数字图像处理
Digital Image Processing

Image as A Function

Discrete Sampling of A Function

Spatial Convolution

Most slides are courtesy of Juyong Zhang

Mathematical Convolution?

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

$$= \int_{-\infty}^{\infty} f(t-\tau)\, g(\tau)\, d\tau.$$

Meaning of Convolution?

滑动平均

- Mathematics
- Signal
  - 激励信号、响应函数
- Statistics
- Life
  - 下棋
  - 复利
  - …

Spatial Filtering

Let **I** and **J** be images such that **J** = T[**I**].

T[·] represents a transformation, such that,

$$\mathbf{J}(r,c) = \mathrm{T}[\mathbf{I}](r,c) =$$
$$f\left(\left\{\mathbf{I}(\rho,\chi) \,\middle|\, \rho \in \{r-s,\ldots,r,\ldots r+s\}, \chi \in \{c-d,\ldots,c,\ldots c+d\}\right\}\right).$$

That is, the value of the transformed image, **J**, at pixel location (r,c) is a function of the values of the original image, **I**, in a $2s+1 \times 2d+1$ rectangular neighborhood centered on pixel location (r,c).

Moving Windows

- The value, $\mathbf{J}(r,c) = \mathrm{T}[\mathbf{I}](r,c)$, is a function of a rectangular neighborhood centered on pixel location (r,c) in $\mathbf{I}$.
- There is a different neighborhood for each pixel location, but if the dimensions of the neighbor-hood are the same for each location, then trans-form T is sometimes called a *moving window transform*.

Moving-Window Transformations

Neutral Buoyancy Facility at NASA Johnson Space Center

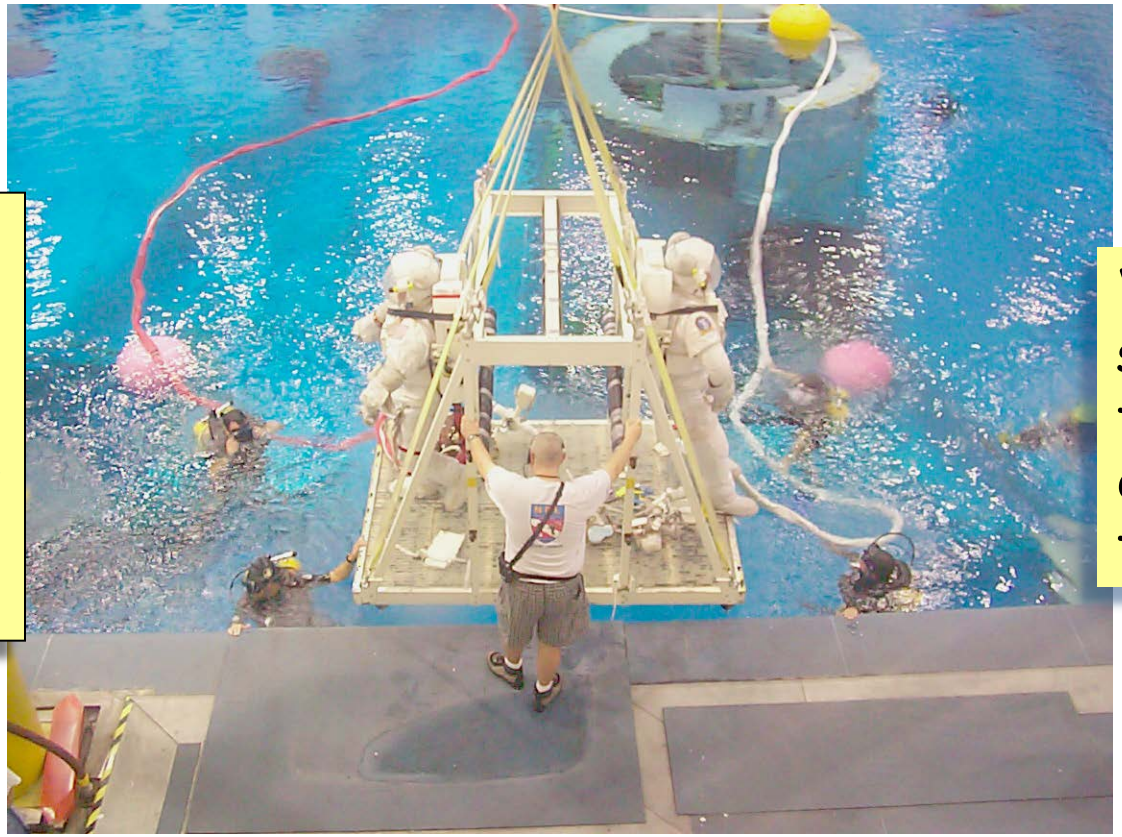
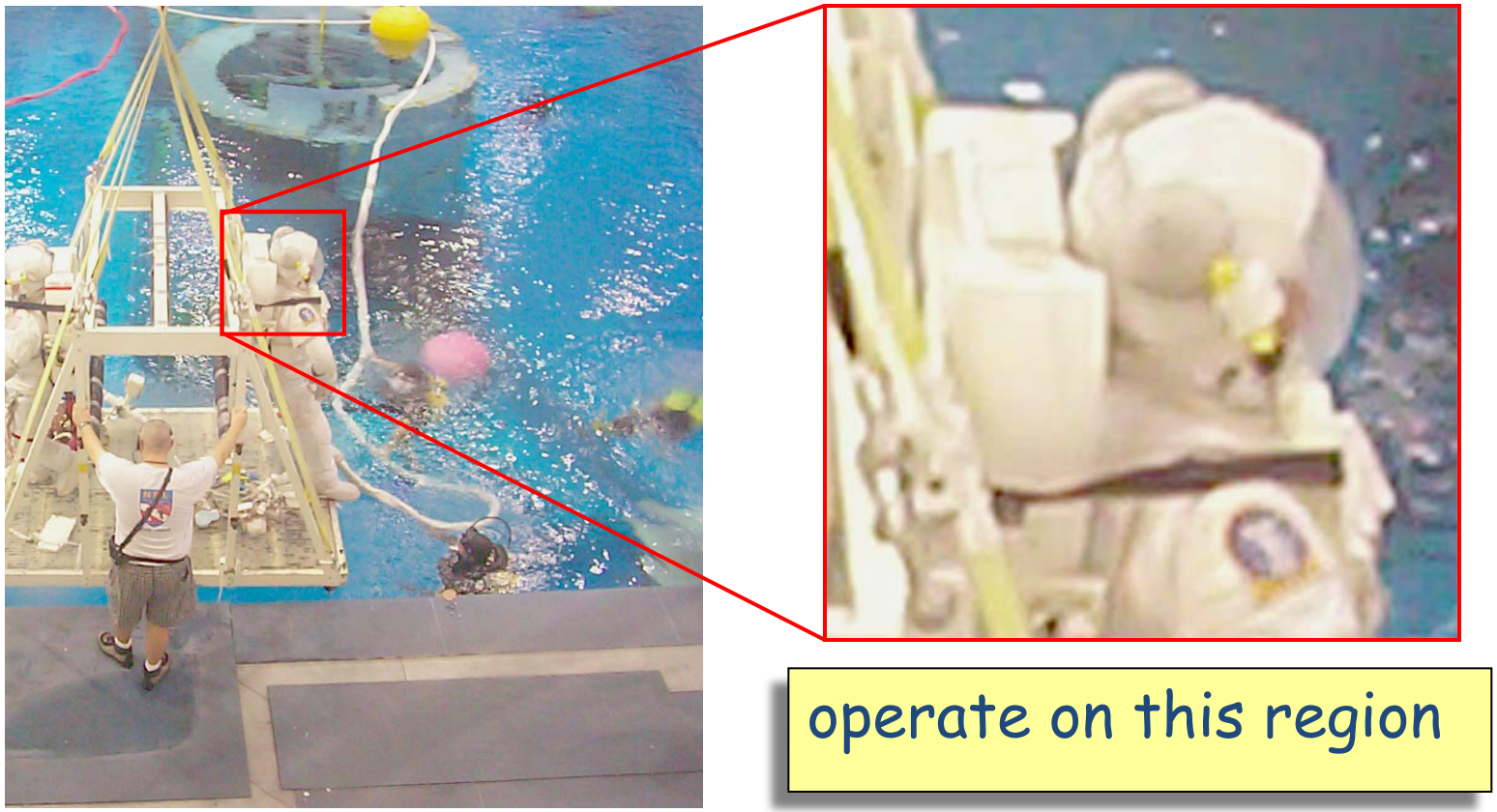


photo: R.A.Peters II, 1999

We' ll take a section of this image to demonstrate the MWT

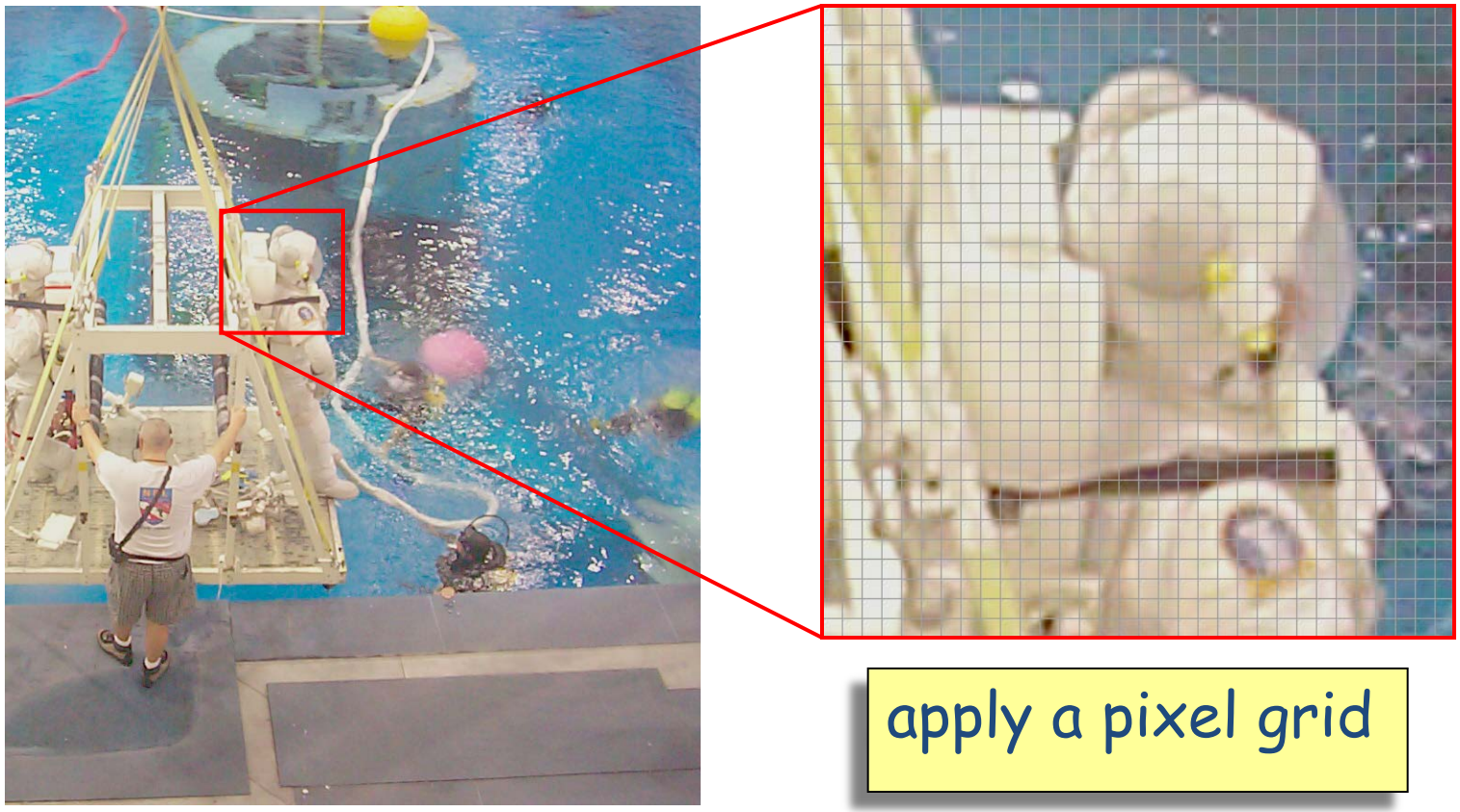
Moving-Window Transformations



operate on this region

Pixelize the section to better see the effects.

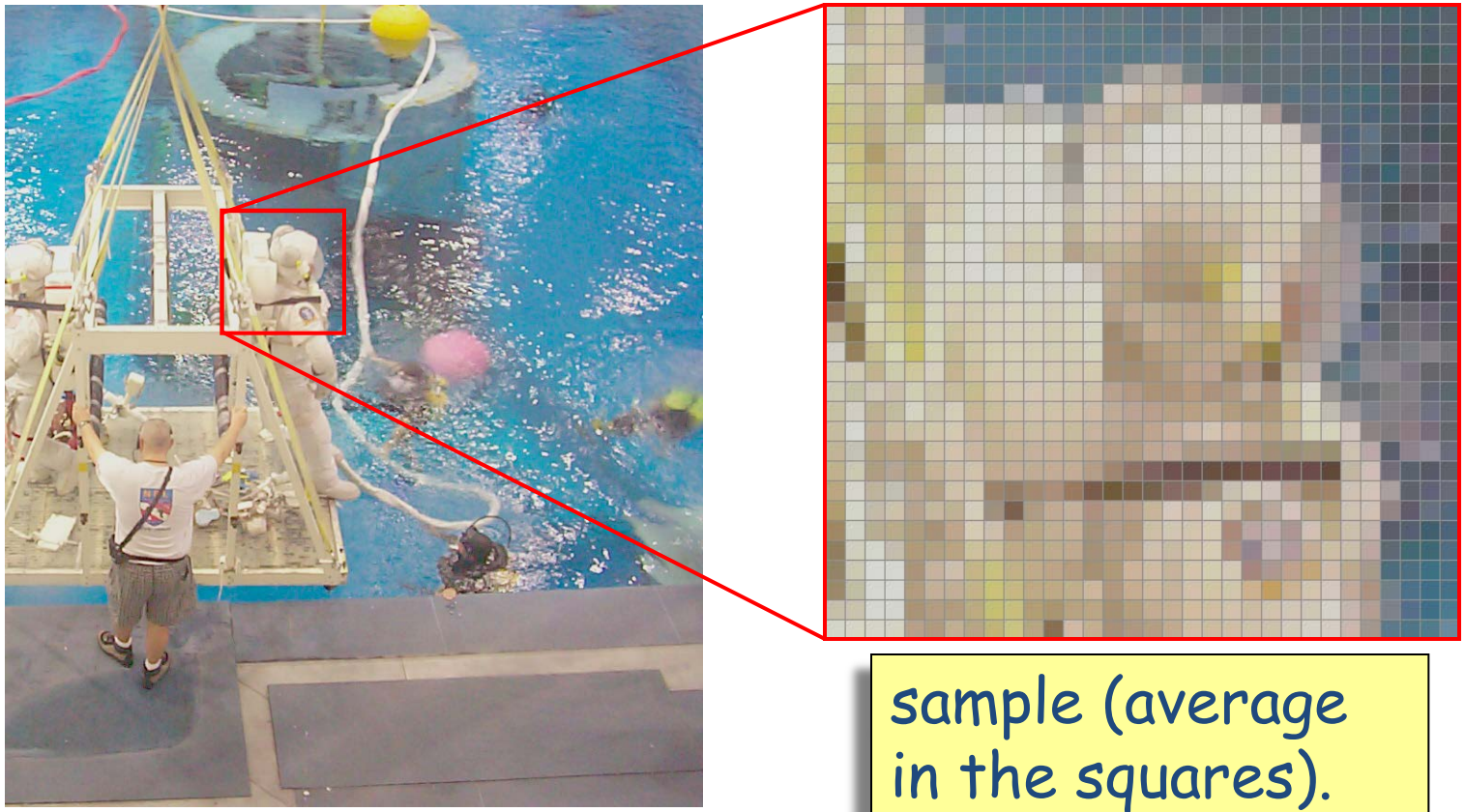
Moving-Window Transformations



apply a pixel grid

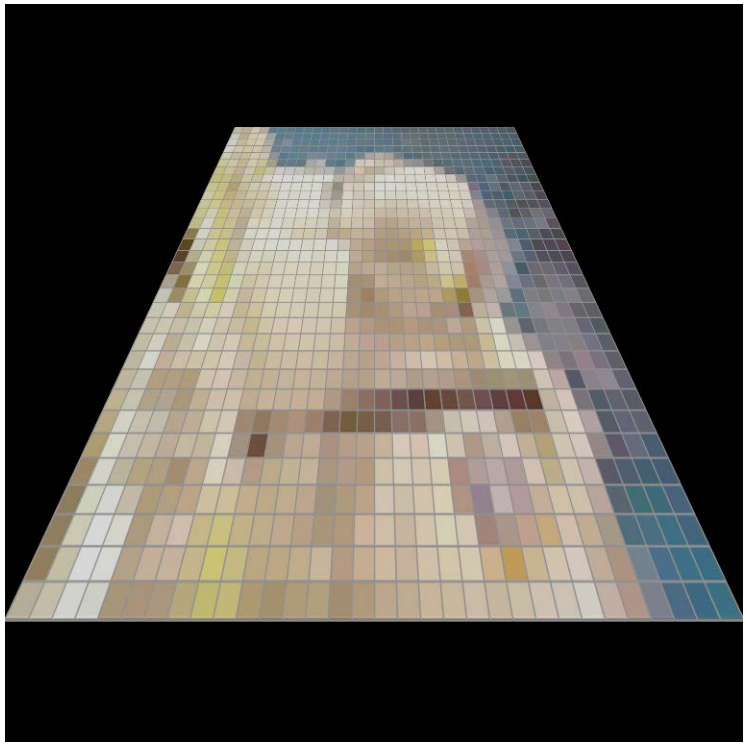
Pixelize the section to better see the effects.

