# Cloud Detection in All-Sky Images via Multi-scale Neighborhood Features and Multiple Supervised Learning Techniques

Hsu-Yung Cheng [a*] , Chih-Lung Lin [b]

[a]Department of Computer Science and Information Engineering, National Central University. No.300 Jhongda Rd., Jhongli City, Taoyuan 32001, Taiwan

[b]Department of Electronic Engineering, Hwa Hsia University of Technology, Taiwan

***Abstract*: Cloud detection is important for providing necessary information such as cloud cover in many applications. Existing cloud detection methods include red-to-blue ratio thresholding and other classification based techniques. In this paper, we propose to perform cloud detection using supervised learning techniques with multi-resolution features. One of the major contributions of this work is that the features are extracted from local image patches with different sizes to include local structure and multi-resolution information. The cloud models are learned through the training process. We consider classifiers including random forest, support vector machine and Bayesian classifier. To take advantage of the clues provided by multiple classifiers and various levels of patch sizes, we employ a voting scheme to combine the results to further increase the detection accuracy. In the experiments, we have shown that the proposed method can distinguish cloud and non-cloud pixels more accurately compared with existing works.**

***Keywords*: All-sky Image; Cloud Detection; Multi-resolution; Classifier; Supervised Learning**

## 1 Introduction

[*] Corresponding author. Tel.: +886-3-4227151 ext35306, fax: +886-3-422-2681

Email address: chengsy@csie.ncu.edu.tw

With the trend of sustainable and green energy, there is a growing demand for solar energy technology. To utilize solar energy effectively, integrated and large scale photovoltaic systems need to overcome the unstable nature of solar resource (Gueymard, 2004; Heinemann et al., 2006; Lorenz et al., 2009). The ability to forecast surface solar irradiance is helpful for planning and deployment of electricity generated by different units. Numerical weather prediction information or satellite images are popular materials used for wide-range prediction (Marquez and Coimbra, 2011; Perez et al., 2002; Perez et al., 2010; Remund et al., 2008). However, the resolution of prediction with respect to space and time obtained by weather prediction information or satellite cloud images is relatively coarse compared to the resolution desired for photovoltaic grid operators. For more refined spatial and temporal resolution of irradiance prediction, researches that analyze images obtained from devices capturing skies have emerged. Ground-based sky camera systems have been proposed to capture the images of the sky (Sabburg and Wong, 1999), allowing researchers to study the relationship between the sun and clouds and the effect of clouds. Devices developed to monitor the sky presented in some of the pioneering works include Whole Sky Imager (Kassianov et al., 2005; Li et al., 2004), Whole Sky Camera (Long et al., 2006), All-Sky Imager (Kubota et al., 2003), and Total Sky Imager (Pfister et al., 2003). More recent commercial products include all-sky cameras by Eko Instruments, Oculus, SBIG, etc. These devices are useful to make up the deficiency of satellite cloud observations in terms of spatial and temporal resolutions.

Cloud coverage, configurations and types are critical factors that influence the solar irradiance. A category of research works are devoted to detecting (Long et al., 2006), classifying (Calbo and Sabburg, 2008; Heinle et al., 2010; Martínez-Chico et al., 2011), and tracking clouds (Marquez and Coimbra, 2013; Tapakis and Charalambides, 2013; Wood-Bradley et al., 2012). The relationships between cloud coverage and surface solar irradiance have been explored (Feister and Shields, 2005; Fu and Cheng, 2013; Pfister et al., 2003). It has been shown that cloud fraction and surface irradiance are negatively correlated under most conditions. In addition to providing cloud coverage information, accurate cloud detection result could further improve the cloud type classification accuracy (Cheng and Yu, 2015). It has been established that employing cloud type information in the process of short-term irradiance prediction cloud yield more accurate prediction results (Cheng and Yu, 2015).

Cloud detection in all-sky image is to decide if a pixel belongs to a cloud. Traditionally, red to blue ratio (RBR) of each pixel is used to indicate whether the dominant source of the pixel is from clear sky or clouds (Chow et al., 2011; Johnson et al., 1989, 1991; Long et al., 2006; Shields et al., 2007, 2009). Then, a threshold is applied to RBR to determine cloud pixels in a sky image. The pixels whose RBRs are lower than the threshold are classified as clear sky and the pixels whose RBRs are higher the threshold are labeled as clouds. Selecting a good threshold is very important for RBR method. The work by Long et al. (Long et al., 2006) suggested that different thresholds should be selected depending on the relative position of the pixel being classified in contrast to the positions of sun and horizon. In addition to pure color characteristics, Roy et al. (Roy et al., 2001) tried a neural network approach with a wider range of variables for cloud segmentation. West et al. (West et al., 2014) also used a neural network to classify pixels. The features they used are colors and the distance of the pixel to the sun. Under lower-visibility conditions, aerosol and thin clouds tend to cause errors in cloud determination. To improve the accuracy of the single threshold method, Huo and Lu proposed an integrated method for cloud determination under low visibility conditions (Huo and Lu, 2009). The integrated cloud-determination algorithm uses fast Fourier transform, symmetrical image features, and self-adaptive thresholds. Li (Li et al., 2011) proposed a hybrid thresholding algorithm (HYTA) for cloud detection on ground-based color images, aiming at complementing fixed thresholding and adaptive thresholding algorithms. HYTA identifies the ratio image as either unimodal or bimodal according to its standard deviation. Then, the unimodal and bimodal images are handled by fixed and minimum cross entropy (MCE) thresholding algorithms, respectively. Kazantzidis (Kazantzidis et al., 2012) tuned multiple heuristic thresholds on RGB color components to detect clouds. The above mentioned works mostly consider the features extracted from each single pixel, but do not consider the local image patch and structure around the pixel. Bernecker et al. (Bernecker et al. 2013) used color and texture as features. After applying deep belief networks to learn the structure of the features, a random forest classifier is used to classify image patches into three classes: sky, cloud, and thick cloud. Bernecker et al. proposed to utilize information of image patch. However, they used fixed-size patches for training and classification without considering multiresolution information. Patches with sizes that are too large would include features from both sky and clouds. On the other hands, patches with sizes that are too small might not include enough information to represent the appearance of the clouds.

In this paper, we propose to perform cloud detection via extracting features from local image patches with various sizes. Patches of different sizes extract information at different levels of resolution. For classification, we utilize multiple supervised learning techniques. We regard the cloud detection problem as a two-class classification problem. In other words, we classify each pixel in the image as cloud or non-cloud. The cloud models are learned through the training process. We consider classifiers including Support Vector Machine (SVM), random forest, and Bayesian classifier. To extract features from each pixel, we calculate the red and blue ratio (RBR) as well as the color components of various color models including RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), and YCbCr. To take advantage of the clues provided by multiple classifiers and multi-level resolution, we employ a scheme to combine multiple classification results to further increase the cloud detection accuracy. The methodology, including the features and the classifiers, is elaborated in Section 2. In Section 3, the proposed system framework is validated using a set of experimental images with manually labeled ground truth. The experimental results using different classifiers are demonstrated and discussed. Finally, conclusions are made in Section 4.

# 2 Methodology

The proposed system framework is illustrated in Fig. 1. For each all-sky image, Hough line transform is performed first to detect the vertical line of the sun, which is caused by the CCD device when capturing all-sky images. The pixels on this line often has bright intensities and could be confused as cloud pixels. After detecting and eliminating the vertical line of the sun, the rest of the pixels in the image are classified as cloud or non-cloud. The input images are RGB color images. For each all-sky image, the color components in various color space are computed. The color models considered in this work include RGB, HSV, and YCbCr. In addition to the above mentioned color components, the RBR of each pixel is also calculated and considered as a feature. To perform pixel-wise classification, all the color components and the RBR of the local image patches around a pixel are collected and concatenated as a feature vector for the pixel. Training samples are obtained from manually labeled ground truth images.

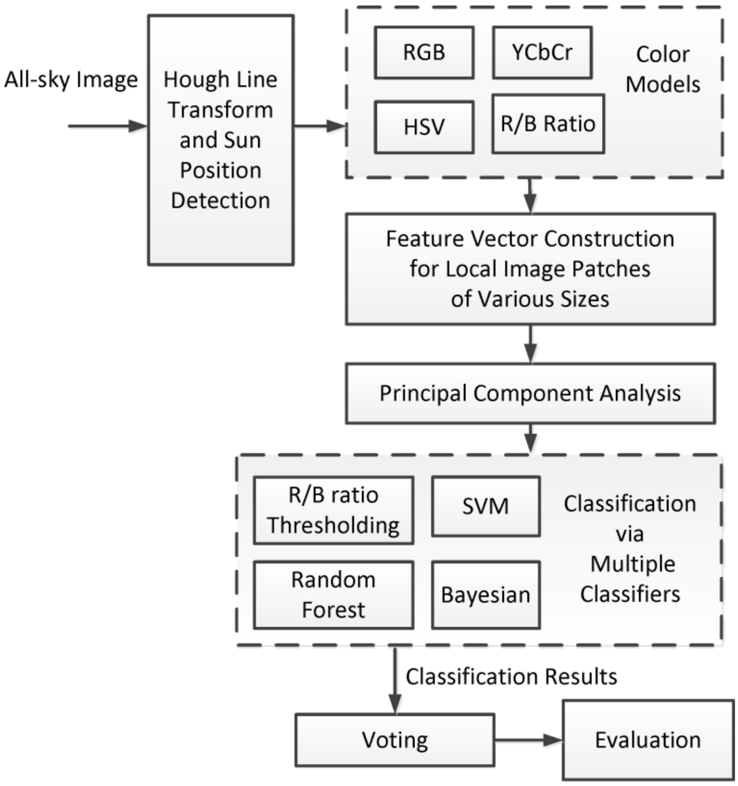
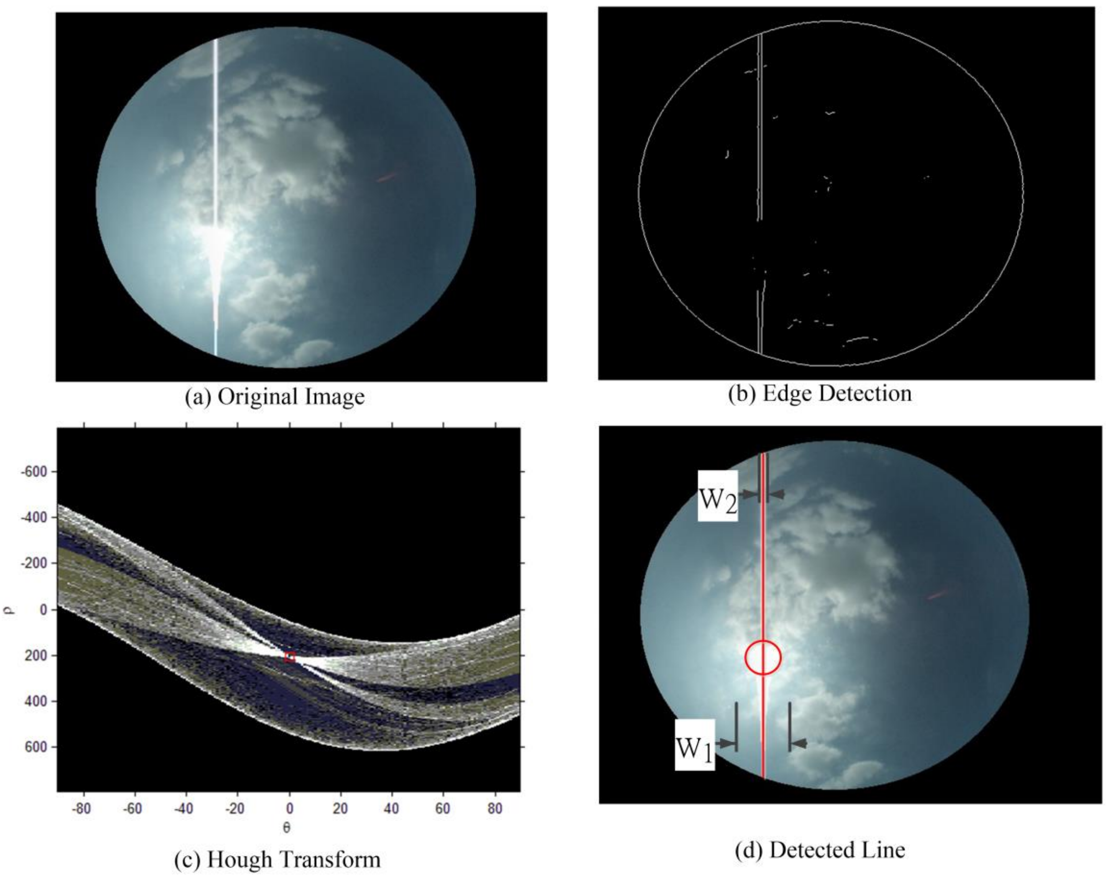



Fig. 1. System framework

## 2.1 Hough Line Transform and Sun Position Detection

Hough transform (Shapiro, 2011) is used to detect the vertical line of sun in an all-sky image. The procedure of detecting lines can be regarded as finding the coefficients of the line equations using a voting mechanism. The procedure of detecting lines via voting in the parameter space can be achieved by dividing the parameter space into grids. Because all the pixels satisfying a certain line equation would vote to the same grid, a high vote would appear in the corresponding grid in the parameter space. Hough transform re-parameterizes the line equation as $x\cos\theta + y\sin\theta = \rho$ to avoid uing the slope parameter for line equation y=*m*x+*b*. Because possible values for the slope parameter *m* ranges from minus infinity to infinity, it would be infeasible to find the slope parameter *m* via grid search. After, re-parameterizing the line equation, the range of the parameter $\rho$ can be set according to the width and height of the image. And the range of the parameter $\theta$ is from $-180^\circ$ to $180^\circ$. Fig. 2 displays an example of Hough line detection on an image. After detecting the vertical line, the sun position is determined by accumulating the intensities of the pixels along x direction in a window with width $w_1$. The position with the highest accumulated intensity is the center of the sun. The pixels in the line window with a fixed width $w_2$ are eliminated from the image. The pixels within the sun position and the line window with width $w_2$ are determined as non-cloud pixels and do not have to go through the subsequent classification steps. The values of $w_1$ and $w_2$ are determined depending on the size of the all-sky images. In our experiments, we set $w_1$ and $w_2$ as 60 and 12 pixels, respectively.




