Linear filtering

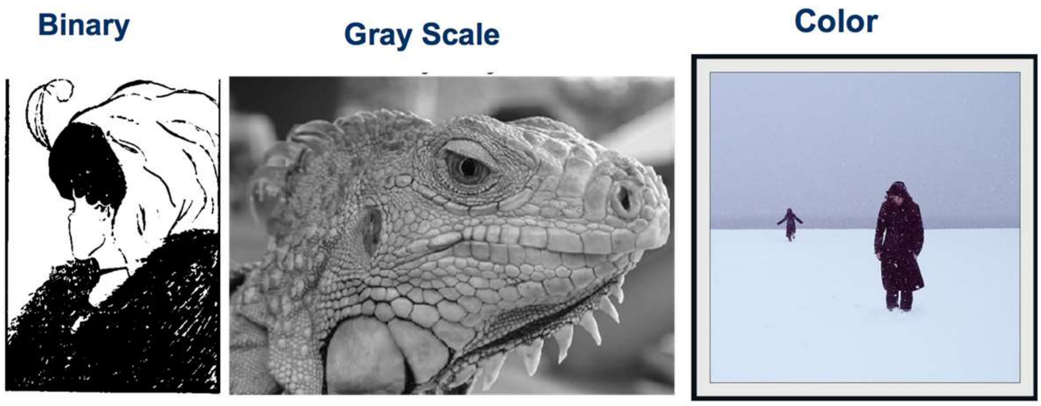
1

Roadmap

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM

2

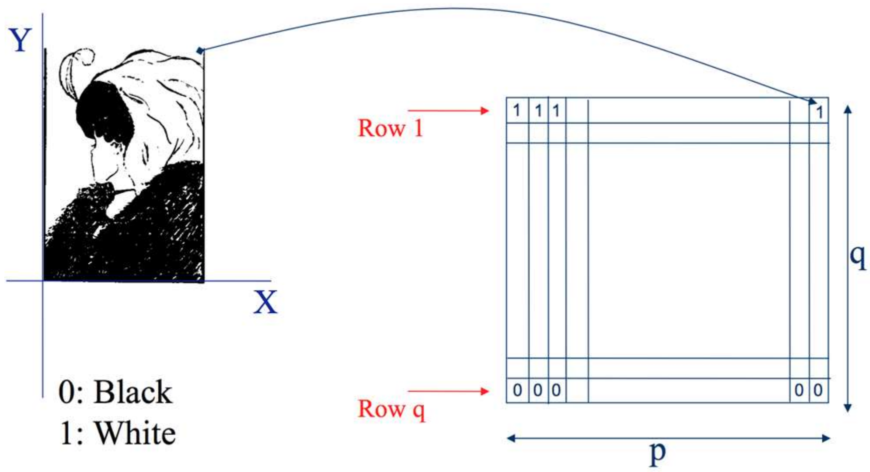
Types of Images



Source: Ulas Bagci

3

Binary image representation

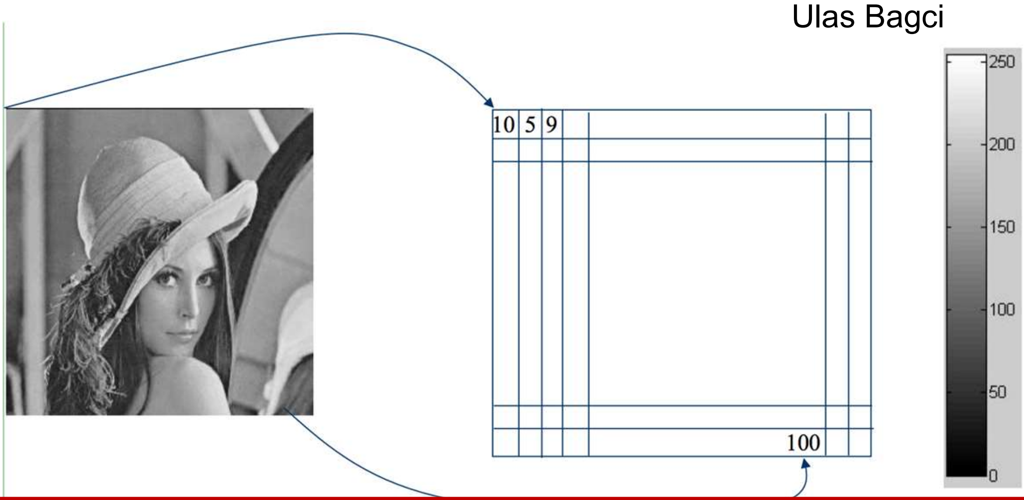


Source: Ulas Bagci

4

Grayscale image representation

Slide credit:
Ulas Bagci

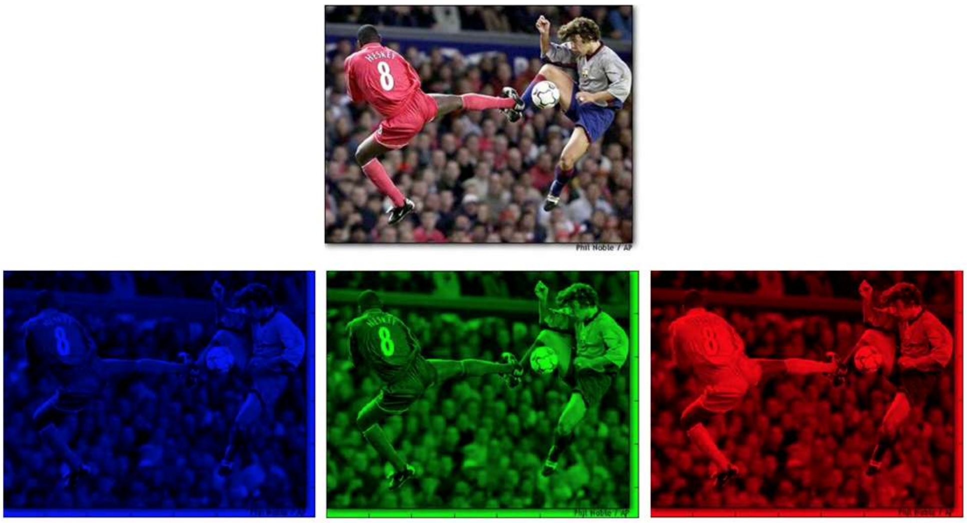


Source：Ulas Bagci

Color Image - one channel

Source：Ulas Bagci

Color image representation



Source: Ulas Bagci

7

Motivation: Image denoising 为处理图像噪声 --> 卷积

- How can we reduce noise in a photograph?

噪声点的像素和周围点
的像素存在较大差异

Source: S. Lazebnik

8

Moving average

- Let's replace each pixel with a *weighted* average of its neighborhood
- The weights are called the *filter kernel* 卷积核/滤波核：权值赋予
- What are the weights for the average of a 3x3 neighborhood?

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

"box filter"

Source：S. Lazebnik

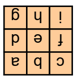
Defining convolution

- Let f be the image and g be the kernel. The output of convolving f with g is denoted $f * g$.

(m, n）表示卷积核中心值

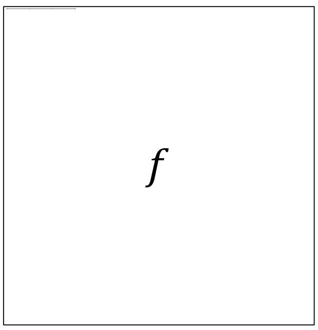
$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$

真正的卷积：从数学定义上看，是需要将卷积核翻转后，再进行运算的，如下图所示。



Convention:
kernel is "flipped"

深度学习中使用的卷积操作则没有进行翻转



Key properties

- **Linearity**: filter($f_1 + f_2$) = filter(f_1) + filter(f_2)

具有平移不变性
- **Shift invariance**: same behavior regardless of pixel location: filter(shift(f)) = shift(filter(f))
- Theoretical result: any linear shift-invariant operator can be represented as a convolution

Source： S. Lazebnik

Properties in more detail

交换律
- Commutative: $a * b = b * a$
  - Conceptually no difference between filter and signal

结合律
- Associative: $a * (b * c) = (a * b) * c$
  - Often apply several filters one after another: $(((a * b_1) * b_2) * b_3)$
  - This is equivalent to applying one filter: $a * (b_1 * b_2 * b_3)$

分配律
- Distributes over addition: $a * (b + c) = (a * b) + (a * c)$

实数乘法
- Scalars factor out: $ka * b = a * kb = k (a * b)$
- Identity: unit impulse $e = [\ldots, 0, 0, 1, 0, 0, \ldots]$, $a * e = a$

如果一个信号和脉冲向量进行卷积会得到信号本身

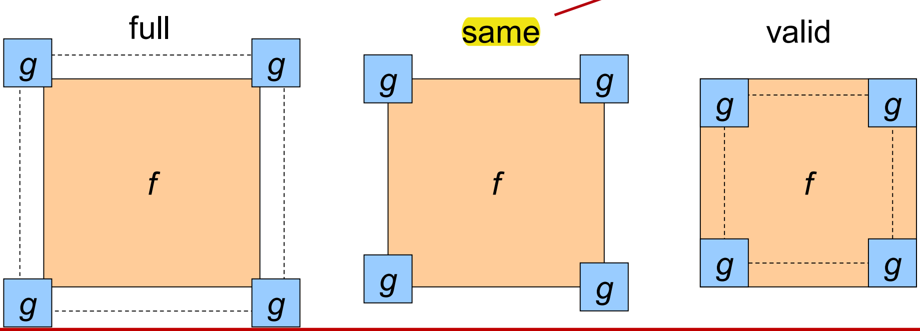
Source： S. Lazebnik

Annoying details

What is the size of the output?

- MATLAB: filter2(g, f, *shape*)
  - *shape* = 'full': output size is sum of sizes of f and g
  - *shape* = 'same': output size is same as f
  - *shape* = 'valid': output size is difference of sizes of f and g

填充处理



Source： S. Lazebnik

Annoying details

What about near the edge?

- the filter window falls off the edge of the image
- need to extrapolate
- methods:
  - clip filter (black)
  - wrap around
  - copy edge
  - reflect across edge

Source: S. Marschner

Annoying details

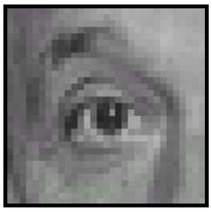
What about near the edge?

- the filter window falls off the edge of the image
- need to extrapolate
- methods (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')

- 填补一圈0，深度学习常用
- 桶状围绕式填充
- 复制填充，直接将最外层像素拉伸
- 镜像填充

Source: S. Marschner

Practice with linear filters



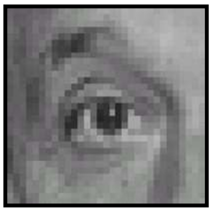
Original

0	0	0
0	1	0
0	0	0

?

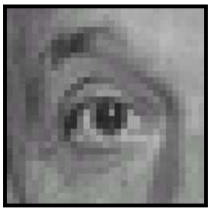
Source: D. Lowe

Practice with linear filters



Original

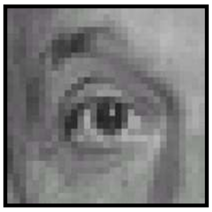
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters



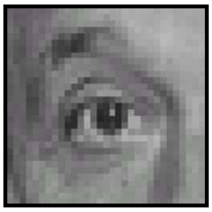
Original

0	0	0
0	0	1
0	0	0

?

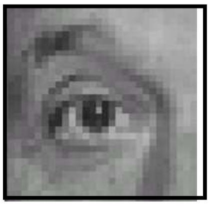
Source: D. Lowe

Practice with linear filters



Original

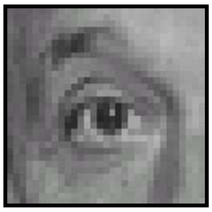
0	0	0
0	0	1
0	0	0



Shifted *left*
By 1 pixel

Source: D. Lowe

Practice with linear filters



Original

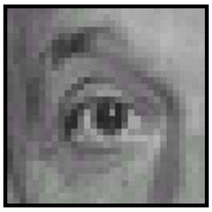
$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

?

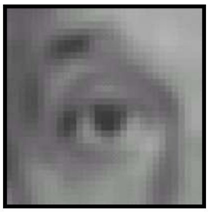
Source: D. Lowe

Practice with linear filters



Original

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

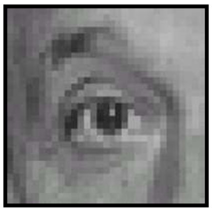


Blur (with a box filter)

图像平滑操作 --> 去噪

Source: D. Lowe

Practice with linear filters



Original

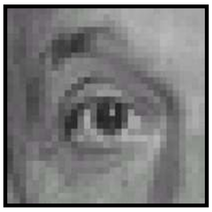
$$\begin{bmatrix}0 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

?

(Note that filter sums to 1)

Source: D. Lowe

Practice with linear filters



Original

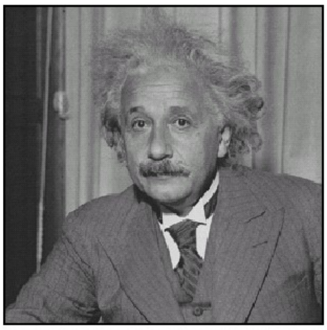
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter 锐化操作 --> 棱角更加分明

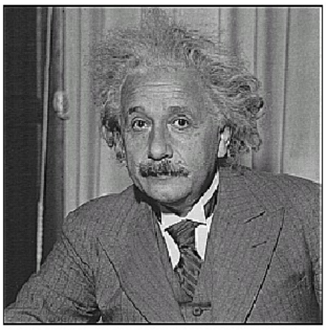
- Accentuates differences with local average

Source: D. Lowe

Sharpening



before



after

Source: D. Lowe

Sharpening

What does blurring take away?

注: I - image, e - 脉冲核（参考Page-12），g - 卷积核（图像平滑）

相当于 I * e - I * g = I * (e - g)

Source: D. Lowe

Sharpening

What does blurring take away?

Let’s add it back:

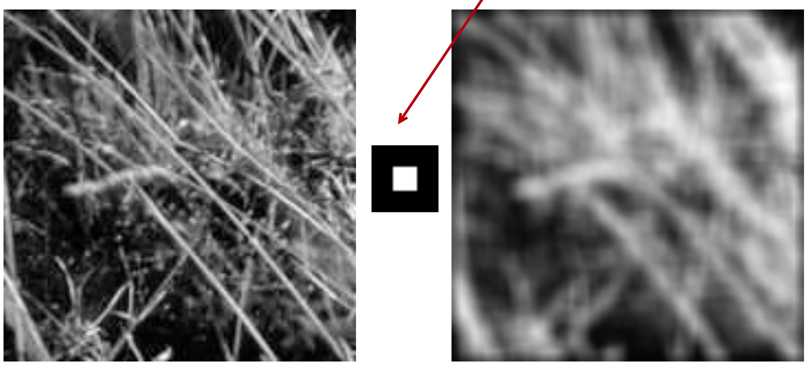
相当于 I * e + I * (e - g) = I * （2e - g）

Source: D. Lowe

Smoothing with box filter revisited

- What’s wrong with this picture?
- What’s the solution?

