Lecture 3. Filters and Convolutions

Convolution and correlation

Juan Carlos Niebles and Jiajun Wu

CS131 Computer Vision: Foundations and Applications

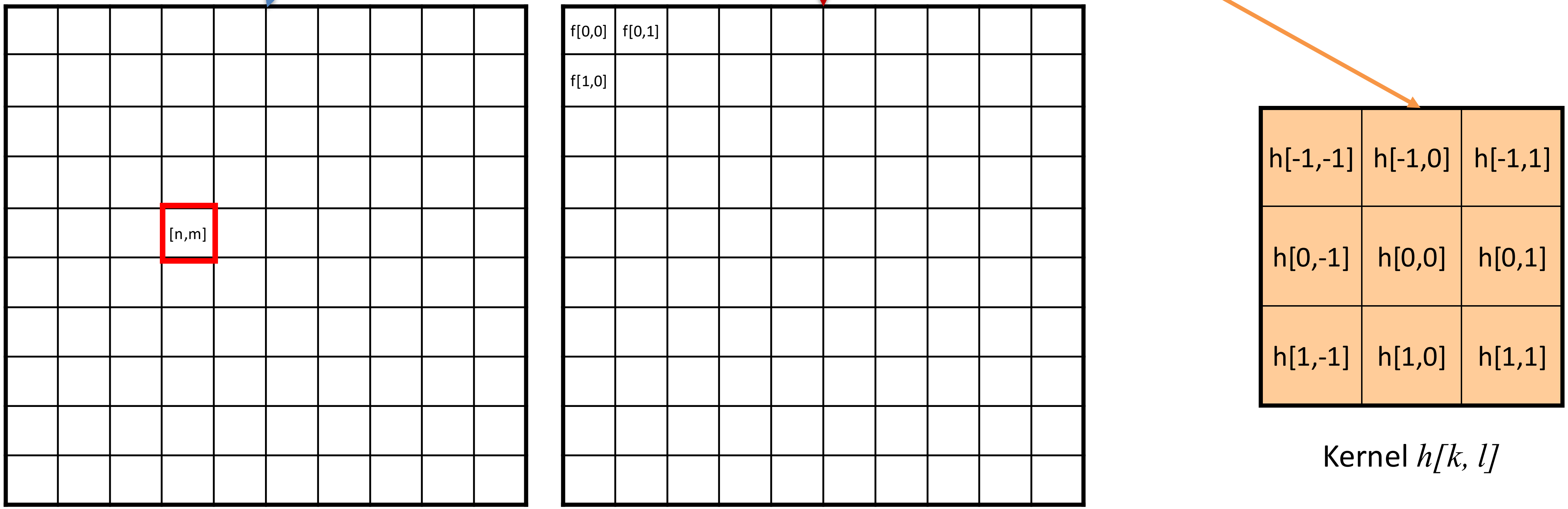
What we will learn today?

- Convolution
- Correlation

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

2D Discrete Convolution

$$\boxed{f[n,m]*h[n,m]} = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} \boxed{f[k,l]}\cdot \boxed{h[n-k,m-l]}$$



Output $f*h$

Image $f[k, l]$

Kernel $h[k, l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot \boxed{h[n-k, m-l]}$$

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]

