Research on Cloud Cover Monitoring Strategies for Radio Astronomy Observatory Sites (Postprint)

Authors: Zhang Hailong, Lu Dengrong, Sun Jixian, Li Jibin, Zhang Xuguo

Date: 2025-10-11T00:00:00+00:00

Abstract

In the site selection process for millimeter-wave and submillimeter-wave radio telescopes, designing an all-sky camera system applicable to field environments is essential in order to fully understand the cloud cover information of candidate astronomical sites. Therefore, according to the characteristics of radio telescopes and the specific conditions of field sites, the scheme innovatively utilizes a planetary camera and an embedded microcontroller to develop a full-time all-sky camera. It can operate long-term in the field using solar power, and its most important feature is the ability to achieve unattended, autonomous operation. In the data processing section, deep learning neural network algorithms are also innovatively employed to extract data feature values, establish a machine learning model library, and automatically calculate statistics for the cloud cover information of the site, which is more efficient and simpler than manual and general image processing algorithms. These studies provide important references for more comprehensively evaluating millimeter-wave and submillimeter-wave radio astronomical sites.

Full Text

Preamble

Vol. 66 No. 5

September 2025

Acta Astronomica Sinica

Research on Cloud Monitoring Scheme for Radio Telescope Observatory Sites

ZHANG Hai-long†, LU Deng-rong‡, SUN Ji-xian, LI Ji-bin, ZHANG Xu-guo
(Qinghai Station of Purple Mountain Observatory, Chinese Academy of Sciences,
Delingha 817000)

Abstract

In the process of selecting sites for millimeter and submillimeter-wave radio astronomical telescopes, it is essential to design an all-sky camera system for field environments to fully understand cloud cover information at candidate observatory sites. Therefore, according to the characteristics of radio telescopes and the specific conditions of field sites, this scheme innovatively utilizes a planetary camera and embedded microcontroller to develop a full-time all-sky camera. The system can operate long-term in the field using solar energy, with the most important feature being its capability for unmanned, autonomous operation. In the data processing phase, the scheme also innovatively employs deep learning neural network algorithms to extract data feature values, establish a machine learning model library, and automatically generate statistics on cloud cover information at the site, achieving higher efficiency and simplicity compared to manual and general image processing algorithms. These studies provide important references for more comprehensive evaluation of millimeter and submillimeter-wave radio observatory sites.

Keywords telescopes: radio, instrumentation: detectors, methods: measurement and evaluation, techniques: image processing, site testing

1 Introduction

Millimeter and submillimeter-wave radio astronomical telescopes have stringent requirements for observatory sites, particularly submillimeter telescopes. In addition to demanding conditions such as low temperature, low atmospheric water vapor content, and large-scale radio-quiet zones, these sites must also meet certain requirements regarding altitude, topography, atmospheric effects, and cloud cover. Submillimeter-wave radio telescopes require sites with minimal cloud cover because cloud amount indirectly reflects atmospheric water vapor content. Consequently, cloud cover significantly impacts the quality of data produced by millimeter-wave radio telescopes, potentially reducing observation efficiency and affecting telescope operations. Among site monitoring methods, using all-sky cameras to capture cloud images for monitoring sky cloud cover is one of the most common approaches. To understand cloud cover information at candidate field sites, the system design draws on the all-sky cloud monitoring scheme from China' s SONG (Stellar Observations Network Group) project and other optical observatory all-sky camera development plans. To achieve automated all-sky monitoring at the site, the scheme employs a ZWO ASI224MC color planetary camera with 1280$×$960 pixels and 3.75 m pixel size, providing extremely low read noise and high sensitivity. This camera is much smaller than typical DSLR cameras, facilitating integration, and has low power consumption (maximum 2.8 W), allowing direct power supply and control via USB interface. To monitor cloud cover across all sky regions, the camera is paired with a 5-megapixel fisheye lens with a 1.55 mm focal length. For efficient processing of large data volumes (at least hundreds of cloud images daily), the scheme uses Python-based deep learning algorithms that train on extensive samples to extract features from various cloud images and establish a model library for fully automatic weather condition statistics.

