applied sciences

MDPI

Article

Combination of Spatial and Frequency Domains for Floating Object Detection on Complex Water Surfaces

Xin Sun, Hao Deng, Guihua Liu * and Xin Deng

School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China; sunxin@mails.swust.edu.cn (X.S.); denghao@mails.swust.edu.cn (H.D.); dengxin@mails.swust.edu.cn (X.D.)
* Correspondence: liuguihua@swust.edu.cn

Received: 27 October 2019; Accepted: 28 November 2019; Published: 30 November 2019

Featured Application: We proposed a method of floating object detection, which is characterized by small sample training, strong anti-interference ability, low time consumption and high precision. The method combines spatial-based texture detection and frequency-based saliency detection. Our research is based on the natural images collected by an unmanned surface cleaning robot, which is one of the important applications in water source conservation.

Abstract: In order to address the problems of various interference factors and small sample acquisition in surface floating object detection, an object detection algorithm combining spatial and frequency domains is proposed. Firstly, a rough texture detection is performed in a spatial domain. A Fused Histogram of Oriented Gradient (FHOG) is combined with a Gray Level Co-occurrence Matrix (GLCM) to describe global and local information of floating objects, and sliding windows are classified by Support Vector Machines (SVM) with new texture features. Then, a novel frequency-based saliency detection method used in complex scenes is proposed. It adopts global and local low-rank decompositions to remove redundant regions caused by multiple interferences and retain floating objects. The final detection result is obtained by a strategy of combining bounding boxes from different processing domains. Experimental results show that the overall performance of the proposed method is superior to other popular methods, including traditional image segmentation, saliency detection, hand-crafted texture detection, and Convolutional Neural Network Based (CNN-based) object detection. The proposed method is characterized by small sample training and strong anti-interference ability in complex water scenes like ripple, reflection, and uneven illumination. The average precision of the proposed is 97.2%, with only 0.504 seconds of time consumption.

Keywords: floating object detection; complex water surface; combination; spatial-frequency; texture; saliency detection

1. Introduction

Salvage of floating objects is an important part of water source conservation. At present, interception and manual salvage are the main methods for the salvage task [1]. These methods are labor consuming and dangerous. Although manipulating robots overcomes the shortcomings noted above [2,3], it requires skilled operators and cannot operate around the clock. With the development of image processing and computer vision, the use of autonomous salvage technology in surface cleaning robots has attracted great interest from researchers, and one of its most important elements is the fast and robust detection of floating objects. However, small sample acquisition is a tough problem due to the diversity and rarity of floating objects in different water areas. Additionally, interference factors such as ripple, reflection, and uneven illumination also make it difficult to find a general model for various complex water scenarios.

Few researchers have focused on floating object detection with respect to complex water surfaces. In general, the existing methods that can be applied to the detection task fall into four categories: traditional image segmentation, saliency detection, hand-crafted feature detection, and object detection based on a Convolutional Neural Network (CNN). According to different processing domains, these methods can also be classified into spatial-based methods and frequency-based methods.

In spatial domain, traditional image segmentation has an excellent real-time performance [4]. Wang et al. [5] extracted contours of floating objects according to image grayscale, then framed the contours that met a special size criterion. Xue et al. [6] used the Super-pixel Merging method to segment eligible foreground regions. Tang et al. [7] used Mean-shift clustering and an improved Otsu [8] method to detect floating objects. Although these methods work well in calm water, they are not robust enough when ripples or reflections exist. To this end, Jin et al. [9] proposed an improved Gaussian Mixture Model Based (GMM-based) automatic segmentation method (IGASM) to detect water surface floats, but it is not applicable in static images or in images with severe interferences.

Spatial-based object detection methods can be applied to floating object detection and have a certain anti-interference ability [10,11]. Traditional methods of combining hand-crafted features with classifiers are simple and fast [12]. Hand-crafted texture features with grayscale invariance, e.g., Histogram of Oriented Gradient (HOG) [13], Local Binary Pattern (LBP) [14], and Gray Level Co-occurrence Matrix (GLCM) [15], are suitable for describing floating objects under uneven illumination. But these features do not utilize global information, making feature descriptors inaccurate when other sever interferences exist. Therefore, it is necessary to extract very suitable features with respect to the special detection task, i.e., floating object detection in complex water scenes. Some algorithms based on CNN do not need to manually extract features and perform well in speed and accuracy, e.g., a Region-based Fully Convolutional Network (R-FCN) [16], Single Shot Multi-Box Detector (SSD) [17], Single Shot Refinement Detector (RefineDet) [18], and the You Only Look Once (YOLOv3) method [19]. However, in the case of a small sample, CNN-based methods are prone to over-fitting due to their multi-layer convolutions. To this end, saliency detection methods based on visual attention mechanism can solve the problem of small sample. But spatial-based saliency detection methods, e.g., Global Contrast-based Saliency Detection (RC) [20], Contrast-based Filtering for Saliency Detection (SF) [21] and Boolean Map for Saliency Detection (BMS) [22] are generally sensitive to parameter selection, which limits their generalization in multiple scenarios.

In the frequency domain, frequency-based saliency detection methods, such as the Spectral Residual (SR) approach [23], Frequency-tuned (FT) method [24], and Maximum Symmetric Surround (MSS) method [25] have fewer parameters and less time consumption than those saliency detection methods in a spatial domain. However, in the case of large ripple, reflection, and other interference factors, redundancies of salient maps are difficult to eliminate, resulting in extreme high false alarm rates.

In order to overcome the problems of small sample acquisition and several interference factors, i.e., ripple, reflection, and uneven illumination in floating objects detection, a simple but effective method combining spatial and frequency domains is proposed. In the spatial domain, after training with few positive samples, sliding image blocks are roughly classified by Support Vector Machines (SVM) [26–28] with a Fused Histogram of Oriented Gradient (FHOG) features [29] and GLCM features to detect floating objects. In the frequency domain, Gaussian high-pass filtering and phase spectrum reconstruction are implemented to preliminarily suppress the interferences. Then, global and local low-rank decompositions [30,31] are carried out to remove the residual redundant regions. Finally, bounding boxes from different processing domains are combined by analyzing Intersection-over-Unions (IoUs) to frame real objects accurately.

The main contributions of this paper are as follows:

An effective combination of hand-crafted texture features, i.e., FHOG and GLCM. The new combined features describe floating objects more accurately than traditional HOG, GLCM, and LBP features in complex water scenes, while its time consumption of feature extraction is comparable to the traditional ones.

A new anti-interference saliency detection method. The method processes a frequency spectrum and adopts low-rank matrix decomposition to accurately detect floating objects without any training samples. Note that the method effectively solves the problems caused by ripple, reflection, and uneven illumination, and performs better than other sample-free saliency detection methods.

A novel floating object detection algorithm combining spatial-based texture detection and frequency-based saliency detection. The combined algorithm is characterized by small sample training, low time consumption, strong anti-interference ability, and high precision.

The rest of this paper is organized as follows. Section 2 introduces the methodology, including the proposed algorithm framework, detailed implementations in different processing domains, and the strategy of combination. Section 3 presents and analyzes the comprehensive results. Parameter selections and comparative experiments are also conducted in Section 3. Section 4 draws the conclusion.

2. Methodology

All images applied are collected by a surface cleaning robot. One of the key tasks of the robot is to detect floating objects in different water areas. Algorithm 1 is the framework of our floating object detection algorithm for each video frame.

Algorithm 1 Floating Object Detection

Input: I
Output: B_1, B_2, B_3
1: $B_1 \leftarrow$ *TextureDetection*(I)
2: $B_2 \leftarrow$ *SaliencyDetection*(I)
3: $B_3 \leftarrow$ *Combination*(B_1, B_2)
4: return B_1, B_2, B_3

As shown in Algorithm 1, the input is a video frame I, and the detection task is carried out in three parts: texture detection, saliency detection, and combination. The outputs are three vectors of bounding boxes, i.e., B_1, B_2, and B_3, which are the results of the three parts, respectively. Note that the final output B_3 can be an empty vector, indicating that there are no floating objects in this video frame. More details are given in the subsections below.

