

Research Article

A Study on Removing Cloud Drift of Sky-Sea Infrared Image Based on Agent

Jianming Sun,^{1,2} Zhipeng Fan ,^{1,2} Zhihui Sun,¹ Qinghua Miao,¹ and Xiaoyan Li¹

¹Harbin University of Commerce, Harbin, Heilongjiang 150028, China

²Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin, Heilongjiang 150028, China

Correspondence should be addressed to Zhipeng Fan; hsdfzp@126.com

Received 25 September 2018; Accepted 26 December 2018; Published 3 April 2019

Guest Editor: Subramaniam Ganesan

Copyright © 2019 Jianming Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

According to the characteristics of cloud-drift area in an infrared sky-sea line image, we put forward a new shadow extraction method by using a layered reactive agent. The layered reactive agent includes a lower cloud search agent and higher management agent. The lower distributed layer search agent can sense local information of the image and retrieve the local cloud area in the image by locating operating point, enlarging scope, moving, marking, perishing, and so on. The higher management agent can sense the information of the whole and restrict and guide search action to the lower distributed search agent. The result of the simulation test shows the efficiency of this method is high, and this method extracts effectively and removes cloud in the infrared sky-sea image.

1. Introduction

To the infrared image of sky-sea background, we can ensure ship attitude by extracting sky-sea line. Cloud appears when the target image is operated by infrared, which will directly influence the success rate of sky-sea line recognition algorithm in that area. Meanwhile, the cloud-drift line often appears on the sea, which will be mistaken for sky-sea line. Therefore, the study on removing cloud has great significance on extracting sky-sea line. To solve this problem, some scholars propose several solutions. They classify the prior knowledge according to the environment condition [1]. They obtain the law of cloud change by statistical characteristics, compare the real image and this change law, and then build a thin cloud layer and cloud-drift removal model [2]. This method needs to use prior knowledge of cloud in some period before operation, so many image data cannot be operated because of the lack of enough prior knowledge. Another kind of method is recognition algorithm of cloud extraction without prior knowledge, like the search method of histogram peak value [3]. This method is simple, so it cannot be used because of its own shortcoming especially in complicated situation. And, this method can only be used in some special images, and hence, it is not fit for other images.

Through observing enough sky-sea infrared images, we can find that brightness of sky outside the cloud decreases significantly, the grey value in the same cloud changes little compared with the noncloud area, and the grey value between clouds distributed around the image has little change.

Agent refers to an active entity which stays in some situation and acts autonomously and cooperates with other Agents in order to achieve the goal of design. Residence, autonomy, and sociality are the basic features [4]. A multi-agent system is composed of Agents which cooperate with. Because of residence and autonomy of the Agent, it can adapt to environmental change effectively, which makes the system more flexible. Sociality of the Agent is an effective way to solve problems. Therefore, MAS is considered as an enabling technology which supports complicated technology and has great potential [5]. This technology offers many engineering ways like high-level abstraction, resolution of problem, and layer and system organization. Some practices show the advantages and potentials of MAS in some complicated system developments, such as production process, intelligence control, distributed network management, distributed control system, and telemedicine system [6].

Agent has been the focus of the research of artificial intelligence and software science because of its features like

perceptions [7], problem-solving ability, and cooperation. People make the program of Agent which can do random search and grey test of similar area. This program is applied to computer tomography, and the results are very good. In light of cloud scope and its distribution, this paper reveals an image processing method based on Agent. This method overcomes the shortcomings of the above methods and can extract efficiently and remove cloud of infrared images in different conditions.

2. Agent Model

2.1. Individual Agent Layer. In individual Agent layer, the research on AOP design abstract and model is to offer description software for autonomous decision action of the Agent. The inside of the Agent has higher concept and abstract, whose core is how to describe decision of the Agent inside. Then, the corresponding individual Agent software model is built. The individual Agent software model supported by AOP has 4 types: knowledge mode, cognitive mode, reactive mode, and hybrid mode. The knowledge mode takes Agent as a knowledge system [8]. The inside structure is composed of abstract and concept based on belief, knowledge, and distributed knowledge. We can use belief modification, knowledge reasoning, situation calculus, and so on to support autonomous decision of Agents by logic tools. AOP language which supports the knowledge mode includes Golog, AGTGolog, and Con Golog. The cognitive mode takes the Agent as a cognitive system. Based on cognitive science and folk psychology like goal, desire, intention, and plan, the inside structure of the Agent can be described by practical reasoning, BDI, and KARO to support Agent's autonomous decision [9]. Most AOP languages can be integrated into an idea, like Agent-0, 3APL, PLACA, Agent Speak (L), AOPLID, GOAL, Dribble, CLAIM, and 2APL [10]. Agent-0 takes the Agent as an active entity which consists of belief, capability, commitment, and action. PLACA expands intention cognitive component to support goal-directed action [11]. Agent Speak (L) and CLAIM are based on the BDI model. The Agent model of 3APL is based on belief, goal, and plan. Furthermore, target concepts of ADP can be divided into procedural one and declarative one. Procedural target corresponds to specific planning, focusing on goal-to-do [12]. Declarative target focuses on goal-to-be, which introduces GOAL. Dribble and 2APL support these two types of targets. The reactive mode takes the Agent as a reactive system [13], which can sense environment and its change and can response to these changes and process them. The reactive Agent model contains events and reactive rules, which can support autonomous decision of individual Agent's action based on events and events processing. For example, SLABSp uses action rule to define the action of Agent and describes the action that the Agent takes when some scene can be met [14]. The hybrid mode integrates the above software types to support structure and realization of the Agent, which uses various abstract and concepts to describe every element that can construct the model.

2.2. Multiagent Layer. In multiagent layer, the research on AOP design abstract and model focuses on how to provide effective concepts and models to support action of the Agent in MAS and organize and adapt to it, which can ensure its operation of cooperation of MAS and acquire the whole action of MAS. In terms of the software development, organization idea provides a feasible decomposition for the design of MAS [15]. And, a diversity of organization structure provides feasible structure for MAS, including layer organization, holonic organization, league, team, gathering, society, federation, market, and matrix organization. Now, organization structure supported by AOP includes team, regulatory organization, structured organization and hybrid organization. Team takes many Agents, which complete a complicated task by cooperation, as a team. This model often uses the traditional BDI model of the Agent to expand joint intention and team planning to build and describe team and guide or control the decision of Agent to achieve the cooperation of multiagent. For example, Simple Team describes multiagent team by describing some concepts like increasing roles to Jack, team ability, team planning, and so on. The Agent can achieve cooperation by executive team. EAMCORE can fulfill coordination between Agents based on team planning and group belief. A BDI structure of the individual Agent expands content and context in order to describe group concept [16]. Regulatory organization takes organization as a group of Agent and a rule set, which can define organization's control and restriction to the action and interaction of the Agent based on laws from sociology. According to different nature, law includes obligation, permission, and prohibition. In light of different contents of restriction, law can be classified action law and status law. And, in light of different enforcement mechanisms, law can be divided into regimentation law and sanction law. Now, typical AOP language supporting regulatory organization has ISLANDER, NOPL, and AOP. ISLANDER defines law based on action, which can prohibit or forbid the action that the Agent executes. All rules of ISLANDER cannot be obeyed. NOPL is a program design language to organization management infrastructure, not to the language for the programmer. Because of the specialty of OMI (violating laws will bring fatal mistakes to platform), NOPL is a simple law program design language which only supports compulsive obligation. That is, the law of NOPL only describes the action which the Agent executes and all the rules cannot be obeyed. Laws defined by status support three kinds of rules based on obligation, permission, and prohibition and provide punishment mechanisms to the Agent violating rules.

2.3. Reactive Agent Model. Reactive Agent can be shown by symbols, which can respond to the changes of external environment. Reactive structure is designed by corresponding action of the assuming Agent. Its complication of action reflects the one of practical environment of the Agent. Structure of the reactive Agent is shown in Figure 1.

Defined environment is the finite set of all discrete and instantaneous status:

$$E = \{e, e^t, \dots\}. \quad (1)$$

Agent has an action set that can complete. These actions can change the status of the environment. The finite set can be shown as follows:

$$Ac = \{\alpha, \alpha^t, \dots\}. \quad (2)$$

R which is an action in some environment of Agent is a sequence between environment status and action alternate replacement:

$$R : e_0 \xrightarrow{\alpha_0} e_1 \xrightarrow{\alpha_1} \dots \quad (3)$$

The action of reactive Agent can be shown as follows:

$$Ag : E \xrightarrow{see} Per \xrightarrow{plan} Per^* \xrightarrow{action} Ac. \quad (4)$$

In equation (4), see, plan, and action stand for environment perceptor, restriction condition, and corresponding action sequence, respectively. Per and Per* stand for the prior sensing range and anticipation set, respectively.

3. Action Design of Reactive Agent in Cloud Testing

3.1. General Frame Structure of Layer Reactive Agent. The structure of layer agent has higher management agent and lower cloud search agent. These two agents fit to reactive agent model, shown in Figure 2.

The function of Search Agent is to sense part of the image environment and then respond to the content of sense. The Search Agent can search and mark cloud scope by its own intelligent action. However, this agent cannot grasp the whole process and information. The management agent can make up the disadvantage of the search agent, which can acquire the whole information of the image and guide and control the search agent.

3.2. Data Definition and Design of the Search Agent. In order to make sure the working condition of the search agent, we should show three measurements of local cloud similarity of the agent. Their influence equals to E of equation (1). Its definition can be shown as (i).

(i) The number of local grey similarities

$$C_{[i,j]_{-region}} = \sum_{\|(i,j)-(k,l)\| \leq R_{(i,j)_{-region}}} \rho(i, j, k, l). \quad (5)$$

Then,

$$\rho(i, j, k, l) = \begin{cases} 1, & \|I(i, j) - I(k, l)\| \leq \delta, \\ 0, & \text{other.} \end{cases} \quad (6)$$

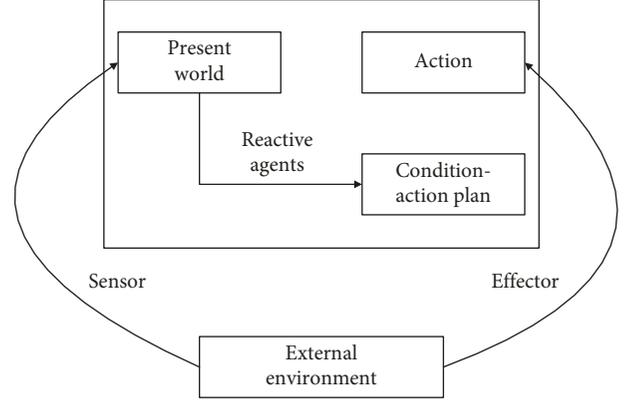


FIGURE 1: Structure of the reactive agent.

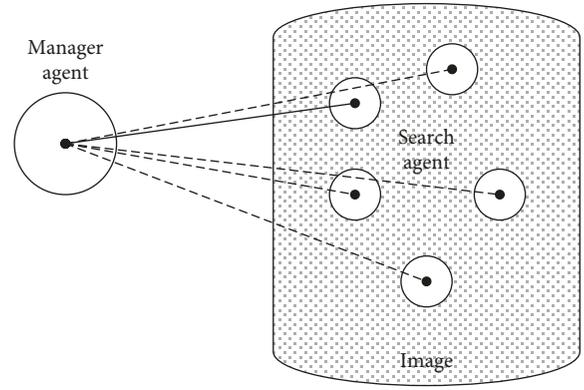


FIGURE 2: Structure of layer agent.

(ii) Expectation of the local area:

$$\text{mean}_{(i,j)_{-region}} = \frac{1}{N} \sum_{\|(i,j)-(k,l)\| \leq R_{(i,j)_{-region}}} I(k, l), \quad (7)$$

$$\text{std}_{(i,j)_{-region}} = \sqrt{\frac{1}{N} \sum_{\|(i,j)-(k,l)\| \leq R_{(i,j)_{-region}}} (I(k, l) - \text{mean}_{(i,j)_{-region}})^2}. \quad (8)$$

In equation (8), $I(i, j)$ is the grey value of (i, j) . In equation (6), δ is the empirical parameter and $R_{(i,j)_{-region}}$ in Figure 3 is the action radius.

The distributed searching agent directly acts on some pixels of the local image and performs calculation to neighboring environment similarity of local cloud. Cloud-line scope can be searched by locating the working site, expanding scope, moving, marking, and dying out. These actions can be designed to touch off in some condition, which is condition design action of the searching agent. Locate working site: the agent can sense and acquire neighboring information and judge whether this environment adapts to retrieve working. When the environment is fit for retrieving, the agent locates the working site. The rule of locating working site is

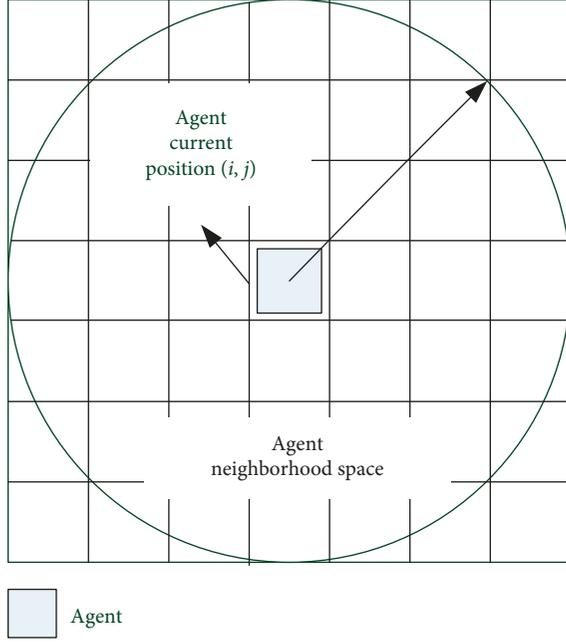


FIGURE 3: Local neighboring region of a shadow-searching agent.

- (i) $R_{(i,j)\text{-region}}$ has characteristics of given images
- (ii) $R_{(i,j)\text{-region}}$ has characteristics of images similar to cloud lines

Rule (i) uses prior knowledge (like the scope of grey value and texture properties) in the environment of practical images in order to guarantee that the agent locates the working site of local images, which can decrease repeat calculation. This paper does not require rule (i) in simulation test algorithm.

Rule (ii) can locate the working site when the agent has pixel of cloud characteristics. Restriction conditions that this algorithm has are as follows:

$$\begin{aligned}
 C_{(i,j)\text{-region}} &\geq C_{\min}, \\
 R_{(i,j)\text{-region}} &\leq M_{\max}, \\
 R_{(i,j)\text{-region}} &\leq S_{\max}.
 \end{aligned} \tag{9}$$

Expanding scope: the searching agent can expand searching space. In this algorithm, the method of expanding searching space is to breed. From the characteristics of cloud, after some searching agent locates working site, we consider its neighboring pixel is the cloud point. The steps by which the searching agent expands scope to breed are as follows. First, we assume the image of A has some agents. A will have 4 offspring after A locates the working site. The prior location of these four offspring is four neighboring points of that agent:

$$A_{(i,j)}^g \longrightarrow \left\{ A_{(i-1,j)}^{(g+1)}, A_{(i+1,j)}^{(g+1)}, A_{(i,j-1)}^{(g+1)}, A_{(i,j+1)}^{(g+1)} \right\}. \tag{10}$$

In this equation, $A_{(i,j)}^{(g)}$ means generation g and working site is the agent of (i, j) .

Moving: the searching agents, which cannot locate working site successfully, need to move in retrieval images to obtain suitable working situation in order to get chances of expanding scope in retrieval work. This is called moving. In the process of moving, the age of them increases gradually. This searching action of the agent can drive the searching agent to find new environment of doing retrieval work.

The specific process of moving in the searching agent can be shown as follows. To the agent that cannot locate the working site and can be defined as $A^{(g+1)}$ to generation $g+1$, we need to search its parent $A^{(g)}$. If its parents do not exist, it shows that agent is the first generation. Then, we can select a direction of moving e and moving distance r randomly. If its parents $A^{(g)}$ exist, we can find $\{B^{(g+1)}\}$ of generation $g+1$ by $A^{(g)}$. We accumulate these points to locate moving direction of agents that have located working site successfully and make up a histogram of counting directions. At last, the ratio between the value of every moving direction and its sum is probability. There is a moving direction e that is made randomly and (e, r) of moving direction r distributed in $[-R, R]$. R is the maximum moving distance, and a new position of $A^{(g+1)}$ in (e, r) appears. This moving mechanism can use family to transfer experience to locate working site of searching site from the aspect of possibility.

Marking: after some searching agent locates working site of a pixel successfully or unsuccessfully, this agent records its result information on the image. If this environment adapts to do retrieval work, then $\text{label}(i, j) = \text{label} A$ or $\text{label}(i, j) = \text{label} B$. $\text{label} A$ and $\text{label} B$ are the given parameters, and label is the marking image.

Dying out: the searching agent will die out after experiencing expanding scope and exceeding maximum of moving. There are two situations in dying-out status:

- (i) After some agent marks and breeds successfully, it can finish algorithm and enter dead state, that is, dying-out status.
- (ii) When some agent cannot still locate the working site of meeting conditions after moving N times, the age of agent is N . If N is greater than maximum age set by the system, then dying-out status appears instantly. Extinction mechanism can eliminate the agent with weak ability and drive population to find new environment suitable to do retrieval work, which avoids optimal solution in local part.

3.3. Management Agent Design. The main task of management agent is to judge whether the searching action of the searching agent can meet requirement. This paper defines the best measure of the whole searching as equation (11) according to the characteristics of cloud:

$$R^{(g)} = \sqrt{\frac{1}{N} \sum_{L(i,j)=\text{label}A} (I(i, j) - M^{(g)})^2}. \tag{11}$$

Then,

$$M^{(g)} = \frac{1}{N} \sum_{L(i,j)=\text{label}A} I(i, j). \quad (12)$$

$I(i, j)$ means the grey value of this image in (i, j) . $\text{label}(i, j)$ is the grey value of the marked image in (i, j) . $\text{label} A$ is the marking value when the searching agent meets the conditions of the working site. $R^{(g)}$ is the best measure of the searching agent of generation g .

The management agent can get $R^{(g)}$ after every iteration according to best measure. If $R^{(g)} > R_M$, it means the current working condition cannot meet the best condition and the working condition needs to be adjusted.

The formulas locating working site include formulas (5)–(7), in which there are three parameters $\{C_{\min}, M_{\max}, S_{\max}\}$. Here, $\{C_{\min}, S_{\max}\}$ often are unchangeable to different sky-sea images. $\{M_{\max}\}$ is related to the brightness of scene and imaging condition. This algorithm is to adjust to $\{M_{\max}\}$, and the process of adjustment can be shown as equation (13).

$K_{\text{threshold}}$ is the threshold of image segmentation when we cluster two kinds of grey K – mean to image $R = \text{label} \& I$.

The existence of the management agent can claim the tolerance of the lower-layer cloud test agent can adjust itself to the environment and make the algorithm suitable to image in some scene.

4. Fulfillment of Algorithm

4.1. Acquisition of Information of Management Agent. When the multiagent handles the similar events, the prior self-testing often executes when we find $|V_m| < (N + 1)/2$ at the end of the task (we call it self-diagnosis). However, results that we get in the recognition decision process described in this paper are instantaneous result. If self-testing can be done after one task period is over, the results that we get in the process no longer exist. In order to know the events happening during the task period, we put forward a method. The corresponding detection code can be inserted in the original task codes dispersedly, and environment information can be extracted real-timely in task operation process. If the original task code sequence is $\{t_1, t_2, \dots, t_n\}$ and detection code sequence is $\{d_1, d_2, \dots, d_m\}$, we get $\{t_1, d_1, t_2, d_2, \dots, d_m, t_n\}$ when $\{d_1, d_2, \dots, d_m\}$ is inserted into $\{t_1, t_2, \dots, t_n\}$ distributedly. If insertion methods are different, sequences are different. Detection code sequence $\{d_1, d_2, \dots, d_m\}$ designed reasonably is the key to voting algorithm. We design a detection sequence and take it as principle verification. This detection sequence is mainly used as an intermediate result that Agent outputs. Detection sequence is as in Algorithm 1.

If recognition result of every Agent in operation process is false (0), Detect Result should be false. If some Agents can recognize right result in some task period q , we should judge the possibility of results according to most voting principles.

When intermediate result of function Agent is greater than one of the common Agents and results are considered real, then the results are real. This voting function can output specific values of results. If three agents consider results real,

the output voting result is 3. The following processing can give further conclusion according to this result. In the system in which reliability requirement of identification recognition is strict, threshold can be improved. Conversely, in the system in which reliability requirement is not strict, this threshold can be decreased appropriately.

4.2. Image Segmentation. In the method of image segmentation based on the agent, the agent points judge whether it meets the consistency criteria by sensing the pixels in its neighborhood. The agent point performs its subsequent behavior through the feedback information of its neighboring regions; it may reproduce offspring and moves to neighboring pixels or from the image disappear.

Usually, we use the following three mathematical criteria to measure whether the consistency region is satisfied: relative contrast, regional average, and regional gray standard variance. Agent point of behavior will be based on the above 3 criteria to determine whether to trigger local stimulation; more detailed consistency criteria are defined as follows:

Definition of contrast formula:

$$g(i, j) = \sum_{\|(i,j)-(k,l)\| \leq R(i,j)} \rho(i, j, k, l), \quad (13)$$

$$\rho(i, j, k, l) = \begin{cases} 1, & \text{if } \|I(i, j) - I(k, l)\| \leq \delta, \\ -1, & \text{others,} \end{cases} \quad (14)$$

where, $g(i, j) \in (\eta_1, \eta_2)$ is a predefined constant; $R(i, j)$ is the neighborhood radius of pixel (i, j) for agent; $I(i, j)$ is the grey scale of pixel (i, j) ; δ is the threshold value of predefinition.

Mean of standard region:

$$\text{mean}_{(i,j)} = \frac{1}{N} \sum_{\|(i,j)-(k,l)\| \leq R(i,j)} I(k, l). \quad (15)$$

Standard region variance:

$$\delta_{\text{stand}(i,j)} = \sqrt{\frac{1}{N} \sum_{\|(i,j)-(k,l)\| \leq R(i,j)} (I(k, l) - \text{mean}_{(i,j)})^2} \quad (16)$$

where $\text{mean}_{(i,j)} \in (\mu_1, \mu_2)$, while μ_1, μ_2 is the constant of predefinition; $\delta_{\text{stand}(i,j)} \in (d_1, d_2)$, d_1, d_2 is the constant of predefinition; N is the number of pixels for region $R_{(i,j)}$.

4.3. Replication and Diffusion. Agent can adopt two different modes of behavior, replication and diffusion, corresponding to different local environments, which become the adaptability of agent. Diffusion is to move up the current pixel to the other ones. Specific process is as follows:

- (1) When an agent searches a pixel satisfying the above three criteria consistency, it will copy the appropriate set of descendant agent in a particular direction on the neighborhood. The copy behavior makes the newly generated offspring locate near the pixels

```

<DetectCode> → <Init>; <DetectPair>*;
<Finish>;
<Init> → DetectI = 0; DetectJ = 0;
<DetectPair> → DetectI = DetectI + 1.0;
DetectJ = DetectJ + 1.0;
<Finish> → DetectResult = DetectI - DetectJ.
(*) means <DetectPair> can be repeated many many times. The typical period task can be described as follows after inserting
detection code:
// states the global variable that detection codes can use out function.
while (1) // period task, repeat operation repeatedly
{
Init // test initialization, test variable zeroes
t1; // the beginning a period, like sensor information
// every task code, insert a detect pair:
Detect Pair
t2;
t3;
Detect Pair
.....
Detect Pair
tn; // when algorithm finishes, voting is prepared
Finish // standardize detect results
r = vote(r, Detect Result); // vote with self-testing
output(r); // output control information
}

```

ALGORITHM 1

which can meet the consistency criteria, for subsequent detection of further regional coherence:

$$\alpha_{(i,j)}^g \Rightarrow \left\{ \alpha_{\nu(w, d_{\text{copy}})} \right\}^{\nu} \quad (17)$$

$$= 1, 2, \dots, m; \omega \in \Omega; d_{\text{copy}} \leq k \}^{g+1}.$$

where g and $g + 1$ is the g and $g + 1$ offspring of the agent, (i, j) is the current position of the agent, α is an agent with active status, ν is the ν offspring of the agent, m is the total amount of the offspring agent, ω is the replication direction of the offspring agent, Ω is the replication direction of a series of possibilities, d_{copy} represents the distance of replicating the offspring agent, and R_c is the replicating radius.

- (2) When an agent finds itself is in a nonuniform region, it will select diffusion pattern, moving along a particular direction to a specific location. Diffusion behavior also plays an important role in the discovery process of consistency area because the diffusion direction is determined by the agents that successfully found consistency area among the parent agent and the agent on behalf of the brothers. The new diffusion agent is still in the neighborhood; its proliferation is not a search for balance but should be seen as looking for a new biased search of consistency pixels:

$$\alpha_{(i,j)}^t \Rightarrow \alpha_{\theta, d_{\text{diffu}}}^{t+1}, \quad \theta \in \Theta, d_{\text{diffu}} \leq R_d. \quad (18)$$

where t and $t + 1$ represent the time of the agent, θ is the spreading direction, Θ is the spreading direction of a series of possibility, d_{diffu} represents the spreading distance, R_d denotes the spreading radius.

- (3) When the agent found a consistency area, which itself is placed in the suppressed state.

Meanwhile, in order to prevent the agent's unlimited searching, its provisions can be a "life cycle," over the life cycle and it is self-suppression.

4.4. Steps of Algorithm

Step one: build an Agent group $\{A_i\}_{i+1}^k$, in which is given the parameter that shows the number of agents. We take the age as 1 and put it to active queue of agent.

Step two: to every agent of active queue, we can make sure whether that agent can locate the working site according to the working condition. If the working site can be located, all corresponding pixel points of that agent can be set as label A , and then the offspring are propagated and added to the active queue of the agent. Meanwhile, it itself will be deleted from the active queue of agent. If the working site cannot be located, corresponding pixel points of that agent can be set as label B first and then we should judge whether its age is greater than the given maximum age. If it is greater, it will be deleted from active queue of agent or it will move. The new coordinate point can be built, and its age will be added to 1.

Step three: Through equation (11), we can get best measure of the current marked image. If measure is greater than the preinstalled parameter, we can correct $\{M_{max}\}$ in the working condition by equation (15) and return to step one. If measure is smaller than the preinstalled parameter, we will begin step four.

Step four: Judge whether active queue of agent is empty or all images are marked. If it is, we can begin step five or return to step two.

Step five: To do subsequent processing of processed images, we can make grey value label A of the original image 1, and other grey values which are not equal to label A are made 0. In this way, we can get an image with two values. Then, we perform dilation to this image and perform dry operation, and we can get target images.

5. Experiment and Result Analysis

In order to test the correctness of this method, we select a group of typical sky-sea pictures as the simulation testing image, including four types of targets. Figure 4 shows a common cloud image including only sea and sky. Figure 4 also reveals a gradual process of removing cloud through this algorithm. The process of operation takes 5.22 seconds.

The parameters that this image uses are as follows: the number of initial agent is 500, the maximum existence age is 5, neighboring search diameter is 4, and the minimum value of similar points of local grey is 3. The maximum initial value in the local area is the grey expectation value of the input image, and the maximum value of local standard deviation is 9.

Simulation platform: simulation software is Matlab 2010a, CPU domain frequency is 3.2 G, and memory is DDR2GB. Figures 5 and 6 are the typical sky-sea image and corresponding cloud test result. Figure 5 is the tilt image for camera angle and the result of testing cloud through that method. Figure 6 is the sky-sea image with occlusion like trees and the result of testing cloud through this method. Furthermore, we select another 9 images to every type of target to do test simulation. In the table of simulation test result, we compare cloud testing result through this method with cloud testing through artificial vision and we perform statistics to ratio in which detection and missed test scopes account for the whole cloud scope, like Table 1.

Through observing simulation results, we find this method can search and remove cloud to different sky-sea images and cannot limit size and shape of cloud in searching. The test result of images with tilt angle and images with occlusions shows that this method can overcome the influence of cloud recognition to complicated scenes. The total operation time of computer is so little, which means time complicity of this algorithm is low. The operation time of a single infrared image (256×256 pixel) can be controlled between 1 and 3 seconds. Therefore, this method can be used widely.



(a)



(b)



(c)

FIGURE 4: Cloud extraction and removal of common images.

6. Conclusion

We can test and remove cloud of sky-sea images, which can improve recognition efficiency of sky-sea line and acquire tilt angle of carriers. This paper puts forward a better algorithm method of testing and removing cloud of infrared images based on cloud characteristics of analyzing infrared sky-sea images. Through grey value characteristics of cloud, we can find and use reactive agent layer structure, classify many agents used for local image cloud searching, and manage agents used for coordinating many cloud searching agents. We can extract and remove cloud of the infrared sky-sea image by intelligent algorithm and cooperation of two agents. Simulation shows this method has good operation effect to various shooting situations. And, time

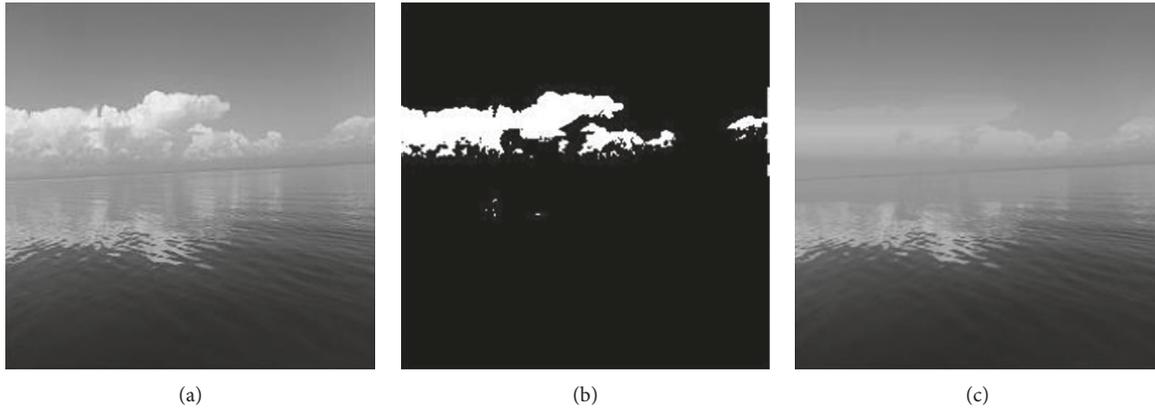


FIGURE 5: Cloud extraction and removal of the sky-sea image with tilt angle.

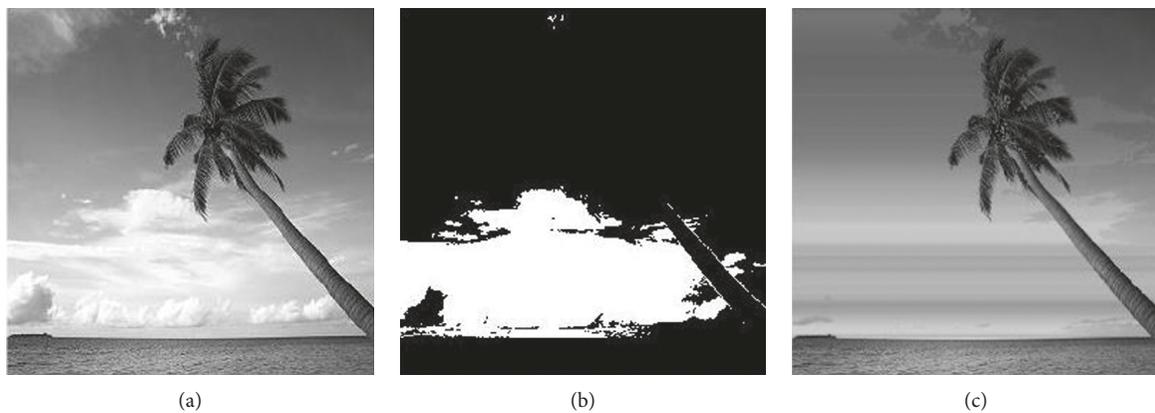


FIGURE 6: Cloud extraction and removal of the sky-sea image with occlusions.

TABLE 1: Experiment result.

Types of image	Total after-detection rate (%)	Total missed detection rate (%)	Average operation time (s)
Common sky-sea image	2.71	3.25	1.56
Image with tilt angle	3.75	4.80	2.12
Image with occlusions	7.53	5.01	2.62

complicity of this algorithm is low, and cloud test and removal task in the infrared sky-sea image can be completed fast and efficiently.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work was supported by the National Natural Science Foundation of China (51476049) and Harbin

University of Commerce Young Creative Talents Support Project (2016QN050 and 17XN005).

References

- [1] J. J. Yoon, C. Koch, and T. J. Ellis, "ShadowFlash: an approach for shadow removal in an active illumination environment," in *Proceedings of the 13rd British Machine Vision Conference*, pp. 636–645, INSPEC, Cardiff, UK, 2002.
- [2] L. Yan and P. Gong, "Cloud test of images based on DSM cloud simulation and ray tracing in high area," *Journal of Remote Sensing*, vol. 9, no. 4, pp. 357–362, 2005.
- [3] P. Gils, "Remote sensing and cast shadows in mountainous terrain," *Photogrammetric Engineering and Remote Sensing*, vol. 67, no. 7, pp. 833–839, 2001.
- [4] H. Guo, Q. Xu, and B. Zhang, "Building cloud extraction in the multiple constraint," *Wuhan University Journal*, vol. 30, no. 12, pp. 1059–1062, 2005.

- [5] Y. Yang, R. Zhao, and W. Wang, "Test of cloud area in air image," *Signal Processing*, vol. 18, no. 3, pp. 228–232, 2002.
- [6] J. Liu and Y. Y. Tang, "Adaptive image segmentation with distributed behavior-based agents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 544–551, 1999.
- [7] H. He and Y. Q. Chen, "Artificial life for image segmentation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 6, pp. 989–1003, 2001.
- [8] C. Pereira, D. Veiga, J. Mahdjoub et al., "Using a multi-agent system approach for microaneurysm detection in fundus images," *Artificial Intelligence in Medicine*, vol. 60, no. 3, pp. 179–188, 2014.
- [9] Y. Fan, L. Liu, G. Feng, and Y. Wang, "Self-triggered consensus for multi-agent systems with zeno-free triggers," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2779–2784, 2015.
- [10] R. H. Baxter, N. M. Robertson, and D. M. Lane, "Human behaviour recognition in data-scarce domains," *Pattern Recognition*, vol. 48, no. 8, pp. 2377–2393, 2015.
- [11] N. C. A. D. Freitas, P. P. R. Filho, C. D. G. D. Moura, and M. P. D. S. Silva, "AgentGeo: multi-agent system of satellite images mining," *IEEE Latin America Transactions*, vol. 14, no. 3, pp. 1343–1351, 2016.
- [12] J.-R. Ruiz-Sarmiento, C. Galindo, and J. Gonzalez-Jimenez, "Scene object recognition for mobile robots through semantic knowledge and probabilistic graphical models," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8805–8816, 2015.
- [13] N. Shiroma, R. Miyauchi, A. Nagafusa, Y. Haga, and F. Matsuno, "Gaze direction based vehicle teleoperation method with omnidirectional image stabilization and automatic body rotation control," *Advanced Robotics*, vol. 29, no. 3, pp. 149–163, 2015.
- [14] D. Wang, L. Liu, X. Wang, and Y. Lu, "A novel feature extraction method on activity recognition using smartphone," *Web-Age Information Management*, vol. 351, pp. 67–76, 2016.
- [15] L. Zhuo, Z. Geng, J. Zhang, and X. G. Li, "ORB feature based web pornographic image recognition," *Neurocomputing*, vol. 173, pp. 511–517, 2015.
- [16] J. M. Beer, C.-A. Smarr, A. D. Fisk, and W. A. Rogers, "Younger and older users' recognition of virtual agent facial expressions," *International Journal of Human-Computer Studies*, vol. 75, pp. 1–20, 2015.



Hindawi

Submit your manuscripts at
www.hindawi.com

