Lab3

Kevin Wang

Chelsey Yin

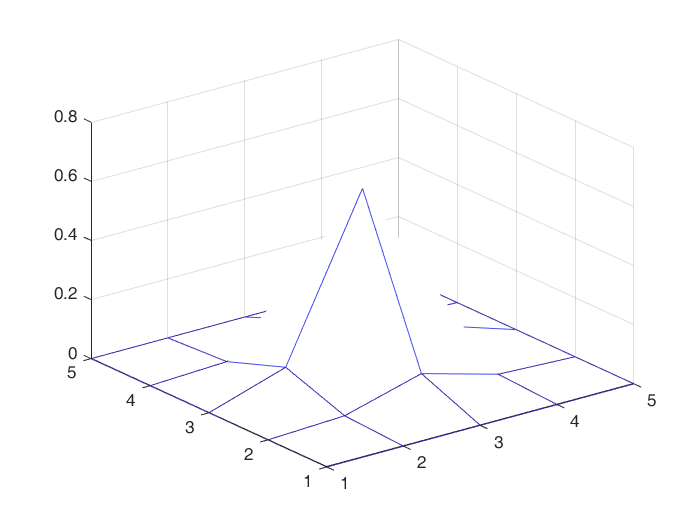
**Type these in and explain the answers you get:**

C = A\*B = [19 22; 43 50]

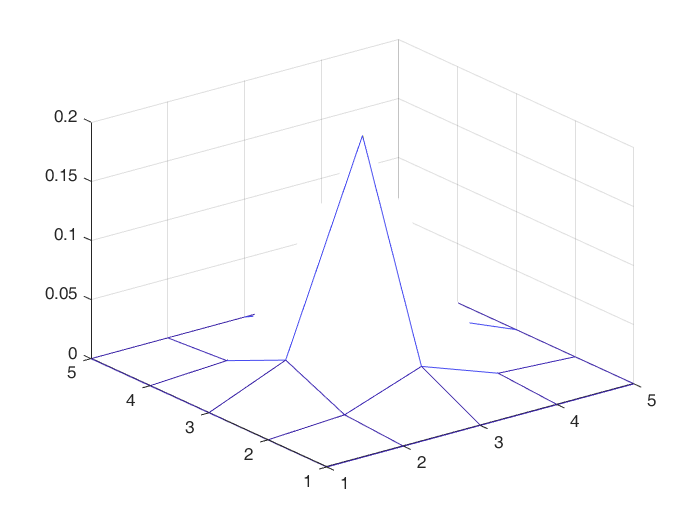
D =A.\*B = [5 12; 21 32]

C works just like the actual matrix multiplication and D times all the numbers at the corresponding place. A is an 2 × 2 matrix and B is an 2 × 2 matrix, their matrix product C is an 2 × 2 matrix, in which the 2 entries across a row of A are multiplied with the 2 entries down a column of B and summed to produce an entry of AB. D multiples value of every position in A and the corresponding position value in B, and put the result into the same position and create a new matrix. From other operations, we find out that the “.” means do the operation for each element in the matrix individually.

**Submit the 5x5 matrix what you used as a filter (it won’t be perfect, that’s OK). Then explain any differences between it and the “true” Gaussian.**

****

mesh(trueGaussian)



mesh(ourGaussian)

We randomly picked “2” as 2D Gaussian filtering coefficient. Later, we compare the image after mesh of trueGaussian and our gfilter. The shapes look the same, but they are different in z-axis which represent weights. Our filter has a smaller weight for the center value than the true Gaussian filter.

**Note that you are not using Matlab’s edge function here. Write your own function called sobel** that takes a grayscale image as input and **returns** 6 matrices, as discussed in class:

**Now (if you haven’t so far), create a single script that reads an image, converts it to grayscale, calls the sobel function on your image, displays the 6 results using imtool, and saves them to appropriate file names. You can just copy the commands from MATLAB’s Command History if you like.**

**Original image**



We get the picture from: <https://blogs.wsj.com/chinarealtime/2013/05/31/giant-yellow-ducks-creator-not-flattered-by-chinese-imitators/>

