Linear Image Filtering

C306

Martin Jagersand

Filtering

- Filtering can be expressed as a convolution in spatial coordinates, and a pointwise multiplication in frequency coordinates.
- Low pass filtering can suppress noise.
- High pass filtering can enhance detail.

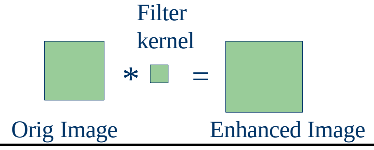
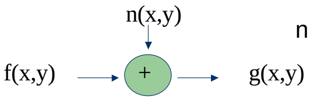
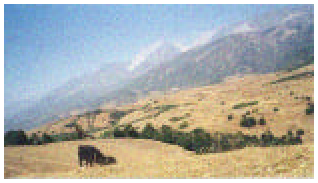


Image noise



n = Additative noise term

n due to e.g. electrical noise in video camera



Noise Models

- Noise is commonly modeled using the notion of “additive white noise.”

  I(i,j,t) = I*(i,j,t) + n(i,j,t) where

  n(i,j,t) is independent of n(i’,j’,t’) unless i’=i,j’=j,t’=t.

- Properties of temporal image noise:

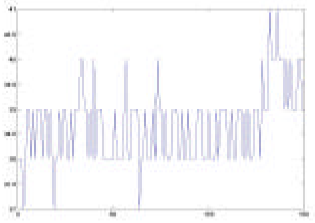
Mean $\mu = \sum_{t} I(i,j,t)$

Standard Deviation $\sigma = \mathrm{Sqrt}(\sum_{t} (\mu - I(i,j,t))^2)$

Signal-to-noise Ratio $\frac{\quad}{\sigma_{\sigma}}$

Image Noise

- An experiment: take several images of a static scene and look at the pixel values



mean = 153.6
std = 2.99

snr = 153/3
max snr = 255/3 = 85

Noise reduction by averaging

PROPERTIES OF TEMPORAL IMAGE NOISE (i.e., successive images)

If standard deviation of grey values at a pixel is σ for a pixel for a single image, then the laws of statistics states that for independent sampling of grey values, for a temporal average of n images, the standard deviation is: $\frac{\sigma}{\mathrm{Sqrt}(n)}$

How to reduce noise

- Averaging is a common way to reduce noise
  - instead of temporal averaging, how about spatial
- For a pixel in image **I** at i,j

$$I'(i,j) = 1/9 \sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} I(i',j')$$

Low pass filtering

- More generally: LP is spatial averaging:

$$\bar{I}(m;n) = \sum_{(k;l)\in W} a(k;l)\, I(m-k)(n-l)$$

- But this is just the convolution with a mask A

$$\bar{I} = I * T; \quad \text{for } T = \begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}; \text{e.g. } T = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

– Note convolution is

- Associative
- Commutative
- Linear

Low pass filtering

- Convolution with a mask A

$$\bar{I} = I\ \tilde{a}\ T;\ \ \text{for}\ \ T = 4\begin{matrix} a_{11} & \acute{a}\acute{a}\acute{a} & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{m1} & \acute{a}\acute{a}\acute{a} & a_{mn} \end{matrix}5;\ e{:}g\ \ T = \hat{o}\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}\tilde{o}$$

–Note convolution is

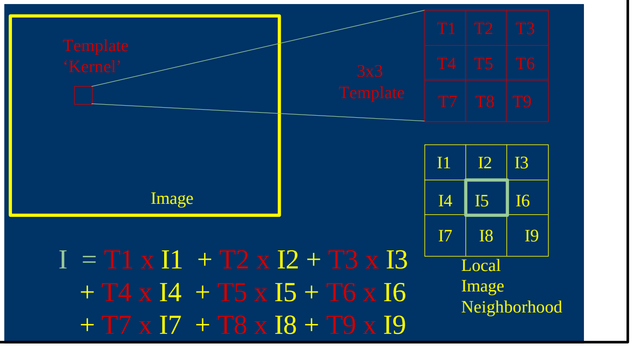
- •Associative
- •Commutative
- •Linear

•Noise properties:

Variance: $\hat{u}_{filt}\ \grave{u}\ p\frac{1}{mn}\hat{u}_{orig}$

For m$\times$n image filter mask

Convolution graphically



T1	T2	T3
T4	T5	T6
T7	T8	T9

I1	I2	I3
I4	I5	I6
I7	I8	I9

$$\begin{aligned} I &= T1 \times I1 + T2 \times I2 + T3 \times I3 \\ &+ T4 \times I4 + T5 \times I5 + T6 \times I6 \\ &+ T7 \times I7 + T8 \times I8 + T9 \times I9 \end{aligned}$$

Frequency domain

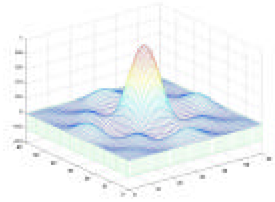
- Taking the Fourier transform and using the convolution theorem:

$$=(\bar{I}) = =(I\ \tilde{a}\ T) = =(I)=(T)$$

- In particular, if we look at the power spectrum, then we see that convolving image I by T attenuates frequencies where T has low power, and amplifies those which have high power.