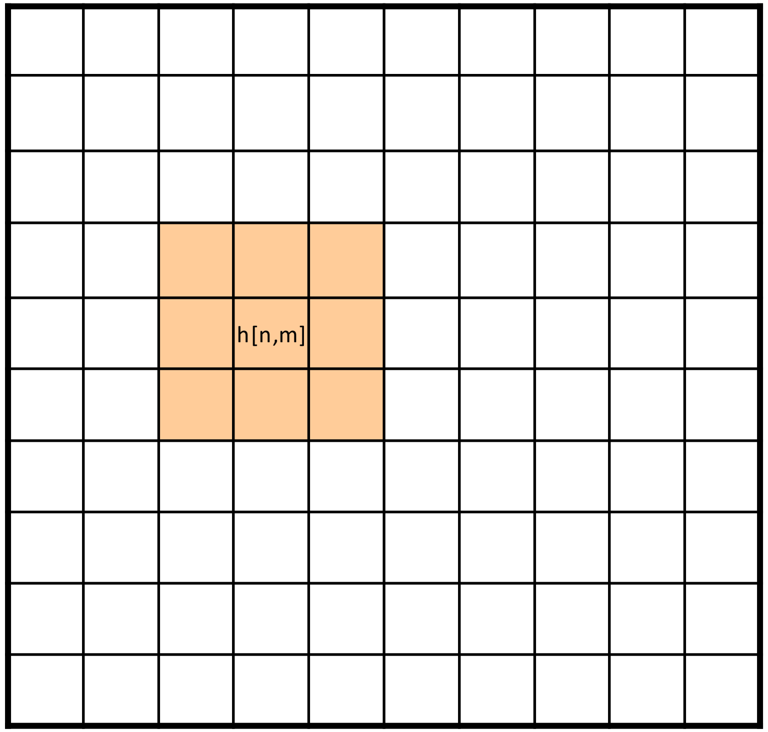
Kernel *h[k, l]*

Fold

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]