Moving-Window Transformations



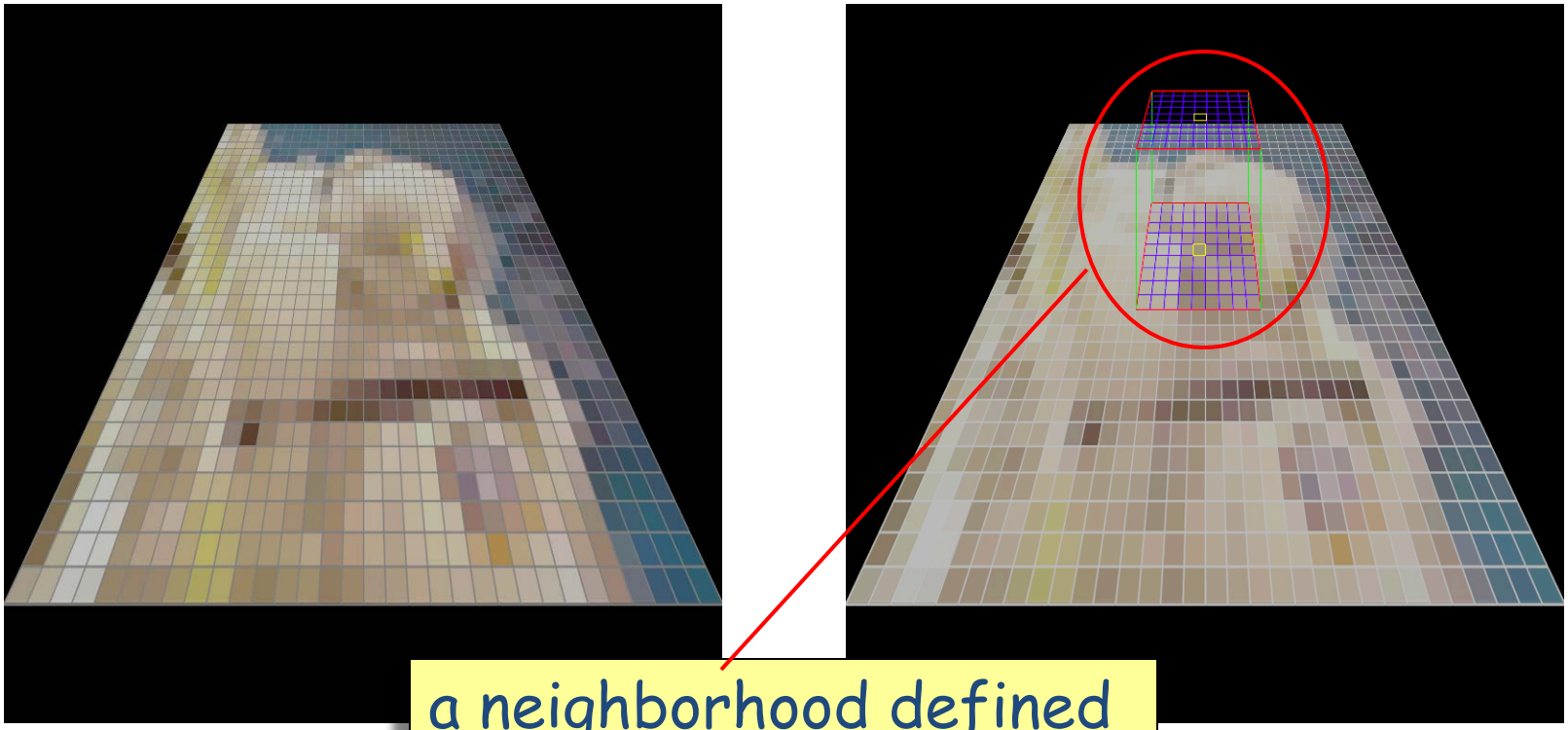
sample (average in the squares).

Moving-Window Transformations

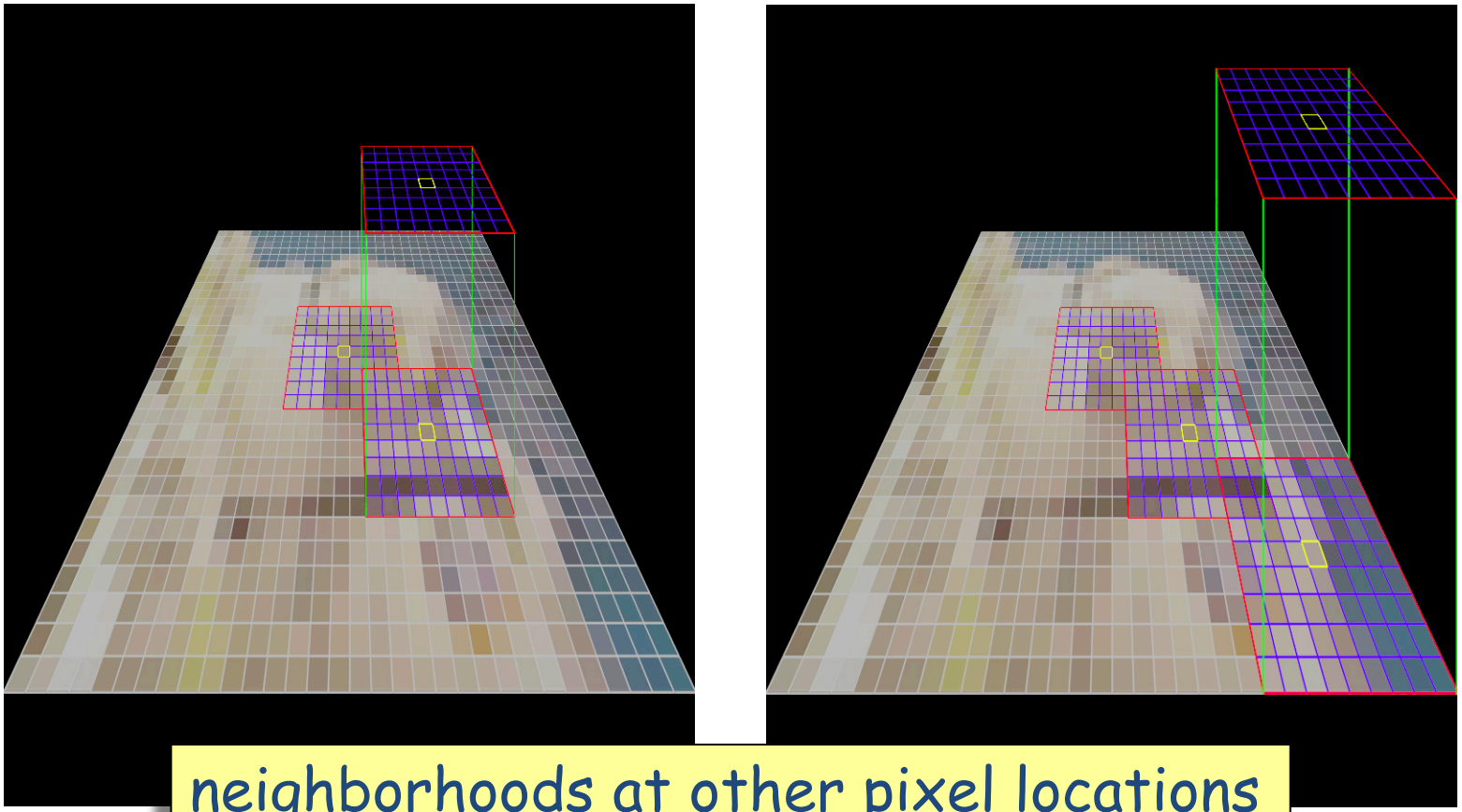


lets get some perspective on this

Moving-Window Transformations



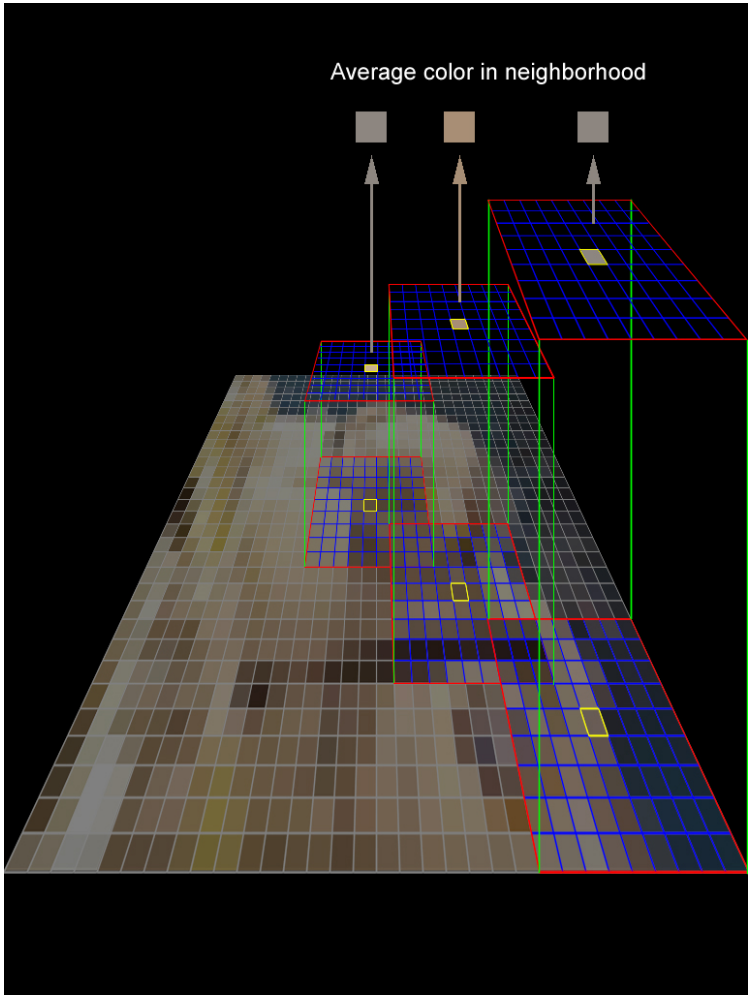
Moving-Window Transformations



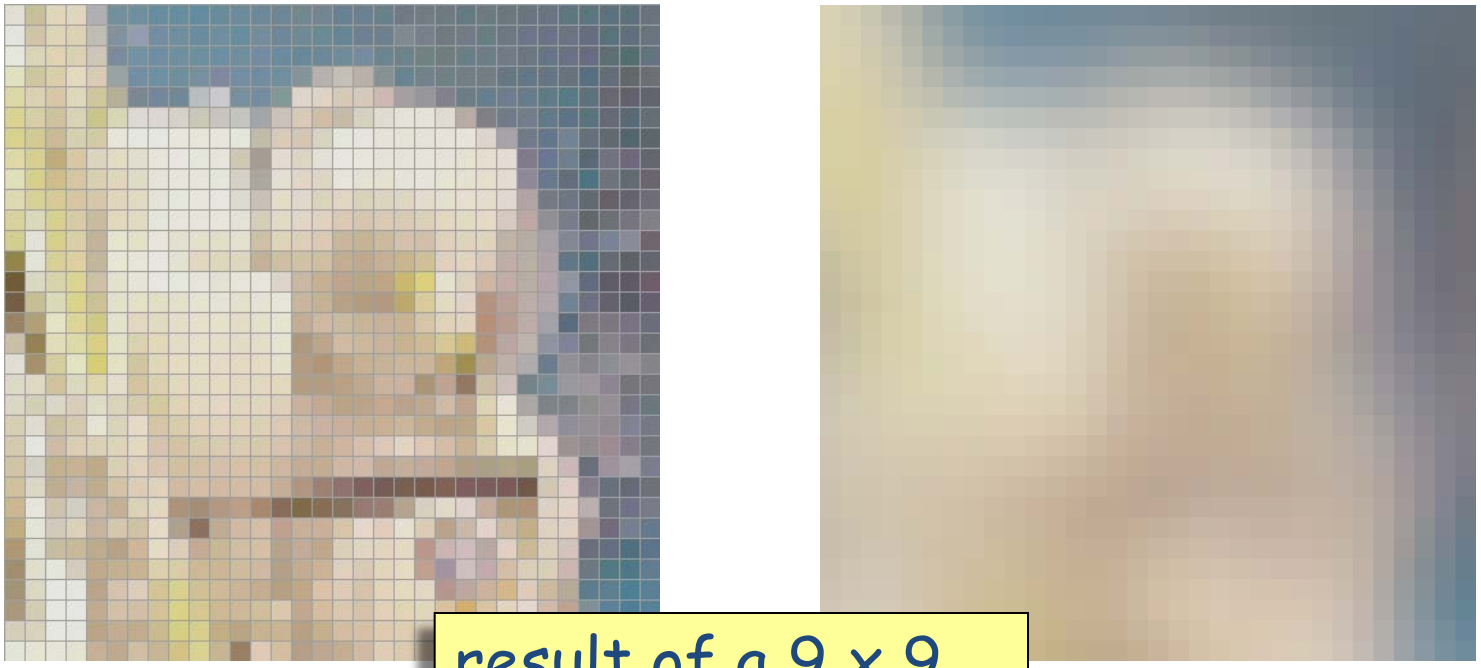
Linear Moving-Window Transformations

(*i.e.* convolution)

The output of the transform at each pixel is the (weighted) average of the pixels in the neighborhood.



Moving-Window Transformations



result of a 9 x 9 uniform averaging

Convolution: Mathematical Representation

注：
1）滤波器核h(i,j)定义了对应I(r-i,c-j)处的加权值
3）采用这种定义式的卷积，本质是相关
2）如果采用h(-i,-j)，则是相关运算

If a MW transformation is *linear* then it is a *convolution*:

$$\mathbf{J}(r,c)=\Big[\mathbf{I}*\mathbf{h}\Big](r,c)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathbf{I}(r-\rho,c-\chi)\mathbf{h}(\rho,\chi)\,d\rho d\chi,$$

for a real image ($\mathbf{I}: R\times R\to R$), or for a digital image ($\mathbf{I}: Z\times Z\to Z$):

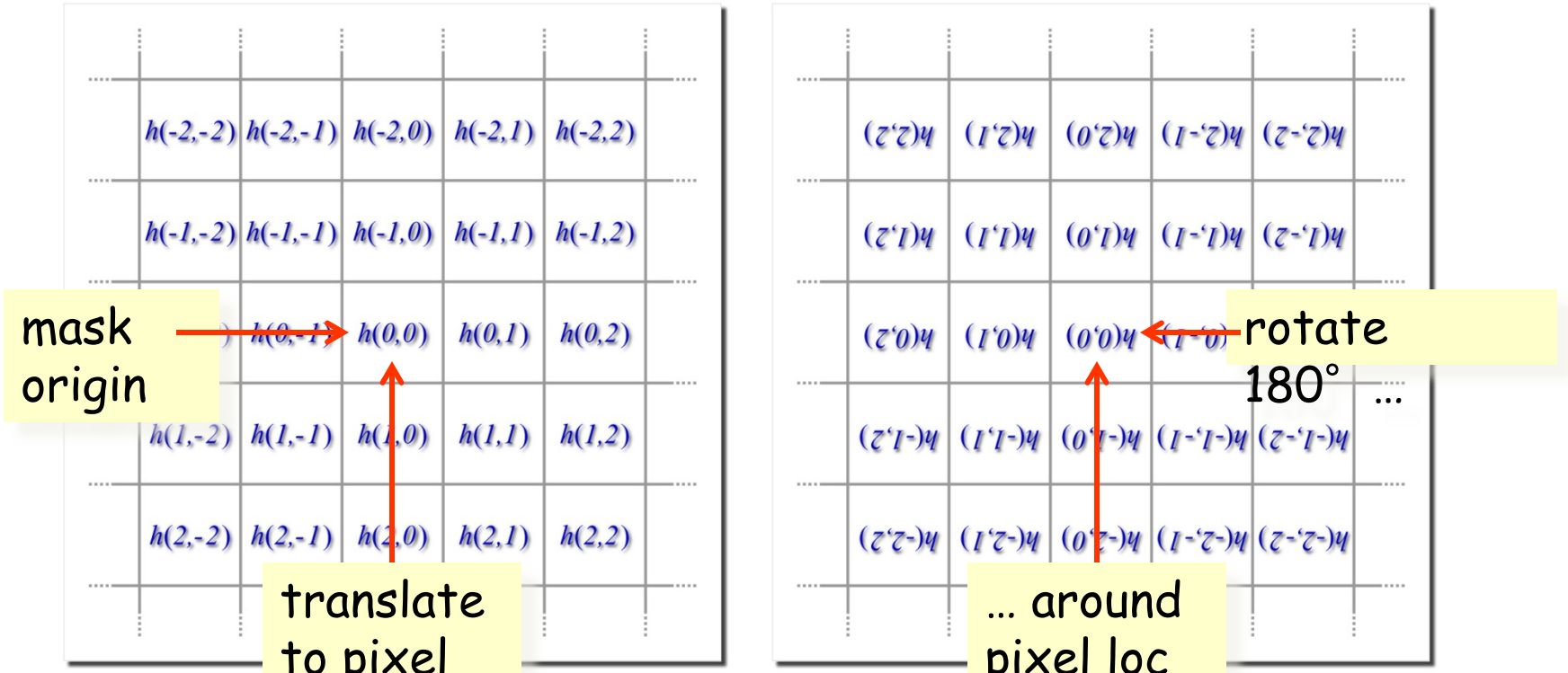
$$\mathbf{J}(r,c)=\Big[\mathbf{I}*\mathbf{h}\Big](r,c)=\sum_{\rho=-s}^{s}\sum_{\chi=-d}^{d}\mathbf{I}(r-\rho,c-\chi)\mathbf{h}(\rho,\chi)$$

h(i,j)：以3x3为例 - h(-1,-1) h(-1,0) h(-1,1)　　h(1,1) h(1,0) h(1,-1)
h(0,-1) h(0,0) h(0,1) —》 h(0,1) h(0,0) h(0,-1)
h(1,-1) h(1,0) h(1,1)　　h(-1,1) h(-1,0) h(-1,-1)

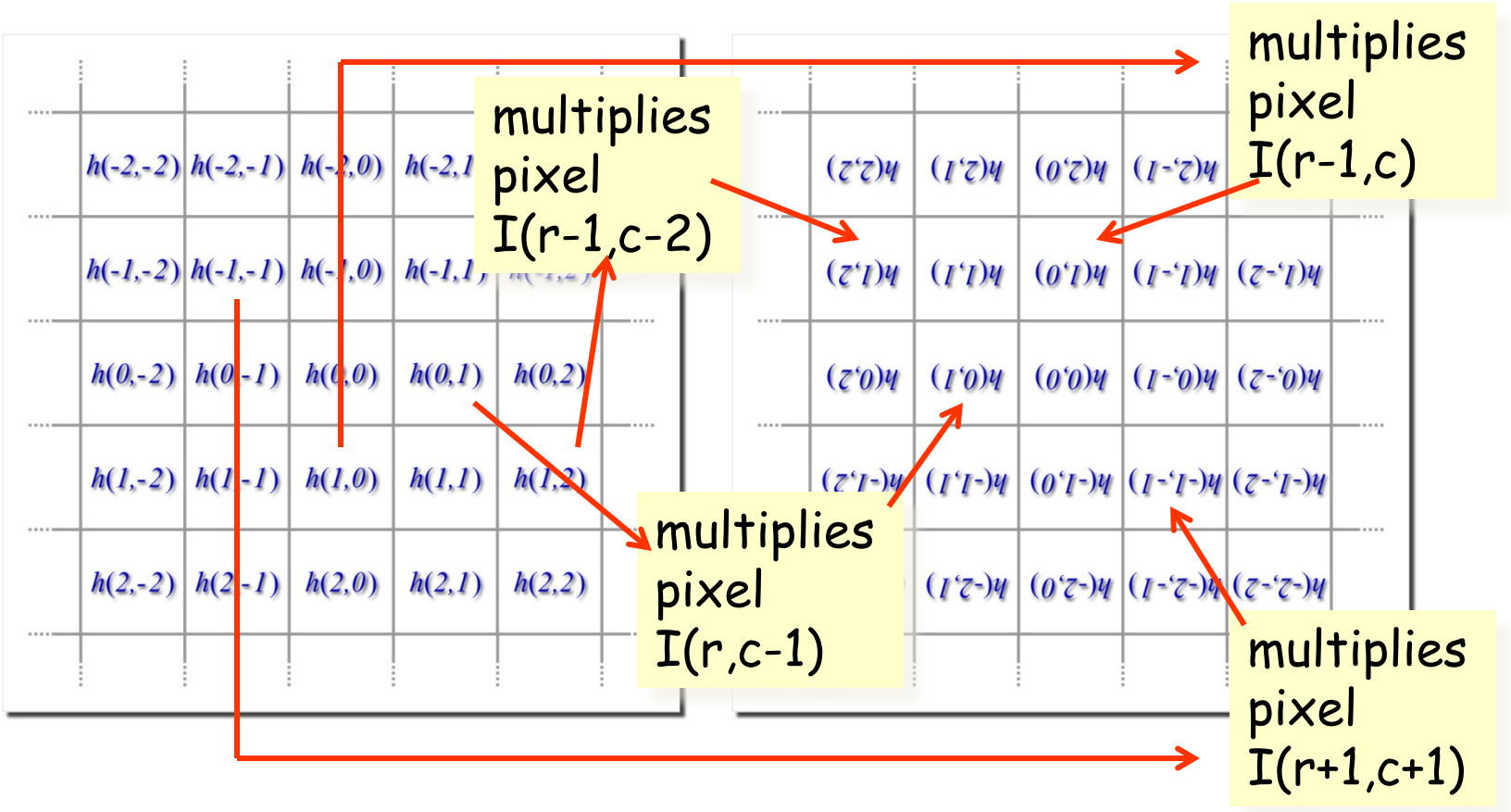
Convolution Mask (Weight Matrix)

