

Introduction

The goal of this assignment is to detect pedestrian crossings in images taken from a car's point of view. Firstly, I will describe in depth the theory behind my process, and then present results and analyses both successes and failures.

1 Theory of the Processing

1.1 Smoothing

To help cope with noise, all my images went through an initial stage of smoothing. In the context of this problem, it was crucial that the edges of the pedestrian crossings remained well-defined and as sharp as possible. The reason for this is because I wanted my edge detection to work as good as possible. As a result, I settled with **median smoothing**, primarily because it does not blur edges too much.

I experimented with multiple iterations of median smoothing, however I found the best results with just one iteration. Additionally, I opted for a filter size of 3x3 as I wanted the edges to be as sharp as possible. With larger filter sizes, I was worried it would corrupt the image too much and deviate too far from the original.

1.2 Edge Detection

At first, I experimented with region processing using Mean Shift Segmentation (MSS). However, I found that it took about two seconds to process the image using MSS. In the context of the problem, I opted for a more efficient approach as it is vital that images are produced rapidly in a car imaging system. This is how I settled with edge processing.

I chose to go with **Canny Edge Detection** as opposed to Boundary Chain Codes, because with Canny I could have more customisation over the results by changing the parameters, whereas with Boundary Chain Codes it was more limiting.

The low and high thresholds were chosen mostly based on trial-and-error. I ensured that my low threshold value was conservative, as this is indicative of the number of contours detected by Canny, and I wanted to be sure that my edge detection would not miss out on any potential pedestrian crossings.

2 Results and Analysis

Conclusion