Kernel *h[-k, -l]*

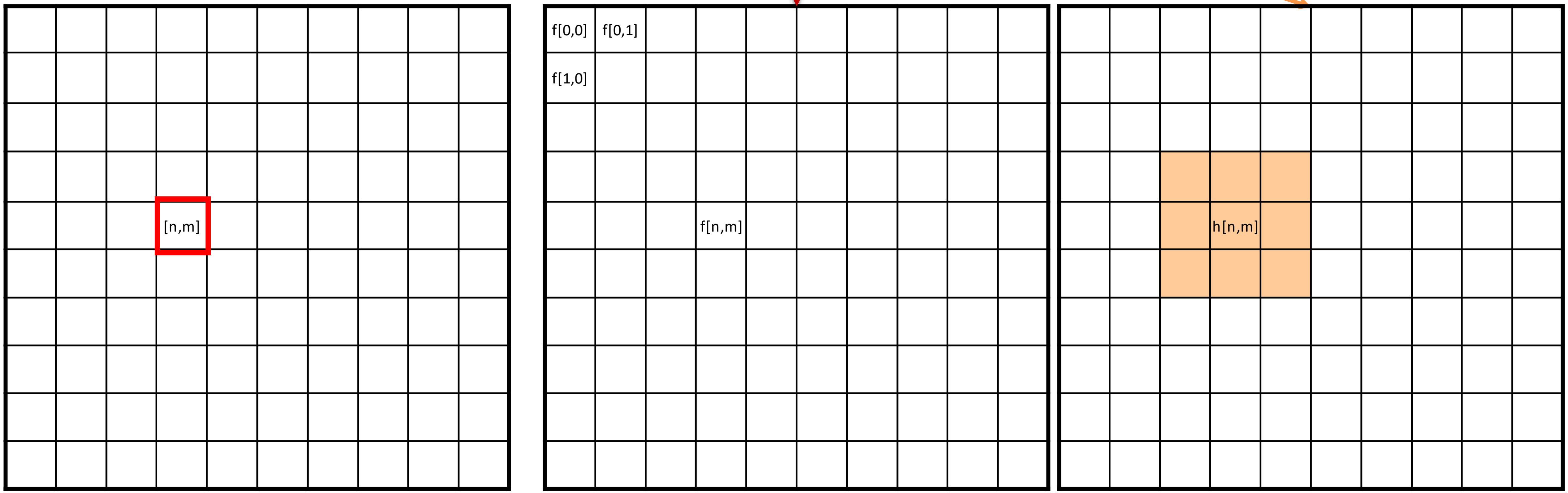
Shift



Kernel *h[n-k, m-l]*

2D Discrete Convolution

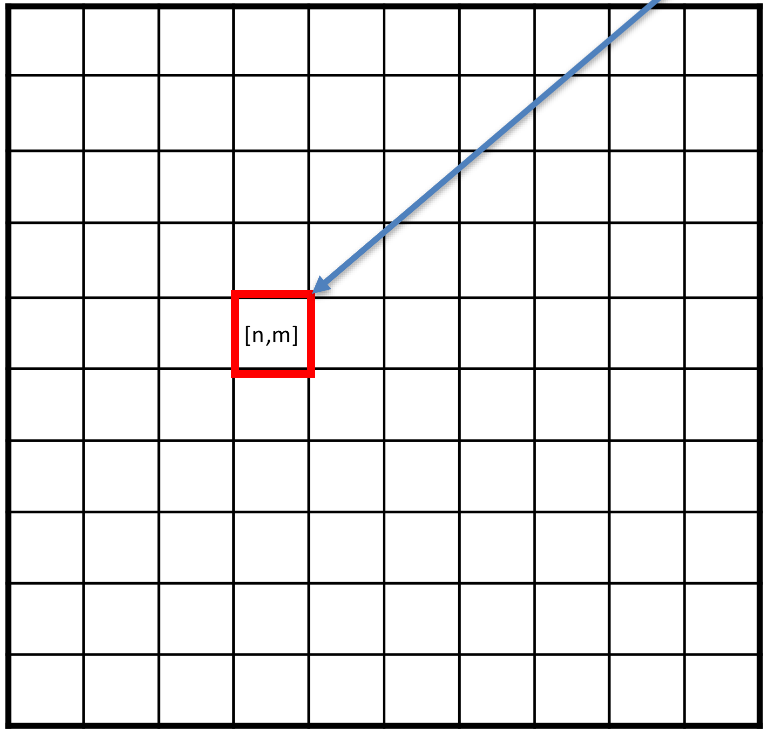
$$f[n,m]*h[n,m]=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} f[k,l]\cdot h[n-k,m-l]$$



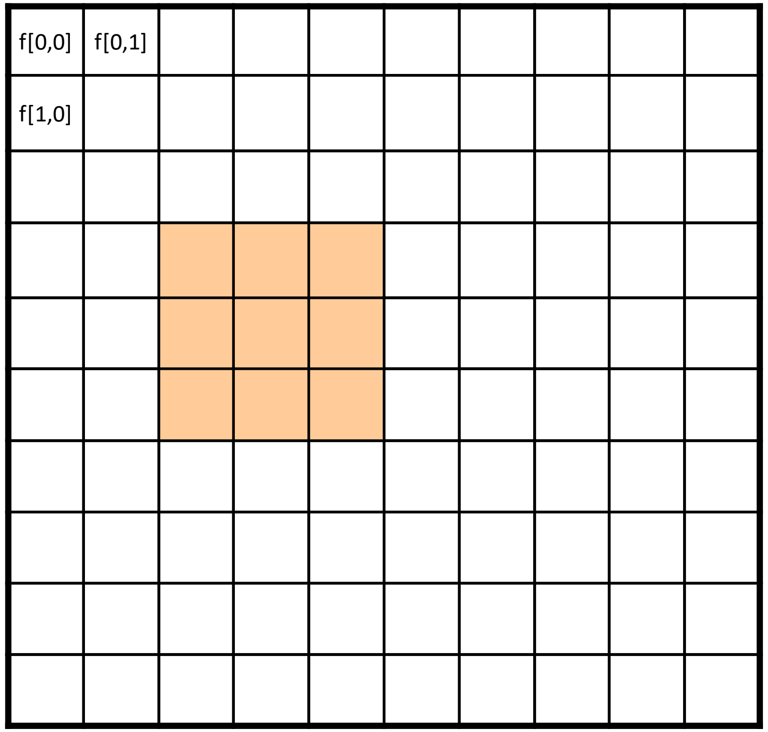
Output $f*h$ | Image $f[k, l]$ | Kernel $h[n-k, m-l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \boxed{f[k,l] \cdot h[n-k, m-l]}$$