(a) Original Image (b) Edge Detection

(c) Hough Transform (d) Detected Line

Fig. 2. Hough Line Detection and Sun Position Detection

## 2.2 Color Models

RGB is a very common color model, being used in most computer systems. It is an additive color model based on tri-chromatic theory. RGB is easy to implement. However, it is non-linear with visual perception, and the specification of colors is semi–intuitive. HSV is a color model that describes colors in terms of hue, saturation and value components (Gonzalez, 2002). Hue is expressed as a number from 0 to 360 degrees. The hue component of red starts at 0, green starts at 120, and blue starts at 240. Saturation is the amount of gray in the color. And the value component describes the brightness or intensity of the color. YCbCr is a color space used in video and digital photography systems. Y is the luminance component, and Cb and Cr are the blue-difference and red-difference chroma components. HSV and YCbCr color components can be obtained from RGB color components using color model transformation equations (Gonzalez, 2002; Poynton, 2003). Although the color models are not independent and including color components from different color models may introduce redundancy in the feature vector, considering various color models still provides the classifier more information that is beneficial to performing classification.

### 2.3 Feature Vector Construction for Local Image Patches of Various Sizes

For each pixel, local image patches with various sizes are used to extract features. The size of the image patch at level $i$ is $L_i \times L_i$, $i = 1 \cdots \ell$, where $\ell$ denotes the total number of levels. For each local image patch, the color components and the red to blue ratio (RBR) of all the pixels in the patch are concatenated to form a feature vector. Consequently, the dimension of the feature vector is $L_i \times L_i \times 10$. There are $\ell$ feature vectors constructed for each pixel.

### 2.4 Dimension Reduction

We apply Principal Component Analysis (PCA) (Duda et al., 2001) on the feature vectors to reduce their dimensions. Based on the assumption that the importance of the features lies in the variability of the data, PCA chooses principal components along the directions with the largest variance of the data distribution first. The principal components are a set of new orthogonal bases that can be used to re-express the data in order to reduce the correlation among different variables. Suppose that the original dataset has $N_{\mathrm{Samples}}$ samples and each sample has $D_1$ variables. The data matrix $X$ is established with each sample as a column vector. Therefore the data matrix $X$ has $N_{Samples}$ columns and $D_1$ rows. If we would like to reduce the feature dimension to $D_2$, then we need to select $D_2$ principal components. PCA constructs a matrix $X^TX$, which is a matrix proportional to the sample covariance matrix of the dataset *X*. The first $D_2$ eigenvectors of $X^TX$ whose corresponding eigenvalues are largest are chosen as principal components. To determine the desired number of dimensionality $D_2$, we check the eigenvalue ratio $R_{Eigenvlaue}$

$$R_{Eigenvalue} = \frac{\sum_{k=1}^{D_2} |\lambda_k|}{\sum_{k=1}^{D_1} |\lambda_k|} \quad (1)$$

In Eq. (1), $\lambda_k$ denotes the $k^{\text{th}}$ Eigenvalue of $X^TX$. The first $D_2$ Eigenvectors are preserved so that $R_{Eigen}$ is larger than a threshold $Thr_{PCA}$. The selection of $Thr_{PCA}$ is discussed in the experiments in Section 3.

## 2.5 Classifiers

### 2.5.1 Random Forest

Classification and Regression Tree (CART) is a systematic procedure that learns decision trees proposed by Breiman (Breiman et al., 1984). The splitting rules of the tree include an attribute value test at each node of the tree. Starting from the root node, all training data is used to split the root node. And the tree is built recursively. Considering all the possible splitting rules, CART would construct the tree by selecting the splitting rule that can maximize the impurity drop when a node is added. The impurity measures the condition of mixed class labels at each node. The goal is to make the class labels at each node as "pure" as possible. The splitting process stops when all the samples in a node have the same class label, or when the measure of purity at the child nodes cannot be improved compared with its parent node. After a decision tree is built, it might need to be pruned using a cross-validation procedure. The reason for pruning is that some branches of the tree might over-fit the training data. In our experiment, we use 10 fold cross validation. Instead of growing a single decision tree, random forest grows an ensemble of trees and lets them vote for the most popular class label. In this work, we adopt random split selection (Dietterich 1999) to build the ensemble of trees. At each node, the split is selected at random from the K best splits. The features for the split rules are randomly selected. It reduces the correlation between the trees and improves the efficiency of training.

### 2.5.2 Support Vector Machine

The Support Vector Machine (SVM) learns a set of hyperplanes that maximize the margins between the hyperplanes and the training samples in order to lower the classification error of unknown testing samples. The motivation of SVM is that an ideal decision boundary should have the largest distance to the nearest training sample of all the classes. However, it might be infeasible to separate data samples using linear hyperplanes in practice. Therefore, soft margins and kernel functions are applied in the SVM in practice. We apply SVM with radial basis functions (RBF) as one of the classifiers in this work. For the details of SVM, please refer to the work by Cristianini and Shawe-Taylor (Cristianini and Shawe-Taylor, 2000).

### 2.5.3 Bayesian Classifier

Bayesian classifier aims at minimizing the probability of misclassification by classifying a sample $x$ to the class $\omega_k$ with the largest posterior probability $P(\omega_k \mid x)$. Since the posterior probability $P(\omega_k \mid x)$ itself is unknown, we need to transform the problem using the probabilities that can be obtained via training samples. Bayesian classifier uses the Bayes' theorem to re-express the posterior probability using

$$P(\omega_k \mid x) = \frac{P(\omega_k)P(x \mid \omega_k)}{P(x)} \tag{2}$$

In Eq. (2), $P(\omega_k)$ denotes the prior probability, which is independent of the testing sample. In other words, $P(\omega_k)$ states how likely a pixel belongs to cloud or non-cloud regardless of its observed feature vector. It is possible to use meteorological conditions and weather forecast report to determine different prior probabilities $P(\omega_k)$ for each day. However, we use the same prior probabilities for both cloud and non-cloud classes for simplicity, and no meteorological information is required to be involved as prior knowledge in our decision process. The class conditional probability $P(x \mid \omega_k)$ in Eq. (2) can be learned from the training samples. We use Gaussian distributions

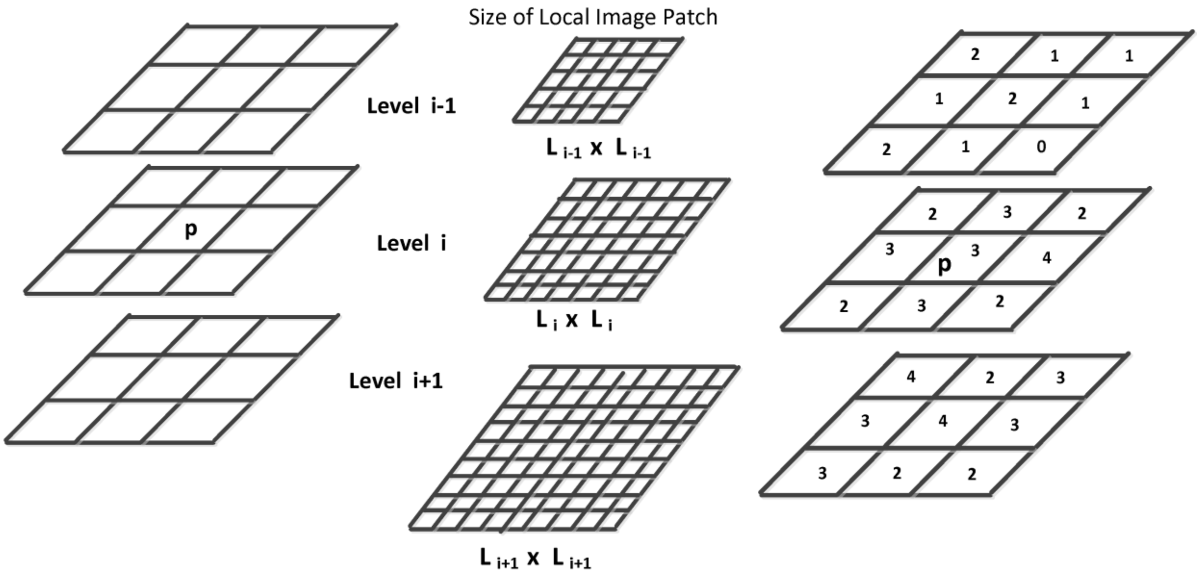
$$P(x \mid \omega_k) = \frac{1}{(2\pi)^{p/2} \mid \Sigma_k \mid^{1/2}} e^{-\frac{1}{2}(x-\mu_k)\Sigma_k(x-\mu_k)^T} \tag{3}$$

to model the class conditional probability $P(x \mid \omega_k)$ for each class. To learn the parameters of Gaussian functions, training samples from each class are used to calculate the sample mean vector $\mu_k$ and the sample covariance matrix $\Sigma_k$ for the class. The probability of the sample $P(x)$ in Eq. (2) does not depend on the class label and can be neglected in the decision process.

## 2.6 Combining Results of Multiple Level Neighborhoods and Classifiers

The concept of a multiple expert system is to take advantage of the clues provided by multiple classifiers. Instead of majority voting, we use a different voting scheme to combine the results of multiple-level patches and classifiers. As shown in Fig. 3, considering a $3\times3$ neighborhood around a pixel *p* at level *i*, its previous level $i-1$ and its next level $i+1$, voting is performed in the scale space of its $3\times3\times3$ neighborhood. For the pixels in level $i-1$ in Fig. 3 (a), the size of the local image patch used for feature vector construction is $L_{i-1}\times L_{i-1}$ in Fig. 3 (b). Similarly, image patches of size $L_i \times L_i$ and $L_{i+1}\times L_{i+1}$ are used for level *i* and level *i+1*, respectively. The voting scheme takes into account the classification results from 4 classifiers: RBR thresholding, SVM, random forest, and Bayesian classifier. In other words, there are $27\times4$ votes for the pixel. Suppose $x_{Level_i}$ denotes the feature vector of a pixel at level *i*, and the number of votes in the neighborhood classified as cloud at level *i* is denoted as $V_{cloud}(x_{Level_i})$. The decision for a pixel at level *i* is determined by $V_{cloud}(x_{Level_i}) > N_v$. In other words, if there are more than $N_v$ votes in the $3\times3\times3$ neighborhood of a pixel at level *i*, the pixel is classified as a cloud pixel at this level. Considering the example illustrated in Fig. 3 (c), the numbers represent the votes in the $3\times3\times3$ neighborhood of pixel *p* at level *i*. Summing up the numbers in Fig. 3 (c), we can obtain that $V_{cloud}(x_{Level_i})$ =61. If the threshold $N_v$ equals to 57, then pixel *p* is classified as cloud at level *i*. To combine the decision at different levels, the probability $P(x \in cloud \mid \underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))$ is computed. Suppose $\underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud)$ denotes the number of levels at which the pixel is determined as cloud among all levels $i$=1 to $\ell$ . $P(x \in cloud \mid \underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))$ states the probability of a pixel belonging to cloud given the number of levels that the pixel is determined as cloud. If $P(x \in cloud \mid \underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))$ is larger than $P(x \in noncloud \mid \underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))$, the final decision would classify the pixel be a cloud pixel. The probability $P(x \in cloud \mid \underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))$ can be expressed as Eq. (4) using Bayesian rules of conditional probability. In Eq. (4), the term $P(\underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))$ is independent of class label and would not affect the decision. The prior probabilities $P(x \in cloud)$ and $P(x \in noncloud)$ are assumed to be equal as stated in Section 2.4.3. The likelihood term $P(\underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud) \mid x \in cloud)$ is learned from the training dataset by constructing the normalized histogram of $\underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud)$ using all ground truth cloud pixels.

$$P(x \in cloud \mid \underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud)) = \frac{P(x \in cloud)P(\underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud) \mid x \in cloud)}{P(\underset{i=1\sim\ell}{Num}(x_{Level_i} \in cloud))} \quad (4)$$




(a) (b) (c)

Fig. 3. Voting in the scale space of a $3 \times 3 \times 3$ neighborhood: (a) Structure of the scale space neighborhood (b) Size of the local image patch at different layers (c) Number of votes in the scale space neighborhood.

## 3 Experimental Results

In this work, the device used to capture the all-sky images is the all-sky camera manufactured by the Santa Barbara Instrument Group (SBIG). The field of view is 185º. The focal length is 1.44 mm. And the focal ratio range is f/1.4–f/16. The resolution of the bitmap images is 640 x 480. We manually marked the ground truth of cloud pixels in 250 images for training and testing. These images are collected from January to June, 2014. With the ground truth labels of the images, we are able to calculate the detection accuracy at pixel level. We adopt 10-fold cross validation to calculate the average detection accuracy, precision and recall rate. The definitions of detection accuracy, precision and recall rate are listed in Eq. (5) to Eq. (7).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

In Eq. (5) to Eq. (7), true positive *TP* is the number of cloud pixels correctly detected. True negative *TN* is the number of non-cloud pixels that are correctly classified. False positive *FP* is the number of non-cloud pixels that are incorrectly classified as clouds. False negative *FN* is the number of cloud pixels that are incorrectly classified as non-cloud.

In this work, the RGR thresholding method proposed by Long (Long et al., 2006) will be used as the baseline method for comparison. In Long's work, an RBR threshold is recommended for the Whole Sky Camera and several thresholds are suggested to be used for the Total Sky Imager. Since the desired threshold varies due to different devices and weather conditions, we perform an experiment to test the best threshold for our all-sky camera. Also, to avoid false positive detection at highlighted regions around the sun, we employ an upper bound threshold. Therefore, two thresholds, $Thr_{upper}$ and $Thr_{lower}$, are used in the experiments. A pixel is classified as cloud if its RBR is higher than $Thr_{lower}$ and lower than $Thr_{upper}$. We perform experiments on several thresholds to select the best thresholds for our dataset. As shown in Fig. 4, we have observed that $Thr_{lower}$ = 0.8 and $Thr_{upper}$ = 0.9 yield the best detection accuracy for our dataset. In the rest of the experiments, we use RBR thresholding with $Thr_{lower}$ = 0.8 and $Thr_{upper}$ = 0.9 as a baseline method for comparison. However, even with the best selected RBR thresholds, the cloud detection result is not satisfying. The thresholds $Thr_{lower}$ = 0.8 and $Thr_{upper}$ = 0.9 might cause some false positives for certain images while causing some false negatives for other images. Therefore, neither raising or lowering the threshold could improve the detection results by thresholding.

