A NOVEL CLOUD DETECTION AND SATELLITE ATTITUDE PLANNING METHOD FOR CAPTURING CLOUDLESS REMOTE SENSING IMAGES

[1]CHIN-SHYURNG FAHN, [2]GER LING, [2]MING-YUAN YEH, [1]MENG-LUEN WU, [1]YI-CHUNG LAN, [1]WEI-YU CHAO, [1]HSIAO-CHI WU, [1]YU-CHI HUANG, [1]TING-YU LU

[1]National Taiwan University of Science and Technology, Taipei, Taiwan (R.O.C)
[2] National Space Organization, Hsinchu, Taiwan (R.O.C)
Email: [1]csfahn@mail.ntust.edu.tw, [2]se01@nspo.narl.org.tw, [2]marco@nspo.narl.org.tw, [1]d10015015@mail.ntust.edu.tw

---

Abstract: Clouds are unwanted objects in satellite images because they cover up the ground contents. To avoid unnecessary clouds appear in satellite's view, we develop a cloud detection system that helps satellite system avoid or remove the clouds. To detect the clouds correctly, in this paper, many cloud detection methods are proposed. First, apply the Otsu's method to select the appropriate threshold for binarization to filter out possible cloud layers because the clouds are often high brightness regions in satellite images. However, the threshold is only for brightness filtering, considering the color components, we also use Grab cut and k-means algorithm to perform more detailed cloud segmentation. The texture of clouds has strong characteristics among other objects in a satellite image. Therefore, after segmentation, extract the texture features such as local binary pattern (LBP) and Histogram of oriented gradients (HOG) as materials for building cloud classifiers. The classifier is built upon the support vector machine (SVM), and the classifier is used to determine whether an image is within a cloud region. The experimental results show that the proposed method can effectively detect the clouds in a satellite image. With the combination of HoG texture feature and the linear SVM, the proposed cloud classifier can reach 99% of accuracy.

Keywords: Cloud Detection, Satellite Attitude Planning, Satellite Image, Grab Cut, Convex Hull, Support Vector Machine, Multilayer Perceptron

---

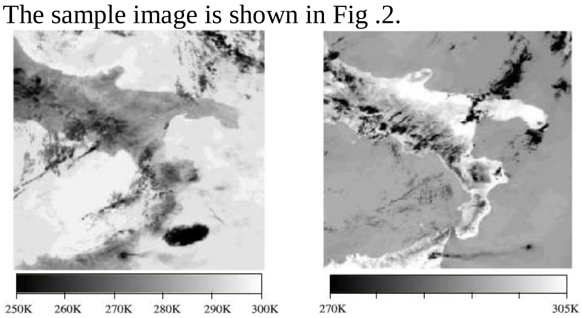
1. INTRODUCTION

In mid-2017, the National Space Organization (NSPO) successfully launched the completely self-developed satellite FORMOSAT-5, and obtained high-quality optical satellite images through self-made CMOS digital image sensing modules. This is a major milestone in the aerospace technology in Taiwan. The digital image obtained by FORMOSAT-5 can be widely used in land telemetry, disaster prevention, disaster relief, and commercial services. However, the satellite can provide limited power, so the satellite camera should avoid shooting the unwanted cloud-occluded images. To save the power of satellite camera, this paper propose a cloud detection method and a satellite attitude planning algorithm to help reduce the cloud appearance in satellite images.

In order to avoid consuming too much energy in the cloud detection, we use subsidiary low-resolution camera with lower energy consumption to perform cloud detection and decrease the shots in high-resolution camera. The method developed by this research can be divided into three parts, one is image processing, highlighting the possible areas of the cloud layer, followed by Grab cut and SVM for cloud layer detection, confirming the cloud layer area, and finally, importing machine learning and cloud weights. Through the cloud detection system we developed, we can find out the locations of the cloud layer and find a reasonable satellite attitude planning to obtain high-resolution satellite images in a more power-saving way. Fig. 1 shows the main flow of the proposed system.

