<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Cloud classification of ground-based images using texture–structure features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Zhuo, Wen; Cao, Zhiguo; Xiao, Yang</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2014</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/18802">http://hdl.handle.net/10220/18802</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2014 American Meteorological Society. This paper was published in Journal of Atmospheric and Oceanic Technology and is made available as an electronic reprint (preprint) with permission of American Meteorological Society. The paper can be found at the following official DOI: [<a href="http://dx.doi.org/10.1175/JTECH-D-13-00048.1">http://dx.doi.org/10.1175/JTECH-D-13-00048.1</a>]. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper is prohibited and is subject to penalties under law.</td>
</tr>
</tbody>
</table>
Cloud Classification of Ground-Based Images Using Texture–Structure Features

WEN ZHUO AND ZHIGUO CAO
National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan, China

YANG XIAO
Institute for Media Innovation, Nanyang Technological University, Singapore

(Manuscript received 8 February 2013, in final form 23 October 2013)

ABSTRACT

Cloud classification of ground-based images is a challenging task. Recent research has focused on extracting discriminative image features, which are mainly divided into two categories: 1) choosing appropriate texture features and 2) constructing structure features. However, simply using texture or structure features separately may not produce a high performance for cloud classification. In this paper, an algorithm is proposed that can capture both texture and structure information from a color sky image. The algorithm comprises three main stages. First, a preprocessing color census transform (CCT) is applied. The CCT contains two steps: converting red, green, and blue (RGB) values to opponent color space and applying census transform to each component. The CCT can capture texture and local structure information. Second, a novel automatic block assignment method is proposed that can capture global rough structure information. A histogram and image statistics are computed in every block and are concatenated to form a feature vector. Third, the feature vector is fed into a trained support vector machine (SVM) classifier to obtain the cloud type. The results show that this approach outperforms other existing cloud classification methods. In addition, several different color spaces were tested and the results show that the opponent color space is most suitable for cloud classification. Another comparison experiment on classifiers shows that the SVM classifier is more accurate than the $k$–nearest neighbor ($k$-NN) and neural networks classifiers.

1. Introduction

In recent years, ground-based imaging devices have been widely used for obtaining information on sky conditions. These devices, including the whole-sky imager (WSI) (Shields et al. 2003; Heinle et al. 2010), total-sky imager (TSI) (Pfister et al. 2003; Long et al. 2001) and all-sky imager (ASI) (Lu et al. 2004; Long et al. 2006; Cazorla et al. 2008), can provide continuous sky images from which one can infer cloud macroscopic properties—for example, cloud height, cloud cover (or cloud fraction), and cloud type. Most cloud-related studies require such information. According to the specifications of the China Meteorological Administration (CMA 2003) for ground-based meteorological observations, cloud height, cloud cover, and cloud fraction are three basic meteorological elements that weather stations must observe and record. These observations have been historically performed by human observers, which is expensive. Therefore, automatic observations are needed. Cloud height can be measured using paired ground imagery with photogrammetric methods (Allmen and Kegelmeyer 1996; Seiz et al. 2002; Kassianov et al. 2005), and cloud cover can be estimated using many algorithms (Pfister et al. 2003; Long et al. 2006; Kalisch and Macke 2008). The automatic cloud type classification of ground-based images is a challenging task and is still under development. Typically, an automatic cloud classification algorithm comprises three main stages: preprocessing, feature extraction, and a classifier.

The first stage is the preprocessing of images, which includes converting red, green, and blue (RGB) to grayscale images (Isosalo et al. 2007; Calbó and Sabburg 2008; Heinle et al. 2010), image segmentation methods (Singh and Glennen 2005; Calbó and Sabburg 2008; Heinle et al. 2010; Liu et al. 2011), image smoothing methods, and image enhancement methods (Liu et al. 2011). The main
The purpose of image preprocessing is to obtain certain image types for extracting features. Some preprocessing methods (e.g., image segmentation) can be considered part of a feature extraction due to their high dependency (e.g., the features computed directly from segmented images).

