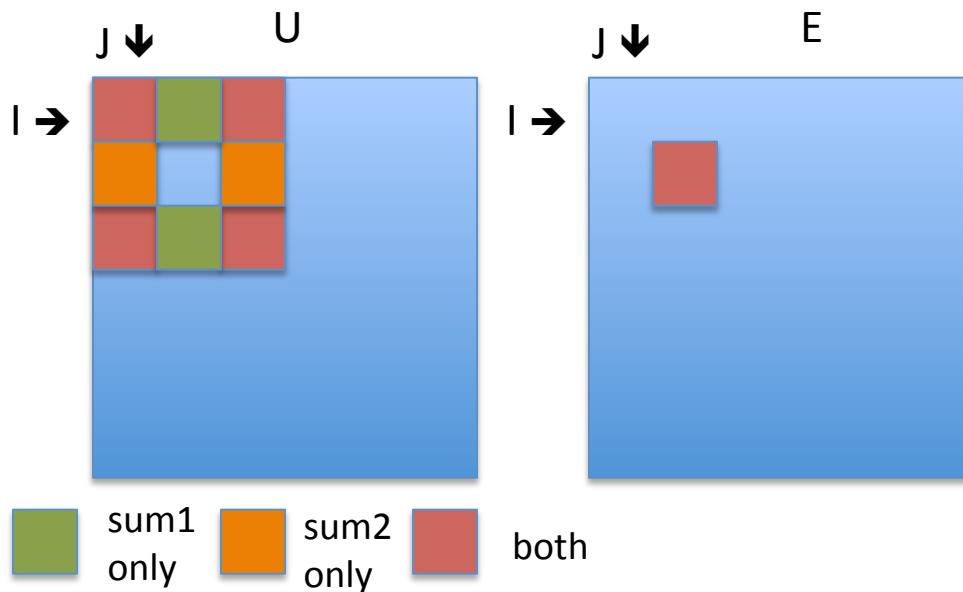


Assignment 2: Memory Hierarchy Optimization

Due Friday, January 31 at 5PM

Sobel edge detection:

Find the boundaries of the image where there is significant difference as compared to neighboring "pixels" and replace values to find edges



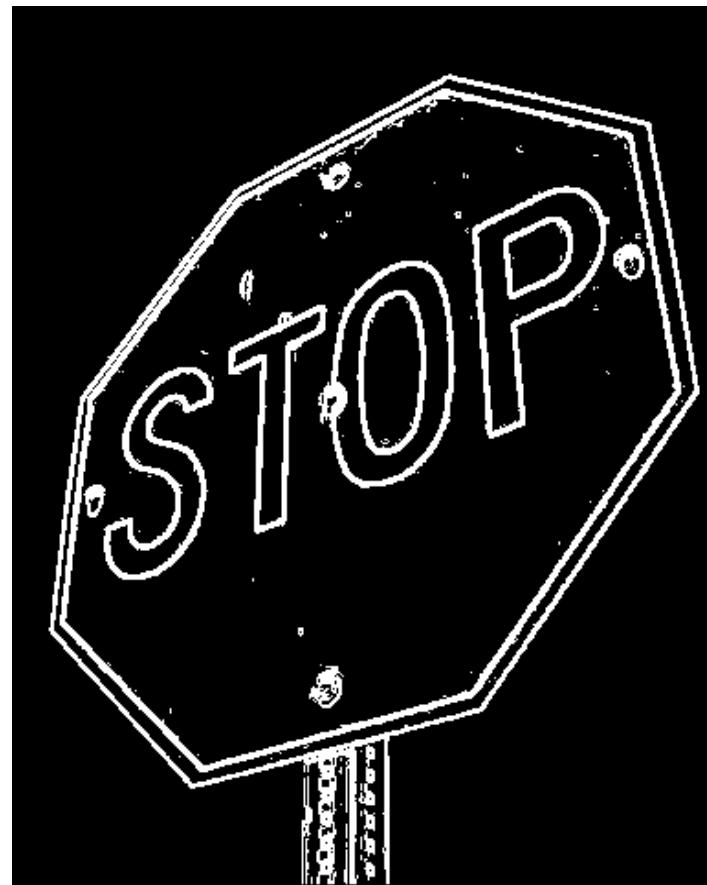
```
for (i = 1; i < ImageNRows - 1; i++)  
    for (j = 1; j < ImageNCols - 1; j++)  
    {  
        sum1 = u[i-1][j+1] - u[i-1][j-1]  
            + 2 * u[i][j+1] - 2 * u[i][j-1]  
            + u[i+1][j+1] - u[i+1][j-1];  
        sum2 = u[i-1][j-1] + 2 * u[i-1][j] + u[i-1][j+1]  
            - u[i+1][j-1] - 2 * u[i+1][j] - u[i+1][j+1];  
  
        magnitude = sum1*sum1 + sum2*sum2;  
        if (magnitude > THRESHOLD)  
            e[i][j] = 255;  
        else  
            e[i][j] = 0;  
    }
```

Example

Input



Output



General Approach

0. Provided

- a. Input file
- b. Sample output file
- c. CPU implementation

1. Structure

- a. Compare CPU version and GPU version output [[compareData in <helper_image.h>](#)]
- b. Time performance of two GPU versions (see 2 & 3 below) [[EventRecord](#)]

2. GPU version 1 (partial credit if correct) implementation using global memory

3. GPU version 2 (highest points to best performing versions) use memory hierarchy optimizations from previous, current lecture

4. Extra credit: Try two different block / thread decompositions. What happens if you use more threads versus more blocks? What if you do more work per thread? Explain your choices in a README file.