2.1. Texture Detection in Spatial Domain

Currently, the state-of-the-art methods of universal object detection are usually based on the data-driven CNN. Taking the commonly used 16-depth network by Visual Geometry Group (VGG16-Net) as an example, the convolutional neural network has about 138 million parameters, so thousands of training samples are required to solve the over-fitting problem. However, due to the diversity and rarity of floating objects on different water surfaces, only hundreds or even dozens of effective source images can be collected, which makes it difficult to robustly detect floating objects with CNN-based methods. Although there are a series of Transfer Learning methods [32–34] and Data Augmentation strategies [35,36] that help CNN models fit methods without plenty of source images, these training skills cannot essentially improve the detection precision and maximize the performance of CNN. Therefore, the popular CNN-based methods may not be the best choices of floating object detection in the case of small sample acquisition.

To solve the problem of over-fitting caused by insufficient training samples, we adopted Latent SVMs as the classifiers instead of convolutional neural networks. Unlike neural networks, Latent SVMs adopt the optimization goal of minimizing structural risk rather than empirical risk, and use the principle of maximum margin to reduce the requirements of data size and data distribution [26–28]. Therefore, small sample training with SVM is able to complete a rough detection task. More importantly, SVM is insensitive to the problem of sample imbalance when all background windows are treated as

negative samples. Experiments shown in Section 3 will prove that SVM is effective enough for the rough texture detection task, and more details of SVM training are provided in the later experiment section.

In order to ensure that our classifier has a high hit rate (while high false alarm rate is acceptable at this stage), it is necessary to extract very reliable features which helps to clearly distinguish foregrounds and backgrounds. To this end, the combination of FHOG and GLCM features is adopted. The new texture features have good grayscale invariance and rotation invariance, which effectively describe global and local information of floating objects in complex water scenes.

2.1.1. FHOG Feature Extraction

The traditional HOG method firstly divides the image into many cells (e.g., size 8×8 pixels), and then calculates the histogram of nine unsigned gradient directions for each cell [13]. However, this method considers two regions separated at 180 degrees to be identical, which is not accurate to describe floating objects with 360-degree rotation directions. To guarantee the ability of feature description, the HOG method extracts texture features by sliding image blocks with a small stride. Therefore, the dimension of final feature descriptor is very high, which leads to high time complexity and over-fitting, especially in the case of small sample training.

As shown in Figure 1, an improved HOG named FHOG is adopted. First, the nine cells in a sliding image block are divided into four groups. In each group, an 18-dimensional signed HOG eigenvector and a 9-dimensional unsigned HOG eigenvector are combined into a 4×27 matrix. Then, columns of the matrix are added up to get a 1×27 vector, while rows of the matrix are added up to get a 4×1 vector. Finally, through connecting the two vectors, a 31-dimensional FHOG eigenvector (describing a block with nine cells) is obtained. Note that when describing the same block with nine cells, traditional HOG method needs an 81-dimensional descriptor, which is 50 dimensions larger than that of FHOG. More importantly, FHOG not only improves the speed of feature extraction, but also has a better feature description ability.

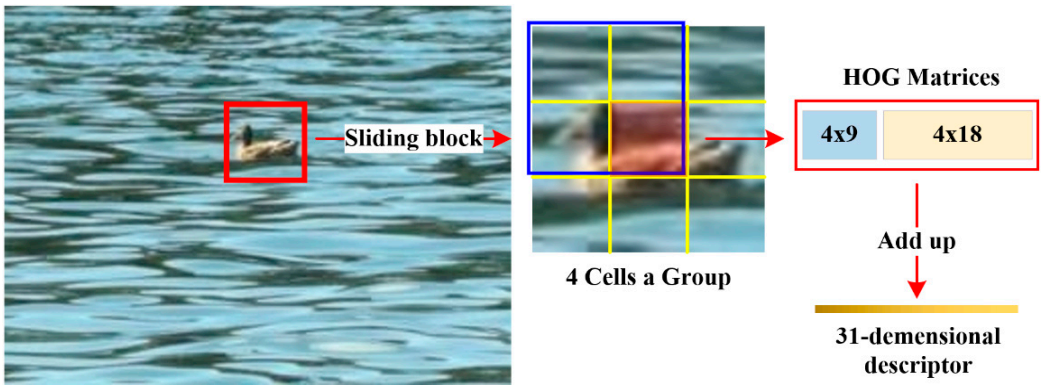


Figure 1. Fused Histogram of Oriented Gradient (FHOG) for feature extraction.

2.1.2. GLCM Feature Extraction

To further improve the ability of feature description, GLCM features are extracted. An HOG-like feature is the statistical result of pixels at different gray levels, while GLCM describes grayscale relationships between pixels and their adjacent pixels [15]. GLCM provides information about the directions and the extents of grayscale changes, but it cannot provide the information in the form of descriptors. Therefore, statistical attributes are used to quantitatively describe texture characteristics on the basis of the GLCM matrix.

Based on the priori knowledge that textures of image blocks containing floating objects change dramatically, while textures of other background blocks change relatively gently, four statistical attributes, i.e., Energy E_g, Entropy E_g, Homogeneity H_o, and Contrast C_t are calculated. The definition is Equation (1):

$$\begin{cases} E_g = \sum_i \sum_j P^2(i,j) \\ E_p = -\sum_i \sum_j P(i,j) \log P(i,j) \\ H_o = \sum_i \sum_j \frac{P(i,j)}{1+P(i,j)^2} \\ C_t = \sum_i \sum_j (i-j)^2 P(i,j) \end{cases}, \quad (1)$$

where P is the probability matrix obtained by normalizing the GLCM matrix. Energy, Contrast, and Homogeneity reflect the thickness, depth, and variability of the texture, respectively, and Entropy measures the amount of information in an image. At the same time, textures of the foreground image blocks with floating objects are more delicate and more dramatic than the background blocks only with water surfaces. Therefore, the energy, uniformity, and contrast of the foregrounds are small, while the entropy values are large. Through these four attributes, foreground, and background areas can be well distinguished. Our calculation step is 2, and the directions are 0 degrees, 45 degrees, 90 degrees, and 135 degrees. Finally, a 16-dimensional descriptor can be extracted from each sliding window. Experimental results (in Section 3) show that the combination of GLCM features and FHOG features is more conducive to detecting floating objects than traditional texture features.

2.2. Saliency Detection in Frequency Domain

A novel frequency-based saliency detection method for floating objects in complex scenes is proposed. Different from the mainstream saliency methods such as BMS [22], the Multi-task Sparsity Pursuit method (MBP) [37], RC [20], MSS [25], SR [23], and FT [24], our method can better solve the problems caused by waves, reflections, and uneven illumination. Algorithm 2 is the framework of our saliency detection algorithm for each video frame.

Algorithm 2 Saliency Detection

Input: I
Output: B_2
Stage 1: Initial Saliency Map
$G \leftarrow GammaCorrection(I)$
$H \leftarrow GaussianHighpassFitering(G)$
$P \leftarrow PhaseReconstruction(H)$
Stage 2: Redundancy Removal
$P_g \leftarrow MatrixDecomposition(P)$
$N_1, N_2, \ldots, N_t \leftarrow CreateBlocks(P)$
for $i = 1 \rightarrow k$ **do**
  $M_i \leftarrow MatrixDecomposition(N_i)$
  $S_i \leftarrow ComputeNorm(M_i)$
  $N_i \leftarrow Supression(N_i, S_i)$
end for
$P_l \leftarrow Merge(N_1, N_2, \ldots, N_k)$
Stage 3: Ultimate Saliency Map
$P_f \leftarrow \gamma Pg + (1-\gamma)P_l$
$B_2 \leftarrow BoundBox(P_f)$
return B_2

In saliency detection, as shown in Algorithm 2, the processing steps fall into three stages. Aiming at obtaining an initial saliency map with obvious foreground and background distinctions, the first stage includes gamma correction, Gaussian high-pass filtering, and phase spectrum reconstruction. The second stage aims to remove the scattered redundancies in the initial saliency map, and obtain the global and local saliency maps. At this stage, splitting and merging image blocks are needed for local

low-rank decompositions. The third stage is to obtain the ultimate saliency map by fusing global and local saliency maps, and output the bounding box vectors as the final results.