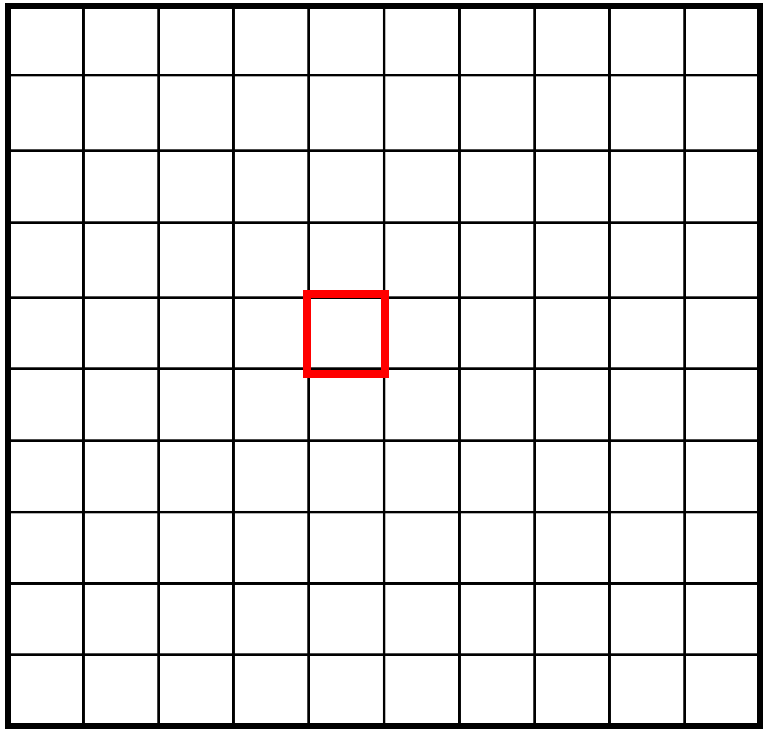
Output $f*h$



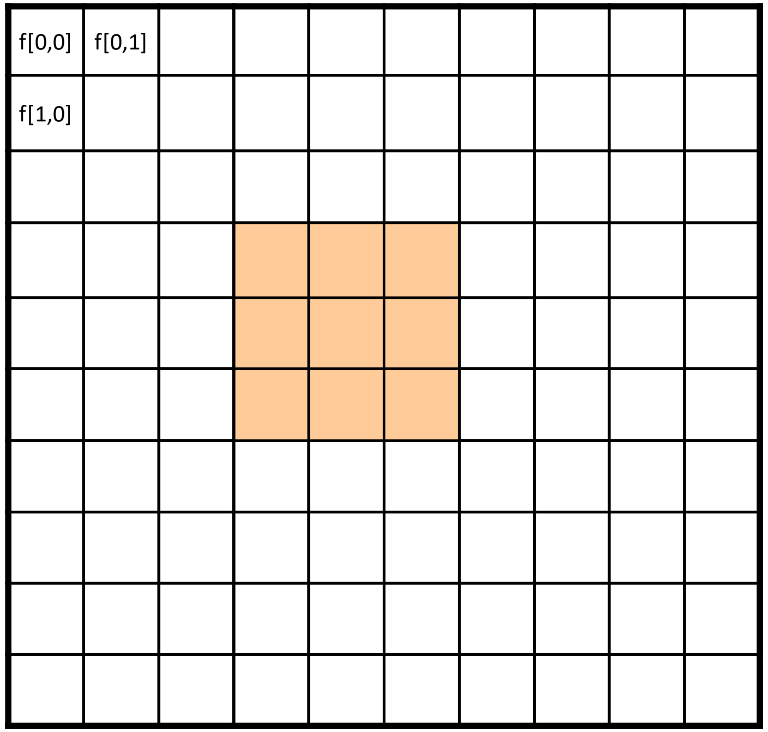
Element-wise multiplication
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