2. RELATED WORK

In 2004, Pergola et al. proposed a method to detect volcanic ash clouds using thermal infrared information coming from AVHRR bands captured by satellite [1], and the sample images are shown in Fig. 1. In 2006, Lu et al. proposed a method to detect clouds and hazes in IKONOS image with 0.8-meter panchromatic and 4-meter multispectral bands [2]. The sample image is shown in Fig .2.



(a) (b)

Fig.1. Infared cloud image with pseudo colors in different range: (a) 250K~300K; (b) 270K~305K

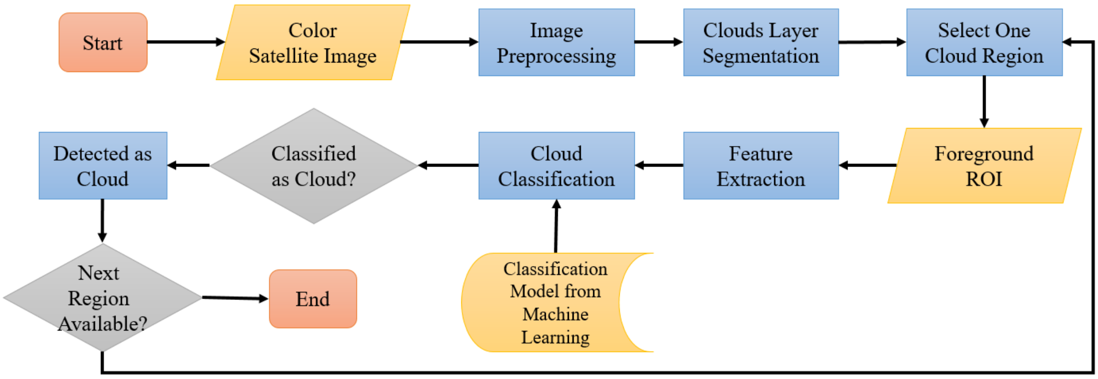


Fig.1. The flow chart of our proposed cloud detection method in satellite image

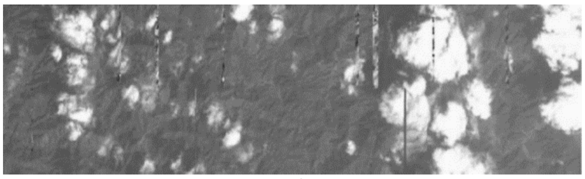
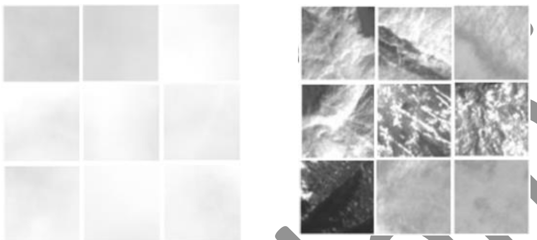


Fig.2. A sample image from IKONO satellite

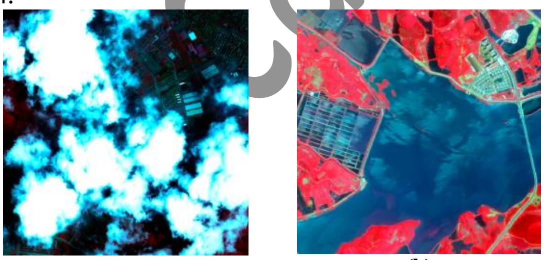
In 2015, Li et al. proposed a method to find thick clouds in satellite images of spectrum reflection [3]. They first divide the image into multiple small blocks, as shown in Fig. 3., and use multiple support vector machines (SVM) to detect clouds.



(a) **(b)**

Fig.3. Cloud and ground object samples: (a) clouds; (b) ground

In 2016, Bao et al. proposed a cloud detection method using the Gaofen datasets [4], which is composed of panchromatic image (one band), and multispectral image (four bands), as illustrated in Fig. 4.



(a) **(b)**

Fig.4. Scenarios for images chosen from Gaofen-1 dataset: (a) clouds; (b) water.

In this paper, we propose a method to detect clouds in RGB color image, and infrared, spectrum information are not required.

3. CLOUD DETECTION

This section proposes a method for cloud detection. The purpose of image pre-processing is to filter complex background images, leaving only the part of the cloud layer. Then, using image segmentation, contour extraction, and convex hull algorithm to obtain simplified cloud contours.