The second stage, feature extraction, plays an important role in achieving comparative classification results; numerous feature extraction methods have been studied in the literature. Buch et al. (1995) first applied Law’s texture measures (125 features), which was also used by Singh and Glennen (2005). The other four well-known texture feature extraction approaches that were adopted by Singh and Glennen (2005) are autocorrelation (99 features), co-occurrence matrices (14 features), edge frequency (50 features), and run length encoding (5 features). Isosalo et al. (2007) compared two texture features, local binary patterns (LBP) (256 features) and local edge patterns (LEP) (407 features), finding that LBP performed better than LEP. Calbó and Sabburg (2008) applied statistical texture features (12 features), pattern features based on a Fourier spectrum (4 features), and features based on the threshold image (6 features). Heinle et al. (2010) tested a large number features and selected 12 features with Fisher distances, a selection criterion. These 12 features included spectral features (7 features), co-occurrence matrix-based features (4 features), and a cloud cover feature (1 feature). Liu et al. (2011) proposed structure features of infrared images for cloud classification. These features included cloud fraction (2 features), edge sharpness (1 feature), and cloud maps and gaps (4 features).

It should be noted that the feature selection step is needed if a large number of features are extracted because pattern recognition algorithms are known to degrade in classification accuracy when faced with many features that are not necessary for predicting the desired output. The feature selection step consists of finding a subset of features to improve the classification accuracy and to reduce the computational burden. Heinle et al. (2010) used a feature selection method based on the Fisher distance. In our algorithms, we adopted principal component analysis (PCA), an alternate way to reduce the number of features (see section 2c).

In the third stage, a trained classifier receives an extracted feature vector and determines the cloud type. In previous studies, few classifiers have been explored because most researchers have focused on designing new sky image features and are less concerned about classifiers. The most frequently used classifier is the $k$–nearest neighbor ($k$-NN) (Singh and Glennen 2005; Isosalo et al. 2007; Heinle et al. 2010). Both Calbó and Sabburg (2008) and Liu et al. (2011) used a simple classifier that was developed based on the two-dimensional parallelepipeded technique. Other classifiers—for example, a binary decision tree (Buch et al. 1995) and linear classifier (Singh and Glennen 2005)—have also been applied.

To visually show the workflow of typical cloud classification algorithms, we take examples from previous studies (Calbó and Sabburg 2008; Liu et al. 2011) and present them in Figs. 1 and 2. Here, both methods have preprocessing, feature extraction, and a classifier. The detailed description of each feature calculation procedure can be found in the corresponding study (Calbó and Sabburg 2008; Liu et al. 2011).

In this paper, our goal is to propose an algorithm that can automatically distinguish different cloud types. This classification algorithm should also contain the three stages described above. Our primary concern is how to describe sky image features. In previous studies, there are generally two ways: choosing appropriate texture features (Singh and Glennen 2005; Calbó and Sabburg 2008; Heinle et al. 2010) and constructing structure features (Liu et al. 2011). In the first approach, clouds are treated as a texture type. However, despite using many texture features, the classification accuracy is not satisfactory, that is, below 75%. For example, Singh and Glennen (2005) achieved only a mean accuracy of approximately 70%. Moreover, Calbó and Sabburg (2008) achieved
approximately 62% accuracy and Heinle et al. (2010) achieved approximately 74% accuracy. A possible reason for these results is that these studies considered only texture information while ignoring the cloud structure information. Liu et al. (2011, p. 411) suggested that “manual cloud classification takes cloud shape as the basic factor, together with considering the cause of its development and the interior microstructure.” Therefore, they developed structure features, including edge sharpness, cloud maps, and gaps, and the total classification accuracy increased to 90.97% on their selected infrared image sets. However, the structure features that Liu et al. (2011) proposed may not be robust. For example, cloud boundaries and sky are usually blurred—that is, across several pixels—and it is hard to define a clear cloud edge. Thus, edge detection images may be very different when imaging conditions, such as illumination and visibility, change. For the same reason, segmentation images are not robust, which results in unreliable segmentation-image-based information—that is, cloud maps and gaps.

We believe that clouds that appear in ground-based images contain both texture and structure information. For example, cirrus is feather like, which can be interpreted as texture. Moreover, cumulus clouds resemble cotton balls. Similarly, we can easily determine that other cloud types—for example, cirrocumulus, cirrostratus, altocumulus, and stratocumulus—also have texture and structure information. More precisely, the structure should be rough because of the large intraclass variations of clouds. Thus, our main concern with describing features is to find an approach that can capture both texture and rough structure information. Fortunately, in the field of computer vision and image processing, census transform (Zabih and Woodfill 1994) and spatial pyramid strategy (Lazebnik et al. 2006) meet our requirements. Census transform can encode texture and local shape structure in a small image—that is, a $3 \times 3$ image patch—while spatial pyramid strategy can capture the global structure on larger scales. To achieve higher performance for the specific cloud classification task, these two techniques are extended and improved in this paper. Specifically, the standard census transform is extended to a color census transform, and a novel automatic block assignment method is developed for spatial representation. In our approach, we adopt the support vector machine (SVM) classifier, a powerful and popular classifier used in recent years.

