Scene learning for cloud detection on remote sensing images

Zhenyu An and Zhenwei Shi*, *Member IEEE*

***Abstract*—Cloud detection plays a major role for remote sensing image processing. To accomplish the task, a novel automatic supervised approach based on the "scene learning" scheme is proposed in this paper. Scene learning aims at training and applying a cloud detector on the whole image scenes. The cloud detector herein is a special classifier that is used to separate clouds from the backgrounds. Concretely, scene learning regards each pixel of scenes in training image as a sample, and use it to train a cloud detector. Accordingly, the detecting process is also implemented on each pixel of testing image using the trained detector. Generally, scene learning scheme contains two modules: feature data simulating, cloud detector learning and applying. We first simulate a kind of cubic structural data (also named feature data) by stacking different fundamental image features, including color, statistical information, texture and structure. Such data synthesizes different image features, and it is used for cloud detector training and applying. Cloud detector is designed based on minimizing the residual error between the feature data and its labels. The detector is easily to be trained because of its closed-form. Apply the detector and some necessary cloud refinement methods to the testing images, we could finally detect clouds. We also theoretically analyze the influence of feature number and prove that more features lead to better performance of scene learning under certain circumstance. Comparisons of qualitative and quantitative analyses of the experimental results are implemented. Results indicate the efficacy of the proposed method. *Index Terms*—Cloud detection, scene learning, feature data, cloud detector, cloud refinement.**

I. Introduction

With the development of image acquisition technology, high resolution remote optical images can be more easily obtained. These images are widely applied to city surveying, military target recognition, meteorology, change detection, mineral development, and many other fields. However, analyses on the optical images are often disturbed by clouds. According to Q. Zhang and C. Xiao's belief [1], "cloud covers more than 50% of the surface of the earth", and "many aerial photographs will contain cloud regions", which implies the universal presence of clouds in optical images. Clouds usually cause negative influence on the surface studies as they cover the objects on the grounds. The images may not be negligible for some special image processing tasks, like change detection, although the clouds only possess a low percentage in the whole scene. Therefore, as a crucial preprocessing step for many subsequent image analyses, cloud detection is a meaningful work.

In the early studies, researchers usually focused on detecting clouds in images with low spatial resolution, like NOAA/AVHRR images with about one kilometer square per pixel. Orthogonal transformation [2]—-"Tasseled Cap" transform was therefore proposed to locate the mist and clouds. Zhang et al. improved the method and developed a haze optimized transformation (HOT) [3], and used it for detecting and characterizing haze/cloud spatial distributions in Landsat scenes. O. Hagolle et al. proposed a multi-temporal cloud detection method [4]. It exploited the "sudden increase of reflectance in the blue wavelength" and obtains effective results. However, the method needs a set of images for in the same grounds at different time, and the demand is not always satisfied. High temporal and spectral resolutions were also widely applied in cloud detection. Cihlar et al. exploited the normalized difference vegetation index (NDVI) to detect cloud-contaminated pixels, and optionally replace them with interpolated values [5], and the method is supported by the AVHRR data. Related work could also be referred to deshadow or dehaze. Richter and Muller developed a de-shadowing technique for multispectral and hyperspectral imagery over land [6]. Richter proposed a haze removal method for multispectral resolution satellite sensors [7]. Long et al. [8] proposed an effective and fast dehazing method for single remote sensing image. However, those methods are not designed for detecting clouds in remote sensing images and they may face problems.

Recent developments in machine learning provide more available approaches to cloud detection. Some researchers treated cloud detection as segmentation problem, namely, segmenting cloudy areas in the images. Based on prior knowledge of spectral properties, M. S. De Ruyter et al. separated clouds from backgrounds by applying multiple thresholds [9]. R. Rossi et al. proposed to extract features using SVD from the cloudy image and then use SVM to accomplish the detection task [10], but it required the images from the QuickBird and Landsat 7 satellites be co-registered. Q. Zhang and C. Xiao [1] proposed a progressive refinement scheme by using a series of steps to segment clouds from backgrounds in RGB color aerial photographs, but the parameters setting in the process

The work was supported by the National Natural Science Foundation of China under the Grants 61273245 and 91120301, the Beijing Natural Science Foundation under the Grant 4152031, the Program for New Century Excellent Talents in University of Ministry of Education of China under the Grant NCET-11-0775, the funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University under the Grant VR-2014-ZZ-02, and the Fundamental Research Funds for the Central Universities under the Grant YWF-14-YHXY-028 and the Grant YWF-15-YHXY-003. *(Corresponding author: Zhenwei Shi.)*

Z. An (e-mail: anzhenyu@sa.buaa.edu.cn) and Z. Shi (e-mail: shizhenwei@buaa.edu.cn) are with State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China and with Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China and also with Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China.

of segmentation should be set carefully. On the other hand, some researchers regarded the cloud detection problem as a classification task, which aimed at categorizing all pixels in the image as cloudy or noncloudy. S. Le Hégarat-Mascle and C. André used Markov random fields for cloud detection on high resolution optical images [11] based on three assumptions of clouds: clouds and shadows are connected; the image location of the shadow of a cloud is known if we know the geometry of the acquisition and the sun location; each cloud and its associated shadow have the same shape and area. G. Vivone et al. improved the classification rate by introducing a novel penalty term within the classical maximum a posteriori probability-Markov random field (MAP-MRF) [12]. However, the above methods face two main problems: 1) cloudy regions cannot be accurately distinguished from bright noncloudy regions 2) both of the algorithms are usually time-consuming

Since existing methods cannot solve the above problems, we propose a "scene learning" scheme to accomplish the task. In this paper, we consider cloud detection in the remote sensing images with only RGB colors which makes it still a challenging work because there are less spectral channels and relatively more complicated backgrounds in the images. The core principle of the "scene learning" scheme is to learn a cloud detector from the training data and then apply the detector in the testing data. One will see that the proposed scheme effectively overcomes difficulties in cloud detection task. The whole processing chain is illustrated in Fig. 1. Generally, the scheme belongs to the supervised learning approach, and it aims at training and applying a cloud detector on the whole image scenes. It contains two main modules: feature data simulating; cloud detector designing and applying. Feature data is a kind of new data with cubic structure. We extract different fundamental features (color, statistical information, texture and structure) from the original image and stack them together, then the feature data is formed. Therefore, each plain of the feature data is an image feature map as shown in Fig. 1. Such data synthesizes different image features which provides more information than original image. In the cloud detector designing and applying module, a cloud detector is designed and applied with a closed-form. To train the detector, we label all the pixels of training image, and then minimize the residual error of the label map and the feature data to train the detector. Next, applying the trained detector to the testing image, we could then obtain a saliency map, where cloudy areas are outstanding and backgrounds are effectively suppressed (as shown in Fig. 1). Finally, the detection results can be obtained by using effective cloud refinement methods.

There are some differences between the scene learning scheme and conventional learning approaches. First, we label all the pixels rather than small sampled patches. In conventional learning methods, researchers usually sample different small patches and learn classifier from them. However, in the proposed method, each pixel is labeled and the whole scenes of image are applied to obtain cloud detector in training step. Such implementation could exploit each pixel's information and benefit the subsequent cloud boundary refinement. Second, a closed-form cloud detector is designed which has a concise form and easily to be implemented. More importantly, it is more effective than other methods in the cloud detection task. Third, to apply the detector under the scheme, we propose to simulate feature data using different fundamental image features. To explore influence of feature number, we theoretically prove that more features of the data lead to better performance of the learning scheme.

The main contributions of our method can be summarized as follows:

i. A scene learning based cloud detecting frame is proposed. Under the frame, we design a novel closed-form cloud detector for remote sensing optical images, and the detector could be learned from all the pixels in the image instead of sample patches.
ii. We theoritically calculate the residual error for training data, and we also prove that more features will lead to better performance of detector.
iii. By utilizing the characteristics of clouds, cloud refinement based on optimal thresholding method and subsequent detail refinement is proposed.

The rest of this paper is organized as follows: In Section II, scene learning scheme is proposed. In Section III, we mainly discuss the feature data simulating and cloud detector learning, the approach for choosing and calculating image features, and properties of the detector. Also, theoretical demonstration on the influence of data dimensions is provided. In Section IV, cloud refinement work is presented. An optimal thresholding algorithm and subsequent necessary processing steps, including small object elimination and internal hole filling are combined to finally locate clouds. In Section V, experiments on the real data are implemented. The proposed method and the state-of-art cloud detection methods are compared in both subjective and quantitative evaluations. Finally, the paper comes to the conclusion in Section VI.

II. Scene learning scheme

In the section, scene learning scheme is discussed, which provides us a novel view for training and applying detector. It is superior to conventional learning scheme via exploiting the whole scenes of image and details about the scheme is discussed in detail as follows.

A. Conventional patch learning scheme and their problems

Supervised learning methods are widely applied for target detection and classification. Detector training is the most important step in the task. To train a classifier in conventional learning and training frame, the positive and negative data are segmented in small patches (usually smaller than the whole image). These patches usually have rectangle shape and are designed according to the object size. Different features, like histograms of oriented gradients (HOG) [13], local binary patterns (LBP) [14] and other feature descriptors, are then extracted from the patches, thus obtaining feature vectors for the patches (sample images). Obviously, patch size should be carefully designed and it is usually coincide with the target size. Once the size is fixed, we should first segment a large amount of positive and negative samples. Also, the detecting step is based on extracting and applying features

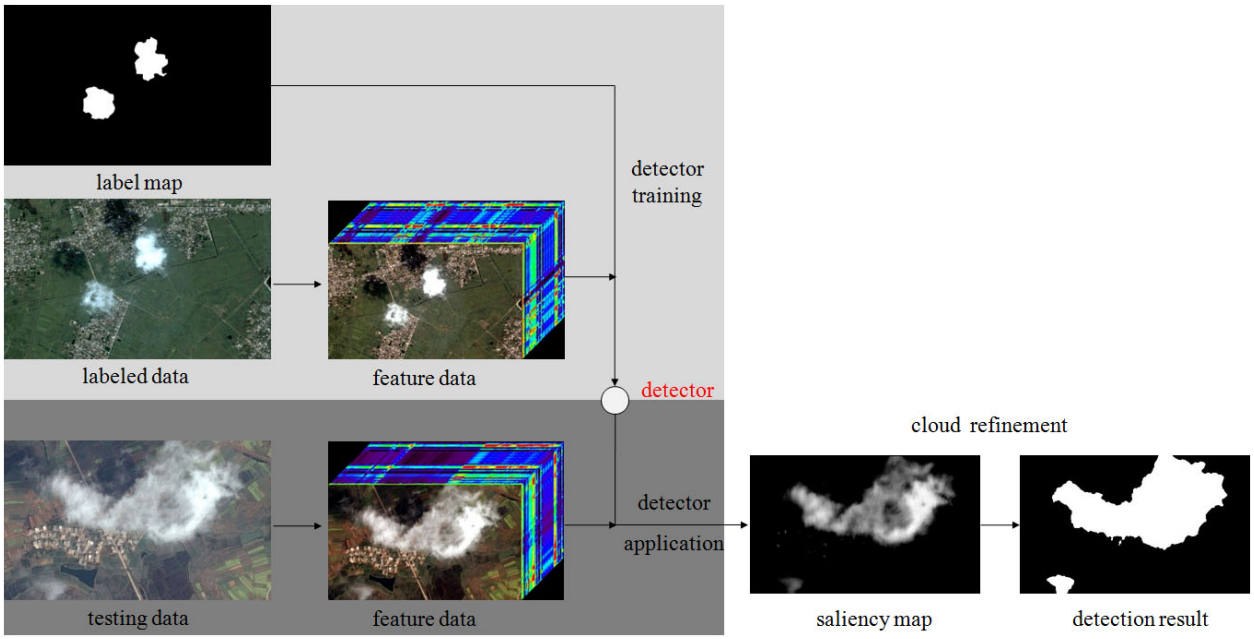

Fig. 1. Proposed processing chain for cloud detection. Feature data with high dimension is first simulated using labeled image and then applied to training a cloud detector with the corresponding label map. Clouds in the testing image could be finally detected using the detector along with some necessary refinement process.

on the different patches. This feature learning scheme, which takes the patch as a unit, could be named "patch learning" and it achieves successes in pedestrian detection [13] and face detection [15].

However, the above training scheme may face problems in cloud detection task for the following reasons: 1. Unlike conventional targets like pedestrians and faces, clouds in the image are usually polymorphic, which means clouds cannot be covered by the fixed size and shape. As illustrated in Fig. 2, clouds sometimes have so various shapes that they cannot even be covered by rectangles. Therefore, it brings difficulties for detecting clouds with different scales. 2. Cloud boundaries are also crucial for the detection task. However, in conventional patch learning scheme, researchers focused on finding the centers of targets and they usually do not clearly segment object boundaries. As a result, it is not approximate to apply conventional method to cloud detection in remote sensing images.

B. Scene learning scheme for cloud detection

To solve the above two problems of conventional "patch learning", we propose a new "scene learning" scheme. Scene learning aims at training and applying a cloud detector on the whole image scenes. The cloud detector herein is a special classifier that is used to separate clouds from the backgrounds. Concretely, scene learning regards each pixel of scenes in training image as a sample, and use them to train a cloud detector. Accordingly, the detecting process is also implemented on each pixel of testing image using the trained detector. Therefore, under the scene learning scheme, each pixel of image will be exploited as a sample, which could effectively avoid the problem caused by the small patches in traditional patch learning method. Generally, scene learning scheme contains two main modules: feature data simulating and cloud detector designing.

For cloud detection, four features including color information, statistical information, structural information, and texture information are extracted from the original image. All these features have the same spatial size, so they could be stacked into a new data. As shown in Figs. 3(b) and (e), the new data has a cubic structure, and each plane of the data is a fundamental feature of original image. Each pixel of such data could be represented by a column vector and each element of the vector is a feature value. So the data has the similar shape as spectra and named "feature data". We then label all the pixels of the input training image instead of different patches in conventional learning method, as shown in Fig. 3(c) (white pixels stand for clouds and black pixels stand for backgrounds). Therefore, we have a training data set of both positive and negative samples, and the number of the training data set is exactly the pixel number of the image.

All the pixels in the feature data and the labeled image will be applied for training a cloud detector. In the paper, a closed-form cloud detector is designed and it is also a column vector. To apply the detector for detecting clouds, we calculate the inner product of the testing feature vectors and the trained detector, thus obtaining a map. In the map, pixels of cloud areas have much higher gray levels than those of backgrounds, so the cloud are salient as shown in Fig. 3(f). The map will be named "saliency map" and will be quite helpful for segmenting clouds from the backgrounds. Using effective cloud refinement methods, we could finally locate clouds. So the core principle of the proposed method is to train a cloud detector in the whole scenes rather than only some patches with specific sizes. That is also the reason why we name the proposed learning scheme "scene learning" frame.

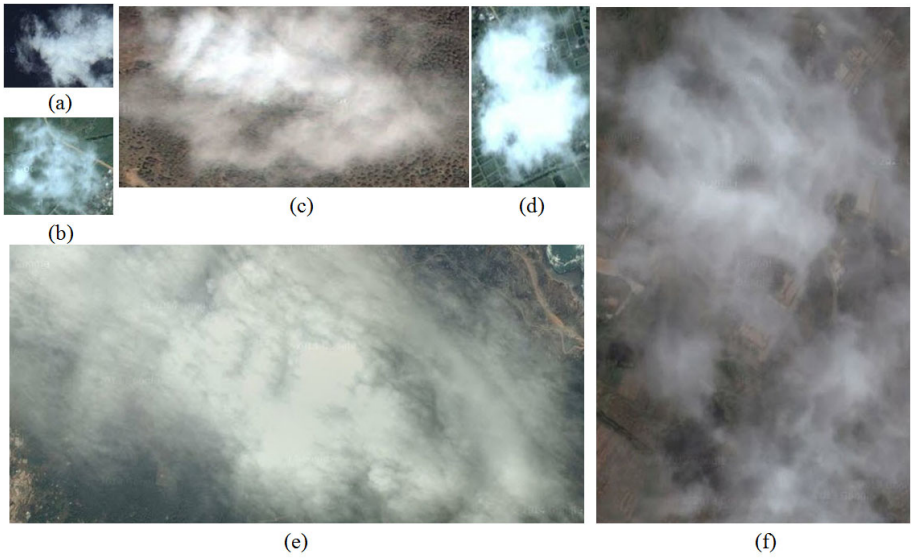

Fig. 2. Clouds in the real world. Clouds in (a),(b) and (d) are small while they are large in (c), (e) and (f), and they also have different shapes.

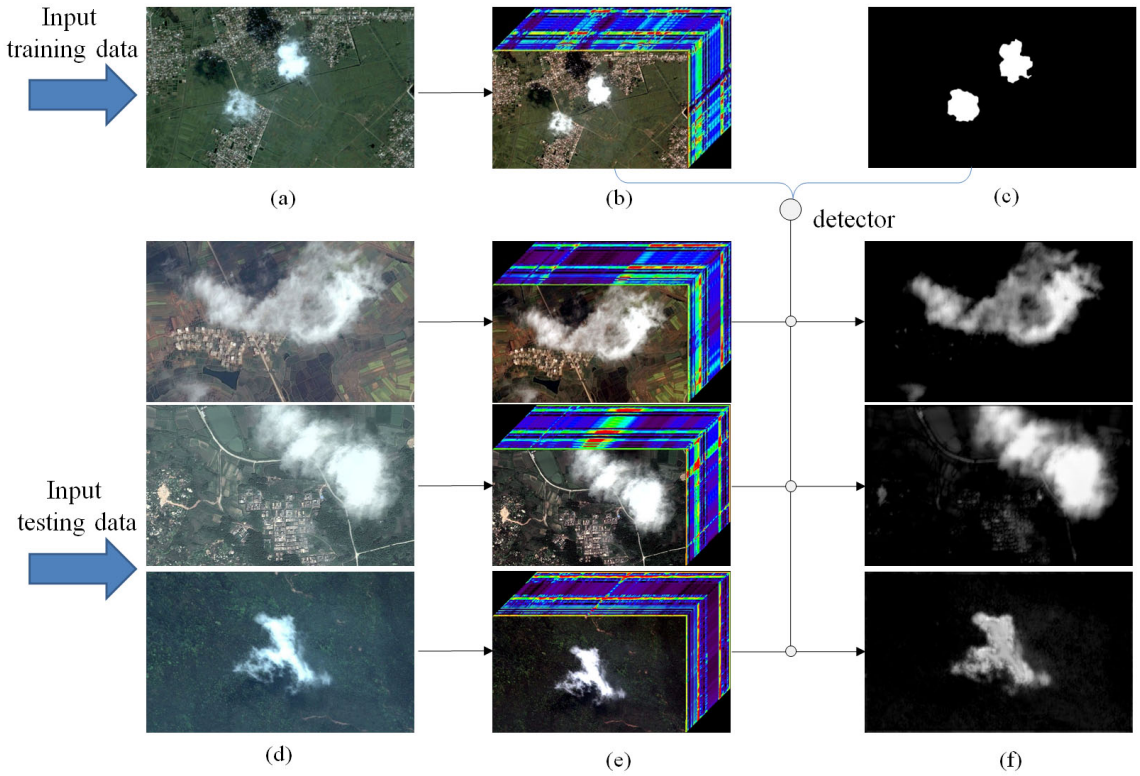

Fig. 3. Detector training and application. (a) is the labeled training data and (c) is its labeled map. (d) shows some input testing images. (f) illustrates the saliency maps using the trained detector. (b) and (e) show the simulated feature data.

Under the scene learning scheme, two main problems should be solved: 1. Feature selection for cloud detection. Simulating feature data requires different proper fundamental features. 2. Cloud detector designing. How to train a cloud detector is the key problem of the paper. In the next section, we will first mainly discuss the features of color, statistics, structure and texture, and the the cloud detector designing method.

III. Feature data simulating and cloud detector designing

A. Feature selection and feature data simulating

Feature data is simulated by stacking different features of input image. As stated above, we have selected four different features including color, statistics, structure and textures. Apparently, they are carefully chosen and they could represent different characteristics of image. Details of the extraction method of features are discussed as follows.

1) Color: Color plays an important role for cloud detection, as cloud has obviously different intensity (white) from that of backgrounds. Three channels in RGB color space, as well as the hue and saturation components in HSI space [16], [17] are extracted for each pixel, thus producing 5 color features. The transformation from RGB to HSI color space is expressed as follows:

$$\mathbf{H} = \begin{cases} \theta, & \mathbf{I}^b \leq \mathbf{I}^g \\ 360 - \theta, & \mathbf{I}^b > \mathbf{I}^g \end{cases} \tag{1}$$

$$\mathbf{S} = 1 - \frac{3 \times \min(\mathbf{I}^r, \mathbf{I}^g, \mathbf{I}^b)}{\mathbf{I}^r + \mathbf{I}^g + \mathbf{I}^b} \tag{2}$$

$$\widehat{\mathbf{I}} = (\mathbf{I}^r + \mathbf{I}^g + \mathbf{I}^b)/3 \tag{3}$$

$$\theta = \cos^{-1}\left\{ \frac{[(\mathbf{I}^r - \mathbf{I}^g) + (\mathbf{I}^r - \mathbf{I}^b)]/2}{\sqrt{(\mathbf{I}^r - \mathbf{I}^g)^2 + (\mathbf{I}^r - \mathbf{I}^b)(\mathbf{I}^g - \mathbf{I}^b)}} \right\} \tag{4}$$

where $\mathbf{I}^r$, $\mathbf{I}^g$ and $\mathbf{I}^b$ are respectively the red, green and blue channel of input image $\mathbf{I}$. $\mathbf{H}$, $\mathbf{S}$ and $\hat{\mathbf{I}}$ are respectively the hue, saturation and intensity components in HSI space. Each feature plane is normalized by subtracting its mean value over the entire image.

2) Local statistical information: Local statistical information is widely used in different image processing areas. It is not a special local measure for image but contains different matrices. For cloud detection task, one can observe two facts: 1. Cloud areas usually have higher intensity, as they have larger reflectivity than other regions. 2. The regions inside the clouds are generally smooth and have low intensity differences. Therefore, in the paper, two matrices will be applied to describe the statistical information to some extent. They are local mean value and local variance value, which could be respectively written as:

$$\mathbf{M}(i) = \frac{1}{W} \sum_{j \in \mathfrak{R}(i)}^{W} \mathbf{I}_j \tag{5}$$

$$\mathbf{V}(i) = \sqrt{\frac{1}{W} \sum_{j \in \mathfrak{R}(i)}^{W} (\mathbf{I}_j - \bar{\mathbf{I}})^2} \tag{6}$$

where $\mathbf{I}$ is the input image, $\mathbf{M}(i)$ and $\mathbf{V}(i)$ represent the mean and variance value of pixel i, $\mathfrak{R}(i)$ is a local window, W is the total pixel number in $\mathfrak{R}(i)$. In practical application, $\mathfrak{R}(i)$ is usually a square window with its center at pixel i. In our experiments, the window width of is $\mathfrak{R}(i)$ set to 3, 7, 11. The different widths help to obtain statistical results in different scales. For an image in RGB color space, we obtain 6 dimensions for each band (including mean value and variance value for three different window widths). Therefore, there will be 18 dimensions for the RGB image.

3) Texture: Texture provides us with information about the spatial arrangement of color or intensities in an image. It is a more descriptive concept than a precise mathematical definition. It contains quite a lot of different information in the image and could be categorized into two parts: the structured approach and the statistical approach, and the latter one will be adopted in the paper. Concretely, we use Gabor filter [18], [19], [20], a advantageous and wildly applied method in image texture and edge description, to obtain the textures..

Gabor filter could be written as:

$$g(x, y; \lambda, \theta, \sigma) = \exp\left(-\frac{x'^2 + y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda}\right)\right)$$

where

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned}$$

here, λ represents the wavelength of the sinusoidal function, θ represents the orientation, σ is the standard deviation of the Gaussian envelope. In our work, $\lambda = \{0.8, 1, 1.2\}$, $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and $\sigma = \{1, 1.5, 2, 2.5, 3, 3.5, 4\}$. Therefore, there are 84 responses for each pixel, and Fig. 4 shows part of filtering results of Fig. 3(a).

4) Structure: Compared with texture information, structure information of an image, the "primary data of human perception, not the individual details", is a higher level feature in feature selection and feature data simulating. J.-F. Aujol et al. regard images as "structure+texture" images [21], as they share the similarity that semantically meaningful structures are formed by texture elements. Thus, if well separated, image structure could provide researchers with the core information about the image. Since information inside the cloud regions is much less than its structural information, the structural analysis will benefit the cloud detection. To exploit structure image $\mathbf{S}$ for an input image $\mathbf{I}$, a relative total variation model will be employed, which could be mathematically written as:

$$\mathbf{S} = \arg\min_{\mathbf{S}} \sum_{i=1}^{N} (\mathbf{S}_i - \mathbf{I}_i)^2 + \lambda \left(\frac{\Phi_x(i)}{\Psi_x(i) + \varepsilon} + \frac{\Phi_y(i)}{\Psi_y(i) + \varepsilon} \right) \tag{7}$$

where ε is a small constant, N is the total number of image, λ is a pre-setting parameter for balance. $\Phi_x(i)$ and $\Phi_y(i)$ are the general pixel-wise windowed total variation measure. They represent the absolute spatial difference within the window $R(i)$ and could be written as:

$$\begin{aligned} \Phi_x(i) &= \sum_{j \in R(i)} g_{i,j} |(\partial_x \mathbf{S})_j| \\ \Phi_y(i) &= \sum_{j \in R(i)} g_{i,j} |(\partial_y \mathbf{S})_j| \end{aligned} \tag{8}$$

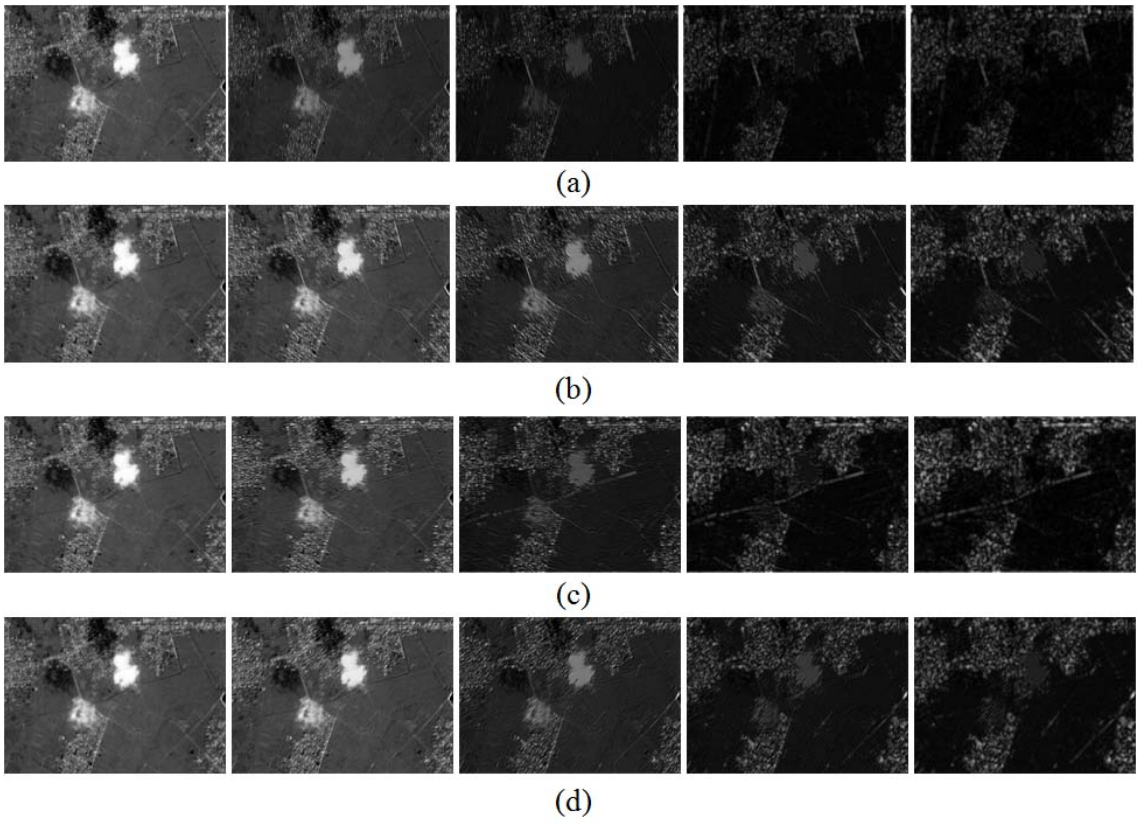

Fig. 4. Part of filtering maps of Fig. 3(c) using Gabor filter. Parameter $\lambda = 0.8$ is used for all the maps. From top to the bottom, $\theta = \{0°, 45°, 90°, 135°\}$, respectively. From left to the right, $\sigma = \{1, 1.5, 2, 2.5, 3\}$, respectively.

where j belongs to a window $R(i)$, $(\partial_x \mathbf{S})$ and $(\partial_y \mathbf{S})$ respectively calculate the partial derivative in x and y directions of image $\mathbf{S}$. $g_{i,j}$ is a weighting function, and it is defined as

$$g_{i,j} = \exp\left(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma^2}\right) \tag{9}$$

$\Psi_x(i)$ and $\Psi_y(i)$ is defined different from $\Phi_x(i)$ and $\Phi_y(i)$, they are written as:

$$\begin{aligned} \Psi_x(i) &= \sum_{j \in R(i)} |g_{i,j}(\partial_x \mathbf{S})_j| \\ \Psi_y(i) &= \sum_{j \in R(i)} |g_{i,j}(\partial_y \mathbf{S})_j| \end{aligned} \tag{10}$$

According to the L. Xu et al. belief [22], equation (8) and equation (10) respectively form the windowed total variation map and windowed inherent variation map. Detail and texture information are visually salient in windowed total variation maps, while they are indistinctive in inherent variation map. Their combination map, namely, the expression $\left(\frac{\Phi_x(i)}{\Psi_x(i)+\varepsilon} + \frac{\Phi_y(i)}{\Psi_y(i)+\varepsilon}\right)$ is named relative total variation (RTV) map. In RTV map, meaningful structures are penalized much less than textures, so it makes main structures stand out. Thus, the optimization problem (7) could extract the structure map $\mathbf{S}$. More details about RTV model could be seen in the work [22]. In our work, we set the parameter $\lambda = \{0.0005, 0.001, 0.0015\}$, so we obtain 9 structure maps.

All the above 116 feature maps are stacked vertically to form feature data, including color, statistical information, texture and structure features. A group of parameters have been carefully tuned and provided in the above discussions, and they could be directly applied to cloud detection. In the next section, details on detector designing and application are provided.

B. Cloud detector designing

In the section, we design a cloud detector using the above simulated feature data. The detector has a closed-form by minimizing the residual error between the data and its labels. Details of the process could be mathematically analyzed as follows.

Given a training data set of N points:

$$D = \{(\mathbf{x}_1, z_1), (\mathbf{x}_2, z_2), ..., (\mathbf{x}_N, z_N)\} \tag{11}$$

$\mathbf{x}_i \in \mathbb{R}^{L \times 1}$ is the observed data (L is the data dimension) and $z_i \in \{0, 1\}_{i=1}^N$ is label for data i. $\mathbf{x}_i$ belongs to cloud if the label $z_i = 1$, otherwise $z_i = 0$. Then a detector $\mathbf{w}^{L \times 1}$ is expected to be obtained to distinguish the clouds from the backgrounds, and it could be written as:

$$\mathbf{w}^T \mathbf{x}_i = z_i, \quad i = 1, ..., N \tag{12}$$

In such circumstance, the residual error (also named empirical risk) $\epsilon(\mathbf{w})$ could be defined as

$$\epsilon(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} 1_z\{\mathbf{w}^T \mathbf{x}_i \neq z_i\} \tag{13}$$

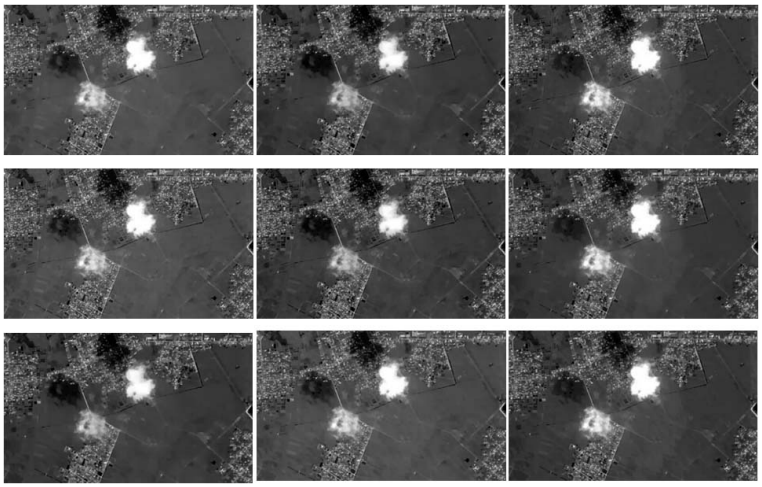

Fig. 5. Structure maps of Fig. 3(c). Parameters λ for first and second rows are respectively 0.0005, 0.001 and 0.0015. From left to the right are the red, green and blue bands, respectively.

where 1_z is an indicator function. Obviously, $\mathbf{w}$ could be obtained by minimizing the error. In equation (12), there are N equations and usually $N \gg L$, so they form overdetermined equations. To calculate $\mathbf{w}$, the least square method is applied. Then the original N equations could be re-grouped with a quadratic form.

$$\mathbf{w} = \arg\min J(\mathbf{w}) = \arg\min \frac{1}{2N} \sum_{i=1}^{N} \left(\mathbf{w}^T \mathbf{x}_i - z_i\right)^2 \quad (14)$$

here, $J(\mathbf{w})$ could be written using an expectation form.

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{E}((\mathbf{w}^T \mathbf{x}_i - z_i)^2) \quad (15)$$

where $\frac{1}{2}\mathbf{E}((\mathbf{w}^T \mathbf{x}_i - z_i)^2) = \frac{1}{2N} \sum_{i=1}^{N} \left(\mathbf{w}^T \mathbf{x}_i - z_i\right)^2$. According to the Karush-Kuhn-Tucker condition [23], we have:

$$\begin{aligned} \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= \frac{1}{2} \frac{\partial \mathbf{E}(\mathbf{w}^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{w} - 2\mathbf{w}^T \mathbf{x}_i z_i + {z_i}^2)}{\partial \mathbf{w}} \\ &= \mathbf{E}(\mathbf{x}_i \mathbf{x}_i^T)\mathbf{w} - \mathbf{E}(\mathbf{x}_i z_i) = 0 \end{aligned} \quad (16)$$

Finally, we have

$$\mathbf{w} = \mathbf{C}^{-1}\mathbf{d} \quad (17)$$

where $\mathbf{C} = \mathbf{E}(\mathbf{x}_i \mathbf{x}_i^T)$ and $\mathbf{d} = \mathbf{E}(\mathbf{x}_i z_i)$. Obviously, vector $\mathbf{w}$ is exactly the detector we need, and it possesses some advantages as follows.

Remark 1: This detector has a closed-form solution, and it is easy to be implemented. To solve the problem that the original residual error is the summation of indicator functions, a quadratic form is applied instead. The quadratic form could effectively approximate the original problem while providing a close-form solution, which will be quite convenient to be implemented in practical application.

Remark 2: To find clouds in a testing image, we should simulate feature data using the same features as in the training process. For each pixel $\mathbf{x}$ of the data, the inner product between $\mathbf{x}$ and the detector vector $\mathbf{w}$ could be calculated using the equation

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \quad (18)$$

The projected value $y(\mathbf{x})$ is large if the input pixel is in cloudy area, otherwise, the value is small. We could obtain projected results for all the pixels, and they form a new map as illustrated in Fig. 3(f). In the map, pixels of cloudy areas have much higher gray levels than those in backgrounds. Therefore, cloudy areas in the map are salient. The map is thus named "saliency map". In such circumstance, simple thresholding method could help to determine the cloudy area in the image. Details will be discussed in the next section.

Remark 3: Both $\mathbf{C}$ and $\mathbf{d}$, the two core factors of detector $\mathbf{w}$, have clear physical meanings. For $\mathbf{d}$, it is the mathematical expectation of multiplication between data $\mathbf{x}_i$ and its label z_i. Since the label z_i has only two possible value, namely, 0 and 1, $\mathbf{d}$ is in proportional to the mean value of the positive samples. Assume $\hat{D} = \{(\hat{\mathbf{x}}_1, 1), (\hat{\mathbf{x}}_2, 1), ..., (\hat{\mathbf{x}}_{N'}, 1)\} \subseteq D$ is the positive set of the training set with N' points, then we have $\mathbf{d} = \mathbf{E}\{\mathbf{x}_i z_i\} = \frac{N'}{N}\mathbf{E}\{\hat{\mathbf{x}}_i\}$. For $\mathbf{C}$, it is the mean value of sample correlation matrix without removing the samples' means. Physically, $\mathbf{C}$ synthesizes the energy of backgrounds and targets. With the above knowledge of $\mathbf{C}$ and $\mathbf{d}$, the physical meaning of detector $\mathbf{w}$ can be well understood. Inverse of $\mathbf{C}$ suppresses all the scenes of image, including backgrounds and foregrounds. However, by multiplying $\mathbf{d}$, the objects are effectively enhanced, thus making the clouds be outstanding in the saliency map.

Remark 4: The closed-form solution is easy to be extended for new input samples. Assume we have calculated $\mathbf{C}$ and $\mathbf{d}$ for N points. For a new input sample $\{\tilde{\mathbf{x}}, \tilde{z}\}$, to calculate the new $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{d}}$, it is not necessary to calculate all the elements again. We could only calculate the covariance matrix $\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T$ and $\mathbf{x}\tilde{z}$, then add them to the original $\tilde{\mathbf{C}}$ and $\mathbf{d}$

with a proper proportion $\frac{1}{N+1}$. So we have:

$$\tilde{\mathbf{C}} = \frac{\mathbf{C}\times N+\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T}{N+1} \quad (19)$$

$$\tilde{\mathbf{d}} = \frac{\mathbf{d}\times N+\tilde{\mathbf{x}}\tilde{z}}{N+1} \quad (20)$$

The calculation could be recurrently implemented until all the new samples are involved. Therefore, the process of calculating $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{d}}$ is uncorrelated to original specifical samples, and it could reduce a large amount of unnecessary calculations.

C. Influence of feature number

To analyze how the feature number affects the performance of the detector, we should theoretically calculate the residual error of the proposed detector in the first place.

We begin with calculating the quadratic residual error $J(\mathbf{w})$ in equation (15). So we have:

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}\mathbf{E}((\mathbf{w}^T\mathbf{x}_i - z_i)^2) \\ &= \frac{1}{2}\mathbf{E}(\mathbf{w}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w} - 2\mathbf{w}^T\mathbf{x}_i z_i + {z_i}^2) \\ &= \frac{1}{2}(\mathbf{E}({z_i}^2) + \mathbf{E}(\mathbf{w}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{w} - 2\mathbf{w}^T\mathbf{x}_i z_i)) \\ &= \frac{1}{2}(\mathbf{E}({z_i}^2) + \mathbf{w}^T\mathbf{E}(\mathbf{x}_i\mathbf{x}_i^T)\mathbf{w} - 2\mathbf{w}^T\mathbf{E}(\mathbf{x}_i z_i)) \quad (21) \\ &= \frac{1}{2}(\mathbf{E}({z_i}^2) + \mathbf{w}^T\mathbf{C}\mathbf{w} - 2\mathbf{w}^T\mathbf{E}(\mathbf{x}_i z_i)) \end{aligned}$$

Substitute (17) into (21), we have

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{2}(\mathbf{E}({z_i}^2) + (\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i))^T\mathbf{C}\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i) \\ &- 2(\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i))^T\mathbf{E}(\mathbf{x}_i z_i)) \\ &= \frac{1}{2}(\mathbf{E}({z_i}^2) + (\mathbf{E}(\mathbf{x}_i z_i))^T\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i) \\ &- 2(\mathbf{E}(\mathbf{x}_i z_i))^T\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i)) \\ &= \frac{1}{2}(\mathbf{E}({z_i}^2) - (\mathbf{E}(\mathbf{x}_i z_i))^T\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i)) \\ &= \frac{1}{2}(\mathbf{E}(z_i) - (\mathbf{E}(\mathbf{x}_i z_i))^T\mathbf{C}^{-1}\mathbf{E}(\mathbf{x}_i z_i)) \quad (22) \end{aligned}$$

With the assumption that $\mathbf{C} = \mathbf{E}(\mathbf{x}_i\mathbf{x}_i^T)$ and $\mathbf{d} = \mathbf{E}(\mathbf{x}_i z_i)$ as in the Section II, the above equation could be written as:

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{E}(z_i) - \mathbf{d}^T\mathbf{C}^{-1}\mathbf{d}), i = 1, 2, ..., N \quad (23)$$

Thus, we finally obtain the residual error. Note that, the residual error in (23) is obtained when the feature dimension is fixed. However, in the proposed scene learning scheme, different features will be added for cloud detection, so how the new features affect the performance of detector $\mathbf{w}$ is an important issue, and it will be further discussed.

For simplicity, we first explore the influence if one more dimension is introduced. In such circumstance, the original L-dimensional data in (11) becomes a $(L+1)$-dimensional data, and the spectral dimension could be represented as $\Phi = \{1, ..., L, L+1\}$. For the i-th pixel, $\mathbf{x}_i$ is its vector in the original space. Assume $x_{i,L+1}$ is its value in $(L+1)$-th dimension, then the i-th pixel in the new space could be represented as

$$\mathbf{x}_{i,\Phi} = \begin{bmatrix} \mathbf{x}_i \\ x_{i,L+1} \end{bmatrix}$$

Similar to the result in equation (23), in the new space, the residual error could be represented as:

$$J(\Phi) = \frac{1}{2}(\mathbf{E}(z_i) - \mathbf{d}_\Phi^T\mathbf{C}_\Phi^{-1}\mathbf{d}_\Phi) \quad (24)$$

where $\mathbf{C}_\Phi = \mathbf{E}(\mathbf{x}_{i,\Phi}{\mathbf{x}_{i,\Phi}}^T)$ and $\mathbf{d}_\Phi = \mathrm{E}(\mathbf{x}_{i,\Phi}z_i)$. Then an important theorem in describing the influence of feature dimension is written as follows.

Theorem 1 (Influence of feature number)**.** *The residual error $J(\mathbf{w})$ in L feature dimensions is not smaller than the residual error J_Φ in $L+1$ feature dimension if $s_{L+1} - \mathbf{s}^T\mathbf{C}^{-1}\mathbf{s} \neq 0$. Mathematically, the following inequality holds:*

$$J(\mathbf{w}) - J(\Phi) = \frac{1}{2}\frac{(d_{L+1} - \mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2}{s_{L+1} - \mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}} \geq 0 \quad (25)$$

where $\mathbf{s} = \mathbf{E}(\mathbf{x}_i x_{i,L+1})$, $d_{L+1} = \mathbf{E}(x_{i,L+1}z_i)$ and $s_{L+1} = \mathbf{E}(x_{i,L+1}^2)$.

Proof for the Theorem 1 could be seen in Appendix.

Remark 1: One can obviously conclude from Theorem 1 that the residual error is reduced by $\frac{1}{2}\frac{(d_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2}{s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}}$ if a new feature number is added into the feature data. Once the feature number of feature data is gained, the proposed scene learning scheme performs better if the condition $d_{L+1} - \mathbf{s}^T\mathbf{C}^{-1}\mathbf{d} \neq 0$ holds. It explains why we use four features in total for cloud detection rather than a single feature.

Remark 2: If $d_{L+1} - \mathbf{s}^T\mathbf{C}^{-1}\mathbf{d} = 0$ holds, then the new added feature could not improve the performance of detector. It indicates that too many features may only cause computational problem without improving the performance of proposed method. In our study, four different features are extracted to comprise the feature data. These features represent most fundamental image features, so we do not add more features.

IV. Cloud refinement

As stated above, clouds are usually well separated from the backgrounds in the saliency map. To further confirm the cloud boundaries and eliminate the tiny noise, refinement step will be applied in the section.

A. Cloud segmenting via optimal thresholding method

Since clouds have been well extracted from the backgrounds, histogram of saliency maps is usually bimodal distribution. Fig. 6 shows an example of saliency map and its histogram. Fig. 6(a) is the original color image, Fig. 6(b) is the saliency map and Fig. 6(c) is its histogram. From the histogram, one can see that one peak usually appears at a low gray level, and it represents the cluster of the backgrounds, as the H1 area shows; the other appears at a high gray level, which represents the cluster of clouds, as the H2 area shows. To segment such

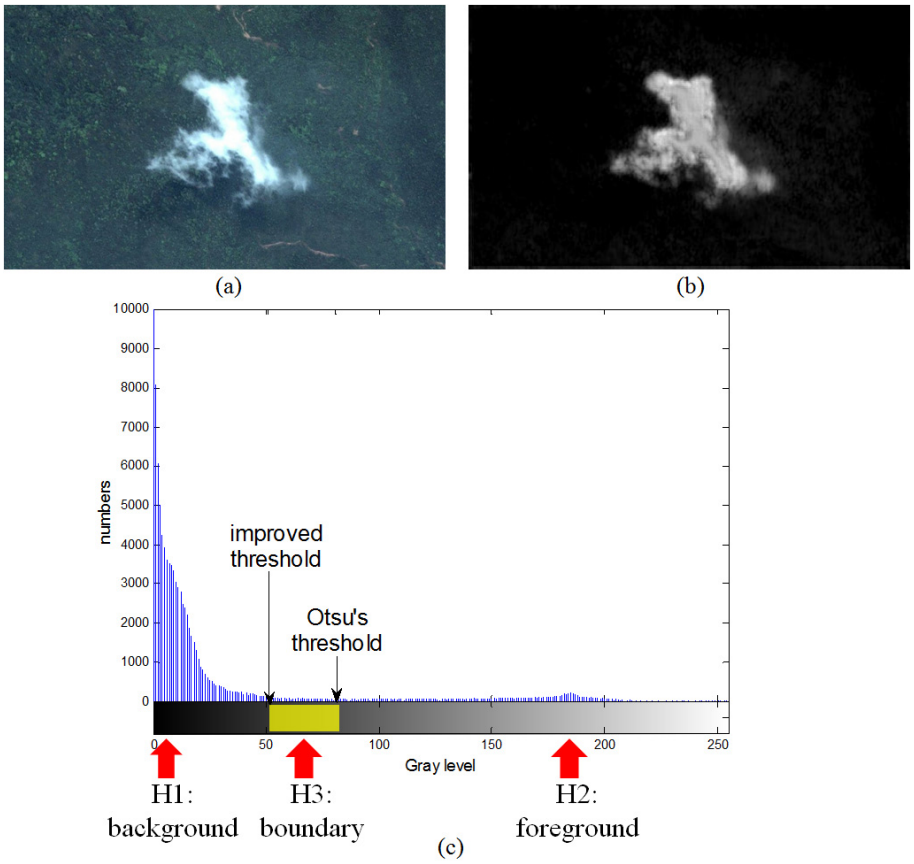

Fig. 6. Illustration of saliency map and histogram of an image. (a) is the original color image, (b) is the candidate map and (c) is the histogram of the map.

an image with bimodal distribution, Otsu's method [24] seems to be a proper choice.

Otsu's method assumes that histogram of image is bimodal (like the saliency map). It calculates a threshold value, which could separate the input image $\mathbf{G}$ into two classes—-the foreground and background, and their interclass variance is maximum. Mathematically, the Otsu's threshold value t should satisfy the optimization problem:

$$t = \arg\max_{t}\{\omega_0(\mu_0 - \overline{\mu})^2 + \omega_1(\mu_1 - \overline{\mu})^2\} \tag{26}$$

where $\omega_0 = N_0/N, \omega_1 = N_1/N, \mu_0 = \sum_{i=1}^{t} i \cdot p_i, \mu_1 = \sum_{i=t}^{256} i \cdot p_i, \overline{\mu} = \omega_0\mu_0 + \omega_1\mu_1$. Here, N_0, N_1 and N are respectively the foreground, background, and total pixel numbers. p_i is the frequency of gray level i. μ_0, μ_1 and $\overline{\mu}$ respectively represent mean gray values of the foregrounds, backgrounds and whole image. For cloud detection in RGB color image, t will be an value between 1 and 256. Traversal strategy is adopted to obtain the result. Therefore, we could obtain a segmentation map R^{seg} using the Otsu's threshold, and it is written as:

$$R^{seg}_{(i)} = \begin{cases} 1, & \mathbf{G}_{(i)} \geq t \\ 0, & otherwise \end{cases} \tag{27}$$

where t is the calculated threshold value using Otsu's method. Here, cloud pixels are all labeled with value 1, and background pixels are labeled with 0. As marked in the histogram (it is 82) of Fig. 6(c), the threshold of Fig. 6(b) can be calculated by using the Otsus method. From the result of the corresponding binary map shows in Fig.7(a), we see that the center area of clouds are located. However, compared with the reference map in Fig. 7(d), the cloud boundaries are eliminated, as marked in the red circle. It implies that the original Otsu's threshold is needed to be revised for cloud detection task.

In the paper, an optimal thresholding method is proposed based on revising the Otsu's value t. Note that cloud boundaries have lower intensities than pixels in the center part of clouds. Therefore, to find the eliminated cloud boundaries, we should reduce the Otsu's value t. Assume the revised threshold value is $\hat{t}$, then the following two facts about the histogram in Fig. 6(c) will be exploited in finding the proper $\hat{t}$: 1. The gray levels within the yellow rectangle have the similar frequency and the variance of the frequency is low enough. 2. The variance will be dramatically larger if $\hat{t}$ is smaller. Based on the two facts, traversal strategy is also applied to find $\hat{t}$. We traverse pixel levels from the Otsu's threshold t to the minimum pixel level (it is 1 for RGB color image), until the variance of histogram frequency between the t and the present level $\hat{t}$ is larger than a pre-setting constant.

It is worth mentioning that, although only one map (with simple background) is shown, most saliency maps have the similar histogram as illustrated in Fig. 6(c). They also have the bimodal distribution and the similar cloud boundary property,

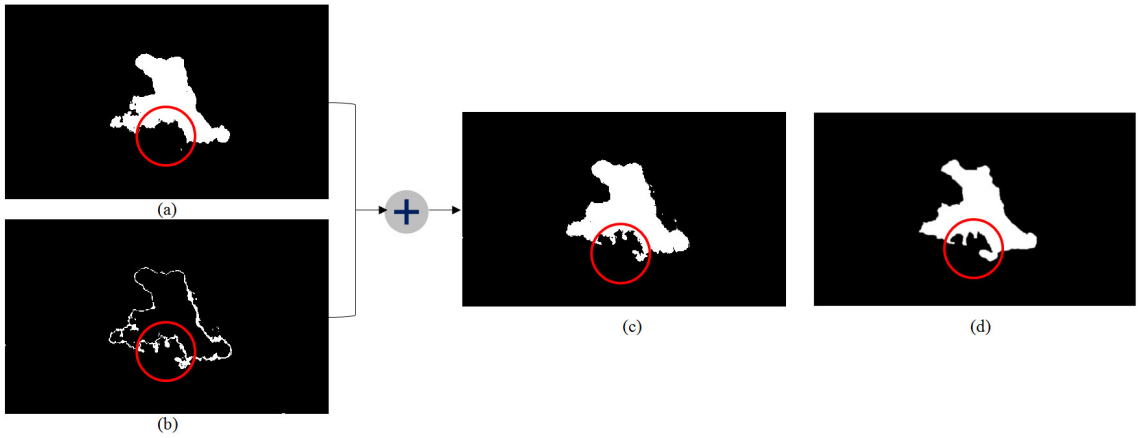

Fig. 7. Illustration for the improved Otsu's threshold. (a) is the binary map using the original Otsu's threshold. (b) shows the boundaries of clouds. (c) is the binary map using the improved Otsu's threshold. (d) is the reference map.

therefore, the above optimal thresholding method could be widely applied. Pseudocode of the whole process for searching optimal threshold is shown as in Algorithm 1. In the algorithm, $\mathbf{h}(\hat{t}:t)$ represents the $\hat{t}$-th to t-th elements of $\mathbf{h}$.

Algorithm 1 Optimal threshold approximation.

1: Input saliency map $\mathbf{G}$, and its gray levels range from 1 to 256.
2: Obtain the Otsu's threshold t via solving the problem (26).
3: Obtain the gray histogram vector $\mathbf{h}^{256\times 1}$ of the image $\mathbf{G}$. Each element $\mathbf{h}(i)$ represents the total number of gray level i in $\mathbf{G}$.
4: Initialize a constant parameter $v_0 = 400$, and two variables $v_1 = 0$, $\hat{t} = t$.
5: **while** ($v_0 > v_1$ AND $\hat{t} > 0$) **do**
6: $\hat{\mathbf{h}} = \mathbf{h}(\hat{t}:t)$, $num = t - \hat{t} + 1$.

$$\overline{\mathbf{h}} = \frac{\sum_{l=1}^{num} \hat{\mathbf{h}}(l)}{num} \tag{28}$$

$$v_1 = \frac{\sum_{l=1}^{num} (\hat{\mathbf{h}}(l) - \overline{\mathbf{h}})^2}{num} \tag{29}$$

7: **if** $v_0 > v_1$ **then**
8: $\hat{t} \leftarrow \hat{t} - 1$
9: **else**
10: **Break**;
11: **end if**
12: **end while**
13: Output the optimal threshold $\hat{t}$.

We calculate the optimal threshold $\hat{t}$ (it is 52) of Fig. 6(b) using the proposed method, and mark it in Fig. 6(c). One can observe the fact that, in the histogram, there are some gray levels between the original Otsu's threshold t and the threshold $\hat{t}$, as the yellow rectangle shows. We threshold the saliency map and set the pixel value as 1 if its gray value is between $\hat{t}$ and t, and 0 otherwise. Fig. 7(b) shows the binarized map and we see that it actually represents the boundaries that have been removed in Fig. 7(a). It indicates that the revised optimal threshold $\hat{t}$ is more appropriate for detecting clouds and the saliency map could be binarized into $\hat{R}^{seg}$ with $\hat{t}$ using the equation (30).

$$\hat{R}^{seg} = \begin{cases} 1, & \mathbf{G}(i) \geq \hat{t} \\ 0, & otherwise \end{cases} \tag{30}$$

Fig. 7(c) shows the binary map using the improved threshold. Compared with Fig. 7(a), Fig. 7(c) preserves more boundaries, and it is more similar to the reference map as shown in Fig. 7(d). More thresholding maps could be seen in Fig. 8(a) - (f), and cloud boundaries are also preserved.

B. Detail refinement

As shown in the binary maps in Fig. 7(c) and Fig. 8(a) - (f), the cloud areas can be generally obtained, and the boundary information of cloud is more accurate compared with the detection result by using Otsu's threshold. However, three problems should be solved before the detection task is finally accomplished: 1. Cloud isolation. As shown in the red circle of Fig. 8(d), the joint area between the two parts of one cloud may be isolated because of the thin cloud in these areas. Therefore, we should first close the gaps and make the isolated parts connected. 2. Tiny interference removal. Although the proposed algorithm suppresses backgrounds, some interferences still exist as shown in Fig. 8(e), and they should be excluded before we obtain the final results. 3. Internal holes filling. As shown in Fig. 8(f), some thin cloud may appear in the internal regions, and it may lead to the appearance of holes which should be filled because they are also covered by clouds.

To address the above problems, three image processing approaches will be implemented. First, morphological close operator [25], [26] is used to implement on the binary map $\hat{R}^{seg}$. The close operator is comprised of two fundamental morphological operators: image dilation and corrosion. Note that, corrosion operator is applied after dilation operator. So the close operator could connect the neighboring objects, filling small holes and smooth the boundaries without changing the shape much. In the cloud detection task, a mask of disk shape with radius 4 is applied. In Fig. 8(g), we see that the two

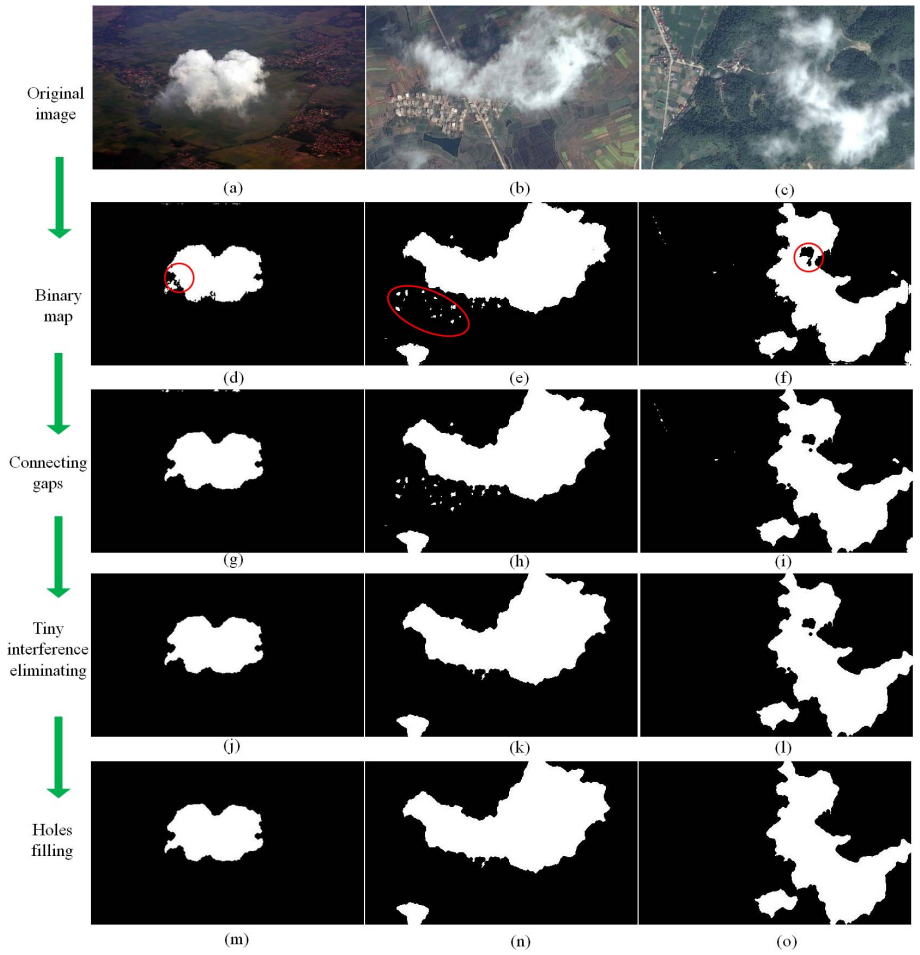

Fig. 8. Cloud detail refinement process. (a) - (c) are three color images. (d) - (f) are their binary maps with gaps, tiny interference objects, and internal hole, respectively. (g) - (i) are the binary maps after using morphological close operator. (j) - (l) are the binary maps after removing tiny objects. (m) - (o) are the final detection results after filling internal holes.

parts in the red circle are connected. Moreover, the boundaries in Fig. 8(g) - (i) are all smoothed to some extent. Second, to remove the tiny interferences, the similar method in the paper [1] is applied. Connected regions are first extracted from the binary map. For each region, its pixel number is counted. The region will be classified into interference if the number is smaller than a given constant. Otherwise it is cloud. Fig. 8(j) - (l) present the results after removing tiny objects. Third, to fill the holes of each region, the function $imfill()$ [27] in MATLAB will be applied. This is an effective function that performs a filling operation on the input binary image. Fig. 8(m) - (o) are the binary images after filling holes. Fig. 8(m) - (n) do not change compared with Fig. 8(j) - (l), as they do not have internal holes. However, we see that the hole in Fig. 8(l) is filled as shown in Fig. 8(o). In conclusion, the above three procedures are sequentially implemented on the binary map $\hat{G}^{seg}$, and the final cloud detection results are then obtained as in Fig. 8(m) - (o).

V. Numerical Experiments

In this section, we report results of experiments, aiming at evaluating the performance of the proposed method. Comparisons between the proposed method and the method in [1] are implemented. Besides, as the work in the paper [1] did, some popular automatic image segmentation methods including K-means [28], [29], Mean-shift [30], [31], [32] and Chan-Vese [33], [34], [35], are also used to evaluate the performance because of the their close relationships. These segmentation methods are implemented on the original RGB color image. Executable program of method [1] could be available online http://graphvision.whu.edu.cn/. MATLAB codes for K-means, Mean-shift and Chan-Vese are respectively available at http://www.mathworks.cn/cn/help/stats/kmeans.html, http://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering and http://www.ipol.im/pub/art/2012/g-cv/. Both visual comparisons and quantitative evaluations are implemented to demonstrate the efficacy of the proposed method. MATLAB codes are all implemented in Mathworks MATLAB R2009b and a desktop computer equipped with Intel(R) Core(TM) 2 Duo 2.80GHz CPU with 4GB memory. Some of the original RGB color images are obtained by the satellite Quickbird http://glcf.umd.edu/data/quickbird/ and their spatial resolution is about 2.44-2.88 meters. Some color images are from the Flickr.com. The others are obtained

from the Google map [1], with the size of 1000 × 600 pixels and spatial resolution of 1 meter. Ground truth of the images are all manually marked since it is impossible to field survey the cloud boundaries. We first label the map using the popular software Environment for Visualizing Images (ENVI) individually, so we obtain several ground truth for one image, and then using their average as the final ground truth.

Four metrics are used to quantitatively assess the algorithms. They are the right rate (RR), error rate (ER), false alarm rate (FAR) and ratio of right rate to error rate (RER). RR is defined as:

$$RR = \frac{CC}{GN} \tag{31}$$

where CC is the number of correctly detected cloud pixels, GN is the number of cloud pixels in ground truth.

ER is defined with the same form as in the paper [1]:

$$ER = \frac{CN + NC}{TN} \tag{32}$$

where CN represents the number of cloud pixels identified as noncloud pixels, NC represents the number of noncloud pixels identified as cloud pixels, and TN denotes the number of pixels of the input image.

FAR is defined with the same form as in the papers [36] and [37]:

$$FAR = \frac{NC}{TN} \tag{33}$$

where NC and TN have the same meanings as in equation (32).

RER is defined as the ratio of RR to ER.

$$RER = \frac{RR}{ER} \tag{34}$$

From the definition, one can see that RR is used to provide us with information of correctly detected results, while ER and FAR are used to provide incorrect information. The reference values for RR, ER and FAR are respectively 1, 0 and 0. FAR is one part of ER and it explicitly represents the false alarm rate. Using only one of them to assess algorithms is insufficient, as some methods may obtain high right rate but bring too many false alarms. On the contrary, some methods may obtain low error rate but also low right rate. Therefore, RER is defined to obtain an integrated result as it considers the right rate and error rate. The higher it is, the better.

A. Visual analysis

In the experiments, only one image will be used to train the detector, and it is shown in Fig. 3(a). To make the comparisons be fair for other methods, this training image will not be used as the testing image. 38 images in total are used as testing images. To save space, we display only five groups of detection results using different algorithms. Fig. 9(a) shows the original RGB color images. Clouds in these five images have quite different shapes and thicknesses. They are thick in the left three images, and semitransparent in the right two images. Fig. 9(b) is the ground truth images. Figs. 9(c) - (g) are the detection results using K-means, Mean-shift, Chan-Vese, method in [1] and the proposed method, respectively.

All the three conventional segmentation methods could automatically divide the images into two categories: backgrounds and foregrounds. We then calculate their mean value and take the classification with higher values as the clouds, because clouds usually possess much higher brightness. One can observe that conventional segmentation methods, the method in [1] and our method could generally obtain clouds if clouds and backgrounds have high contrast, as shown in the first, second and third columns of Fig. 9. However, segmentation methods face the problem in accurately finding the cloud boundaries because of the different intensity in these areas. Mean-shift tends to miss some clouds while K-means and Chan-Vese tend to obtain some interferences. Besides, all segmentation methods fail to eliminating interferences in the third images, there the brightness of backgrounds and clouds are similar. Method of [1] performs better than segmentation methods but also could not precisely locate the boundaries (see the first and second columns of Fig. 9(f)).

The fourth column images show a challenging case in cloud detection since the existence of snows. In such circumstance, all the three segmentation methods obtain false results and mistake snows as clouds. Method of [1] performs better but also brings false alarms. On the other hand, since our method exploits different image features and obtains a salience map, where snows and clouds have different details, satisfactory result is obtained as shown in Fig. 9(g). Meanwhile, since our method adopts the optimized threshold, cloud boundaries are also better located compared with existing methods. The fifth column images illustrate an extreme case, where the brightness of clouds and backgrounds are quite similar. K-means, Chan-Vese and method of [1] tend to obtain more false alarms, while Mean-shift tends to miss some clouds. Although our method also could not obtain ideal result, it obtains the result that is closest to the ground truth.

B. Quantitative analysis

TABLE I
CALCULATED RR, ER, FAR AND RER USING DIFFERENT METHODS

	K-means	Mean-shift	Chan-Vese	method of [1]	proposed method
RR	0.6884	0.5492	**0.9584**	0.7890	0.8477
ER	0.0820	0.1184	0.3636	0.0773	**0.0458**
FAR	0.0528	0.0553	0.2472	0.0289	**0.0139**
RER	16.4407	7.5397	6.0220	17.4380	**23.0686**

Four quantitative measures, namely, RR, ER, FAR and RER are applied to evaluate the different methods. All the average values for the 38 testing images are tabulated in Table. I. Indices with bold types represent the best ones in the same row.

Segmentation methods could not obtain satisfactory results as the results of RR, ER and FAR are not simultaneously the best. For example, Chan-Vese performs well in index RR but it has too high ER and FAR, which implies that it obtains more clouds but bringing more false alarms. On the contrary,

[1] Available online: https://maps.google.com/

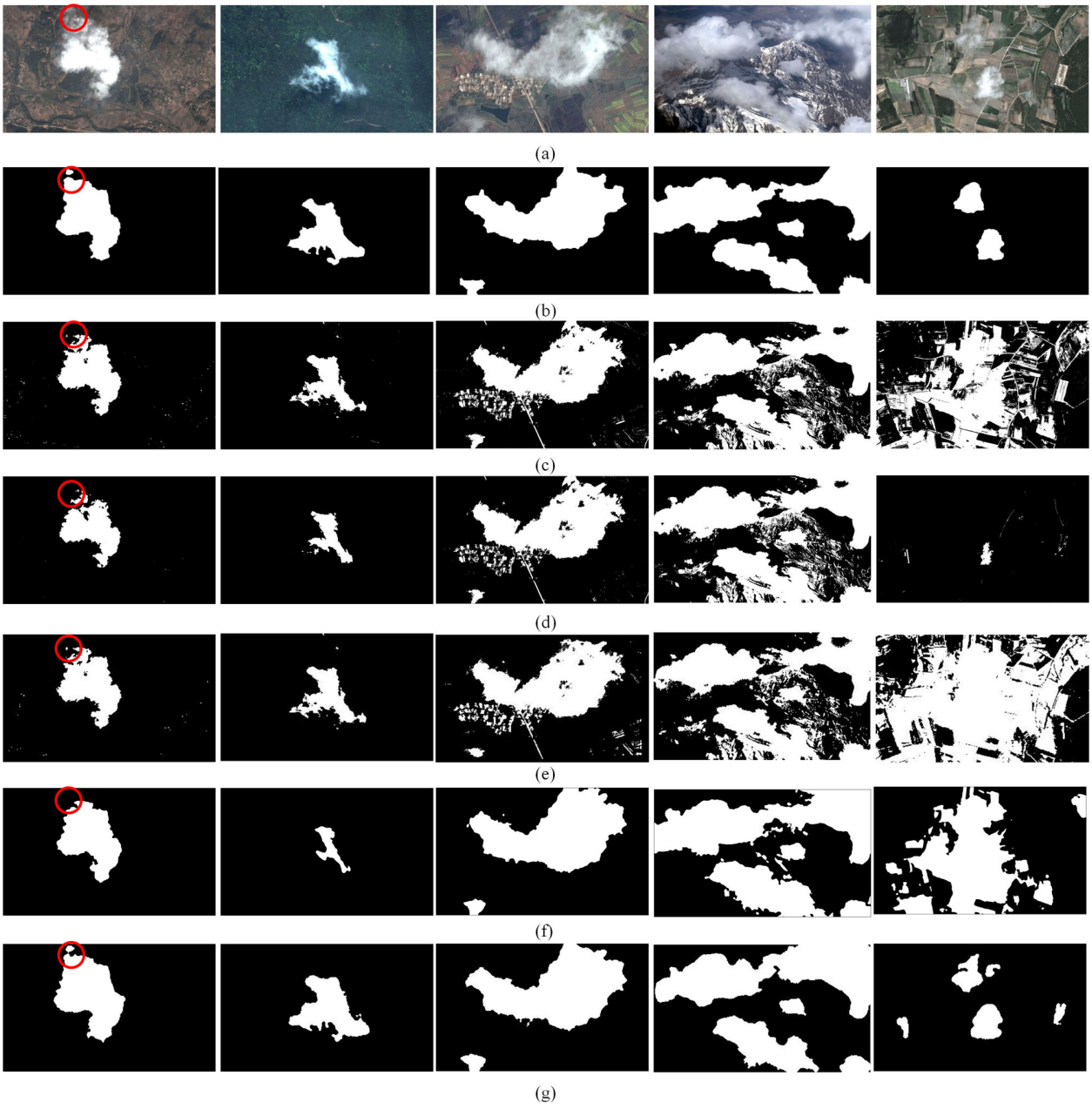

Fig. 9. Comparisons of cloud detection results using different methods on part of data set. (a) Original testing images. (b) Ground truth. (c) Detection results by K-means. (d) Detection results by Mean-shift. (e) Detection results by Chan-Vese. (f) Detection results by [1]. (g) Detection results by the proposed method.

method of [1] has low ER and FAR but not high RR. On the other hand, the proposed method obtains better results, because it has the lowest ER and FAR, and the second highest RR, which implies that the proposed method detected precise cloud areas with the lowest error rate. Meanwhile, the best average value of index RER obtained using the proposed method confirms the above conclusion.

C. Experimental results on influence of feature number

In the section, two groups of experiments are implemented, aiming at deeply exploring the influence of features. In the first group, we will use various individual features to detect clouds. In the second group, experimental results using the proposed methods with different feature combinations are reported.

Fig. 10(a) and (b) are the original images and the corresponding ground truth images. Figs. 10(c)-(f) respectively show the detection results using individual features, including color (Fig. 10(a)), structure (Fig. 10(b)), statistics (Fig. 10(c)) and texture (Fig. 10(d)). Obviously, all the features could detect part of clouds. However, the missing or the false detected regions also happen. Concretely, color feature tends to miss clouds as shown in Fig. 10(c). Structure feature tends to mistake noncloud regions as clouds (see the right result in Fig. 10(d)). Statistics performs slightly better than color and structure but still obtains false alarms (see the left result in Fig. 10(e)). Texture seemly obtains a compromised results, it

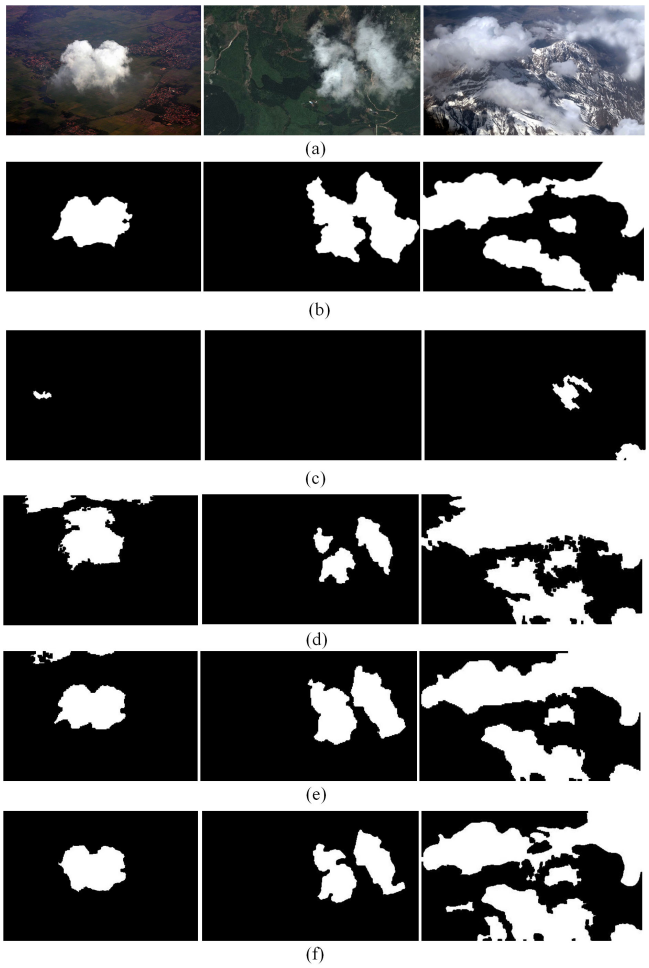

Fig. 10. Visual comparisons of detection results using different individual features. (a) Original testing images. (b)-(e) Detection results using the color feature, structure features, statistics features, texture features, respectively.

misses some clouds in the middle image of Fig. 10(f), but mistakes some backgrounds as clouds in the right image of Fig. 10(f). As a result, although the features show their abilities in cloud detection, ideal results could not be obtained using individual feature. On the other hand, theorem 1 in section III-C has demonstrated that the residual error is reduced if a new proper feature number is added into the feature data, which theoretically implies that the feature combination would be better for detection task. Therefore, we will experimentally confirm the conclusion in the next experiments.

Fig. 11 shows the detection results using different feature combinations. From the top to the bottom rows, the feature combinations color, color + structure, color + structure + statistics, and color + structure + statistics + texture are respectively used. Compared with the source images and ground truth images in Fig. 10(a) and (b), one can easily see that the detection results become better along with the increase of feature number. Fig. 12 illustrates the results of indices ER, RR and RER using different features. The vertical axis is logarithmic for visual convenience. Obviously, the values of RR and RER increase along with the number of features, while the ER values decrease. It also demonstrates the conclusion of theorem 1, namely, if the feature number of feature data is gained, the proposed scene learning scheme performs better.

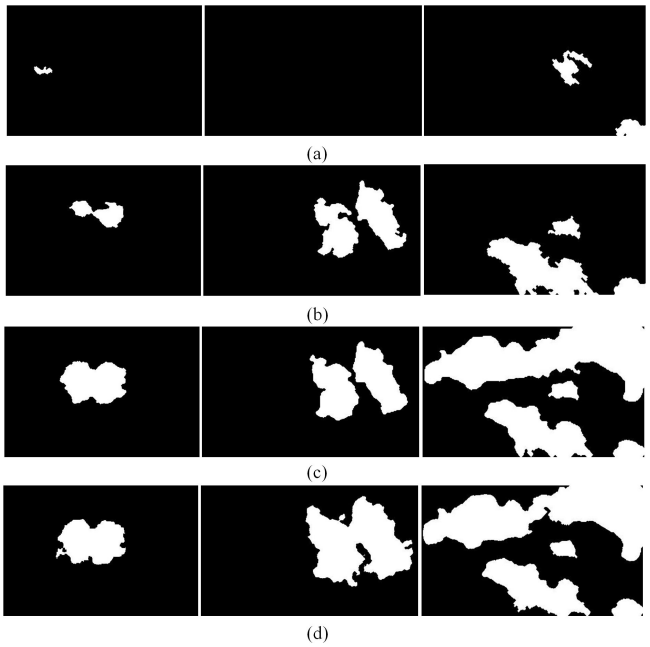

Fig. 11. Visual comparisons of detection results using different feature combinations. (a) Original testing images. (b)-(e) Detection results using the color feature, color + structure features, color + structure + statistics features, color + structure + statistics + texture features, respectively. (f) Ground truth.

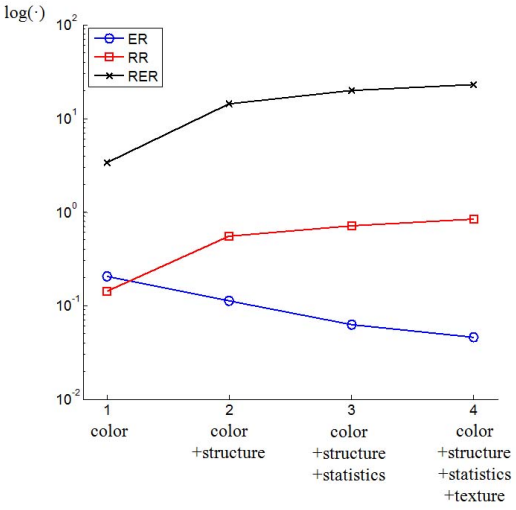

Fig. 12. Quantitative comparisons of detection results using different features.

D. Limitation

It is still a quite difficult problem to detect clouds in remote sensing image with only RGB colors, especially for the image if its background regions have similar brightness as clouds. In such circumstance, all the features of the backgrounds, including colors, statistics, structures and textures are inseparable from those of clouds. On the other hand, the proposed scene learning frame is essentially a supervised method. Results are inevitably affected by the training and testing data. Generally,

diversity of training data is encouraged as it would help to obtain a more universal cloud detector. If the scenes between training data and testing data are extremely different, ideal detection results may be not obtained.

Fig. 13 illustrates an example using different training images, aiming at discussing their influences. Fig. 13(a) is the training image which contains cumulus clouds. Fig. 13(b) is another labeled training with cirrus clouds. Fig. 13(c) are two testing images. Clouds in the upper image are much thinner and have relative much lower brightness than Fig. 13(a), and the cirrus clouds in the bottom image are quite different from Fig. 13(a). Fig. 13(e) are the results when we just use Fig. 13(a) as the training image. Although some clouds are correctly located, more are missed or falsely detected. Fig. 13(f) are the results when both Figs. 13(a) and (b) are applied to train the detector. Obviously, results are promoted and they are much closer to the ground truth. Generally, although the proposed method behaves better than conventional cloud detection algorithms, its performance acquires proper training images, which may limit its application to some extent.

VI. Conclusions and future work

In the paper, we present an automatic algorithm for cloud detection on remote sensing images. It is built on exploiting a novel scene learning frame. The core principle is to train a cloud detector and then apply it to the testing image. Before training and applying the detector, feature data is simulated by stacking different fundamental features. Such data provides more information than original RGB image, so the method could effectively enhance the cloud scenes while suppressing the backgrounds using the data. Applying the trained detector to testing images, saliency maps will be generated, where clouds are well separated from backgrounds. For clouds refinement, an optimal thresholding algorithm is proposed based on revising Otsu's threshold. Effective detail refinement processes are subsequently implemented to finally locate the clouds. Experimental results demonstrate the efficacy of the proposed method. Residual error in designing detector is calculated. Furthermore, we theoretically and experimentally demonstrate that more features lead to better performance.

For further work, influence of feature will be more deeply studied. In the paper, only four fundamental image features are applied to simulating feature data. Other features, like Haar or HOG could be considered. These features have been proved to be effective in computer vision, and they may help to improve the algorithm. Besides, how the method behaves in handling images with more than 3 visible bands could be researched. Concretely, since the existence of near-infrared bands, multispectral images provide more information compared with images with only RGB colors, and the special band would be helpful for the performance of the proposed method. On the other hand, one can find that band number of the simulated feature data is about 40 times of the original image. It is a quite large data size. Therefore, when applying the method to cloud detection in hyperspectral images (HSI), how to choose features and generate should be taken into consideration, because HSI has much more bands than RGB image.

VII. Appendix

Proof for Theorem 1: To begin with, we calculate the $\mathbf{C}_\Phi^{-1}$. Expand the matrix $\mathbf{C}_\Phi$, we have

$$\begin{aligned}\mathbf{C}_\Phi &= \mathrm{E}\{\mathbf{x}_{i,\Phi}\mathbf{x}_{i,\Phi}{}^T\} = \mathrm{E}\left\{\begin{bmatrix}\mathbf{x}_i\\ x_{i,L+1}\end{bmatrix}\begin{bmatrix}\mathbf{x}_i\\ x_{i,L+1}\end{bmatrix}^T\right\}\\ &= \mathrm{E}\left\{\begin{bmatrix}\mathbf{x}_i\mathbf{x}_i^T & \mathbf{x}_i x_{i,L+1}\\ (\mathbf{x}_i x_{i,L+1})^T & x_{i,L+1}^2\end{bmatrix}\right\}\\ &= \begin{bmatrix}\mathrm{E}(\mathbf{x}_i\mathbf{x}_i^T) & \mathrm{E}(\mathbf{x}_i x_{i,L+1})\\ \mathrm{E}(\mathbf{x}_i x_{i,L+1})^T & \mathrm{E}(x_{i,L+1}^2)\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{C} & \mathbf{s}\\ \mathbf{s}^T & s_{L+1}\end{bmatrix}\end{aligned} \tag{35}$$

Then its inverse is calculated as follows:

$$\begin{aligned}\mathbf{C}_\Phi^{-1} &= \begin{bmatrix}\mathbf{C} & \mathbf{s}\\ \mathbf{s}^T & s_{L+1}\end{bmatrix}^{-1}\\ &= K\begin{bmatrix}\frac{\mathbf{C}^{-1}}{K}+\mathbf{C}^{-1}\mathbf{s}\mathbf{s}^T\mathbf{C}^{-1} & -\mathbf{C}^{-1}\mathbf{s}\\ -\mathbf{s}^T\mathbf{C}^{-1} & 1\end{bmatrix}\end{aligned}$$

where $K=(s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s})^{-1}$. So the quadratic form of J_Φ could be obtained as:

$$\begin{aligned}&\mathbf{d}_\Phi^T\mathbf{C}_\Phi^{-1}\mathbf{d}_\Phi\\ &= K\begin{bmatrix}\mathbf{d}\\ d_{L+1}\end{bmatrix}^T\begin{bmatrix}\frac{\mathbf{C}^{-1}}{K}+\mathbf{C}^{-1}\mathbf{s}\mathbf{s}^T\mathbf{C}^{-1} & -\mathbf{C}^{-1}\mathbf{s}\\ -\mathbf{s}^T\mathbf{C}^{-1} & 1\end{bmatrix}\begin{bmatrix}\mathbf{d}\\ d_{L+1}\end{bmatrix}\\ &= K(\tfrac{\mathbf{d}^T\mathbf{C}^{-1}\mathbf{d}}{K}+\mathbf{d}^T\mathbf{C}^{-1}\mathbf{s}\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d}\\ &-d_{L+1}\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d}-d_{L+1}\mathbf{d}^T\mathbf{C}^{-1}\mathbf{s}+d_{L+1}^2)\\ &=\mathbf{d}^T\mathbf{C}^{-1}\mathbf{d}+\tfrac{(d_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2}{s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}}\end{aligned}$$

Calculate the difference between $J(\mathbf{w})$ and J_Φ, we have

$$\begin{aligned}&J(\mathbf{w})-J(\Phi)\\ &=\tfrac{1}{2}(\mathbf{d}_\Phi^T\mathbf{C}_\Phi^{-1}\mathbf{d}_\Phi-\mathbf{d}^T\mathbf{C}^{-1}\mathbf{d})\\ &=\tfrac{1}{2}((\mathbf{d}^T\mathbf{C}^{-1}\mathbf{d}+\tfrac{(d_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2}{s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}})-\mathbf{d}^T\mathbf{C}^{-1}\mathbf{d})\\ &=\tfrac{1}{2}\tfrac{(d_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2}{s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}}\end{aligned} \tag{36}$$

Since the numerator in equation (36) is a square form, $(d_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2\geq 0$ holds. Meanwhile, according to the rule of the partitioned matrix determinant, we have

$$\begin{aligned}|\mathbf{C}_\Phi| &= \left|\begin{bmatrix}\mathbf{C} & \mathbf{s}\\ \mathbf{s}^T & s_{L+1}\end{bmatrix}\right|\\ &= |\mathbf{C}|\left|s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}\right| = (s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s})\,|\mathbf{C}|\end{aligned}$$

where $|\cdot|$ is the determinant operator. Note that both $\mathbf{C}_\Phi$ and $\mathbf{C}$ are symmetric positive-definite matrices. If $|\mathbf{C}|$ is invertible, then we have

$$\frac{|\mathbf{C}_\Phi|}{|\mathbf{C}|}=(s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s})>0$$

Thus,

$$J(\mathbf{w})-J(\Phi)=\frac{1}{2}\frac{(d_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d})^2}{s_{L+1}-\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}}\geq 0$$

and the equality sign holds while $d_{L+1}=\mathbf{s}^T\mathbf{C}^{-1}\mathbf{d}$. The proof of Theorem 1 is thus completed. ■

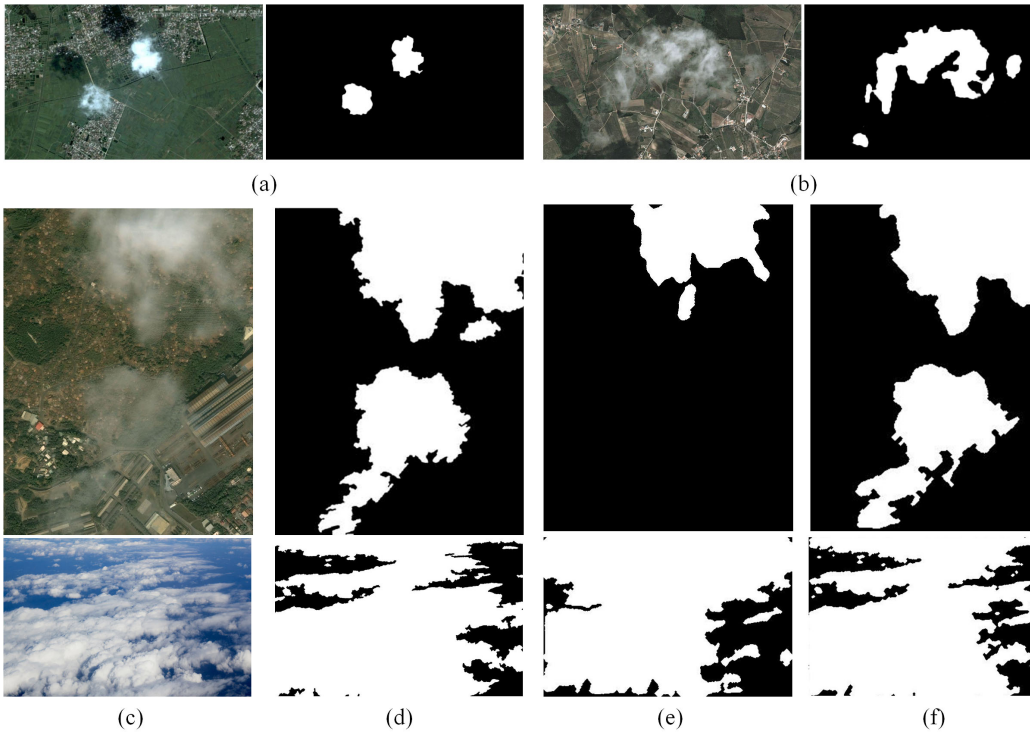

Fig. 13. Detection results using different training images. (a)-(b) Two training images and their ground truths. (c) Two testing images. (d) Ground truths of testing images. (e) Failed detection results using only one training image. (f) Successful detection results using the two training images.

REFERENCES

[1] Q. Zhang and C. Xiao, "Cloud detection of rgb color aerial photographs by progressive refinement scheme," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7264 – 7275, 2014.

[2] R. J. Kauth and G. Thomas, "The tasselled cap–a graphic description of the spectral-temporal development of agricultural crops as seen by landsat," in *LARS Symposia*, pp. 41–51, 1976.

[3] Y. Zhang, B. Guindon, and J. Cihlar, "An image transform to characterize and compensate for spatial variations in thin cloud contamination of landsat images," *Remote Sensing of Environment*, vol. 82, no. 2, pp. 173–187, 2002.

[4] O. Hagolle, M. Huc, D. V. Pascual, and G. Dedieu, "A multi-temporal method for cloud detection, applied to formosat-2, venμs, landsat and sentinel-2 images," *Remote Sensing of Environment*, vol. 114, no. 8, pp. 1747–1755, 2010.

[5] J. Cihlar and J. Howarth, "Detection and removal of cloud contamination from avhrr images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, no. 3, pp. 583–589, 1994.

[6] R. Richter and A. Müller, "De-shadowing of satellite/airborne imagery," *International Journal of Remote Sensing*, vol. 26, no. 15, pp. 3137–3148, 2005.

[7] R. Richter, "Atmospheric correction of satellite data with haze removal including a haze/clear transition region," *Computers & Geosciences*, vol. 22, no. 6, pp. 675–681, 1996.

[8] J. Long, Z. Shi, W. Tang, and C. Zhang, "Single remote sensing image dehazing," *Geoscience and Remote Sensing Letters, IEEE*, vol. 11, no. 1, pp. 59–63, 2014.

[9] M. de Ruyter de Wildt, G. Seiz, and A. Gruen, "Operational snow mapping using multitemporal meteosat seviri imagery," *Remote Sensing of Environment*, vol. 109, no. 1, pp. 29–41, 2007.

[10] R. Rossi, R. Basili, F. Del Frate, M. Luciani, and F. Mesiano, "Techniques based on support vector machines for cloud detection on quickbird satellite imagery," in *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pp. 515–518, IEEE, 2011.

[11] S. Le Hégarat-Mascle and C. André, "Use of markov random fields for automatic cloud/shadow detection on high resolution optical images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 4, pp. 351–366, 2009.

[12] G. Vivone, P. Addesso, R. Conte, M. Longo, and R. Restaino, "A class of cloud detection algorithms based on a map-mrf approach in space and time," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 8, pp. 5100–5115, 2014.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[15] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2006.

[16] E. Welch, R. Moorhead, and J. Owens, "Image processing using the hsi color space," in *Southeastcon'91., IEEE Proceedings of*, pp. 722–725, IEEE, 1991.

[17] A. R. Weeks and G. E. Hague, "Color segmentation in the hsi color space using the k-means algorithm," in *Electronic Imaging'97*, pp. 143–154, International Society for Optics and Photonics, 1997.

[18] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," in *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, pp. 14–19, IEEE, 1990.

[19] R. Mehrotra, K. R. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, no. 12, pp. 1479–1494, 1992.

[20] T. P. Weldon, W. E. Higgins, and D. F. Dunn, "Efficient gabor filter design for texture segmentation," *Pattern Recognition*, vol. 29, no. 12, pp. 2005–2015, 1996.

[21] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition‡modeling, algorithms, and parameter selection," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 111–136, 2006.

[22] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 139, 2012.

[23] G. Gordon and R. Tibshirani, "Karush-kuhn-tucker conditions," *Optimization*, vol. 10, no. 725/36, p. 725.

[24] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[25] I. Pitas and A. N. Venetsanopoulos, "Morphological shape representation," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pp. 2381–2384, IEEE, 1991.

[26] P. Soille, *Morphological image analysis: principles and applications*. Springer-Verlag New York, Inc., 2003.

[27] P. I. Corke, "The machine vision toolbox: a matlab toolbox for vision and vision-based control," *Robotics & Automation Magazine, IEEE*, vol. 12, no. 4, pp. 16–25, 2005.

[28] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 881–892, 2002.

[29] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[30] D. Comaniciu and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 750–755, IEEE, 1997.

[31] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1197–1203, IEEE, 1999.

[32] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[33] T. F. Chan and L. A. Vese, "Active contours without edges," *Image processing, IEEE transactions on*, vol. 10, no. 2, pp. 266–277, 2001.

[34] T. F. Chan, B. Y. Sandberg, and L. A. Vese, "Active contours without edges for vector-valued images," *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 130–141, 2000.

[35] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the mumford and shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

[36] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 3, pp. 629–640, 2011.

[37] S. Yang, Z. Shi, and W. Tang, "Robust hyperspectral image target detection using an inequality constraint," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3389–3404, 2015.