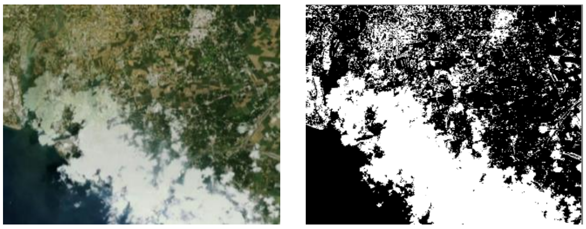
3.1. Image Pre-processing

The image pre-processing aims at filtering the clouds in the image. First, the image is binarized by the Otsu's method [5]. Since the cloud layer is usually a light-colored and high-brightness pixel, the Otsu algorithm can be used to filter out the pixels of the cloud layer. However, Otsu's method only uses brightness as the segmentation basis, in Figure 2, the white pixels is the cloud layer detected using Otsu's method, and some pixels are misjudged. Therefore, we also consider using thresholds in HSV color space instead of OTSU only. A flowchart of the proposed method is shown in Fig. 5.

Compared with the cloud layer obtained by OTSU, the result obtained by computing the HSV color space is better, as shown in Fig. 6. In addition, the HSV operation can be combined with the standard deviation to automatically select the HSV threshold. We have obtained ideal thresholds by experiment: H: 0~255, S: 0~50, V: 100~255. A result by applying the thresholds to a satellite image with cloud is shown in Fig. 7.

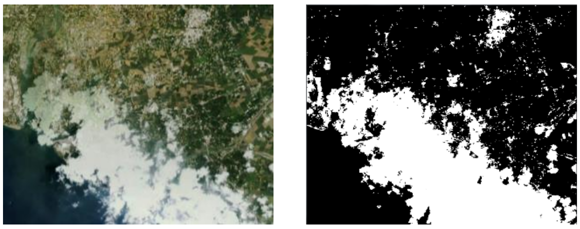
In addition to the HSV color space, we also conducted experiments in the RGB color space [6]. We found that the optimal thresholds are Red: 120~255, Green: 120~255, Blue: 120~255, and the segmentation result is shown in Fig. 8.

In the above results, there are much noises included. To remove the noise, the mathematical morphological methods including erosion and dilation operation are applied. In an erosion operation, if a target is too small, it will be eroded. To remove the noise and keep the shape of the detected cloud, we use an erosion operation followed by a dilation operation, and the result is as shown in Fig. 9, where the structural element used in the operations is a 3 x 3 square.



(a) (b)

Fig.6. The preprocessing result using Otsu's method: (a) original image; (b) processed binarized image with white pixels above the threshold computed by Otsu's method

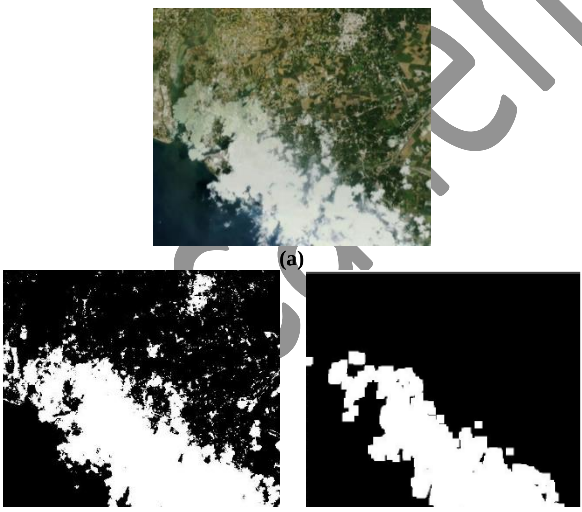


(a) (b)

Fig.7. Preprocessing by setting thresholds in channels of HSV color space: (a) original image; (b) processed binarized image with white pixels above thresholds in HSV color space

(a) (b)

Fig.8. Preprocessing by setting thresholds in the RGB color space: (a) original image; (b) processed binarized image with white pixels above thresholds



(b) (c)

Fig.9. Noise removal using opening operation: (a) original image; (b) binarization using HSV threshold; (c) after using opening operation

3.2. Clouds Layer Segmentation

A natural image has multiple Gaussian components. To find the foreground, we use the Grab cut method [7] to find the appropriate cutting points that separate the Gaussian components belonging to the foreground and background to segment the foreground of the image.

For advanced image segmentation, we apply the Gaussian mixture model (GMM) theory. In the GMM theory, there are multiple Gaussian components in a natural color image that some belong to the foreground while others belong to the background. To differentiate the two parts, the energy minimization is applied to find an optimal cut for these components.

The image consists of pixels z_n in RGB color space with n pixels. We denote π as the weights, μ as the mean value, and Σ as the covariance. Vector $\boldsymbol{k} = \{k_1, k_2, \dots, k_n, \dots, k_N\}$, with $\mathrm{k_n} \in \{1,2,\dots,N\}$, typically

$$\mathrm{E}(\alpha, \boldsymbol{k}, \theta, \boldsymbol{z}) = U(\alpha, k, \theta, z) + V(\alpha, z) \quad (1)$$

where

$$U(\alpha, \boldsymbol{k}, \theta, \boldsymbol{z}) = \sum_n -\log \pi(\alpha_n, k_n) + \frac{1}{2}\log \det \Sigma(\alpha_n, k_n) + \frac{1}{2}[z_n - \mu(\alpha_n, k_n)]^{\mathrm{T}} \Sigma(\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)] \quad (2)$$

The parameters are $\theta = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0,1, k = 1, \dots, K\}$ with 2K Gaussian components for the background and foreground distributions

From the definitions above, the flow of the Grab Cut algorithm is formulated as follows:

Algorithm 2: FP-growth

1 Initialize a trimap T with only T_B, and the foreground is set to $T_F = \emptyset$. Set $T_U = \overline{T_B}$, which is the complement of the background.

2 For $n \in T_B$, set $\alpha_n = 0$; for $n \in T_U$, set $\alpha_n = 1$

3 For each n in T_U, assign GMM components to pixels. $k_n \leftarrow \operatorname*{argmin}_{k_n} D_n(\alpha_n, k_n, \theta, z_n)$

4 Learn GMM parameters from data z $\theta = \operatorname*{argmin}_{\theta} U(\alpha, \boldsymbol{k}, \theta, \boldsymbol{z})$

5 Use minimum cut to solve:

$$\min_{\{\alpha_n : n \in T_U\}} \min_{k} E(\alpha, \boldsymbol{k}, \theta, \boldsymbol{z})$$

6 Repeat the process until convergence.

Fig.10. Flowchart of Grab Cut

In the graph theory, the optimal cut is the minimum cut in the component graph, and it is equivalent to the maximum flow. The Graph Cut is a method to find the minimum cut, which is also a useful method for image segmentation. An illustration of the Graph Cut concept is shown in Fig. 11.

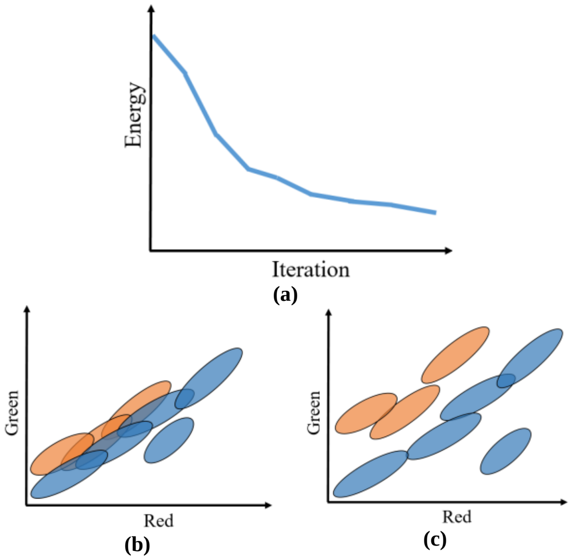


Fig.11. Energy minimization of an image by Grab Cut, each elliptic object represent a Gaussian component: (a) the relation between energy and iteration; (b) before energy minimization; (c) after energy minimization

However, sometimes it is unable to find the optimal cut by a single operation. Therefore, the iterative Graph Cut, called Grab Cut, is a method to perform energy minimization iteratively. Grab Cut segments the foreground and background of the satellite image through multiple iterations. In each iteration, it optimizes the segmentation from its last iteration. The optimization is performed by decrementing the energy of the objective function of GMM. When there is no obvious difference between two iterations, the result is converged, and an optimized segmentation is obtained. An optimized segmentation result from Grab Cut is shown in Fig. 12.

