Edge Detection and Normalized Cross-Correlation Report

Aman Hogan-Bailey

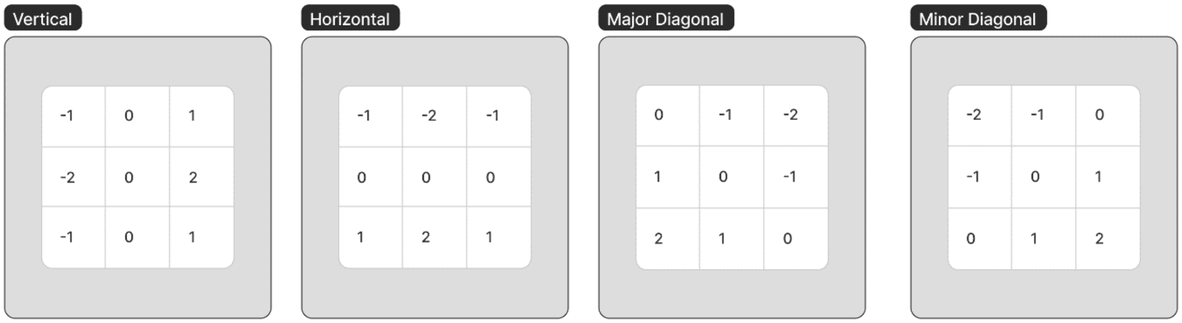
2238-CSE-5331-002, Professor Manfred Huber

The University of Texas at Arlington

Contents

Edge Detection Using Sobel Templates

1. Defined four Sobel templates for detecting vertical, horizontal, and diagonal edges. These will later be used to calculate the individual gradients of the original. Below are the Sobel templates used:



Vertical

-1	0	1
-2	0	2
-1	0	1

Horizontal

-1	-2	-1
0	0	0
1	2	1

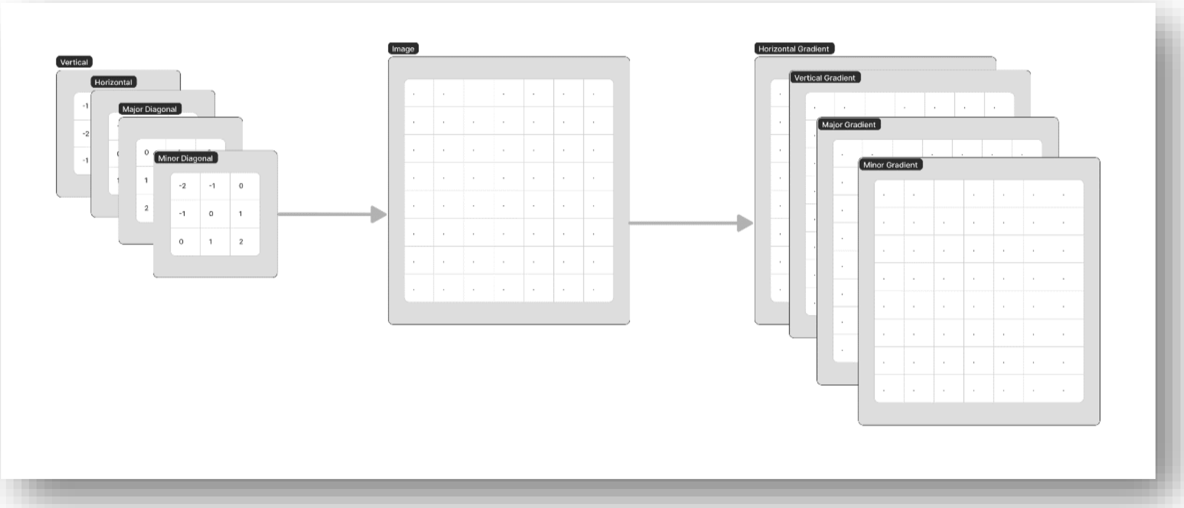
Major Diagonal

0	-1	-2
1	0	-1
2	1	0

Minor Diagonal

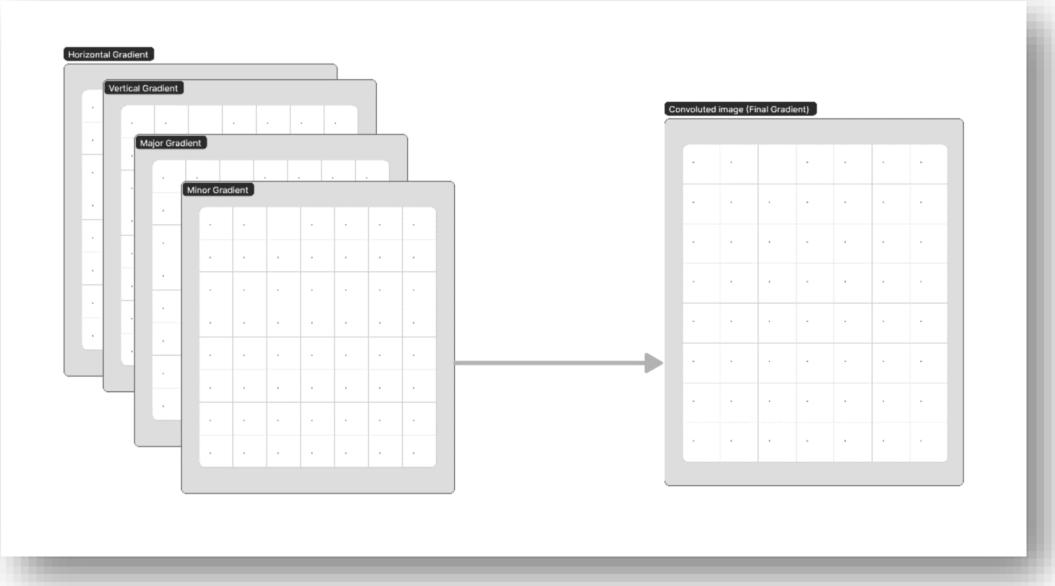
-2	-1	0
-1	0	1
0	1	2

2. Performed convolution for each Sobel template and stored that gradient in a temporary 2D array. For each pixel in the image, the corresponding gradient was calculated by convoluting the Sobel template with the image around that pixel.
3. The formulation used for convolution is as follows:
   a. $G_{template} = \sum_{u=-1}^{1} \sum_{v=-1}^{1} Image(x+u, y+v) * Sobel(u+1, v+1)$
      i. $G_{template} - \ convolution\ gradient$
      ii. $Image(x+u, y+v) - \ pixel\ intensity\ in\ image$
      iii. $Sobel(u+1, v+1) - \ pixel\ inensity\ in\ Sobel\ template$
      iv. $x, y - \ image\ coords.$
      v. $u, v - sobel\ template\ coords.$
4. After the convolution process, four different gradients of the original image were created:
   a. G_x – horizonal gradient
   b. $G_y -$ vertical gradient
   c. $G_{Major} -$ Major Diagonal gradient
   d. $G_{Minor} -$ Minor Diagonal gradient
5. Here is diagram that illustrates the convolution process:



6. After calculating the gradients, I found the final gradient of the processed image using the equation:

    a. $|G| = \sqrt{{G_x}^2 + {G_y}^2 + {G_{Major}}^2 + {G_{Minor}}^2}; Range[Global\ min, Global\ max]$



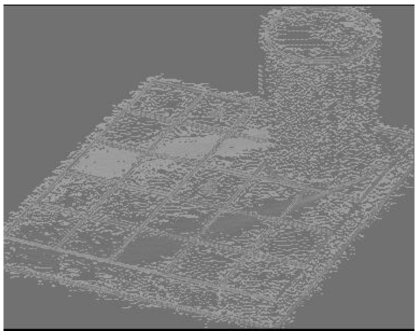
7. While looping through to find the final gradient, I updated the global max and min so I could have a well-defined interval for normalization:

8. Looped through the Gradient and normalized each pixel for the range of [0, 255] which maps the original gradient magnitudes to the standard 8-bit range. The formula used was:
    a. $G_{scaled} = \frac{255*(Original\ value - Global\ Min)}{Global\ Max - Global\ Min}; Range[0, 255]$
9. Finally, I set these normalized values to the pixel in the new image to get the processed image values.

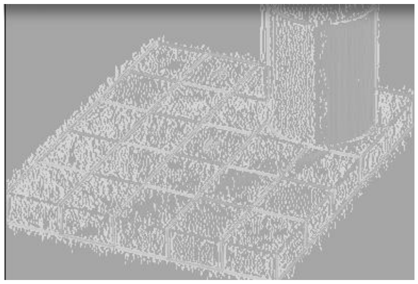
Template Matching Using Normalized Convolution

1. Defined an array to store the normalized cross correlation values.
2. Defined an array to store the template array (selected region of pixels).
3. Defined a global max and min that will help to scale the NCC.
4. Calculated the mean and standard deviation of the template (selected region of pixels).
5. Looped through the image array, calculated image region mean and standard deviation, calculated the NCC values, updated max and min values, and stored the NCC values into the NCC array.
6. Below is the formula used for the NCC score:
    a. $NCC = \frac{\sum_{i=0}^{h}\sum_{j=0}^{w}[(I(i,j)-\mu_I)*(T(i,j)-\mu_I)]}{\sigma_I*\sigma_T}; Range[NCC\ min, NCC\ max]$
        i. $h, w - height\ and\ width\ of\ template$
        ii. $I(x, y) - Image\ pixel$
        iii. $T(x, y) - template\ pixel$
        iv. $\sigma_I, \sigma_T - Image\ std\ and\ tempalte\ std$
        v. $\mu_I, \mu_T - Image\ mean\ and\ template\ mean$
        vi. $NCC - normalized\ cross - correaltion\ value$
7. Loop through the NCC array, and scaled the values into the grayscale [0, 255] range.
    a. $NCC_{scaled}\frac{255*(NCC\ value-Global\ Min)}{Global\ Max-Global\ Min}; Range[0, 255]$

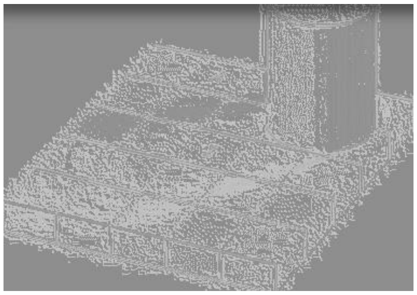
Generated Images
Convolution Edge detection for Chess.lpr



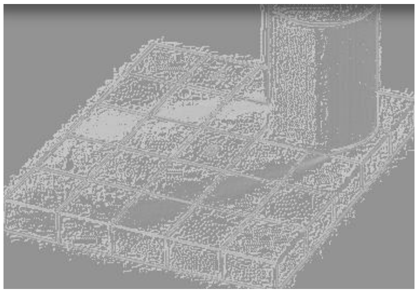
Vertical Sobel Template convolution for Chess.lpr



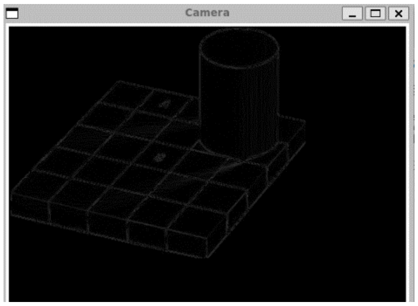
Horizontal Sobel Template convolution for Chess.lpr



Major diagonal Sobel Template convolution for Chess.lpr

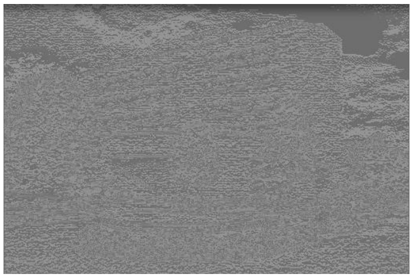


Minor diagonal Sobel Template convolution for Chess.lpr

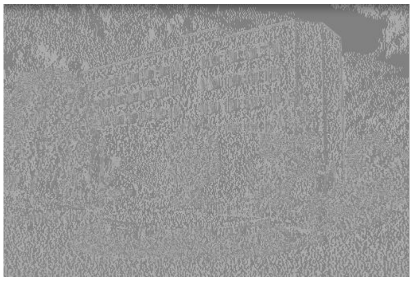


All Sobel Template convolution for Chess.lpr

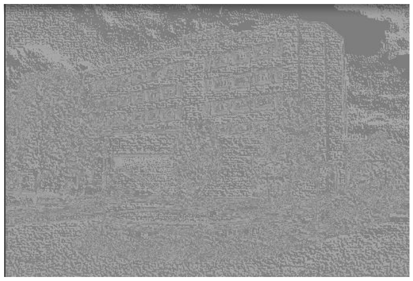
Convolution Edge detection for Nedderman.lpr



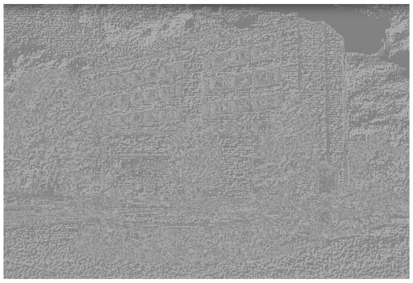
Vertical Sobel Template convolution for nedderman.lpr



