Automatic Cloud Detection for All-Sky Images Using Superpixel Segmentation

Shuang Liu, Linbo Zhang, Zhong Zhang, Chunheng Wang, and Baihua Xiao

***Abstract*—Cloud detection plays an essential role in meteorological research and has received considerable attention in recent years. However, this issue is particularly challenging due to the diverse characteristics of clouds. In this letter, a novel algorithm based on superpixel segmentation (SPS) is proposed for cloud detection. In our proposed strategy, a series of superpixels could be obtained adaptively by SPS algorithm according to the characteristics of clouds. We first calculate a local threshold for each superpixel and then determine a threshold matrix for the whole image. Finally, cloud can be detected by comparing with the obtained threshold matrix. Experimental results show that our proposed algorithm achieves better performance than the current cloud detection algorithms.**

***Index Terms*—Cloud detection, superpixel segmentation (SPS), threshold matrix.**

I. Introduction

CLOUDS play an important role in hydrological cycle and affect the energy balance on local and global scopes via interacting with solar and terrestrial radiation. Most cloud-related research requires ground-based cloud observation, such as cloud cover. At present, cloud cover is still estimated by human observers at meteorological observation stations [1]. However, this method takes high cost in terms of human resources, and the results obtained from different observers are often inconsistent. Thus, automatic estimation techniques for cloud cover are eagerly required in this field.

To achieve this goal, some instruments for capturing ground-based clouds, such as whole sky imager [2], [3], total sky imager [4], and infrared cloud analyzer [5], have been developed. Those instruments could obtain continuous all-sky images with a set time interval. Moreover, a lot of algorithms are proposed to estimate the cloud cover based on these captured all-sky images. Cloud detection, which classifies each pixel of all-sky images into cloud or clear sky element, is needful for cloud cover estimation. Currently, most cloud detection algorithms treat color as the primary characteristic for distinguishing cloud from clear sky. This is due to the scattering difference between cloud particles and air molecules. More specifically, cloud particles scatter similar blue (B) and red (R) intensity, whereas clear sky scatters more B than R intensity [6]. Based on this, Long *et al.* propose a threshold algorithm for cloud detection according to a certain radio of R over B intensity using red-green-blue (RGB) image [6]. Specifically, pixels with R/B greater than 0.6 are identified as cloud, otherwise as clear sky. Kreuter *et al.* report a different threshold of 1.3 on the B/R radio for identifying cloud [7], which is illustrated to be a more suitable choice than the previously mentioned value of 0.6. Heinle *et al.* treat $R - B = 30$ as the threshold for cloud detection [8], where better performance is obtained than the radio techniques. This kind of methods can be referred to as the fixed threshold algorithm, i.e., each pixel in a cloud image is classified by a fixed value. Moreover, other algorithms using fixed thresholds are proposed, such as saturation [9] and Euclidean geometric distance [10]. However, the fixed threshold algorithms are unsuitable for different sky conditions. To overcome this shortcoming, Yang *et al.* propose to use a global adaptive threshold for cloud detection [1]. Although improved results can be achieved, this method is obviously sensitive to nonuniform illumination for the whole all-sky image. Then, Yang *et al.* propose another cloud detection algorithm based on local threshold interpolation [11], in which each cloud image is divided into several subimages with the same size according to the spatial position, as showed in Fig. 1(b). Then, they calculate the local threshold for each subimage. However, the rigid division is not flexible, as it is not adaptive for the diverse shape, size, and location of clouds.

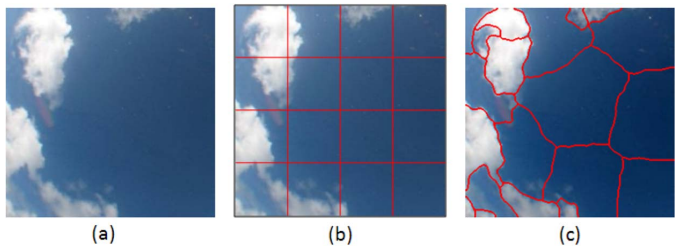

Fig. 1. (a) Original cloud image. (b) Rigid division of cloud image. (c) SPS of cloud image.