The Properties of the Box Filter

F(mean filter) =



Thus, the mean filter enhances low frequencies
but also has “side lobes” that admit higher frequencies

The Gaussian Filter: A Better Noise Reducer

- Ideally, we would like an averaging filter that removes (or at least attenuates) high frequencies beyond a given range

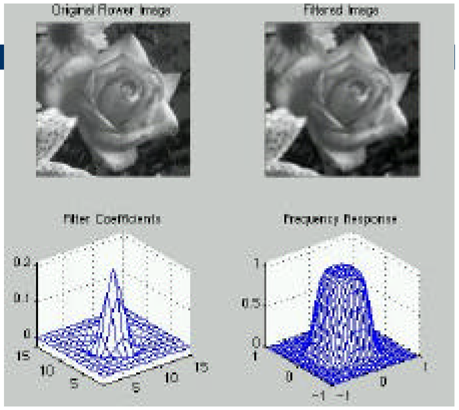
$$g(i, j, s) = e^{-\left(i^2+j^2\right)/2s^2}$$

- It is not hard to show that the FT of a Gaussian is again a Gaussian. Hence, it operates as a low pass filter.

Computational Issues: Separability

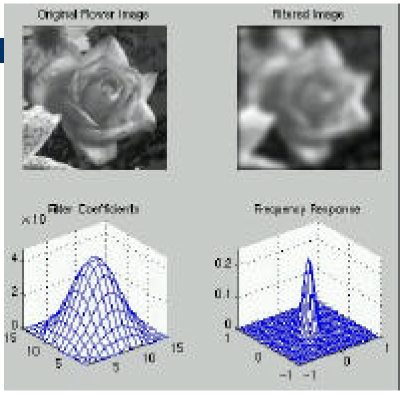
- Recall that convolution is commutative. Suppose I use the templates gx = exp(-i^2/2 σ^2) and
  By = gy = exp(-j^2/2 σ^2) Then
  - gx * (gy * I) = (gx * gy) * I
  - but, it is not hard to show that the first convolution is simply the 2-D Gaussian that we defined previously!
- In general, this means that we can “separate” the 2-D Gaussian convolution into 2 1-D convolutions with great computational cost savings.
- A good exercise is to show that the box filter is also separable.

Example: Moderate smoothing



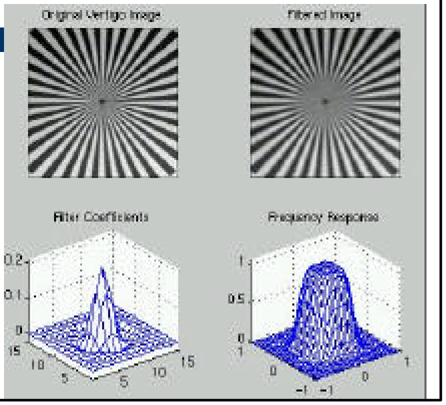
Ex2: More smoothing

- Note: Details disappear
- Note2: Scaling property of F-t



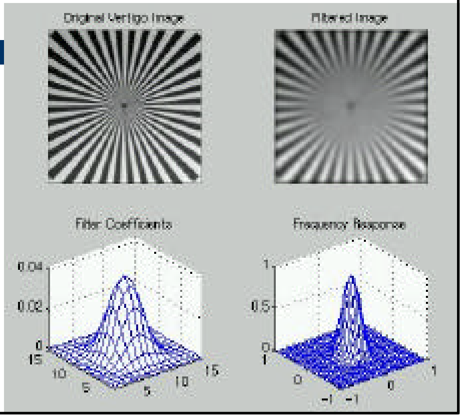
Ex3 Frequency diagram

- Can visually quantify "blurring"



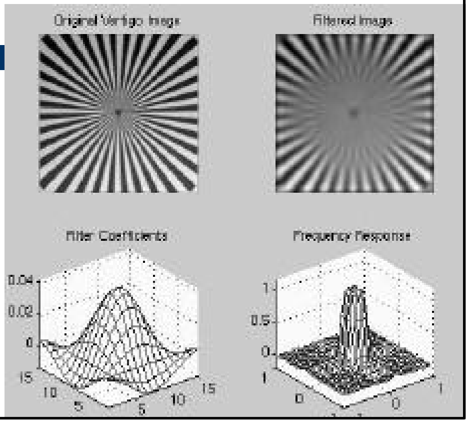
More "blurring"

- Note that in general, we truncate the filter mask
- a good general rule is that the width (w) of the filter is at least such that $w > 5\sigma$.



Filter with "ringing"

- Notice sharp frequency cutoff
- Causes ringing in spatial domain
- And odd looking image



In practice: smooth masks

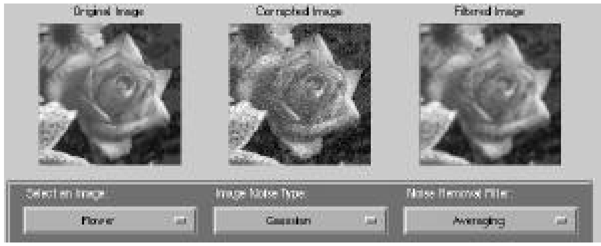
- Avoids ringing!
- 2D Gaussian:

$$G(u;v) = \frac{1}{2ùû^2}\exp\left(\frac{àu^2àv^2}{4û^2}\right)$$

- Butterworth:

$$B(\mathbf{u};\mathbf{v}) = \frac{1}{1+[(\mathbf{u}^{2+}\mathbf{v}^2)=\mathbf{d}]^{\mathbf{n}}}$$

Noise reduction



Select an Image: Flower

Image Noise Type: Gaussian

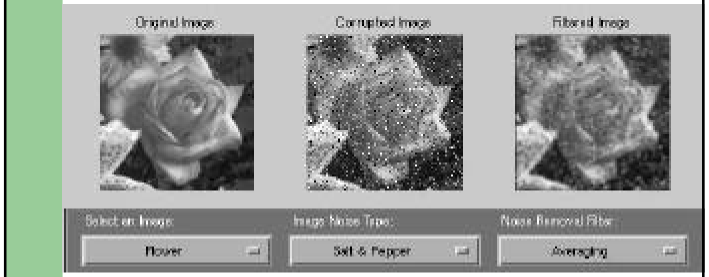
Noise Removal Filter: Averaging

- Only Gaussian high frequency noise is effectively reduced by low pass filtering

Other Types of Noise

- Shot noise (also called salt and pepper noise)
- Quantization effects
  - Often called noise although it is not statistical
- Unanticipated image structures
  - Also often called noise although it is a real repeatable signal.

Digital noise



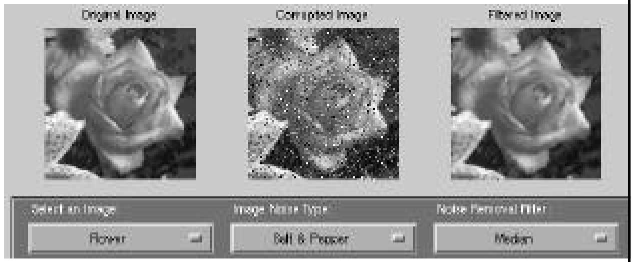
Select an Image: Flower

Image Noise Type: Salt & Pepper

Noise Removal Filter: Averaging

- Linear average or Gaussian LP filter not so good at removing this

Solution: Median filter

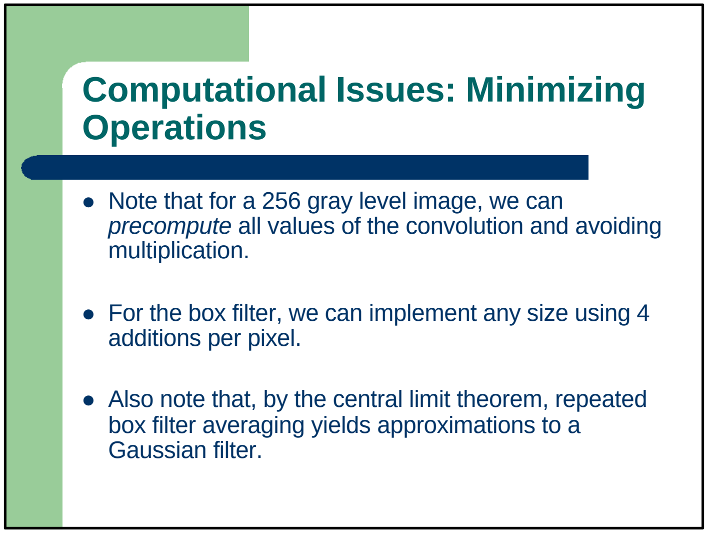


Select an Image: Flower

Image Noise Type: Salt & Pepper

Noise Removal Filter: Median

- $I_{out}(x,y) = median(I_{in}(x-k,y-l),-s<k,l<s)$



Computational Issues: Minimizing Operations

- Note that for a 256 gray level image, we can *precompute* all values of the convolution and avoiding multiplication.
- For the box filter, we can implement any size using 4 additions per pixel.
- Also note that, by the central limit theorem, repeated box filter averaging yields approximations to a Gaussian filter.