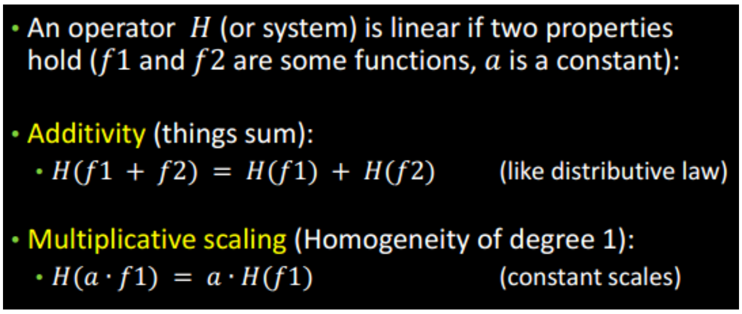
2A-L3 Linearity and convolution

2017/11/04 00:55

1. Sum
    a. linearity
        i. additivity & multiplicative scaling
    b. impulse, impulse response
    c. convolution
        i. unlike correlation, it rotates the img 180 degree
        ii. property
            1. linear and shift invariant
            2. commutative
            3. associative
            4. identity
            5. differentiation
        iii. seperability to reduce the computation
    d. edge issue
        i. choose replicate or symmetric, which keeps the original statistics
        ii. matlab
            1. imfilter(img, filter, 'type')
    e. noise
        i. independent, mean zero noise
            1. guassian or box average
            2. smooth the edge
        ii. spike, totally random noise
            1. median filter
                a. code
                    i. noiseimg = imnoise(img, 'salt & pepper', 0.02)
                    ii. medimg = medfilt2(noiseimg)
                b. perserve the edge

2. linear intuition
    a. Additivity & Multiplicative scaling



- An operator H (or system) is linear if two properties hold ($f1$ and $f2$ are some functions, a is a constant):
- Additivity (things sum):
  - $H(f1 + f2) = H(f1) + H(f2)$ (like distributive law)
- Multiplicative scaling (Homogeneity of degree 1):
  - $H(a \cdot f1) = a \cdot H(f1)$ (constant scales)

b. Because it is sums and multiplies, **the “filtering” operation we were doing is linear.**

c. What linearity is going to allow us to do is to build up a signal or a function, i.e. an image, a piece at a time and then be able to say how a linear operator affects that whole image.

1. Impulse Function and Response

a. it’s a building block function

b. representation

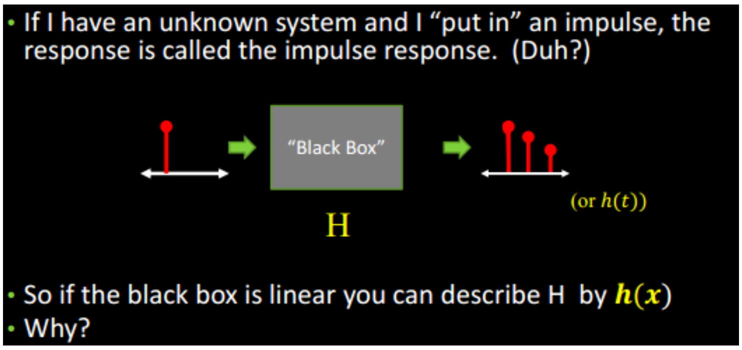
i. In the discrete world, it’s just a value of 1 at a single location.

1. this is what concerned

ii. • In the continuous world, an impulse is an idealized function that is very narrow and very tall so that it has a unit area.

c. An impulse response

i. the response outputed from a system with an impluse input is called the impulse response



- If I have an unknown system and I “put in” an impulse, the response is called the impulse response. (Duh?)

- So if the black box is linear you can describe H by $h(x)$
- Why?

d. the cool thing

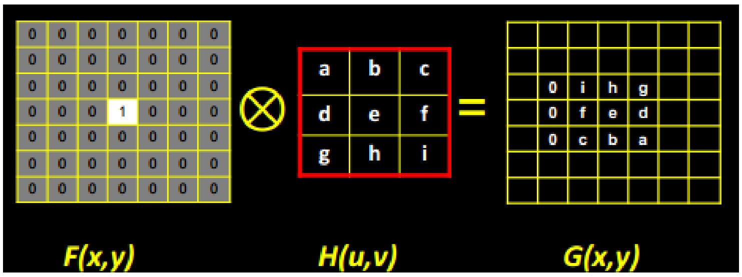
i. if we know the impulse response and the impulse, we can discribe what is the operation of H(x). Since:

1. any sequence of pulses can be described by just adding in a shifted set and scaled set of those single impulses.

2. If I know how this black box effects just the single impulse. I’ll be able to say how it affects the entire image.

1. Filtering an Impulse Signal

a. it flips the original image upside down and leftside to right



b. normally, the center, e, is called reference point.

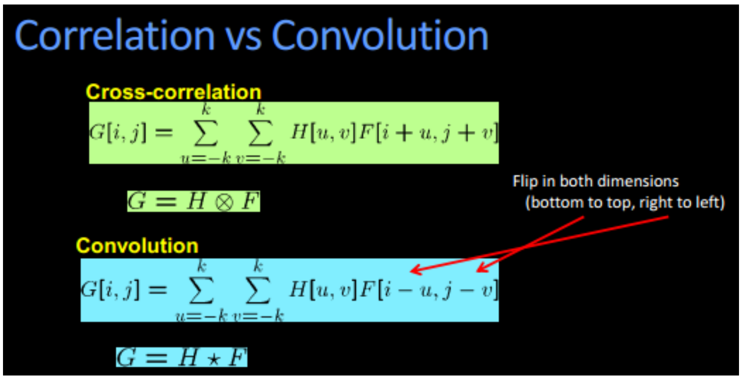
1. Quiz: Kernel Quiz

a. Suppose our kernel was size MxM and our image was NxN. How many multiplies would it take to filter the who image with the filter?

i. MxM x NxN

2. Correlation vs Convolution

a. Now we have a less pleasant thing that the impulse response is fliped upside down and leftside right. So we introduce another filter Convolution, which rotates itself 180° before applying the filtering.



Correlation vs Convolution

Cross-correlation

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

$G = H \otimes F$

Flip in both dimensions (bottom to top, right to left)

Convolution

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

$G = H \star F$

b. For a Gaussian or box filter, how will the outputs differ?

i. no, both the correlation and convolution work the same with circular symmetric filters, e.g. Gaussian

1. Quiz: Convolution Quiz

a. When convolving a filter with an impulse image, we get the filter back as a result. So if we convolve an image with an impulse we get:

i. The original image

ii. Since digital images and filters are both essentially matrices, so they behave in the same manner.

2. Properties of Convolution

a. shift invariance

i. means the operator behaves everywhere the same way, i.e. the value of the output depends on the pattern in the image neighborhood, not the position of the neighborhood.

b. convolution and correlation are built in multiplication and addition which are linear operators, so these filters are linear filters

c. properties

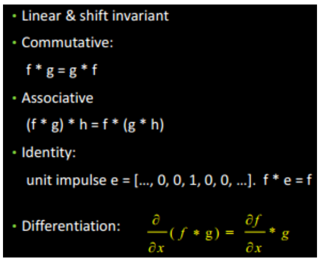
i. linear & shift invariant

ii. commutative

iii. associative

iv. identity

v. differentiation



- Linear & shift invariant
- Commutative:

  f * g = g * f
- Associative

  (f * g) * h = f * (g * h)
- Identity:

  unit impulse e = [..., 0, 0, 1, 0, 0, ...]. f * e = f
- Differentiation: $\frac{\partial}{\partial x}(f * g) = \frac{\partial f}{\partial x} * g$

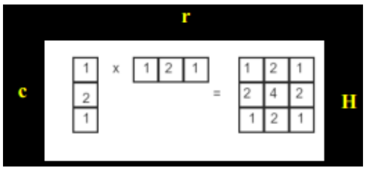
9. Computational Complexity and Separability

   a. Previously, if an image is NxN and a kernel (filter) is WxW, how many multiplies do you need to compute a convolution?

      i. N*N*W*W

   b. Separability

      i. • In some cases, filter is separable, meaning you can get the square kernel H by convolving a single column vector by some row vector:



   c. Now, with separability, G = H * F = (C * R) * F = C * (R * F)

      i. 2*W*N*N is enough, which used to be very important

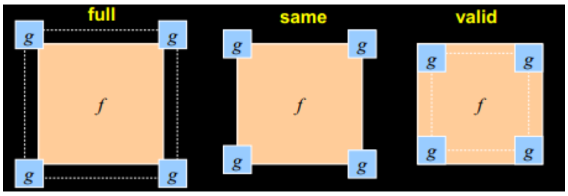
         1. it's more efficient when W > 2

      ii. so we prefer to use separable filters.

1. Boundary Issues

   a. the size of the output

      i. with Matlab we have 3 choice: full, same, valid. We prefer the same.



   b. Methods

      i. 0 clip filter: regard the outside is Black,

         1. so the image would be affected with the black leakage

      ii. **circular** /wrap around: treat the image as periodic signal, so, e.g., the right edge has the elements from the left.

         1. would affect the image

      iii. **replicate** /copy edge: just the copy the content on the same edge

         1. this won't affect the original image

      iv. **symmetric** /reflect across edge: take the reflection

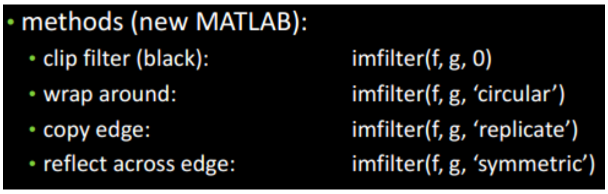
         1. this won't affect the statistic of the

original image

v. In order not to affect the image, choose copy or reflect

1. The created imagery has the same statistics as the original image
2. the distribution of color or intensity values is not likely to change much across a small region, so padding with pixel values along the boundary is a good approximation of what might've actually been there if the image was larger.
3. Also, the reflection method avoids introducing a possibly hard boundary.
4. here no statistical change = no hard boundary

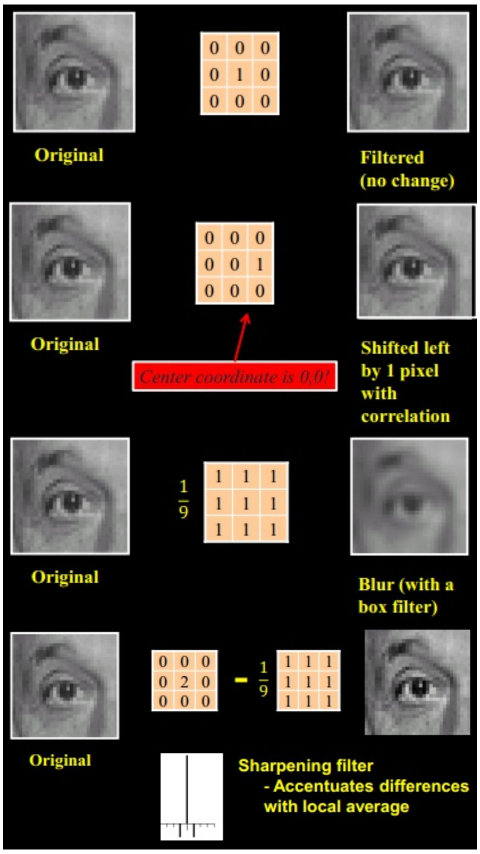
vi. matlab



- methods (new MATLAB):
  - clip filter (black): imfilter(f, g, 0)
  - wrap around: imfilter(f, g, 'circular')
  - copy edge: imfilter(f, g, 'replicate')
  - reflect across edge: imfilter(f, g, 'symmetric')

11. Practicing with Linear Filters

a. The current filters' effect



b. Quiz

i. the weights can be rescaled before or after the filtering, since it's linear operations.

1. Different Kinds of Noise

a. what method should be used to remove the specific kind of Noise?

i. noise is as sort of arbitrary value i.e. **if the noise was independent at each pixel and centered about zero**

1. How: replace the value of on pixel by the local average

a. box average or Gaussian average

ii. noise as totally random values, e.g. salt and pepper noise

1. median

2. Median Filter

a. AD

i. median filter just replace the center value with some other value around, which means **no weird value is introduced.**

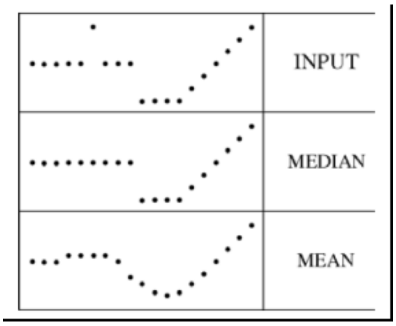
ii. remove spikes

1. so it's good for the weird spike noise, e.g. salt and pepper noise

b. Not linear!!!

c. VS. mean filter

i. median filter preserve the edge, while mean filter blur the edge



14. Matlab

a. salt & pepper noise

i. imgnoise = imnoise(img, 'salt & pepper', 0.02)

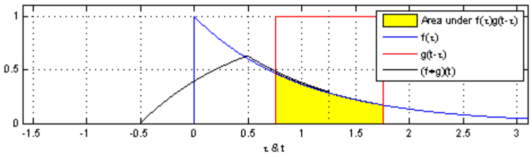
b. median filter

i. med = medfilt2(imgnoise)

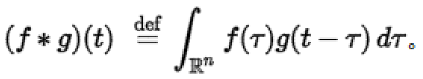
15. Convolution

a. definition

i. 表征函数f与经过翻转和平移的g的乘积函数所围成的的曲边梯形的面积。

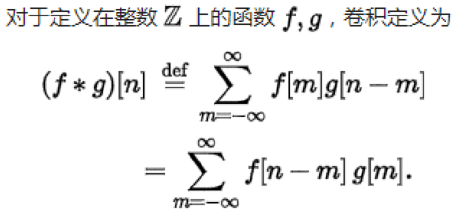


ii. 定义在R上



$$(f * g)(t) \overset{\text{def}}{=} \int_{\mathbb{R}^n} f(\tau) g(t-\tau)\, d\tau。$$

iii. 定义在Z上



对于定义在整数 $\mathbb{Z}$ 上的函数 f, g，卷积定义为

$$\begin{aligned}(f * g)[n] &\overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m] g[n-m] \\ &= \sum_{m=-\infty}^{\infty} f[n-m]\, g[m].\end{aligned}$$

b. 计算方法

i. 快速傅里叶变换， 时域卷积可变换成频域乘积，计算后再回到时域。共作2次DFT和1次IDFT。

c. Reference

i. https://zh.wikipedia.org/wiki/%E5%8D%B7%E7%A7%AF

ii. https://www.zhihu.com/question/22298352