**Horizontal-edge finding masks**



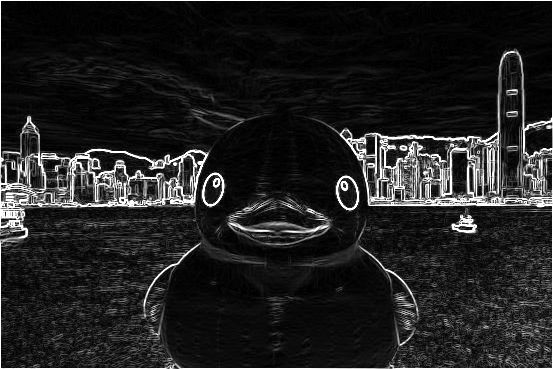
**Vertical-edge finding masks**



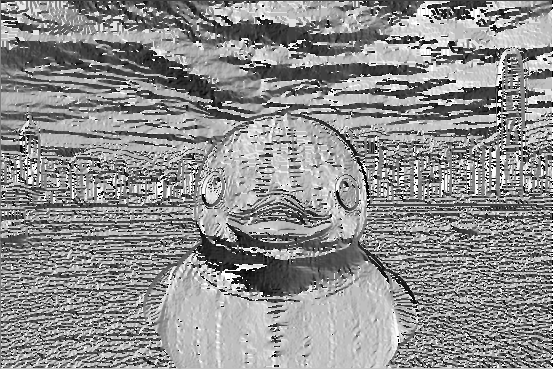
**Sum**

****

**Raw magnitude of the gradient**

****

**Raw direction of the gradient**

****

**Directions**

****

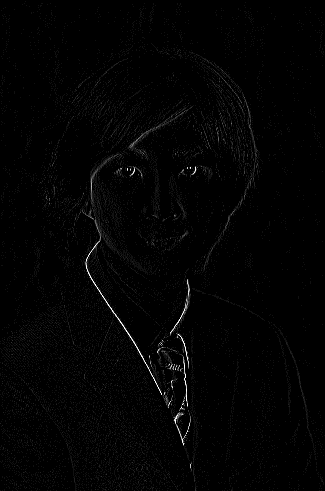
**We applied our sobel function in another pic.**

**Original image**

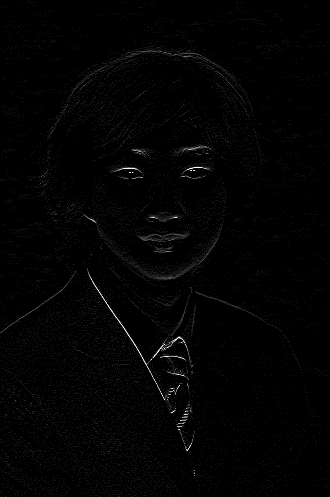
****

Credit to Jizhou Huang

**Horizontal-edge finding masks**

****

**Vertical-edge finding masks**

****

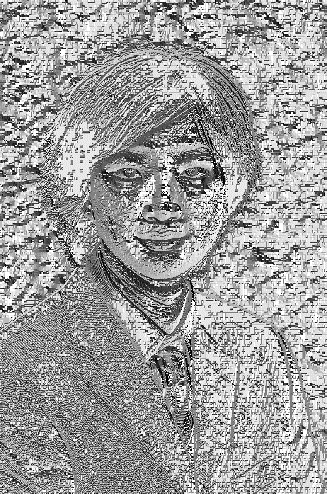
**Sum**

****

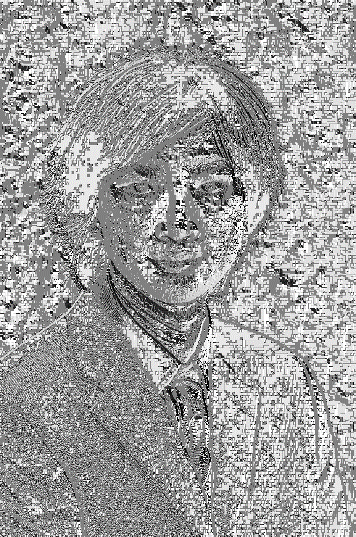
**Raw magnitude of the gradient**

****

**Raw direction of the gradient**

****

**Directions**

****