Cloud Classification Using Whole-Sky Imager Data

K. A. Buch, Jr., C.-H. Sun, and L. R. Thorne
Sandia National Laboratories
Livermore, California

Introduction

Clouds are one of the most important moderators of the earth radiation budget and one of the least understood. The effect that clouds have on the reflection and absorption of solar and terrestrial radiation is strongly influenced by their shape, size, and composition. Physically accurate parameterization of clouds is necessary for any general circulation model (GCM) to yield meaningful results. The work presented here is part of a larger project that is aimed at producing realistic three-dimensional (3D) volume renderings of cloud scenes based on measured data from real cloud scenes. These renderings will provide the important shape information for parameterizing GCMs. The specific goal of the current study is to develop an algorithm that automatically classifies (by cloud type) the clouds observed in the scene. This information will assist the volume rendering program in determining the shape of the cloud.

Much work has been done on cloud classification using multispectral satellite images (e.g., Ebert 1992; Garand 1986; Inoue 1987; Lamei et al. 1993; Lee et al. 1990; Smotroff 1993; Yool and Brandley 1992). Most of these references use some kind of texture measure to distinguish the different cloud types and some also use topological features (such as cloud/sky connectivity or total number of clouds). A wide variety of classification methods has been used, including neural networks, various types of clustering, and thresholding. The work presented here uses binary decision trees to distinguish the different cloud types based on cloud feature vectors.

The images used in this study were obtained from two whole-sky imagers (WSIs) located at White Sands, New Mexico, during the months of July, September, and October 1989 and May 1992. These ground-based imagers are spaced approximately 5 km apart and, together, provide a stereo view of the sky. The data acquired from these imaging systems can be used in a wide variety of applications including cloud base height determination and cloud-type occurrence statistics. These applications can be augmented by automatic determination of the type of cloud present in a cloud scene.

WSI data present some unique challenges to cloud classification for several reasons. First, the image is not a flat representation of the cloud scene. Instead, equal distances in the image are equal angles in the scene. Therefore, while the angular resolution of the data ($\approx 1/3^\circ$) is constant, the length resolution decreases from the center to the edge of the image. Also, due to the perspective of ground observations, the center of the image views the bottom of the clouds but the image edge views either the bottom (for clouds of large spatial extent) or the side (for modular or broken clouds). Another challenging aspect of the WSI data is the relatively limited field of view. Compared with satellite data, the WSI cloud scene is a very local one. Many of the clouds extend beyond the edge of the image and, therefore, very few clouds are seen in their entirety. This precludes such features as perimeter measures, shape measures, etc., from being used in the classification.

Classification Method

The objective of this work is to automatically classify the different types of clouds observed in WSI data. To do this we use binary decision trees (Breiman et al. 1984) to classify each pixel in the cloud scene as either clear sky or one of four cloud types: altocumulus, cirrus, cumulus, or stratus. Binary decision tree algorithms use a set of training data, for which the cloud class is known, and generate a hierarchy of thresholds based on a series of measured features. For each point in the training sample, *n* different features are measured. These measurements are placed in an *n*-dimensional space, and successive splits (which are equivalent to thresholds on individual feature values) are used to separate the points into different regions or nodes. At every level, each region is individually analyzed to find the optimal split (i.e., the split which most reduces some measure of the node impurity). In the resulting tree structure, each node has an associated misclassification rate estimate which can be used to evaluate the relative performance of the tree and also the confidence of the final classification.

The features described here are all measured for each pixel in the WSI data. Most of the features are computed as an average over a neighborhood around each pixel. The resulting binary decision tree, then, actually classifies individual pixels as belonging to one of four cloud classes or the clear sky class. This result can be generalized to give the most likely class to which each cloud or cloud image belongs.

The features used in this study can be grouped into three categories: texture measures, position information, and pixel brightness. One of the texture measures is the standard deviation of the image brightness computed over a neighborhood around each pixel. The remaining texture measures are computed using the LAWS kernels (Laws 1980). These 5 x 5 matrix kernels are used to quantify the amount of 25 different types and orientations of texture in the cloud scenes and are derived by taking the outer product of combinations of two of the following five vectors: L5 = [1 4 6 4 1], E5 = [-1 -2 0 2 1], S5 = [-1 0 2 0 -1], W5 = [-1 2 0 -2 1], and R5 = [1 -4 6 -4 1]. The letter names for the vectors refer to the type of texture that each vector is sensitive to, namely Level, Edge, Spot, Wave, and Ripple, respectively. For example, the kernel L5E5 is the outer product of the vectors L5 and E5, and E5L5 is its transpose, the outer product of the vectors E5 and L5.

The method used to obtain a measure of the texture energy in the neighborhood of each pixel closely follows that found in Laws (1980). First, each cloud scene is convolved with one of the 5 x 5 texture kernels. Next, the absolute value of the result is averaged over a neighborhood around each pixel to give a measure of the texture energy at each point. The nature of the WSI camera systems gives rise to two important distance measures in the acquired images. First is the distance from the center of the image, which is equivalent to the zenith angle. Since the linear (not angular) resolution of the image is degraded as the zenith angle increases, the texture of the clouds will also change with increasing zenith angle. Second is the difference between the sun angle and viewing angle, which we approximate as the distance from the center of the occultor. These two distance measures are also included as features used in the classification. Finally, the pixel brightness of the acquired image is also included as a feature.

The texture measures, as outlined above, only use information from a small neighborhood relative to the sizes of the clouds. In order to include the available large-scale information in the classification, the resolution and sample density of the data are recursively reduced to form a pyramid-type structure following Burt (1984). In this pyramid structure, each successive level represents low-pass filtered copies of the original image with an additional decrease in sampling density and a corresponding decrease in physical size (each level being one-half as big as the preceding level). The same sized kernels and neighborhoods (in pixels) are used to compute the texture energy at each level with the result that each level contains information collected over a larger extent than the previous level. After the texture measures are computed on the reduced images, they are expanded by interpolation (Burt 1984) to the original size so that information from different spatial resolution levels can be used simultaneously at each point. Four levels were used for this work with the original image being level 0, with a diameter of about 450 pixels, and level 3 being eight times smaller in size than level 0, with a diameter of about 56 pixels.

Of the features outlined above, there is no way to determine *a priori* which ones are more important to the final classification and which ones can be left out without introducing errors. Furthermore, it would be very computationally expensive to grow the decision tree using all the features and let the algorithm choose the best ones. Therefore, an iterative process was used to estimate which of the above features allowed the optimal tree to be grown. First, the training data were classified using each feature independently and the resulting resubstitution error rates compared (the resubstitution error rate is the error incurred when the data that were used to grow the decision tree are classified by the decision tree). This error rate can be non-zero because in some instances the training data do not warrant the amount of splitting required to produce a zero error rate. The five features that gave the lowest error rates were then passed on to the next step. The training data were classified using one of the best five features from the previous step and all the other features, taking one at a time so that each tree was grown using two features. The five trees with the lowest error rates were then passed on to step three. To each pair of features from step two, all other features were added one at a time and a tree with three features grown. This procedure was repeated until the nine most important features were discovered. This is actually an estimate of the most important features. To determine the absolute best features, an exhaustive search would have to be made where every combination of nine features would be used to classify the training data and the resubstitution error rates compared for all trees. This is an enormous task for even a small number of features.

Table 1. The composition and size of the training data and test data sets.

Cloud Class	Number of Training Images	Number of Training Points	Number of Test Images	Number of Test Points
Altocumulus	3	80,197	1	32,264
Cirrus	5	153,760	1	52,347
Cumulus	5	114,955	2	75,235
Stratus	5	187,394	2	152,595
Clear sky	16	469,152	4	125,606

Results

Table 1 lists the number of images and pixels for each cloud type used in the training data set. The method outlined above was used to determine the nine most important features for classifying the training data. The resubstitution error rate of the resulting decision tree was 6.4%. The nine features used for the classification, in order from most to least important, are

1. the LAWS feature L5L5 for level 2
2. the distance from the center of the occultor
3. the distance from the center of the image
4. the local standard deviation at level 0
5. the local standard deviation at level 1
6. the pixel brightness
7. the LAWS feature L5E5 for level 3
8. the LAWS feature L5R5 for level 3
9. the LAWS feature E5L5 for level 3

Several things should be noted from this list. First, the four LAWS texture measures included are for levels 2 and 3. This implies that these important textures are more large-scale textures, probably inter-cloud rather than intra-cloud. Also note that the transpose pair L5E5 and E5L5 are included, indicating that this particular texture occurs with more than one orientation. Second, the local standard deviation features are for levels 0 and 1, indicating that intra-cloud texture is also important for the classification. Finally, both distance measures are very important features, indicating the strong dependence of cloud appearance on position within the image.

The resubstitution error rate is a measure of how well the decision tree classifies the training data. But a more strict performance measure of the decision tree is the error rate resulting from classifying data not included in the training set. (Table 1 lists the number of images and pixels for each cloud type used in this test data set.) This error rate is the misclassification error rate, which is defined as the percentage of class I points that are misclassified as class j, with $j \neq I$. The misclassification error rate would be zero for a perfect classification. It is also instructive to generate a confusion matrix C_{ij} which gives the fraction of class I points that are classified as class j. This matrix would be diagonal for a perfect classification and shows which classes are confused most often.

Since the classification is performed on the individual pixels in each test image, the output from the decision tree algorithm is a very noisy image. Several methods can be used to clean up the raw output and produce a more usable and informative result. For the present, a k-nearest neighbor filter is used to remove the noise in the raw classified images. The class of each point is changed to the most probable class in a 31 x 31 neighborhood around that point. This has the effect of absorbing small islands of one class into the larger regions of the surrounding class. Visual inspection of both the raw classified and the filtered images shows that the basic trends are maintained but the noise is greatly reduced. Figure 1 shows one example of the classification result with Part a showing the cloud scene, Part b showing the classification result before filtering, and Part c showing the result of filtering with the

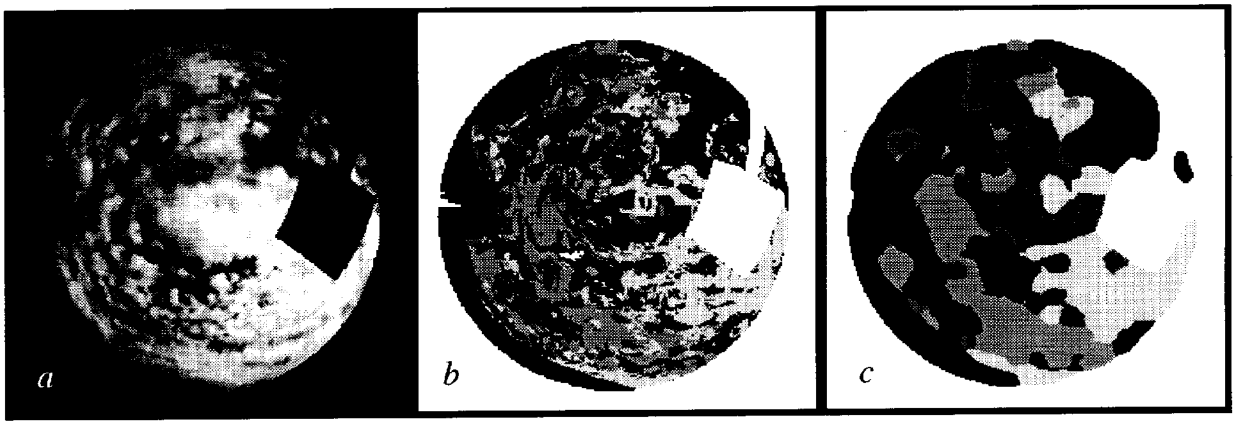

Figure 1. An example of the classification results for one of the test data images. Part a shows the WSI cloud scene. Part b shows the classification result before filtering. Part c shows the classification result after processing with the k-nearest neighbor filter. In Parts b and c, the five gray levels, starting with black and increasing to light gray, correspond to pixels classified as clear sky, cirrus cloud, stratus cloud, altocumulus cloud, and cumulus cloud, respectively.

k-nearest neighbor filter. In Parts b and c, black denotes clear sky pixels, dark gray denotes cirrus cloud pixels, medium dark gray denotes stratus cloud pixels, medium light gray denotes altocumulus cloud pixels, and light gray denotes cumulus cloud pixels. Table 2 gives the confusion matrix for the filtered results which has an overall misclassification rate of 39%.

Conclusions/Discussion

The decision tree grown using the training data has a very low resubstitution error rate (6.4%), indicating that the features used to perform the classification represent the training data well. The misclassification error rate is higher (39%) for several reasons. First, there is the possibility that the training data are not representative of the test data. This difficulty can be overcome by including more samples in the training data, thus supplying the tree growing algorithm with information about a larger range of clouds in each cloud class. With the immense variability that exists in clouds, this is almost certainly a contributing factor in this work and also one of the hardest to overcome. Second, when the test data classification results are analyzed, several common errors that contribute to misclassified pixels are easily recognized. The two

Table 2. Confusion matrix for the test data after filtering with the k-nearest neighbor filter.

True Class	**(%) altocumulus**	**(%) cirrus**	**(%) cumulus**	**(%) stratus**	**(%) clear sky**
Altocumulus	43	19	12	10	17
Cirrus	7	35	21	24	12
Cumulus	8	14	54	14	11
Stratus	4	12	12	46	26
Clear sky	1	0	1	0	98
Overall misclassification rate = 39%					

most obvious are problems in correctly determining the class of pixels near the edge of the image and near the occultor. This is not an unexpected result since the resolution is the poorest near the edge of the image, which greatly reduces the sensitivity of the texture measures, and the brightness is generally the highest near the occultor, which can tend to make different cloud types appear similar. Another factor contributing to the high misclassification error rate is related to the use of the k-nearest neighbor filter. Passing this filter over the classified images does decrease the misclassification error rate (for the test data, the error rate was reduced from 45% to 39%), but it does so without any knowledge specific to the geometry or physics of cloud fields. Another method that does use some basic rules about typical cloud sizes, cloud occurrence statistics etc., would probably give a more dramatic reduction in error rate. This assertion is motivated by the fact that, in the test data classification results, the correct cloud class is present somewhere within the region occupied by almost every cloud. If this correct class could be identified by some rule-based decision algorithm, then another filtering operation with the correct cloud class preferentially weighted could be used with an associated decrease in the error rate.

This study has succeeded in developing an algorithm that can classify pixels in WSI data according to cloud type through the use of binary decision trees. The classification is fairly accurate when tested on data other than those used for training, and analysis of the results has suggested several ways of improving the overall accuracy of the classification. Future work on this project will be focused on two main areas. First, we will assemble a training data set that is more representative of the cloud scenes collected by the WSIs and thereby improve the accuracy of the initial classification step. Second, we will develop a more physically based process of determining the actual cloud class from the output of the decision tree algorithm and thereby improve the accuracy of the second classification step.

Acknowledgments

This research is sponsored by the Laboratory Directed Research and Development Program (LDRD), Sandia National Laboratories. The Marine Physical Laboratory (MPL) of the Scripps Institute of Oceanography provided the whole-sky imagers and the processed cloud decision images used in this study. We thank Janet Shields of MPL who set up and operated the whole-sky imagers and Bob Endlick of the U.S. Army Research Laboratory, White Sands Missile Range, who provided for the siting of the WSIs at White Sands, New Mexico.

References

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and regression trees*. Wadsworth International Group, Belmont, California.
Burt, P. J. 1984. The pyramid as a structure for efficient computation. In *Multiresolution image processing and analysis*, ed. A. Rosenfeld. Springer, Heidelberg, Germany, pp. 6-35.

Ebert, E. E. 1992. Pattern recognition analysis of polar clouds during summer and winter, *Int. J. Remote Sensing*, **13**, 97-109.

Garand, L. 1986. Automated recognition of oceanic cloud patterns and its application to remote sensing of meteorological parameters, Ph.D. Thesis, University of Wisconsin, Madison, Wisconsin.

Inoue, T. 1987. A cloud type classification with NOAA 7 split-window measurements, *J. Geophys. Res*., **92**, 3991-4000.

Lamei, N., M. M. Crawford, K. D. Hutchison, and N. Khazenie. 1993. Cloud type discrimination via multispectral textural analysis, *SPIE* Proc., **1934**, 49-61.

Laws, K. I. 1980. Textured image segmentation, Ph.D. Thesis, University of Southern California, Los Angeles, California.

Lee, J., R. C. Weger, S. K. Sengupta, and R. M. Welch. 1990. A neural network approach to cloud classification, *IEEE Trans. Geosci. Remote Sens*., **28**, 846-855.

Smotroff, I. G. 1993. *Neural network cloud classification research*, Report No.: MTR-93B0000039. MITRE Corp., Bedford, Massachusetts.

Yool, S. and M. Brandley. 1992. Remote discrimination of clouds using a neural network, *SPIE* Proc., **1766**, 497-503.