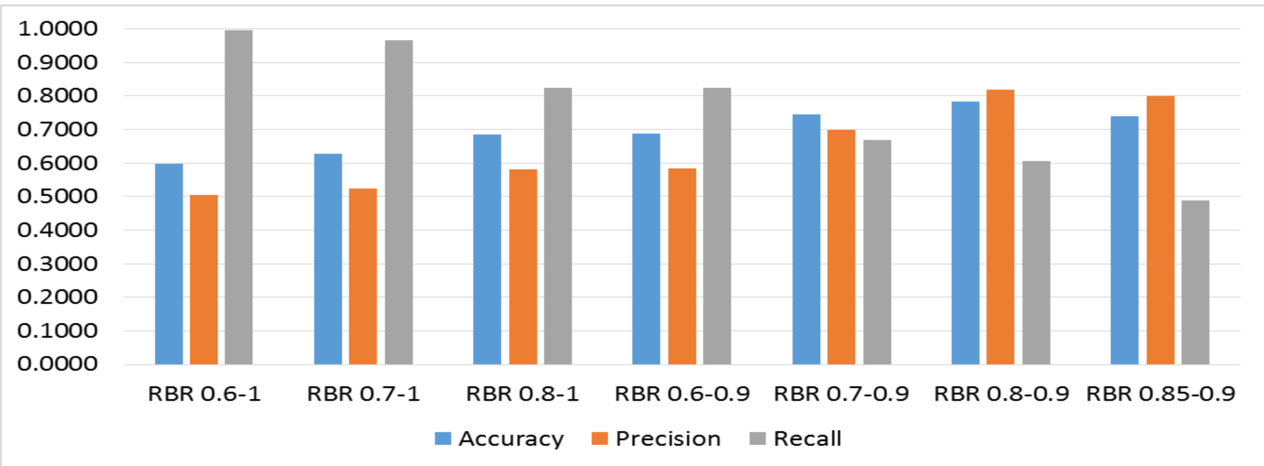



Fig. 4. Cloud detection accuracy using various RBR thresholds

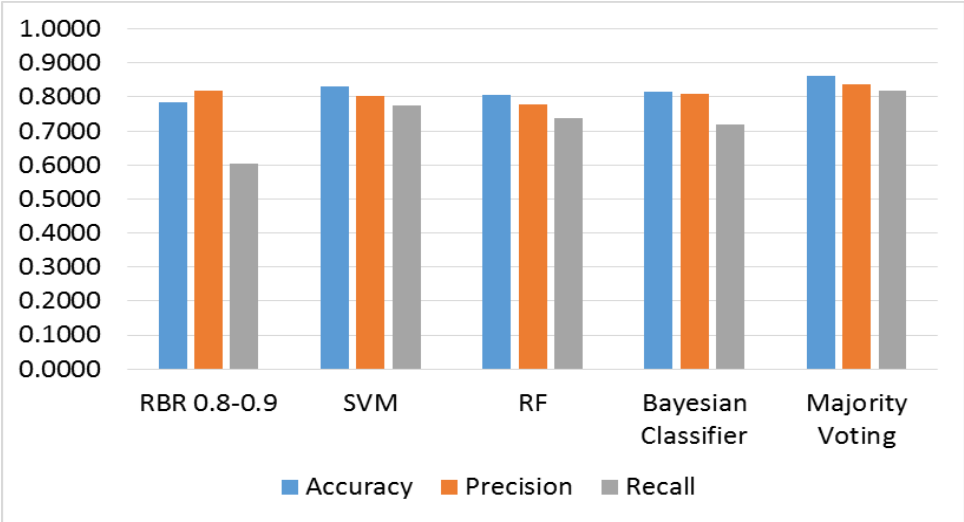



Fig. 5. Comparisons of detection accuracy using different classifiers with single pixel color information

To observe classification results of different classifiers, the detection accuracy of different classifiers based on single pixel color information are plotted in Fig. 5. Compared with other classifiers, RBR thresholding with $Thr_{lower}$ = 0.8 and $Thr_{upper}$ = 0.9 has the lowest detecction accuracy. Majority voting of the four detection methods can yield both better precision and recall rates. With voting schemes that combine the information from multiple classifiers, the accuracy can be enhanced compared with individual single classifiers. However, utilizing only single pixel color information is not sufficient to give satisfying detection accuracy. Applying features extracted from local image patch is able to further enhance the detection results.

When applying the proposed cloud detection method, we use five levels of local image patches with different sizes, i.e. $\ell$=5. The size at each level is $L_1 = 5\times5$, $L_2 = 10\times10$, $L_3 = 15\times15$, $L_4 = 20\times20$, $L_5 = 25\times25$. To observe the effect of parameter $Thr_{PCA}$ for dimension reduction at each level, we perform an experiment using feature vector constructed at each single level with SVM as the classifier for different settings of $Thr_{PCA}$. A proper $Thr_{PCA}$ usually locates between 90%~99% and is selected empirically. Typically, the accuracy of classification would increase as $Thr_{PCA}$ goes from 100% (which means no dimensionality reduction at all) to 99%. The accuracy of classification would continue increasing until $Thr_{PCA}$ reaches a certain value, which is caused by the benefit of dimensionality reduction. After that, the accuracy of classification would start to decrease due to too much information loss. We plot the cross-validated detection accuracy in Fig. 6. From Fig. 6, we can observe that the detection accuracy at single level using SVM is highest for $Thr_{PCA}$=97% at levels $L_1$ and $L_2$. At levels $L_3$, $L_4$ and $L_5$, the parameter $Thr_{PCA}$=95% yields better results. Therefore, for levels $L_1$ and $L_2$, $Thr_{PCA}$=97% is selected; for levels $L_3$, $L_4$ and $L_5$, $Thr_{PCA}$=95% is selected.

