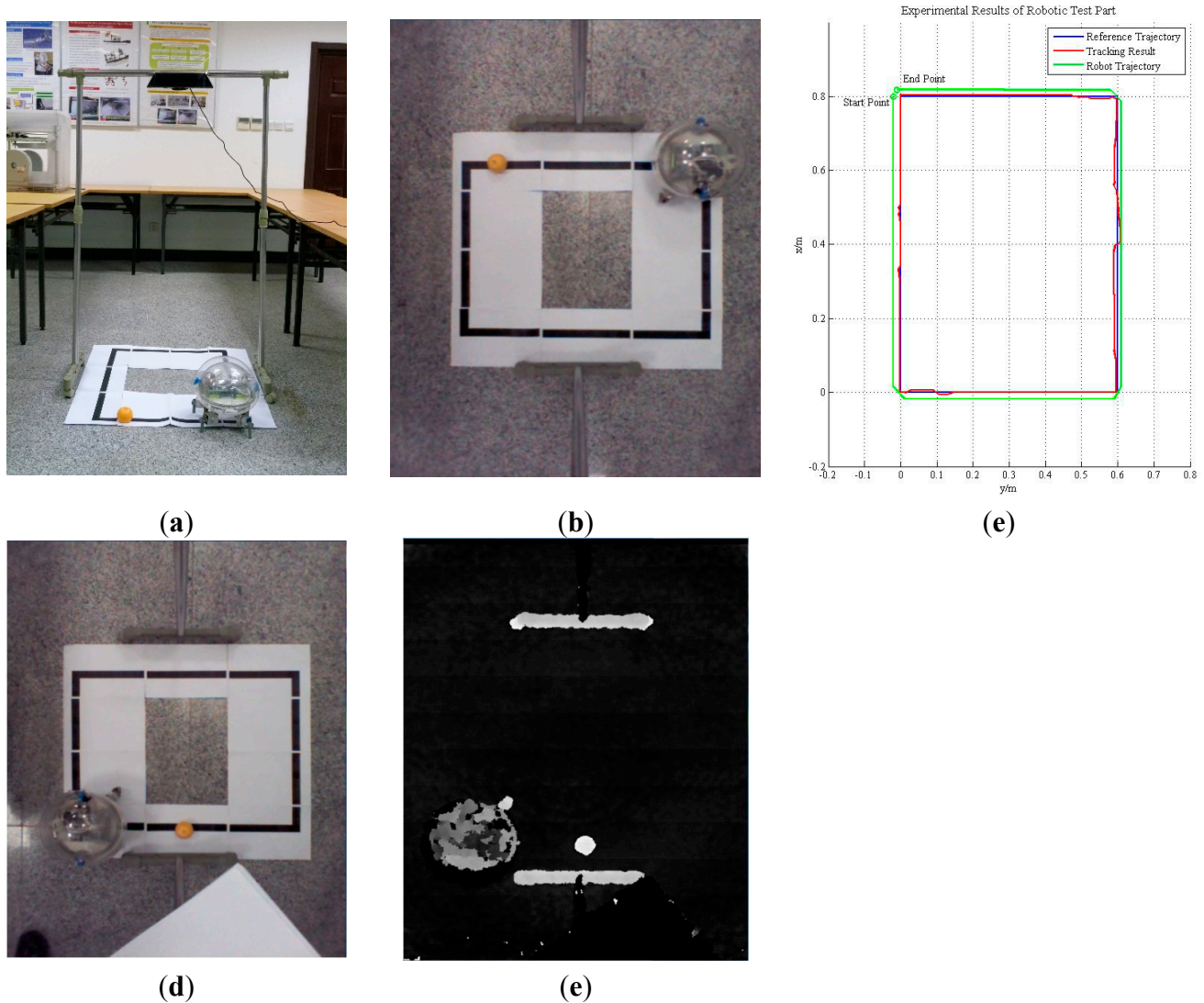


Figure 11. Robotic tracking experiment with our amphibious spherical robot. (a) Experimental scenario; (b) Perspective of Kinect; (c) Experimental Results; (d) Colour Image before Occlusion; and (e) Depth Image before Occlusion.

5. Conclusions

In this paper, a real-time visual tracking algorithm was proposed and implemented for our amphibious spherical robot. To meet the requirements of robots in robustness, adaptability and effectiveness of tracking systems, Microsoft Kinect was adopted to capture colour and depth images. An improved CT tracker with a Kalman prediction mechanism was deployed to process colour images from the Kinect, and a VR-V tracker was implemented to process depth images from the Kinect. The update rate of the CT tracker was adjusted using a feedback strategy, which partly solved the problem of its ineffectiveness in situations of full occlusion, irregular target motion and illumination variation.

The experimental results of virtual, actual and robotic tests verified the effectiveness, efficiency and robustness of the proposed algorithm.

This paper presents only a preliminary prototype of the robotic tracking system. Kinect cannot be installed in our spherical robot due to size limitations. Also, Kinect is not able to provide depth images of high quality in outdoor and underwater environment because it acquire depth information with an infrared speckle dot pattern projector and an infrared camera [33]. Furthermore, the mechanisms of parameter adjustment and adaptive classifier updating in the feedback strategy were experience-based and primitive in some measures. Future work will focus on realisation of a small Kinect-like compound image sensor which consists of a colour camera and a laser-based depth camera for applications in amphibious environments [43].

Acknowledgments

This work was supported by the Excellent Young Scholars Research Fund of Beijing Institute of Technology and the Basic Research Fund of the Beijing Institute of Technology (No.3160012211405). This research project was also partly supported by National Natural Science Foundation of China (61375094), Key Research Program of the Natural Science Foundation of Tian-jin (13JCZDJC26200) and National High Tech. Research and Development Program of China (No.2015AA043202). Ping Guo and Yanlin He fabricated the amphibious spherical robot used in this paper and helped to conduct the experiments.

Author Contributions

Shaowu Pan has developed the algorithm, conducted the experiments and drafted the paper. Liwei Shi has contributed significantly to data analysis, manuscript preparation, and revising it critically for important intellectual detail. Shuxiang Guo has contributed to the conception of the study and helped perform the analysis with constructive discussions.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Fabian, J.; Young, T.; Peyton Jones, J.C.; Clayton, G.M. Integrating the microsoft kinect with simulink: Real-time object tracking example. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 249–257.
2. Capi, G.; Toda, H.; Nagasaki, T. A vision based robot navigation and human tracking for social robotics. In Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE), Phoenix, AZ, USA, 15–16 October 2010; pp. 1–6.
3. Wirbel, E.; Steux, B.; Bonnabel, S.; de la Fortelle, A. Humanoid robot navigation: From a visual SLAM to a visual compass. In Proceedings of the 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), Evry, France, 10–12 April 2013; pp. 678–683.

4. Bischoff, B.; Duy, N.-T.; Streichert, F.; Ewert, M.; Knoll, A. Fusing vision and odometry for accurate indoor robot localization. In Proceedings of the 12th International Conference on Control Automation Robotics & Vision (ICARCV), Guangzhou, China, 5–7 December 2012; pp. 347–352.
5. Lin, R.; Li, M.; Sun, L. Image features-based mobile robot visual SLAM. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013; pp. 2499–2504.
6. Siradjuddin, I.; Behera, L.; McGinnity, T.M.; Coleman, S. A position based visual tracking system for a 7 DOF robot manipulator using a Kinect camera. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012; pp. 1–7.
7. Wang, P.; Su, J.; Li, W.; Qiao, H. Adaptive visual tracking based on discriminative feature selection for mobile robot. In Proceedings of the IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Hong Kong, China, 4–7 June 2014; pp. 55–61.
8. Oonishi, S.; Yanou, A.; Minami, M. Human-face-tracking using visual servoing by patient robot. In Proceedings of the SICE Annual Conference (SICE), Nagoya, Japan, 14–17 September 2013; pp. 2266–2271.
9. Gupta, M.; Behera, L.; Subramanian, V.K.; Jamshidi, M.M. A Robust visual human detection approach with UKF-based motion tracking for a mobile robot. *IEEE Syst. J.* **2014**, *PP*, 1–13.
10. Liu, Q.; Zhao, X.; Hou, Z. Survey of single-target visual tracking methods based on online learning. *IEEE Trans. Comput. Vis.* **2014**, *8*, 419–428.
11. Salti, S.; Cavallaro, A.; Di Stefano, L. Adaptive appearance modeling for video tracking: Survey and evaluation. *IEEE Trans. Image Process.* **2012**, *21*, 4334–4348.
12. Liu, Y.F.; Li, Q.; Fang, H.; Xu, H.C. Research on embedded system with implementation of a moving object tracking algorithm based on improved meanshift on DM6437. *Adv. Mater. Res.* **2014**, *1003*, 207–210.
13. Zhang, M.; Xin, M.; Yang, J. Adaptive multi-cue based particle swarm optimization guided particle filter tracking in infrared videos. *Neurocomputing* **2013**, *122*, 163–171.
14. Ross, D.A.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141.
15. Liu, J.; Subramanian, K.R.; Yoo, T.S. An optical flow approach to tracking colonoscopy video. *Comput. Med. Imag. Gr.* **2013**, *37*, 207–223.
16. Pirzada, N.; Nayan, M.Y.; Hassan, F.S.M.F.; Khan, M.A. Device-free localization technique for indoor detection and tracking of human body: A survey. *Proced. Soc. Behav. Sci.* **2014**, *129*, 422–429.
17. Avidan, S. Support vector T racking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1064–1072.
18. Chen, X.; Wu, J. Scalable compressive tracking based on motion. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, 12–14 December 2013; pp. 504–509.
19. Qi, Y.; Suzuki, K.; Wu, H.; Chen, Q. EK-means tracker: A pixel-wise tracking algorithm using kinect. In Proceedings of the Third Chinese Conference on Intelligent Visual Surveillance (IVS), Beijing, China, 1–2 December 2011; pp. 77–80.

20. Shi, Q.; Ishii, H.; Sugahara, Y.; Sugita, H.; Takanishi, A.; Huang, Q.; Fukuda, T. Design and control of a biomimetic robotic rat for interaction with laboratory rats. *IEEE/ASME Trans. Mechatron.* **2014**, *PP*, 1–11.
21. Shi, Q.; Ishii, H.; Kinoshita, S.; Konno, S.; Takanishi, A.; Okabayashi, S.; Iida, N.; Kimura, H.; Shibata, S. Modulation of rat behaviour by using a rat-like robot. *Bioinspir. Biomim.* **2013**, *8*, 46002–46012.
22. Wang, K.; Liu, Y.; Li, L. Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement. *IEEE Trans. Robot.* **2014**, *30*, 1026–1035.
23. Wang, K.; Liu, Y.; Li, L. Visual servoing based trajectory tracking of underactuated water surface robots without direct position measurement. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, IL, USA, 14–18 September 2014; pp. 767–772.
24. Guo, S.; Mao, S.; Shi, L.; Li, M. Development of an Amphibious Mother Spherical Robot Used as the Carrier for Underwater Microrobots. In Proceedings of the International Conference on Complex Medical Engineering (ICME), Kobe, Japan, 1–4 July 2012; pp. 758–762.
25. Shi, L.; Guo, S.; Li, M.; Yue, C.; Mao, S.; Asaka, K. Development of an amphibious turtle-inspired spherical mother robot. *J. Bion. Eng.* **2013**, *10*, 446–455.
26. Guo, S.; Shi, L.; Xiao, N.; Asaka, K. A Biomimetic underwater microrobot with multifunctional locomotion. *Robot. Auton. Syst.* **2012**, *60*, 1472–1483.
27. Pan, S.; Guo, S.; Shi, L.; He, Y.; Wang, Z.; Huang, Q. A Spherical Robot based on all Programmable SoC and 3-D Printing. In Proceedings of the IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 150–155.
28. Wang, H.-S.; Chen, W.-D.; Wang, Z.-L. Visual tracking of robots in uncalibrated environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 2959–2964.
29. Zhang, K.; Zhang, L.; Yang, M.-H. Real-Time Compressive Tracking. In Proceedings of the European Conference on Computer Vision (ECCV 2012), Florence, Italy, 7–13 October 2012; pp. 864–877.
30. Xu, R.; Gu, X.; Wei, B. An improved real time compressive tracking. In Proceedings of the 2nd IAPR Asian Conference on Pattern Recognition (ACPR), Okinawa, Japan, 5–8 November 2013; pp. 692–696.
31. Yan, Q.-S.; Li, L.-S. Kernel sparse tracking with compressive sensing. *IET Comput. Vis.* **2014**, *8*, 305–315.
32. Zhang, K.; Zhang, L.; Yang, M.-H. Fast compressive tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 2002–2015.
33. Han, J.; Shao, L.; Xu, D.; Shotton, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Trans. Cybern.* **2013**, *43*, 1318–1334.
34. Smisek, J.; Jancosek, M.; Pajdla, T. 3-D with kinect. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Barcelona, Spain, 6–13 November 2011; pp. 1154–1160.

35. Tang, S.; Wang, X.; Lv, X.; Han, T.-X.; Keller, J.; He, Z.; Skubic, Z.; Lao, S. Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor. In Proceedings of the Asian Conference on Computer Vision (ACCV), Daejeon, Korea, 5–9 November 2012; pp. 525–538.
36. Wang, P.; Ma, S.; Shen, Y. Performance study of feature descriptors for human detection on depth map. *Int. J. Model. Simul. Sci. Comput.* **2014**, *5*, 13–29.
37. Spinello, L.; Arras, K. People detection in RGB-D data. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 3838–3843.
38. Song, S.; Xiao, J. Tracking revisited using RGB-D camera: Unified benchmark and baselines. In Proceedings of the IEEE International Conference on Computer Vision (ICCV2013), Sydney, Australia, 1–8 December 2013; pp. 233–240.
39. Ren, H.-L.; Liu, W.; Lim, A. Marker-based surgical instrument tracking using dual kinect sensors. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 921–924.
40. Collin, R.T.; Liu, Y.; Leordeanu, M. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1631–1643.
41. Babenko, B.; Yang, M.-H.; Belongie, S. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632.
42. Wu, Y.; Lim, Jongwoo; Yang, M.-H. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
43. Prats, M.; Fernandez, J.J.; Sanz, P.J. Combining template tracking and laser peak detection for 3d reconstruction and grasping in underwater environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Algarve, Portugal, 7–12 October 2012; pp. 106–112.