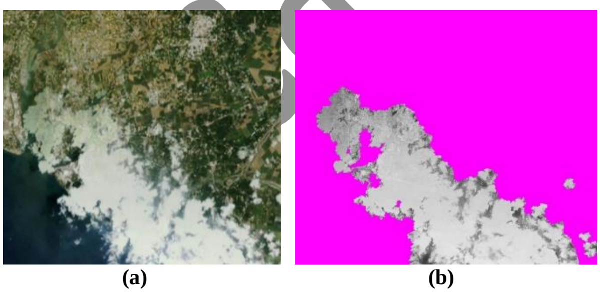


Fig.12. Cloud region segmentation by Grab Cut: (a) original image; (b) result image;

Grab Cut performs foreground segmentation on a pixel basis rather than setting thresholds on the entire image. As a result, there are less noise pixels in the Grab Cut segmentation result compared to the RGB and HSV segmentation methods. The comparison is shown in Fig. 13.

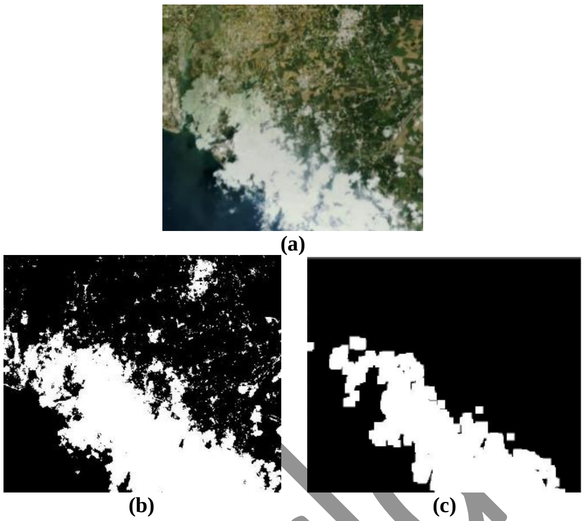


Fig.13. Comparison of cloud segmentation; (a) original image; (b) binarization using RGB thresholds; (c) binarization using Grab Cut

3.3. Clouds Contours Extraction

After the cloud layer is segmented, examine each separated cloud region to determine whether it is actually cloud or not. To separate every region, the contour analysis is conducted on the whole segmented cloud layer. A cloud region is a connected component with no pixels adjacent to other regions in the binarized image obtained from previous steps. The connected components are extracted in the binary image through the border following method proposed by Suzuki et al. [8]. The connected components are called as contours, and are shown in Fig. 14.

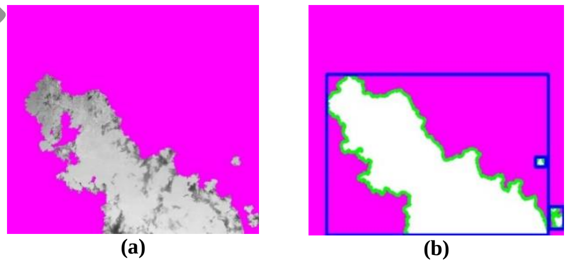
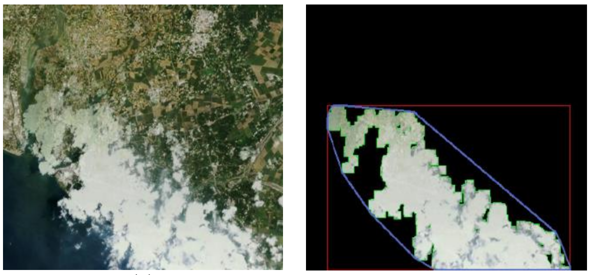


Fig.14. Finding the connected component of a cloud through border following; (a) a segmented cloud image using Grab Cut; (b) borders obtained by border following method marked in green, and the minimal bounding boxes of the borders are marked in blue

The border following method often extract cloud regions with coarse and complex edges, which is not suitable for contour analysis. Therefore, use the convex hull algorithm to simplify the contours of the cloud. First, use the border following algorithm to find the coarse edge of the cloud, and use the edge pixels as input for the convex hull algorithm. The convex hull algorithm finds the smallest polygon that frames the contours, and a sample result is shown in Fig. 15(b).