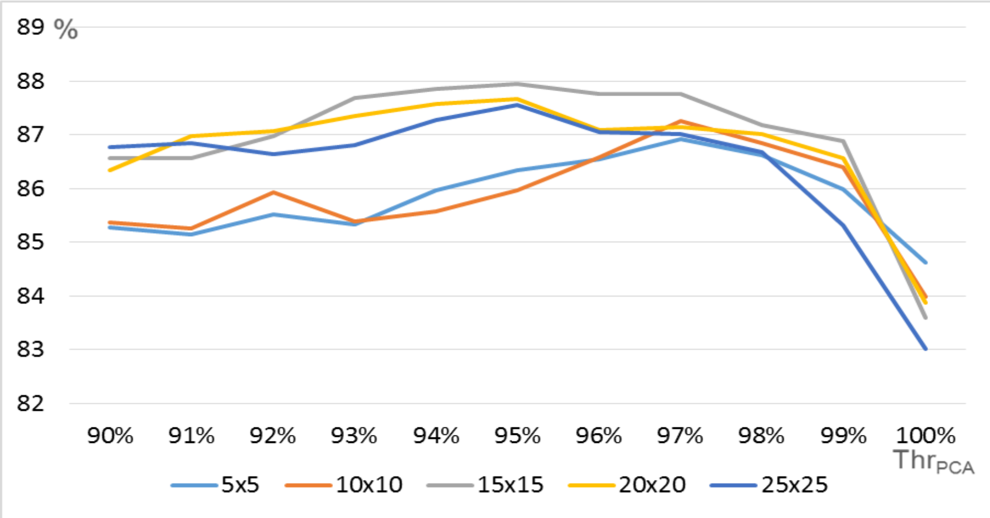



Fig. 6. Detection accuracy with different $Thr_{PCA}$ settings at each level using SVM

To combine results of multiple level patches and classifiers, the threshold for voting $N_v$ needs to be determined. The detection accuracy of combining the results using different $N_v$ settings is plotted in Fig. 7. As shown in Fig. 7, when $N_v$ ranges from 50 to 70, the detection accuracy is higher. We select $N_v$=57 for the proposed method.

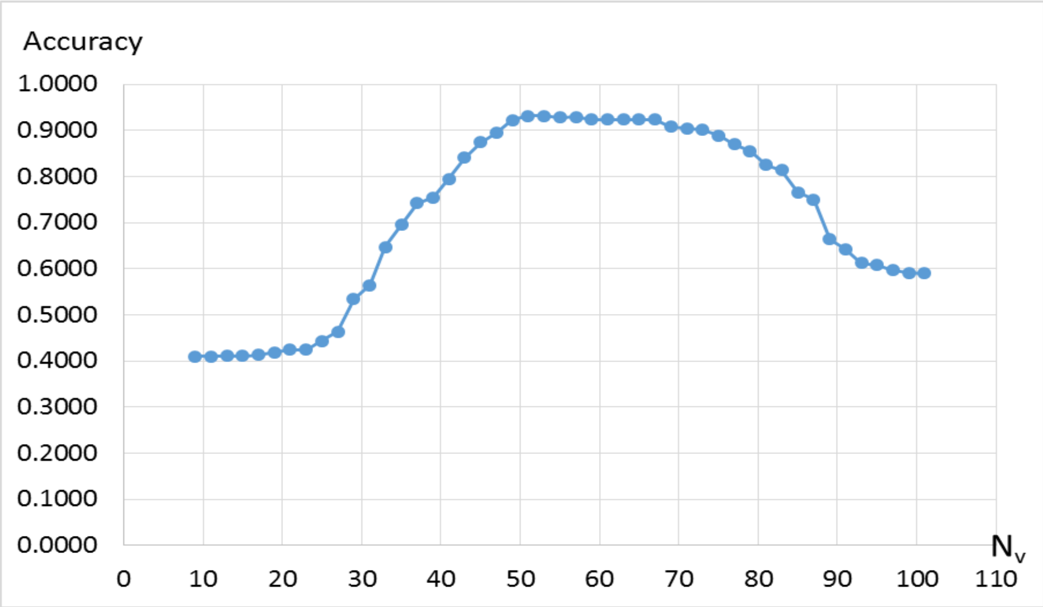



Fig. 7. Detection accuracy with different $N_v$ settings

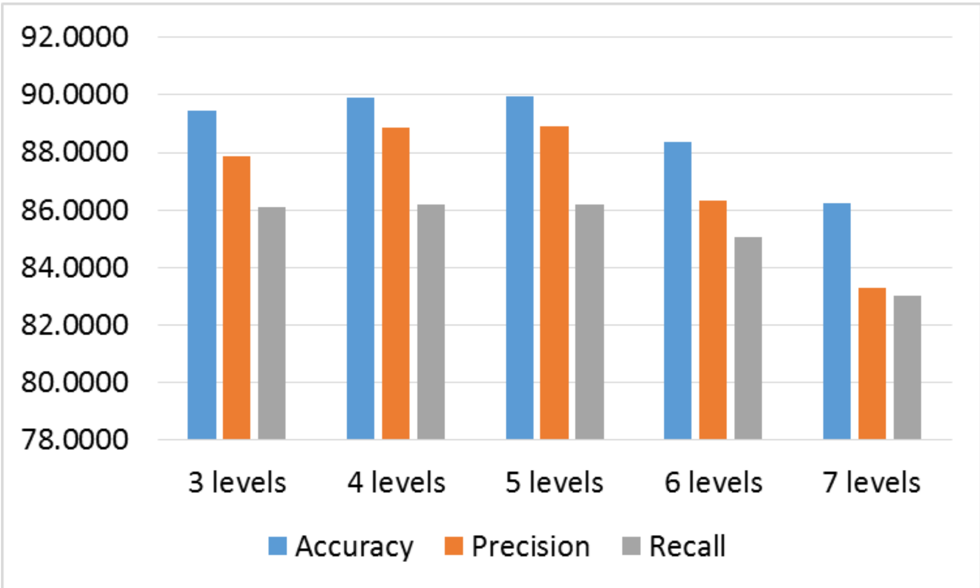
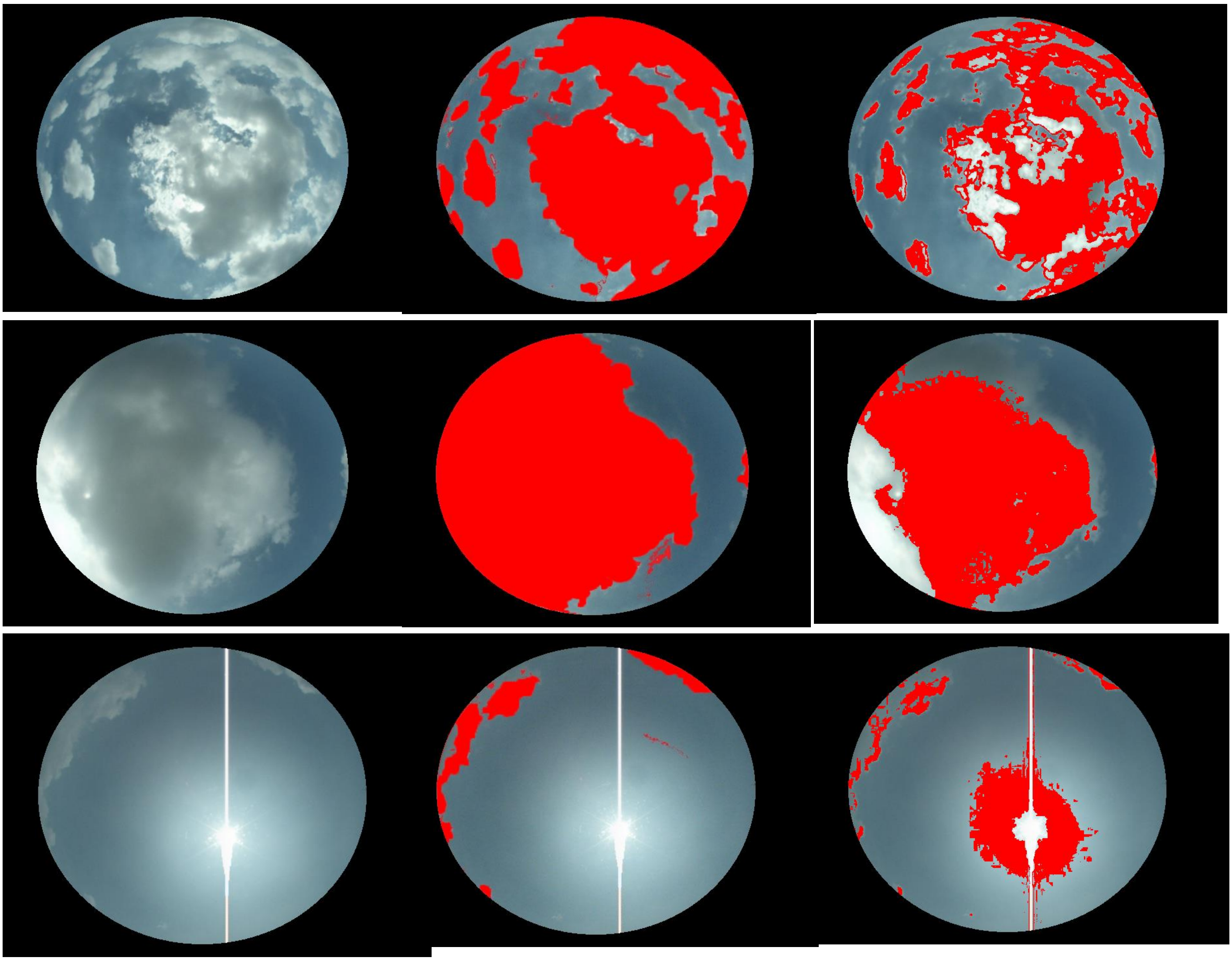