In this paper, we make three main contributions: 1) we introduce census transform into the cloud classification task for ground-based images and extend it to a color census transform; 2) we propose an automatic block assignment method for spatial representation; and 3) we apply, for the first time to our knowledge, the SVM classifier in the field of ground-based imagery.

2. Methodology

In this section, we provide details on our cloud classification algorithm, including the three main stages: color census transform (CCT) for preprocessing, a spatial representation method to form features, and the SVM as a classifier. Figure 3 shows a flowchart of our proposed algorithm.

a. CCT for color sky images

Census transform (CT) is a nonparametric local transform (Zabih and Woodfill 1994) that is applied to grayscale images, that is, images whose pixels are described by a single value. Census transform compares a pixel value with its eight neighboring pixels. If the center pixel is larger than or equal to one of its neighbors, then a bit is set to 1 at the corresponding location [see Eq. (1) below]. Otherwise, a bit is set to 0. Then, the eight bits are put together from left to right and top to bottom. The ordered eight bits represent a base-2 number that is consequently converted into a base-10 number in $[0, 255]$ and is the CT value for this center pixel. Notice that the boundary pixels and corner pixels (i.e., those that do not have eight surrounding pixels) are simply ignored because they are relatively few in number. An example of our algorithm is given below:
As illustrated in Eq. (1), the census transform operation maps a $3 \times 3$ image patch to one of 256 cases, each corresponding to a special local structure type. Such a local transform relies only on pixel value comparison. Therefore, it is robust to, for example, illumination changes and gamma variations (Poynton 2003, 260–265). Note that the census transform has the ability to capture texture information because it is equivalent to the local binary pattern code LBP$_{8,1}$ (Ojala et al. 2002), which has been widely used for texture image analysis and adopted by Isosalo et al. (2007) for cloud texture measures. Another important property is that the CT values of the neighboring pixels are highly correlated; this constraint makes the CT values capture not only local structure information but also implicitly contain information describing global structures (Wu and Rehg 2011). Because the sky images in our research are captured in RGB color space—that is, pixels are described by three values (red, blue, and green)—we require a method to convert them into one or a group of grayscale images. In previous research, Calbó and Sabburg (2008) converted an RGB image into a single grayscale image in two ways: computing a red-to-blue components ratio (R/B) and intensity values. However, color information is lost using these two conversion methods. Heinle et al. (2010) partitioned an RGB image into three grayscale images, each of which contains only one component (R, G, or B). However, the three components R, G, and B are highly dependent. Thus, the RGB space is not an optimal choice. As a result, we need a color transformation that can retain the full color information. Moreover, the transformed components should be as independent as possible. We investigated several color transformations and their corresponding color spaces and found that the opponent color space (van de Weijer et al. 2005; van de Sande et al. 2010) performed best for cloud classification (see experiments in section 3b). The opponent color space is an orthonormal transformation given by (excluding scaling factors) the following:

$$\left(\begin{array}{c}
O_1 \\
O_2 \\
O_3
\end{array}\right) = \left(\begin{array}{c}
\frac{R - G}{\sqrt{2}} \\
\frac{R + G - 2B}{\sqrt{6}} \\
\frac{R + G + B}{\sqrt{3}}
\end{array}\right). \quad (2)$$

In opponent color space, the intensity information is represented by component $O_3$ and the color information by $O_1$ and $O_2$. We applied the census transform to each component (i.e., $O_1$, $O_2$, and $O_3$) and obtained three corresponding census-transformed images, respectively, by replacing a pixel with its CT value. The example in Fig. 4 shows that the difference between components becomes larger when converting RGB color space to opponent color space. The census transform for each component retains global structures of the picture (especially discontinuities) besides capturing the local structures and textures. Although the CCT is adopted for the preprocessing stage, it can also be seen as part of feature extraction. The following section describes how to encode the global structures and form a feature vector.

b. Spatial representation

The goal of spatial representation is to encode rough structure information from a sky image on a larger scale.
The structure information is related to physical size. In other words, we should determine the spatial size of image regions in which the structure information is represented. Here, we do not attempt to use a specific region size and instead use a spatial pyramid strategy that considers structure information from different region sizes.