Output $f*h$

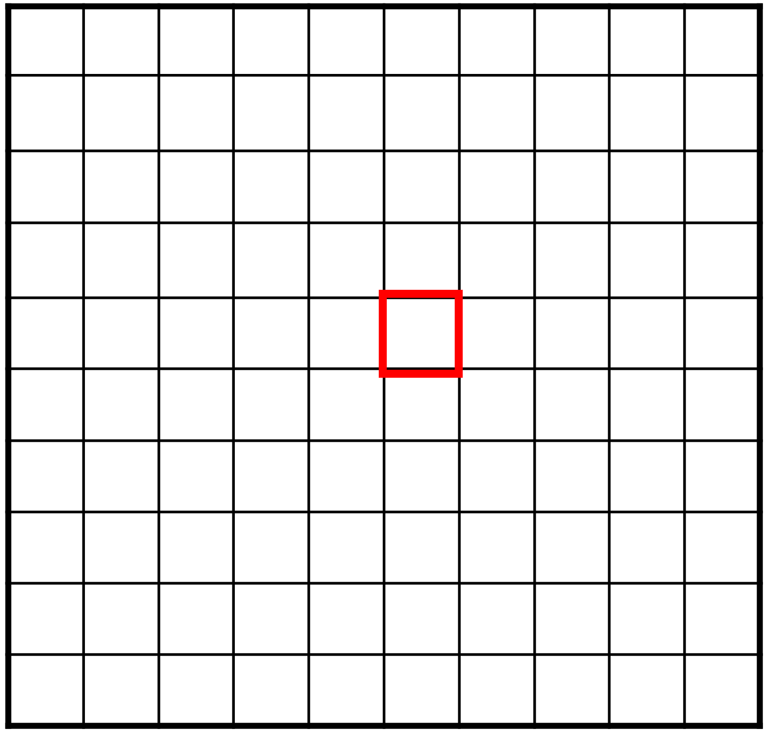


Element-wise multiplication

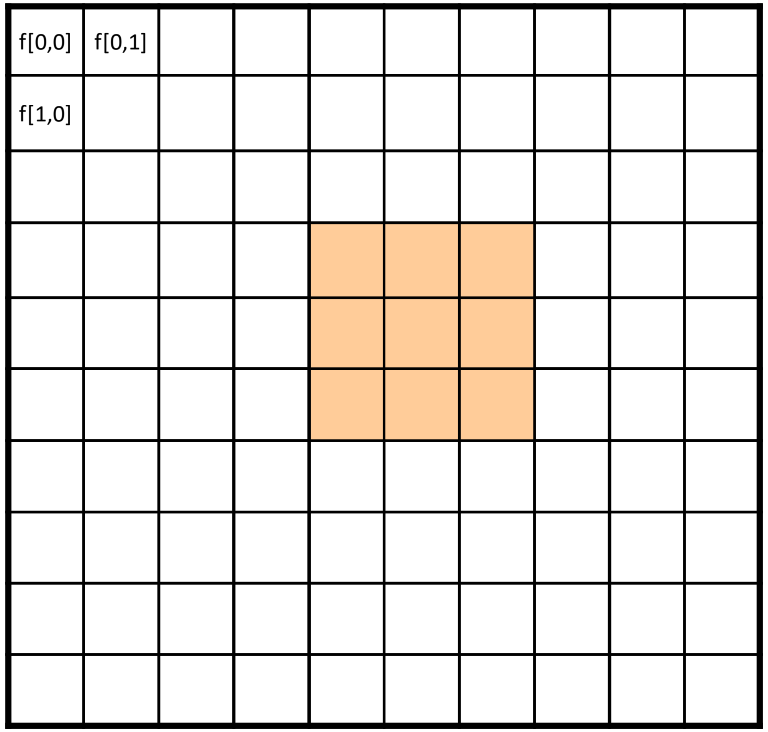
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