- The object, **h**(ρ,χ), in the equation is a weighting function, or in the discrete case, a rectangular matrix of numbers.
- The matrix is the moving window.
- Pixel (*r*,*c*) in the output image is the weighted sum of pixels from the original image in the neighborhood of (*r*,*c*) traced by the matrix.
- Each pixel in the neighborhood of (*r*,*c*) is multiplied by the corresponding matrix value — after the matrix is rotated by 180º.
- The sum of those products is the value of pixel (*r*,*c*) in the output image

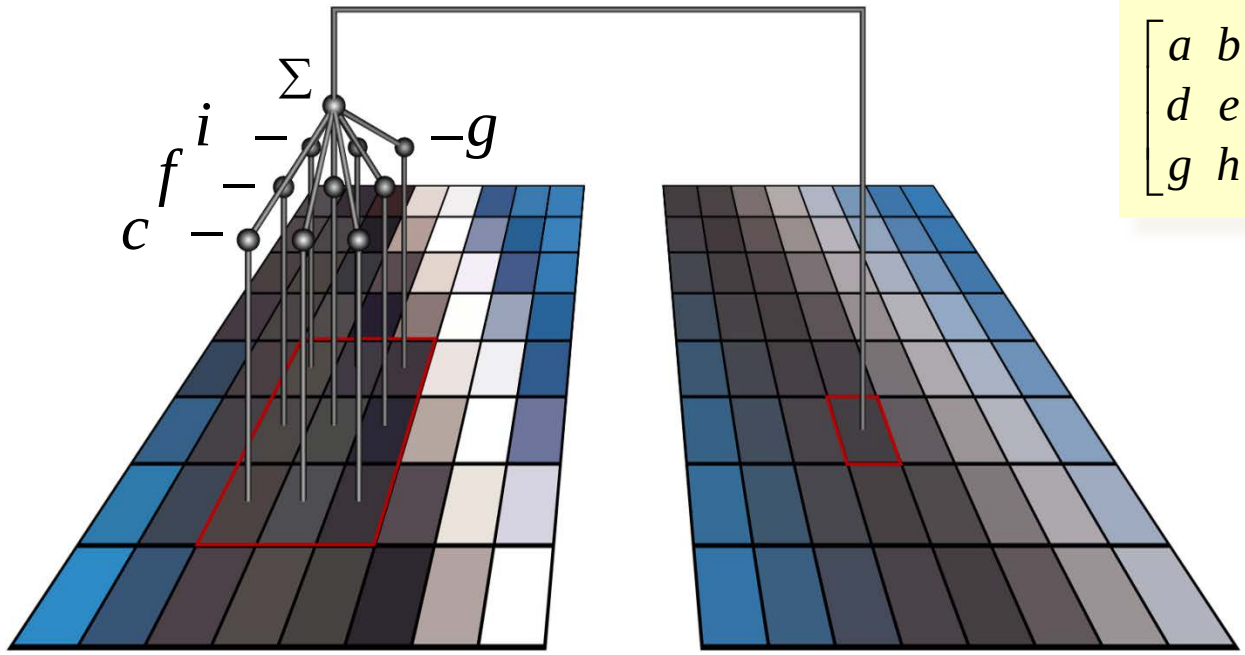
Convolution Masks: Moving Window



Convolution Masks: Moving Window



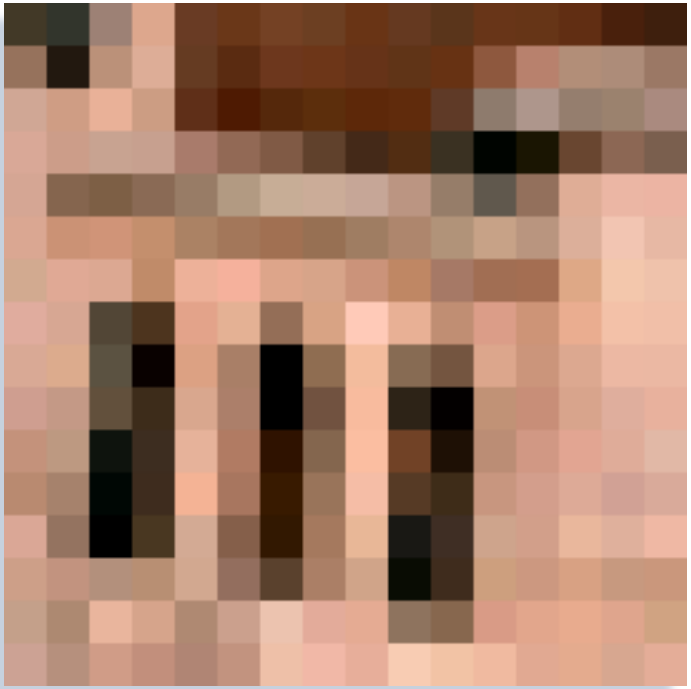
Convolution by Moving Window



$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Another example

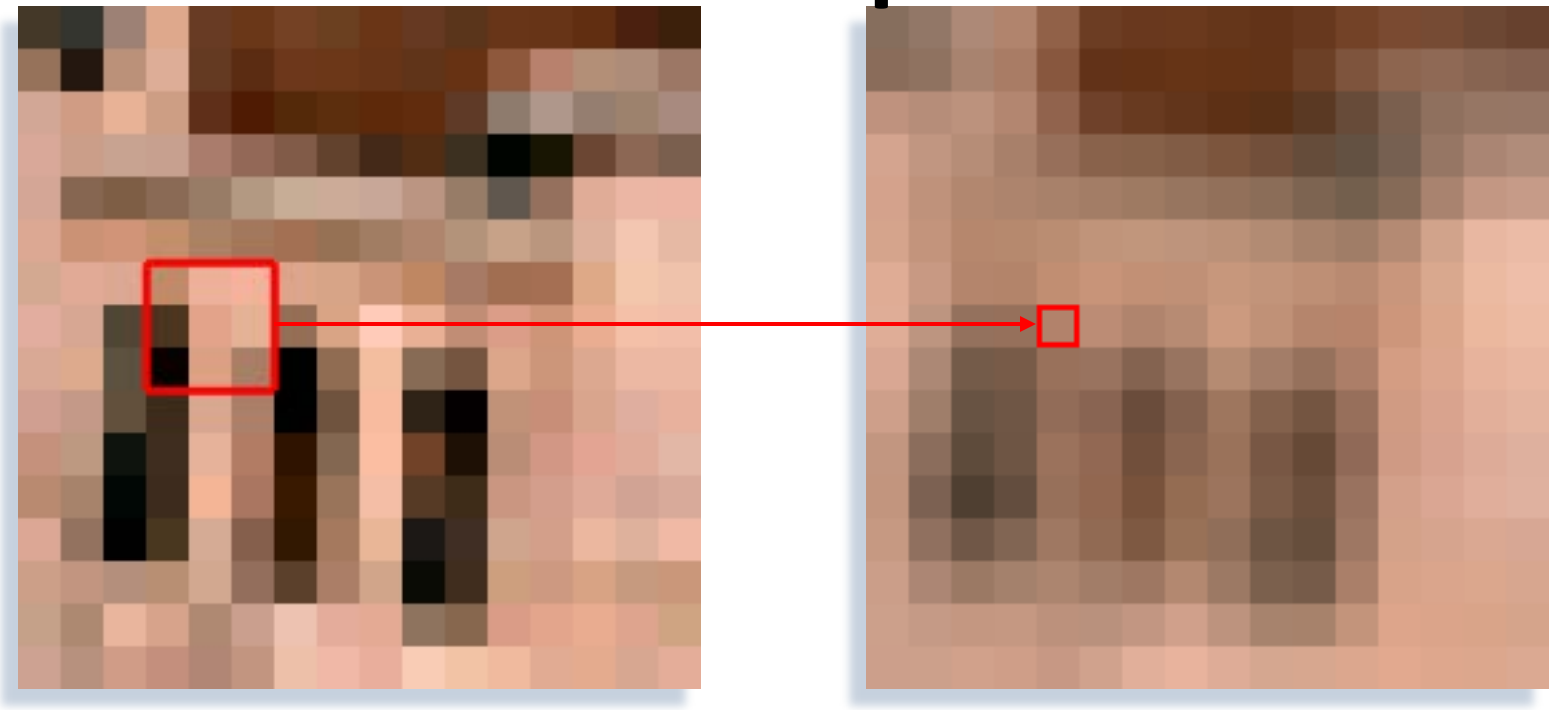
Moving Window Transform: Example



original

3x3 average

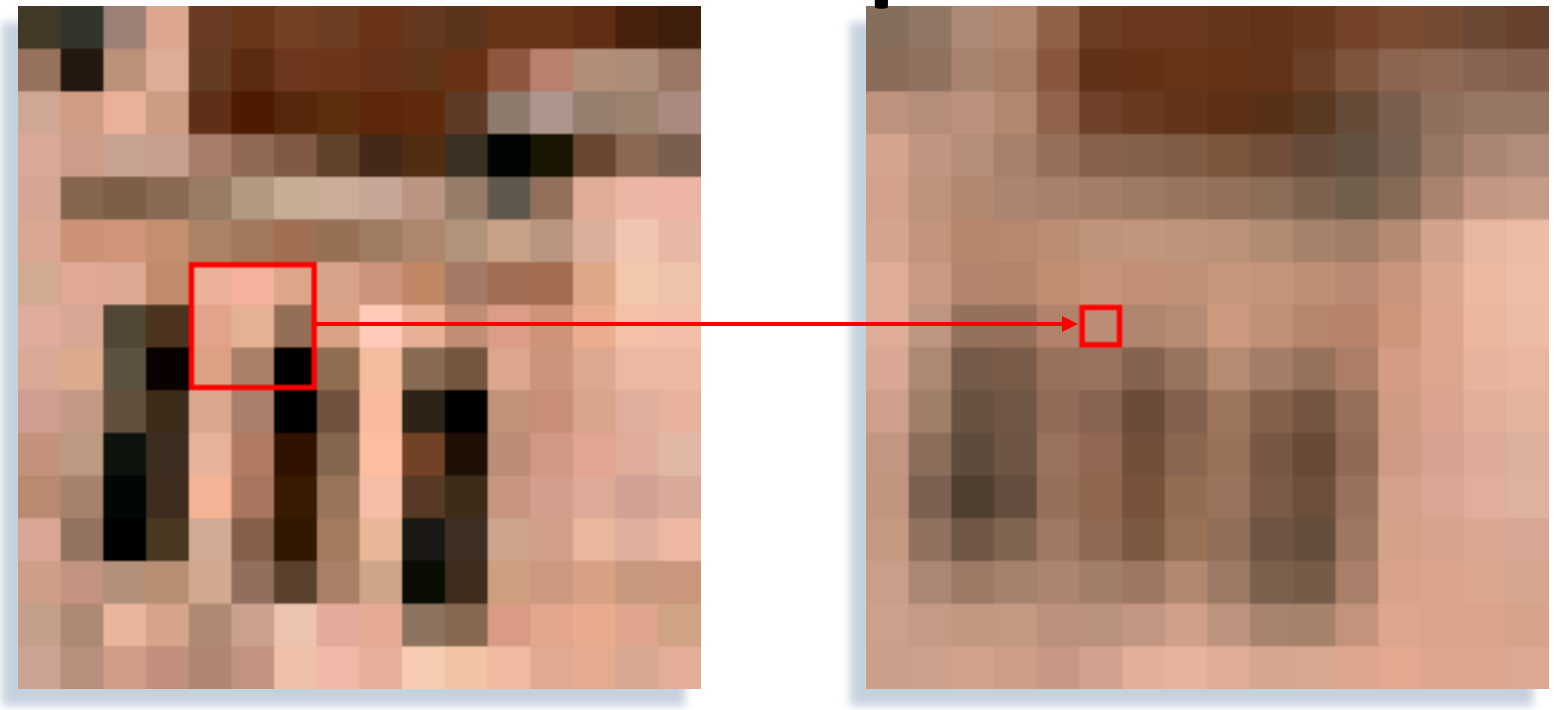
Moving Window Transform: Example



original

3x3 average

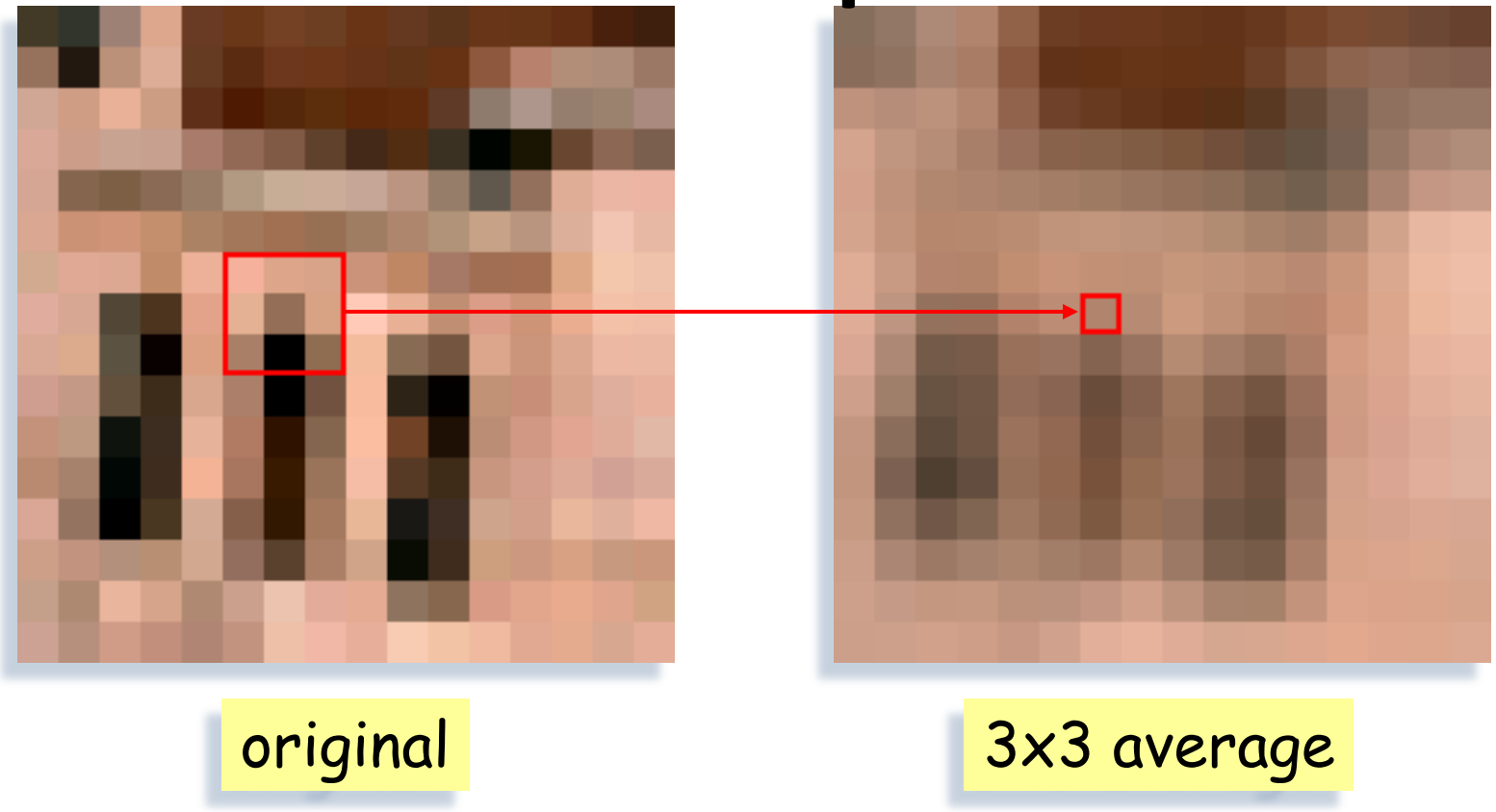
Moving Window Transform: Example



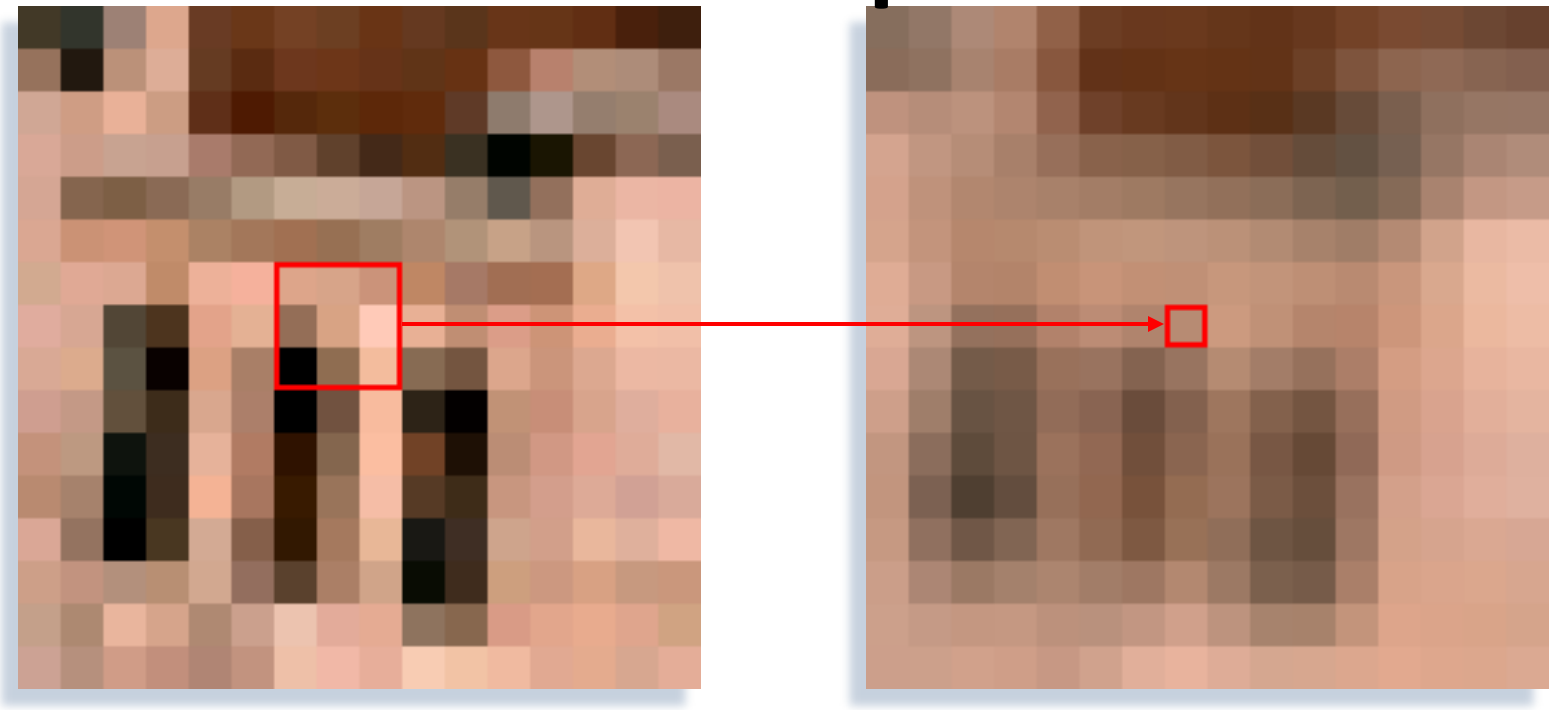
original

3x3 average

Moving Window Transform: Example



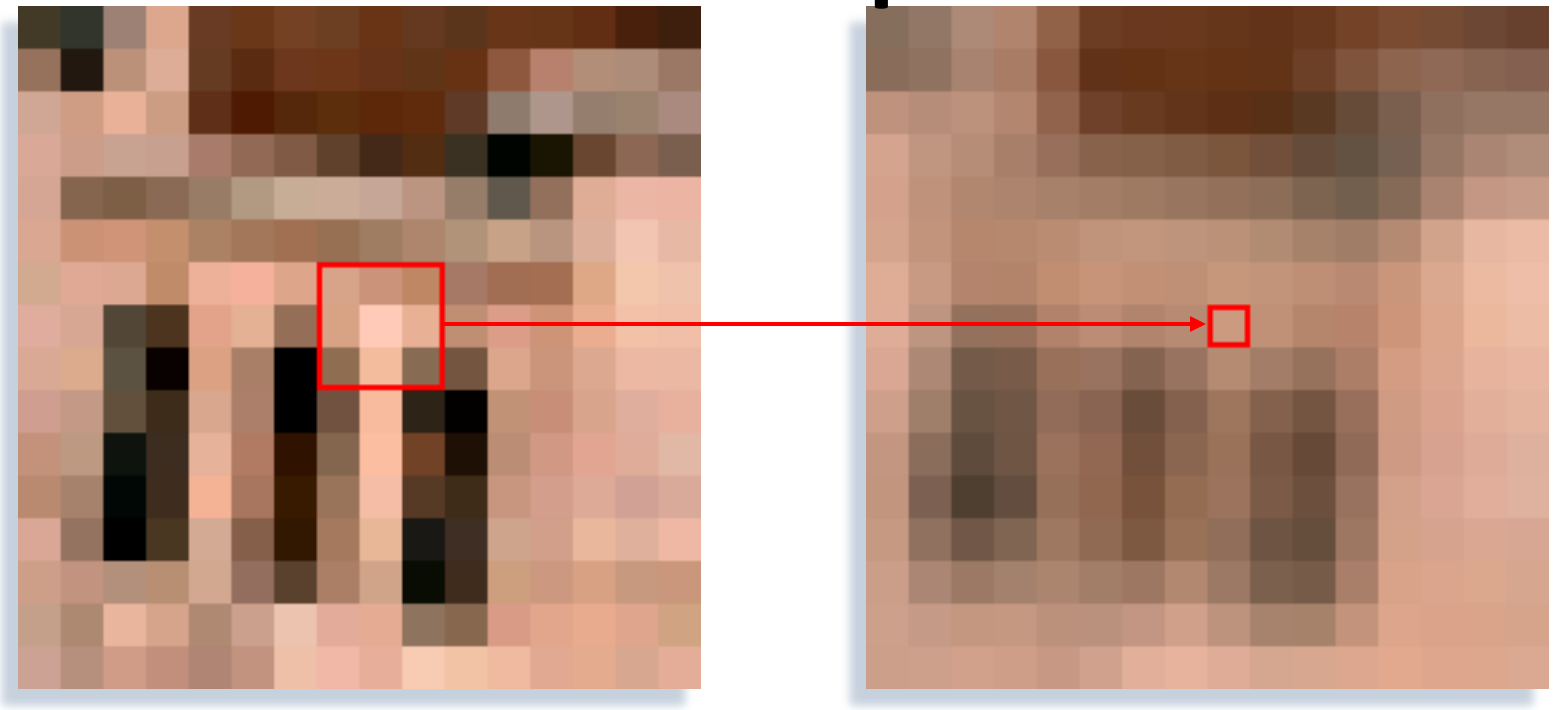
Moving Window Transform: Example



original

3x3 average

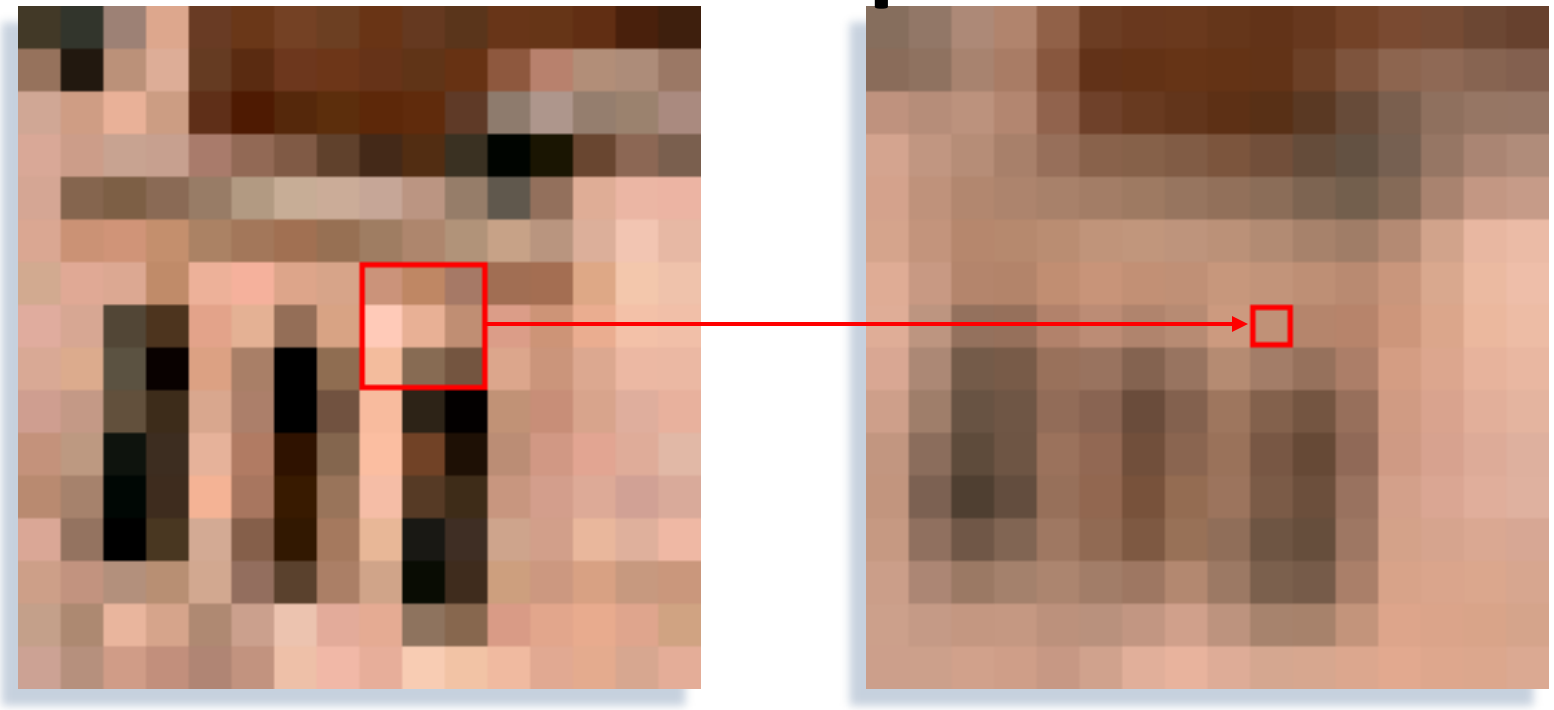
Moving Window Transform: Example



original

3x3 average

Moving Window Transform: Example



original

3x3 average

The Convolution

- The average filter is a particular example of a more general operation: **Image Convolution.**
- Let **A, B** be images. B is typically smaller than A.
- B is typically called the **mask** or the **kernel**.
- The convolution for 1D signal

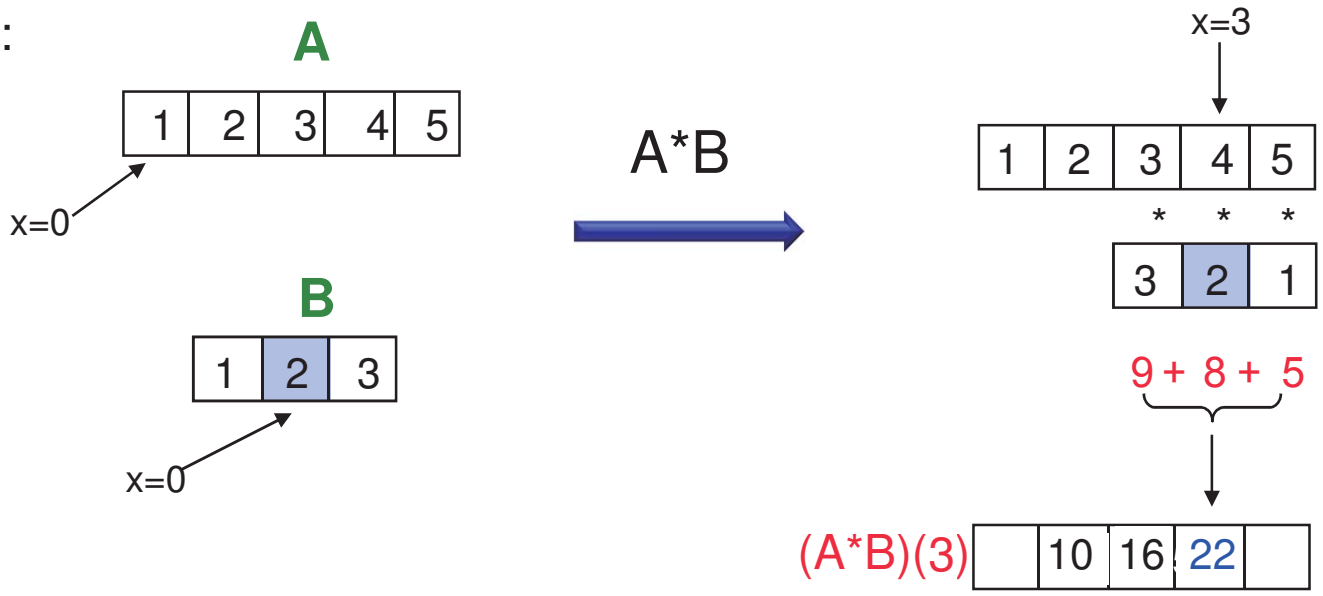
$$(A * B)(x) = \sum_i A(i)B(x-i)$$

1D Convolution

$$(A * B)(x) = \sum_i A(i)B(x-i)$$

注：
1）对于B(i), -1=< i <= 1
2）对于 x=3, -1=< 3-i<= 1
2=< i <=4
3）A(2) - B(1), A(3) - B(0)
A(4) - B(-1)
B()反折后与A()相关

Example:

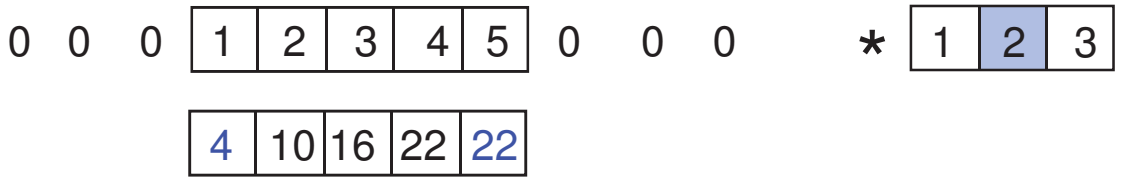


x

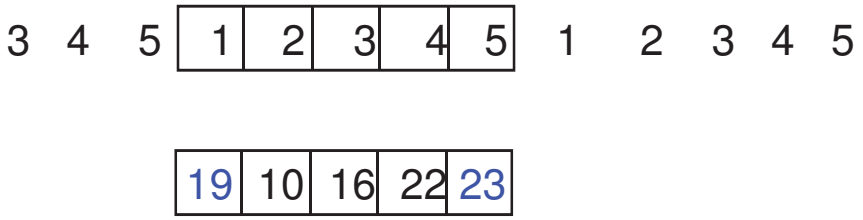
What happens near the edges?

注意：kernel需要反折—》3 2 1

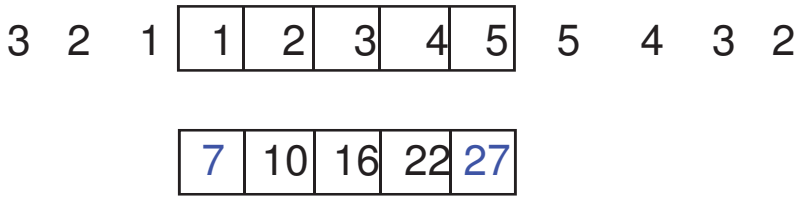
- Option 1: Zero padding



- Option 2: Wrap around

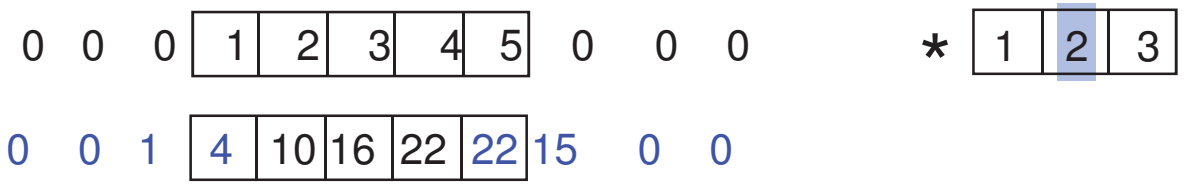


- Option 3: Reflection

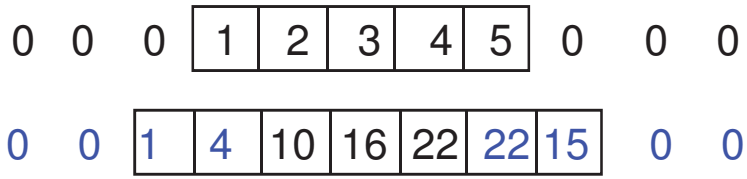


What is the length of the result?

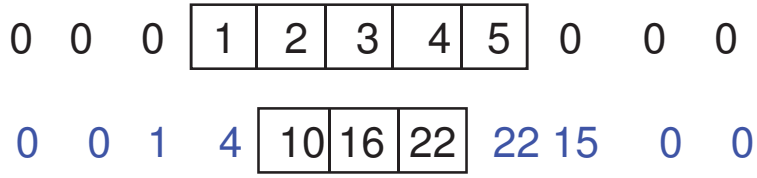
- Option 1: “same” (size A)



- Option 2: “full” (size A + size B - 1)

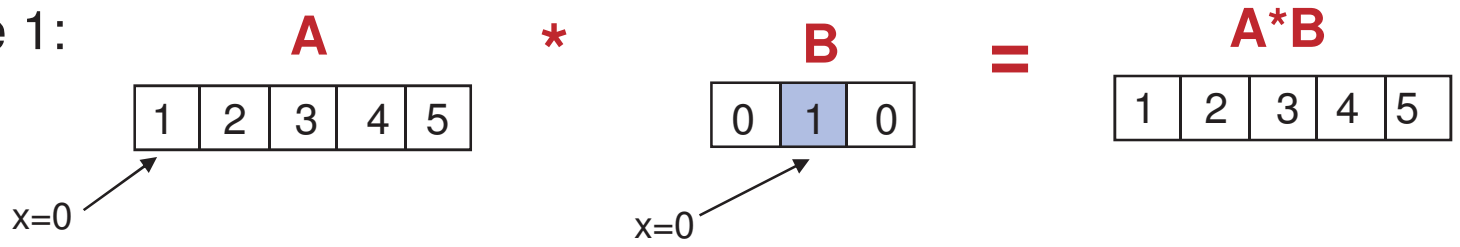


- Option 3: “valid” (size A – size B +1)

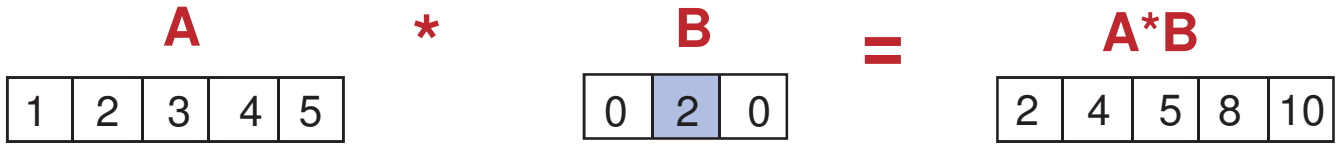


Examples

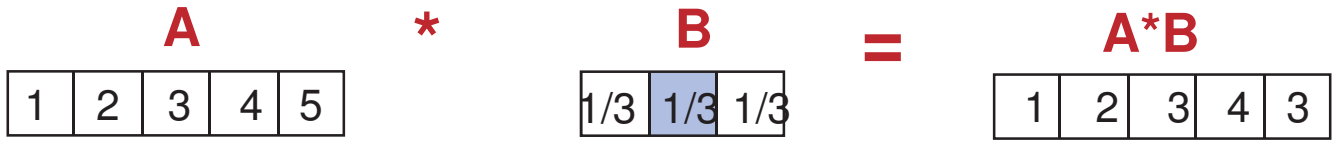
Example 1:



Example 2:

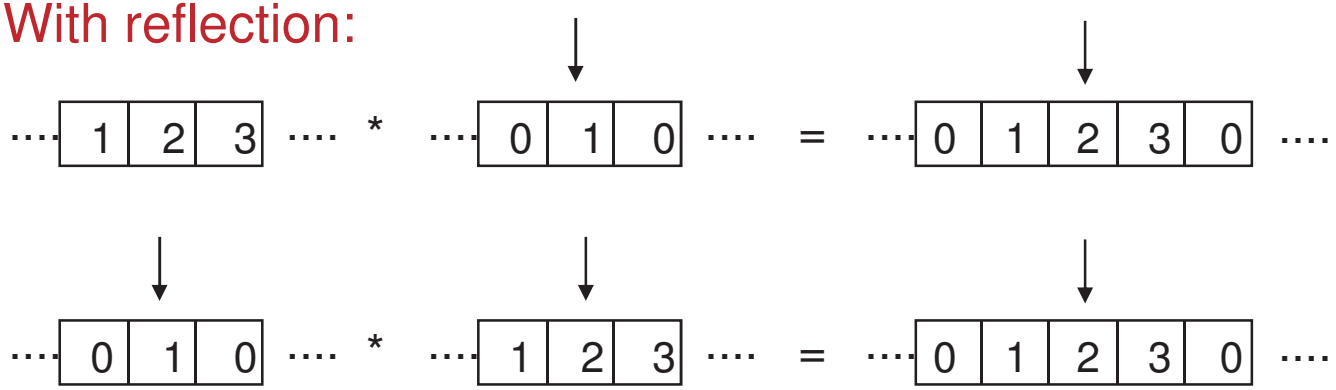


Example 3:

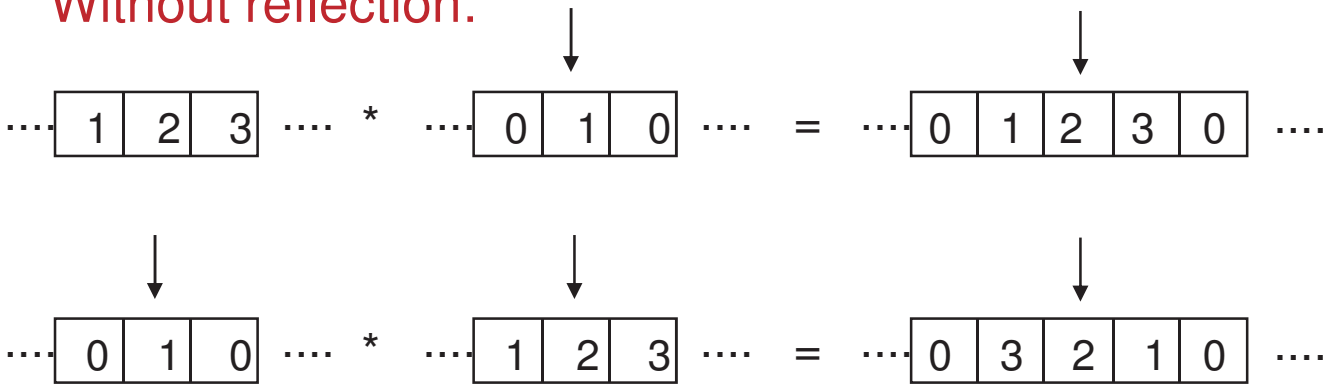


- Why should we flip the mask before the convolution?

With reflection:



Without reflection:

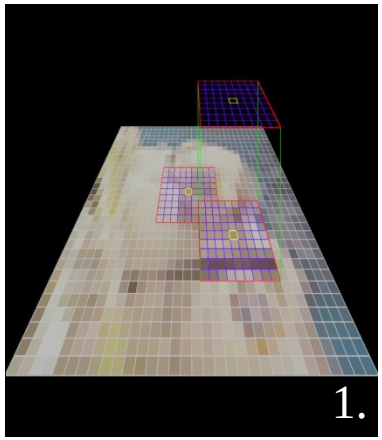


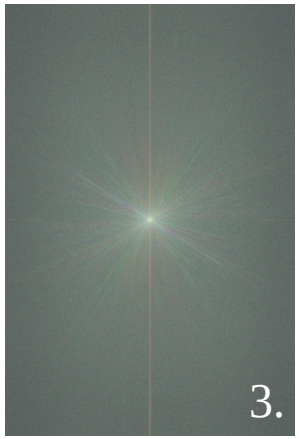
- Reflection is needed so that convolution is commutative:

A*B=B*A

Three ways to compute a convolution

1. Moving window transform as just shown.
2. Shift multiply add.
3. Fourier transform.





Shift-Multiply-Add Approach

- The image is copied 1 time for each element in the convolution mask.
- Each copy is shifted relative to the original by the displacement of its associated mask element.
- Each copy is multiplied by the value of its associated mask element.
- The set of shifted and multiplied images is summed pixel wise.

Convolution by an Impulse

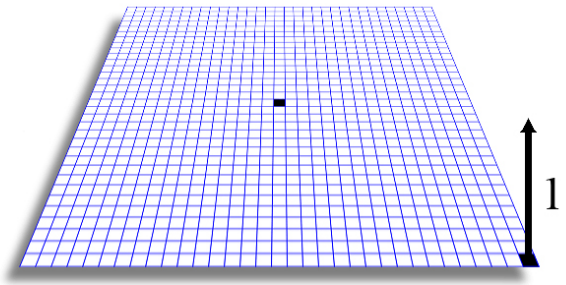
An *impulse* is a digital image, that has a single pixel with value 1; all others have value zero. An impulse at location (ρ, χ) is represented by:

$$\delta(r-\rho, c-\chi) = \begin{cases} 1, & \text{if } r=\rho \text{ and } c=\chi \\ 0, & \text{otherwise} \end{cases}$$

If an image is convolved with an impulse of weight *w* at location (ρ, χ), then the image is multiplied by *w* and shifted in location down by ρ pixels and to the right by χ pixels.

$$\Big[\mathbf{I} * w\delta\big(r-\rho, c-\chi\big)\Big](r,c) = w\mathbf{I}\big(r-\rho, c-\chi\big).$$

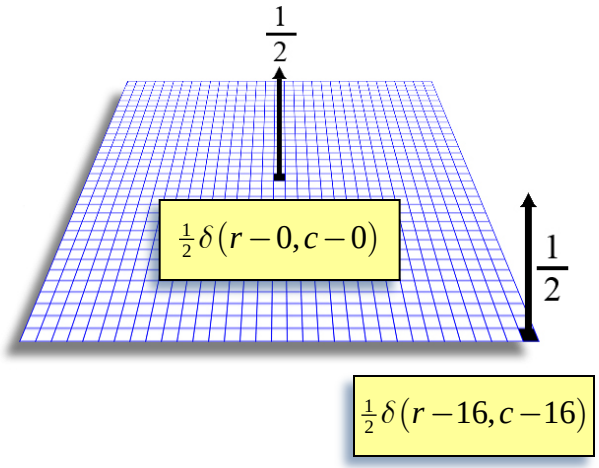
Convolution by an Impulse



$\delta(r-16, c-16)$

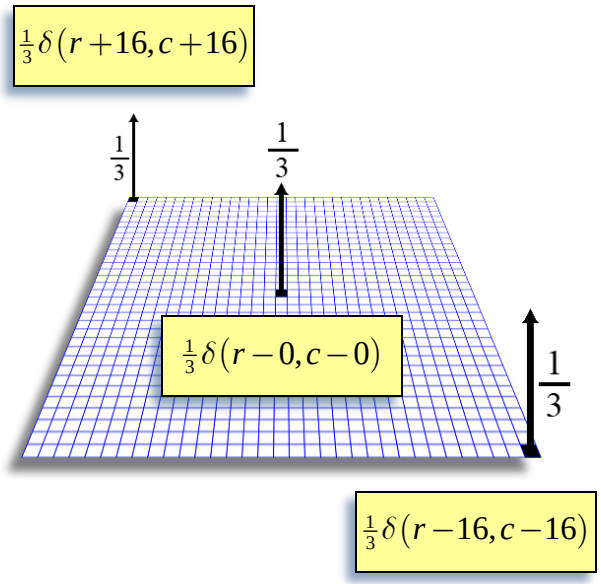
Shifted down and to the right by 16 pixels.

Convolution by Two Impulses



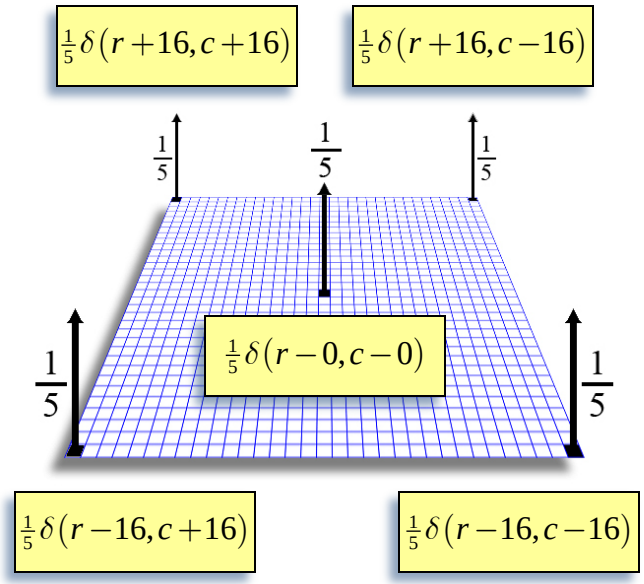
Two copies, one moved,
one not moved, averaged.

Convolution by Three Impulses



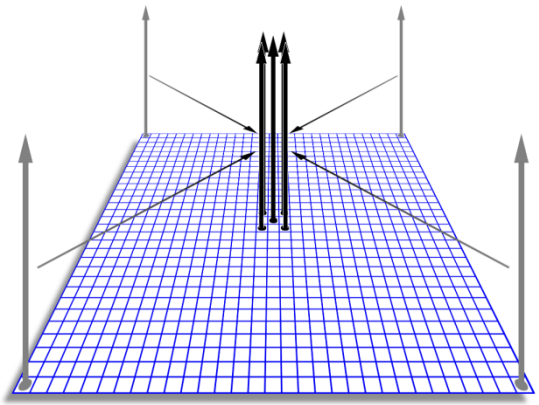
Three copies, two moved, one not moved, averaged.

Convolution by Five Impulses



Five copies, four moved, one not moved, averaged.

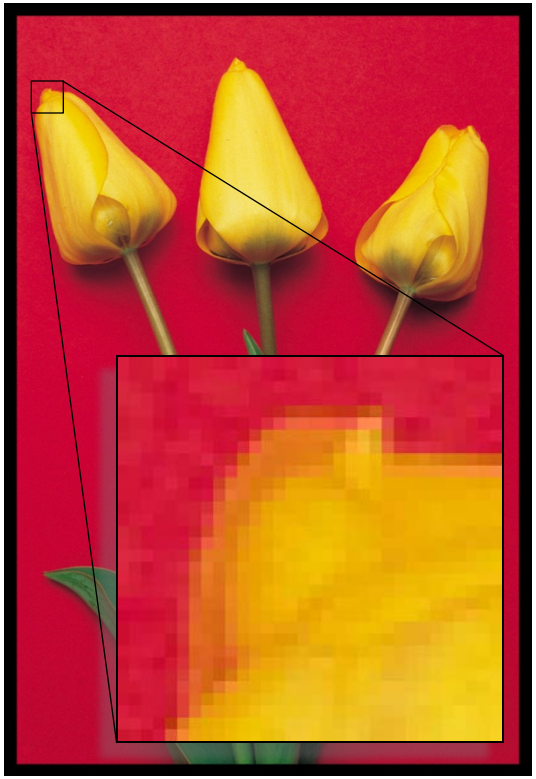
Convolution by Five Impulses

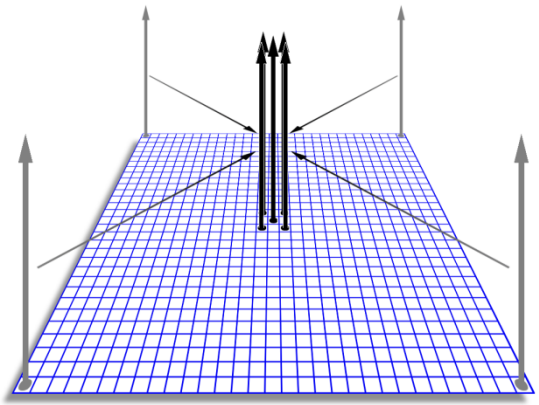


Moved adjacent to each other, the convolution becomes a blurring filter.

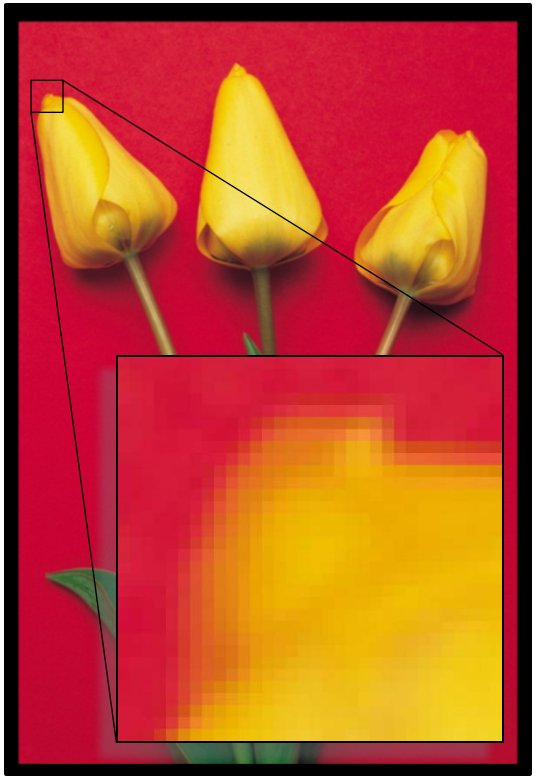
又一个例子，用于仿真图像运动模糊模型

Convolution by Five Impulses

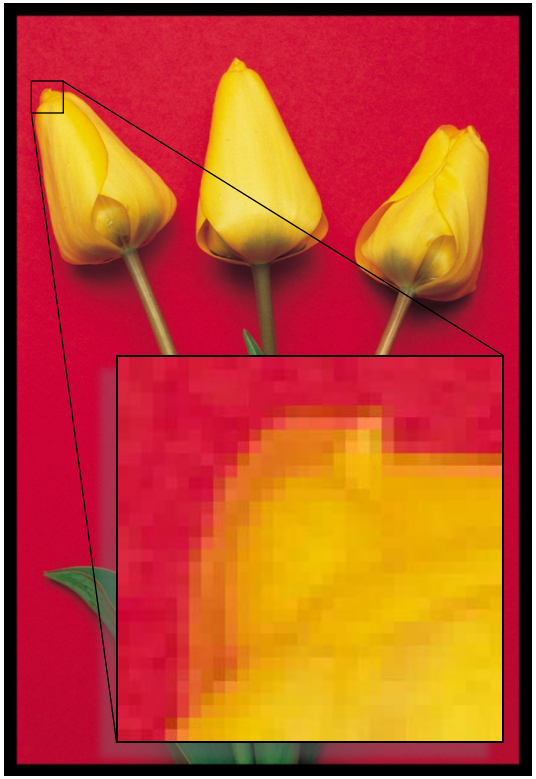


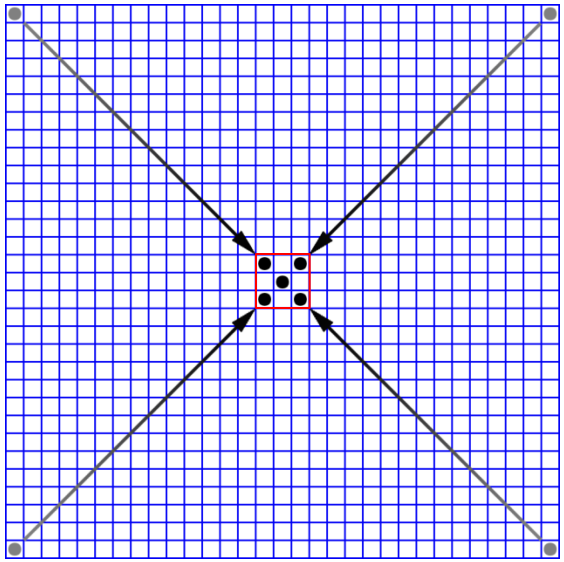


The impulses become values in a 3x3 neighborhood.

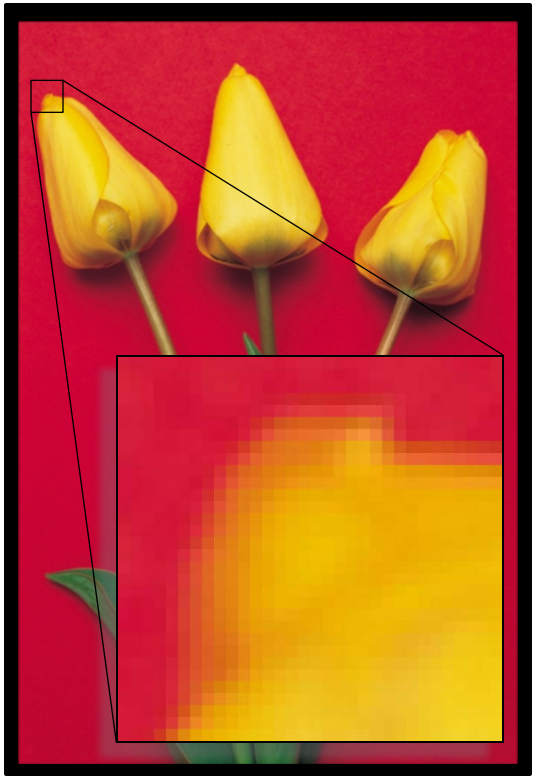


Convolution by Five Impulses



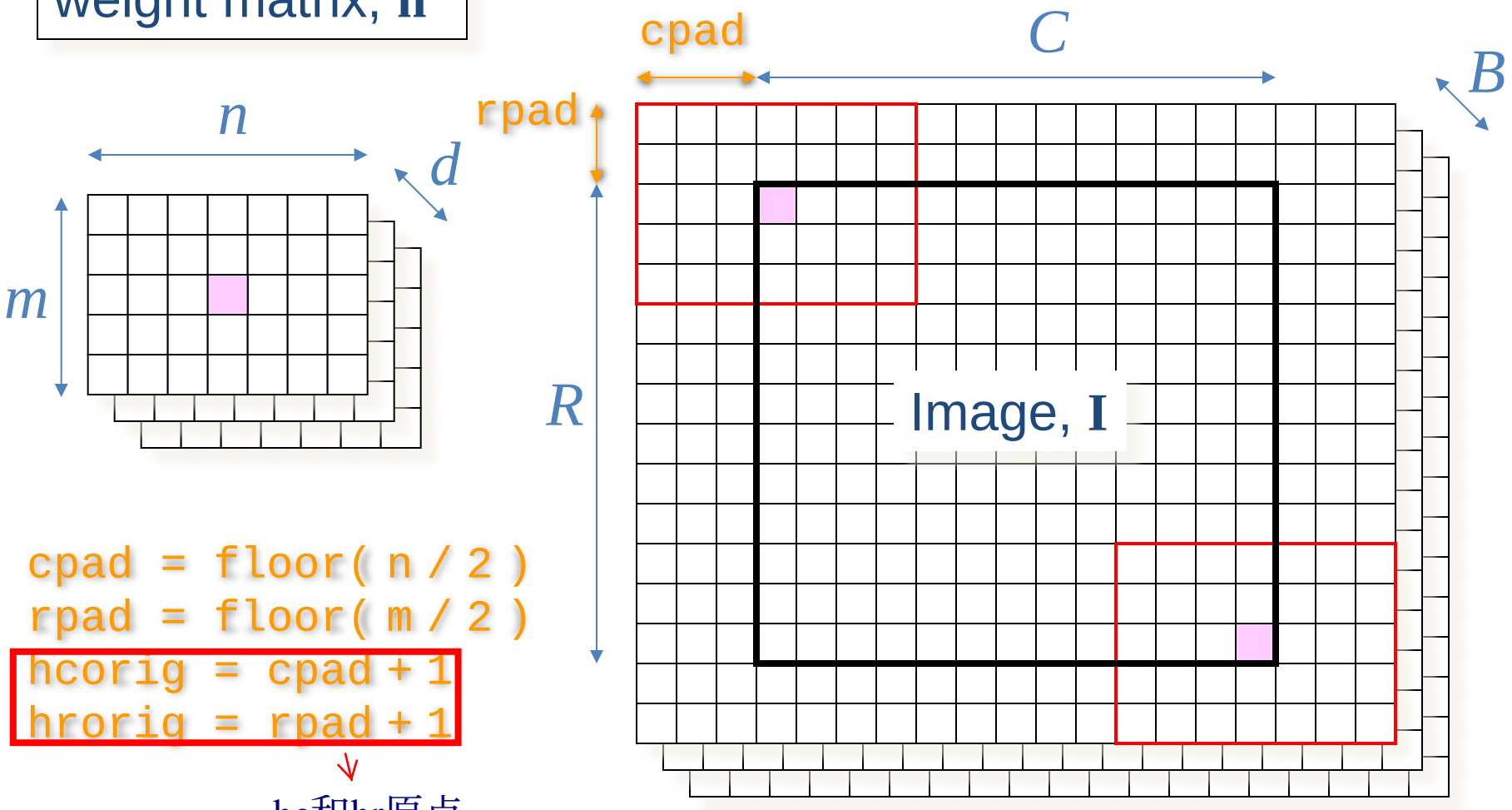


The convolution mask has five elements at 1/5 and four at 0.



Zero Padding an Image for Convolution: Variable Names.

weight matrix, **h**

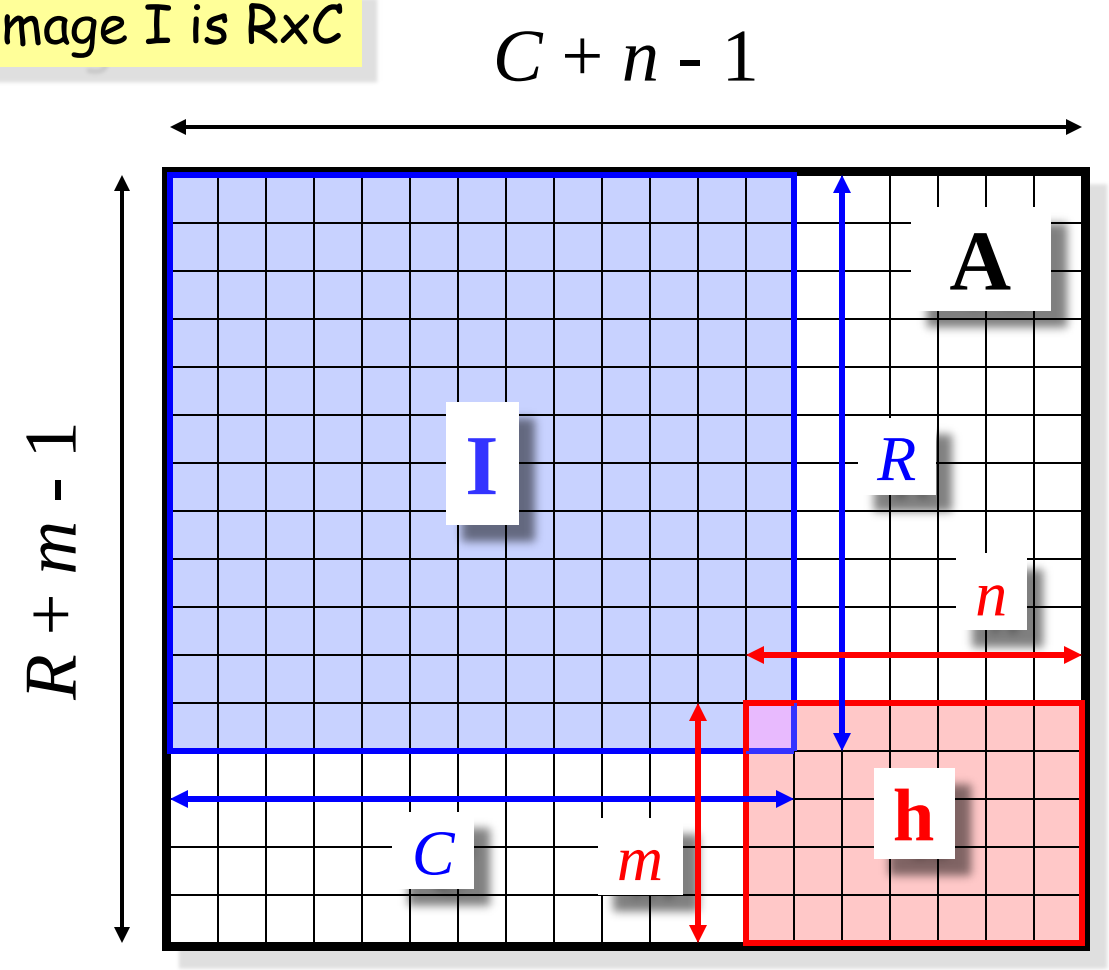



```
cpad = floor( n / 2 )
rpad = floor( m / 2 )
hcorig = cpad + 1
hrorig = rpad + 1
```

hc和hr原点，
也是图像卷积模板的原点

Convolution by Copying and Shifting the Image

Image I is RxC



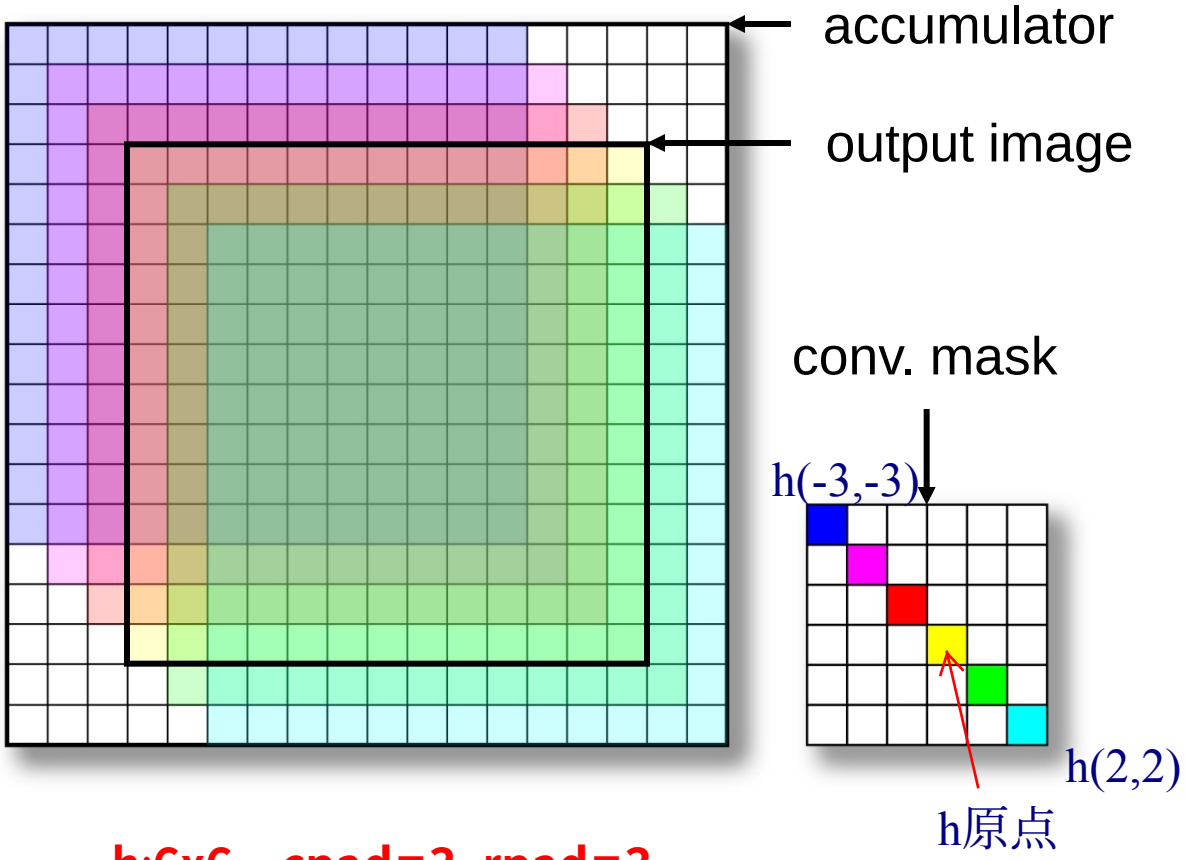
To use the image shift-multiply-accumulate algorithm, create an accumulator image, A, that is R+m-1 rows by C+n-1 columns

Convolution by Copying, Multiplying, and Shifting the Image

13x13 image convolved by 6x6 mask.

Image is constant; mask has only 6 nonzero values all on the diagonal.

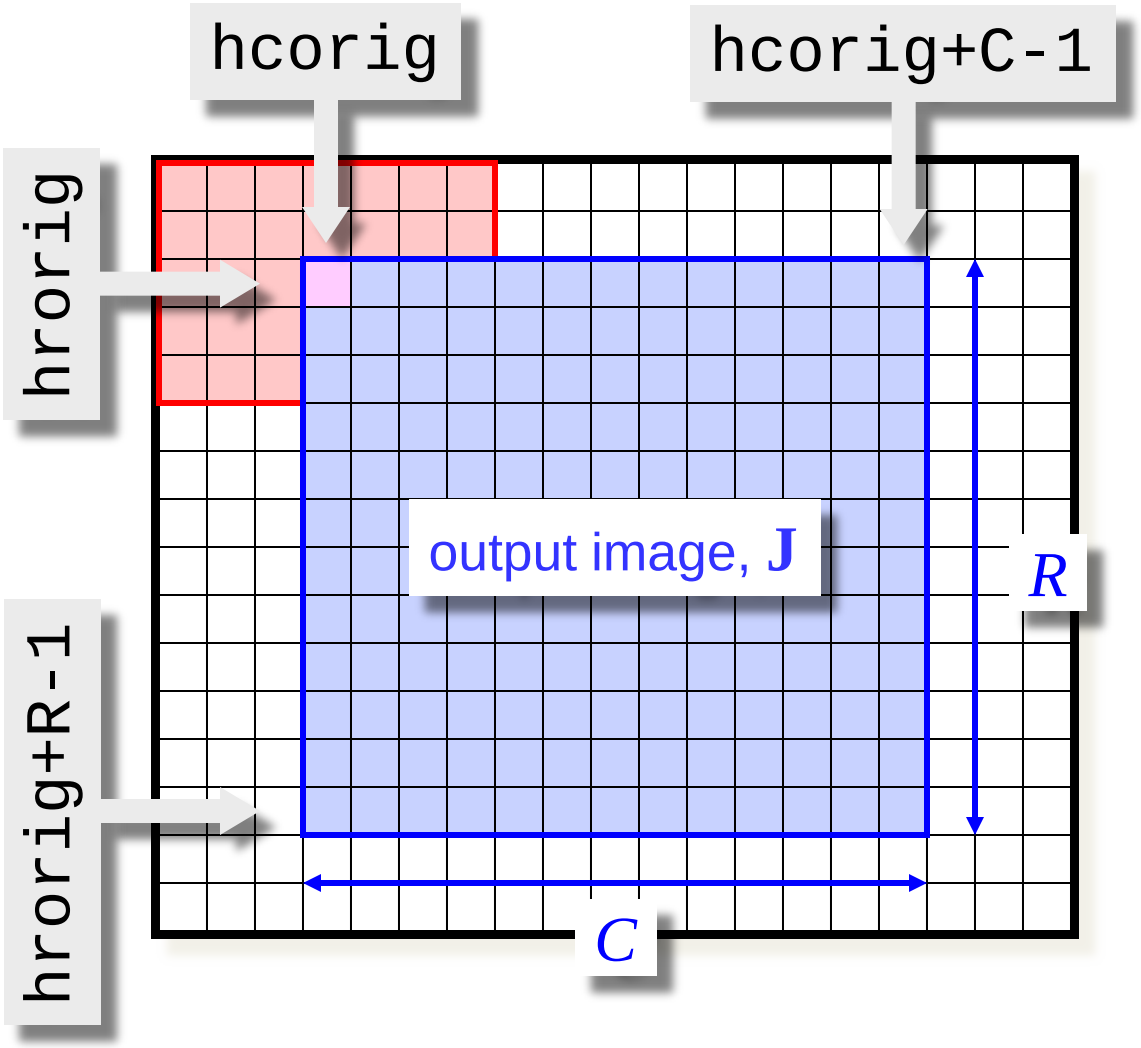
Image is shifted to mask location, multiplied by value, and accumulated.



h:6x6，cpad=3, rpad=3

模板原点（3+1,3+1）

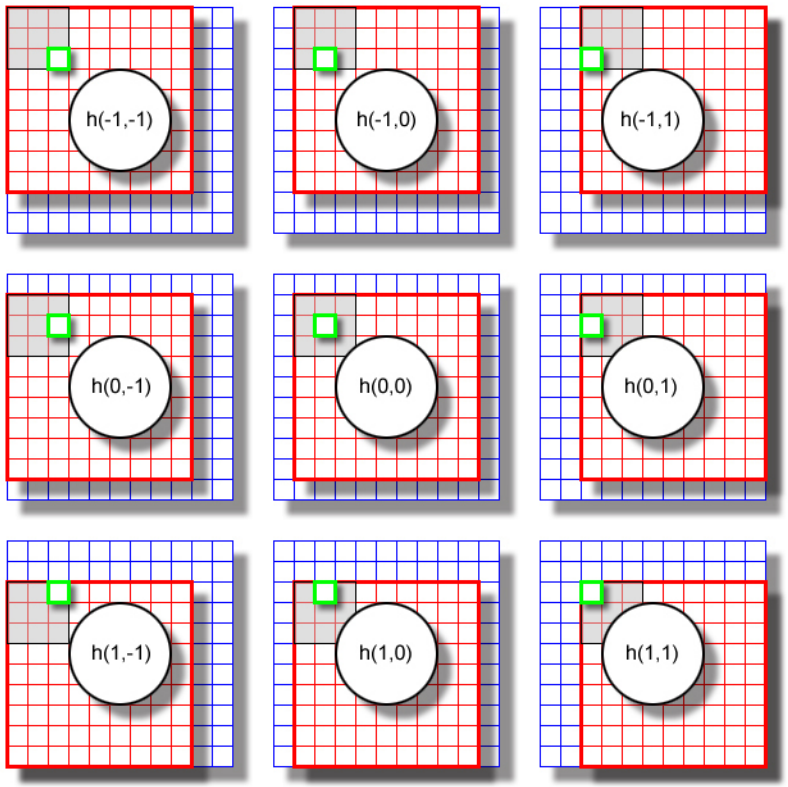
Convolution by Copying and Shifting the Image



When done, copy the output image from the accumulator starting at (hrorig, hcorig) and ending at (hrorig+R-1, hcorig+C-1)

Convolution by Copying, Multiplying, and Shifting the Image

For each element $\mathbf{h}(r_h,c_h)$ in weight matrix, **h,** image **I** is copied into a zero-padded image, **P**, starting at (r_h,c_h).
Each **P** is multiplied by the corresponding weight, $\mathbf{h}(r_h,c_h)$.
All the **P** images are summed pixel-wise then divided by the sum of the elements of **h**. The result is cropped out of the center of the accumulated **P**s.



original image, **I**

padded image, **P**

effective neighborhood

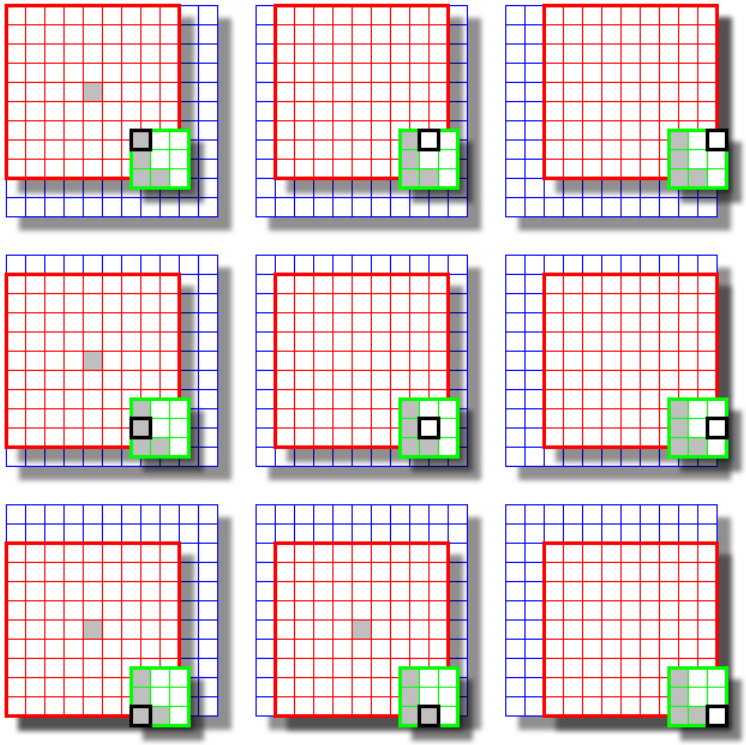
h(-1,-1)	h(-1,0)	h(-1,1)
h(0,-1)	h(0,0)	h(0,1)
h(1,-1)	h(1,0)	h(1,1)

weight matrix

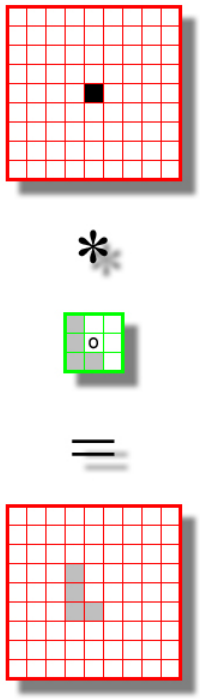
aligned pixels to be summed

weight for image ◯

Convolution by Copying, Multiplying, and Shifting the Image



original image, **I**
effective neighborhood
padded image, **P**



The original image has a black impulse at the center and zeros (white) elsewhere.

The weight matrix has a gray 'L' at its left and zeros (white) elsewhere.

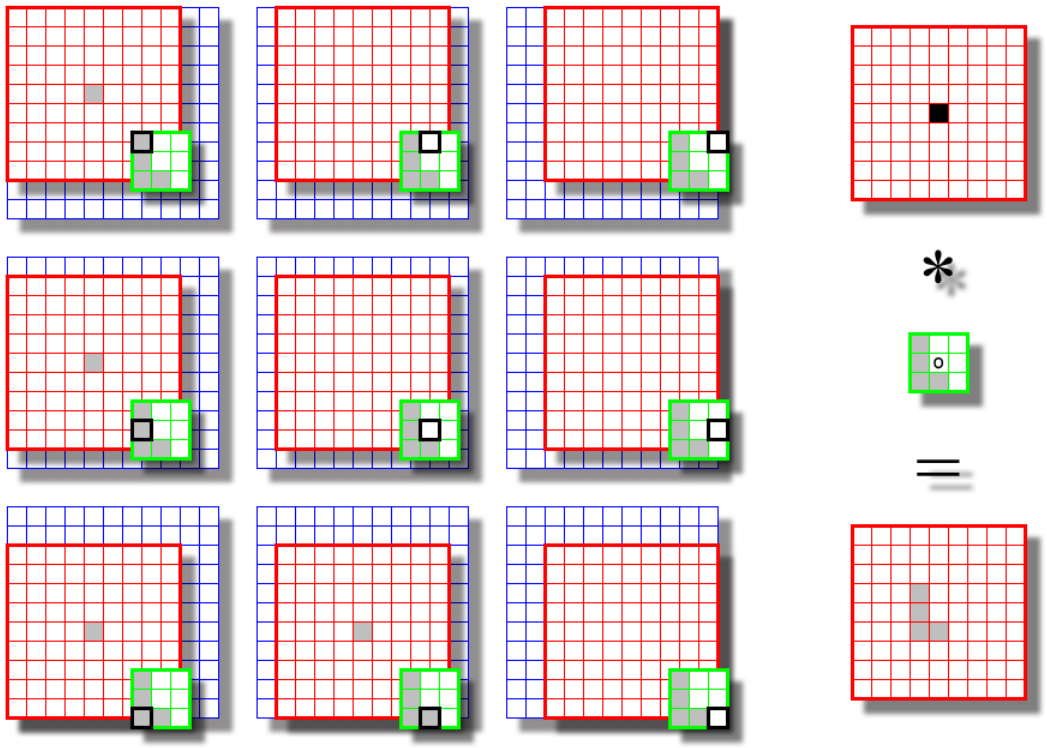
The resulting image has a copy of the weight matrix pegged to the impulse location.

In the result, the origin of the weight matrix coincides with the original location of the impulse.

Convolution by Copying, Multiplying, and Shifting the Image

Each copy of the (entire) image is multiplied by the value of the weight matrix in black square (here, white = 0) before being accumulated (pixelwise) in the padded image

The position of the black square relative the center of the weight matrix indicates the shift of the original image relative to the middle of the padded image.