2.2.1. Initial Saliency Map

Gamma correction (i.e., power operation on each pixel) is performed to reduce the impacts of uneven illumination, while Gaussian high-pass filtering is to suppress the ripples and reflections to some extent. The transfer function of Gaussian high-pass filter is:

$$H(u,v) = 1 - \exp(\frac{-D^2(u,v)}{2{D_0}^2}), \tag{2}$$

where $H(u, v)$ is the transfer function of the filter, $D(u, v)$ is the distance from point (u, v) to the origin of the Fourier transform center, and D_0 is the cut-off frequency of the filter.

Amplitude spectrum of the image mainly contains the information of brightness, and phase spectrum mainly contains the information of texture structure [30]. Therefore, a phase spectrum reconstruction is performed to reduce the impacts caused by ripples and uneven illumination. Then, an initial saliency map is obtained.

2.2.2. Redundancy Removal

Although most of the redundant regions have been obviously suppressed by the operations above, the scattered redundancies still need to be removed. Based on Chandrasekaran's matrix decomposition theory [31] that an input matrix can be decomposed into a sparse part and a low-rank part, an image can be represented as the following two parts:

$$I(x,y) = L(x,y) + M(x,y), \tag{3}$$

where $L(x, y)$ is the low-rank part corresponding to the background, and $M(x, y)$ is the sparse noises corresponding to the salient regions. The decomposition is implemented by Equation (4):

$$\begin{cases} \min\limits_{L,M} rank(L) + \|M\|_0 \\ s.t.\ P = L + M \end{cases}, \tag{4}$$

where P is the input matrix. This is a Non-deterministic Polynomial Hard (NP-Hard) problem, so Equation (4) is optimized by Equation (5):

$$\begin{cases} \min\limits_{L,M} \|L\|_* + \varepsilon\|M\|_1 \\ s.t.\ P = L + M \end{cases}, \tag{5}$$

where ε is a balance coefficient, and we set ε as 0.01. When the input matrix P is an initial saliency map, a global saliency map $P_g = L$ is obtained. However, global decomposition cannot remove all of the redundancies. To this end, local decomposition is performed as follows.

First, initial saliency map is divided into k blocks of the same size (i.e., N_1, N_2, N_k). Next, grayscale features from all the blocks are extracted independently and combined into a matrix Y, as shown in Equation (6):

$$Y = [y_1, y_2, \ldots, y_k] \tag{6}$$

where y_i is the column vector obtained by averaging each row of pixels in the i^{th} image block N_i. Then, Y is decomposed by Equation (5), and the saliency level S of N_i is obtained by Equation (7):

$$S(N_i) = norm(Y(:,i)) = \sqrt{\sum (Y(:,k))^2}. \tag{7}$$

We have difference strategies to process the blocks according to S, which is a key step of obtaining the local saliency map:

$$\begin{cases} \mu_1 N_i & S(N_i) \leq \sigma_1 \\ \mu_1 N_i & \sigma_1 < S(N_i) \leq \sigma_2 \\ N_i & others \end{cases}, \tag{8}$$

where σ_1 and σ_2 are the thresholds, while μ_1 and μ_2 are the suppression coefficients. Experiment results show that the best thresholds are 0.3 and 0.7, and the suppression coefficients are 0.2 and 0.8. Note that when $S_i > \sigma_2$, the block N_i will not be processed.

2.2.3. Ultimate Saliency Map

We fuse the global and local saliency maps in a simple way:

$$P_f = \gamma P_g + (1 - \gamma) P_l, \tag{9}$$

where γ is the weight of the global saliency map, P_f is the ultimate saliency map, i.e., the result of saliency detection.

Figure 2 intuitively shows the entire process. In Figure 2a, huge and abrupt reflections (at the top and middle of the image) make floating objects (at the bottom of the image) inconspicuous. Figure 2b shows that gamma correction handles the illumination problem well. In Figure 2c, most of the non-salient parts are obviously suppressed by Gaussian high-pass filtering. Figure 2d shows that phase spectrum reconstruction makes salient regions relatively concentrated and makes redundant parts dispersed. Figure 2e,f shows the binary images of global saliency map, local saliency map, and ultimate saliency map, respectively. Note that the method of binarization are all Otsu. In Figure 2e,f, global or local decomposition can remove most of the redundant parts, but it is not able to remove all redundancies. However, as shown in Figure 2g,h, through a fusing operation, the final result is much closer to the ground truth.

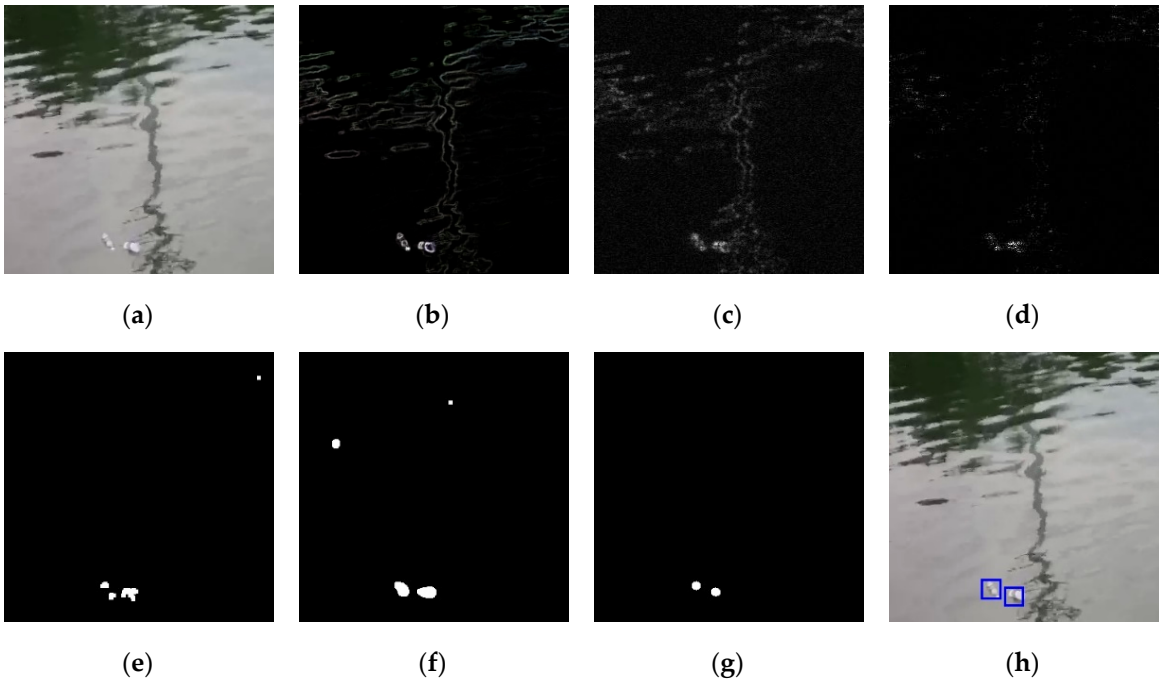
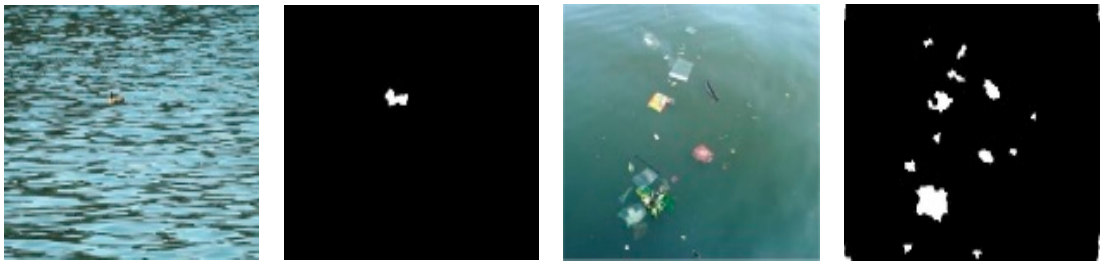


Figure 2. Each stage of saliency detection. They are (**a**) the source image, (**b**) Gamma corrected image, (**c**) Gaussian high-pass filtered image, (**d**) phase reconstructed image, (**e**) global saliency map, (**f**) local saliency map, (**g**) ultimate saliency map, (**h**) result of our saliency detection.

Since it is completely independent of training samples, our saliency detection method can be applied to detect any floating objects. As shown in Figure 3, even if the floating object in the scene is indistinguishable to the naked eye, or the scene has many targets, our method can successfully detect different unknown floating objects. In Figure 3a,b, the low contrast duck on the turbulent surface was detected. In Figure 3c,d, multiple floating litters of different sizes on the clam surface were detected. Note that the smallest object detected in Figure 3d is approximately 10x10 pixels in size, indicating that our saliency detection method has potential in small object detection.



(**a**) (**b**) (**c**) (**d**)

Figure 3. Saliency detection in two different water scenes. They are (**a**) the turbulent surface with a low contrast object, (**b**) the detection result of (**a**), (**c**) the clam surface with multiple floating objects of different sizes, (**d**) the detection result of (**c**).

2.3. Combination

Multiple candidate regions can be obtained by the above methods in spatial and frequency domains. However, these candidate regions may intersect or overlap with each other. So confirmation is needed to get the final bounding boxes, i.e., B_3. First, Intersection-over-Unions (IoUs) of the bounding boxes from spatial and frequency domains are calculated by Equation (10):

$$IoU = \frac{Area(A \cap B)}{Area(A \cup B)}, \tag{10}$$

where A and B are two bounding boxes from texture detection and saliency detection, respectively, and they constitute a pair of combinations. Note that in a certain video frame, there are $M \times N$ similar combinations. M and N are the number of bounding boxes from texture detection and saliency detection, respectively. The final result from the candidates is obtained by an Intersection-over-Unions Based (IoU-based) strategy, as shown in Equation (11):

$$C = \begin{cases} \text{A}, & \tau_1 < IoU \leq \tau_2 \cap R_1 > R_2 \\ \text{B}, & \tau_1 < IoU \leq \tau_2 \cap R_1 \leq R_2 \\ A \cap B, & IoU > \tau_2 \\ \varnothing, & others \end{cases} \tag{11}$$

where C is the result of combining texture detection and saliency detection; τ_1 and τ_2 are the thresholds; R_1 and R_2 are replacement rates. Experiments results show that the best thresholds are 0.2 and 0.8, respectively. The replacement rate is similar to the confidence of a bounding box, which is calculated through Equation 12:

$$\begin{cases} R_1 = \frac{Area(A,B)}{A} \\ R_2 = \frac{Area(A,B)}{B} \end{cases} \tag{12}$$

where R_1 and R_2 are the replacement rates of the results of texture detection and saliency detection, respectively. The combination strategy not only strictly judges the existence of floating objects, but also makes the final detection result close to the ground truth.

3. Experiments

In this section, the detection results in spatial domain, frequency domain and spatial-frequency domain are firstly shown intuitively. Then, through a series of ablation experiments, the optimal parameters are given. Finally, we compare our final results with other floating object detection methods. All experiments are conducted in an unmanned surface cleaning robot shown in Figure 4. The robot is composed of an image processing unit, a navigation unit, a conveyor caterpillar, etc. The visual system consists of two HIKVISION B12-IPOE surveillance cameras, and the CPU of the image processing unit is Intel (R) Xeon (R) E5-2660 2.20 GHz. Note that no GPU is involved in running the program. The programming libraries used are OpenCV 4.0.1 and Eigen 3.3.7.

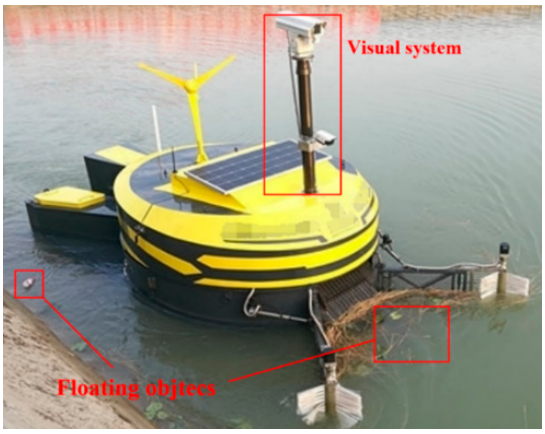


Figure 4. Experimental platform: unmanned surface cleaning robot.

Experimental images were sampled from 10 different water surfaces, and each area has 40 images resized into 350 × 350 pixels. Interference factors in these samples include different illumination intensities, waveform shapes, turbulence levels, reflection sizes, etc. Of these 400 source images, 280 were for training and 120 for testing, and all of them have appropriate ground truth labels. Note that in the testing set consisting of 120 source images, 140 distinct floating objects are labeled as positive samples, because some images contain multiple floating objects. Additionally, there are 10 images (1 image each water scene) without any floating objects.

3.1. Intuitive Results of Texture Detection

To intuitively compare the effects of our feature with other mainstream features (LBP [14], HOG [13], GLCM [15], FHOG [29]) in the task of floating object detection, a comparative experiment is conducted. The experiment occurred under seven challenging water scenes, including large waves, circular ripples, spray, turbulent water surface, strip ripples, near-view reflection, and far-view reflection. In this comparative experiment, the feature extraction method was the only variable, so it was necessary to keep sample generation and classification methods consistent.

Sample generation. The positive samples are not directly cropped from ground truth labels, because one of the most important features of our classification was is the dramatic change of texture features. Instead, positive samples were generated by cropping special images blocks from the 280 source images. These image blocks had the same centers of the corresponding ground truth labels, while their sizes are all 48 × 48 pixels. From each source image, negative samples were generated by cropping sliding image blocks (48 × 48 pixels in size and 20 pixels in stride) that did not intersect with any ground truth labels.

Classification. As mentioned in the Section 2.1, Latent SVMs are adopted as our classifiers for the small sample problem. The uniformed sample size (i.e., 48 × 48 pixels) guaranteed the same feature dimensions, which as beneficial for SVM training. In addition, the cell sizes of HOG, FHOG and LBP were all 16 × 16 pixels, and the calculation stride of GLCM was 2 pixels. More details of SVM training

(built by the Application Programming Interface (API) of opencv4.0.1) are given: the SVM type was C-Support Vector Classification (C-SVC) [27], kernel was Linear [28], and the termination criteria was 1000 iterations or the accuracy reached $1 + 1.192092896^{-7}$.

The detection strategy was to classify multi-scale sliding windows in the testing set. We adopted three scaling ratios: 0.8, 1, and 1.2. The initial window size was 48×48 pixels and the strides were all 10 pixels. Figure 5 shows the results.

