Sky Pixel Identification in Images

Yiwen Xu

This project focuses on identifying sky pixels in an image using traditional computer vision techniques. The chosen methods include gradient-based sky region extraction and morphological operations. The primary motivation is to enhance the accuracy of skyline detection and provide a cleaner identification of the sky region.

Key Techniques

1. Gradient-Based Sky Region Extraction

Rationale:

Utilizes the Laplacian filter to enhance edges, emphasizing transitions from the sky to other regions. The Laplacian filter is employed to emphasize edges present in the image, particularly focusing on the transition areas from the sky to other regions. This is crucial in identifying the distinct boundaries that define the sky region. By applying the Laplacian filter, the resulting output effectively accentuates transitions in pixel intensity. These transitions are indicative of the edges present in the image, with a specific emphasis on the delineation between the sky and other elements.

The gradient mask generated based on the Laplacian output highlights potential sky regions in the image. The Laplacian output serves as the basis for generating a gradient mask. This mask is designed to capture and highlight potential sky regions within the image. The gradient mask acts as a binary indicator, distinguishing areas with significant intensity changes, which often correspond to the sky.

Implementation:

- Conversion of the image to grayscale.
- Blurring to reduce noise.
- Median blur for additional noise reduction.
- Application of the Laplacian filter to enhance edges.
- Generation of a gradient mask based on the Laplacian filter output.
- Erosion to remove small artifacts in the gradient mask.

```
height, width, _ = img.shape

# Convert the image to grayscale
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
# Blur the image to reduce noise
img_blurred = cv2.blur(img_gray, (9, 3))

# Apply median blur to further reduce noise
cv2.medianBlur(img_blurred, 5)

# Apply Laplacian filter to enhance edges
laplacian_output = cv2.Laplacian(img_blurred, cv2.CV_8U)

# Create a gradient mask based on Laplacian filter output
gradient_mask = (laplacian_output < 6).astype(np.uint8)
```

2. Morphological Operations

Rationale

The incorporation of morphological operations, specifically erosion, is a crucial step in the sky pixel identification process. The primary objectives are to mitigate noise, eliminate small artifacts, and ensure a more accurate and cleaner identification of the sky region. The application of morphological operations also aids in smoothing the boundaries of the identified sky region.

Implementation

- Definition of a morphological kernel.
- Erosion of the gradient mask using the defined kernel to remove small artifacts.

```
# Define a morphological kernel
morph_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 3))

# Erode the gradient mask to remove small artifacts
gradient_mask_eroded = cv2.morphologyEx(gradient_mask, cv2.MORPH_ERODE,
morph_kernel)
```

3. Skyline Refinement

Rationale

Its primary rationale is to address potential inaccuracies or noise in the initial skyline detection. The refinement process aims to enhance the precision of identifying the transition from sky to non-sky regions within each column of the binary mask. By analyzing column-wise transitions and applying adjustments, the function contributes to a more accurate delineation of the sky region.

Implementation

- Apply median filter for smoothing
- Identify indices of first zero and first one in the smoothed column
- Adjust skyline if there is a significant transition

```
num_rows, num_cols = mask.shape

   for col_index in range(num_cols):
       current_column = mask[:, col_index]
       smoothed_column = medfilt(current_column, 19)

       try:
           first_zero_index = np.where(smoothed_column == 0)[0][0]
           first_one_index = np.where(smoothed_column == 1)[0][0]

           # Adjust skyline if there is a significant transition
           if first_zero_index > 20:
               mask[first_one_index:first_zero_index, col_index] = 1
               mask[first_zero_index:, col_index] = 0
               mask[:first_one_index, col_index] = 0
```

4. Gradio Interface for Real-Time Interaction

Rationale

Integration of Gradio for an interactive user interface, allowing users to visually identify sky pixels in real-time. Enhances the user experience and facilitates easy exploration of the sky pixel identification process.