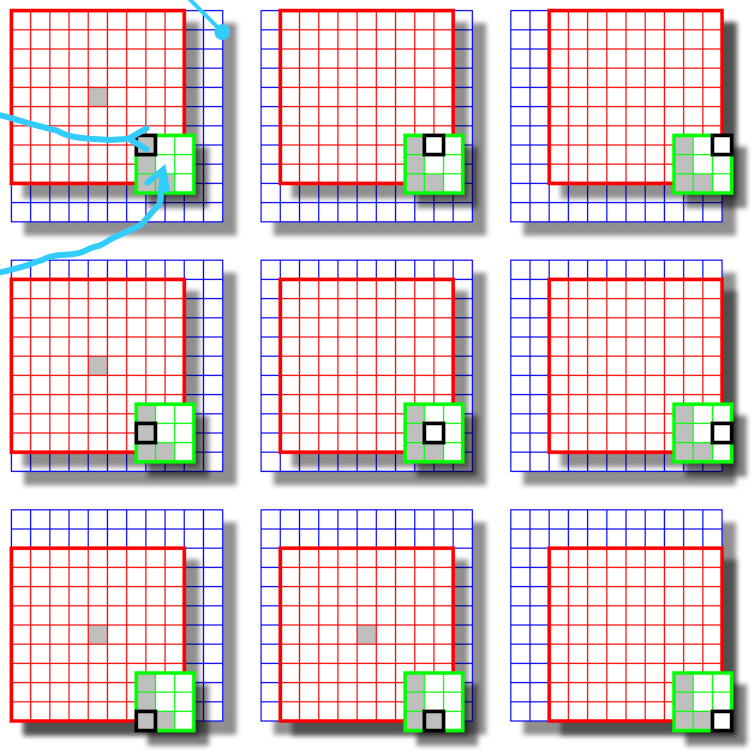


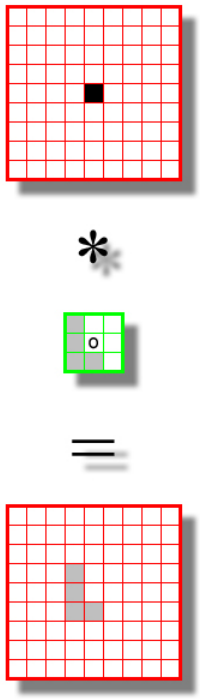
In this image, only the pixel in the center is nonzero so only it shows a result when the image is multiplied by a nonzero value

Convolution by Copying, Multiplying, and Shifting the Image

Each copy of the (entire) image is multiplied by the value of the weight matrix in black square (here, white = 0) before being accumulated (pixelwise) in the padded image

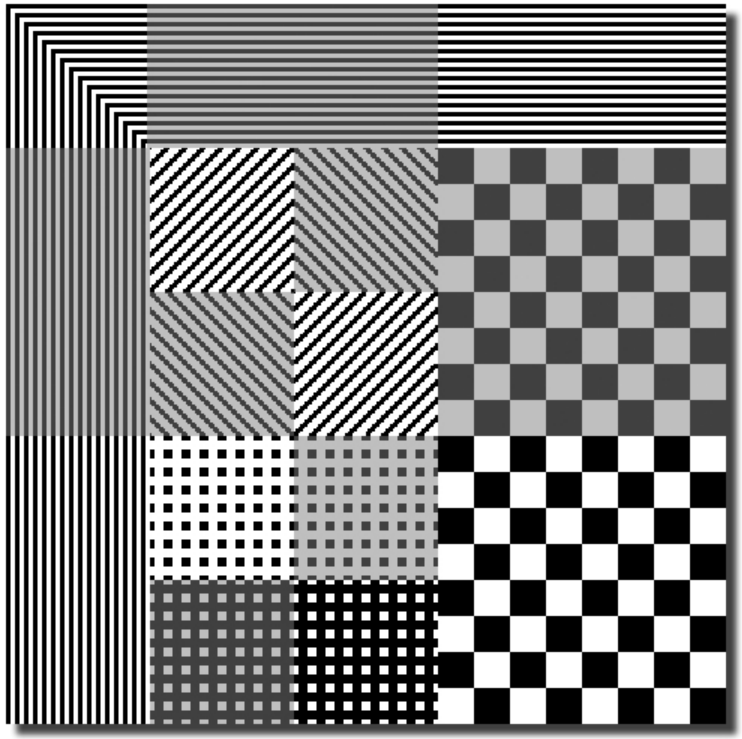
The position of the black square relative the center of the weight matrix indicates the shift of the original image relative to the middle of the padded image.





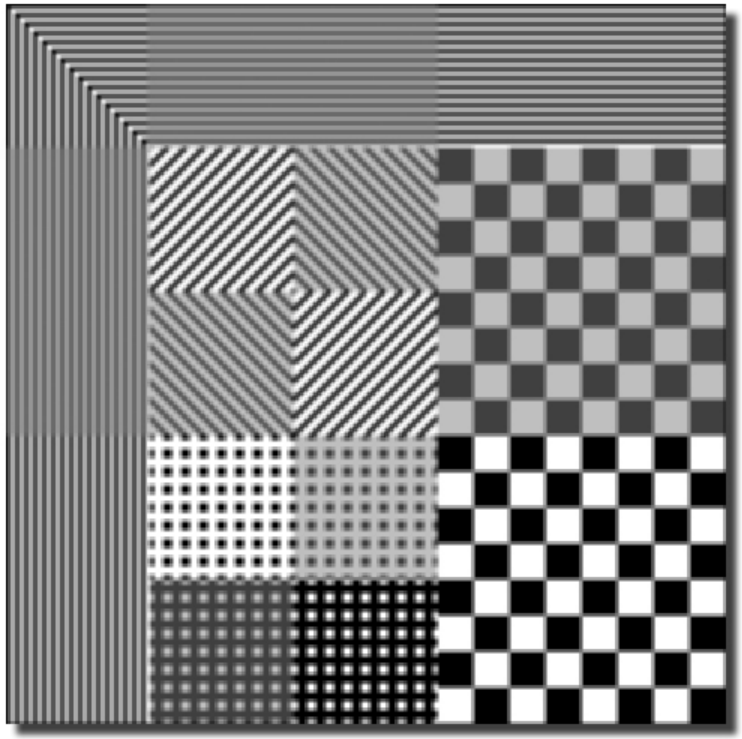
In this image, only the pixel in the center is nonzero so only it shows a result when the image is multiplied by a nonzero value

Convolution Examples: Original Images



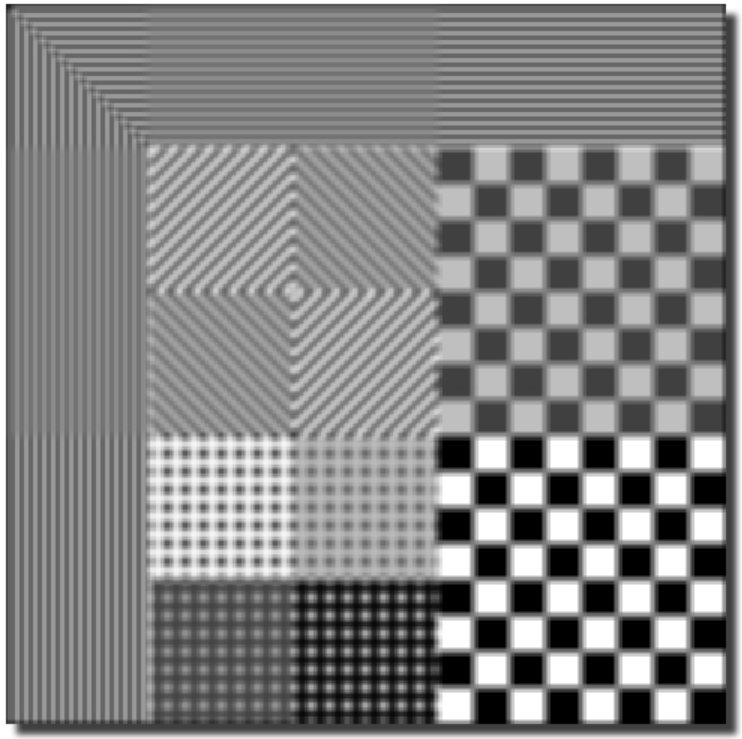
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Convolution Examples: 3×3 Blur



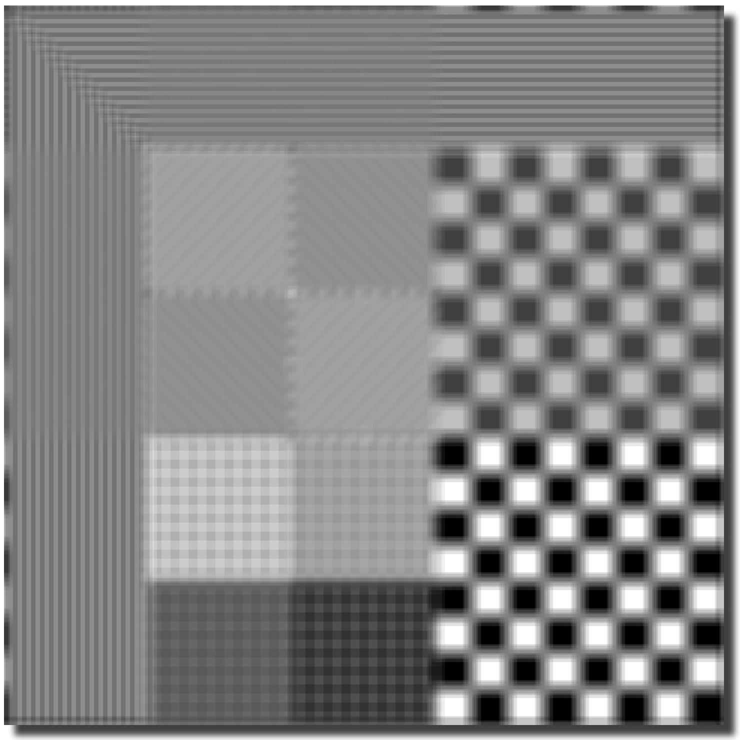
$$\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Convolution Examples: 5×5 Blur



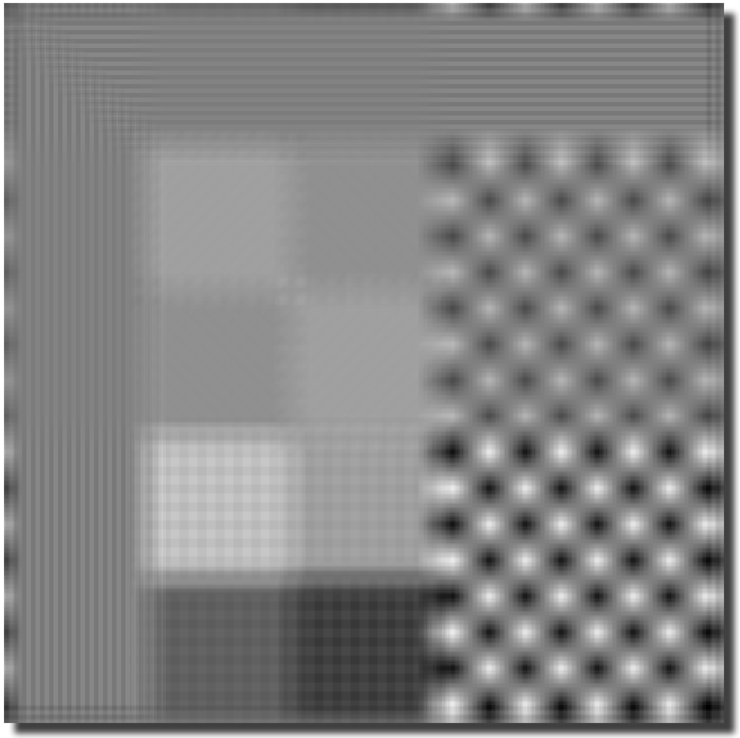
$$\frac{1}{81}\begin{bmatrix}1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\\1&1&1&1&1&1&1&1&1\end{bmatrix}$$