Fig. 8. Detection accuracy using different number of levels

To test the number of levels required to yield better detection results, we plot the detection accuracy using different number of levels in Fig. 8. Note that for the 6$^{th}$ level and 7$^{th}$ level, the size of the local image patch is $L_6 = 30 \times 30$ and $L_7 = 35 \times 35$. We can observe that using 4 or 5 levels results in better detection accuracy. When involved with levels with image sizes that are too large, the detection accuracy drops.



(a) (b) (c)

Fig. 9. Selected results: (a) Original images; (b) Detection results of the proposed method; (c) Detection results of RBR 0.8-0.9

Selected cloud detection results are shown in Fig. 9 (b). The proposed method using features from multi-scale local image patches can accurately detect clouds in the all-sky images. The pixels within the vertical line and the solar disk would not be detected as clouds even though their intensities are high. The Hough line detection and sun position detection successfully eliminated those pixels before performing classification. Compared with detection results of RBR 0.8-0.9 in Fig. 9 (c), the proposed method can detect cloud pixels with satisfying accuracy with the proposed multi-level local patch feature extraction mechanism and combination of multiple expert decision.

To summarize the detection accuracy, the detection accuracy of various methods are plotted in Fig. 10. We compare the proposed method with ANN (Roy et al., 2001) and HYTA (Li et al., 2011). ANN utilized a feedforward back-propagation neural network to perform detection. HYTA employs dynamic thresholding based on minimum cross entropy when necessary. The ANN and HYTA methods outperform traditional RBR thresholding. Nevertheless, the accuracy of ANN and HYTA still has room for improvement. Using the single pixel color components described in Section 2.2 and utilizing SVM as the classifier can yield slightly improved accuracy compared with ANN and HYTA. Incorporating feature vector extracted from single level 15x15 neighborhood patch can further improve the accuracy compared with using only information from single pixel. The proposed method utilizing features extracted from multi-level neighborhood yields the best accuracy since multiscale information is considered.

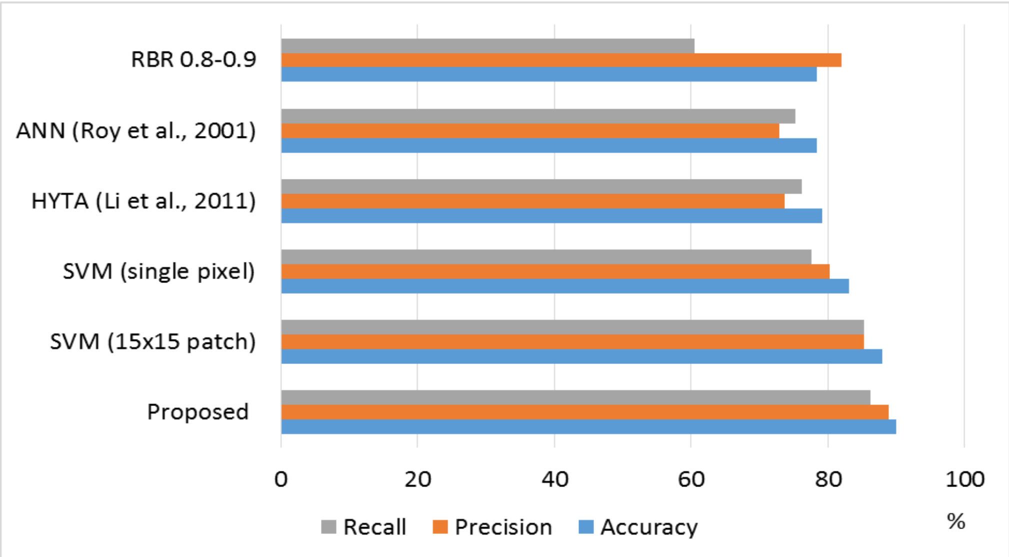



Fig. 10. Comparisons of different methods

# 4 Conclusions

With the development of all-sky cameras, the cloud conditions in the sky can be monitored and useful information can be extracted for solar irradiance prediction with refined spatial and temporal resolutions. Clouds play a critical role in affecting the amount of solar irradiance penetrating the atmosphere. With more accurate cloud detection schemes, subsequent prediction modules that forecast solar irradiance could benefit a lot from the enhanced detection results. In this work, supervised learning methods are utilized to train various classifiers that can distinguish cloud pixels from non-cloud pixels in all-sky images. The classifiers implemented in this work include RBR thresholding, SVM, random forest, and Bayesian classifier. We propose to use features extracted from multi-level local image patches with different sizes to include local structure and multi-resolution information. Final decision is made according to multi-level classification results by various classifiers. A challenging dataset with ground truth labels is used to validate the detection schemes. Experiments have also shown that the proposed detection method yields better results than both fixed and dynamic RBR thresholding. Combining the information of multiple classifiers using voting can improve the detection accuracy. It is also validated that using color information in multi-level local neighborhood instead of only a single pixel is very helpful to improve the detection accuracy. To apply the proposed method on different all-sky cameras, images captured by various cameras can be added into the training set to enhance the robustness of the detector. For the selection of parameters $Thr_{PCA}$ and $N_v$ for different devices and sites, if the number of levels and feature length are fixed, the desired parameters should not be seriously affected even if the training samples are changed.


## Acknowledgements

The financial support provided by the Ministry of Science and Technology of Taiwan is gratefully acknowledged.


## References


Bernecker, D., Riess, C., Christlein, V., Angelopoulou, E., Hornegger, J., 2013. Representation learning for cloud classification. Lecture Notes in Computer Science 8142, 395-404.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 1984. Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Calbo, J., Sabburg, J., 2008. Feature extraction from whole-sky ground-based images for cloud-type recognition. J. Atmos. Ocean. Tech. 25, 3-14.

Cheng, HY, Yu, C.C., 2015. Multi-Model Solar Irradiance Prediction Based on Automatic Cloud Classification. Energy 91, 579-587.

Cheng, H.Y., Yu C.C., 2015. Block based cloud classification with statistical features and distribution of local texture features. Atmospheric Measurement Techniques 8, 1173–1182.

Chow, C.W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., Washom, B., 2011. Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. Solar Energy 85, 2881-2893.

Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, 2000.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern classification. John Wiley & Sons, 2001; 2nd edn.

Feister, U. and Shields, J., 2005. Cloud and radiance measurements with the VIS/NIR daylight whole sky imager at Lindenberg (Germany). Meteorol. Z. 14, 627-639.

Fu, C.L., Cheng, H.Y., 2013. Predicting solar irradiance with all-sky image features via regression. Solar Energy 97, 537-550.

Gonzalez, R.C., Woods, R.E., 2002. Digital Image Processing 2nd Edition, Prentice Hall.

Gueymard, C.A., 2004. The sun’s total and spectral irradiance for solar energy applications and solar radiation models, Solar Energy 76 (4), 423-453.

Heinemann, D., Lorenz, E., Girodo, M., 2006. Solar Irradiance Forecasting for the Management of Solar Energy Systems, Solar 2006, Denver, CO, USA.

Heinle, A., Macke, A., Srivastav, A., 2010. Automatic cloud classification of whole sky images. Atmos. Measur. Technol. 3, 557-567.

Huo, J., Lu, D., 2009. Cloud determination of all-sky images under low visibility conditions. J. Atmos. Ocean. Technol. 26 (10), 2172-2180.

Johnson, R., Hering W., Shields, J., 1989. Automated visibility and cloud cover measurements with a solid-state imaging system. Tech. Rep., University of California, San Diego, Scripps Institution of Oceanography, Marine Physical Laboratory, SIO Ref. 89-7, GL-TR-89-0061, 128.

Johnson, R., Shields, J., Koehler, T., 1991. Analysis and interpretation of simultaneous multi-station whole sky imagery. Marine Physical Laboratory. Scripps Institution of Oceanography, University of California San Diego, SIO 91-3, PL-TR-91-2214.

Kassianov, E., C. N. Long, and M. Ovtchinnikov, 2005. Cloud sky cover versus cloud fraction: Whole-sky simulations and observations. J. Appl. Meteor. 44, 86-98.

Kazantzidis, A., Tzoumanikas, P., Bais, A.F., Fotopoulos, S., Economou, G., 2012. Cloud detection and classification with the use of whole-sky ground-based images, Atmospheric Research 113, 80-88,

Kubota, M., T. Nagatsuma, and Y. Murayama, 2003: Evening corotating patches: A new type of aurora observed by high sensitivity all-sky cameras in Alaska. Geophys. Res. Lett. 30, 1612.

Li, Z., M. C. Cribb, F.-L. Chang, and A. P. Trishchenko, 2004: Validation of MODIS-retrieved cloud fractions using whole sky imager measurements at the three ARM sites. Proc. 14th ARMScience Team Meeting, Albuquerque, NM, Atmospheric Radiation Measurement Program 6, 2-6.

Li, Q., Lu, W., Yang, J., 2011: A Hybrid Thresholding Algorithm for Cloud Detection on Ground-Based. Journal of Atmospheric and Oceanic Technology 28, 1286–1296.

Long, C. N., Sabburg, J., Calb´o, J., and Pag`es, D., 2006. Retrieving cloud characteristics from ground-based daytime color all-sky images. J. Atmos. Ocean. Tech. 23, 633-652.

Lorenz, E., Hurka, J., Heinemann, D., Beyer, H.G., 2009. Irradiance Forecasting for the Power Prediction of Grid-Connected Photovoltaic Systems, IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing 2, 2-10.

Marquez, M., Coimbra, C.F.M., 2011. Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the NWS database. Solar Energy 85, 746-756.

Marquez, R., Coimbra, C.F.M., 2013. Intra-hour DNI forecasting based on cloud tracking image analysis, Solar Energy 91, 327-336

Martínez-Chico, M., Batlles, F.J., Bosch, J.L., 2011. Cloud classification in a mediterranean location using radiation data and sky images, Energy 36 (7), 4055-4062.

Perez, R., Ineichen, P., Moore, K., Kmiecik, M., Chain, C., George, R., Vignola, F., 2002. A new operational model for satellite-derived irradiances: description and validation. Solar Energy 73, 307-317.

Perez, R., Kivalov, S., Schlemmer, J., Hemker Jr., K., Renne´, D., Hoff, T., 2010. Validation of short and medium term operational solar radiation forecasts in the US. Solar Energy 84, 2161-2172.

Pfister, G., McKenzie, R. L., Liley, J. B., Thomas, A., Forgan, B. W., and Long, C. N., 2003. Cloud coverage based on all-sky imaging and its impact on surface solar irradiance. J. Appl. Meteorol. 42, 1421-1434.

Remund, J., Perez, R., Lorenz, E., 2008. Comparison of solar radiation forecasts for the USA. Proc. of the 23rd European PV Conference, 1.9-4.9, Valencia, Spain.

Roy, G., Hayman, S., Julian, W., 2001. Sky analysis from CCD images: cloud cover. Lighting Research and Technology 33 (4), 211-222.

Sabburg, J., Wong, J., 1999. Evaluation of a Ground-Based Sky Camera System for Use in Surface Irradiance Measurement. Journal of Atmospheric and Oceanic Technology 16, 752-759.

Shapiro, L. and George, C. S., 2001. Computer Vision. Prentice Books, Upper Saddle River.

Shields, J., Karr, M., Burden, A., Johnson, R., Hodgkiss, W., 2007. Continuing support of cloud free line of sight determination including whole sky imaging of clouds. Final Report for ONR Contract N00014-01-D-0043DO #13, Marine Physical Laboratory, Scripps Institution of Oceanography, University of California San Diego, Technical Note 273.

Shields, J., Karr, M., Burden, A., Johnson, R., Mikuls, V., Streeter, J., Hodgkiss, W., 2009. Research toward Multi-Site Characterization of Sky Obscuration by Clouds. Final Report for Grant N00244-07-1-009, Marine Physical Laboratory, Scripps Institution of Oceanography, University of California San Diego, Technical Note 274.

Tapakis, R., Charalambides, A.G., 2013. Equipment and methodologies for cloud detection and classification: A review. Solar Energy 95, 392-430.

West, S.R., Rowe, D., Sayeef, S., Berry, A., 2014. Short-term irradiance forecasting using skycams: Motivation and development. Solar Energy 110, 188–207.

Wood-Bradley, P., Zapata, J., Pye, J., 2012. Cloud tracking with optical flow for short-term solar forecasting. 50th conference of the Australian Solar Energy Society, Melbourne, December.