Thin Cloud Detection of All-Sky Images Using Markov Random Fields

Qingyong Li, *Member, IEEE*, Weitao Lu, Jun Yang, and James Z. Wang, *Senior Member, IEEE*

***Abstract*—Thin cloud detection for all-sky images is a challenge in ground-based sky-imaging systems because of low contrast and vague boundaries between cloud and sky regions. We treat cloud detection as a labeling problem based on the Markov random field model. In this model, each pixel is represented by a combined-feature vector that aims at improving the disparity between thin cloud and sky. The distribution of each label in the feature space is defined as a Gaussian model. Spatial information is coded by a generalized Potts model. During the estimation, thin cloud is detected by minimizing the posterior energy with an iterative procedure. Both subjective and objective evaluation results demonstrate higher accuracy of the algorithm compared with some other algorithms.**

***Index Terms*—All-sky cloud image, cloud detection, Markov random fields (MRFs).**

I. Introduction

CLOUDS are crucially important in the atmospheric energy balance and the hydrological cycle. Most cloud-related research and applications require some ground-based cloud observations [1], such as cloud cover. Conventionally, the cloud cover is determined by human observers. Manual observation is often subjective and inconsistent. The shortcomings have led to the development of automatic cloud observation techniques, which utilize sky-imaging systems to capture sky visual conditions and to analyze cloud characteristics.

A typical sky-imaging system includes two main parts: a sky imager and an image analysis module. A sky imager is an optical device that automatically takes series of hemispheric sky images, called all-sky images, with a set time interval. Examples include the whole-sky imager series [2] and the total-sky imager series [1]. Fig. 1(a) shows the sky imager used in our study. Fig. 1(b) shows an all-sky image produced. An image analysis module processes all-sky images and determines cloud characteristics, e.g., cloud cover and cloud type. In this module, *cloud detection*, which is a process to classify each pixel of an all-sky image into "cloud" or "sky" elements, is a fundamental task because it is a precondition to determining cloud characteristics [1].

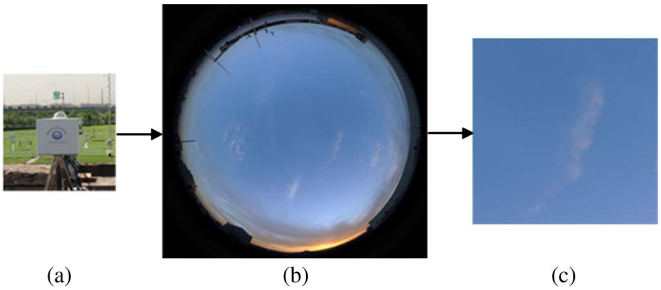

Fig. 1. Sky imager and all-sky cloud images. (a) WSC used in this study. (b) All-sky image produced by the WSC. (c) Cropped thin cloud image.

Existing cloud detection techniques are generally based on thresholding techniques, in which an red–green–blue (RGB) cloud image is transformed into a single-channel feature image and each pixel is classified by thresholding the feature. Common features for fixed thresholding algorithms include the ratio of red over blue (or blue over red) [1], saturation [3], and Euclidean geometric distance (EGD) [4]. Fixed thresholding methods, however, are not flexible for different sky conditions and sky imagers [1]. Alternatively, an adaptive thresholding method based on the Otsu algorithm was investigated [5]. In addition, Cazorla *et al.* proposed a model based on a neural network for cloud detection [6]. Although these methods achieved good performance in their corresponding circumstances, they mostly acknowledged that thin cloud detection remained a challenge [1], [3]–[6].

Thin cloud refers to a form of cloud that is light and somewhat transparent with low optical depth, such as a cirrus cloud. Fig. 1(c) shows an example of a thin cloud image. Thin cloud images have the following characteristics. First, thin cloud images have relatively low contrast between cloud and sky elements, unlike many other cloud genera. There is often an overlap between the distribution of sky and that of cloud within a single feature space. We compute the distribution of cloud and that of sky in our ground-truth data set (details about the data set are in Section IV) for the three features, namely, normalized blue/red ratio [5], saturation [3], and EGD [4], and show the distributions in Fig. 2. We observe notable overlaps between cloud and sky in all the three feature spaces. It implies that linear thresholding algorithms are not capable of accurately detecting thin cloud. As a result, nonlinear discriminative models in a higher feature space should be considered.

Second, thin cloud images are often piecewise smooth with a small number of regions, although there can be outliers caused by complicated sky conditions and electronic noise. In a piecewise smooth image, a pixel is not independent of others. On the contrary, it tends to have the same class (referring to

Manuscript received June 27, 2011; revised August 14, 2011; accepted September 15, 2011. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Date of publication November 7, 2011; date of current version March 7, 2012.

Q. Li is with the School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China (e-mail: liqy@bjtu.edu.cn).

W. Lu and J. Yang are with the Institute of Atmospheric Sounding, Chinese Academy of Meteorological Sciences, Beijing 100081, China (e-mail: wtlu@cams.cma.gov.cn; jyang@cams.cma.gov.cn).

J. Z. Wang is with the College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802-6823 USA, and also with the National Science Foundation, Arlington, VA 22230 USA (e-mail: jwang@ist.psu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2011.2170953

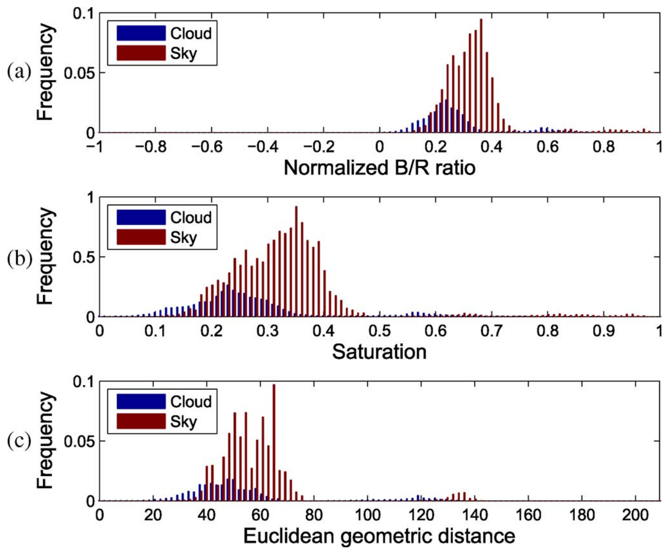

Fig. 2. Distributions of cloud and sky in three feature spaces. (a) Normalized blue/red ratio, (b) saturation, and (c) EGD.

cloud or sky in our context) with its neighboring pixels. Such a spatial constraint, usually modeled by Markov random fields (MRFs) [7], is proved to be useful for image segmentation, such as synthetic aperture radar image segmentation [8]. Therefore, we argue that thin cloud detection should take advantage of a smoothness constraint.

Cloud detection is essentially an application of image segmentation. Many sophisticated methodologies for image segmentation have been proposed recently, including maximum *a posteriori* (MAP)–maximum-likelihood (ML) estimation [9], hidden MRFs [7], Gaussian mixture model (GMM) [10], normalized cut (NC) [11], and level set (LS) method [12]. Their usefulness for cloud detection needs to be explored. In the MAP–ML estimation approach [9], data penalty for observed pixel features and smoothness penalty for hidden class labels of pixels are used together to model the probability of an image, and the MAP estimation and the ML estimation are iteratively applied to optimize the image segmentation. This method provides an effective resolution for image segmentation under the MRF framework.

Motivated by the challenges of thin cloud images and inspired by the principle of MAP–ML estimation [9], our work proposes a thin cloud detection algorithm under the MAP–MRF framework. In this algorithm, each pixel of a cloud image is characterized by combined features and modeled by a multivariate Gaussian model. The smoothness constraint is coded by an MRF model as *a priori*. Thin cloud is finally detected by minimizing the posterior energy with the iterative MAP–ML estimation.

The remainder of this letter is organized as follows. The feature representation is described in Section II. The cloud detection algorithm is proposed in Section III. Experimental results are presented in Section IV. Finally, we conclude in Section V.

II. Combined-Feature Representation

Feature representation of pixels is a fundamental factor for thin cloud detection. Three features, covering the physical, visual, and statistical properties, are used together in this work.

The first feature, denoted ro, is the normalized blue/red ratio, defined as $ro(s) = (b - r)/(b + r)$, where b and r are the intensities of the blue and red channels in the RGB color space and s is the pixel. While this feature is similar to the popular blue/red (or red/blue) ratio in [1] and [6], it is more suitable for adaptive thresholding algorithms [5]. The underlying principle of this feature comes from the scattering difference between air molecules and cloud particles. Clear sky is molecular Rayleigh scattering and scatters more blue light than red light, whereas cloud particles scatter the blue and red light almost equally [13]. Therefore, clear sky appears blue with high ro, and cloud appears white or gray with low ro.

The second feature takes the value of the saturation component of the hue–saturation–intensity color space as used in [3]. It is defined as $sa(s) = 1 - 3 \times \min(r, g, b)/(r + g + b)$. Generally, cloud has low sa, whereas clear sky often has high sa according to human visual perception.

The third feature is EGD defined as the distance between the color vector (r, g, b) of a pixel and the diagonal of the RGB cube [4]. EGD $ed(s)$ is given by

$$ed(s) = \sqrt{r^2 + g^2 + b^2 - \frac{(r + g + b)^2}{3}}.$$

This feature is supported by a statistical observation that cloud pixels distribute linearly and closely parallel to the diagonal of the RGB cube, whereas sky pixels locate farther from the diagonal. That is, cloud often has low ed while clear sky has high ed.

As shown in Fig. 2, a single feature was incapable of distinguishing cloud from sky for thin cloud images. Hence, we combine the three features and characterize a pixel s by a 3-D vector

$$F(s) = (ro(s), sa(s), ed(s)) . \tag{1}$$

Note that $ro(s)$, $sa(s)$, and $ed(s)$ range in [−1, 1], [0, 1], and [0, 208], respectively. They are all linearly rescaled to the interval [0, 255].

III. Cloud Detection Based on MRF Model

We describe the other aspects of cloud detection: the probability model of cloud images under the MAP–MRF framework and the cloud detection algorithm using MAP–ML estimation.

A. Probability Model of Cloud Images

Let S be a finite index set corresponding to N pixels of an all-sky image. Let $F = \{F_s | s \in S\}$ denote any family of observed pixel features and $C = \{C_s | s \in S\}$ be an unknown *configuration* of class labels of pixels with $C_s \in \{0, 1\}$ in which label 1 represents cloud and label 0 represents sky. According to the MAP–MRF principle [7], the posterior probability $P(C|F)$ of a configuration C given the image features F can be written as $P(C|F) \propto P(F|C)P(C)$, where $P(F|C)$ is the likelihood function and $P(C)$ denotes the prior probability modeled by MRFs in the next.

Likelihood function $P(F|C)$ is modeled by a Gaussian distribution. We adopt the typical assumption that random feature vectors F_s are conditionally independent given their class labels C_s, i.e.,

$$P(F|C) = \prod_{s\in S} P(F_s|C_s). \qquad (2)$$

The conditional distribution $P(F_s|C_s)$ is defined using the data penalty function [9] as follows:

$$P(F_s|C_s) \propto \exp\left(-D\left(F_s, \phi(C_s)\right)\right) \qquad (3)$$

where $\phi(C_s)$ is the parameter vector denoting class features of label C_s and is described by a 3-D vector

$$\phi(C_s) = \left(\overline{ro}(C_s), \overline{sa}(C_s), \overline{ed}(C_s)\right). \qquad (4)$$

The data penalty function $D(F_s, \phi(C_s))$, which measures the penalty of a pixel s labeled with C_s for a given $\phi(C_s)$, is defined as

$$\begin{aligned} D\left(F_s, \phi(C_s)\right) &= \|F_s - \phi(C_s)\|^2 \\ &= \left(ro(s) - \overline{ro}(C_s)\right)^2 + \left(sa(s) - \overline{sa}(C_s)\right)^2 \\ &\quad + \left(ed(s) - \overline{ed}(C_s)\right)^2. \end{aligned} \qquad (5)$$

The data penalty function essentially measures the Euclidean distance between the pixel feature F_s and the class feature $\phi(C_s)$. A larger distance leads to a lower probability of designating pixel s to a class C_s. Substituting (5) into (3), we find that $P(F_s|C_s)$ is a Gaussian distribution which takes the mean $\phi(C_s)$ and the unit covariance.

Second, we assume that a class configuration C is an MRF since all-sky images are almost piecewise smooth. Based on the Potts model and the eight connectivities [7], we describe the distribution of MRF C as

$$P(C) \propto \exp\left(-\sum_{s\in S}\sum_{t\in N(s)} V_{s,t}(C_s, C_t)\right) \qquad (6)$$

where $N(s)$ is the eight-pixel neighborhood of the pixel s. $V_{s,t}(C_s, C_t)$ denotes the clique potential function, which defines a smoothness penalty [9] of the clique (s, t) using a generalized Potts model as

$$V_{s,t}(C_s, C_t) = c\left(\exp\left(\frac{-D(F_s, F_t)}{\mu}\right)\right)\left(1 - \delta(C_s, C_t)\right) \qquad (7)$$

where $D(F_s, F_t)$ denotes the Euclidean distance between features of s and t as in (5) and μ denotes the mean of the distance of neighboring pixels in the whole image. $\delta(\cdot)$ is the Kronecker delta function, i.e., $\delta(C_s, C_t)$ is one if $C_s = C_t$ and zero if otherwise. As a result, the penalty function $V_{s,t}(C_s, C_t)$ enforces the spatial smoothness. If C_s and C_t are not equal, a penalty will be imposed. However, if the distance $D(F_s, F_t)$ is larger, the penalty will be smaller. Therefore, s and t tend to be labeled differently and form a boundary.

In terms of weight c that is used to tune the significance of the smoothness penalty, it is not easy to determine directly [7]. Based on our experiments, we determine that it can be a good choice to adaptively set c in order to put the smoothness penalty the same significance as the data penalty and set

$$c = \frac{\sum_{s\in S} \left|D\left(F_s, \phi(0)\right) - D\left(F_s, \phi(1)\right)\right|}{\alpha N} \qquad (8)$$

where α is a positive constant.

Finally, the posterior probability of $P(C|F, \Phi)$ given the observed features F can be derived by (2), (3), and (6) according to Bayes' rule as

$$\begin{aligned} P(C|F, \Phi) \propto &\prod_{s\in S} \exp\left(-D\left(F_s, \phi(C_s)\right)\right) \\ &\times \exp\left(-\sum_{s\in S}\sum_{t\in N(s)} V_{s,t}(C_s, C_t)\right). \end{aligned} \qquad (9)$$

Taking the negative logarithm of (9), we get the following posterior energy function:

$$E(C, \Phi) = \sum_{s\in S} D\left(F_s, \phi(C_s)\right) + \sum_{s\in S}\sum_{t\in N(s)} V_{s,t}(C_s, C_t). \qquad (10)$$

Under the MAP framework, the goal of cloud detection is, given the features F, to determine the class configuration C that maximizes the posterior probability in (9). It can be shown that maximizing $P(C|F, \Phi)$ is equivalent to minimizing the posterior energy $E(C, \Phi)$. In the next section, we will present the cloud detection algorithm by minimizing the posterior energy.

B. Cloud Detection Algorithm Using MAP–ML Estimation

Class labels C and model features $\Phi = \{\phi(0), \phi(1)\}$ in (10) are unknown, and they are strongly interdependent. In the literature, parameter estimation for such model is regarded as "incomplete-data" problem. Many techniques have been proposed to address this problem, such as the expectation–maximization algorithm [10] and iterative MAP–ML estimations [9]. Here, the MAP–ML estimation is adopted because it is more suitable for energy minimization.

First, if Φ is known, the optimal class configuration $\hat{C}$, which minimizes the posterior energy in (10) according to the MAP principle, is given by

$$\hat{C} = \arg\min_{C} E(C, \Phi). \qquad (11)$$

It is difficult to achieve the global optimal $\hat{C}$. Usually, it is obtained using a Markov-chain Monte Carlo method [7] and a graph cut algorithm [9]. Such methods, however, require a large amount of computation. Alternatively, approximate optimal solution can be efficiently achieved by well-established methods [7]. We adopt the iterated conditional mode (ICM) algorithm [14], which uses a greedy strategy in the iterative local minimization procedure and often achieves a convergence after only a few iterations.

Second, if the class configuration C is given, the optimal Φ should minimize $E(C, \Phi)$ with the principle of ML estimation. Since $D(F_s, \phi(C_s))$ is a quadratic function and $V_{s,t}(C_s, C_t)$ is

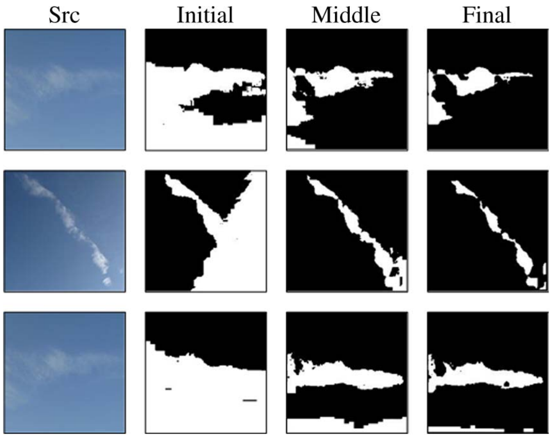

Fig. 3. Class configurations are updated during the iterative optimization. (From left) Original image, initial class configuration by adaptive thresholding, middle class configuration, and final class configuration.

independent of Φ, a closed-form solution for Φ can be obtained as follows:

$$\phi(i) = \frac{1}{Cnt(i)} \sum_{C_s = i} F(s) \tag{12}$$

where $Cnt(i)$ denotes the number of pixels labeled with i.

Finally, given the estimation C^n and Φ^n for the nth iteration, C^{n+1} and Φ^{n+1} can be iteratively estimated by

$$\Phi^{n+1} = \arg\min_{\Phi} E(C^n, \Phi) \tag{13}$$

$$C^{n+1} = \arg\min_{C} E(C, \Phi^{n+1}). \tag{14}$$

Note that initialization of C can be obtained by thresholding algorithms [1], [3]–[5].

As a summary, the cloud detection algorithm, based on the combined features and the MAP–ML estimation, is described as follows. In our experiment, $T = 0$, and $It_{\max} = 50$.

```
Algorithm 1 Cloud Detection
Input: an RGB cloud image, the convergence threshold $T$, and the maximum number of iterations $It_{\max}$.
1) Extract the features $F$ according to (1).
2) Initialize $C^0$ using the Otsu thresholding algorithm [5].
3) Iterative optimization.
3.1) Update $\Phi^{n+1}$ in (13) using (12) and the current $C^n$.
3.2) Estimate $C^{n+1}$ in (14) using the ICM algorithm [14] and the updated $\Phi^{n+1}$.
If $Ham(C^{n+1}, C^n) \leq T$, where $Ham(\cdot)$ denotes the Hamming distance and $T$ is the convergence threshold, or $It_{\max}$ is reached, go to the output step; otherwise, go to step 3.
Output: the class configuration $C$.
```

Fig. 3 shows examples of class configurations C for the initial, midpoint, and final estimations. This figure demonstrates that our algorithm can refine cloud regions with the assistance of the spatial smoothness constraint and the Gaussian model in the 3-D feature space.

TABLE I
SUBJECTIVE EVALUATION ON THE DATA SET

	Fixed	Adaptive	GMM	LS	NC	**Ours**
$\overline{N}_g$	32	43	35	45	1	**56**
$\overline{N}_m$	54	94	57	98	15	**97**
$\overline{N}_b$	114	63	108	57	184	**47**

IV. Experimental Results

The all-sky images used in this experiment were obtained by the whole-sky cameras (WSCs) [shown in Fig. 1(a)] which are located in Beijing (39.80° N, 116.47° E) and Conghua (23.57° N, 113.62° E), China, during the periods of May 2009 to June 2010. Because we focus on the cloud detection algorithm for thin cloud images, the zenith square area which is recognized as typical thin cloud by an expert is considered and cropped. In addition, some thin cloud images, which are manually acquired by several photographers with regular digital cameras, are also considered to increase the diversity of the image set. A data set with 200 images (resized to 200 × 200) is constructed to evaluate the performance; furthermore, 40 typical thin cloud images are selected and manually segmented into binary masks by the expert, and these 40 images are utilized as the ground truth.

The algorithms that we compared in our experiments include the following: fixed thresholding [1], adaptive thresholding [5], GMM [10], LS method [12], and NC [11]. The feature used in fixed thresholding, adaptive thresholding, and LS is ro defined earlier; the features defined by (4) are applied for GMM, NC, and our method. The threshold in the fixed thresholding method is set to 0.20 that is the best value after an exhaustive search in [0.1, 0.5] with a step of 0.02. The GMM and LS are initialized with the adaptive thresholding [5] just as our algorithm. The NC algorithm keeps the same as in [11] except that the features are replaced by (4) and the parameters are set as $\sigma_I = 1$, $\sigma_X = 15$, and $r = 10$. The parameters of LS are the default values in [12], and the number of iterations is 300. In our algorithm, the parameter α is set to $\alpha = 8$. Note that similar results can be achieved with $\alpha \in [6, 10]$.

A. Subjective Evaluation

We first visually evaluate the detection results. There are two reasons to estimate subjectively. First, some thin cloud images are too wispy to precisely generate ground truth, for example, cirrus fibratus is such a kind of clouds that are fibrous and curved like a mare's tail. Second, it is better to allow variation of criteria for cloud detection since thin cloud images have rather vague boundaries. Three experts who are experienced in cloud observation are asked to estimate the detection results. Each evaluator marks every detected result in the data set with a label, namely, good, medium, or bad, and records the number of results labeled with good (N_g), medium (N_m), and bad (N_b). The average numbers $\overline{N}_g$, $\overline{N}_m$, and $\overline{N}_b$ of the three evaluators are applied to measure the performance of the algorithms.

The values of $\overline{N}_g$, $\overline{N}_m$, and $\overline{N}_b$ for the six algorithms are given in Table I. From these results, we can see that our algorithm outperforms the other approaches because it obtains the largest $\overline{N}_g$ and the smallest $\overline{N}_b$. Among the other algorithms, LS and adaptive thresholding achieve a close performance to our algorithm, and NC is the worst.

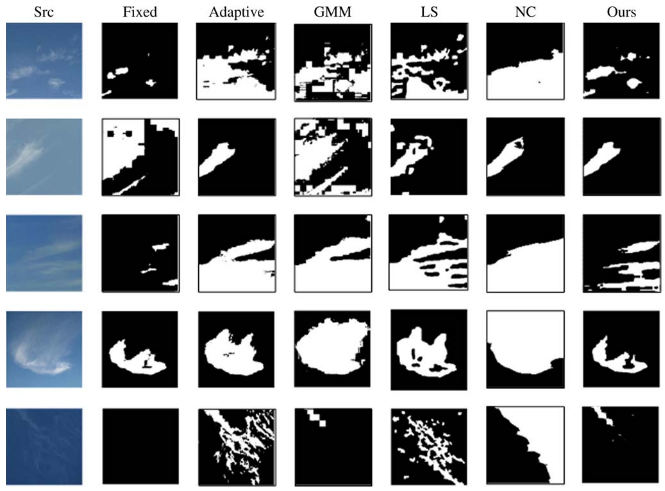

Fig. 4. Examples of the detection results with the six methods. The results of our method in the top three rows are labeled with good, while those of the fourth and the fifth rows are graded with medium and bad, respectively.

TABLE II
AVERAGE ACCURACY FOR THE SIX ALGORITHMS ON THE DATA SET

	Fixed	Adaptive	GMM	LS	NC	**Ours**
Ac	0.7767	0.8066	0.7976	0.8134	0.6466	**0.8340**

Fig. 4 shows some examples of the detection results. We have the following observations. The fixed thresholding method is not flexible for thin clouds. For example, the threshold is too small for the image in the first row, whereas it is too large for the one in the second row. NC tends to partition an image into regions of similar sizes and gives the worst segmentation mostly. The results generated by LS, adaptive thresholding, and GMM are exaggerated and noisy, although they visually obtain better results than fixed thresholding and NC. This is because adaptive thresholding and GMM ignore the spatial information, and LS takes only one feature into account although it considers spatial information. Our method takes spatial constraint and combined features into account and achieves compact and precise cloud regions in most cases as the top three rows. We noticed that our method has a risk of shrinking the cloud elements when they are too thin, as shown in the fifth image.

B. Objective Evaluation

To quantitatively compare the detection results, accuracy is applied to measure the performance of the algorithms. Accuracy is defined as $Ac = (TP + TN)/N_{\text{all}}$, where TP denotes the number of correctly detected cloud pixels, TN shows the number of correctly detected sky pixels, and N_{all} is the number of pixels in the detected image.

The average of the accuracy values for the six algorithms is given in Table II. We can observe that, by taking advantage of both the combined-feature space and spatial contextual information, our method is more accurate than the other five approaches. Among the other algorithms, LS, adaptive thresholding, and GMM achieve close accuracy to our method, and NC gives the worst performance.

V. CONCLUSION

This letter has put forward a thin cloud detection algorithm based on the MAP–MRF framework, taking combined features and spatial contextual information into account. In this algorithm, each pixel is represented by the combined features, including normalized blue/red ratio, saturation, and EGD, and the distribution of the observed features for a class is modeled by a multivariate Gaussian distribution. In addition, the spatial contextual information is coded with a generalized Potts model in the MRF framework. Cloud detection is formulated as a labeling problem, for which thin cloud is detected by minimizing the posterior energy with the MAP–ML estimation. Both the subjective and objective evaluation results verify that our algorithm is more accurate than some other methods.

REFERENCES

[1] C. N. Long, J. M. Sabburg, J. Calbo, and D. Pages, “Retrieving cloud characteristics from ground-based daytime color all-sky images,” *J. Atmos. Ocean. Technol.*, vol. 23, no. 5, pp. 633–652, 2006.

[2] J. E. Shields, R. W. Johnson, M. E. Karr, and J. L. Wertz, “Automated day/night whole sky imagers for field assessment of cloud cover distribution and radiances distributions,” in *Proc. 10th Symp. Meteorol. Observ. Instrum.*, Phoenix, AZ, 1998, pp. 165–170.

[3] M. P. Souza-Echer, E. B. Pereira, L. S. Bins, and M. A. R. Andrade, “A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera,” *J. Atmos. Ocean. Technol.*, vol. 23, no. 3, pp. 437–447, 2006.

[4] S. L. M. Neto, A. V. Wangenheim, E. B. Pereira, and E. Comunello, “The use of Euclidean geometric distance on RGB color space for the classification of sky and cloud patterns,” *J. Atmos. Ocean. Technol.*, vol. 27, no. 9, pp. 1504–1517, Sep. 2010.

[5] J. Yang, L. Weitao, M. Ying, Y. Wen, and L. Qingyong, “An automatic ground-based cloud detection method based on adaptive threshold,” (in Chinese), *J. Appl. Meteorol. Sci.*, vol. 20, no. 6, pp. 713–721, 2009.

[6] A. Cazorla, J. Olmo, and L. Alados-Arboledas, “Development of a sky imager for cloud cover assessment,” *J. Opt. Soc. Amer. A*, vol. 25, no. 1, pp. 29–39, Jan. 2008.

[7] S. Li, *Markov Random Field Modeling in Image Analysis*. New York: Springer-Verlag, 2009.

[8] M. Picco and G. Palacio, “Unsupervised classification of SAR images using Markov random fields and g_i^0 model,” *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 2, pp. 350–353, Mar. 2011.

[9] S. Chen, L. Cao, Y. Wang, J. Liu, and X. Tang, “Image segmentation by MAP–ML estimations,” *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2254–2264, Sep. 2010.

[10] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image segmentation using expectation–maximization and its application to image querying,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1026–1038, Aug. 2002.

[11] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[12] C. Li, C.-Y. Kao, J. C. Gore, and Z. Ding, “Minimization of region-scalable fitting energy for image segmentation,” *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1940–1949, Oct. 2008.

[13] J. M. Sabburg and C. N. Long, “Improved sky imaging for studies of enhanced UV irradiance,” *Atmos. Chem. Phys. Discuss.*, vol. 4, pp. 2543–2552, 2004.

[14] J. Besag, “On the statistical analysis of dirty pictures,” *J. R. Stat. Soc. Ser. B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.