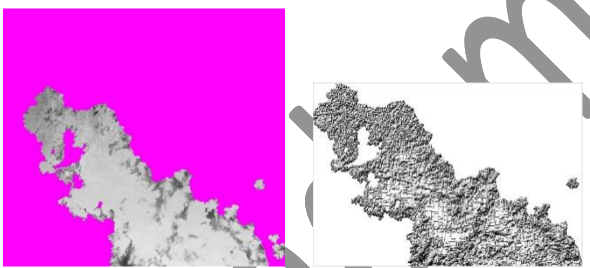


(a) (b)

Fig.15. Simplify the cloud contour using convex hull: (a) original image; (b) cloud with simplified contour marked in blue

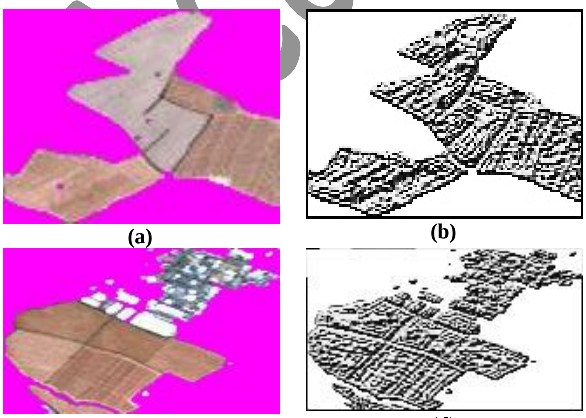
3.4. Clouds Texture Analysis

After extracting each separated cloud regions, use the local binary patterns (LBP) method [9] to analyze the texture of a region. The LBP is performed as follows: choose a pixel as the center point, and compare its brightness value with its 8 neighboring pixels. For the 8 pixels, if its value is greater than the center point, set its value to 1, otherwise set to 0; second, the form the 8 pixels into a binary sequence, and convert it to a decimal digit as the new value of the center point. There are two advantages of this method: (1) illumination changes will not affect the results of LBP; (2) calculation is simple and fast, and can be used on real-time systems. Fig. 16 shows the texture extracted by LBP, Fig. 17 shows the feature extraction results of two non-cloud samples, and Fig. 18 shows the feature extraction results of three cloud samples.



(a) (b)

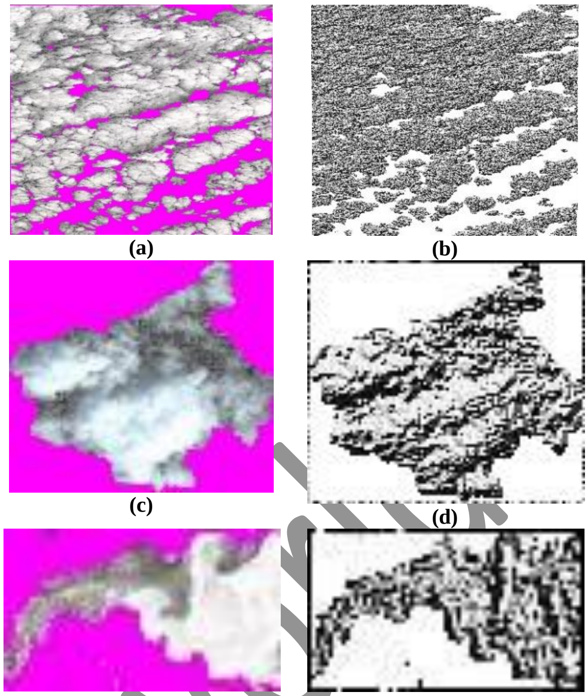
Fig.16. Using LBP to extract the texture feature of the cloud: (a) target cloud region; (b) texture feature by applying LBP