振铃现象：因为模版是“方的”，即卷积核的各个点权值是一致的。

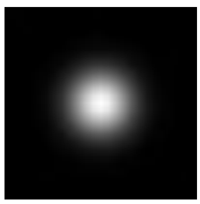


Source: D. Forsyth

27

Smoothing with box filter revisited

- What’s wrong with this picture?
- What’s the solution?
  - To eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center



“fuzzy blob”

Source： S. Lazebnik

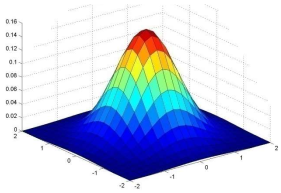
28

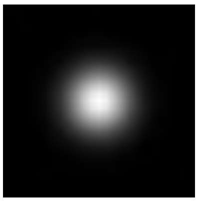
Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

(x, y)

(-2, -2)　(0, 0)　(2, 2)





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, σ = 1

窗宽

确保权值和为1，否则，原图的像素值会出现“衰减现象”

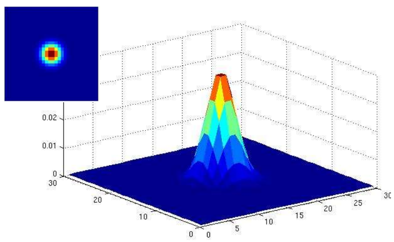
- Constant factor at front makes volume sum to 1 (can be ignored when computing the filter values, as we should renormalize weights to sum to 1 in any case)

Source: C. Rasmussen

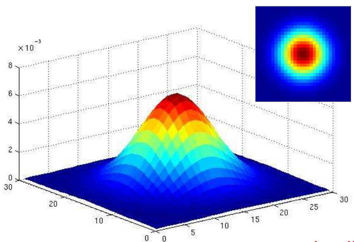
29

Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



σ = 2 with 30 x 30 kernel



σ = 5 with 30 x 30 kernel

标准差越小表示数据越集中，表明中心点的权值越大，说明平滑的程度越小

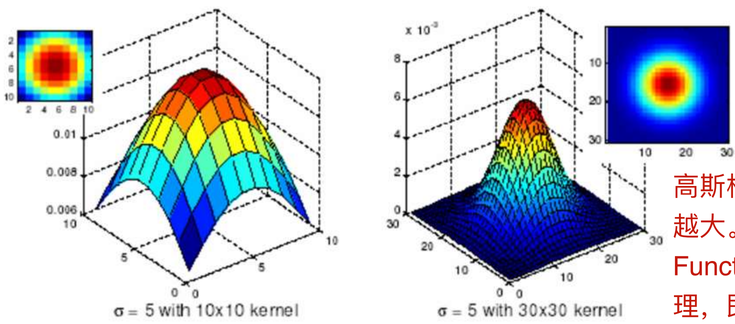
- Standard deviation σ: determines extent of smoothing

Source: K. Grauman

30

Choosing kernel width

- The Gaussian function has infinite support, but discrete filters use finite kernels



