1.

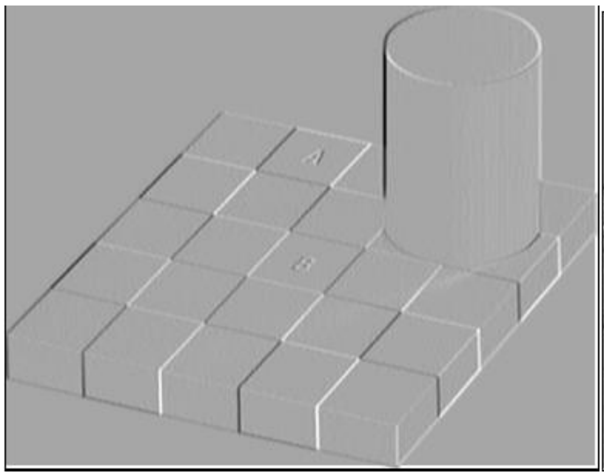
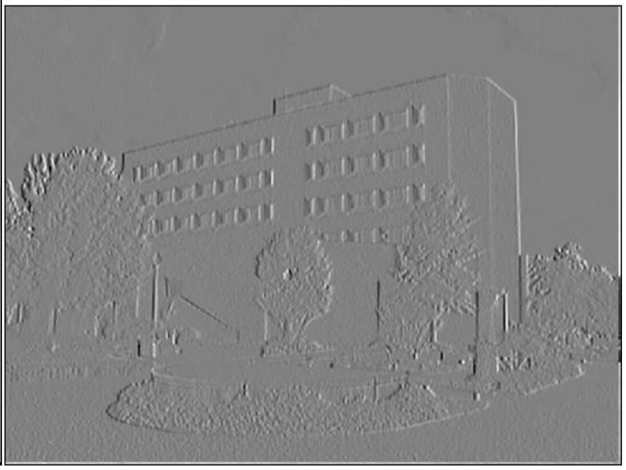
The code for this part is contained in process_image_p1.c

First I define all the Sobel templates in a list and use an index to reference them. I made a function called apply_sobel_template which uses a given template, the image, and an image x and y location. This function is used to calculate values for an intermediate temp_image. Each template is applied by multiplying the values of the template with the values of the image and summing everything up.

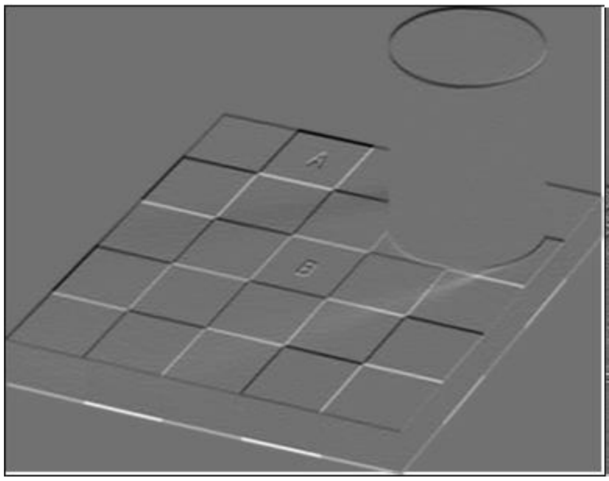
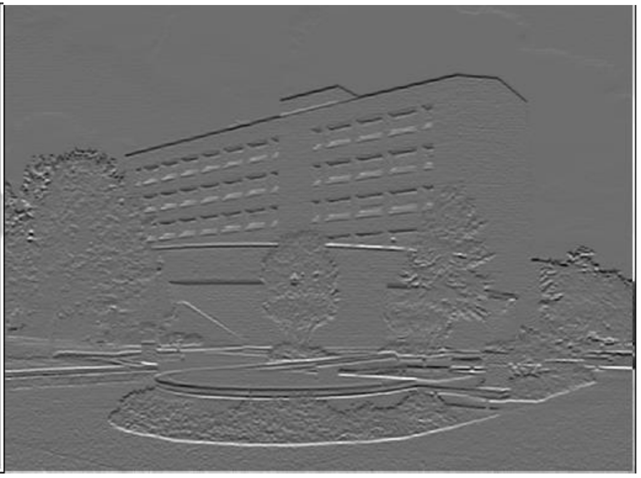
After the raw values for all the pixels are calculated, the values are normalized linearly between the min and max values. The values have the range of 0-255. These values are then put into the proc_image.

Finally, the size of the image is changed but this causes graphical artifacting due to a change in size. I talked to the professor and he said this is ok.