Implementation

The gr_sky_region_extraction function is a critical component of the Gradio interface designed for interactive sky pixel identification. It involves the following steps:

1. Convert Gradio Image to Numpy Array
2. Sky Region Extraction
3. Conversion to RGB Format
4. Output

```
def gr_sky_region_extraction(image):
   """
```

```
    Converts a Gradio Image type to a numpy array, extracts the sky region,
and returns the resulting image in RGB format.

    Parameters:
    - image: Gradio Image type.

    Returns:
    - numpy.ndarray: Image in RGB format with identified sky pixels.
    """
    # Convert Gradio Image type to numpy array
    image_array = np.array(image)

    # Sky region extraction
    img_gray = cv2.cvtColor(image_array, cv2.COLOR_BGR2GRAY)
    masked_image, binary_mask = get_sky_region_gradient(image_array)

    # Convert the output image to RGB for Gradio compatibility
    output_image_rgb = cv2.cvtColor(masked_image, cv2.COLOR_BGR2RGB)

    return output_image_rgb
```

Challenges

One significant challenge was the sensitivity of the initial skyline detection process to noise, leading to inaccuracies in delineating the boundaries of the sky region. To address this issue, the refine_skyline function was introduced. By applying a median filter and adaptively adjusting the binary mask column-wise, the function successfully mitigated noise and improved the precision of skyline identification.

The presence of small artifacts in the gradient mask after Laplacian filtering and erosion presented another challenge in maintaining the overall accuracy of the sky region identification. To overcome this, morphological operations, particularly erosion, were applied to the gradient mask. The choice of the kernel size in erosion played a key role in removing artifacts while preserving the integrity of the identified sky region.

Reflections

<u>Effectiveness of the Approach</u>

The adopted approach has proven to be effective in enhancing skyline detection accuracy and providing a cleaner identification of the sky region. The combination of gradient-based sky region extraction and morphological operations has demonstrated robust performance across diverse images.

The refine_skyline function played a pivotal role in addressing noise and refining the binary mask, contributing to the overall effectiveness of the approach.

The effectiveness of the approach is particularly evident in its adaptability to different skyline characteristics. By employing techniques that emphasize transitions and incorporating user-friendly interfaces, the system offers a versatile solution for identifying sky pixels in various scenarios.

Limitations

The system exhibits several limitations that impact its overall performance.

One limitation is the sensitivity of the system to variations in lighting conditions and complex skyline patterns. Very limited effectiveness in extreme lighting or dark night conditions.

Another noteworthy limitation arises from the system's challenge in distinguishing building reflections or objects with colors similar to the sky. The reliance on color and gradient-based features may lead to misclassifications, especially when dealing with reflections or structures that share similar hues with the sky.

Additionally, the dependence on traditional computer vision methods introduces constraints when faced with images featuring unconventional sky shapes or drastic lighting changes.

These limitations collectively underscore the need for further refinements, adaptive algorithms, and advanced techniques to enhance the system's robustness across diverse environmental scenarios.

Potential Improvements

Implement dynamic parameter adjustment based on the image's lighting conditions. This could involve automatically adapting filter sizes, threshold values, or kernel sizes to accommodate varying illumination levels.

Explore and implement advanced lighting correction techniques, such as histogram equalization or adaptive histogram matching, to enhance the system's resilience to diverse lighting conditions.

Develop algorithms or techniques specifically designed to handle unconventional sky shapes. This could involve segmenting the image based on color, texture, or other features that are indicative of the sky region, enabling the system to adapt to a wider range of skyline variations.

Learning Outcomes

The assignment provided hands-on experience in applying traditional computer vision techniques, such as gradient-based methods and morphological operations, to solve a real-world problem—sky pixel identification. This practical application enhanced my understanding of how these techniques function in image-processing tasks.

The project emphasized the significance of developing systems that can adapt to diverse image characteristics. The adaptive nature of the refine_skyline function and the exploration of parameter tuning highlighted the need for versatility in image processing solutions. Dealing with challenges like small artifacts and noise in image processing pipelines required the implementation of morphological operations and adaptive techniques. This experience enhanced my skills in addressing common issues encountered in real-world image analysis.

Integrating Gradio for real-time user interaction provided insights into designing user-friendly interfaces for image processing applications. Understanding the importance of clear visualizations and intuitive controls is crucial for creating effective user interfaces.

In conclusion, the implemented approach showcases a solid foundation for sky region extraction, with room for improvement in handling diverse environmental conditions and complex skyline scenarios. The project provided valuable insights into image processing techniques and interface design.