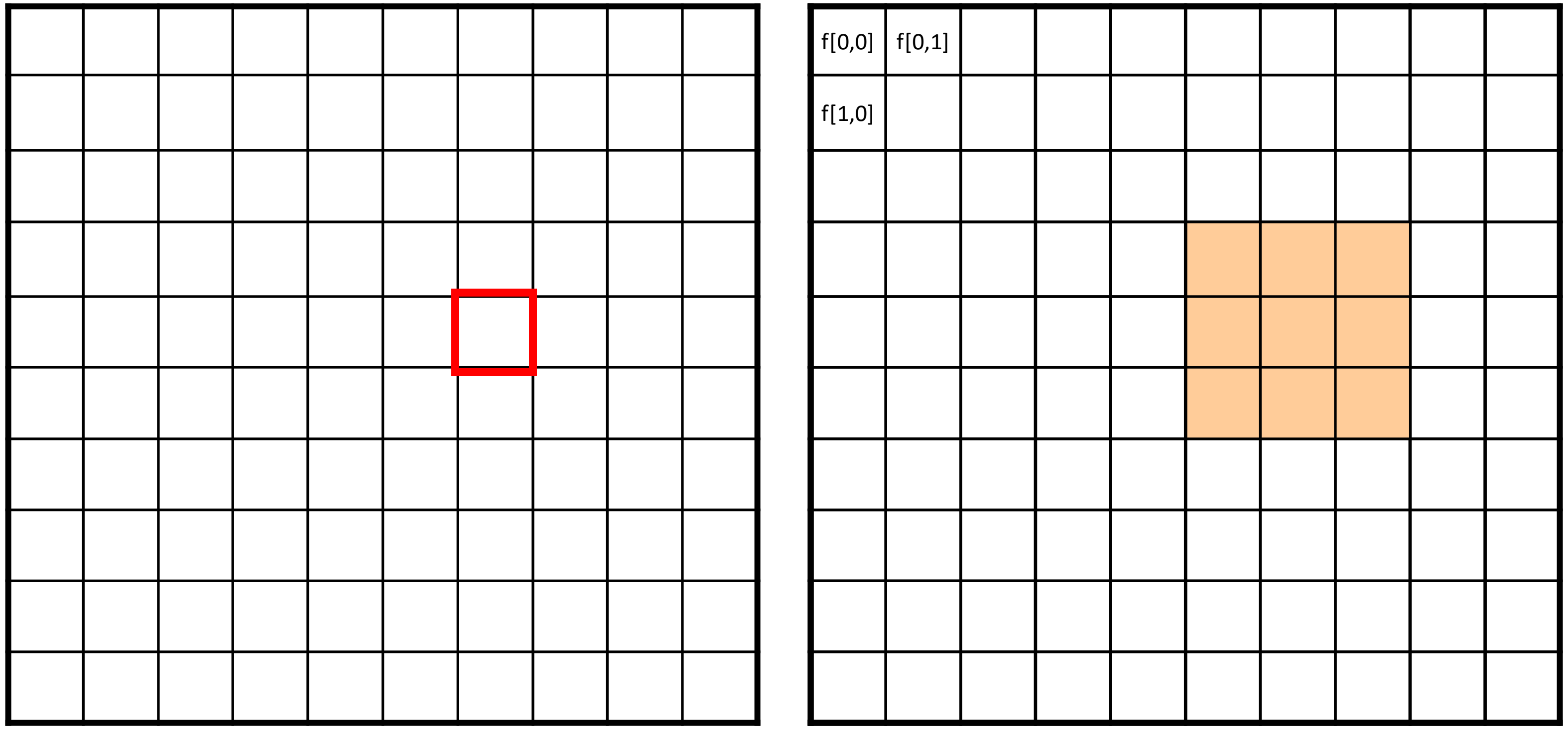
Output $f*h$



Element-wise multiplication
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$



Output $f*h$

Element-wise multiplication
Image $f[k, l]$ • Kernel $h[n\text{-}k, m\text{-}l]$

2D Discrete Convolution

$$f[n,m] * h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n-k, m-l]$$

Algorithm:

- Fold $h[k, l]$ about origin to form $h[-k, -l]$
- Shift the folded results by n, m to form $h[n-k, m-l]$
- Multiply $h[n-k, m-l]$ by $f[k, l]$
- Sum over all k, l
- Repeat for every n, m