高斯核越大，表明平滑程度越大。因为计算了Gaussian Function之后会做归一化处理，即高斯核越大，表明归一化中的分母越大。

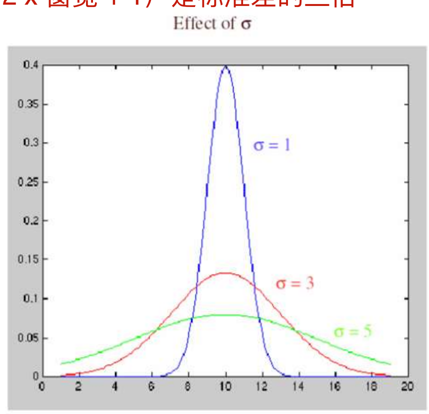
Source: K. Grauman

Choosing kernel width

- Rule of thumb: set filter half-width to about 3σ

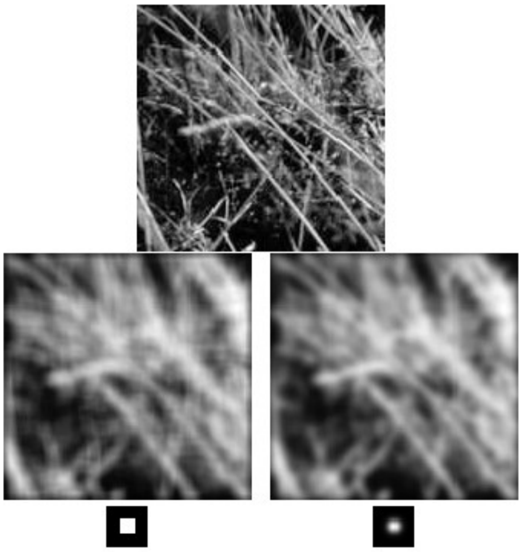
一般规则： 窗口的大小（高斯核中心到边缘的距离，即 kernel - width = 2 x 窗宽 + 1）是标准差的三倍

例如，当标准差为1时，高斯核尺寸为 7 x 7



Source：S. Lazebnik

Gaussian vs. box filtering



Source： S. Lazebnik

33

Gaussian filters

- Remove “high-frequency” components from the image (low-pass filter) 去噪
- Convolution with self is another Gaussian
  - So can smooth with small-σ kernel, repeat, and get same result as larger-σ kernel would have
  - Convolving two times with Gaussian kernel with std. dev. σ is same as convolving once with kernel with std. dev. $\sigma\sqrt{2}$ 用一个大高斯核对图像进行卷积，等价于用两个小高斯核进行卷积
- *Separable* kernel
  - Factors into product of two 1D Gaussians
    高斯核可以分解

Source: K. Grauman

34

Separability of the Gaussian filter

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp^{-\frac{x^2+y^2}{2\sigma^2}}$$
$$= \left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{y^2}{2\sigma^2}}\right)$$

The 2D Gaussian can be expressed as the product of two functions, one a function of x and the other a function of y

In this case, the two functions are the (identical) 1D Gaussian

Source: D. Lowe

Separability example

2D convolution
(center location only)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix}$$

= 2 + 6 + 3 = 11
= 6 + 20 + 10 = 36
= 4 + 8 + 6 = 18

= 65

The filter factors
into a product of 1D
filters:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Perform convolution
along rows:

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} = \boxed{\begin{matrix} 11 \\ 18 \\ 18 \end{matrix}}$$

*

= 1x11 + 1x18 + 1x18 = 65

= 1x2 + 2x3 + 1x3 = 11
= 1x3 + 2x5 + 1x5 = 18
= 1x4 + 2x4 + 1x6 = 18

Followed by convolution
along the remaining column:

Source: K. Grauman

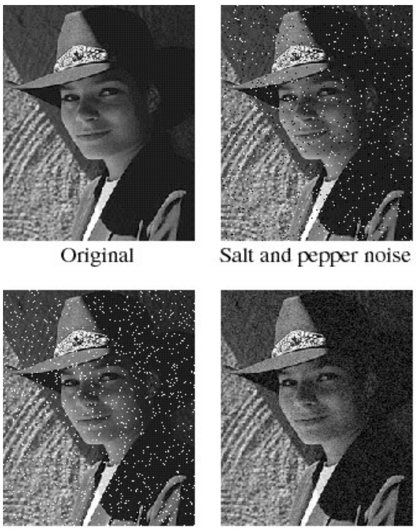
Why is separability useful? 加速运算

- What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?
  - $O(n^2 m^2)$ → 可分离则意味着复杂度变为了 2m
- What if the kernel is separable?
  - $O(n^2 m)$

Source： S. Lazebnik

37

Noise



Original | Salt and pepper noise | Impulse noise | Gaussian noise

- **Salt and pepper noise**: contains random occurrences of black and white pixels 椒盐噪声 — 随机黑白点
- **Impulse noise:** contains random occurrences of white pixels 脉冲噪声 — 随机白点
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution 高斯噪声

Source: S. Seitz

38

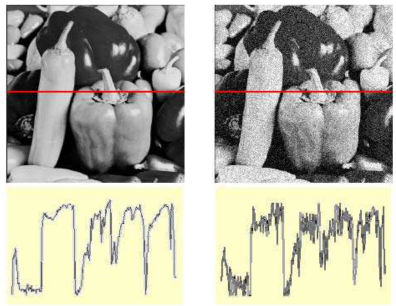
Gaussian noise

- Mathematical model: sum of many independent factors
- Good for small standard deviations
- Assumption: independent, zero-mean noise

高斯噪声是指它的概率密度函数服从高斯分布（即正态分布）的一类噪声

产生原因：
1）图像传感器在拍摄时市场不够明亮、亮度不够均匀；
2）电路各元器件自身噪声和相互影响；
3）图像传感器长期工作，温度过高。

Image
Noise

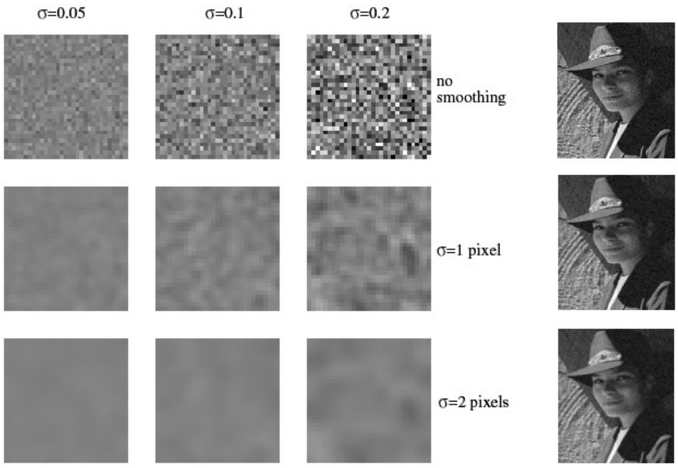


$$f(x,y) = \overbrace{\tilde{f}(x,y)}^{\text{Ideal Image}} + \overbrace{\eta(x,y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise: $\eta(x,y) \sim \mathcal{N}(\mu,\sigma)$

Source: M. Hebert

Reducing Gaussian noise



对于标准差较大的高斯噪声使用的高斯核的标准差相应地也会较大

Smoothing with larger standard deviations suppresses noise, but also blurs the image

Source：S. Lazebnik

Reducing salt-and-pepper noise

3x3

5x5

7x7

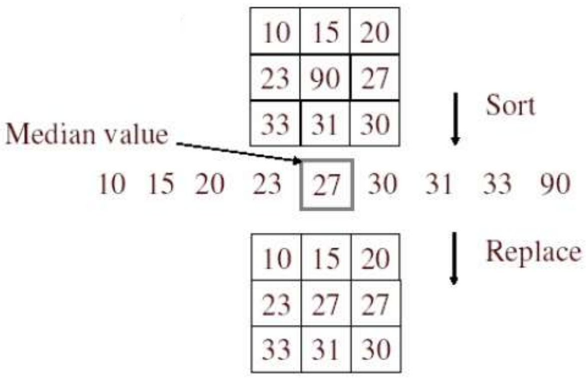
What’s wrong with the results?

高斯滤波器并不能很好的解决“椒盐噪声”

Source：S. Lazebnik

Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window 非线性操作

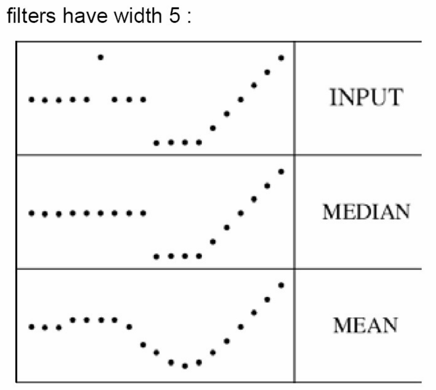


- Is median filtering linear?

