MAY 8, 2024

Bauhaus-Universität Weimar

ASSIGNMENT NO. 2

IMAGE ANALYSIS AND OBJECT RECOGNITION – GROUP 13

KLAURENT MADHI, BHAGATH MAMINDLAPELLY, ABDUL RAHMAN

Table of Contents

Brief

This documentation explains the inner workings of the Gaussian Filter and Förstner Operator, exploring each step-in detail and explaining its contribution to the overall goal of detecting corners within an image.

Setting up

We start by loading the image named "*ampelmaennchen.png*". Since corners are often characterized by sharp transitions in intensity, the code prioritizes grayscale information over color. Therefore, it converts the loaded RGB image, to a grayscale representation.

Furthermore, to ensure precise calculations throughout the process, we convert the grayscale image from its default integer format to double precision. This conversion elevates the data type to floating-point numbers with a wider range and higher accuracy, minimizing errors during calculations involving decimals.


Figure 1: Original Image provided for the task

Figure 2: Grayscale Image

Gaussian Filtering

The second step focuses on identifying areas in the image where the intensity (brightness) of pixels undergoes significant changes. These regions often correspond to edges and corners, which stand out due to their contrasting brightness values compared to their surroundings.

The code tackles this challenge by employing a Gaussian filter with a predefined standard deviation (sigma) of 0.5. This sigma acts as a control, dictating the level of blur applied by the filter. A higher sigma results in a "smoother" filter, meaning it blurs the image more.

The Gaussian filter is then applied to the grayscale image in both the horizontal and vertical directions using convolution. This mathematical operation blurs the image in the specified direction while preserving areas with substantial intensity variations. The processed image, now highlighting potential edges and corners, is stored in the variable "gradient magnitude" and saved as a separate image named "GoG_Image.png".

Figure 3: Image after Gaussian Filter to separate the edges

Förstner Interest Operator

This step introduces the Förstner Interest Operator. The operator leverages the concept of local intensity variations to achieve this objective.

The code iterates through each pixel in the image. For each pixel, it carves out a small window 5x5 square, centered at that specific pixel. This window zooms in on the local neighborhood surrounding the pixel.

Next, the code extracts information from the previously computed "gradient magnitude" image, which now holds the edge and corner information. It retrieves two crucial components: the horizontal gradient (Ix) and the vertical gradient (Iy) within the extracted window. These gradients represent the rate of intensity change in the horizontal and vertical directions, respectively.

Then, the code constructs a 2x2 matrix (M) for each pixel. This matrix encapsulates valuable information about the local intensity variations around the pixel. The specific elements within matrix M are calculated based on the extracted gradients within the window.

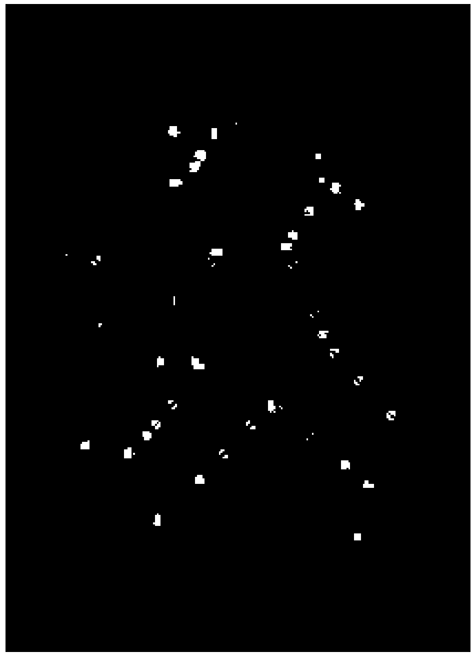

Figure 4: Interest Point Mask after Förstner Operator

W and Q Values

Here we calculate the two critical values for each pixel: W and Q. These values play a pivotal role in discerning true corners from other intensity variations.

A higher W value indicates a sharper corner, with significant intensity changes in both horizontal and vertical directions. Conversely, a low W value suggests a flatter region or an edge, where the intensity variation is less pronounced in one direction.

A high Q value suggests a round corner, where the intensity changes gradually in both directions. In contrast, a value closer to 0 indicates a sharper corner, where the intensity transitions abruptly in both horizontal and vertical directions.

We threshold for both W and Q (0.004 for W and 0.5 for Q) to refine the corner detection process. These thresholds function as filters, eliminating false positives and ensuring a more robust identification of true corners.

Visualization

The final step focuses on visualizing the corners identified throughout the process. The code overlays red plus signs on the original color image ("ampelmaennchen.png"). These plus signs pinpoint the locations designated as potential corners based on the Förstner Interest Operator, the calculated W and Q values, and the applied thresholds.

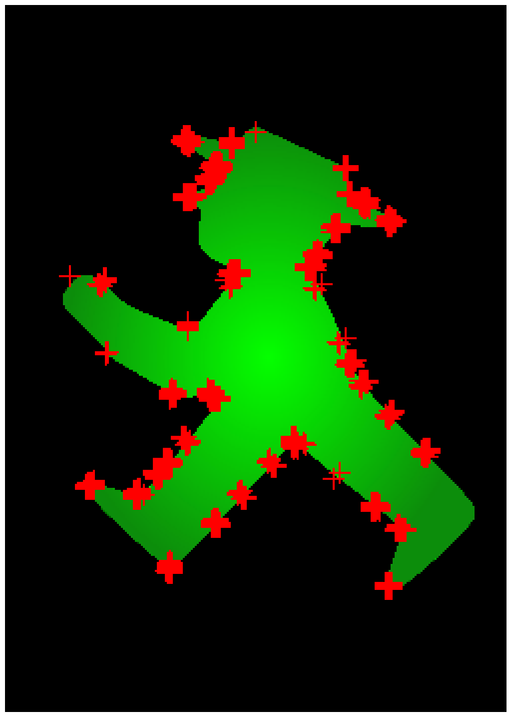

Figure 5: Final Overlayed Image

References

- Code Debugging and testing through ChatGPT and Google Gemini
- 3-D Gaussian filtering of 3-D images - MATLAB imgaussfilt3 - MathWorks Deutschland
- Find gradient magnitude and direction of 2-D image - MATLAB imgradient - MathWorks Deutschland
- Förstner operator – StachnissLab (uni-bonn.de)
- OpenCV: Smoothing Images