(a) (b)

(c) (d)

Fig.17. Some examples of non-cloud texture extraction: (a), (c) are the cloud region samples; (b), (d) are the extracted texture features



(a) (b)

(c) (d)

(e) (f)

Fig.18. Some examples of cloud texture extraction: (a), (c), (e) are the cloud region samples; (b), (d), (f) are the extracted texture features of the clouds

3.5. Clouds Layer Classification

After the cloud image is extracted by the above features, we use the statistics of the Histogram of oriented gradients (HOG) [10] to numericalize the features. We use the Support Vector Machine (SVM) [11] to build a classification model for clouds; SVM finds a maximum margin between the cloud data points and non-cloud data points in the feature space. However, the training data may present a non-linear distribution in the feature space, which is linearly inseparable and is impossible to divide by a hyper-plane from SVM. Therefore, we choose linear core transfer function and RBF function to map the data points to a higher dimension feature space as a solution [12], and the implementation is shown in Fig. 19.

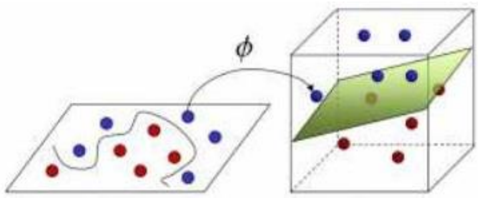


Fig.19. Feature space mapping example: original data are into the feature space of higher dimension

The experimental result for the building the classifier is shown in Table 1. According to the results, the best combination of building a cloud classifier is the using HoG feature with linear core transfer function in SVM. The accuracy of cloud classification is 99.8% with the best combination.

Table.1. The Accuracy of the Cloud Classifier Using SVM

Feature / Kernel	Gray Histogram	HoG
Linear	645/749 = 86.1%	748/749 = 99.8%
RBF	674/749 = 90.0%	692/749 = 92.3%

4. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method for cloud detection in satellite images. First, we use thresholds from Otsu's method, RGB and HSV color space. Subsequently, Grab Cut, a GMM based method is used to separate foreground and background Gaussian components. Also, LBP is used to extract texture features. To build a cloud classifier, we use the LBP combined with SVM with linear core transfer functions to establish a classifier which 99.8% of cloud classification accuracy. However, according to our statistics on 13,037 cloud image datasets, the Grab Cut algorithm currently used has an average execution time of 27.6 seconds in a single cloud image. Therefore, the detection speed should be reduced for real-time applications. In the future, we will introduce the object recognition method based on deep learning that does not require preprocessing steps, and has the capability of generating feature and locating the clouds automatically.

ACKNOWLEDGMENTS

The authors thank the National Space Organization of National Applied Research Laboratories of Taiwan (R. O. C.) for supporting this work in part under Grant NSPO-S-107100.

REFERENCES

1. N. Pergola et al., "Improving volcanic ash cloud detection by a robust satellite technique," *Remote Sensing of Environment*, vol. 90, no. 1, pp. 1-22, 2004.
2. D. Lu, "Detection and substitution of clouds/hazes and their cast shadows on IKONOS images," *International Journal of Remote Sensing*, vol. 28, no. 18, pp. 4027-4035, 2007.
3. P. Li, L. Dong, H. Xiao, and M. Xu, "A cloud image detection method based on SVM vector machine," *Neurocomputing*, vol. 169, no. 2, pp. 34-42, 2015.
4. T. Bai et al., "Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion," *Remote Sensing*, vol. 8, no. 9, pp. 715-721, 2016.
5. N. Otsu, "A Threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.
6. Q. Zhang and C. Xiao, "Cloud detection of RGB color aerial photographs by progressive refinement scheme," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7264-7275, 2014.
7. C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *Association for Computing Machinery Transactions on Graphics*, vol. 23, no. 3, pp. 309-314, 2004.
8. S. Suzuki, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32-46, 1985.
9. T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.
10. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, pp. 886-893, 2005.
11. C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
12. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 9, pp. 533-536, 1986.