Source: K. Grauman

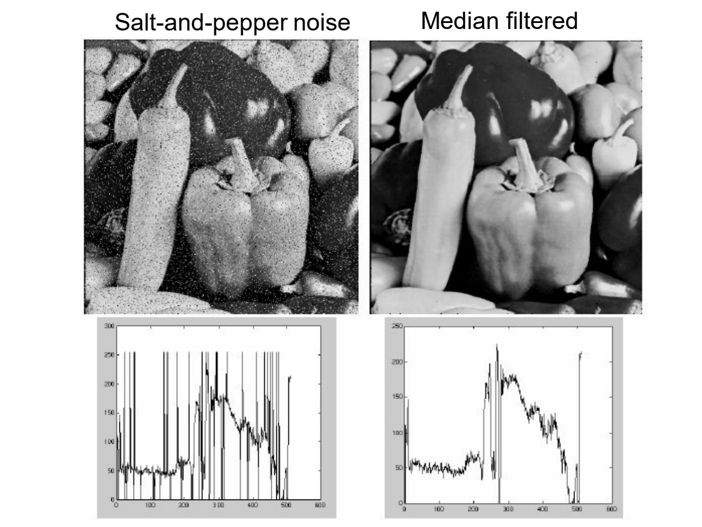
Median filter

- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers



Source: K. Grauman

Median filter



Source: M. Hebert

Gaussian vs. median filtering

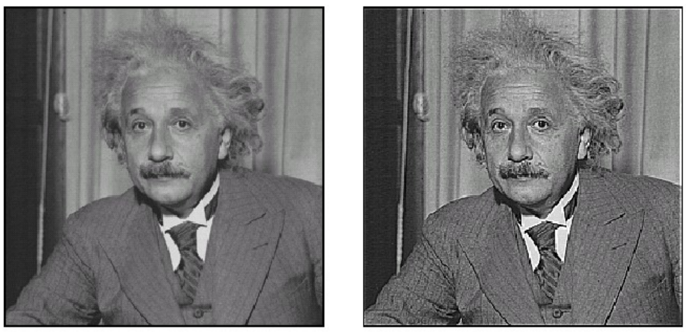
3x3 5x5 7x7

Gaussian

Median

Source: S. Lazebnik

Sharpening revisited



before **after**

Source: D. Lowe

Sharpening revisited

What does blurring take away?

−

=

Let’s add it back:

+ α

=

Source： S. Lazebnik

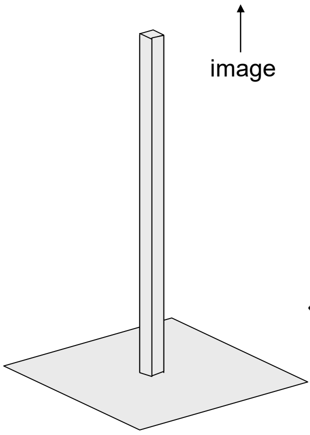
Unsharp mask filter

$$f + \alpha(f - f * g) = (1+\alpha)f - \alpha f * g = f * ((1+\alpha)e - \alpha g)$$

image

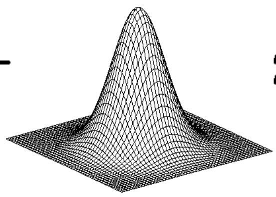
blurred image

unit impulse (identity)



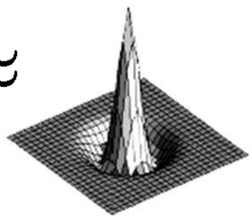
unit impulse

—



Gaussian

≈



Laplacian of Gaussian

Source： S. Lazebnik