2.1 Monitoring Requirements

Monitoring at potential astronomical candidate sites has the fundamental requirement of unmanned operation to obtain long-term data. Particularly when selecting sites for millimeter and submillimeter-wave radio telescopes, candidate locations are typically in high-altitude, hypoxic, uninhabited wilderness areas without any support infrastructure. This necessitates a fully automated monitoring system capable of automatic monitoring, automatic camera parameter adjustment based on light conditions, automatic data saving, and later automatic data transmission.

The all-sky camera monitoring system must continuously monitor sky cloud cover changes and save data locally. In environments with network connectivity, the system supports remote login to display cloud cover data. Cloud cover data can also be saved locally in image format according to sampling intervals for subsequent processing. The sampling interval is configurable; based on comprehensive analysis of image size, storage capacity, and cloud change timescales, the scheme selects a 5-minute interval for saving cloud images. Due to extended unmanned operation, high reliability is required, and with candidate site temperatures reaching as low as -35°C, the system must guarantee normal operation in low-temperature environments. The monitoring control system must be integrated, lightweight, and low-power to facilitate field installation, operation, and maintenance.

2.2 System Composition

The all-sky camera monitoring system consists of an embedded microcontroller (Raspberry Pi 4B), an astronomical camera (ASI120MM-S from ZWO), a heating control module, a fisheye lens (FS15520FEMP), an acrylic hemisphere, an enclosure, and DC power supply (including solar panels, batteries, and charge controller), as shown in [FIGURE:1].

The entire all-sky camera enclosure is wrapped in acrylic panels, integrating the embedded microcontroller and planetary camera inside. The embedded controller uses a Raspberry Pi microcontroller powered by 5V DC with maximum power consumption of 6-7W, running a Linux operating system with 128GB storage space. The planetary camera is powered and communicates via the Raspberry Pi’ s USB port, with maximum power consumption of 2.86W.

Considering factors such as camera and lens protection, light transmission, fabrication, installation, and overall weight, the all-sky camera enclosure structure entirely adopts acrylic material (plexiglass). Acrylic panels offer excellent light transmission (over 92%), superior weather resistance, and strong adaptability to natural environments, maintaining performance without degradation under long-term sunlight exposure, wind, and rain. For fabrication, acrylic is suitable for both mechanical processing and thermoforming, and can be easily cut. Both acrylic hemispheres and enclosures are available as off-the-shelf products requiring no complex processing, while other support panels can be cut from acrylic sheets according to installation requirements. For field installation and operation, the structural design principle for the all-sky camera is to achieve miniaturization, integration, and lightweight design, with the 3D model shown in [FIGURE:2].

During winter operation, significant temperature differences between the inside and outside of the acrylic enclosure can cause frost formation on the inner wall of the hemisphere. Therefore, gaps are left at the bottom contact surface during hemisphere installation, and ventilation holes are opened around the perimeter to facilitate temperature balance and eliminate frost formation on the inner hemisphere wall, as shown in [FIGURE:3].

To improve thermal insulation, the entire acrylic enclosure is also wrapped with thermal insulation cotton to reduce heat loss, with the actual device shown in [FIGURE:5]. The camera's minimum operating temperature is -5°C, but considering that winter temperatures at field sites can reach -35°C, flexible polyimide (PI) film heaters are installed on the camera, powered by 12V DC for heating and insulation. Through heater operation and the Raspberry Pi controller' s self-heating, the camera temperature is maintained above -5°C. Temperature control uses a simple temperature switch chip installed near the camera, which controls heating based on camera temperature: below 0°C the switch closes to power the heater, while above 0°C the switch remains open to stop heating, as shown in [FIGURE:4].

3.1 All-Sky Camera Installation and Monitoring

The all-sky camera was initially tested from early December 2023 to mid-February 2024, installed at the Qinghai Observatory Station of Purple Mountain Observatory. In March 2024, it was installed at the Delingha Snow Mountain Ranch radio multi-band candidate site of Purple Mountain Observatory for long-term monitoring, as shown in [FIGURE:6].