A traditional spatial pyramid (Lazabnik et al. 2006; Wu and Rehg 2011) divides an image into subregions of different sizes and provides integrated correspondence results in these regions. As shown in Fig. 5a, there are four level splits in the spatial pyramid. Level \(l (l = 1, 2, 3, 4)\) divides the image into \(l \times l\) blocks, for example, level 4 has 16 blocks; a total of 30 blocks are created. However, the block shapes are rectangular with boundaries that are artificially specified. Therefore, errors may occur when a homogeneous region is divided into two adjacent blocks.

To overcome this limitation, we propose an automatic block assignment method in which the blocks are not rectangular regions and instead automatically computed in accordance with the image content. Specifically, for an image in opponent color space, \(O = [O_1, O_2, O_3]\) at level \(l\), we first set center points for each \(l \times l\) block as \(p(c_i, c_j), i, j = 1, 2, \ldots, l\) where \(c_i = [(i - 0.5)N/l]\), \(c_j = [(j - 0.5)M/l]\), and \(M\) and \(N\) denote the image size. Then, every \(M \times N\) pixels are assigned to a block with the closest distance as shown:

\[
\arg \min_{i,j} \alpha \|O(x, y) - p(c_i, c_j)\|^2 + (1 - \alpha) \frac{l^2}{M \times N} \|(x, y) - (c_i, c_j)\|^2,
\]

where \(\alpha\) is a parameter, which allows us to weigh the relative importance between color similarity and spatial proximity. When \(\alpha\) is small, the spatial proximity is more important and the resulting blocks are more compact. In particular, when \(\alpha = 0\), the result of (3) is equivalent to the traditional spatial pyramid with rectangle blocks. When \(\alpha\) is large, the resulting blocks adhere more tightly to cloud boundaries and have a less regular size and shape. In our sky image set, \(\alpha\) can be in the range [0.02, 0.5].

Figures 5b and 5c show an example of two partitions for the sky image shown in Fig. 4. Unlike the rectangular block approach (Fig. 5b), the boundary computed using the automatic block assignment method (Fig. 5c) is deformable, which can ensure that a relatively small

---

**Fig. 4.** Example opponent color space and corresponding census-transformed image. Values of opponent color space are adjust to an integer range [0,255] to produce the 8-bit grayscale image.
homogeneous region will not be divided into different blocks. It should be noted that for some cloud types—for example, stratus, altostratus, and clear sky—there are no strong edges. Thus, the automatic block assignment method will assign blocks similar to rectangles.

c. Feature vectors and SVM classifiers

A histogram of CT values will be counted for blocks of a census-transformed image. Because the CT values have an integer value in the range $[0, 255]$, the dimension of a counting histogram is 256. However, the histogram bins are strongly correlated. As described in section 2a, the census transform compares neighboring pixel values to generate the CT values. Therefore, adjacent CT values are dependent. In the classification task, it is important to use independent features. Therefore, we utilize PCA to reduce the dimensions to 40.

Because the CT values are based solely on pixel intensity comparisons, it might be helpful to include a few image statistics. Inspired by the previous studies of Calbó and Sabburg (2008) and Heinle et al. (2010), we use four statistical features of pixels, including the average value, standard deviation, skewness, and entropy, in a block of opponent color images $O_k$. Then, we concatenate the histograms of dimensionality reduction and four statistics features in all blocks to form the final feature vector.

In our proposed approach, we use a level 4 pyramid and a total of 30 blocks ($16 + 9 + 4 + 1 = 30$) (see Fig. 5a) for each opponent color channel. For each block, a histogram of 40 dimensions and four statistics are computed. Thus, in one opponent color channel, we have a feature of $(40 + 4) \times 30 = 1320$. Because the opponent color space has three channels, the final feature vector has $1320 \times 3 = 3960$ dimensions for one sky image.

The classifier we used in our algorithms is the SVM (Boser et al. 1992; Cortes and Vapnik 1995), which is a powerful classifier that has achieved great success in the field of pattern classification. The basic SVM is a two-class classifier. For example, suppose a training set has $n$ dimensional feature vectors $x_i \in \mathbb{R}^n$, and the corresponding label $y_i \in \{-1, 1\}$. Then, the SVM solves the following optimization problem:

$$
\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i
$$
subject to $y_i [w^T \phi(x_i) + b] \geq 1 - \xi_i$, $\xi_i \geq 0$.

(4)

where $\phi(x_i)$ maps $x_i$ into a higher dimensional space to obtain the nonlinear classification and $C > 0$ is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is defined as the kernel function. In our method, we use the radial bases function (RBF) kernel, that is, $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$. $\gamma > 0$. The kernel parameter and the penalty parameter are chosen by cross validation on the training set.

Notice that the dimension of a feature vector is 3960, and the kernel function maps the input feature vectors into a higher dimensional feature space. Such a high dimensionality does not increase the difficulty for training an effective SVM classifier with a relatively small training set because special properties of the decision surface ensure the high generalization of the SVM (Cortes and Vapnik 1995).

For multiclass classification, the one-against-one strategy is used (Knerr et al. 1990). If $K$ is the number of classes, then $K(K - 1)/2$ two-class classifiers are constructed, one for every possible pair of classes. The final classification is made using voting strategy. In the following section, the LibSVM software package (Chang and Lin 2011) was used throughout the experiments.

![Fig. 5. Illustrations of spatial representation: (a) spatial pyramid splitting an image into four levels, (b) splitting the sky image with rectangular blocks at level 4, and (c) splitting result of the automatic block assignment method at level 4.](image-url)
3. Experiments

In this section, we report experimental results based on a number of sky images taken by a sky camera in Beijing, China. The sky camera consists of an ordinary charge-coupled device (CCD) and an auto iris lens; its viewing angle is approximately 60°. Images acquired by the sky camera are stored in 8-bit color JPEG format with a resolution of 352 × 288 pixels. To cover the entire sky, the camera is fixed on a pan-till platform and scans the entire sky in horizontal intervals of 30° and vertical intervals of 40°. Controlled by a servo motor, the sky camera can move fast and the time of capturing two adjacent images is approximately 2 s. Therefore, information on the entire sky is contained in an image sequence of 28 images (see Fig. 6) with a time less than 1 min. This device is advantageous because unlike the WSI, TSI, or ASI, which use a fish-eye lens, it can capture images similar to a human eye—that is, without the so-called fish-eye distortion. Another advantage is that the sun disk (if it exists) only appears in a few images in a sequence; for example, as shown in Fig. 6, only six images contain the sun disk.

The sky images used in the experiment were acquired from August 2010 to May 2011. A subset of these images was selected with the help of Xiangang Wen, a meteorological observer from the China Meteorological Administration who has more than 10 years of sky observation experience. The selected images should be of good quality and can be well recognized by visual screening. If the image contains other objects or the sun disk, then an image mask that sets all corresponding pixel values to zero is used. After selecting a subset of sky images, we need to label them to evaluate our proposed approach. However, how to classify cloud types remains an unresolved problem. Liu et al. (2011) argued that the criteria developed by the WMO (1975) for classifying different cloud types is unsuitable for automatic cloud classification. Moreover, there is still no consensus on defining different cloud types or sky conditions in recent studies (Isosalo et al. 2007; Calbó and Sabburg 2008; Heinle et al. 2010; Liu et al. 2011). In this study, similar to Heinle et al. (2010), we classified sky images into six categories based on visual similarity: 1) cumulus (172 images); 2) cirrus and cirrostratus (241 images); 3) cirrocumulus and altocumulus (181 images); 4) clear sky (195 images); 5) stratocumulus (262 images); and 6) stratus and altostratus (180 images). Here, each selected image (or sample) is a color image of 352 × 288 pixels, similar to the RGB image in Fig. 4. In other words, we selected only one or more representative images (i.e., not all 28 images) of a whole-sky area. The minimum mean intensity (darkness) of all selected images was 80. Notice that the cumulonimbus, nimbostratus, and towering cumulus classes were not included here due to a lack of available data. Details on the characterization of each cloud type can be found in Heinle et al. (2010).

A cloud classification experiment should be repeated over 10 runs to obtain an average performance. For each run, we chose a fixed number of training images at random from each category, placing the remaining images in the testing set. For example, if the number of training images is 5, then the number of testing images is, for example, 167 for cumulus and 236 for cirrus and cirrostratus. As mentioned before, the whole-sky area contains 28 images. Therefore, images selected from the same whole-sky area may be placed in both the training and testing sets. However, adjacent images from the same whole-sky area overlap in only a few parts (see Fig. 4). Therefore, each image can be considered as a different
sample. Additionally, multiple rounds of experiments using randomly selected training samples can reduce the impact of the specific samples when evaluating the performance of the classification algorithms.

a. **Comparison results of different methods**

In this experiment, we performed a quantitative comparison of our proposed approach and five alternate methods, including Isosalo et al. (2007), Calbó and Sabburg (2008), Heinle et al. (2010), Liu et al. (2011), and the CT value-based method with rectangle blocks (CTR). Because these previous approaches used different classifiers that lead to incomparable classification results, we only implemented the feature stage described in their methods and adopt the SVM classifier with the same training images in each run. Note that the Liu et al. (2011) method was designed for infrared images (only one channel). Therefore, we first converted our RGB images to grayscale images (more precisely, intensity images), and then evaluated their method.

The classification results are shown in Fig. 7 and Table 1 (rows 1–6). As the number of training images increased, the classification accuracy of all methods increased. However, our approach always outperformed the other methods. After improving the rectangle block (in CTR) with the adaptive boundaries (in ours), the mean classification accuracy increased, especially when the training images were large (e.g., 40 or 80 training images). The accuracy of the Liu et al. (2011) method was the lowest, which indicates that the extracted structure features may not work for visible images. The texture features of the Isosalo et al. (2007) method, using only LBP (or census transform), outperformed traditional texture features of the Calbó and Sabburg (2008) and Heinle et al. (2010) methods. However, these methods were still less accurate than our approach, and CTR, which improves the structure information on a larger scale, was helpful for cloud classification.

The confusion matrix for our approach from one run on 80 training images is shown in Fig. 8, where row and column names are true and predicted labels, respectively. The biggest confusion occurred between the cirrus and cirrostratus class and the cumulus class. Moreover, the cirrus and cirrostratus class had the lowest classification accuracy. This is because cirrus and cirrostratus are thin clouds with some of the internal fine structure that our approach occasionally failed to extract discriminative features in this situation. The cirrus and cirrostratus class is not misclassified as clear sky more frequently compared to misclassification as cumulus because the clear sky contains almost no texture or structure patterns. This indicates that our algorithms have an ability to capture information from thin cloud images. The next largest error occurred when classifying stratocumulus. This is
because stratocumulus clouds are very complex, which vary in color from dark gray to light gray and may appear as, for example, rounded masses or rolls. These results suggest that we should increase the training data to cover enough variations. Moreover, future studies with more robust features are necessary.

To further analyze the classification results quantitatively, we utilized paired $t$ tests. Here, we let $X$ and $Y$ denote the classification accuracy of our approach and one of the other methods, respectively. Then, we assumed that $X$ and $Y$ have difference variances. Thus, the $t$ statistic to test whether the means $X$ and $Y$ are different can be calculated as follows:

$$ t = \frac{\overline{X} - \overline{Y}}{\sqrt{s_X^2 + s_Y^2} \sqrt{n}} $$

where $s_X$ and $s_Y$ are the standard divisions of $X$ and $Y$, respectively; and $n$ is the number of runs. For use in significance testing, the distribution of the test statistic was approximated as being an ordinary Student’s distribution with the degrees of freedom (df) calculated using

$$ df = (n - 1) \frac{(s_X^2 + s_Y^2)^2}{s_X^2 + s_Y^2} $$

Once a $t$ statistic and the corresponding degrees of freedom were determined, a $p$ value was found using a table of values corresponding to the Student’s distribution. The right-tailed test $p$ values are shown in Table 2. In most cases, except for CTR with 5 and 10 training images, the $p$ values are far below 0.01, which means that the mean of our classification accuracy was higher than the other methods at the 99% confidence level. However, when 5 and 10 training images were used, the $p$ values are 0.131 and 0.057, respectively. Because these values are larger than 0.05, our approach does not significantly improve
the CTR method. The primary reason is that when few training images were used (e.g., 5 or 10), many different in-class patterns were not included. Therefore, the trained classifier was not reliable.