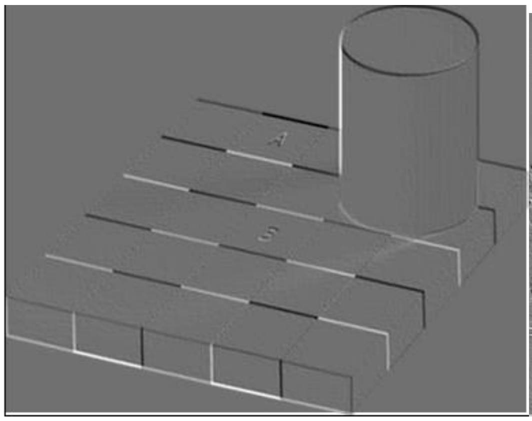
Vert:

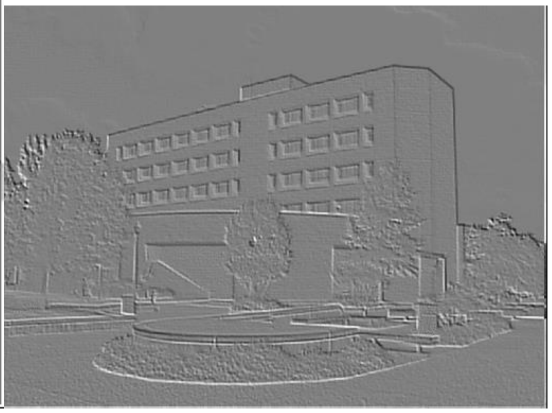



Hor:

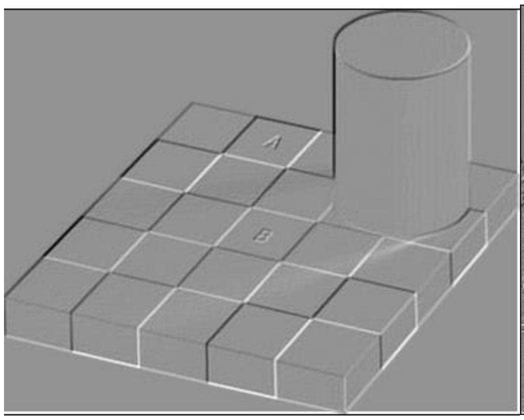



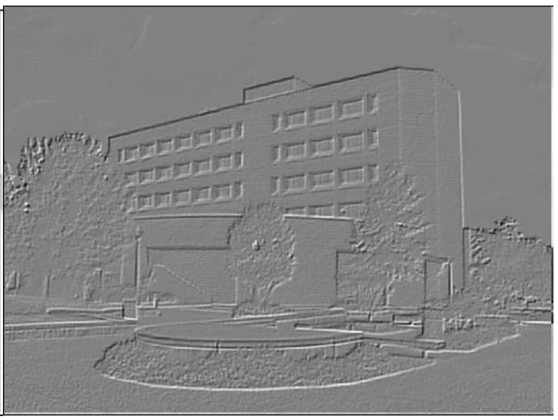
Maj. Diag:





Min. diag:





2.

The code for this part is contained in process_image_p2.c

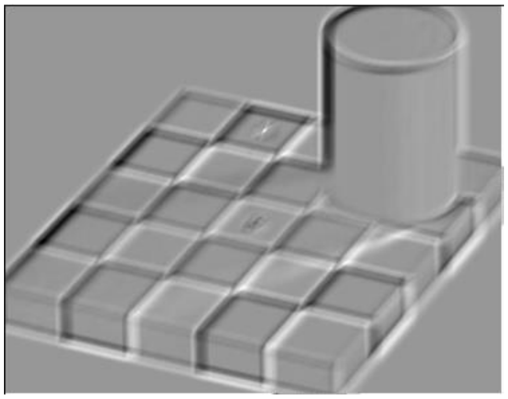
Part two works very similarly as part one. Now instead of applying a sobel template, the code has changed to handle different sized templates.

A temp_image is used to hold the raw image data. This data is retrieved by applying the apply_selected_template function to each applicable pixel. The selected template is normalized by subtracting the mean of the template, this way the total of all the values in the template are equal to 0. Additionally I scaled by the standard deviation of the template to standardize different templates.

Same as before, the temp_image data is then used and resiszed to be within the range of 0-255. This is done by scaling linearly between the min and max values in the temp image

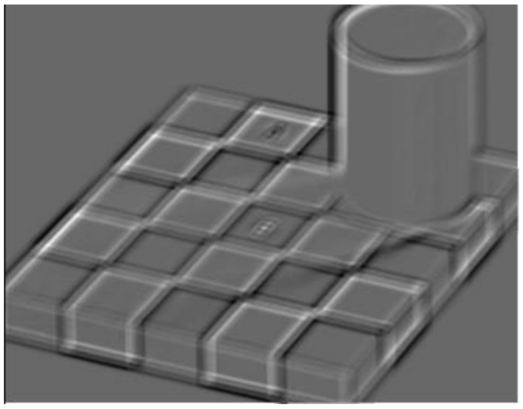
Examples:

Selected A:



You can see that the A letter has a center white dot along with ghosting lines that look like an x. This is because the template and the letter line up along the left and right angled supports (like / and \)

Selected B:



You can see the tripple B ghosting due to the template being able to match with the letter 3 times. This is because B can match with itself in the top half, bottom half, and in both halves.

Selected Window: