Sky Pixel (Region) Detection using Computer Vision Algorithm

Landy Ko Zi Ying
School of Science and Technology
Sunway University
Selangor, Malaysia
20043923@imail.sunway.edu.my

Abstract — This paper proposes a computer vision algorithm that can automatically identify pixels that belong to the sky region using Python and OpenCV. The algorithm processes images from different folders (623, 684, 9730, 10917), each containing images taken during either daytime or nighttime from different cameras at different locations. The sample dataset is taken from the SkyFinder dataset. The proposed algorithm starts with image preprocessing to classify each image as either daytime or nighttime based on its brightness level. Then, the sky region detection algorithm leverages Sobel edge detection and energy optimization techniques to identify the optimal sky border position. The algorithm then applies post-processing to refine the segmentation result, followed by skyline detection using edge detection on the post-processed mask. For nighttime images, the algorithm applies the best daytime mask to obtain the sky region. Lastly, the algorithm evaluates the accuracy and Mean Squared Error (MSE) of the detected sky region compared to the ground truth mask provided in the dataset. The output is visualized and saved to a result folder. Moreover, considering only daytime images, the overall average accuracy and MSE for all folders are computed. The execution time of the algorithm is also calculated to assess the effectiveness of the sky region detection algorithm.

Keywords—computer vision, Python, sky region detection, Sobel edge detection, skyline detection, accuracy, Mean Squared Error (MSE)

I. INTRODUCTION

A. Problem Definition

Sky region detection challenges vision algorithms to identify pixels belonging to the sky region accurately. Another problem is accurately detecting the sky region in images taken during either daytime or nighttime. This is challenging, especially for outdoor imagery affected by various conditions. For example, the existing algorithm might perform well in detecting a clear blue sky but not when affected by weather, season, and time. However, it is important to solve the problem as sky region detection is a preprocessing step for many important outdoor-based computer vision and image processing applications, including horizon estimation, robot navigation, image geolocalization, acquisition and understanding, weather forecasting, solar exposure prediction [1].

B. Describe the differences between the proposed algorithm and existing algorithms

The proposed algorithm distinguishes itself from existing techniques in several aspects. Firstly, it incorporates Sobel edge detection and energy optimization, offering a robust approach to identifying the sky region. A more detailed description will be presented in the literature review section.

Secondly, the proposed algorithm is implemented so that the sky region of the image is the region of interest, which is displayed in the resulting binary mask, where the sky region is labeled with 255. The non-sky region is labeled with 0. After that, the post-processing step is applied to the detected mask to improve the segmentation accuracy. The step includes the application of the morphological closing algorithms involving dilation and erosion to fill the gaps in edges after edge detection.

Lastly, most existing algorithms pose challenges in handling dark sky images. The proposed algorithm can indirectly handle dark sky, nighttime images. The algorithm first processes all daytime images to obtain the sky region mask and saves the output, while nighttime images are stored in an array list. After processing all daytime images, the algorithm identifies the best daytime mask with the highest accuracy. This mask is applied to the nighttime images to identify the sky region, ensuring consistency and accuracy across lighting conditions. By leveraging the best daytime mask for nighttime images, the algorithm effectively detects the sky region even in low-light or challenging weather conditions. Utilizing the best daytime mask guarantees consistently high accuracy, making the accuracy of nighttime images less relevant in this context. As the modification utilises the best daytime mask, thus the accuracy is always ensured to remain high and optimal. Hence, the accuracy of the nighttime image will not be considered here.

II. LITERATURE REVIEW

This section presents an overview of systems, methods, and algorithms proposed for sky region detection. By exploring the strengths and weaknesses of the existing approaches, the review contributes ideas to the proposed algorithms.

A. *Overview of Related System, Methods, and Algorithms*

One sky region detection method is defining the boundaries between sky and non-sky regions. The boundary is usually identified using **the edge detection technique**, which defines the most significant changes in intensity or color in an image. Examples of edge detection methods include Sobel, Roberts Cross, Prewitt, Laplace, and Canny edge detectors [2]. The most used methods are Canny and Sobel edge detection. Sravanthi and Sarma [3] utilised a conventional method for the sky segmentation problem using canny-based edge detection. Few studies implement the Sobel edge detection algorithm or Sobel operator in sky region detection [4,5].

Besides, **the morphology closing algorithm**, although not directly involved in sky region segmentation, can significantly improve the algorithm's accuracy in the post-processing step. Sravanthi and Sarma [3] and Laungrungthip et al. [6] applied the morphological operation in their proposed sky region detection algorithm. The method is used to fill the gaps in edges after edge detection. It is applied to the binary image representation of the edges. The algorithm consists of two operations: dilation and erosion. They are performed using a structuring element (SE).

As mentioned earlier, it is important to consider weather conditions affecting visibility, such as foggy days, when detecting the sky region. In such cases, a defogging approach can be applied before sky region detection. Song et. Al. [7] proposed a **hybrid image defogging approach** consisting of three phases for defogging both the sky and non-sky regions. In the first phase, the sky region is extracted and segmented from the hazy image using edge detection embedded with a confidence measurement strategy and the mean shift approach. After that, a morphological closing algorithm is applied to extract the sky area and obtain the binary image. In the second phase, a refined dark channel prior defogging algorithm is applied to remove fog from non-sky regions. This involves estimating atmospheric light and the transmission map of the hazy image, refining the transmission map using a guided filter, and restoring the image based on the transmission map. In the third phase, an improved DehazeNet defogging algorithm is applied to enhance the defogging approach. Notably, the defogged image may exhibit low contrast due to the removal of fog, necessitating image enhancement techniques like brightness adjustment to improve the overall brightness and appearance of the restored image.

K-means clustering is an image segmentation algorithm that can also be applied in sky region detection. The main idea of image segmentation is grouping pixels based on their similarities or characteristics. Nice et al. [8] conducted K-means clustering in sky region detection. The study highlights that a different number of clusters, K, affects the accuracy of image segmentation. The findings, already defined in the literature review before, could be a reference for determining K when developing sky region detection algorithms. Dhanachandra et al. [9] highlighted the importance of choosing the initial centroid, which will produce different results.

Horizon Detection Algorithm can also be used in sky region detection, where the horizon line e identifies the boundary between the sky and the rest of the images. A study proposes a horizon detection algorithm that applies edge-based and color-based methods for accurate horizon detection [10]. The edge-based detector uses Hough transformation and Canny edge detection to pinpoint the most prominent edge or line that best represents the horizon. When obstacles obscure the horizon, the color-based detector calculates the horizon line by examining colour transitions that correlate to a clear sky. To categorize pixels as the sky or non-sky regions, the algorithm builds a continuous sky probability map turned into a binary map.

B. *Summary of strengths and weaknesses of the algorithms*

The strengths and weaknesses of each algorithm are the key factors in determining their suitability in the context of sky region detection. **Canny Edge Detection** is accurate and can eliminate false positives, making it ideal for determining boundaries between the sky and non-sky regions. Its adaptability to recognizing complicated and irregular shapes, and ensuring exact border identification, is beneficial when dealing with outdoor scenes. However, it may demand more processing resources due to its multi-step procedure, making it less appropriate for real-time applications with a large dataset.

On the other hand, **Sobel Edge Detection** provides simplicity and computing efficiency in the context of the sky region. Its ability to provide edge direction information aids in determining edge orientation and slope, making it useful for distinguishing sky regions from other objects in the scene. Nonetheless, Sobel is susceptible to noise, leading to false positives and mistakes in sky region segmentation. Furthermore, its tendency to overlook weak or faint edges could impact segmentation accuracy in low-contrast photos or scenes with subtle transitions between sky and non-sky regions.

The **morphology closing algorithm** plays a crucial role in the post-processing step to improve the accuracy of the sky detection algorithm. The algorithm has the advantage of filling gaps by improving the detected objects' connectivity and completeness, enhancing the sky region's accuracy. However, its limitation in handling complicated or irregular shapes and the potential to overbalance smooth regions might affect segmentation accuracy, particularly in scenes with various or challenging sky structures.

The **hybrid image defogging approach** addresses the visibility problem brought on by haze in sky area detection. It enhances the clarity of the sky region by reducing haze, allowing for more precise segmentation. However, the output image might have less brightness and contrast, which could impact how it looks overall and make it more difficult to notice objects in other parts of the image.

K-means Clustering is a desirable option for image segmentation tasks like sky region detection because of its simplicity and scalability. However, the accuracy of segmentation results might be affected due to its sensitivity to the initial centroid selection and the assumption of equal-sized and spherical clusters.

In the **horizon detection algorithm**, both edge-based and color-based detector have their strengths and weaknesses. The edge-based detector relies on identifying boundaries and edges in the image, which can effectively capture the distinct edges between the sky and non-sky regions. However, it may need help with certain images needing clear edges. On the other hand, the color-based detector focuses on analysing the colour information in the image to differentiate the sky region from the rest. This approach can be helpful in scenarios where the colour of the sky exhibits consistent characteristics. However, it may face challenges in cases where there are variations in lighting conditions or when other objects in the scene share a similar colour with the sky. Combining both detectors can enhance the accuracy and reliability of the sky region detection results.

III. SYSTEM DESIGN / PROPOSED ALGORITHM

There are several key steps in this proposed algorithm to accurately detect the sky region in images during daytime and nighttime. The algorithm starts with identifying whether the image was daytime or nighttime using **is_daytime()** function. The function involved calculating the average brightness of the image and comparing it with a predefined threshold value. If the average brightness exceeds the threshold, the image is daytime and will proceed with sky region detection. Otherwise, it is considered nighttime and stored in an array list.

Next, **return_gradient_img()** function convert the image into a gradient image using Sobe operator. The Sobel operator is applied to the grayscale image, producing horizontal and vertical edges. These edges are combined to create a gradient image highlighting potential boundaries between sky and non-sky regions. The resulting gradient image is then passed to **return_border_position()**, called within **border_optimisation()**, to compute the optimal sky border position using energy optimization. Three parameters need to be defined: minimum threshold, maximum threshold, and sampling step. The number of sample points in the search space is calculated based on these parameters. Following Shen & Wang's study[4], threshold values above 600 result in nearly constant energy optimization function values. Hence, 600 is chosen as the maximum threshold and five as the minimum threshold, as the extrapolated graph starts from there. To strike a balance between search accuracy, program complexity, and time complexity, a moderate search step size of 5 is used.

After the search phase, the algorithm determines the number of sample points in the search space using the maximum and minimum threshold values. This information is crucial in finding the ideal threshold value for the sky border function. Additionally, the algorithm initializes the energy function values and the list of optimal border points. A for loop is constructed to iterate over each sample point in the search space. After each iteration, the function used to determine the threshold value, which plays a significant role in finding the best value for the sky border function, is returned. This function is implemented based on equation (1).

$$t = thresh\_\min + \frac{threshmax - threshmin}{n-1}\, x\, (k-1) \quad (1)$$

While the **border_optimisation()** function searches for the optimal border position that maximizes the energy value, it can be calculated using the **calc_energy_optimization()** function for a given set of border points. Initially, the **create_mask()** function is employed to highlight the sky region in the image. The function checks whether the detected sky region contains any sky pixels. In case the sky region is empty, it returns a very low energy value (1e-20) to signify an improper sky region configuration. Subsequently, masked array functionality extracts the ground and sky regions from the binary mask. The function ensures enough pixels in the sky region to proceed with covariance calculations. If there are insufficient samples, it returns a low energy value. However, if sufficient samples are available, the function uses the cv2.calcCovarMatrix function to determine the covariance matrices and average RGB values for the sky and ground regions.

Finally, the energy optimization function J_n is computed based on equation (2) from a study by Shen & Wang [4], incorporating the covariance matrices, average RGB values, and a gamma parameter. The function returns the resulting energy value, where a higher value indicates a higher accuracy in representing the sky region.

$$J_n = \frac{1}{\gamma \cdot |\Sigma s| + |\Sigma g| + \gamma \cdot | \lambda_1^s | + | \lambda_1^g |} \qquad (2)$$

Besides, the **create_mask()** function generates a binary mask to separate the sky region in the image based on the given border points. A mask is initialised based on the image dimension, and all pixels are set to 0. The function is then iterated through the list (x,y) border points, and an intensity value of 255 is assigned to the pixels below these points, representing the ground region. The original mask is then reversed to create a new binary mask, highlighting the sky region with an intensity value of 255 while the ground region is 0.

The detected binary mask is then applied to the **postprocess_mask()** function to enhance the continuity and completeness of the sky region. The function utilises the morphological closing algorithms from the OpenCV library. The algorithms help to fill small gaps and holes in the non-sky region of the mask. The final binary mask is returned.

For the nighttime image, a best_daytime_mask variable is initialised. When iterating the dataset, the algorithm compares each mask's accuracy with the previous best accuracy and saves the mask to the variable. The mask is then applied to the nighttime image to detect the sky region.

To evaluate the effectiveness of the algorithm, accuracy and Mean Squared Error(MSE) are calculated by the **evaluation()** function based on the detected mask and ground truth mask given by the SkyFinder dataset. The function first ensures that both masks have the same dimension. Then, it calculates the confusion matrix, which is True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) values. The accuracy is determined by equation (3), and MSE is determined by equation (4). Higher accuracy and lower MSE indicate better segmentation results.

$$accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \qquad (3)$$

$$np.mean((ground_{truth_{mask}} - detected\_sky\_mask) \; ** \; 2) \qquad (4)$$

The **visualize_and_save_results()** function will visualize the output in a figure using the matplotlib library, showing the original image, detected sky region and detected skyline. A line of text indicating the image filename, daytime or nighttime, accuracy and MSE is also displayed on top of the figure for easy reference. Afterwards, the figure is saved into the result folder based on the folder name, followed by "_result.png". If the result folder does not exist, the algorithm will create the folder in the same directory as the code.

All functions have been included in the **main().** To run the algorithm, call the function and the result will automatically saved in the result folder.

IV. RESULTS AND DISCUSSIONS

A. *Experimental Setup*

The proposed algorithm was evaluated using a dataset of images taken in variety of weather situations, including daytime and nighttime. The dataset consists of four folders, named 623(3790 images), 684(4094 images), 9730(1805 images) and 10917(3568 images). A random number of images was randomly selected from each folder for evaluation. The evaluation metrics are accuracy and Mean Squared Error (MSE).

B. *Discussions of the results*

The following table displays the accuracy and MSE for 100 images randomly selected from each dataset.

TABLE I. AVERAGE ACCURACY AND MEAN SQUARED ERROR (MSE) FOR RANDOM 100 IMAGES

Dataset	Average Accuracy (%)	Mean Squared Error (MSE)
623	88.78	0.1122
684	92.71	0.0729
9730	90.13	0.0987
10917	93.47	0.0653

The following table displays the accuracy and MSE for 500 images randomly selected from each dataset

TABLE II. AVERAGE ACCURACY AND MEAN SQUARED ERROR (MSE) FOR RANDOM 500 IMAGES

Dataset	Average Accuracy (%)	Mean Squared Error (MSE)
623	88.85	0.1115
684	95.07	0.0493
9730	89.77	0.1023
10917	93.07	0.0693

The following table displays the overall accuracy, MSE, and program execution times for 100 and 500 images randomly selected from each dataset.

TABLE III. OVERALL AVERAGE ACCURACY, OVERALL MEAN SQUARED ERROR (MSE) AND PROGRAM EXECUTION TIME FOR RANDOM 100 AND 500 IMAGES

Dataset	Overall Average Accuracy (%)	Overall Mean Squared Error (MSE)	Program Execution Time (s)
Total of 100 sample images per folder	91.29	0.0871	2435.0357
Total of 500 sample images per folder	91.65	0.0835	11831.1031

Based on the results, the algorithm demonstrated effectiveness in detecting the sky region, achieving accuracy above 80%, with an overall average accuracy of approximately 90%. The result indicate that the algorithm performs well in accurately identifying the sky region in images. However, it can be observed that the program execution time is relatively high, around 6 seconds per image. Hence, the algorithm was tested on a smaller dataset to reduce the program execution time.

C. Limitations

As mentioned in the result section, further optimization might be required to improve the algorithm's efficiency when applied to larger datasets. Additional optimization strategies should be investigated, such as hardware specification. Besides, misclassification of daytime and nighttime image. As the classification relies on the average brightness of the image, images could be misclassified due to strong artificial lighting or low natural light.

Furthermore, using a fixed kernel size for the post-processing step may not be optimal. It may cause the sky region to be either over-smoothing or under-smoothing, which could result in biases in the final mask.

Lastly, the proposed algorithm does not directly process the nighttime images but relies on the best daytime mask. A more complex approach might be needed to overcome this limitation, such as using a machine learning classifier trained on nighttime images.

[1] R. P. Mihail, S. Workman, Z. Bessinger, and N. Jacobs, “Sky Segmentation in the wild: An empirical study,” 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016. doi:10.1109/wacv.2016.7477637

[2] Mohamed Roushdy, “Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter,” GVIP Journal, vol. 6, no. 4, Dec. 2006.

[3] R. Sravanthi and A. Sarma, "Efficient Horizon Line Detection using Clustering and Fast Marching Method" International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 4, pp. 2913-2918, 2020.

[4] Y. Shen and Q. Wang, “Sky region detection in a single image for Autonomous Ground Robot Navigation,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 10, p. 362, 2013. doi:10.5772/56884

[5] K. A. Nice *et al.*, “Sky Pixel detection in outdoor imagery using an adaptive algorithm and machine learning,” *Urban Climate*, vol. 31, p. 100572, 2020. doi:10.1016/j.uclim.2019.100572

[6] N. Laungrungthip, A. McKinnon, C. Churcher and K. Unsworth, "Edge-Based Detection of Sky Regions in Images for Solar," 23rd International Conference Image and Vision Computing New Zealand, pp. 1-6, 2008.

[7] Y. Song, H. Luo, J. Ma, B. Hui, and Z. Chang, “Sky detection in Hazy Image,” Sensors, vol. 18, no. 4, p. 1060, 2018. doi:10.3390/s18041060

[8] K. A. Nice et al., “Sky Pixel detection in outdoor imagery using an adaptive algorithm and machine learning,” Urban Climate, vol. 31, p. 100572, 2020. doi:10.1016/j.uclim.2019.100572

[9] N. Dhanachandra, K. Manglem, and Y. J. Chanu, “Image segmentation using K -means clustering algorithm and subtractive clustering algorithm,” Procedia Computer Science, vol. 54, pp. 764–771, 2015. doi:10.1016/j.procs.2015.06.090].

[10] B. Zafarifar, H. Weda, and P. H. de With, “Horizon detection based on Sky-color and edge features,” SPIE Proceedings, 2008. doi:10.1117/12.766689