Cloud detection is actually treated as an application of image segmentation, and therefore, applying image segmentation techniques for cloud detection is a natural consideration [12]. Superpixel segmentation (SPS) [13] proposed by Ren and Malik is a popular algorithm in the field of image segmentation [14], [15]. The algorithm divides an image into a series of irregular image blocks, and each image block is called a superpixel. This irregular division is based on the texture similarity,

Manuscript received September 28, 2013; revised January 7, 2014 and May 21, 2014; accepted June 11, 2014. Date of publication August 8, 2014; date of current version August 21, 2014.

S. Liu and Z. Zhang are with the College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin 300387, China.

L. Zhang is with China Academy of Transportation Sciences, Beijing 100029, China.

C. Wang and B. Xiao are with the State Key Laboratory of Management and Intelligent Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

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.2014.2341291

brightness similarity, and contour continuity in an image. As a kind of natural texture, it is reasonable to describe the cloud appearance using contour and texture cues. In this letter, a novel algorithm based on SPS is proposed for cloud detection. In the framework of our cloud detection method, SPS can adaptively divide the cloud image into a series of superpixels according to the characteristics of clouds (shape, size, and location) shown in Fig. 1(c), which is designed to guarantee the consistency in each superpixel most of the time, i.e., each superpixel is covered by cloud or clear sky. Then, a local threshold value for each superpixel is obtained, and a threshold matrix is computed by interpolating the thresholds of all of the local thresholds. Finally, cloud detection is achieved by comparing an R-B feature image with the threshold matrix pixel by pixel.

The remainder of this letter is organized as follows. Section II introduces the data in our experiments and then proposes the cloud detection algorithm based on the SPS. Section III presents the experimental results, which show the superperformance of our algorithm. Finally, we conclude this letter in Section IV.

II. Data and Our Algorithm

A. Data

In this letter, the performance of our proposed SPS algorithm is evaluated on two ground-based cloud data sets. The first cloud data set is the Kiel data set provided by Kiel University, Kiel, Germany. The main equipment used to capture the all-sky images is a digital camera equipped with a fisheye lens, which can provide a field of view larger than 180°. Its aperture is f/1.2, the exposure time is 1/200, ISO is 100, and focal length is 8 mm. The camera is set to capture one cloud image per 15 s. More information about the camera can be found in [16]. In this letter, the square area of an all-sky image is considered and cropped, as shown in Fig. 1(a). Then, each image can be resized to 200 × 200 pixels. A set of 500 images (Kiel for short) is constructed to evaluate the performance of the proposed SPS algorithm. In addition, we also verify our method on the more challenging IapCAS data set, which is provided by IAPCAS. The capturing conditions are different from that of Kiel. Its aperture is f/1.4, the exposure time is 1/250, ISO is 100, and focal length is 6 mm. The camera is set to capture one cloud image per 10 s. Similarly, we also crop the square area of an all-sky image and resize it to 200 × 200 pixels. A set of 300 images (IapCAS for short) is constructed for our experiment, which can verify the robustness and versatility of the SPS algorithm effectively.

B. SPS Algorithm

In this letter, a novel cloud detection algorithm based on the SPS is proposed. In SPS, the detection results can be obtained by two steps. First, we apply SPS on ground-based cloud images, which acts as a kind of clustering algorithm. As the SPS based on the characteristics of the image, such as contour and texture, the pixels in each superpixel could guarantee the consistency most of the time, i.e., the pixels in each superpixel will belong to the same group (cloud or clear sky) as much as possible. Then, we further detect cloud based on these superpixels.