The camera control program is written in Python, calling the camera' s built-in function library to enable automatic photography at set intervals and save images. The program can also control camera exposure time and gain parameters in real-time based on sky brightness changes. Common algorithms for camera gain and exposure control include average brightness method, weighted mean method, and brightness histogram, with the average brightness method being the most widely used and implemented in this scheme. The specific control process is shown in [FIGURE:7].

The control program calculates the mean brightness of each captured image using the median function from Python' s numpy library. Based on measured results, a brightness range is established: values below the minimum are considered too dark, requiring increased exposure time and gain, while values above the maximum are considered too bright, requiring decreased gain and exposure time. New gain and exposure parameters are then set, and the adjusted image is acquired for another round of mean brightness calculation and judgment. After multiple iterations, the image mean brightness falls within the established range.

The brightness range must be determined based on actual site conditions, which can be assisted by long-term measurements using sky brightness sensors, night sky brightness monitors, and other instruments to calculate setting parameters. In this scheme, based on specific conditions such as presence or absence of sun and moon, clear skies, cloudy conditions, and various weather scenarios, optimal empirical brightness ranges were obtained through multiple parameter adjustments. After determining the brightness adjustment range, images were captured. [FIGURE:8] shows cloud images taken during daytime with sun, clear sky, cloudy, and overcast conditions. [FIGURE:9] shows images captured at night with moon, clear sky, cloudy, and overcast conditions.

During the testing phase, it was discovered that direct sunlight would reflect the camera' s bottom image onto the top hemisphere, as shown in the top-left photo of [FIGURE:8]. Later considerations included laying black anti-reflection cloth at the bottom of the camera lens to eliminate this reflection. In traditional cloud image processing, the grayscale values of sun and moon regions are often similar to cloud features, causing data processing algorithms to mistakenly identify these regions as clouds and produce false positives. Therefore, image data requires preprocessing. Solutions to this problem include time segmentation method and differential method, with the basic idea being that in adjacent images, the position changes of the sun and moon are smaller than cloud changes, allowing removal via differential method. The time segmentation method identifies and removes sun and moon trajectories by recognizing their relatively constant grayscale values across multiple consecutive frames. While these methods can effectively eliminate sun and moon effects, they complicate data processing. For applications that do not require precise analysis of cloud size and coverage, deep learning neural network algorithms offer greater advantages, as discussed below.

3.2 Data Processing and Statistical Analysis

All-sky camera data is saved in image format, and the data volume is extremely large, requiring efficient algorithms for processing and statistical analysis. Cloud image analysis has traditionally relied on manual statistics, with limited algorithmic processing that generally requires specialized image processing knowledge and yields unsatisfactory efficiency. Therefore, this scheme innovatively employs image recognition technology from the field of artificial intelligence. The basic approach involves first extracting image features, then establishing a model library through machine learning, and finally classifying and performing statistical analysis on target images based on this foundation. The traditional image recognition process is shown in [FIGURE:10].

With the development of AI technology, image recognition has become an important field of artificial intelligence, with increasingly stringent requirements revealing many shortcomings in traditional image recognition, such as weak adaptability, susceptibility to noise interference, and dependence on manual design. Particularly in complex application environments like autonomous driving, computers must quickly identify various objects. Deep learning is highly suitable for such scenarios, as extensive sample training and long training periods significantly improve image recognition accuracy and efficiency.

Deep learning research can be traced back to the M-P neuron model proposed by American psychologist McCulloch and mathematician Pitts in 1943 during their investigation of artificial neural networks. Deep learning mainly includes algorithms such as Deep Belief Network (DBN), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN). A typical neural network structure is shown in [FIGURE:11], including an input layer, several hidden layers, and an output layer. Hidden layers contain numerous neurons with corresponding weight connections and activation functions. The number of neurons and hidden layers indicates network complexity, with more numerous networks demonstrating stronger adaptability and more significant nonlinear effects.

CNN structure is similar to traditional neural networks, also designed based on brain neurons. Its essence is to connect simple neurons that extract image information at the 底层, then combine many such neurons to extract highly complex image information, ultimately obtaining high-level semantic features of images. According to relevant experimental results, training models designed with CNN can achieve 96% accuracy on datasets, providing fundamental support for advancing image recognition technology.

The data processing workflow is shown in [FIGURE:12]. The scheme first uses CNN to generate a machine learning model library, dividing extensive sample cloud images into four categories (represented by numbers: 0: overcast; 1: cloudy; 2: sunny; 3: snow). The data processing program converts all cloud images to grayscale, obtains pixel matrices, and uses deep learning CNN tools for feature extraction to obtain feature matrices W for the four sample categories of overcast, sunny, cloudy, and snow. After extensive sample training through multiple iterations, the training model parameter library is finally obtained. The cloud image recognition process also converts cloud images to grayscale, then calls the training model library, extracts cloud image feature values through CNN, and finally classifies them into the predefined four categories based on feature values to determine weather conditions.

CNN feature extraction is a crucial component in both training model library generation and cloud image recognition, forming the core of image recognition. Its basic structure is shown in [FIGURE:13].

As shown in [FIGURE:13], the CNN image feature extraction employs multiple classic network layers including Convolutional layers, Max pooling layers, ReLU (Rectified Linear Unit) activation layers, and Fully connected layers. The convolutional layer functions to extract image features and constitutes the core of CNN. Pixel matrix data from images undergoes convolution operations with convolutional kernels containing weight parameters, moving left-to-right and top-to-bottom through the image according to equation (1) to obtain the image feature matrix, as shown in [FIGURE:14].

In equation (1), the leftmost matrix represents the image pixel matrix denoted as a, while the convolution kernel' s weight values represent the final result of image recognition, namely the model library. Initial values are random, and after training iterations, the machine learning model library is ultimately obtained.

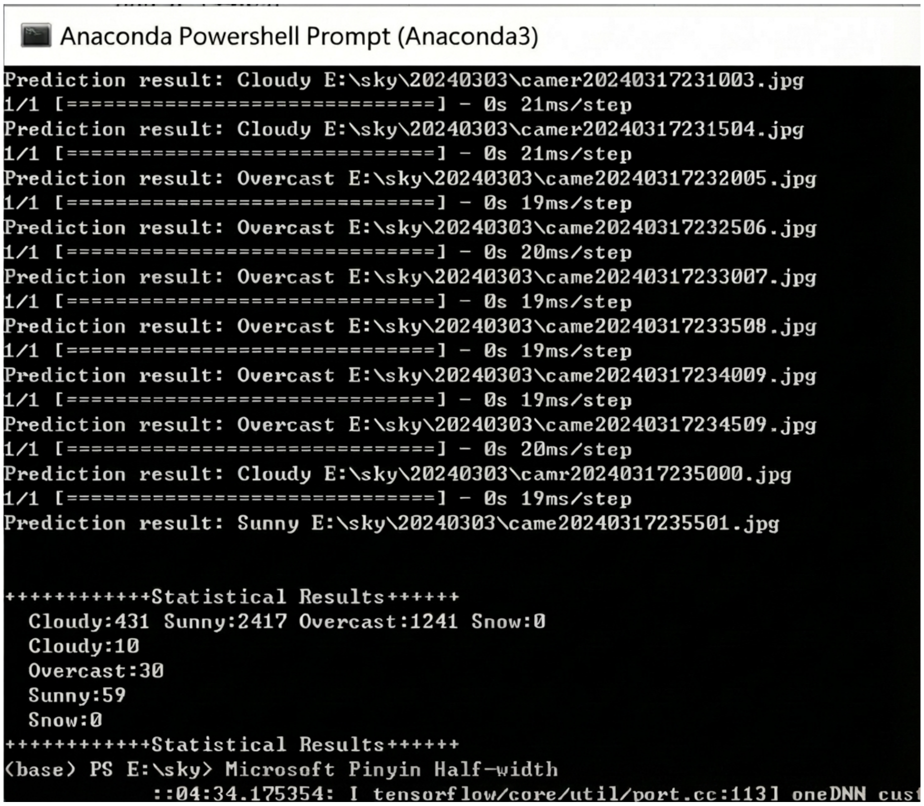
The pooling layer' s primary function is to select features extracted by convolutional layers, choosing image information 不受 position interference. Secondly, it reduces feature dimensions and decreases the number of feature variables, thereby reducing computational load. Common pooling operations include max pooling and mean pooling. [FIGURE:15] divides the extracted feature matrix into regions according to bold boxes, then extracts the maximum value from each region to form a new feature matrix—this is the max pooling operation.

In neurons, input undergoes a series of weighted summations before acting on another function called the activation function. Similar to neuron-based models in the human brain, the activation function ultimately determines whether to transmit signals and what content to transmit to the next neuron. Common activation functions include Sigmoid, Tanh, and ReLU functions. The first two are prone to overfitting during operation, so the ReLU function is generally used, with the form:

The fully connected layer serves a classification function in neural network models as the final layer. During training model library generation, training efficiency can be improved by optimizing parameters such as the number of CNN layers, number of convolutional kernels, whether to perform padding operations, the Dropout random deactivation ratio for neural network model regularization, and the number of data traversal epochs. Among these, the number of convolutional layers and traversal epochs have greater impact.

[FIGURE:16] shows the influence of CNN layer numbers on training effectiveness, where Train_{loss} and Train_{acc} represent neural network loss function values and accuracy, respectively. It can be observed that with the same sample data volume, more convolutional layers increase the risk of loss function jumps and overfitting. [FIGURE:17] shows training results achieved with Epochs set at 10, 30, 60, and 100 iterations, demonstrating that accuracy exceeds 90% when data traverses 30 times. Excessive traversal makes loss function values less smooth, risks overfitting, and increases computer processing time.

From March 4 to 17, 2024, 4,089 data groups were measured at the candidate site, and weather statistics were obtained through the training model. [FIGURE:18] and show model training and weather data statistics, respectively.



```
Prediction result: Cloudy E:\sky\20240303\camer20240317231003.jpg
1/1 [==============================] - 0s 21ms/step
Prediction result: Cloudy E:\sky\20240303\camer20240317231504.jpg
1/1 [==============================] - 0s 21ms/step
Prediction result: Overcast E:\sky\20240303\came20240317232005.jpg
1/1 [==============================] - 0s 19ms/step
Prediction result: Overcast E:\sky\20240303\came20240317232506.jpg
1/1 [==============================] - 0s 20ms/step
Prediction result: Overcast E:\sky\20240303\came20240317233007.jpg
1/1 [==============================] - 0s 19ms/step
Prediction result: Overcast E:\sky\20240303\came20240317233508.jpg
1/1 [==============================] - 0s 19ms/step
Prediction result: Overcast E:\sky\20240303\came20240317234009.jpg
1/1 [==============================] - 0s 19ms/step
Prediction result: Overcast E:\sky\20240303\came20240317234509.jpg
1/1 [==============================] - 0s 20ms/step
Prediction result: Cloudy E:\sky\20240303\camr20240317235000.jpg
1/1 [==============================] - 0s 19ms/step
Prediction result: Sunny E:\sky\20240303\came20240317235501.jpg


++++++++++++Statistical Results++++++
  Cloudy:431 Sunny:2417 Overcast:1241 Snow:0
  Cloudy:10
  Overcast:30
  Sunny:59
  Snow:0
++++++++++++Statistical Results++++++
(base) PS E:\sky> Microsoft Pinyin Half-width
            ::04:34.175354: I tensorflow/core/util/port.cc:113] oneDNN cus
```

Figure 1: Figure 19

After obtaining the training model, to evaluate its quality, four groups of real sample sets (gold standard) were extracted from the sample data, including overcast, cloudy, sunny, and snow conditions, which were then analyzed using the model. The open-source artificial neural network library keras written in Python was used, employing keras's built-in model evaluation function Evaluate and the confusion matrix commonly used in evaluation classification algorithms.

Analysis results show: the built-in evaluate function assesses model accuracy at 94%, while the confusion matrix evaluates accuracy for each classification set at 93.7%, as shown in [FIGURE:20]. [FIGURE:21] compares manual statistics and neural network algorithm statistics results, with the maximum difference occurring between cloudy and overcast classifications, primarily because sample sizes were small during model training, with each sample set containing fewer than 100 data groups. However, in processing time, the machine learning algorithm is much more efficient than manual processing. For the aforementioned 4,089 data groups, the algorithm can complete statistics in approximately 1 minute, while manual processing would take at least 1 hour. This demonstrates that machine learning algorithms are more efficient for large-scale batch processing.