2D convolution example

1	2	3
4	5	6
7	8	9

Input

n \ m	-1	0	1
-1	-1	-2	-1
0	0	0	0
1	1	2	1

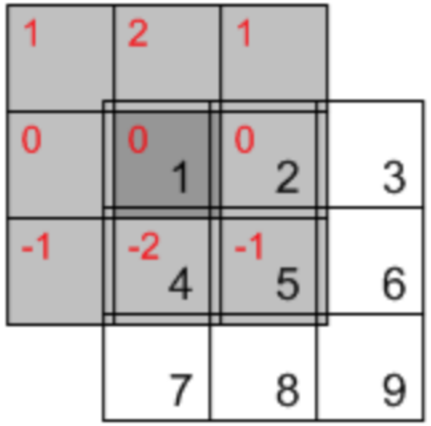
Kernel

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



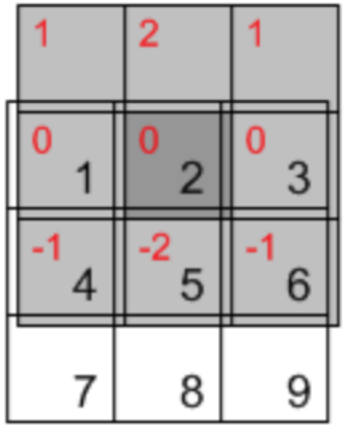
$$
\begin{aligned}
&= x[-1,-1]\cdot h[1,1] + x[0,-1]\cdot h[0,1] + x[1,-1]\cdot h[-1,1] \\
&\quad + x[-1,0]\cdot h[1,0] + x[0,0]\cdot h[0,0] + x[1,0]\cdot h[-1,0] \\
&\quad + x[-1,1]\cdot h[1,-1] + x[0,1]\cdot h[0,-1] + x[1,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 0\cdot 0 + 1\cdot 0 + 2\cdot 0 + 0\cdot(-1) + 4\cdot(-2) + 5\cdot(-1) = -13
\end{aligned}
$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



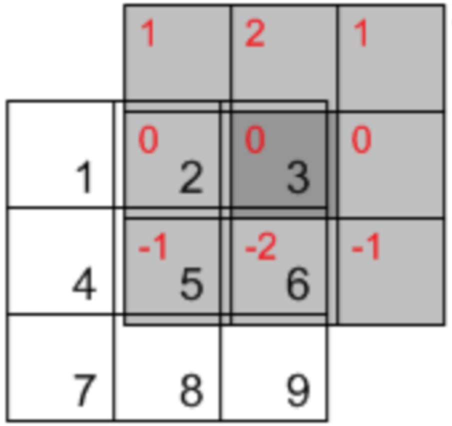
$$
\begin{aligned}
&= x[0,-1]\cdot h[1,1] + x[1,-1]\cdot h[0,1] + x[2,-1]\cdot h[-1,1] \\
&\quad + x[0,0]\cdot h[1,0] + x[1,0]\cdot h[0,0] + x[2,0]\cdot h[-1,0] \\
&\quad + x[0,1]\cdot h[1,-1] + x[1,1]\cdot h[0,-1] + x[2,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 1\cdot 0 + 2\cdot 0 + 3\cdot 0 + 4\cdot(-1) + 5\cdot(-2) + 6\cdot(-1) = -20
\end{aligned}
$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



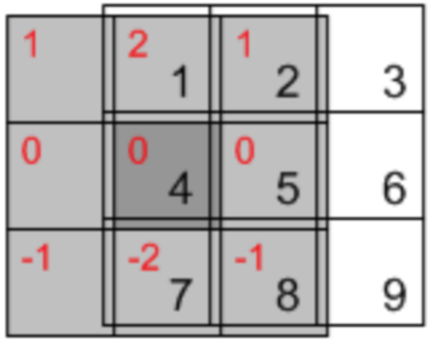
$$
\begin{aligned}
&= x[1,-1]\cdot h[1,1] + x[2,-1]\cdot h[0,1] + x[3,-1]\cdot h[-1,1] \\
&\quad + x[1,0]\cdot h[1,0] + x[2,0]\cdot h[0,0] + x[3,0]\cdot h[-1,0] \\
&\quad + x[1,1]\cdot h[1,-1] + x[2,1]\cdot h[0,-1] + x[3,1]\cdot h[-1,-1] \\
&= 0\cdot 1 + 0\cdot 2 + 0\cdot 1 + 2\cdot 0 + 3\cdot 0 + 0\cdot 0 + 5\cdot(-1) + 6\cdot(-2) + 0\cdot(-1) = -17
\end{aligned}
$$