1) SPS: SPS [13] divides an image to a series of irregular blocks according to texture similarity, brightness similarity, and contour continuity. For a ground-based cloud image, the segmentation can be treated as a graph partitioning problem. The nodes of the graph are the entities that we want to partition, and here, they are the pixels; the edges between two nodes correspond to the similarity with which these two nodes belong to one group. Let $G = \{V, E\}$ be a weighted undirected graph, where V denotes the nodes and E denotes the edges; the segmentation can be formulated by the following objective function:

$$y = \arg\min Ncut = \arg\min_{y} \frac{y^T(D-W)y}{y^T D y} \tag{1}$$

where $W = \{w_{ij}\}$ is the association matrix and w_{ij} is the weight between nodes i and j in the graph. D is the diagonal matrix, where $D_{ii} = \sum_j w_{ij}$, i.e., D_{ii} is the sum of the weights of all of the connections to node i. $y = \{a, b\}^N$ is a binary indicator vector specifying the group identity for each pixel, i.e., $y_i = a$ if pixel i belongs to group A, and $y_i = b$ if pixel i belongs to group B. Here, A (or B) is a partition of the graph nodes; combining A and B is the set of all nodes. N is the number of pixels.

From the objective function of SPS (1), we know that the segmentation fundamentally depends on the weights (w_{ij}), which should be large for pixels from the same group, while small otherwise. Since contour and texture are the main characteristics of clouds, the weight is defined as the product of the contour and texture cues [13], i.e., $w_{ij} = w_{ij}^C \times w_{ij}^T$. w_{ij}^C denotes contour cue, which is adopted from the intervening contour method; w_{ij}^T denotes texture cue, which is computed by χ^2 test. Normalized cut algorithm [17] is applied to solve (1).

After applying the SPS algorithm, it can be seen that each cloud image can be adaptively divided into a series of superpixels according to the characteristics of clouds, as shown in Fig. 1(c). In most situations, the pixels maintain a high degree of consistency in each superpixel, i.e., they belong to cloud or clear sky elements.

2) Cloud Detection Scheme: In this section, we further detect cloud on the basis of SPS. The pseudocode of our proposed SPS is presented in Algorithm 1.

Algorithm 1: SPS algorithm

Input: an RGB cloud image, a division of cloud image after applying superpixel segmentation.
Output: detection result.
- **step1**: transfer RGB image to R-B feature image, which is simply difference of the R-channel and B-channel;
- **step2**: normalize the elements of R-B feature image to the range of $0 \sim 255$;
- **step3**: calculate local threshold for each superpixel on the R-B feature image;
- **step4**: calculate the threshold matrix by bilinear interpolation using the local thresholds of all of the superpixels;
- **step5**: compare the threshold matrix with R-B feature image pixel by pixel to get the final detection result.

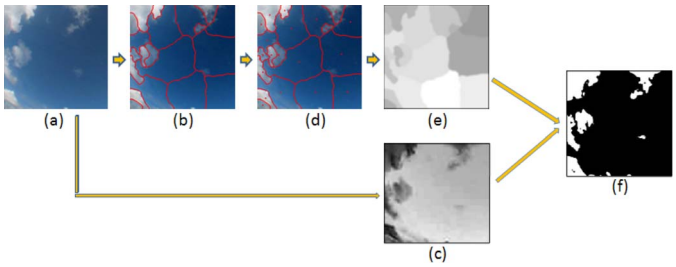

Fig. 2. Whole cloud detection process in this letter. (a) Original image. (b) Superpixels. (c) R-B feature image. (d) Location for threshold of all superpixels. The red dots are the location of local thresholds. (e) Threshold matrix. (f) Detection result.

To implement the cloud detection algorithm, we need to calculate the local threshold for each superpixel and the threshold matrix. Specifically, the local threshold can be obtained by the following four steps.

1) Compute the global threshold value T_g on the feature image by the Otsu algorithm [18], which is based on the different distributions of cloud and clear sky elements in a cloud image, i.e., the distinct bimodal distribution in the histogram derived from the cloud image. We need to find a value which can maximize the variance between cloud and clear sky elements. Formally, the problem can be solved as

$$V(T) = P_C P_S(\mu_C - \mu_S) \tag{2}$$

where P_C and P_S are the occurrence probabilities of the cloud and clear sky elements, respectively. μ_C and μ_S are the mean of cloud and clear sky, respectively. Finally, T_g is set to be $\max\{V(T)\}(1 < T < 255)$.
2) Calculate the maximum value $W_{\max}$ and minimum value $W_{\min}$ for the whole feature image.
3) Calculate the maximum value $L_{\max}$ and minimum value $L_{\min}$ for each superpixel.
4) Determine the local threshold for each superpixel by the following criteria:

$$\begin{array}{ll} T_l = L_{\max}, & \text{if } L_{\max} < T_g \\ T_l = L_{\min}, & \text{if } L_{\min} < T_g \\ T_l = \frac{1}{2}S_l + \frac{1}{2}T_g, & \text{otherwise} \end{array}$$

where S_l refers to the threshold of the superpixel, which is also computed by Otsu algorithm.

After getting the local threshold T_l for each superpixel, the corresponding threshold matrix based on all local thresholds can be calculated. In our experiment, we treat the center of each superpixel as the location of local threshold. Based on this, a threshold matrix with the same size as the R-B feature image is obtained by bilinear interpolation.

Based on the threshold matrix, we can implement cloud detection in the cloud image. In particular, we compute the difference between the threshold matrix and feature image pixel by pixel. If the result is greater than 0, the corresponding pixel is classified as a cloud element, otherwise as a clear sky one.

Fig. 2 shows an example for the whole procedure of our cloud detection scheme.

Several advantages of our proposed SPS algorithm can be concluded.

1) Our SPS can adapt with the diverse shape, size, and location of clouds and can adaptively divide the image into a series of irregular regions.
2) We use R-B as feature image, which can increase the differences of cloud and clear sky elements.
3) For each superpixel, we calculate its local threshold, which is suitable for all kinds of cloud images, especially when cloud and clear sky pixels exist in the same superpixel simultaneously.
4) The threshold matrix is calculated by bilinear interpolation, which can guarantee the smoothness of thresholds between different superpixels and avoid the boundary effect in cloud detection. Correspondingly, better detection results should be achieved.

III. Experimental Results

To evaluate the efficiency of the proposed SPS algorithm for cloud detection, we carry out a series of experiments on the data sets mentioned in Section II-A, which includes 500 and 300 samples, respectively. Furthermore, meteorological experts manually segment the cloud images from two data sets into binary masks, which are assigned as the ground truth, for objective evaluation.

A. Methods for Comparison

In this letter, SPS will be compared with three current cloud detection methods. In order to make the results comparable to others, we use R-B as feature image for all of the experiments.

1) Fixed threshold (fixed) [8]: we assign $\mathrm{R} - \mathrm{B} = 30$ as threshold. In particular, pixels with $\mathrm{R} - \mathrm{B} > 30$ are identified as cloud elements, otherwise as clear sky ones.
2) Global threshold (global) [1]: the global threshold is calculated by Otsu algorithm, which is described in (2).
3) Local threshold interpolation (local) [11]: the local threshold interpolation algorithm is applied on the R-B feature image, as introduced in Section I. Here, each feature image is divided into 16 subimages.

The computational complexity can be analyzed as follows.

1) "Fixed": This method use R-B as the threshold; thus, the computational complexity of "fixed" is linear, which can be regarded as a constant.
2) "Global": As the computation of the "global" method is mainly based on (2), the computational complexity of "global" is m^2 (m is the number of pixels in a detected cloud image) [18].
3) "Local": The computational complexity of the "local" method is mainly determined by two facts: the first fact is the number of divided subimages in a cloud image, which is set to be 16 in this letter; the second fact is mainly based on (2).
4) "SPS": The computational complexity of "SPS" mainly based on two facts: the SPS algorithm, which is shown in (1); the second fact is based on (2).

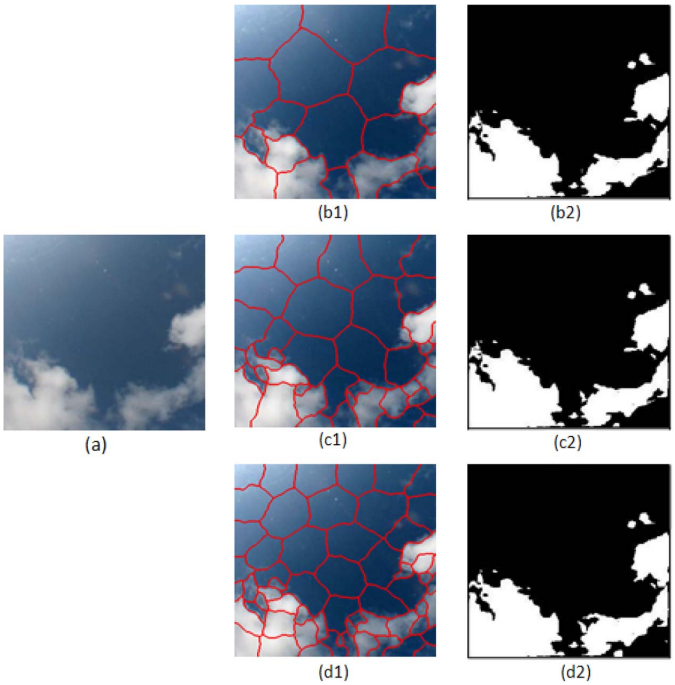

Fig. 3. (a) Original cloud image. (b1) Superpixel blocks obtained with $N = 20$. (c1) Superpixel blocks obtained with $N = 40$. (d1) Superpixel blocks obtained with $N = 60$. (b2) Detection result with $N = 20$. (c2) Detection result with $N = 40$. (d2) Detection result with $N = 40$.

TABLE I
SUBJECTIVE EVALUATION FOR KIEL DATA SET

	Fixed	Global	Local	SPS
$\overline{N_g}$	135	155	204	301
$\overline{N_m}$	104	113	126	137
$\overline{N_b}$	261	232	170	62

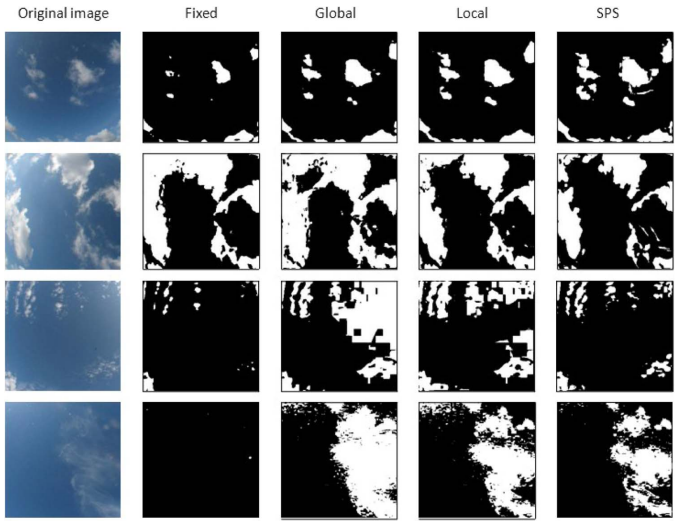

Fig. 4. Examples of detection results with four algorithms.

For a more obvious comparison, each method is realized using MATLAB on a 750-MHz Pentium III machine. The results show that the processing times used for each image by "fixed," "global," "local," and "SPS" are 0.4, 1, 1.2, and 2.6 s, respectively. Although more time is consumed by our "SPS," the highest accuracy rate is obtained. In addition, as the cloud-measuring devices on the ground taking one image of the clouds every 15 s in our experiment, our approach has no impact on the efficiency of the cloud detection process.

B. Results and Analysis

We first verify the performance of the SPS algorithm on Kiel data sets. Before presenting the experimental results, the parameter N, which is the number of superpixels for each cloud image, should be analyzed. An example about the influence of N is shown in Fig. 3. From the division results with different N values [Fig. 3(b1), (c1), and (d1)], we observe that, with the increase of N, better consistency in each superpixel is achieved but with more computational time. In addition, we calculate the local threshold for each superpixel in the following step, which can make up the differences caused by different N values in a certain degree. From Fig. 3(b2), (c2), and (d2), it can be seen that similar detection results with different N values are obtained. Therefore, N is set to be 20 empirically in our research.

After discussing the influence of parameter N, we show the experimental results in the following sections. Two ways are carried out for evaluating the detection results: subjective evaluation and objective evaluation.

1) Subjective Evaluation: We first evaluate the detection results by observation. There are two reasons to evaluate subjectively. First, there are some thin clouds and break clouds in cloud images, which is difficult to generate ground truth precisely. Second, cloud cover evaluation by human observers is the major mode for meteorological observations. Considering the subjectivity differences between different observers, 15 experts who have rich experience in cloud observation are asked to evaluate the detection results. Each expert labels the detected results for images in the data set with good, medium, or bad and records the number of results 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 15 experts are used to assess the performance of each detection algorithm.

Table I lists the values of $\overline{N_g}$, $\overline{N_m}$, and $\overline{N_b}$ for the four cloud detection algorithms. From the results, we can see that our proposed SPS algorithm achieves better detection results than the other three methods as our method gets the highest value for $\overline{N_g}$ and the lowest value for $\overline{N_b}$. In addition, $\overline{N_g} = 301$ illustrates that SPS can achieve the most consistent results with human observations.

Fig. 4 shows some examples of the results by the four detection algorithms. From the results, several conclusions can be drawn. First, the fixed threshold algorithm is not adaptable for different sky conditions. For example, the threshold is too large for the first row, whereas it is too small for the second row. Second, the performance of the global threshold algorithm is better than the fixed threshold algorithm; however, it cannot solve the conditions with nonuniform illumination. For instance, the detection results in the second row. Third, the results generated by the local threshold interpolation algorithm achieves closely similar results with our method; however, its performance is impaired when there is a break or thin clouds in the detected cloud images, which are shown in the third and fourth rows. In a

TABLE II
AVERAGE DETECTION ACCURACY FOR THE FOUR ALGORITHMS ON THE KIEL DATA SET

	Fixed	Global	Local	SPS
precision	0.69	0.74	0.82	0.94
recall	0.67	0.70	0.79	0.92
F_value	0.68	0.72	0.80	0.93

TABLE III
SUBJECTIVE EVALUATION FOR THE IAPCAS DATA SET

	Fixed	Global	Local	SPS
$\overline{N}_g$	64	79	88	129
$\overline{N}_m$	87	93	105	93
$\overline{N}_b$	149	128	107	78

TABLE IV
AVERAGE DETECTION ACCURACY FOR THE FOUR ALGORITHMS ON THE IAPCAS DATA SET

	Fixed	Global	Local	SPS
precision	0.59	0.66	0.75	0.84
recall	0.57	0.63	0.72	0.81
F_value	0.58	0.64	0.73	0.82

word, our method obtains the best performance. This is because our proposed SPS algorithm considers the characteristics of clouds and can divide cloud images adaptively.

2) Objective Evaluation: Quantitative evaluations of all of the aforementioned algorithms are also performed in this letter. Three criteria are selected to measure how well each algorithm matches the ground truth: precision, recall, and F-score. Precision and recall are defined as follows: precision $= TP/(TP + FN)$ and recall $= TP/(TP + FP)$, respectively. Here, TP, FP, FN and are the number of correct cloud pixels, false cloud pixels, and false sky pixels in a detected image. In addition, the F-score is defined as follows: $F = (2 \cdot \text{recall} \cdot \text{precision})/(\text{recall} + \text{precision})$. The F-score measures the detection accuracy by considering both the recall and precision. The higher the values of the three criteria are, the better the algorithm performs. Table II lists the values of the three criteria by the four detection algorithms. We can see that SPS achieves the highest detection accuracy for all of the three criteria.

We also verify our method on IapCAS data set, which includes 300 cloud images. Similarly, both the subjective and objective evaluations are carried out. Tables III and IV show the subjective and objective results, respectively. Once again, the best results are achieved by our strategy.

IV. CONCLUSION

In this letter, we have proposed an automatic cloud detection algorithm based on SPS. Compared with the fixed threshold, global threshold, and local threshold interpolation algorithms, the experimental results demonstrate that our algorithm achieves the highest performance for cloud detection.

REFERENCES

[1] J. Yang, W. T. Lv, Y. Ma, W. Yao, and Q. Y. Li, "An automatic ground-based cloud detection method based on adaptive threshold," *J. Appl. Meteorol. Sci.*, vol. 20, no. 6, pp. 713–721, 2009.
[2] J. E. Shield, R. W. Johnson, M. E. Karr, and J. L. Wertz, "Automated day/night whole sky imagers for field assessment of cloud cover distributions and radiance distributions," in *Proc. 10th Symp. Meteorol. Observ. Instrum. Amer. Meteorol. Soc.*, 1998, pp. 11–16.
[3] U. Feister and J. Shield, "Cloud and radiance measurements with the VIS/NIR Daylight Whole Sky Imager at Lindenberg (Germany)," *Meteorol. Zeitschrift*, vol. 14, no. 4, pp. 627–639, 2005.
[4] C. N. Long and J. J. DeLuisi, "Development of an automated hemispheric sky imager for cloud fraction retrievals," in *Proc. 10th Symp. Meteorol. Observ. Instrum.*, 1998, pp. 171–174.
[5] I. Genkova, C. N. Long, T. Besnard, and D. Gillotay, "Assessing cloud spatial and vertical distribution with infrared cloud analyzer," in *Proc. 14th ARM Sci. Team Meet.*, 2004, pp. 22–26.
[6] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagès, "Retrieving cloud characteristics from ground-based daytime color all-sky images," *J. Atmos. Ocean. Technol.*, vol. 23, no. 5, pp. 633–652, May 2006.
[7] A. Kreuter, M. Zangerl, M. Schwarzmann, and M. Blumthaler, "All-sky imaging: A simple, versatile system for atmospheric research," *Appl. Opt.*, vol. 48, no. 6, pp. 1091–1097, Feb. 2009.
[8] A. Heinle, A. Macke, and A. Srivastav, "Automatic cloud classification of whole sky images," *Atmos. Meas. Technol.*, vol. 3, no. 1, pp. 557–567, May 2010.
[9] 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, Mar. 2006.
[10] 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. 7, pp. 1504–1517, Sep. 2010.
[11] J. Yang, W. T. Lv, Y. Ma, W. Yao, and Q. Y. Li, "An automatic ground-based cloud detection method based on local threshold interpolation," *Acta Meteorol. Sinica*, vol. 68, no. 6, pp. 1007–1017, 2010.
[12] Q. Li, W. Lu, J. Yang, and J. Wang, "Thin cloud detection of all-sky images using Markov random fields," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 417–421, May 2012.
[13] X. F. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 10–17.
[14] G. Mori, "Guiding model search using segmentation," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1417–1423.
[15] B. Peng, L. Zhang, and D. Zhang, "Automatic image segmentation by dynamic region merging," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3592–3605, Dec. 2011.
[16] J. Kalisch and A. Macke, "Estimation of the total cloud cover with high temporal resolution and parametrization of short-term fluctuations of sea surface insolation," *Meteorol. Zeitschrift*, vol. 17, no. 5, pp. 603–611, Oct. 2008.
[17] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 7–23, Jun. 2001.
[18] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst., Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.