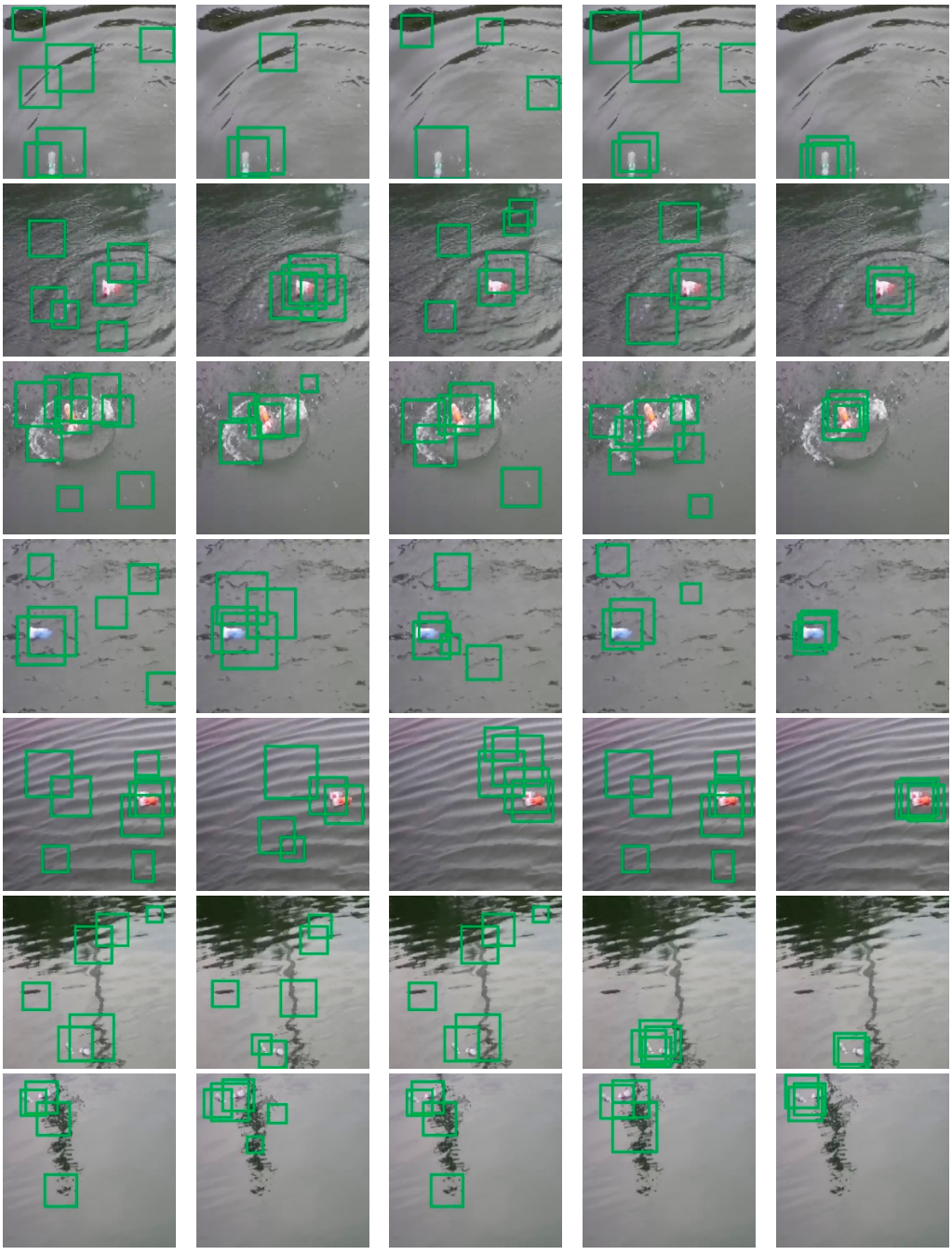


Figure 5. Results of five texture detection methods. They are (**a**) Histogram of Oriented Gradient (HOG), (**b**) Fused Histogram of Oriented Gradient (FHOG), (**c**) Gray Level Co-occurrence Matrix (GLCM), (**d**) Local Binary Pattern (LBP), (**e**) ours (FHOG-GLCM). From top to bottom, the scenes are large waves, circular ripples, spray, turbulent water surface, strip ripples, near-view reflection, and far-view reflection.

From Figure 5a,b, it can be seen that FHOG features make the result more accurate than HOG features. Figure 5b–d show that the three methods (FHOG, GLCM, and LBP) have plenty of false detections, but the results of FHOG and GLCM are better than those of LBP, indicating that the two features we selected are effective. Figure 5e shows that our combined features make the detection task in complex scenes more accurate, and greatly reduce false detections. Inevitably, our texture detection results are larger than their ground truths, as there were too few training samples to set small sliding windows. However, floating objects can be correctly detected, and the problem of texture detection will be effectively solved by saliency detection and combination.

3.2. Intuitive Results of Saliency Detection

To demonstrate the effectiveness of our frequency-based saliency detection method, experiments with existing sample-free saliency detection methods are carried out. The results are shown in 5 challenging surface scenes, including huge reflections with big waves, striped ripples, slender reflections, huge circular waves under uneven illumination, and far-view reflections. Figure 6 shows the results.

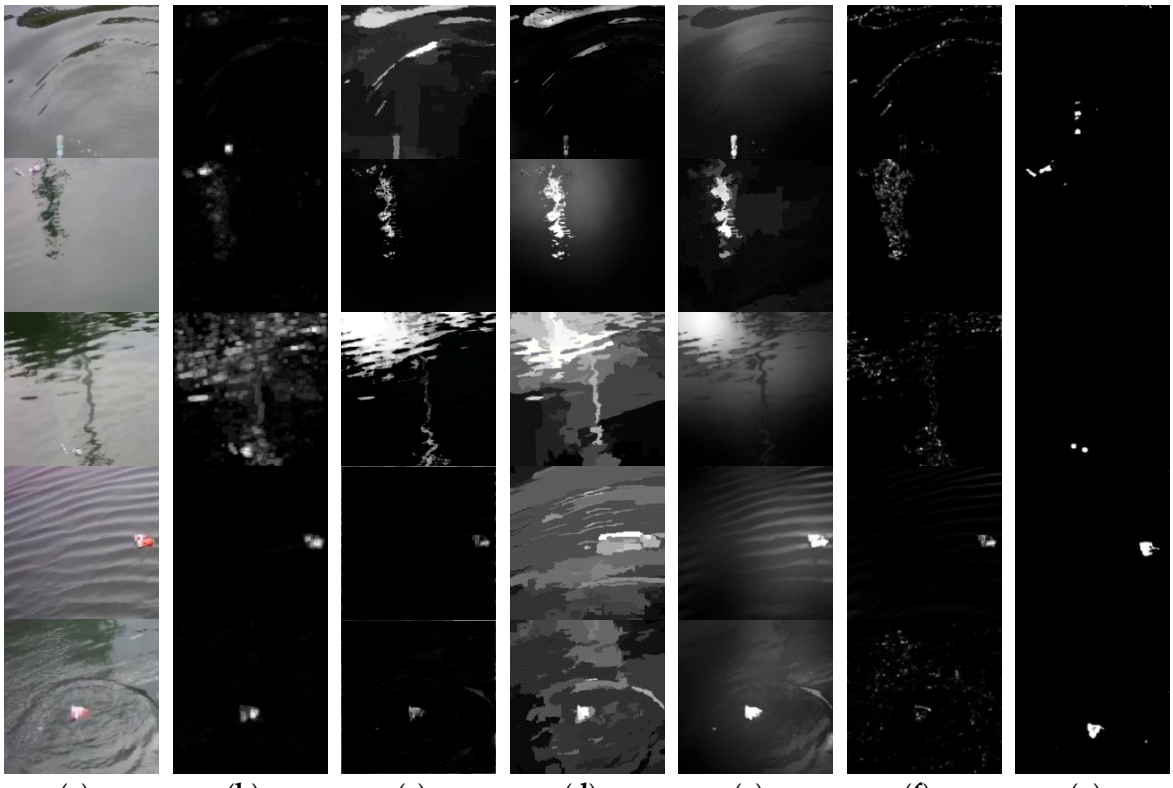


Figure 6. Saliency detection results in complex surface scenes (from top to bottom, they are large ripples, far-view reflections, near-view reflections, strip ripples, circular ripples, respectively). The methods are (**a**) original image, (**b**) result of Boolean Map for Saliency Detection (BMS), (**c**) result of Frequency-tuned method (FT), (**d**) result of Global Contrast-based Saliency Detection (RC), (**e**) result of Contrast-based Filtering for Saliency Detection (SF), (**f**) result of Spectral Residual approach (SR) and (**g**) ours.

Figure 6b–f show the results of five popular sample-free saliency detection methods. In Figure 6c,d, large ripples are mistaken for floating objects. In Figure 6b,e, near-view reflections cause plenty of false detections. Though some ripples are removed by the frequency-based method SR, shown in Figure 6f, important information of floating objects is removed as well. From these intuitive results, a preliminary conclusion can be drawn that the detection results of our method shown in Figure 6g are optimal in the task of floating object detection with respect to various interference factors.

3.3. Intuitive Results of Combination

The results shown in the above subsections indicate that our spatial-based texture detection and frequency-based saliency detection can both obtain location information of floating objects. However, bounding boxes from texture detection are larger than the corresponding ground truths (due to small sample training), and the results of salient detection are smaller than the real sizes (due to the loss of contour information). Through combining bounding boxes from the two methods, closer results to the ground truths can be obtained. To demonstrate the effectiveness of the combination intuitively, results in five challenging scenes are shown in Figure 7.

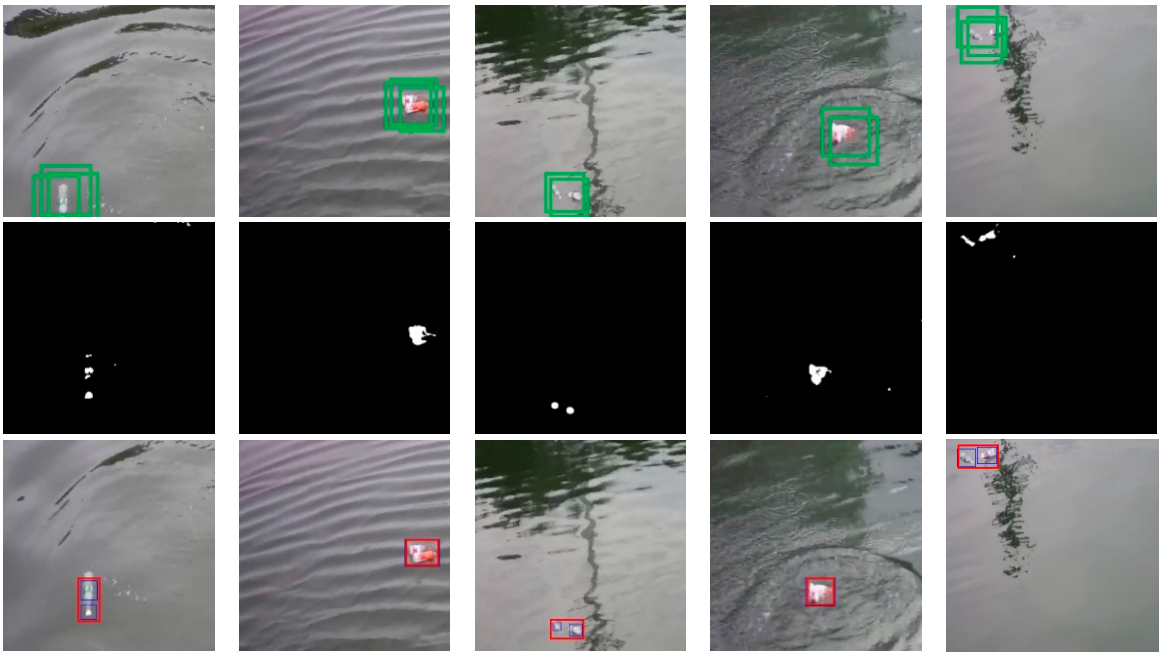


Figure 7. Detection results in 5 challenging scenes: (**a**) large ripples, (**b**) strip ripples, (**c**) near-view reflections, (**d**) circular ripples, (**e**) far-view reflections. From top to bottom, the results are from: spatial-based texture detection, frequency-based saliency detection, and combination.

In Figure 7, the first line shows that our spatial-based texture detection is able to detect floating objects correctly, though their bounding boxes are larger than the ground truths. The second line shows that correct saliency maps with no redundancies are obtained by our frequency-based saliency detection, but some contours are removed (especially in Figure 7a,c). The third line shows that the results of combination have the correct size and location information of floating objects, indicating that our combination strategy overcomes the shortcomings in different processing domains. Meanwhile, as shown in Figure 7a–e, our method is robust in the five challenging scenes, and the final results are close to the ground truths.

3.4. Evaluation Metrics

To evaluate our results quantitatively, four popular evaluation metrics of object detection are calculated, i.e., Precision P, Recall R, F_1-Measure F_1, and average detection time T. Precision and Recall are often a pair of contradiction measures. Generally speaking, when the Precision is high, the Recall tends to be low. Therefore, the comprehensive evaluation metric F_1-Measure is needed. Higher F_1

value indicates better performance of the detection model. The calculation methods of P, R, and F_1 are as follows:

$$\begin{cases} P = \frac{TP}{TP + FP} \\ R = \frac{TP}{TP + FN} = \frac{TP}{G} \\ F_1 = \frac{2P \cdot R}{P+R} \end{cases}, \tag{13}$$

where TP and TN denote the number of true positives and true negatives, respectively, while FP and FN denote the number of false positives and false negatives, respectively. G is the number of ground truths, equal to the number of all positives. In our testing set consisting of 120 source images, the value of G is 140 rather than 120, because some testing images contain multiple floating objects.

In our experiments, a result was considered a true positive only when the IoU calculated from the bounding box and its ground truth was higher than 0.8. This threshold guarantees the practicalities of the testing methods.

3.5. Parameters Selection

In the task of texture detection, FHOG feature can be extracted with different cell sizes, so it is necessary to find the optimal cell size through ablation experiments.

Table 1 shows the results of an ablation experiment to find the best cell size in feature extraction. As shown in Table 1, texture detection with the cell size of 16×16 pixels has the highest precision and recall rate (88.24% and 96.4%, respectively), although they are close to the results when the cell size is 32×32 pixels. However, in terms of computational complexity, when the cell size is 16×16 pixels, the detection time is the shortest (0.333 seconds). This detection time is 0.108 seconds faster than the time when the cell size is 32×32 pixels. We concluded that bigger cells perform better in precision than the smaller ones until the size has reached 16×16, while the time consumption decreases as cell size increases except the size 32×32. Therefore, size 16×16 is the best choice.

Table 1. Comparison of results of texture detection for different cell sizes.

Cell size	Time (s)	P (%)	R (%)
4×4	0.698	81.1	96.1
8×8	0.502	84.2	95.9
16×16	**0.333**	**88.24**	**96.4**
32×32	0.441	88.5	96.5

In the task of saliency detection, ablation experiments are performed for the two tricky problems:

(1) what is the best weight of global saliency map in the fusion operation;
(2) how to solve the problem of high time consumption caused by matrix decomposition.

For the first problem, we made a comparison of the results of saliency detection for different global-map weights. As shown in Figure 8, Precision peaks at the weight value of 0.4, and the Recall remains high at this value. Although the time consumption is very small when the weight is 1 or 0, its Precision and Recall are extremely low. Consequently, the optimal weight value of global saliency map is 0.4.

To solve the problem of time consuming in matrix decomposition, initial saliency maps are down-sampled before global decompositions, as smaller matrices make decompositions faster. Note that the initial saliency maps are not necessary to resize for local decompositions. We make a comparison of the results for different map sizes.

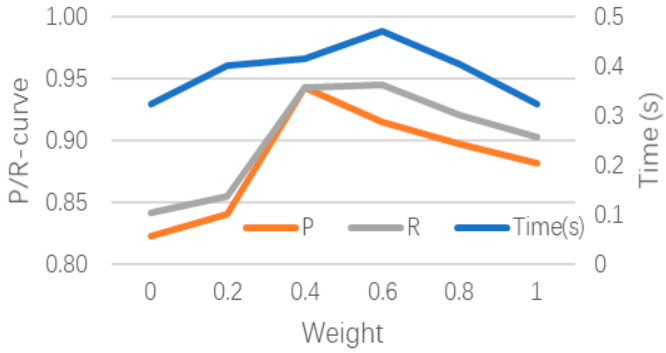


Figure 8. Comparison of results of saliency detection for different weights.

As shown in Table 2, the smaller size for global decompositions not only makes the whole detection task have less time consumption, but also makes it have higher precision and recall rates. When the size is 48 × 48, all of the three parameters are optimal values, i.e., the highest precision rate (94.29%) and recall rate (94.29%), and the lowest time consumption (0.415 seconds). Note that the detection time of size 48 × 48 is 18.335 seconds faster than that of size 350 × 350, and the precision and recall rate are respectively 9.59% and 15.19% higher than those of size 350 × 350. The possible reason is that when an image is down-sampled, the calculated data is reduced, and some noises and redundant regions are removed as well.

Table 2. Comparison of the results for different sizes of the initial saliency map.

Size	Time (s)	P (%)	R (%)
48 × 48	0.415	94.29	94.29
96 × 96	1.701	91.2	92.7
150 × 150	9.12	88.1	83.8
192 × 192	10.68	92.5	88.5
350 × 350	18.75	84.7	79.1

3.6. Comparative Experiments

Comparative Experiments are carried out for the comprehensive evaluation. Among the existing methods in spatial domain, Wang [5], Jang [38], Tang [7] are traditional image segmentation methods; HOG [13], FHOG [29], GLCM [15], and LBP [14] are texture detection methods; RC [20], SF [21], BMS [22] are saliency detection methods; SSD300 [17], the You Only Look Once method (YOLOv3) [19], R-FCN [16], RefineDet321 [18] are popular object detection methods based on CNN. Due to the particularity of the research, only SR [23], FT [24], and MSS [25] are the existing saliency detection methods in the frequency domain.

The experiments have the same configuration as below:

(1) Otsu Method [8] for binarizations;
(2) cell size 16x16, block size 48x48 and classifier SVM for texture detections;
(3) backbone VGG16 and pre-training weight VGG16-VOC2007 [35] for CNNs.

Table 3 shows the comparison of the results of existing methods.

In Table 3, not only is the final result of our method (i.e., row 20) listed, but so is the result of spatial-based texture detection and the result of frequency-based saliency detection (i.e., row 8 and row 19, respectively). Unless otherwise specified, the result of our method refers to the final result. As shown in Table 3, recalls of traditional image segmentation methods (i.e., rows 1–3) are lower than all the others because of interference factors. Texture detection methods (i.e., rows 4–8) have little time consumption, but their recalls are relatively low. It can be shown that our texture detection method (i.e., row 8) performs better than other texture-based methods, indicating that FHOG-GLCM is an

effective hand-crafted texture feature. Frequency-based saliency detection methods (i.e., rows 16–19) are faster than the spatial-based ones (i.e., rows 13–15), and our saliency detection method (i.e., row 19) has the highest precision and recall rates among them. Although the popular CNN-based methods (i.e., rows 9–12) are close to ours in terms of Recall, our final method has higher precision with less time consumption.

Table 3. Comparison of the results of existing methods.

Row	Method	T(s)	True Positive (TP)	False Negative (FP)	P(%)	R(%)
1	Wang	0.277	95	36	72.52	67.86
2	Jang	0.545	105	34	75.54	75.00
3	Tang	0.794	119	20	85.61	85.00
4	Histogram of Oriented Gradient (HOG)	0.337	126	33	79.25	90.00
5	Fused Histogram of Oriented Gradient (FHOG)	0.316	130	23	84.97	92.86
6	Gray Level Co-occurrence Matrix (GLCM)	0.295	129	22	85.43	92.14
7	Local Binary Pattern (LBP)	0.321	128	38	77.11	91.43
8	**Ours-S**	0.333	135	18	**88.24**	**96.43**
9	Single Shot Multi-Box Detector (SSD300)	0.712	133	8	94.33	95.00
10	You Only Look Once (YOLOv3)	0.697	134	11	92.41	95.71
11	Region-based Fully Convolutional Network (R-FCN)	1.351	132	7	94.96	94.29
12	Single Shot Refinement Detector (RefineDet321)	0.987	135	5	96.43	96.43
13	Global Contrast-based Saliency Detection (RC)	0.694	125	40	75.76	89.29
14	Contrast-based Filtering for Saliency Detection (SF)	0.612	117	48	70.91	83.57
15	Boolean Map for Saliency Detection (BMS)	0.797	131	20	86.75	93.57
16	Spectral Residual (SR)	0.215	125	36	77.64	89.29
17	Frequency-tuned method (FT)	0.253	126	37	77.30	90.00
18	Maximum Symmetric Surround (MSS)	0.390	131	24	84.52	93.57
19	**Ours-F**	0.415	132	8	**94.29**	**94.29**
20	**Ours-Final**	0.504	135	4	**97.12**	**96.43**

We calculated the size information of the objects detected by our final method. Statistical results show that in our testing set, when the accuracy is IoU >= 0.8, the largest object detected is 56 × 70 pixels and the smallest object is 27 × 31 pixels. To compensate for the inadequacy of the comparison of the above Precision and Recall (P-R) results, F1-Measures and time consumptions of all methods are intuitively shown in Figure 9.

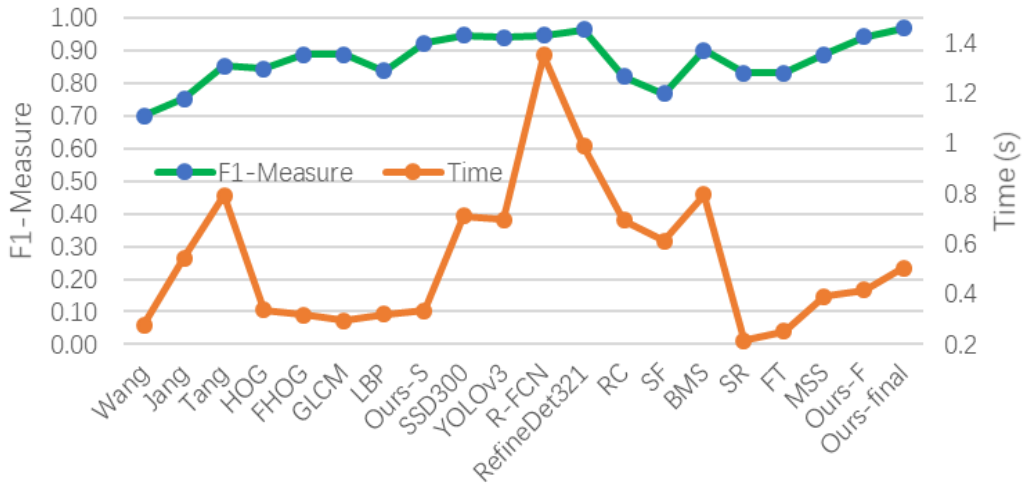


Figure 9. F-curve and time consumption curve of all methods.

As shown in Figure 9, the F_1-Measure of our final method (at around 0.95) is close to those of CNN-based methods, i.e., SSD300, YOLOv3, R-FCN, and RefineDet321. Meanwhile, the F_1-Measure of our final method is higher than those of traditional image segmentation methods, saliency detection methods, and other texture detection methods. In addition, time consumption of our method (at

around 0.5 seconds) is much smaller than those of CNN-based methods. All the experimental results indicate that the overall performance of this method is the best one.

4. Conclusions

Aiming at the problem of floating object detection for the surface cleaning robot in the case of small sample acquisition and interferences factors from waves, reflections, and uneven illumination, an effective detection method is proposed. The method combines texture detection in spatial domain and saliency detection in frequency domain.

In texture detection, a useful combination of hand-crafted texture features is implemented. Compared with 4 traditional texture detection methods, precision of our texture detection is 88.24%, recall is 96.43% (IoU >= 0.8), F_1-Measure is 0.92, and average detection time is 0.333 seconds, which demonstrates that our texture feature (FHOG-GLCM) describes floating objects effectively and efficiently.

In saliency detection, the proposed method combines frequency-domain processing and low-rank decomposition. Compared with six traditional saliency detection methods, the precision of our saliency detection is 94.29%, the recall is 94.29% (IoU >= 0.8), the F_1-Measure is 0.89, and the average detection time is 0.415 seconds, which demonstrates that our saliency detection method performs better than other saliency detection methods in complex water scenes.

Compared with 17 existing methods including traditional image segmentation, saliency detection, hand-crafted feature classification, and CNN-based object detection, our final method has the highest precision (97.12%) and the highest recall (96.43%) with little time consumption (0.504 seconds). The results show the effectiveness of combining spatial and frequency domains, and the reliability and rapidity of our method of floating object detection for surface cleaning robots.

Author Contributions: X.S. designed the algorithm, performed the experiments and wrote the paper. G.L., X.D., and H.D. modified the paper. G.L. supervised the research.

Funding: This work was supported by National Natural Science Foundation of China (Grant No.11602292. 61701421, 61601381) and Postgraduate Innovation Fund Project by Southwest University of Science and Technology (Grant No.19ycx0112).

Acknowledgments: The authors are grateful for the experimental platform and resources provided by the Sichuan Province Key Laboratory of Special Environmental Robotics.

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

References

1. Zhang, D.; Wang, D.; Ren, Y.; Zhou, S. Interception effect of riparian vegetation zone on floats and factors affecting it. *J. Beijing For. Univ.* **2015**, *37*, 98–103.
2. He, Y.; Zhang, X.; Zhu, L.; Sun, G.; Qiao, J. Design and implementation of the visual detection system for amphibious robots. *Int. J. Robot. Autom.* **2019**, *34*, 417–430. [CrossRef]
3. Cao, X.; Sun, C. Multi-AUV cooperative target hunting based on improved potential field in underwater environment. *Appl. Sci.* **2018**, *8*, 973–982.
4. Gan, Z. Research and application of binarization algorithm of QR code image under complex illumination. *Appl. Opt.* **2018**, *39*, 667–673.
5. Wang, M.; Zhou, S.D. Static water object detection and segmentation. *Res. Explore. Lab.* **2010**, *29*, 51–54.
6. Xue, P. Foreground and background segmentation based on super pixel level feature representation. *J. XI'AN Univ.* **2015**, *10*, 731–735.
7. Wei, T.; Si-Yang, L.; Han, G.; Qian, T. A target detection algorithm for surface cleaning robot based on machine vision. *Sci. Tech. Eng.* **2019**, *19*, 136–141.
8. Ostu, N. A threshold selection method from gray-histogram. *IEEE Tran. Syst. Man Cybern.* **2007**, *9*, 62–66.
9. Jin, X.; Niu, P.; Liu, L. A GMM-based segmentation method for the detection of water surface floats. *IEEE Access* **2019**, *7*, 119018–119025. [CrossRef]

10. Becker, D.; Cain, S. Improved space object detection using short-exposure image data with daylight background. *Appl. Opt.* **2018**, *57*, 3968. [CrossRef]
11. Zhang, Z.; Li, D.; Liu, S.; Xiao, B.; Cao, X. Multi-View Ground-Based Cloud Recognition by Transferring Deep Visual Information. *Appl. Sci.* **2018**, *8*, 748. [CrossRef]
12. Ojeda, J.; Nieves, J.L.; Romero, J. How daylight influences high-order chromatic descriptors in natural images. *Appl. Opt.* **2017**, *56*, 3968. [CrossRef] [PubMed]
13. Navneet, D.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 8–10 June 2015; pp. 886–893.
14. Alhindi, T.J.; Kalra, S.; Ng, K.H.; Afrin, A.; Tizhoosh, H.R. Comparing LBP, HOG and Deep Features for Classification of Histopathology Images. *Appl. Sci.* **2018**, *12*, 348–362.
15. Gao, C.C.; Hui, X.W. GLCM-Based Texture Feature Extraction. *Comput. Syst. Appl.* **2010**, *19*, 195–198.
16. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 365–374.
17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. In *Proceedings of European Conference on Computer Vision*; Springer: Cham, Germany; 8–16 October 2016, pp. 21–37.
18. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, 18–22 June 2018; pp. 4203–4212.
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 779–788.
20. Cheng, M.-m.; Mitra, N.J.; Huang, X.; Torr, P.H.; Hu, S.-M. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 569–582. [CrossRef]
21. Perazzi, F.; Krähenbühl, P.; Pritch, Y.; Hornung, A. Saliency filters: Contrast based filtering for salient region detection. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 733–740.
22. Zhang, J.; Sclaroff, S. Saliency detection: A boolean map approach. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 25–27 June 2013; pp. 153–160.
23. Hou, X.; Zhang, L. Saliency detection: A spectral residual approach. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 20–23 June 2007; pp. 110–118.
24. Achanta, R.; Hemami, S.; Estrada, F.; Süsstrunk, S. Frequency-tuned salient region detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Miami Beach, FL, USA, 20–25 June 2009; pp. 1597–1604.
25. Achanta, R.; Süsstrunk, S. Saliency detection using maximum symmetric surround. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 2653–2656.
26. Shawe-Taylor, J.; Cristianini, N. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. *Support Vector Mach.* **2000**, *10*, 93–112.
27. Yang, W.; Wang, Y.; Vahdat, A.; Mori, G. Kernel latent SVM for visual recognition. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, CA, USA, 10–12 December 2012; pp. 809–817.
28. Chen, P.H.; Lin, C.J.; Schölkopf, B. A tutorial on ν-support vector machines. *Appl. Stoch. Mod. Bus. Ind.* **2005**, *21*, 111–136. [CrossRef]
29. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef]
30. Borji, A.; Cheng, M.; Hou, Q. Salient Object Detection: A. Survey. *Comput. Vis. Med.* **2019**, *5*, 117–150. [CrossRef]

31. Li, C.; Hu, Z.; Xiao, L.; Pan, Z. Saliency detection via low-rank reconstruction from global to local. In Proceedings of the Chinese Automation Congress (CAC), Guangzhou, China, 27–29 November 2016; pp. 669–673.
32. Ito, S.; Soga, M.; Hiratsuka, S.; Matsubara, H.; Ogawa, M. Quality Index of Supervised Data for Convolutional Neural Network-Based Localization. *Appl. Sci.* **2019**, *9*, 1983. [CrossRef]
33. Koishi, Y.; Ishida, S.; Tabaru, T.; Miyamoto, H. A source domain extension method for inductive transfer learning based on flipping output. *Algorithms* **2019**, *12*, 95. [CrossRef]
34. Sun, Q.; Liu, Y.; Chua, T.-S.; Schiele, B. Meta-transfer learning for few-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–19 June 2019; pp. 403–412.
35. Chen, C.-H.; Kung, H.-Y.; Hwang, F.-J. Deep learning techniques for agronomy applications. *Agronomy* **2019**, *9*, 142. [CrossRef]
36. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation policies from data. *arXiv* **2018**, arXiv:1805.09501.
37. Lang, C.; Liu, G.; Yu, J. Saliency Detection by Multitask Sparsity Pursuit. *IEEE Trans. Image Process.* **2012**, *21*, 1327–1338. [CrossRef] [PubMed]
38. Jang, J.; Li, G. Research on Automatic Monitoring Method of River Floats. *Yellow River* **2010**, *10*, 13–15.