

CS 425  
Term Project  
Xialei Fang  
200434895

## Project description

This project is to do a linear structure extraction using Hough Transform.

In this project, I did:

1. Apply Laplacian Operator to the "building.raw" and find all zero-crossing points.
2. Apply Sobel Operator to the "building.raw" image to obtain the gradient at all pixels.
3. Generate an edge image where a pixel is an edge point if it is a zero-crossing point and the gradient at this point is greater than or equal to a pre-specified threshold.
4. Implement the Hough transform algorithm to extract three longest linear structures from your edge image.

The C program includes Laplacian operator, zero-crossing detection, Sobel operator, Hough transform, and extraction of three longest linear structures from the edge image.

## Usage:

Compile the program using the following command:

```
gcc Project.c -o project -lm
```

Run the compiled program with the following command:

```
./project input_image.raw output_zero.raw output_gradient.raw output_edge_map.raw  
output_hough.raw output_final.raw
```

For example:

```
./project building.raw zero.raw gradient.raw edge.raw hough.raw final.raw
```

## Functions and Operations:

1. `laplacian_operator()`:
  - This function performs the convolution operation using the Laplacian kernel.
  - It iterates over the image pixels, applies the 3x3 kernel, and stores the result in `out_buf`.
2. `zero_crossing()`:
  - This function detects zero-crossings in the Laplacian output and marks them.
  - It iterates through each pixel in the image except for the boundaries.
  - If a zero-crossing is detected, mark it in `'out_zero'`.
3. `sobel_operator()`:
  - Applies the Sobel operator to the input image to obtain the gradient at all pixels.
  - Generates an edge image where a pixel is an edge point if it is a zero-crossing point and the gradient at this point is greater than or equal to a pre-specified threshold.
4. `hough()`:
  - Apply the Hough transform to the edge map (`out_EdgeMap`) to detect lines.
  - Uses an accumulator array `A` to store votes for different `rho` and `theta` values.
  - The result is stored in the `out_hough` buffer.
5. `drawLines()`:
  - Extracts the three longest linear structures from the edge image based on the Hough transform results.
  - Draws the lines on the final image.

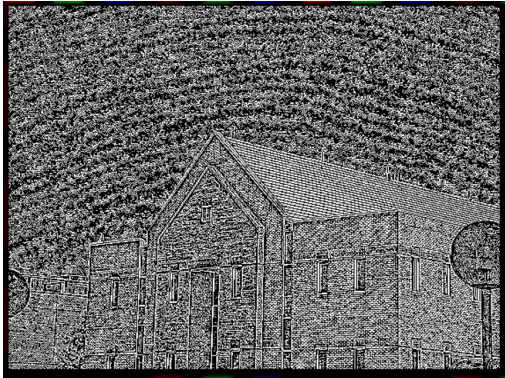
### File I/O:

- The output images include a zero-crossing image, a gradient image, an edge image, an accumulator image after applying the Hough transform, and a final image that shows the three longest lines on the original image.

Note: the final image can be different with different threshold values.

### Output Images:

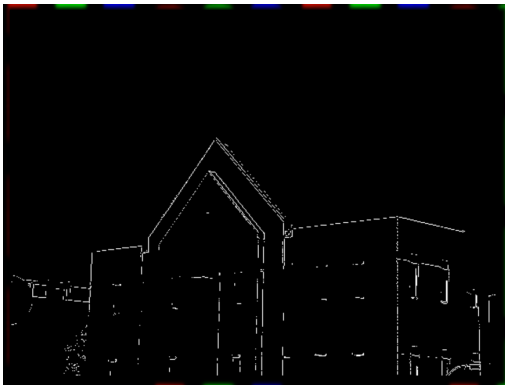
zero-crossing:



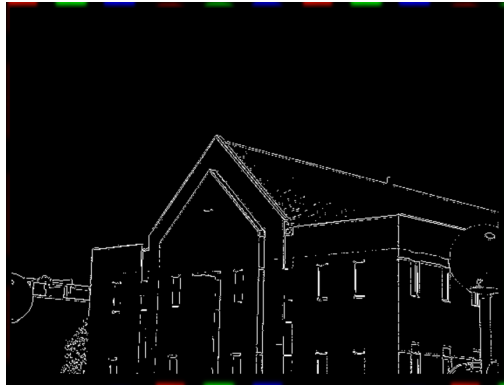
gradient:



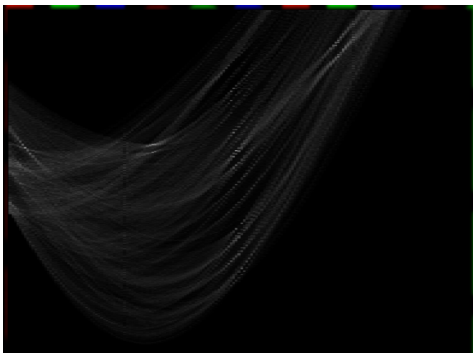
edge (T = 230):



edge (T = 130):



accumulator:



final (T = 230):



final (T = 130):