Convolution Examples: 9×9 Blur

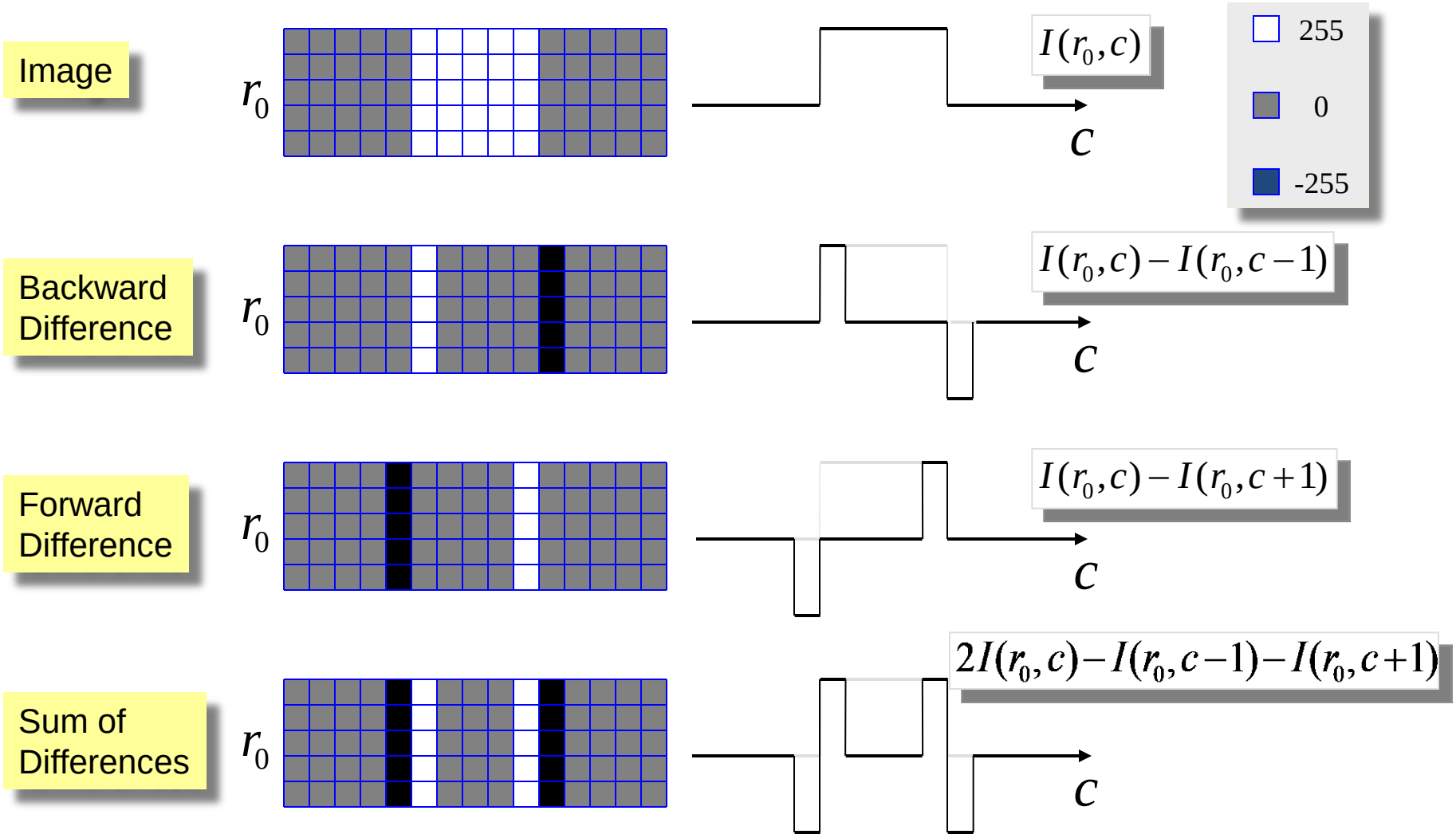


$$\frac{1}{289}\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

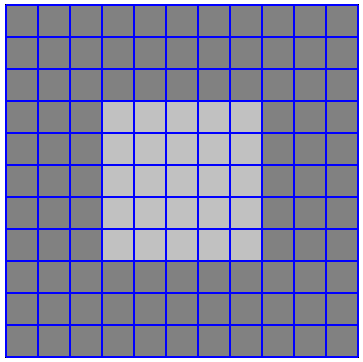
Convolution Examples: 17×17 Blur



Vertical Edge Detection

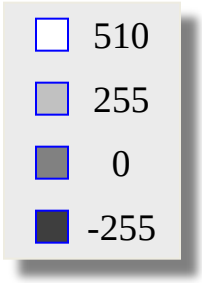


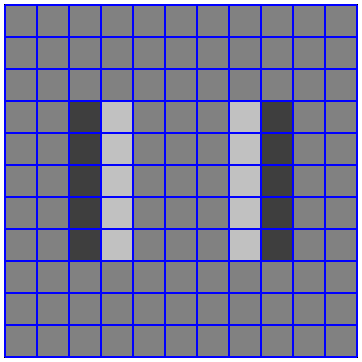
Symmetric Edge Detection （如何构造检测模板）



$$I(r,c)$$

用到3x3邻域

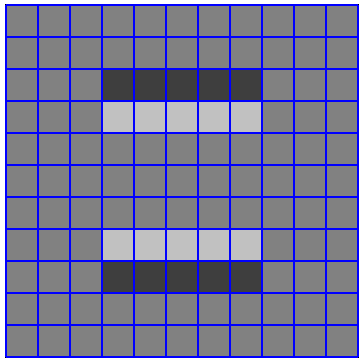




$$2I(r,c)-I(r,c-1)-I(r,c+1)$$

垂直边缘检测

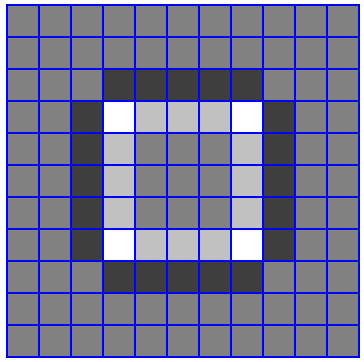
-1	2	-1



$$2I(r,c)-I(r-1,c)-I(r+1,c)$$

水平边缘检测

	-1	
	2	
	-1	

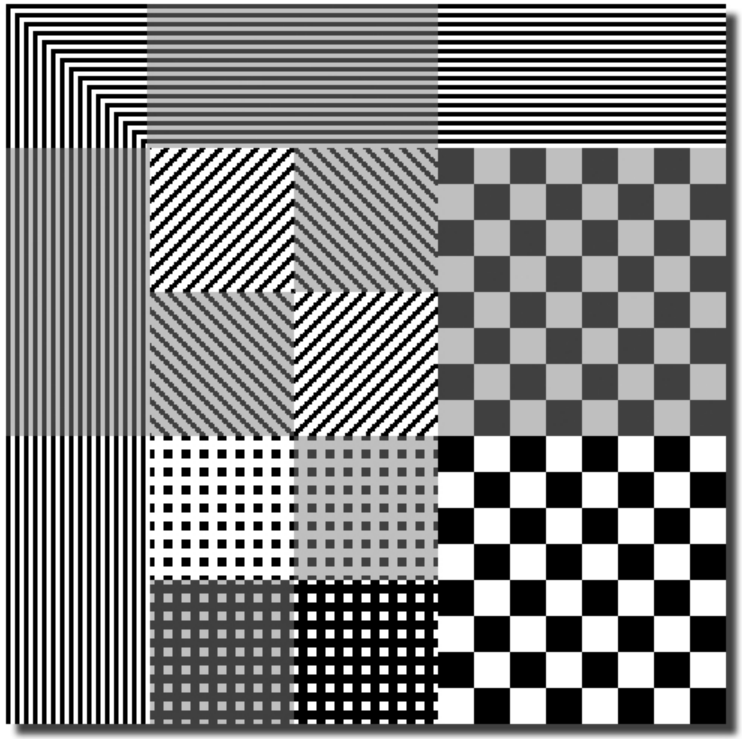


$$4I(r,c)-I(r-1,c)-I(r+1,c)-I(r,c-1)-I(r,c+1)$$

联合边缘检测

	-1	
-1	4	-1
	-1	

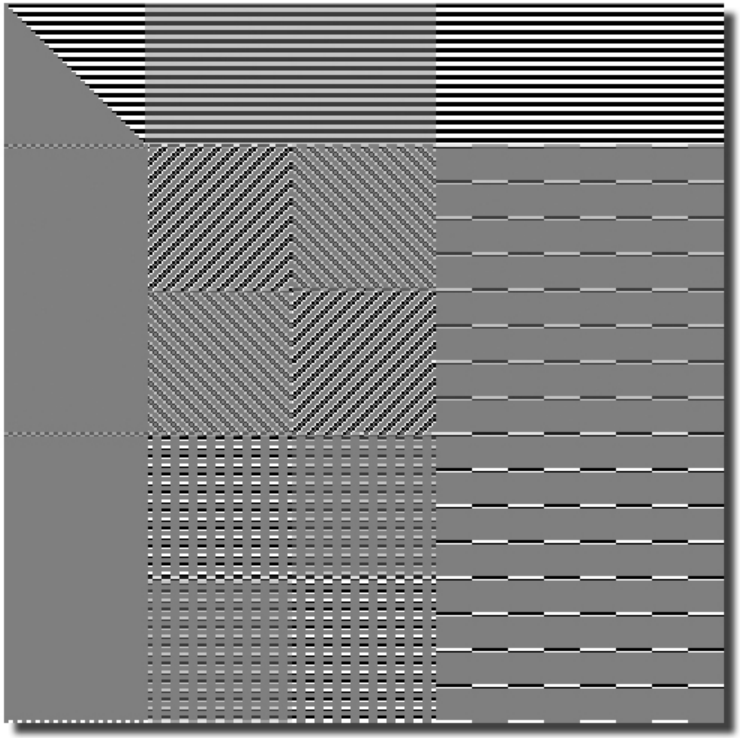
Convolution Examples: Original Images

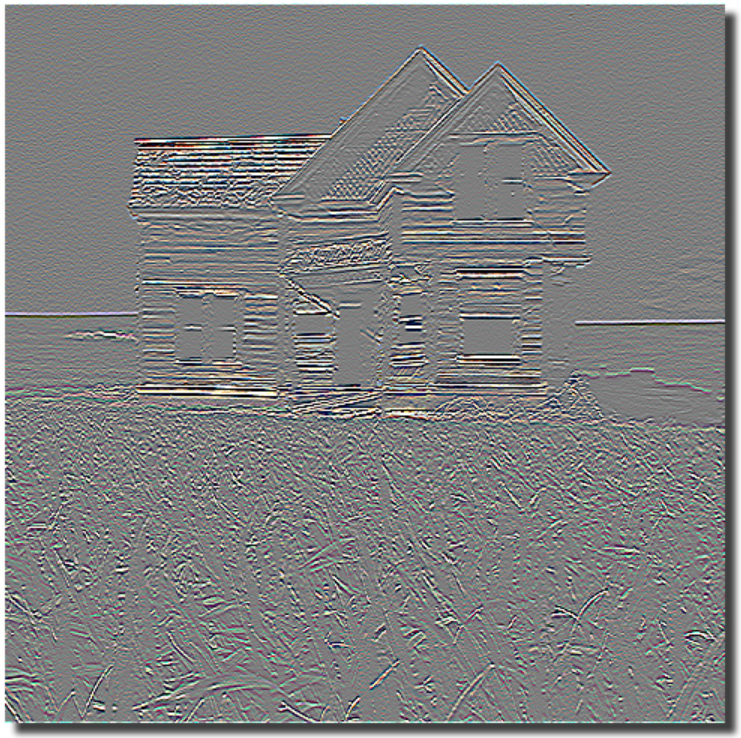


$$\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$$

对于水平边缘检测（垂直变化），其滤波器核是一维的，可以将图像分解为多个列，按列作一维滤波

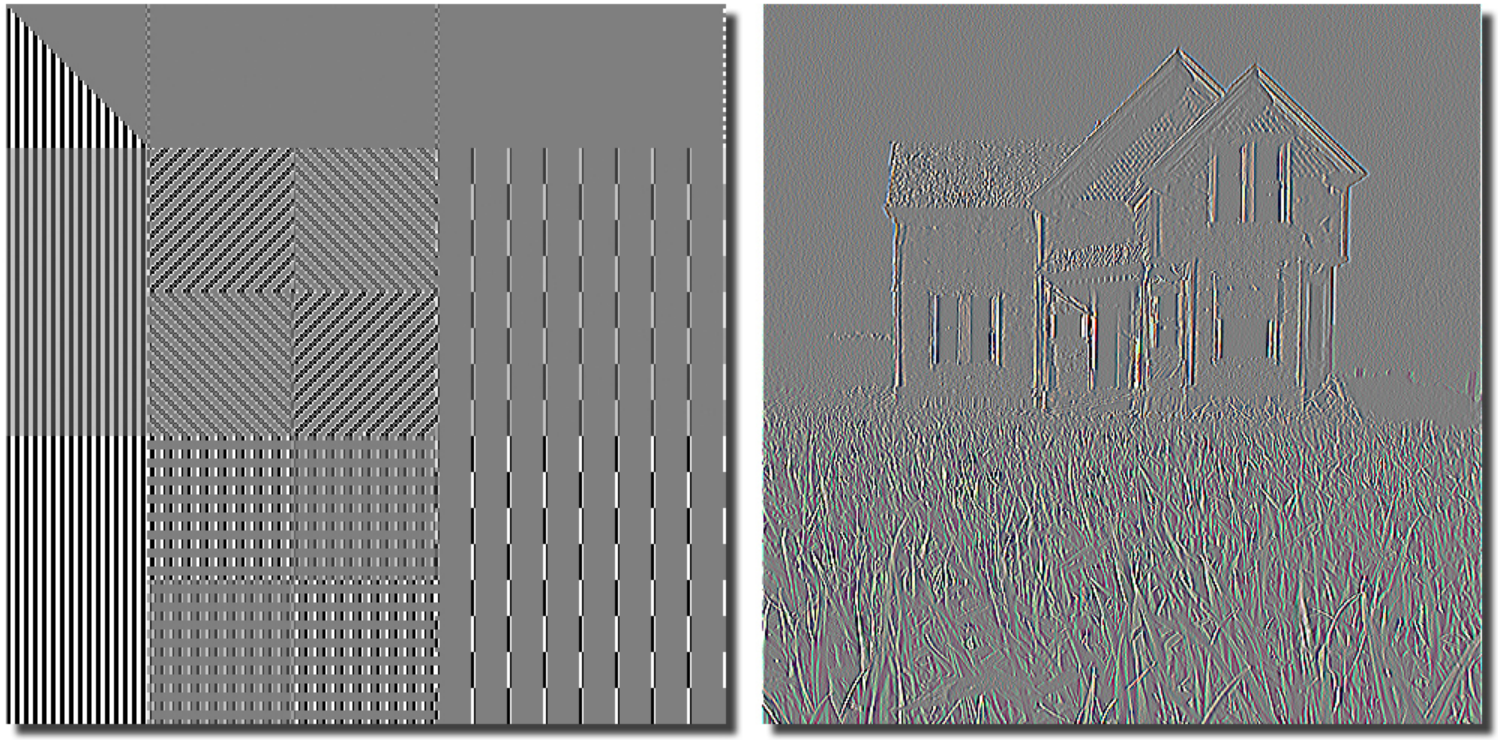
Convolution Examples: Vertical Difference





$[-1 \quad 2 \quad -1]$ 按行作一维滤波

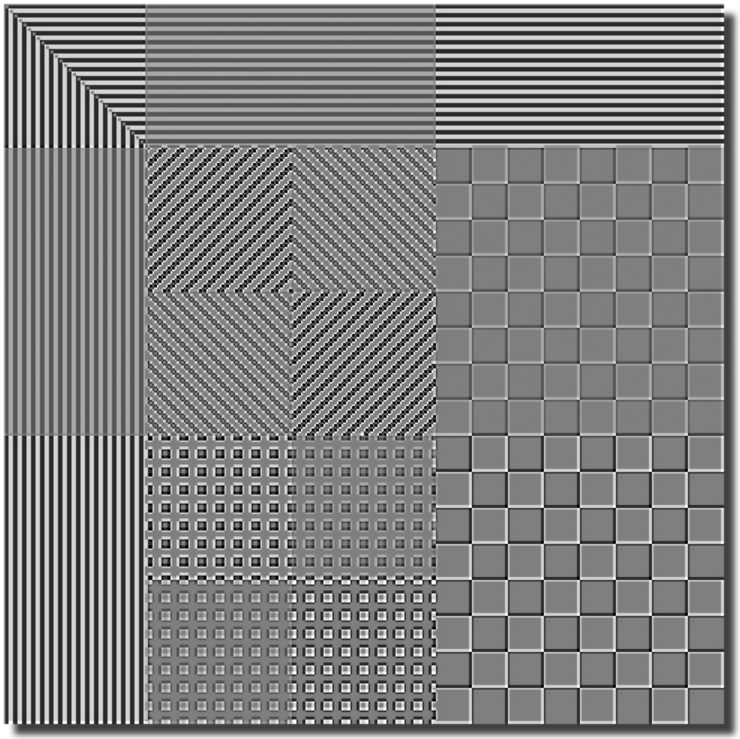
Convolution Examples: Horizontal Difference



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

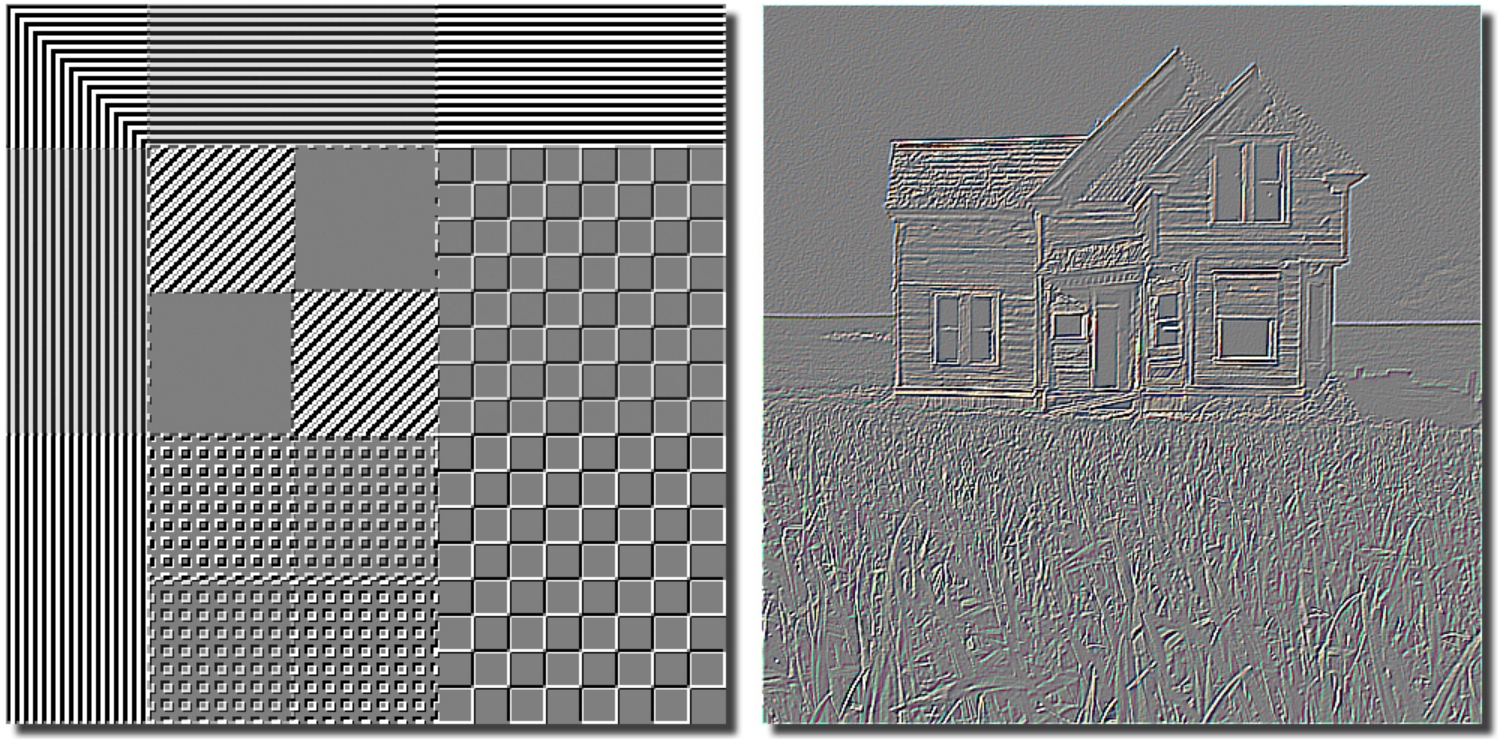
作二维滤波

Convolution Examples: H + V Diff.

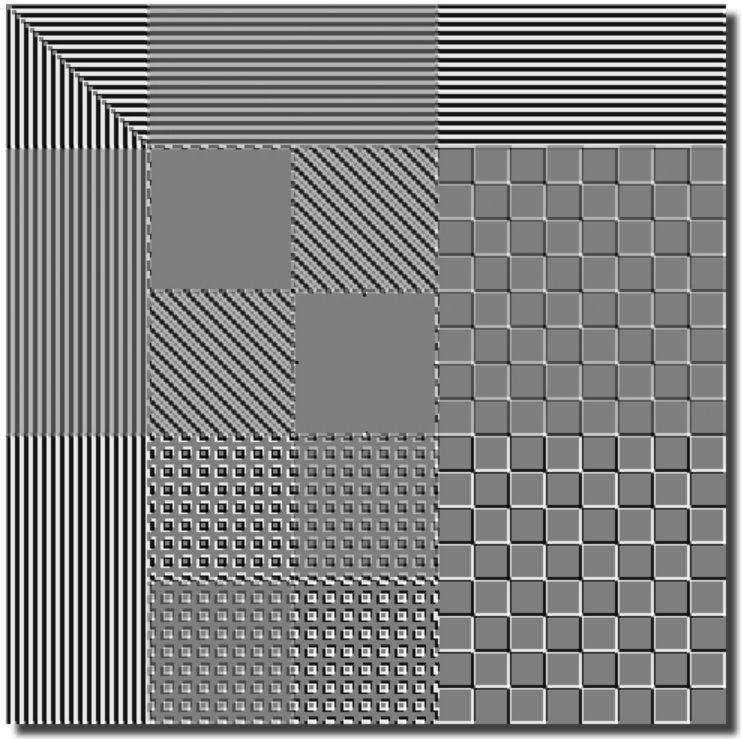


$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Convolution Examples: Diagonal Difference

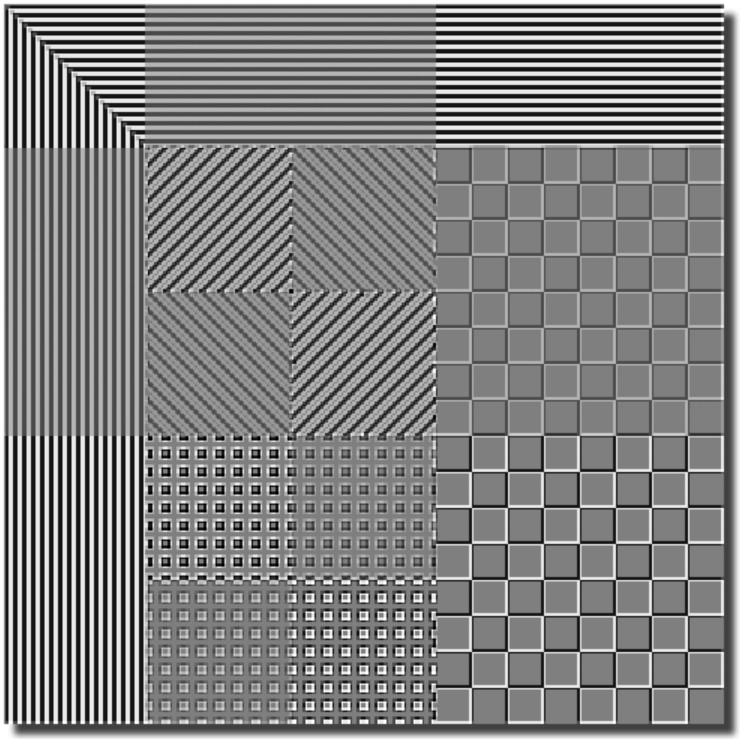


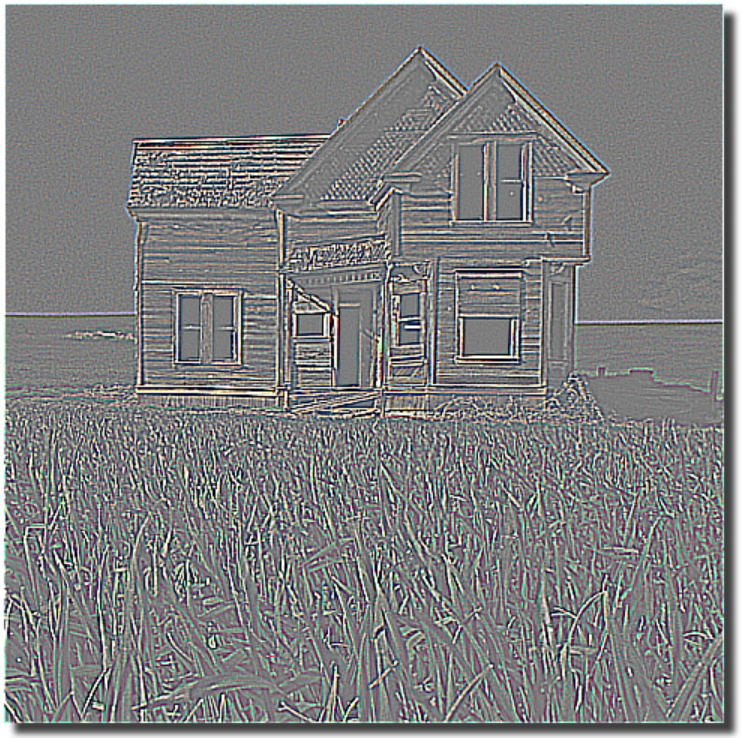
Convolution Examples: Diagonal Difference



$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

Convolution Examples: D + D Difference

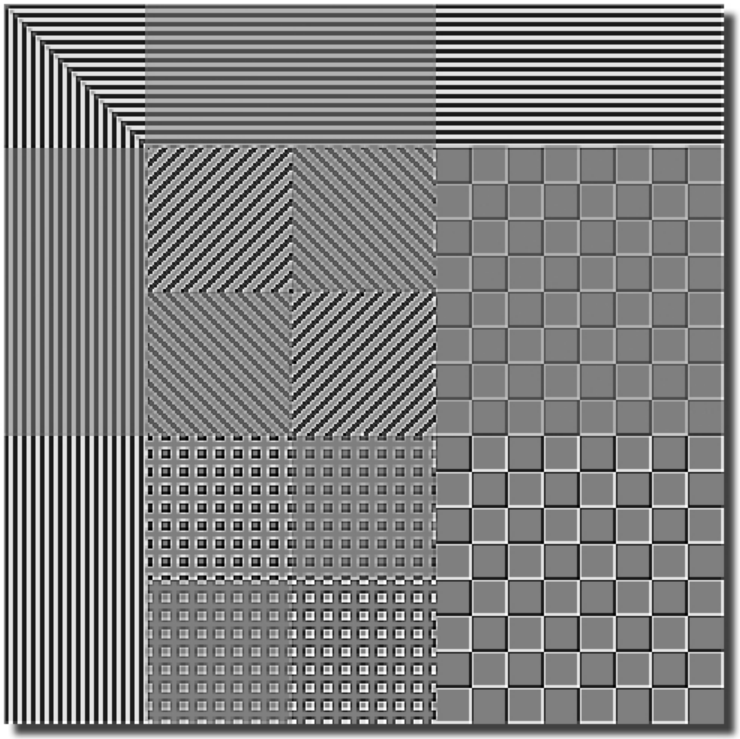




$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

注意：高通滤波器各元素的和为0

Convolution Examples: H + V + D Diff.



Thanks!