To evaluate our method for different cloud type selections, we relabeled the sky images from six to nine categories and compared the experiments. Specifically, all of the categories that contain two cloud forms were divided into two classes. For example, the cirrus and cirrostratus class was relabeled as a cirrus class and a cirrostratus class separately. We used 40 training images per category. The comparison results are shown in Fig. 9. Although the average classification accuracy decreased to 64.1%, our method still outperformed the others in each category. The remarkable difference in classification accuracies comparing six categories (see Fig. 8) and nine categories (see Fig. 9) is the large decrease in accuracy of clear sky (from 95% to 72.8%). The main reason is the difference of training images per category (80 vs 40). We carefully observed the misclassified samples of clear sky and found that the cirrus and the cirrostratus classes were the two classes into which samples were most frequently misclassified. For example, a total of 41 clear-sky samples were misclassified from one run, in which 24 samples were misclassified into cirrus and 10 samples were misclassified into cirrostratus. This happened because sometimes unexpected objects occurred in the image caused by the camera, for example, lens flares (see Fig. 10, left) or polluted lens (see Fig. 10, right). Since all compound cloud classes in six categories were separated, some cloud classes contained a relatively small number of images, for example, stratus and cirrocumulus. Thus, these two cloud classes achieved the highest classification accuracies shown in Fig. 9.

Table 2. The \( p \) values of paired \( t \) tests (right tailed), which were calculated between our approach and other existing methods.

<table>
<thead>
<tr>
<th>Number of training images per category</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>40</th>
<th>80</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTR</td>
<td>0.13</td>
<td>0.06</td>
<td>( 4.2 \times 10^{-3} )</td>
<td>( 7.9 \times 10^{-6} )</td>
<td>( 2.9 \times 10^{-5} )</td>
</tr>
<tr>
<td>Isosalo</td>
<td>( 1.0 \times 10^{-3} )</td>
<td>( 2.6 \times 10^{-3} )</td>
<td>( 5.6 \times 10^{-8} )</td>
<td>( 4.4 \times 10^{-12} )</td>
<td>( 1.2 \times 10^{-11} )</td>
</tr>
<tr>
<td>Calbo</td>
<td>( 1.9 \times 10^{-5} )</td>
<td>( 1.6 \times 10^{-6} )</td>
<td>( 1.3 \times 10^{-12} )</td>
<td>( 1.1 \times 10^{-9} )</td>
<td>( 5.7 \times 10^{-17} )</td>
</tr>
<tr>
<td>Heinle</td>
<td>( 3.6 \times 10^{-7} )</td>
<td>( 3.1 \times 10^{-10} )</td>
<td>( 2.5 \times 10^{-15} )</td>
<td>( 1.5 \times 10^{-17} )</td>
<td>( 9.3 \times 10^{-18} )</td>
</tr>
<tr>
<td>Liu</td>
<td>( 1.5 \times 10^{-9} )</td>
<td>( 2.1 \times 10^{-13} )</td>
<td>( 5.6 \times 10^{-18} )</td>
<td>( 3.8 \times 10^{-24} )</td>
<td>( 2.0 \times 10^{-19} )</td>
</tr>
</tbody>
</table>

Fig. 9. Comparison of different methods on classifying nine cloud classes on 40 training images per category.
Notice that several observers would label the samples differently when the number of cloud classes is large (e.g., nine categories). Garand (1988) showed that the agreement of experts on a 20-class scheme is of the order of 75%. In that case, the experts only need to make a binary decision (i.e., agree or disagree on the machine labeling). Therefore, in the future studies, we should explore how to combine visual features and physical characteristics (e.g., height, fraction, albedo, and directionality).

This would give rise to more objective criteria for labeling and classifying cloud samples.

b. Comparison results of different color space

We compared the performance of several color spaces, including the opponent color space, RGB; normalized RGB (rgb); and hue, saturation, and value (HSV). The components $r$, $g$, and $b$ of the normalized RGB color space are defined as

$$
\begin{pmatrix}
  r \\
  g \\
  b
\end{pmatrix} =
\begin{pmatrix}
  R \\
  G \\
  B
\end{pmatrix}
\left(\begin{array}{c}
  1 \\
  R + G + B \\
  R + G + B
\end{array}\right) .
$$

Moreover, the components $h$, $s$, and $v$ of the HSV color space can be computed as

\[ h = \frac{1}{6} \times \begin{cases} 
0, & \text{if } \max = \min \\
(G - B)/(\max - \min), & \text{if } \max = R \text{ and } G \geq B \\
(G - B)/(\max - \min) + 6, & \text{if } \max = R \text{ and } G < B \\
(B - R)/(\max - \min) + 6, & \text{if } \max = G \\
(R - G)/(\max - \min) + 4, & \text{if } \max = B 
\end{cases} \]

\[ s = \begin{cases} 
0, & \text{if } \max = 0 \\
1 - \min/\max, & \text{otherwise} 
\end{cases} \]

\[ v = \frac{\max}{255}, \]

FIG. 11. Performance of different color space. Each point in the figure shows the mean and standard deviation of the cloud classification accuracy (%).
where \( \text{max} = \text{Max}(R,G,B) \) and \( \text{min} = \text{Min}(R,G,B) \).

In addition, we tested the grayscale space converted in two ways: intensity \((R + G + B)/3\) and \(R/B\). For simplicity, the block regions used in the spatial representation step were the same for all methods, which were calculated with the automatic block assignment method in the opponent color space. The classification accuracies are shown in Fig. 11 and Table 1 (rows 1 and 7–11). The opponent color space outperformed other color spaces and the grayscale space. When the number of training images was small (e.g., 5 or 10), the classification accuracies for different color space were close and greatly varied. With more training images, the performances of the RGB and HSV color spaces remained close. However, both color spaces were better than the other color spaces. It is surprising that the normalized RGB (rgb) color space performed even worse than the intensity space, which indicates that it is crucial to choose an appropriate color space for the cloud classification task. The red-to-blue ratio (R/B) space performed the worst, and thus this space may not be a good candidate for cloud classification.

c. Comparison results of different classifiers

Because the \(k\)-NN classifier has been widely used for cloud classification (Peura et al. 1996; Singh and Glennen 2005; Isosalo et al. 2007; Heinle et al. 2010), we first compared it with the SVM classifier. As suggested by Heinle et al. (2010), the distance measure in the \(k\)-NN classifier was the Manhattan distance, and the number of considered neighbors—that is, the parameter \(k\)—was set to 1, 3, and 5. The comparison of classification results is shown in Fig. 12 and Table 1 (rows 1 and 12–14). For the \(k\)-NN classifier, its performance decreased as \(k\) increased in this case. However, even for \(k = 1\), the classification accuracy of the \(k\)-NN approach remained below the SVM classifier. The reason for the decreasing performance as \(k\) increased is related to the complexity of clouds. In the feature space, cloud features that belong to the same class cover a very complex and irregular area. Therefore, neighbors of one sample that belong to different classes are possible, which leads 1-NN performing the best.

We investigated another popular classifier, neural networks (Singh and Glennen 2005), for comparison. We used a three-layer feed-forward network with sigmoid output neurons. The number of nodes in the input layer was 3960 (the same as the feature dimension), 6 in the output layer (the same as the number of cloud types), and 120 in the hidden layer. The classification results are shown in Fig. 12 and Table 1 (last row). The accuracy of the neural network classifier was significantly lower than that of the SVM and \(k\)-NN, and achieved only 60% accuracy for 80 training images per category. A possible reason is that neural networks are susceptible to getting stuck in local minima for the high dimension of features.
4. Conclusions

In this paper, we proposed a cloud classification approach for ground-based sky images. On the basis of previous studies, we believe that clouds contain both texture and rough structure information. To capture texture and local structure information, we proposed a color census transform (CCT) operation that first converts RGB color space into opponent color space and applies census transform to each color component. To capture global rough structure information more precisely, we proposed an automatic block assignment method. The experiment results showed that our approach outperforms existing cloud classification methods.

In addition, we tested different color spaces and determined that the opponent color space was the optimal choice for cloud classification of color sky images. We also compared different classifiers and the experimental results showed that our SVM classifier outperformed the k-NN and neural networks.

Our approach occasionally confused cirrus and cirrostratus with cumulus. Hierarchical classification processes or more robust features could be adopted in future work to better represent these cloud types. Moreover, the misclassification of stratocumulus could be reduced by increasing the number of training samples to cover more patterns. In the future, we will explore some modern pattern classification techniques—for example, a bag of words model (Wu et al. 2009) and random forest (Breiman 2001)—to classify the cloud types more effectively.

The experimental results showed that our approach can distinguish different cloud types from ground-based images, which are captured during daytime and have good quality to be easily recognized by human observers. At night and in low-visibility situations (e.g., rain, snow, fog, and haze), the sky images captured from ground-based devices will have poor quality (i.e., blurred, low contrast, and noisy) so that even an experienced observer cannot recognize cloud information from it. Therefore, in future work, some specific image enhancement algorithms—for example, the preprocessing stage for the classification method—must be studied.

Acknowledgments. We gratefully acknowledge the assistance of the China Meteorological Administration in providing the sky images. We also thank Xiangang Wen for labeling cloud types.

REFERENCES


