Scribing Feb 5

Olivia Amaral

February 2024

1 CMSC 491 Lecture 3: Image Filtering

This document contains detailed notes from the image filtering lecture, refer to slides as well for even more context when following along with this scribing

2 Recap

- An image is a matrix of pixels

2.1 Subsampling

- To do subsampling, we look at every other row
- Result is grainy because rows and columns are being skipped
- Each pixel looks bigger and zoomed in even though it is actually a smaller image

2.2 Reviewing The Extra Credit

- For every pixel (x,y), f(x,y) there is F(H-x, y)
- Professor showed a Google Colab of what it would look like, with the code included on slide 8.Practice for point processing filters, and these are the steps taken:
  - Separate the red, green, and blue color channels. Then multiply by the X matrices to create red, green, and blue UMBC channels
  - Use flip function
  - Combine blue UMBC and inverted green UMBC image. This is possible because the bottom region pixels are black, which is equal to a value of 0
  - Right shift the white X by 10 pixels
  - Underlay red X below the blue

- We get the shadow by doing rightshift - (x * rightshift)
- Note that white = 1 and black = 0, so it looks like black with a white shadow. Then change white to red and put the blue UMBC where the black is
- Lastly add the two images together that include the flipped X and the X with a shadow

3 Box Filters

- The equation says that for each square, multiply by 0. The value gets assigned to the central pixel
- To apply the box filter, we iterate through the entire image and fill out the values
- On the visuals attached to the slides, the values are shown when stepping through. Also, f represents the input and h represents the output
- A box filter averages an image, by replacing pixel with the average of the neighborhood. This achieves smoothing and removes sharp edges.
- Reminder/Review: Gaussian filter smooths and retains more of the sharp features in comparison to a box filter

4 Sharpening

4.1 Edge Detection

- Edges: incredibly useful image features. As an example, color can be removed from an image, but edges can give enough context that someone looking at the image can still tell what it is
- Image edges are shape information and shape leads to semantic info, such as what the image is. ex: person? cat?
- Edges are locations of rapid change in image intensity
- WHite is the highest brightness and black is the lowest
- Low intensity to high intensity quickly would be an edge. Compute this by doing a first order derivative

4.2 Gradients

- Computed 1st derivative from gray scale imags, where the borders are edges/very sharp changes in intensity
- An edge is computed in the horizontal direction

- There are horizontal, vertical, and 45-degree angle computation of gradients
- Horizontal and vertical gradients showed on slide, where it is moving inside pixels and getting stronger vertical edges or stronger horizontal edges

5 Edge Detection Via Convolution

- Stepped through an example in class of using the popular edge detection filter, known as the Sobel filter
- White box is 0 for no change, and outside the box would either be positive or negative changes. The values represent edges
- In the example, there are some 2s in the matrix. The reason for this is to put more weight into the left and right neighbors when replacing the central value
- There exists a vertical Sobel filter and a horizontal Sobel filter. The two can be combines to count bricks for example in the image of the wall. It adds more detail because more edges are shown

5.1 Gaussian Filter Revisited (Smoothing / Blurring)

- Sigma = 1 is a standard deviation, only pixels in that 1 pixel neighborhood will be averaged. So, if the standard deviation is equal to 1, there won't be any difference to the image. The larger the standard deviation, the wider area that is being averaged, so the image appears to be have much less detail and is blurry
- In the visual of what each standard deviation filter is, the black = 0 and the white = 1

6 Sharpening filter

- High-pass filtering: Smooths the image by applying a Gaussian filter, then just extracts the edges out of the image. Then, the image of the extracted edges is combined with the original image to create a sharpened result
- When an image is blurred, the low frequencies are preserved and the high frequencies are rejected
- First rule is to isolate edges/high frequency and second rule is to increase pixel intensity
  - On the sharpening filter slide (slide 53), the scaled impulse image is representing the 2-dimensional identity matrix
  - Small patches of color are about the same intensity

7 Thresholding

- Thresholding makes storage efficient. It depends on how much more efficient because of variation, but we can say for now about eight times more efficient
- Thresholding is a simple operation where we can assume the highest is 255 and the lowest is 0
- set a to a value, 100 for example. Then any pixel with a value greater than or equal to 100 would be set to 255. Anything less than that would be set to 0.

8 Fourier Domain Filtering

- Fourier Domain Filtering also known as frequency domain. Of a 1 dimensional signal images are 2 dimensional
- Converts signal into its constituent frequency
- For every point on the x-axis, there is some corresponding intensity
- To do this, we add 2 sine waves with different frequencies and amplitudes. A higher intensity sign wave is added to one with a lower intensity
- To generate the square wave, smaller and higher frequency waves are added to the output. By summing up all of the sine waves. Need to find the right frequency as well, not just any random sine waves
- In the equation, the first parameter k is frequency and the other is magnitude

8.1 Fourier Series

- Fourier claims that any signal can be represented by sine waves, and that his formula (slide 65) can be used to get any wave in the world
- Phase of a sine wave is the infinite sum

8.2 Visualizing the frequency spectrum

- 1 dimensional is easy, but 2 dimensional we now have frequencies in X and in Y
- Frequency is horizontal change in intensity
- The signal average for a sine wave with no offset is 0

- Examples show three dots, which represent the frequency spectrum. The central dot is the frequency at 0, then the others represent positive frequency and negative frequency. The outer dots are symmetric to the center one

- Frequency will be in the direction that the intensity changes in

- Can use the frequency spectrum and combine to get both frequencies, similar to combining sine waves from earlier

9 Fourier Transform

- Connection between temp and signals

- Sum over multiple sine waves

- Amplitude and phase spectrum

- Lots of dots combined, looking at the images on the slide of Voldemort and Meg Ryan...

  - Voldemort is lower frequency picture
  - Meg Ryan is higher frequency picture

- Inverse Fourier Transform shows that texture related details are stored

10 Convolution Theorem

- Note the definition on the slides

- Fourier transform of convolution of two functions is the product of their Fourier transforms

- Inverse Fourier transform of product of two Fourier transforms is convolution of the two inverse Fourier transforms

- Convolution is spational domain = multiplication in frequency domain *Important

10.1 Low Pass

- Only low frequency can pass through

- Fourier transfer of a=image and Gaussian filter to get blurred picture, element wise multiplication

- 0's of the Gaussian filter get multiplied

- Center is 0 frequency, low pass because output contains low frequency

- Instead of moving filters across the image, just did two Fourier transforms and multiplies by the number of pixels there are, which is much more efficient

10.2 High Pass

- Allows high frequency, also known as edges, to pass through
- Color is gone, now just see the edges

10.3 Bandpass

- Combination of low and high pass filters, Same thing as bandwidth and can be restricted to certain locations of a picture

11 Summary

- Converting an image to frequency domain is much more efficient and is also easier to extract filters, make design choices in the frequency domain, and to control the output you get