

Hough Transform Report

Task 1: Work your way through the script below. Using online documentation, understand how each function works, from edge detection, Hough Transform and line detection. Write a summary of how this algorithm works.

Summary: The Hough Transform is an algorithm used to detect straight lines in a single-channel image. The steps of the algorithm are as follows:

1. Read the image and convert it to grayscale: The Hough Transform algorithm works only with single-channel images, so the image must be converted to grayscale.
2. Apply the Canny algorithm: This algorithm detects the edges in the image. It involves several steps including Gaussian blurring (to reduce noise), gradient calculation, Non-Maximum Suppression (NMS), double thresholding, and edge tracking to ensure continuous edges.
3. Use `'skimage.transform.hough_line()'`: This function applies the Hough Transform to the edge-detected image. The parameter `theta` represents a predefined range of angles to map each edge point to the parameter space. The accumulator matrix records each (θ, d) pair's votes, and this matrix is one of the outputs of the function.
4. Use `'skimage.transform.hough_line_peaks()'`: This function extracts the peak values from the accumulator matrix, identifying the parameters of each line segment, specifically the angle and the distance from the origin. These two values are key to constructing a line.
5. Draw the detected lines on the original image: Finally, use the two parameter arrays to draw the detected lines on the original image.

Task 2: What is the effect of increasing/decreasing the required number of peaks in `'hough_line_peaks'`?

Answer: Changing the number of peaks in `'hough_line_peaks'` affects the number of lines detected in the image. Increasing the number of peaks results in detecting more lines, which allows more lines to be drawn in the image. However, this may lead to inaccuracies, as more lines can include noise or incorrectly detected features. Conversely, decreasing the number of peaks tends to focus on the most prominent lines, potentially missing some finer details.

Task 3: Replace the Canny Edge detector with other algorithms. Which one do you think performs best and why?

Answer: When different filters like Sobel or Roberts are applied, noticeable differences appear. The Sobel filter is effective in noise control but performs less well when detecting thin edges, as it may pick up irrelevant details, making line detection less precise. The Roberts filter is better at detecting thin edges but is not as robust in handling noise.

The Canny algorithm performs best because it applies Gaussian blurring to reduce noise, uses Non-Maximum Suppression (NMS) to ensure single response edges, and applies hysteresis thresholding to preserve strong, continuous edges.

Task 4: Examine the guide for Probabilistic Hough Transform: https://scikit-image.org/docs/stable/auto_examples/edges/plot_line_hough_transform.html#probabilistic-hough-transform. Apply probabilistic hough transform to the provided image. Describe in the report how this algorithm works, particularly when finding the start/finish of a line.

Answer: The Probabilistic Hough Transform is an extension of the traditional Hough Transform. It uses random point sets near edge points to detect line segments, focusing only on finding the start and end points of each line segment rather than the entire line. This method reads along continuous edge segments to gather information about line segments without needing the complete line data.

When using the ‘`skimage.transform.probabilistic_hough_line()`’ function, three parameters must be set:

1. **Threshold:** The value in the accumulator matrix must exceed this threshold to be considered a valid line.
2. **Minimum line length:** Only line segments longer than this value are considered valid.
3. **Maximum gap:** This parameter sets the maximum allowed gap between points or segments; if the gap is smaller than this value, the segments will be merged into a single line segment.