-13	-20	$\boxed{-17}$
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



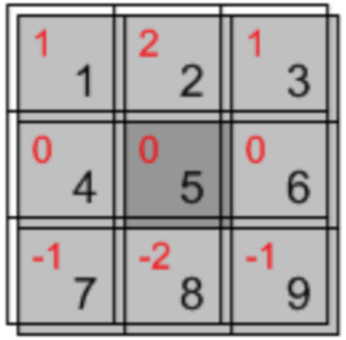
$$
\begin{aligned}
&= x[-1,0]\cdot h[1,1] + x[0,0]\cdot h[0,1] + x[1,0]\cdot h[-1,1] \\
&\quad + x[-1,1]\cdot h[1,0] + x[0,1]\cdot h[0,0] + x[1,1]\cdot h[-1,0] \\
&\quad + x[-1,2]\cdot h[1,-1] + x[0,2]\cdot h[0,-1] + x[1,2]\cdot h[-1,-1] \\
&= 0\cdot 1 + 1\cdot 2 + 2\cdot 1 + 0\cdot 0 + 4\cdot 0 + 5\cdot 0 + 0\cdot(-1) + 7\cdot(-2) + 8\cdot(-1) = -18
\end{aligned}
$$

-13	-20	-17
$\boxed{-18}$	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



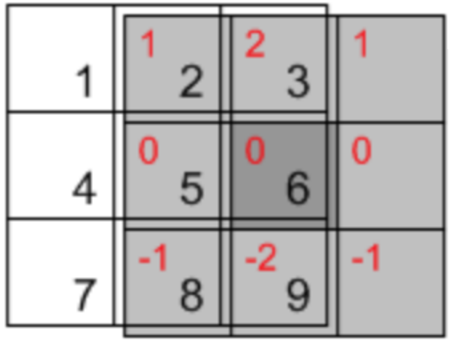
$$
\begin{aligned}
&= x[0,0]\cdot h[1,1] + x[1,0]\cdot h[0,1] + x[2,0]\cdot h[-1,1] \\
&\quad + x[0,1]\cdot h[1,0] + x[1,1]\cdot h[0,0] + x[2,1]\cdot h[-1,0] \\
&\quad + x[0,2]\cdot h[1,-1] + x[1,2]\cdot h[0,-1] + x[2,2]\cdot h[-1,-1] \\
&= 1\cdot 1 + 2\cdot 2 + 3\cdot 1 + 4\cdot 0 + 5\cdot 0 + 6\cdot 0 + 7\cdot(-1) + 8\cdot(-2) + 9\cdot(-1) = -24
\end{aligned}
$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

Slide credit: Song Ho Ahn

2D convolution example



$$
\begin{aligned}
&= x[1,0]\cdot h[1,1] + x[2,0]\cdot h[0,1] + x[3,0]\cdot h[-1,1] \\
&\quad + x[1,1]\cdot h[1,0] + x[2,1]\cdot h[0,0] + x[3,1]\cdot h[-1,0] \\
&\quad + x[1,2]\cdot h[1,-1] + x[2,2]\cdot h[0,-1] + x[3,2]\cdot h[-1,-1] \\
&= 2\cdot 1 + 3\cdot 2 + 0\cdot 1 + 5\cdot 0 + 6\cdot 0 + 0\cdot 0 + 8\cdot(-1) + 9\cdot(-2) + 0\cdot(-1) = -18
\end{aligned}
$$

-13	-20	-17
-18	-24	$\boxed{-18}$
13	20	17

Output

Slide credit: Song Ho Ahn

Convolution in 2D - examples

Original

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \