

Article

Cloud Extraction from Chinese High Resolution Satellite Imagery by Probabilistic Latent Semantic Analysis and Object-Based Machine Learning

Kai Tan, Yongjun Zhang * and Xin Tong

School of Remote Sensing and Information Engineering, Wuhan University, No. 129 Luoyu Road, Wuhan 430079, China; kai_tan@whu.edu.cn (K.T.); xin_tong@whu.edu.cn (X.T.)

* Correspondence: zhangyj@whu.edu.cn; Tel.: +86-27-6877-1101

Academic Editors: Gonzalo Pajares Martinsanz, Clement Atzberger and Prasad S. Thenkabail

Received: 12 August 2016; Accepted: 15 November 2016; Published: 22 November 2016

Abstract: Automatic cloud extraction from satellite imagery is a vital process for many applications in optical remote sensing since clouds can locally obscure the surface features and alter the reflectance. Clouds can be easily distinguished by the human eyes in satellite imagery via remarkable regional characteristics, but finding a way to automatically detect various kinds of clouds by computer programs to speed up the processing efficiency remains a challenge. This paper introduces a new cloud detection method based on probabilistic latent semantic analysis (PLSA) and object-based machine learning. The method begins by segmenting satellite images into superpixels by Simple Linear Iterative Clustering (SLIC) algorithm while also extracting the spectral, texture, frequency and line segment features. Then, the implicit information in each superpixel is extracted from the feature histogram through the PLSA model by which the descriptor of each superpixel can be computed to form a feature vector for classification. Thereafter, the cloud mask is extracted by optimal thresholding and applying the Support Vector Machine (SVM) algorithm at the superpixel level. The GrabCut algorithm is then applied to extract more accurate cloud regions at the pixel level by assuming the cloud mask as the prior knowledge. When compared to different cloud detection methods in the literature, the overall accuracy of the proposed cloud detection method was up to 90 percent for ZY-3 and GF-1 images, which is about a 6.8 percent improvement over the traditional spectral-based methods. The experimental results show that the proposed method can automatically and accurately detect clouds using the multispectral information of the available four bands.

Keywords: satellite imagery; cloud detection; superpixel; probabilistic latent semantic analysis; support vector machine; GrabCut

1. Introduction

According to the International Satellite Cloud Climatology Project-Flux Data (ISCCP-FD), the global annual mean cloud cover is approximately 66% [1]. Clouds are common elements in Chinese high resolution remote sensing satellite imagery which not only can lead to spectral distortion, but also can affect the processing of remote sensing imagery. Cloud pixels are the main source of invalid pixels in satellite imagery and should be detected and excluded from further processing when possible. Reliable cloud masks for remote sensing images therefore can significantly enhance the further analysis and utility of these images by reducing the computational burden and improving productivity by eliminating images with large areas of clouds. Accurate cloud detection also strongly supports the subsequent image processing steps, such as image rectification, image compositing, and image dodging [2].

Most automatic cloud detection methods are based on the radiometric features of clouds. The spectral, frequency, and texture features are the most widely used radiometric features and are extracted from different bands of satellite imagery [3–7]. The spectral features are extracted by multiple bands operations which are primarily based on the characteristics that clouds are generally brighter than most of the Earth's surface in visible bands and the radiances emitted by the clouds can be easily measured by satellite instrument sensors in thermal infrared (TIR) spectral regions. Chylek, et al. (2006) concluded that algorithms based on bands centered around 11.0 and 8.55 μm will be a good starting point in future development of cloud detection for hyperspectral TIR instruments [8]. The distinct frequency features of clouds are usually extracted by wavelet analysis and Fourier transform [9]. Texture features can be extracted by second-moment [10], fractal dimension [11], and a gray-level co-occurrence matrix (GLCM) [12,13]. To distinguish cloud and non-cloud pixels or objects, these features are mainly combined with the threshold [14], support vector machine (SVM) [15,16] or clustering [17] methods. The threshold method is a simple, efficient, and practical approach for cloud detection, but its sensitivity to the background and the range of cloud coverage make it impractical for general use [14]. The SVM based method is widely used for detection purposes in computer vision, and its major task lies in the selection of its kernel. The different kernel functions produce different SVMs and therefore may result in different levels of performance [16]. The clustering method is efficient and appropriate, but usually requires specifying the centers and species of clustering in advance, which limits its popularity in automatic cloud detection.

Several past studies proposed more sophisticated alternatives to adapt, refine, and/or substitute for the traditional cloud detection approaches. Xuemei, et al. (2012) proposed a cloud detection method (denoted as k-means + ICM in this paper) based on Markov Random Field (MRF) [18], which turned the problem of classification into a gradual refinement process by updating the label of each pixel in the image. Their results confirmed the effectiveness of their method and pointed out a variety of advantages to use MRF models for image object detection. Fisher, et al. (2014) investigated the method SPOTCASM to detect clouds and cloud shadows synchronously [19] and achieved high detection accuracy. However their method is limited in the case of shadows near certain water bodies, especially small lakes, which are similar in size to clouds in an image and are aligned with the solar illumination direction [20]. Hu, et al. (2015) studied the method of object-oriented cloud detection [21] and established saliency maps to represent the visual saliency of clouds in images, with which they were able to quickly classify an image as a cloud or non-cloud image. Zhang, et al. (2012) proposed an automatic and effective cloud detection algorithm (denoted as RGB refinement in this paper) based on the properties derived from observations and statistic results on a large number of images with cloud layers [22]. Yuan, et al. (2015) proposed an automatic cloud detection (denoted as SIFT + GrabCut in this paper) based on object classification of image features [23]. A brief introduction of each automatic cloud detection algorithm is provided in Table 1.

Humans usually can identify clouds from satellite images based on their regional characteristics instead of a pixel-by-pixel identification. In this paper, we propose a novel object-pixel two-level algorithm for extracting clouds from Chinese high resolution satellite imagery. The main contributions of this paper are as follow.

- (1) A novel object-pixel two-level machine learning algorithm is proposed, which involves cloud mask detection at the object level and subsequent accurate cloud detection at the pixel level.
- (2) A PLSA model is introduced to extract implicit information, which greatly improves the recognition accuracy of the SVM algorithm and effectively solves the problem of high-dimension in the machine learning field.
- (3) The well-known foreground extraction algorithm in computer vision, GrabCut, which received very little attention in the literature, is utilized in the proposed method to obtain accurate cloud detection results.

The remainder of this paper is organized as follows. In Section 2 the datasets chosen are discussed. The proposed methodology for cloud extraction is introduced in Section 3. In Section 4, the feasibility of the method is validated, and the parameters of the algorithm are analyzed. In Section 5, the proposed methodology is compared to other cloud detection methods, the results are discussed, and the limitations of the algorithm are introduced. Finally, the conclusions are presented in Section 6.

Table 1. Brief introduction of each automatic cloud detection algorithm that published in the past few years.

Algorithm	Published Year	Advantage	Disadvantage
k-means + ICM [18]	2012	<ul style="list-style-type: none"> • Novel. • High-accuracy. • Efficient. 	<ul style="list-style-type: none"> • It is complex, and hard to be implemented.
SPOTCASM [19]	2014	<ul style="list-style-type: none"> • Not only cloud but also shadow masking can be extracted. • Implemented using entirely open source software. • Robust. 	<ul style="list-style-type: none"> • Some errors should be reduced through manual editing. • Not appropriate for images with much cloud contamination.
Object-oriented [21]	2015	<ul style="list-style-type: none"> • Novel. • Efficient. • Robust. 	<ul style="list-style-type: none"> • Semitransparent clouds are a challenge to identify.
RGB refinement [22]	2012	<ul style="list-style-type: none"> • Only R, G, B channels are used, it is easy to be applied widely. • Robust. • High-accuracy. 	<ul style="list-style-type: none"> • Bright non-cloud regions that have little details are very challenging for this algorithm. • Very thin clouds will failed to be extracted.
SIFT + GrabCut [23]	2015	<ul style="list-style-type: none"> • Novel. • Robust. • Not only true color (RGB) but also gray images can be used to do cloud detection. 	<ul style="list-style-type: none"> • Small cloud regions will failed to be extracted. • Semitransparent clouds are a challenge to identify.

2. Datasets

In this study, two categories of Chinese high-spatial resolution remote-sensing images were used for cloud detection: GF-1 and ZY-3 [24]. The detailed parameters of these images are provided in Table 2. These images contain a variety of clouds, such as thin cloud, thick cloud and cirrus cloud, and many types of surfaces, such as vegetation, city, sand, bare land, river, and snow. Generally, thick clouds are easier to extract than thin and cirrus clouds, as thick clouds usually are in distinct contrast with their background. Vegetation usually shows presents in darker colors and can be easily distinguished in images. In a typical Chinese city landscape, buildings are generally rectangular, so they are easily identifiable by their shape features. Two of the typical large rivers of China, the Yangtze River and the Yellow River, are included in the study images. Rivers are usually surrounded by sand in the margin areas. Snow is the most difficult element to distinguish in images, as its appearance is similar to clouds. In the proposed method, snow is distinguished from clouds by taking into account the distribution status, geographic location and season. The graphic examples for the three cloud types are illustrated in Figure 1.

Table 2. Overview of Chinese ZY-3 and GF-1 multispectral datasets.

Satellite Parameters		ZY-3	GF-1
Product Level		1A	1A
Number of bands		4	4
Wavelength (nm)		Blue: 450–520; Green: 520–590; Red: 630–690; NIR: 770–890	
Spatial resolution (m)		5.8	8
Radiometric resolution		1024	1024
Image size (pixel)		8824 × 9307	4548 × 4596
Acquisition Time (year)		2013	2013
Number of images		36	42
Land cover types	Cloud types Surface types	Thin cloud, Thick cloud, Cirrus cloud Forest, City, Sand, Bare land, River, Snow, etc.	



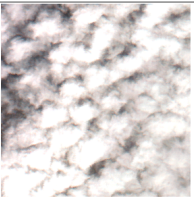
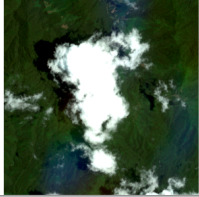

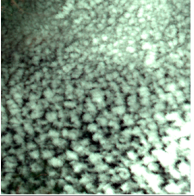
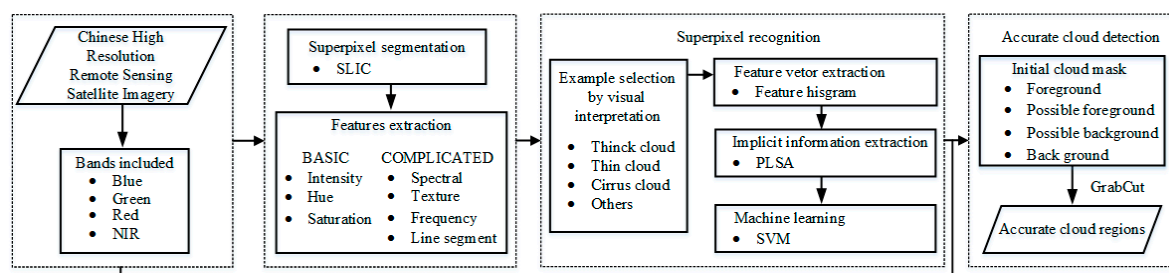
	Thick cloud	Thin cloud	Cirrus cloud
ZY-3 Image			
GF-1 Image			
Characteristics (Key words)	<ul style="list-style-type: none"> ◇ Brightest ◇ Irregular ◇ Has little details 	<ul style="list-style-type: none"> ◇ Semitransparent ◇ Has little details ◇ Brighter 	<ul style="list-style-type: none"> ◇ Has more details ◇ Dark or bright ◇ Cluster

Figure 1. Graphic examples of the three types of clouds visually recognizable from remotely sensed data: thick cloud, thin cloud, Cirrus cloud.

3. Methodology

The workflow of the proposed approach for cloud detection is shown in Figure 2. The major steps include superpixel segmentation, superpixel recognition, and accurate cloud detection.

**Figure 2.** Proposed cloud detection framework.

3.1. Superpixel Segmentation

The term “superpixel” was introduced by Ren, et al. in 2003 [25]. Superpixel algorithms usually group pixels into perceptually meaningful atomic regions which can be used to replace the rigid structure of the pixel grid. Pixels are grouped by the degree of similarity among the neighboring pixels, which provides a convenient method to compute image features and greatly reduces the complexity of

cloud detection. Achanta, et al. [26,27] proposed a simple, convenient, and fast superpixel algorithm named simple linear iterative clustering (SLIC), which generates superpixels based on k-means clustering. The experimental results show that the superpixels segmented by SLIC were compact and uniform in size, and adhered well to the region boundaries. The initial width S of superpixels should be assigned in advance of the segmentation process. When S is set at 30 (the effect of the parameter S for cloud detection is analyzed in Section 4.2), the result of SLIC segmentation is shown in Figure 3.

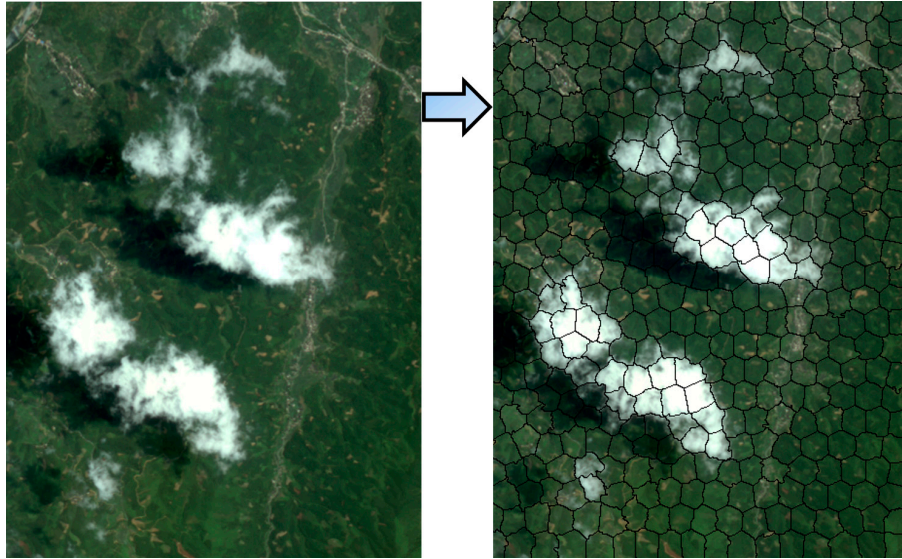


Figure 3. Superpixel segmentation by SLIC.

3.2. Superpixel Recognition

3.2.1. Features for Cloud Detection

The red, green, blue (RGB) color model works well for thick cloud detection, but is not as effective in detecting thin clouds and cirrus clouds. The HIS color model, which follows the human visual perception closely, separates the color components in terms of hue (H), intensity (I) and saturation (S). RGB can be transformed to HIS as follows:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{6} & 0 \\ 0 & \frac{2}{\sqrt{6}} & 1 \end{bmatrix} \begin{bmatrix} \min(R, G, B) \\ \text{mid}(R, G, B) \\ \max(R, G, B) \end{bmatrix} \quad (1)$$

$$H' = \arccos\{0.5 \times [(R' - G') + (R' - B')] / [(R' - G')^2 + (R' - G') \times (G' - B')]^2\} \quad (2)$$

$$I = 1/3 (R + G + B) \quad (3)$$

$$S = 1 - [3 / (R + G + B)] \times \min(R, G, B) \quad (4)$$

$$H = \begin{cases} 0 & \text{if } S = 0 \\ 360 - H' & \text{else if } B > G \\ H' & \text{else} \end{cases} \quad (5)$$

where I , S , and H correspond to the intensity-equivalent, saturation-equivalent, and hue-equivalent components, respectively.

The basic features that are used for cloud detection in this paper are shown in Figure 4. It can be seen that cloud regions usually share the following common properties:

- Most cloud regions often have lower hues than non-cloud regions
- Cloud regions generally have a higher intensity and NIR since the reflectivity of cloud regions is usually larger than that of non-cloud regions
- Cloud regions generally have lower saturation since they are white in a RGB color model
- The ground covered by a cloud veil usually has few details as the ground object features are all attenuated by clouds
- Cloud regions always appear in terms of clustering

In addition, a remote sensing image can be converted from spatial domain to frequency domain by wavelet transform, which can decompose the image into low-frequency and high-frequency components. The objects that change dramatically in gray scale tend to be distributed in the high-frequency components, while those that change slowly, such as clouds, are distributed mainly in the low-frequency component.

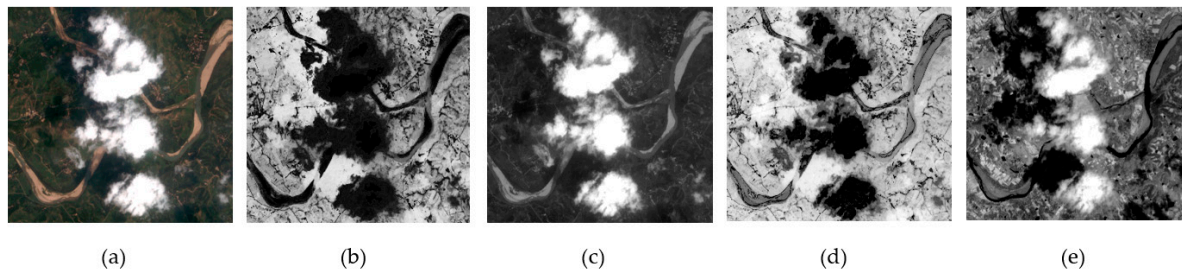


Figure 4. Basic features for cloud detection. (a) Original image; (b) Hue channel; (c) Intensity channel; (d) Saturation channel; (e) NIR channel.

Based on the above features, more complicated features can be introduced as follows shown in Figure 5.

(a) Spectral feature (SF): To highlight the difference between cloud regions and non-cloud regions, SF is extracted by constructing a significance map [22]. This feature fully uses the properties that cloud regions in satellite images: higher intensity and lower hue, which can be written as:

$$SF = \frac{I' + \tau}{S' + \tau} \quad (6)$$

where I' and S' refer to intensity and saturation bounded to (0, 1), respectively. τ is an offset value, which provides an amplification of SF and ensure that the denominator is greater than 0. In this paper, we typically used $\tau = 1.0$. To obtain an intuitive visual description of SF, the value of SF is scaled to the range of (0, 255) (see Figure 5b);

(b) Texture feature (TF): Ideal detail map can be extracted by multiple bilateral filtering [22], which is effective but time-consuming. In the experiment, histogram equalization is presented in the intensity channel to highlight the implicit texture information; then, the texture feature can be extracted by applying bilateral filtering only once. As cloud regions in satellite images usually have fewer details compared to complex ground objects, this feature is an important clue to successful determination of non-cloud regions. The bilateral filtering is defined as:

$$IE'(i, j) = \frac{\sum_{(i,j) \in \Omega_{x,y}} w(i, j) IE(i, j)}{\sum_{(i,j) \in \Omega_{x,y}} w(i, j)} \quad (7)$$

$$w(i, j) = W_s(i, j) \times W_r(i, j) \quad (8)$$

$$W_s(i, j) = e^{-\frac{\| (i, j) - (i', j') \|^2}{2\sigma_s^2}} \quad (9)$$

$$W_r(i, j) = e^{-\frac{\| IE(i, j) - IE(i', j') \|^2}{2\sigma_r^2}} \quad (10)$$

$$TF = |IE - IE'| \quad (11)$$

where IE refers to the intensity enhanced by traditional histogram equalization. IE' denotes the filtered image from IE; $\Omega_{x,y}$ denotes the square window centered in (i, j) ; (i', j') denotes the coordinates of the pixels within $\Omega_{x,y}$; $w(i, j)$ denotes the calculated weight composed of space distance factor $W_s(i, j)$ and similarity factor $W_r(i, j)$; and σ_s and σ_r are the standard deviations of the Gaussian function in the space domain and the color domain, respectively. If σ_s is larger, the border pixels have a greater effect on the values of the center pixel. On the other hand, if σ_r is larger, the pixels with a larger radiation difference with the center pixel are taken into the calculations. In the implementation process, we typically set the size of σ_s to 2, and σ_r to $IE_{\max}/10$, which denotes the maximum value of intensity after histogram equalization. In the same way, the value of TF is scaled to the range of (0, 255) (see Figure 5c);

(c) Frequency feature (FF): As clouds are always a low-frequency component in satellite images, the low-frequency information of the image is extracted by wavelet transform in intensity space [8]. In this paper, we first used biorthogonal wavelet transform to convert an image from the spatial domain to the frequency domain (the number of layer was set to 2). Then the value of the pixels in the low-frequency components remained unchanged, and the pixels in the high-frequency component were set to 0. As can be seen in Figure 5d, many of the non-cloud objects in the high-frequency component were filtered by this method.

(d) Line segment feature (LSF): Clouds are generally irregular in shape, and there are very few line segments in the cloud regions, especially in thick and thin cloud regions. However, most of the human-made objects have a flat surface, so the line segments can be used to distinguish clouds from human-made objects. In this paper, the line segments are identified by line segment detector (LSD) algorithm [28] (see Figure 5e).

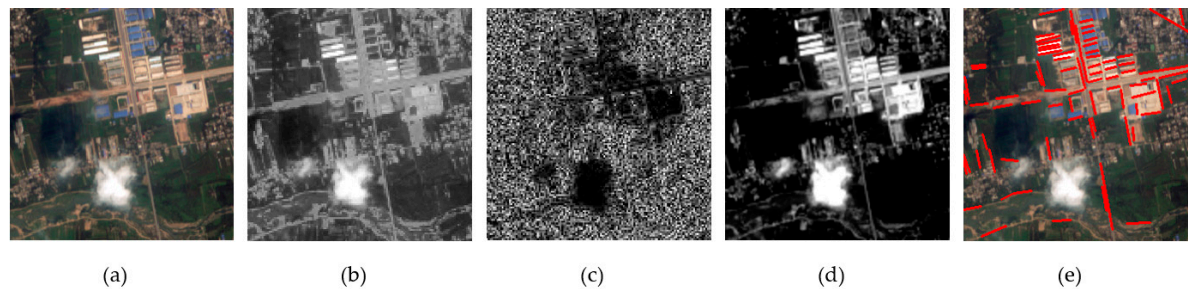


Figure 5. Complicated features of a graphic example. (a) Original image; (b) SF image; (c) TF image; (d) FF image; (e) LSD.

3.2.2. Background Superpixels (BS) Extraction

The basic premise of cloud detection is that cloud regions usually have higher SF, H, and NIR values, but the response in the space of TF and LSF is less than in the non-cloud regions and is always present in the low-frequency component. In fact, the greatest challenges for cloud detection are snow, buildings, sand, and bare land because they have high spectral reflectance in the visible bands. To improve the robustness and efficiency of cloud detection, some obvious background objects, such as water area, shadow, forest, and grass, can be extracted easily using the features introduced in the above section. To identify the distributions law of the features of true cloud superpixels, 500 superpixels for each type of cloud were collected. The average SF, TF, H, and NIR for each superpixel was counted and shown in Figure 6. In order to keep the same intuitive visual description with SF and TF, the value

of H and NIR were scaled to (0, 255). The specific dataset collection process of the proposed method therefore is as follows: (1) cloud images covering every season are selected; (2) all of the images are segmented into superpixels; (3) the average value of each feature is calculated for each superpixel; and (4) 500 superpixels are selected for each type of cloud, and the statistics for their feature values are obtained artificially.

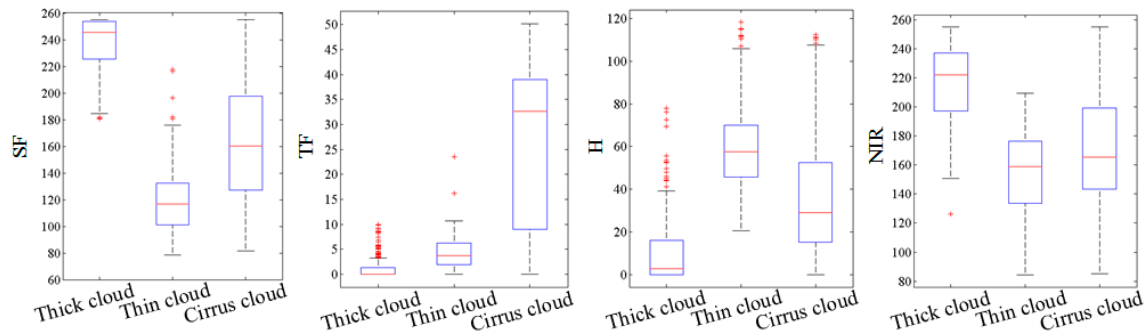


Figure 6. Statistics of the features information for the three cloud types. Each box plot shows the location of the 25, 50, and 75 percentiles using horizontal lines. The two whiskers cover the 99.3% percentile of the data, and the red “+” is the outlier outside the two whiskers.

Based on the SF information, the background superpixels from the input image can be identified coarsely by using a global thresholding method, such as Otsu’s method [29]. However, due to the threshold shifting problem of Otsu’s method, some cloud superpixels may be missed when a prohibitively high Otsu’s threshold T_{otsu} is selected. Conversely, an excessive number of non-cloud superpixels may be mistaken for cloud superpixels when a prohibitively low Otsu’s threshold T_{otsu} is selected [22]. To ensure detection accuracy, this paper proposes an optimal threshold T setting by adding an appropriate constraint to T_{otsu} . Based on the above observation, over 99.3% of the cloud superpixels were found to have a SF not less than 80. Thus, 80 was set as the lowest bound, which means that if Otsu’s threshold is lower than 80, the proposed method takes the optimal threshold of 80. Otherwise, Otsu’s threshold is selected. In the same way, the upper bound was set at 130, which was founded to be suitable for protecting the cloud superpixels. The formula for obtaining optimal threshold T is shown below:

$$T = \begin{cases} 80 & \text{if } T_{otsu} < 80 \\ T_{otsu} & \text{if } 80 \leq T_{otsu} \leq 130 \\ 130 & \text{if } T_{otsu} > 130 \end{cases} \quad (12)$$

In Figure 6, it can be seen that although the feature values of TF, H, and NIR for different cloud types were different, but most all of these cloud superpixels had a TF of less than 50, their H was less than 120, and the NIR was more than 85. Therefore, it was concluded that true cloud superpixels must meet the following conditions:

- Condition 1: The feature value of SF is larger than optimal threshold T ;
- Condition 2: The feature value of TF is less than 50;
- Condition 3: The feature value of H is less than 120;
- Condition 4: The feature value of NIR is not less than 85.

Superpixels are defined as BS if none of these four conditions is satisfied. One example of BS extraction based on four conditions is shown in Figure 7.

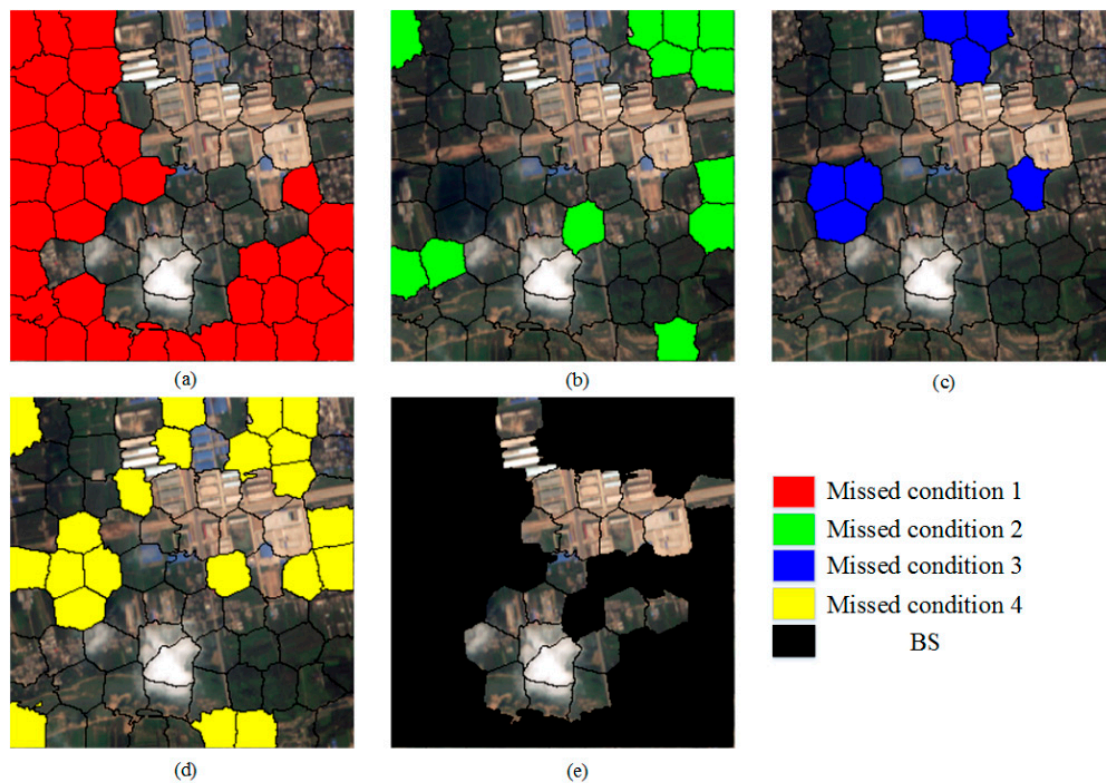


Figure 7. Example showing BS extraction based on four conditions. (a) Superpixels that missed condition 1; (b) Superpixels that missed condition 2; (c) Superpixels that missed condition 3; (d) Superpixels that missed condition 4; (e) BS extraction result.

3.2.3. Cloud Mask Extraction

SF, TF, and FF usually distinguishing different targets well. However, the same target may present different features, and different targets may present similar features in complex satellite images. Thus, problems of ambiguity and similarity between the feature vectors and the target semantics may appear and obtaining the true semantics of the target for machine learning algorithms in the case of inadequate training samples is a difficult process. To solve these problems, bag-of-words (BOW), latent semantic analysis (LSA), PLSA, and other were introduced in computer vision. T. Hofmann, et al. (2001) experimentally verified that PLSA is more principled than BOW and LSA in extracting latent features semantics, since PLSA possess a sound statistical foundation and utilizes the (annealed) likelihood function as an optimization criterion [30]. In this paper, the PLSA model therefore was applied to training typical images, and the targets then were recognized based on the PLSA distribution. The workflow of the cloud mask extraction process is shown in Figure 8.

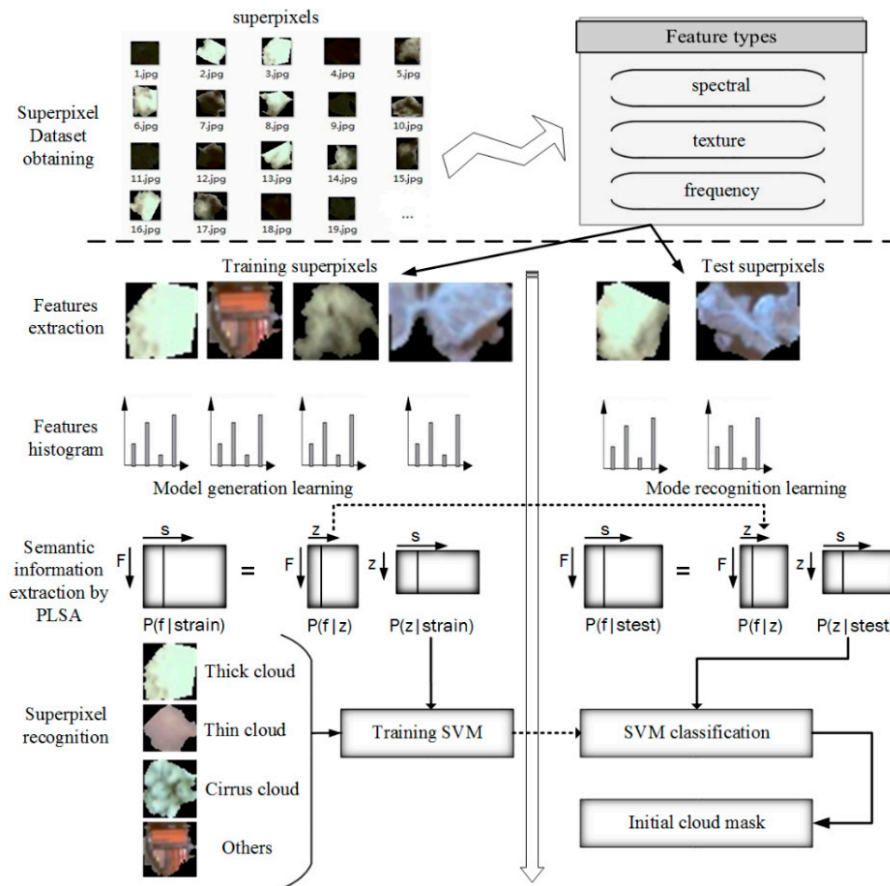


Figure 8. Flow chart of cloud mark extraction.

The detailed recognition steps are as follows:

(a) Superpixel dataset obtaining.

Remote sensing images are segmented into superpixels, and the cloud and non-cloud superpixels are selected as the training dataset. In theory, the more training samples there are, the better the results are; and the number of negative samples should be twice the number of positive samples.

(b) Features extraction.

Feature selection is a very important step in superpixels recognition. In this paper, $STF = \{sf_1, sf_2, \dots, sf_n, 255 - tf_1, 255 - tf_2, \dots, 255 - tf_n, ff_1, ff_2, \dots, ff_n\}$, was extracted for each superpixel (both the training superpixels and test superpixels); and n denotes the number of pixels in the superpixel and sf_n , tf_n , ff_n denote the SF, TF, FF values of pixel n , respectively. As can be seen in Figure 5, the SF and FF values of a cloud pixel were very large, but the TF value was small. To promote the same trend in the experiments in this paper, $255 - tf_n$ was utilized.

(c) Features histogram.

Since the features of each superpixel are extracted, an appropriate vector to describe every superpixel is need. In this paper, the features histogram $STFH = \{f_0, f_1, \dots, f_M\}$ was counted for superpixels $S = \{s_1, s_2, \dots, s_O\}$ as the feature descriptor as shown in Figure 9. Where M is the maximum feature value in the imagery, since SF, TF, and FF were scaled to the range of (0, 255), M was set at 255 in this experiment. f_M denotes the percentage of value M in STF, and O denotes the number of superpixels. Thereafter, the matrix of characteristic frequency $N = (n(s_i, f_j))_{ij}$ was obtained, where $n(s_i, f_j)$ denotes the frequency of feature f_j appears in superpixel s_i . In addition, each (s_i, f_j) corresponds to a set of underlying themes $Z = \{z_1, z_2, \dots, z_K\}$. K indicates the number of themes, which was set at 20 in this paper. The effect of parameter K for cloud detection is analyzed in Section 4.2.

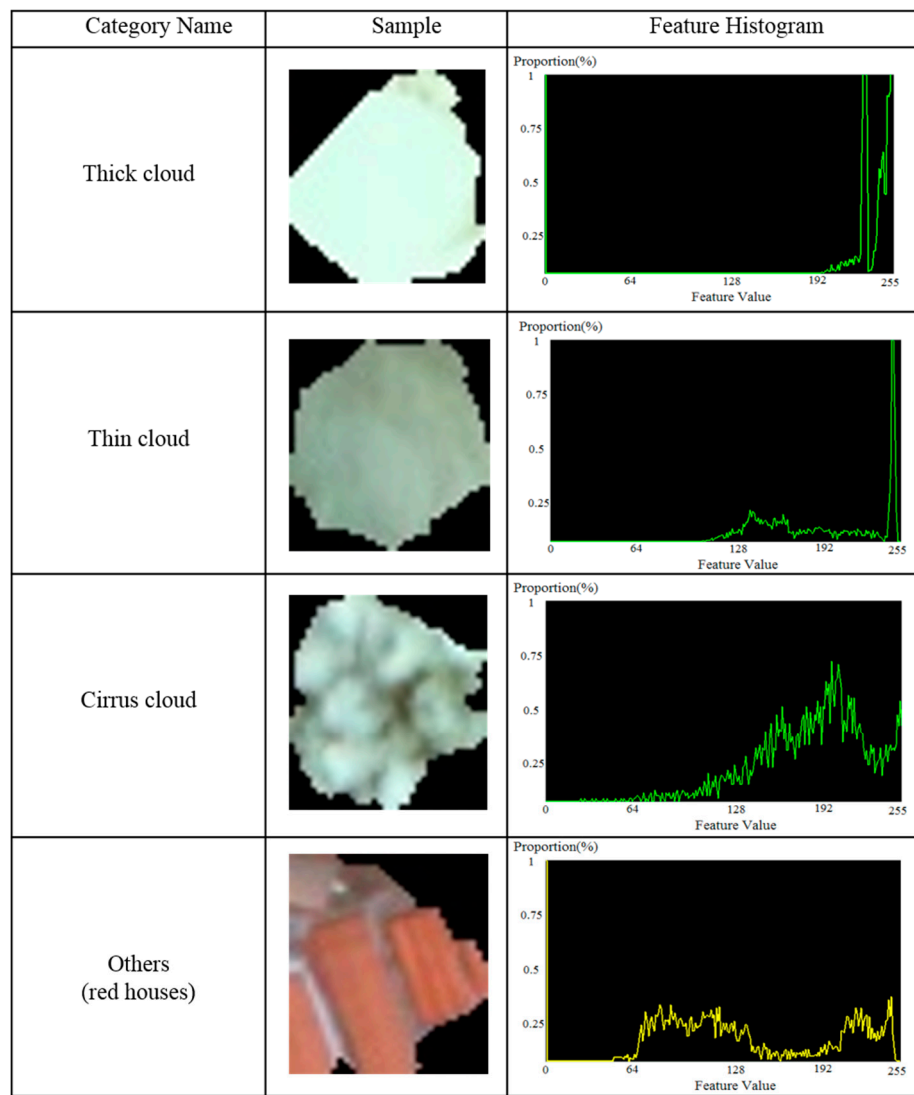


Figure 9. Four categories of superpixels (thick cloud superpixels, thin cloud superpixels, cirrus cloud superpixel, others) and their feature histogram STF. The histogram describes the SF, TF, and FF distribution of the superpixel, the green line histograms denote the positive superpixels, and the yellow ones are the negative superpixels. The x label of the feature histogram is the feature value, ranging from 0 to 255, and the y label is the proportion of the corresponding feature value.

(d) Semantic information extraction by PLSA.

This step includes model generation learning and model recognition learning.

(1) Model generation learning. To obtain the distribution pattern $P(f_j | z_k)$ of the features of the training superpixels when the latent semantic appears, the approximate solution of the PLSA model was calculated by Expectation-Maximization (EM) algorithm iteratively [30]. The E-step and the M-step are included in this process and are described below:

E-step, the posteriori probability of the underlying theme information for each superpixel $P(z_k | s_i, f_j)$ can be computed by

$$P(z_k | s_i, f_j) = \frac{[P(f_j | z_k) P(z_k | s_i)]}{\sum_{l=1}^K [P(f_j | z_l) P(z_l | s_i)]} \quad (13)$$

where $P(f_j | z_k)$ denotes the probability of the underlying theme z_k that appears in f_j , and denotes the probability of superpixel s_i appearing in underlying theme z_k . In this paper; $P(f_j | z_k)$ and $P(z_k | s_i)$ were initialized by using pseudorandom number generators.

M-step, re-estimation of $P(f_j | z_k)$, and $P(z_k | s_i)$ are based on the posteriori probability by

$$P(f_j | z_k) = \frac{\sum_{i=1}^N n(s_i, f_j) P(z_k | s_i, f_j)}{\sum_{m=1}^M \sum_{i=1}^N n(s_i, f_m) P(z_k | s_i, f_m)} \quad (14)$$

$$P(z_k | s_i) = \frac{\sum_{j=1}^M n(s_i, f_j) P(z_k | s_i, f_j)}{\sum_{j=1}^M n(s_i, f_j)} \quad (15)$$

where $P(z_k | s_i, f_m)$ denotes probability of the feature f_m of superpixel s_i appearing in the underlying theme z_k . The EM algorithm is used iteratively, until $E(L)$ (see Formula (16)) becomes stable, then the distribution pattern $P(f_j | z_k)$ of the features of training superpixels are obtained when the latent semantic appears.

$$(L) = \sum_{i=1}^N \sum_{j=1}^M n(s_i, f_j) \times \left(\sum_{k=1}^K P(z_k | (s_i, f_j)) \log[P(f_j | z_k) P(z_k | s_i)] \right) \quad (16)$$

(2) Model recognition learning. Using the distribution pattern $P(f_j | z_k)$ obtained, for any test superpixel s_{ℓ} , a K dimension semantic vector $P(z_k | s_{\ell})$ can be constructed to describe it. First, histogram feature vector $STFH_{\ell}$ of the superpixel is extracted. Then, $STFH_{\ell}$ and $P(f_j | z_k)$ are taken into the EM algorithm as described above. Finally, semantic vector $P(z_k | s_{\ell})$ is calculated, which has the same dimension with Z . Each value in the semantic vector represents the probability of this superpixel belonging to a corresponding theme, and the sum of them should be 1.0. If two superpixels are the same class, such as thick cloud, their semantic vectors will be similar; but the difference between the semantic vectors of a thick cloud superpixel and a vegetation superpixel will be very large.

(e) Superpixel recognition.

K dimensions semantic vector $P(z_k | s_{\ell})$ describes the superpixel, which then can be recognized by the machine-learning algorithm. A series of state-of-the-art algorithms recently were developed for remote-sensing image classification, such as SVM, Extreme Learning Machine (ELM), Decision Tree, Random Forest (RF) and Tree Bagger (TB). Huang, et al. (2015) experimentally verified that SVM outperformed other machine-learning methods in target extraction from remote sensing images. In addition, RBF-SVM demonstrated the best stability for non-linear classification [31]. In this paper, the result is a non-linear classification model that can be used to classify new test superpixel. Thus, the Radial Basis Function (RBF) kernel was selected as the kernel function of SVM and a five-fold cross-validation was conducted to determine the optimal parameters. The steps of recognition by SVM are as follows:

(1) Cloud superpixels are divided into four categories, thin cloud, thick cloud, cirrus cloud, and others. A set of training samples for each category are selected.

(2) The semantic vector of each training sample of each category is extracted by the method described as above (see model recognition learning).

(3) A classification model is obtained using the training samples by SVM.

(4) For any test sample, the semantic vector is extracted, then the sample will be recognized by using the classification model.

A series of GF-1 images were selected as the training examples. All of the images were segmented into superpixels, and a training set was obtained by visual interpretation. In the experiment, there were four categories: thick cloud, thin cloud, cirrus cloud, and others; and 600 positive samples (include 200 thick cloud superpixels, 200 thin cloud superpixels, and 200 cirrus cloud superpixels) and 1000 negative samples (forest, cities and towns, river, barren land, etc.) were collected in advance. Figure 9 shows the four categories and the corresponding features histogram STFHI discussed above.

The semantic vectors $P(z_k|s)$ of the superpixels were counted by PLSA, and the classification model was obtained by RBF-SVM, after which the test superpixels were recognized as one of the categories. In fact, the experiments determined that cirrus clouds were more difficult to distinguish from background superpixels than thick clouds and thin clouds. In Figure 6, it also can be seen that the feature values of SF and TF for cirrus cloud superpixels highly overlapped. Although the proposed method improved the accuracy of cirrus cloud superpixels recognition, 100% accuracy cannot be ensured. Thus, if superpixels were recognized as cirrus clouds, they were designated as possible foreground superpixels (PFS); if superpixels were recognized as thick or thin clouds, they were designated as foreground superpixels (FS); and if superpixels were recognized as others, they were designated as possible background superpixels (PBS).

In the experiments, large white buildings and snow were difficult to distinguish from thick clouds. According to the previous description (see Section 3.2.1), LSF is helpful in distinguishing thick clouds from human-made objects. Thus, in order to refine the cloud mask, if the line segments can be extracted in the FS, then they can be turned as BS forcibly. In addition, snow superpixels were found to always have a lower NIR than thick cloud superpixels. This phenomenon can be clearly observed in Figure 10, where 500 selected thick cloud superpixels and snow superpixels are shown, as well as the relationship between SF and NIR. A clearly boundary was visible between snow and thick clouds. The boundary superpixels were selected and fitted to the reference boundary by the Least Squares algorithm. Then, $ST = NIR - (1.73 \times SF - 219.2)$ was computed; if ST was smaller than 0, FS was turned to PBS forcibly.

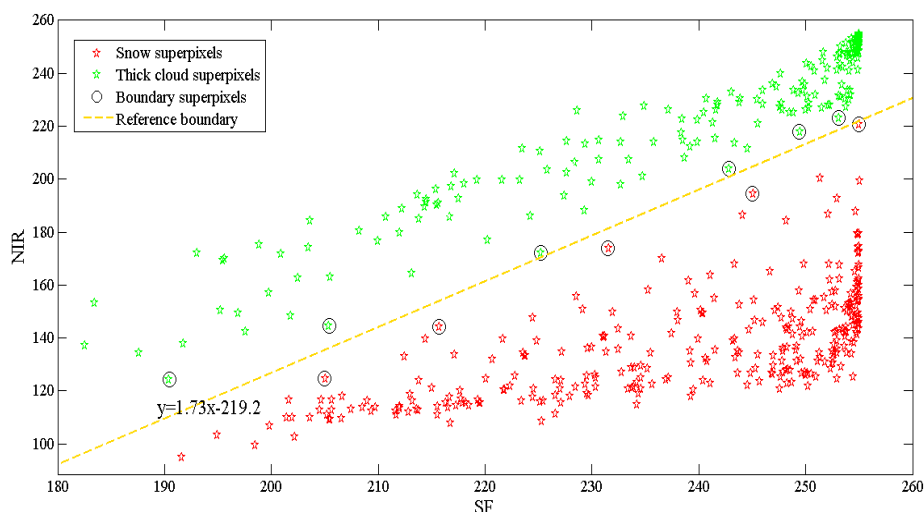


Figure 10. The features distribution of snow and thick clouds. Each pentagram represents a superpixel, plotted by the SF value on the horizontal axis and the NIR value on the vertical. The red pentagrams denote snow superpixels and the green ones are thick cloud superpixels. The circles are the boundary superpixels, and the yellow dotted line denotes the reference boundary fitted by the Least Squares algorithm.

According to the above processes, a good cloud mask can be obtained at the superpixel level. In this study, two typical scenes, city and snow, were selected as the test examples. The results of the cloud mask extraction are shown in Figure 11. It can be seen that the proposed method can generate satisfactory results for complex satellite images. Superpixels with large areas of clouds were accurately recognized as the interest targets, and the superpixels with small or no content of clouds were eliminated effectively. Furthermore, these processes were robust enough to exclude some problems that are difficult to solve with traditional spectral-based methods. As can be seen in Figure 11, the superpixels of white buildings, bright roads, frost, snow, and barren lands were effectively eliminated and semitransparent cloud superpixels were accurately extracted.

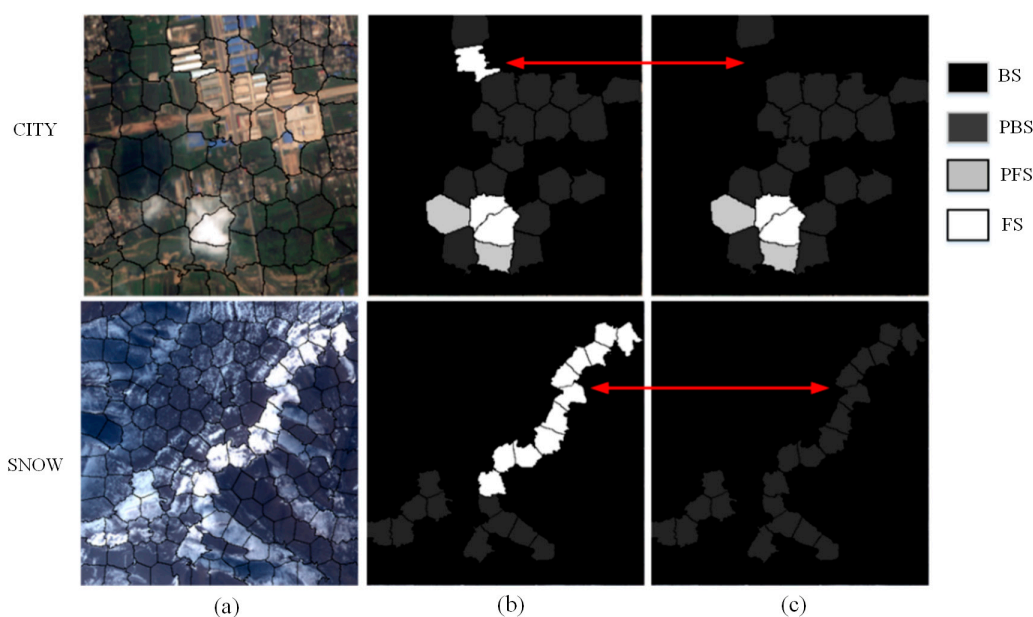


Figure 11. Example of cloud mask extraction using the proposed method. (a) Original image; (b) Cloud mask extracted by SVM. (c) Refined cloud mask. The white superpixels in (b) and (c) denote the foreground superpixels, the light gray ones denote possible foreground superpixels, the dark gray ones denote possible background superpixels, and the black ones denote the background superpixels. The red lines with double arrows between (b) and (c) point out the difference between before and after the refined cloud mask.

3.2.4. Accurate Cloud Detection

Coarse cloud detection results were obtained at the superpixel level using the methods described above. However, there was some missed semitransparent cloud pixels around the boundaries of the cloud regions because the above cloud detection processes belong to the superpixel recognition category. To improve the precision of cloud detection, the cloud pixels were further extracted accurately with the GrabCut algorithm at the pixel level. To obtain ideal foreground extraction results with GrabCut, initial prior knowledge should be provided to tell GrabCut which part of the image is the foreground, and which part is the background. In order to obtain more accurate result, possible foreground and possible background also should be provided. In this paper, clouds were foreground, others were background. The GrabCut algorithm builds a Gaussian mixture model (GMM) for foreground and background, and then splits the image by energy minimization. The GMMs [32] are initialized by the cloud mask extracted by the method described in this section. With the initial labels, GrabCut iteratively minimizes the energy function, which prefers the connected regions with the same label, until the optimal segmentation result was attained. In this study, the GrabCut algorithm was implemented using OpenCV library; and four examples of accurate cloud detection for GF-1 images are shown in Figure 12. More accurate cloud boundary detection results were obtained in the finally cloud detection results, the background pixels and possible background pixels all were recognized as non-cloud pixels, and the foreground pixels and possible foreground pixels all were recognized as cloud pixels. In the process of superpixel recognition, some superpixels with small areas of clouds were recognized as PBS and some superpixels with small area of non-clouds may be recognized as PFS, which are normal cases for the superpixels on the boundary of clouds. Fortunately, the GrabCut algorithm optimized this situation gradually at the pixel level, and improved the precision of the cloud boundary detection effectively. It can be seen in Figure 12 that the GrabCut algorithm repaired some of the hiatuses around the boundaries of the cloud regions, and most of the semitransparent cloud pixels also were successfully extracted.

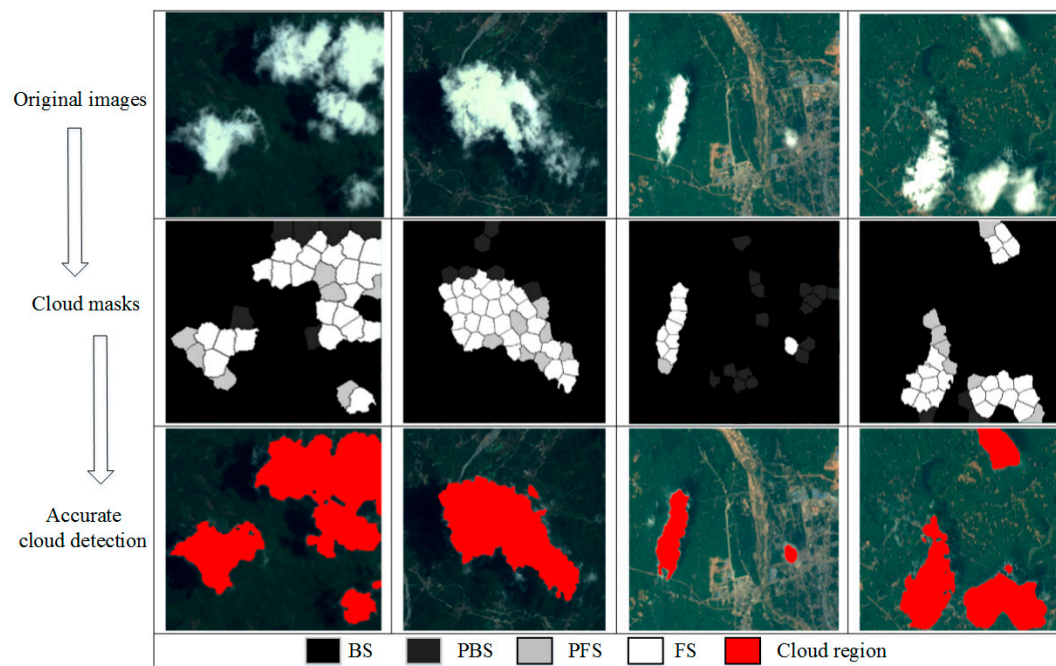


Figure 12. Example of accurate cloud detection by the proposed method. The first row shows the original images, the second row shows the cloud mask extracted by the method described in Section 3.2., and the third row shows the accurate cloud detection results.

4. Method Feasibility Validation

4.1. Internal Feasibility Validation

The proposed methodology detects clouds using STFH + PLSA + SVM + GrabCut, but the combination is not fixed. One of the algorithms can be removed or replaced with other algorithms in practical applications. To evaluate the feasibility of the proposed method, 36 scenes from ZY-3 were chosen. The selected scenes included a wide range of different cloud types and surface features at locations in China. The proposed method was compared to RGBH (RGB Histogram) + PLSA + SVM + GrabCut, STFH + SVM + GrabCut, and STFH + PLSA + SVM methods in the aspects of cloud detection precision and efficiency. A portion of the experimental results is shown in Figure 13.

The ground truths of actual cloud areas were manually extracted. The precision ratio, recall ratio and error ratio were regarded as the metrics, which were defined by the following formulas:

$$PR = \frac{TC}{FA} \quad (17)$$

$$RR = \frac{TC}{TA} \quad (18)$$

$$ER = \frac{TF + FT}{NA} \quad (19)$$

where PR denotes the precision ratio, TC denotes the number of correctly recognized cloud pixels, and FA denotes the number of recognized cloud pixels. RR denotes the recall ratio, and TA denotes the actual number of cloud pixels. ER denotes the error ratio, TF denotes the number of cloud pixels identified as non-cloud pixels, FT denotes the number of non-cloud pixels identified as cloud pixels, and NA denotes the total number of pixels in the image. These detection algorithms were implemented in C++ combined with OpenCV on a PC with a system configuration of Intel(R) Core(TM) i5-3230M CPU @2.60GHz and 4 gigabytes of memory. The average precision ratio, recall ratio, error ratio and runtime of the 36 scene images are shown in Table 3.

Table 3. Cloud detection performance analysis of internal feasibility.

Algorithm	Precision Ratio	Recall Ratio	Error Ratio	Runtime
Proposed method	87.6%	94.9%	2.5%	219 (s)
STFH + SVM + GrabCut	89.0%	82.8%	5.7%	107 (s)
RGBH + PLSA + SVM + GrabCut	82.7%	74.5%	7.6%	209 (s)
STFH + PLSA + SVM	88.1%	74.7%	7.2%	212 (s)

It can be seen that the proposed method produced the best precision ratio and recall ratio and its error ratio was lower than that of the other methods even though the proposed method is relatively time-consuming. STFH + SVM + GrabCut and STFH + PLSA + SVM obtained high precision ratios, their recall ratios were much lower than the proposed method, which means that many more cloud regions were missed during processing. The precision ratio of RGBH + PLSA + SVM + GrabCut was the lowest, which indicates that more non-cloud regions were misjudged as cloud regions than the others.

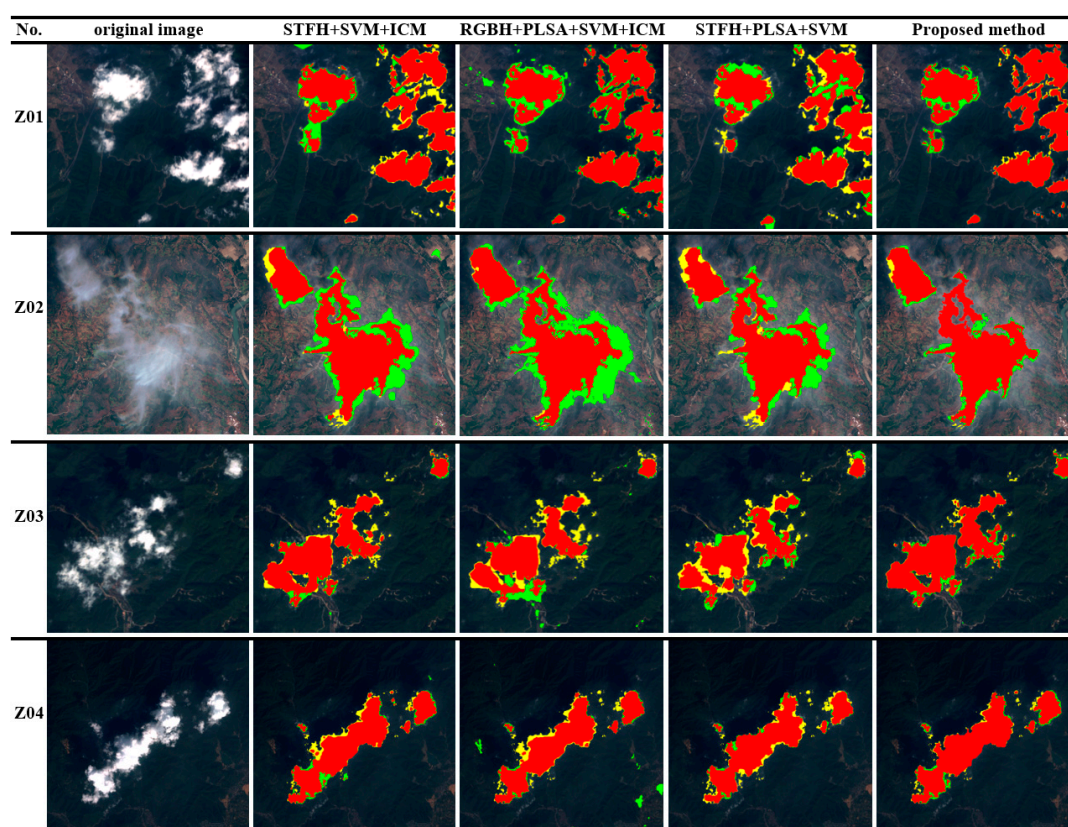


Figure 13. Cloud detection precision analysis of internal feasibility (Red denotes extracted true cloud regions, yellow denotes missed true cloud regions, and green denotes non-cloud regions misjudged as cloud regions).

4.2. Parameters Sensitivity Validation

The initial width of superpixels S and the dimension of PLSA semantic vector K are two important parameters in the proposed method. S was closely related to the efficiency and correctness of cloud detection, as shown in Figure 14, and K was closely related to the correctness of cloud detection, as shown in Figure 15. It can be seen from Figure 14 that the proposed method was relatively robust. Satisfactory results were obtained when S was in the range of 20 to 80, which is suitable for cloud detection in most cases. However, the error ratio increased remarkably if S was larger than 80 or smaller than 20.

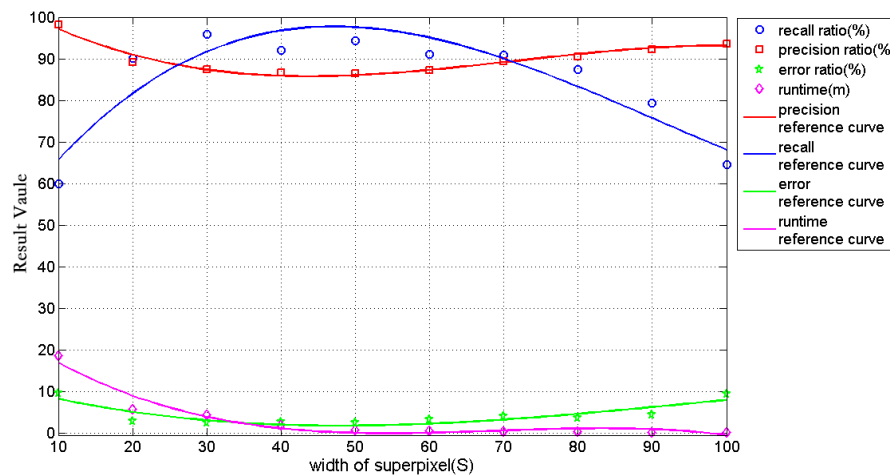


Figure 14. Effects of parameter S for cloud detection.

Figure 15 shows that very stable results were achieved when K was in the range of 20 to 65; and if K was larger than 65 or smaller than 20, the error ratio increased significantly. Based on this evaluation, the K value was set at the range of (20, 65), which is suitable for cloud detection in most cases.

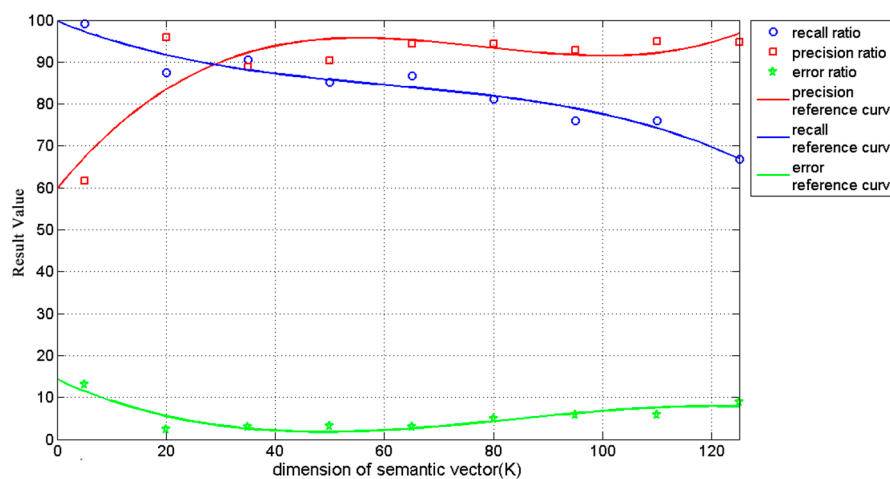


Figure 15. Effects of parameter K for cloud detection.

5. Experimental Results and Discussion

To fully evaluate the efficiency of the cloud detection algorithm, the proposed method was compared to some automatic cloud detection methods [18,22,23], some popular automatic image segmentation methods [33–35], and some interactive image segmentation methods [36,37]. Forty-two scenes of GF-1 images about 4500 pixels \times 4500 pixels in size were used in the experiments. The results were compared and evaluated by both visual inspection and detection accuracy statistics.

5.1. Case 1: Comparison with some Automatic Cloud Detection Algorithms

5.1.1. Methods and Results

First, the proposed method was compared with three other automatic cloud detection methods: k-means + ICM (Iterated Conditional Model) [18], RGB refinement [22] and SIFT + GrabCut [23], as shown in Figure 16. It can be seen that the visual effect of the proposed method was the best result among all four methods. Satisfactory results were quickly obtained by the k-means + ICM algorithm; however, this algorithm relies heavily on accurate initial segmentation results. Unfortunately,

ideal initial segment result may not be possible using only the k-means algorithm, so it dose misjudge, such as the frost on the ridge in G01, some artificial buildings in G03 that were identified as clouds, and parts of the semitransparent clouds in G02 that were identified as non-clouds. The algorithm of RGB refinement obtained coarse cloud detection results by using a global threshold, which were then refined by introducing a texture feature. RGB refinement was effective and worked well when the clouds were distributed evenly, such as in G04, but its results were unsatisfactory for the clouds that were distributed unequally, such as in G02 and G03. Furthermore, this algorithm was greatly affected by frost and snow, as shown in G01. The algorithm of SIFT + GrabCut is also an object-oriented cloud detection method and is innovative in that it separates the foreground from the background by using the GrabCut algorithm. However, sometimes the Harris points were difficult to extracted in the cloud regions, which may indicate that the SIFT feature was unsuitable for describing clouds at that time. As can be seen, parts of clouds were missed in G02, G03, and G04, and parts of frost in G01 were identified as clouds by SIFT + GrabCut algorithm.

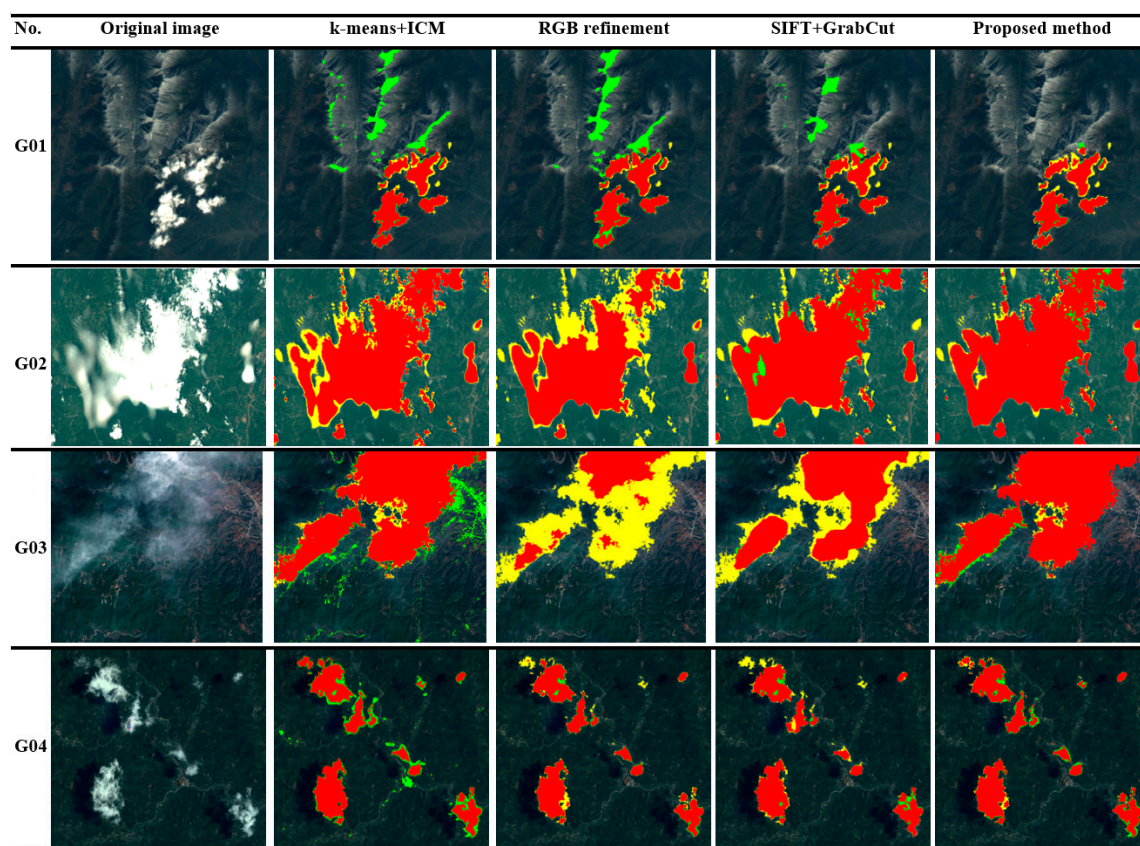


Figure 16. Results of comparison with automatic cloud detection algorithms and comparison of visually extracted clouds with those obtained by automatic detection algorithms (Red denotes the extracted true cloud regions, yellow denotes the missed true cloud regions, green denote the non-cloud regions misjudged as cloud regions).

5.1.2. Analysis and Discussion

The error ratios of the four algorithms in Figure 16 were shown in Table 4. In the four groups of experiments, the proposed method's error ratio was the lowest. For G01 and G02, other algorithms misjudged the frost for cloud pixels, and also were weak in detecting the semitransparent cloud pixels, so their error ratios were higher than the proposed method. For G03, the algorithm of RGB refinement and SIFT + GrabCut missed large areas of semitransparent clouds, so the error ratio reached 20.7% and 10.5%, respectively. For G04, the error ratio of k-means + ICM was quite high because parts of the

artificial buildings and roads were misjudged as clouds. The results show that the proposed method can detect all of the visually salient clouds while some small ones around the main clouds were missed, which possibly resulted from some of the superpixels with a few clouds around the main clouds had been defined as BS in the BS extraction step.

The k-means + ICM, RGB refinement, and SIFT + GrabCut methods can detect clouds only by RGB channels. K-means + ICM and RGB refinement are not only efficient but also can meet the requirement of real-time cloud detection. SIFT + GrabCut is time-consuming, but it is more robust than k-means + ICM and RGB refinement. Although the proposed method is time-consuming and more channel information is needed, one salient advantages was found: the proposed method is robust enough to exclude frost and detect thin clouds accurately. That is because our method uses spectral, texture, and frequency features synthetically. It also utilizes the PLSA model in the process to ensure the precision of superpixel recognition and establish a firm foundation for the GrabCut algorithm. Thus, robust results can be achieved by the proposed method.

Table 4. Error ratio (ER) comparison with automatic cloud detection algorithms.

Algorithm	ER of G01	ER of G02	ER of G03	ER of G04
k-means + ICM	3.8%	7.4%	6.0%	3.1%
RGB refinement	3.9%	12.0%	20.7%	1.6%
SIFT + GrabCut	2.7%	6.0%	10.5%	1.6%
Proposed method	1.5%	3.3%	2.4%	1.7%

5.2. Case 2: Comparison with Some Automatic Image Segmentation Methods

5.2.1. Methods and Results

Figure 17 shows the results of the comparison of the proposed cloud detection algorithm to three popular automatic image segmentation methods: k-means [33], meanshift [34], and Graph-based [35]. The k-means results were obtained by computing two clustering centers for the images based on intensity. The mean-shift or graph-based segmentation results were computed by first segmenting the input images in the mean-shift or graph-based segmentation algorithm and then detecting the cloud regions according to the average intensity of the region. Regions with an average intensity of more than 350 were judged as cloud regions. As shown in Figure 17, the proposed method demonstrated a distinct advantage. For G05, the other three automatic image segmentation methods did well in thick cloud areas, but failed to produce satisfactory results for several semitransparent cloud areas. The brightness of G06 was high and there were only small differences between the foreground and the background, so many areas were misjudged by all three automatic image segmentation methods. For G07, the very bright river and barren land were visually similar to cloud regions compared to the background and therefore were easily misjudged as clouds by the automatic image segmentation methods. Some small clouds were missed by the proposed method in G05 and G07, likely because the clouds were too thin to be detected, and some bright earth surfaces also were misjudged as clouds in G06.

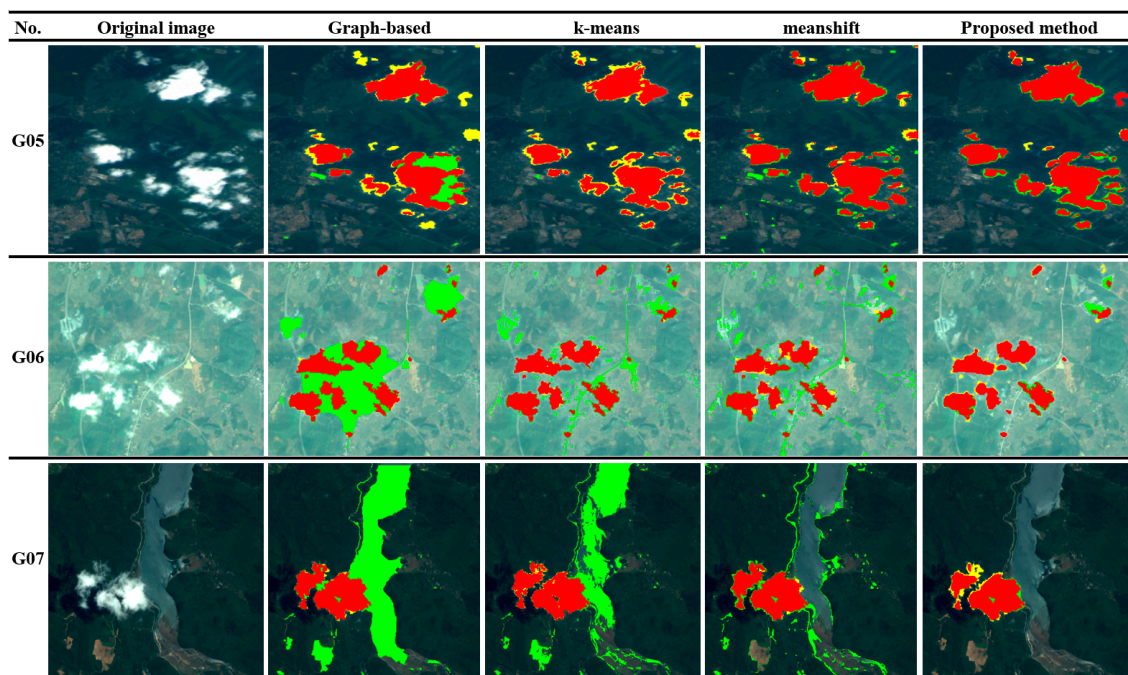


Figure 17. Results of comparison with automatic image segmentation algorithms (Red denotes the extracted true cloud regions, yellow denotes the missed true cloud regions, green denotes the non-cloud regions misjudged as cloud regions).

5.2.2. Analysis and Discussion

The error ratios of the four algorithms in Figure 17 are shown in Table 5. In the three groups of experiments, the proposed method had the lowest error ratio compared to the other three algorithms. For G05, the error ratio of the proposed method was 2.8%, while those of the other methods were all higher than 3% because they missed more small and thin clouds. For G06 and G07, the error ratio of the proposed method was less than the error ratios of the other methods, mainly because very bright objects, such as houses and rivers, were generally identified as clouds by the other three methods. However, although G06 had very bright areas and the non-cloud regions were visually similar to the cloud regions, they were quite different from the aspect of texture, and buildings with line segment features were defined as background in the BS extraction step. Thus, by incorporating the texture and line segment features into the proposed method, they were removed correctly. The river in G07 was very bright and had little detail, so it was difficult to distinguish it by using the ordinary features. Fortunately, the NIR band solved this problem effectively as the NIR value of water usually is low.

Table 5. Error ratio (ER) comparison with automatic image segmentation algorithms.

Algorithm	ER of G05	ER of G06	ER of G07
Graph-based	5.5%	12.8%	14.3%
k-means	3.6%	4.6%	10.6%
meanshift	3.4%	5.0%	4.0%
Proposed method	2.8%	1.7%	1.1%

5.3. Case 3: Comparison with Some Interactive Image Segmentation Methods

5.3.1. Methods and Results

As shown in Figure 18, proposed method was further compared with two interactive image segmentation methods: GrabCut [36] and Watershed [37]. GrabCut and Watershed were generated

using a supervised classification algorithm. In this paper, the results of GrabCut and Watershed were obtained by using OpenCV library, and masks were generated by manually delineating the cloud and non-cloud lines for each scene (the green scribbles indicate non-cloud regions, and the red scribbles indicate cloud pixels). One can see that the proposed method continued to have a distinct advantage. The GrabCut and watershed methods segmented images based on the gray value and space distance pixel by pixel, so the extracted cloud areas show integration and no speckle noise. However, the two methods did not accurately segment the boundaries of clouds. Not only the spectral and space distance were taken into account by the proposed method, but also the implicit texture and frequency information so cloud detection results were better than the other two methods.

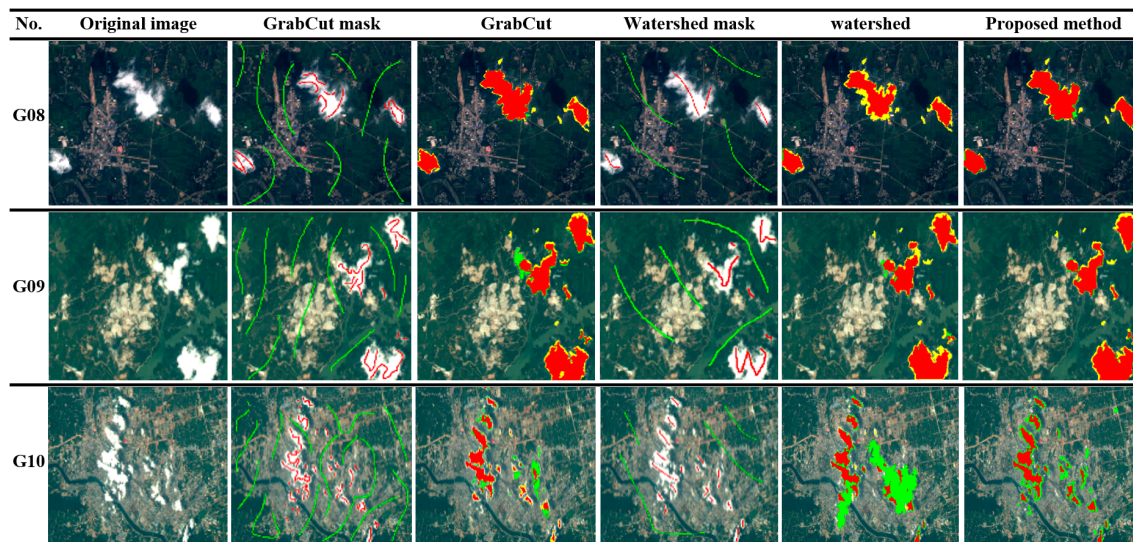


Figure 18. Results of comparison with interactive image segmentation algorithms (Red regions in the columns of GrabCut, watershed and the proposed method denote the extracted true cloud regions, yellow denotes the missed cloud regions, and green denotes the non-cloud regions that were misjudged as cloud regions. The green scribbles in the columns of the GrabCut mask and the Watershed mask indicate non-cloud regions, and the red scribbles indicate cloud pixels).

5.3.2. Analysis and Discussion

In order to compare the proposed method with the interactive image segmentation methods quantitatively, a work ratio was imported, which is defined as:

$$WR = \frac{DR}{NA} \quad (20)$$

where WR denotes the work ratio, DR denotes the number of user scribbled pixels, and NA denotes the total number of pixels in the image. The error ratio and work ratio of the three algorithms are shown in Table 6. One can see that the proposed method was slightly better than GrabCut and much better than watershed. Furthermore, the error ratio of GrabCut was lower than watershed, but the work ratio was relatively high. As for complex cloudy images like G09 and G10, both of them had white non-cloud regions. Even though sufficient interactions were provided for these complex images to perform GrabCut, it misjudged more non-cloud pixels as cloud pixels than the proposed method because the mask for GrabCut was not elaborate enough. In the proposed method, a high-precision mask that each pixel has its own accurate initial value which has been prepared for GrabCut, thus, better results were obtained by the proposed method.

Table 6. Error ratio and work ratio comparison with interactive image segmentation algorithms.

Algorithm	ER\WR of G08		ER\WR of G09		ER\WR of G10	
	ER	WR	ER	WR	ER	WR
GrabCut	1.3%	2.9%	2.8%	2.8%	3.1%	5.3%
Watershed	2.4%	1.2%	3.3%	2.9%	6.6%	1.6%
Proposed method	1.5%	0%	2.4%	0%	2.3%	0%

5.4. Algorithm Limitations

To detect cirrus clouds accurately from Chinese High Resolution Satellite Imagery that have too many details was very challenging for the proposed method. These cloud regions did not fully meet the common cloud features presented in Section 3.2.1, such as higher intensity, lower hue, and fewer details. Thus, the proposed method was relatively weak in detecting this type of cloud. One can see from Figure 19 that the proposed method was inadequate for detecting the cirrus clouds. As a result, the proposed method recognized some cloud regions as non-cloud regions. Cirrus cloud detection has always been an arduous problem for most of the cloud detection algorithms. As shown in Figure 19, the proposed method was further compared with k-means + ICM and RGB refinement. For Z05, some cirrus clouds were also missed by k-means + ICM, and some non-cloud regions were mistaken as clouds by RGB refinement. For G11, the problem of missing cirrus clouds also occurred in k-means + ICM and RGB refinement.

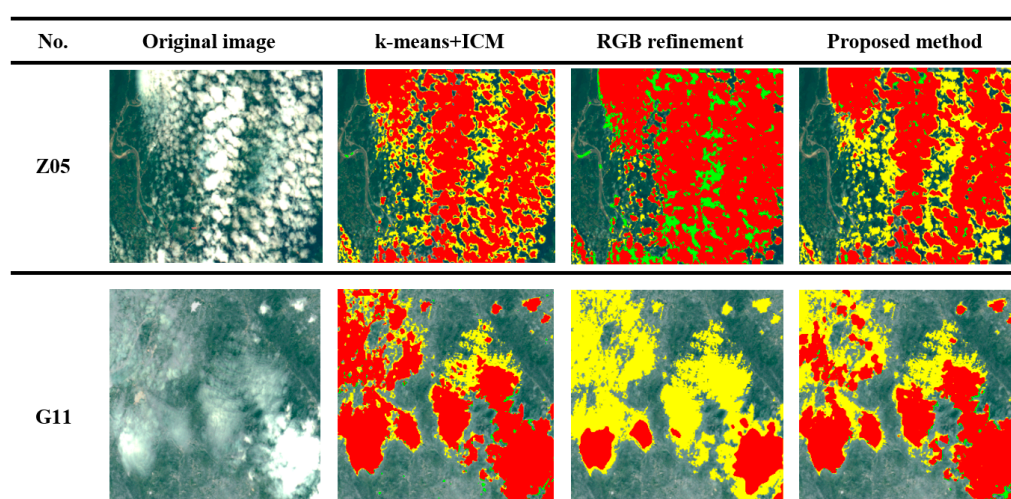


Figure 19. Results of comparison with other algorithms in the failed cases (Red denotes the extracted true cloud regions, yellow denotes the missed true cloud regions, green denotes the non-cloud regions misjudged as cloud regions).

The quantitative comparison of the proposed algorithm with other algorithms in the failed cases is shown in Table 7. The data indicate that, the error ratios of the three algorithms in the failed cases were more than 12.6% and the precision ratios were usually more than 80%, but the recall ratios were usually far below the precision ratio, which indicates that the problem of missing cirrus clouds is common for the most of the cloud detection algorithms. After the comprehensive comparison, k-means + ICM was found to be more suitable for cirrus cloud detection than the proposed algorithm. For Z05, RGB refinement obtained the best result (error ratio was 15.1%), but for G11, the error ratio was around 36.6%, so the algorithm was unstable in the cirrus cloud cases. Future research should continue to address this problem.

Table 7. Quantitative comparison with other algorithms in our failed cases.

No.	Algorithm	Precision Ratio	Recall Ratio	Error Ratio
Z05	Proposed method	97.5%	68.9%	18.9%
	k-means + ICM	99.8%	69.4%	18.3%
	RGB refinement	82.1%	96.1%	15.1%
G11	Proposed method	96.8%	62.7%	17.9%
	k-means + ICM	97.1%	78.5%	12.6%
	RGB refinement	99.9%	20.5%	36.6%

6. Conclusions

Accurate cloud detection for satellite images is a fundamental pre-processing step for a variety of remote sensing applications. This paper introduced a novel object-pixel two-level framework for cloud extraction using Chinese high resolution satellite imagery. The effectiveness of the proposed method was validated on ZY-3 and GF-1 images, both of which are typical Chinese high resolution satellite imagery. The advantages and the novelties of the proposed method are as follows:

(1) The algorithm introduced in this paper is based on probabilistic latent semantic analysis and object-based machine learning. Many ideas from past approaches in the literature were combined and the features of spectral, texture, frequency, and line segment are integrated. Besides, implicit information is extracted with PLSA model, which greatly improved the detection accuracy of the SVM algorithm at the superpixel level and effectively solved the problem of high-dimension in the machine learning field.

(2) The cloud detection results were further refined using the GrabCut algorithm at the pixels level, which effectively improved the cloud detection precision. A pixel-level accuracy assessment was performed, and the results show that the average precision ratio, recall ratio, and error ratio was about 87.6%, 94.5%, and 2.5%, respectively.

(3) In theory, for the satellite images that have R, G, B, and NIR channels and similar spatial resolution, such as ZY-3 and GF-1, it is only necessary to select a training dataset once to generate the classification model. No human interaction is needed in the subsequent cloud detection process.

(4) High accuracy results were achieved. Noises such as bright rivers, white buildings, snows and barren lands were effectively eliminated and semitransparent clouds were extracted accurately. The proposed method provided accurate cloud detection results for both Chinese high resolution satellite imagery and similar satellites with the same temporal and spectral settings.

Generally, the proposed method exhibited high accuracy for cloud extraction from Chinese high resolution satellite imagery and is a great improvement over the traditional spectral-based algorithm.

The proposed method has its limitations: First, PLSA and GrabCut are integrated in the cloud detect system, and both of them are iterative optimization algorithms. Thus, the proposed method is time-consuming when compared to traditional spectral-based methods. However, since the images are segmented into superpixels in the first step, it is still efficient compared to the traditional SVM methods that detect the target at the pixel level. Second, cirrus clouds with fine-texture were recognized as non-cloud areas due to the fact that cloud regions were designated in this method as having little detail. This problem will be addressed in the future work by constraining the weight of the texture feature.

Acknowledgments: This research was partially supported by the National Natural Science Foundation of China under grant 41322010, and National Hi-Tech Research and Development Program under grant 2013AA12A401. The authors would like to thank the editor and reviewers for their instructive comments that helped to improve this manuscript. Besides, they would also like to thank the international scientific data service platform and the image sky company of Suzhou for providing experimental datasets.

Author Contributions: Yongjun Zhang conceived of the study and designed the experiments; Xin Tong taken part in the research and analyzed the data; Kai Tan wrote the main program and most of the paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

PLSA	Probabilistic Latent Semantic Analysis
SLIC	Simple Linear Iterative Clustering
SVM	Support Vector Machine
GLCM	Gray-Level Co-occurrence Matrix
MRF	Markov Random Field
SPOTCASM	SPOT Cloud and Shadow Masking
HIS	Hue Intensity Saturation
SF	Spectral Feature
TF	Texture Feature
FF	Frequency Feature
LSF	Line Segment Feature
BS	Background Superpixels
PBS	Possible Background Superpixels
PFS	Possible Foreground Superpixels
FS	Foreground Superpixels
BOW	Bag-of-words
LSA	Latent Semantic Analysis
RBF	Radial Basis Function
GMM	Gaussian Mixture Model
ICM	Iterated Conditional Model

References

1. Zhang, Y.; Rossow, W.B.; Lacis, A.A. Calculation of radiative fluxes from the surface to top of atmosphere based on ISCCP and other global data sets. *J. Geophys. Res.* **2004**, *109*, 1121–1125. [[CrossRef](#)]
2. Zhang, Y.J.; Wan, Y.; Wang, B. Automatic processing of Chinese GF-1 wide field of view images. In Proceedings of the 36th International Symposium on Remote Sensing Environment, Berlin, Germany, 11–15 May 2015.
3. Chen, P.Y. An automated cloud detection method for daily NOAA-14 AVHRR data for Texas, USA. *Int. J. Remote Sens.* **2002**, *23*, 2939–2950. [[CrossRef](#)]
4. Moustakidis, S.; Mallinis, G.; Koutsias, N. SVM-Based fuzzy decision trees for classification of high spatial resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 149–169. [[CrossRef](#)]
5. Zhu, Z.; Woodcock, C.E. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sens. Environ.* **2012**, *118*, 83–94. [[CrossRef](#)]
6. Hagolle, O.; Huc, M.; Pascual, D.V. A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENμS, LANDSAT and SENTINEL-2 images. *Remote Sens. Environ.* **2010**, *114*, 1747–1755. [[CrossRef](#)]
7. Gao, X.J.; Wan, Y.C.; Zheng, X.Y. Real-Time automatic cloud detection during the process of taking aerial photographs. *Spectrosc. Spectr. Anal.* **2014**, *34*, 1909–1913.
8. Chylek, P.; Robinson, S.; Dubey, M.K. Comparison of near-infrared and thermal infrared cloud phase detections. *J. Geophys. Res.* **2006**, *111*, 4763–4773. [[CrossRef](#)]
9. Hong, G.; Zhang, Y. Wavelet-based image registration technique for high-resolution remote sensing images. *Comput. Geosci.* **2008**, *34*, 1708–1720. [[CrossRef](#)]
10. Agüera, F.; Aguilar, F.J.; Aguilar, M.A. Using texture analysis to improve per-pixel classification of very high resolution images for mapping plastic greenhouses. *ISPRS J. Photogramm. Remote Sens.* **2008**, *63*, 635–646.
11. Tran, D.B.; Puissant, A.; Badariotti, D. Optimizing spatial resolution of imagery for urban form detection—The cases of France and Vietnam. *Remote Sens.* **2011**, *3*, 2128–2147. [[CrossRef](#)]
12. Huang, X.; Liu, X.; Zhang, L. A Multichannel Gray level co-occurrence matrix for multi/hyperspectral image texture representation. *Remote Sens.* **2014**, *6*, 8424–8445. [[CrossRef](#)]
13. Ou, X.; Pan, W.; Xiao, P. In vivo skin capacitive imaging analysis by using grey level co-occurrence matrix (GLCM). *Int. J. Pharm.* **2014**, *460*, 28–32.
14. Liu, J. Improvement of dynamic threshold value extraction technic in FY-2 cloud detection. *J. Infrared Millim. Waves* **2010**, *29*, 288–292.

15. Huang, X.; Zhang, L. An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 257–272. [[CrossRef](#)]
16. Kaya, G.T. A hybrid model for Classification of remote sensing images with linear SVM and support vector selection and adaptation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 1988–1997. [[CrossRef](#)]
17. Shao, P.; Shi, W.; He, P. Novel approach to unsupervised change detection based on a robust semi-supervised FCM clustering algorithm. *Remote Sens.* **2016**, *8*, 264. [[CrossRef](#)]
18. Xu, X.; Guo, Y.; Wang, Z. Cloud image detection based on Markov Random Field. *Chin. J. Electron.* **2012**, *29*, 262–270. [[CrossRef](#)]
19. Fisher, A. Cloud and Cloud-Shadow Detection in SPOT5 HRG imagery with automated morphological feature extraction. *Remote Sens.* **2013**, *6*, 776–800.
20. Zhang, Y.; Guindon, G.; Li, X. A robust approach for object-based detection and radiometric characterization of cloud shadow using haze optimized transformation. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 5540–5547. [[CrossRef](#)]
21. Hu, X.; Wang, Y.; Shan, J. Automatic recognition of cloud images by using visual saliency features. *IEEE Geosci. Remote Sens.* **2015**, *12*, 1760–1764.
22. Zhang, Q.; Xiao, C. Cloud detection of RGB color aerial photographs by progressive refinement scheme. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7264–7275. [[CrossRef](#)]
23. Yuan, Y.; Hu, Y. Bag-of-Words and object-based classification for cloud extraction from satellite imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4197–4205. [[CrossRef](#)]
24. Zhang, Y.J.; Zheng, M.T.; Xiong, J.X. On-Orbit Geometric calibration of ZY-3 three-line array imagery with multistrip data sets. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 224–234. [[CrossRef](#)]
25. Ren, X.; Malik, J. Learning a classification model for segmentation. In Proceedings of the 2003 Ninth IEEE International Conference on Computer Vision, Washington, DC, USA, 13–16 October 2003.
26. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. *SLIC Superpixels*; EPFL Technical Report 149300; École polytechnique fédérale de Lausanne: Lausanne, Switzerland, June 2010.
27. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282.
28. Gioi, R.G.V.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 722–732.
29. Liao, Y.T.; Liao, Y.T. Hierarchical segmentation framework for identifying natural vegetation: A case study of the Tehachapi Mountains, California. *Remote Sens.* **2014**, *6*, 7276–7302. [[CrossRef](#)]
30. Hofmann, T. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.* **2001**, *42*, 177–196. [[CrossRef](#)]
31. Huang, X.; Xie, C. Combining pixel- and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *8*, 2097–2110. [[CrossRef](#)]
32. Bishop, C.M. Pattern recognition and machine learning. *J. Electron. Imaging* **2006**, *16*, 140–155.
33. Kanungo, T.; Mount, D.M.; Netanyahu, N.S. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [[CrossRef](#)]
34. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [[CrossRef](#)]
35. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **2004**, *59*, 167–181. [[CrossRef](#)]
36. Rother, C.; Kolmogorov, V.; Blake, A. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* **2004**, *23*, 309–314. [[CrossRef](#)]
37. Zhang, M.; Zhang, L.; Cheng, H.D. A neutrosophic approach to image segmentation based on watershed method. *Signal Process.* **2012**, *90*, 1510–1517. [[CrossRef](#)]