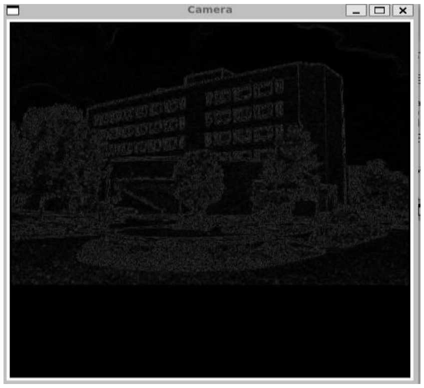
Horizontal Sobel Template convolution for nedderman.lpr



Major Diagonal Template convolution for nedderman.lpr

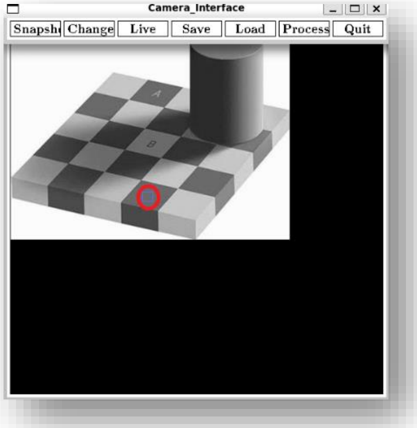


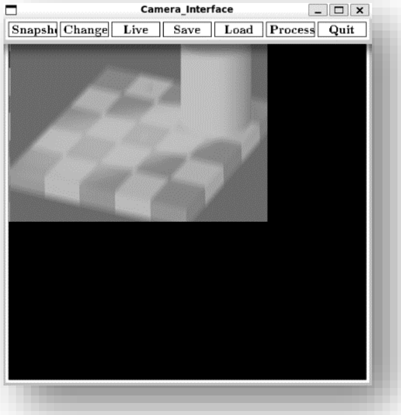
Minor Diagonal Sobel t Template convolution for nedderman.lpr

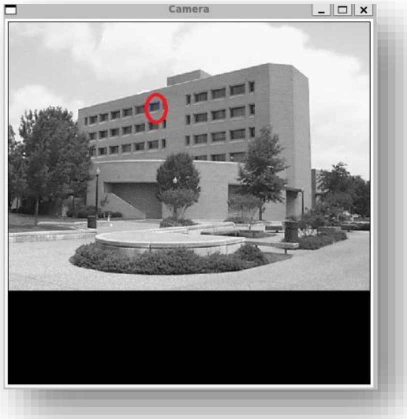


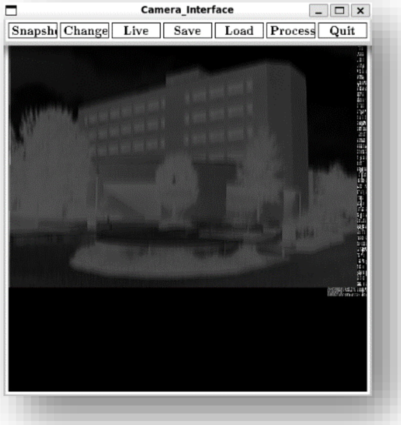
All Sobel Template convolution for nedderman.lpr

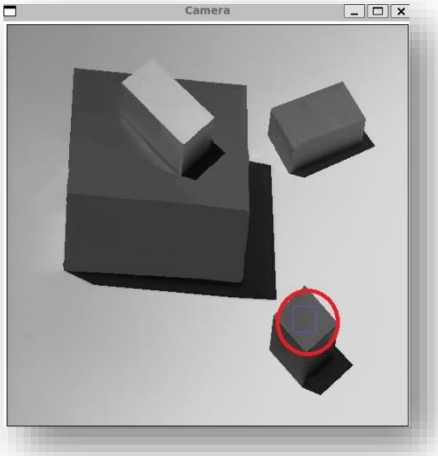
Template Matching using NCC

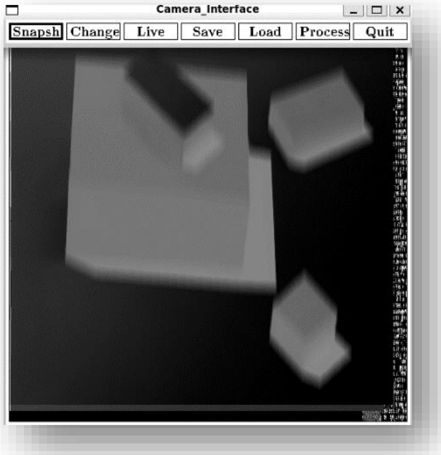









Camera


Camera_Interface
Snapsh
Change
Live
Save
Load
Process
Quit