Compared with other algorithms such as grayscale value aggregation algorithms, CNN demonstrates advantages in complexity, efficiency, and accuracy for specific applications that do not require precise cloud amount calculation. The comparison results are shown in . Of course, for applications requiring precise cloud amount calculation, neural networks would need more refined classification and larger training samples, possibly even requiring complex image processing algorithms. However, for radio telescope site selection, precise cloud amount data is not necessary—only long-term statistical data reflecting the general atmospheric conditions above the site is needed, along with other environmental data to collectively reflect site weather conditions. It is precisely for this reason that CNN algorithms offer significant advantages.

4 Summary

All-sky camera monitoring systems play an important role in real-time cloud cover monitoring and constitute an important component of observatory site monitoring systems for both optical and radio telescopes. This system is designed specifically for the main characteristics of millimeter and submillimeter-wave radio telescope site selection, where candidate sites are typically in high-altitude, harsh environments, uninhabited wilderness areas without any support infrastructure. The design scheme must meet requirements for simple power supply (solar-powered), simple installation, and unmanned automatic operation, with future capabilities for remote monitoring via Beidou short-message communication. Addressing these special requirements, the hardware aspect innovatively selects cost-effective planetary cameras to capture sky images. By utilizing low-power, miniaturized embedded controllers that automatically adjust camera parameters based on actual sky brightness, the monitoring system achieves unmanned operation.

In data processing, the scheme pioneeringly introduces deep learning CNN algorithms, with deep learning algorithm software written in Python. Through extensive sample machine training, feature values from various cloud images are extracted, enabling effective processing of massive data (at the level of tens of thousands) and achieving batch data processing with fully automatic weather condition statistics for the site.

Future research on data processing algorithms can mainly include: image preprocessing, eliminating false positives from sun and moon regions, and removing reflections; researching all-sky cloud amount definition and calculation under

CNN algorithms; increasing neural network model training sample data to improve model reliability; and conducting remote data transmission experiments to transmit processed data results via Beidou short-message communication, achieving remote monitoring of weather conditions at field sites.

References

1. Zhang Hailong, Yao Qijun, Sun Jixian, et al. Qinghai Science and Technology, 2023, 3: 23
2. Peng Huanwen, Xin Yuxin, He Shousheng. Astronomical Research and Technology, 2015, 12: 89
3. Tian Jianfeng, Deng Licai, Yan Zhengzhou. Acta Astronomica Sinica, 2016, 57: 366
4. Fan Liang, Lei Chengming. Acta Astronomica Sinica, 2016, 57: 461
5. Wei Shiya, Qiu Bo, Cao Zihuang. Astronomical Research and Technology, 2019, 16: 85
6. Zhang Yuhong, Bai Renxiang, Meng Fanjun. New Technology and New Process, 2021, 1: 52
7. Zhou Feiyan, Jin Linpeng, Dong Jun. Chinese Journal of Computers, 2017, 40: 1229
8. Guo Lili, Ding Shifei. Computer Science, 2015, 42: 28
9. Liu Fangyuan, Wang Shuihua, Zhang Yudong. Computer Engineering and Applications, 2018, 54: 11
10. Zheng Yin, Chen Quanqi, Zhang Yujin. Journal of Image and Graphics, 2014, 19: 175
11. Cong Shuang, Dai Yi. Computer Applications, 2004, 24: 18
12. Zhang Ronglei, Tian Aikui, Tan Hao. Journal of Shandong University of Technology (Natural Science Edition), 2018, 32: 48
13. Luo Jiajie, Shi Jialin. Science and Technology Innovation Herald, 2019, 24: 130
14. Xu Feng, Lu Jiangang, Sun Youxian. Information and Control, 2003, 32: 344
15. Chang Liang, Deng Xiaoming, Zhou Mingquan. Acta Automatica Sinica, 2016, 42: 1300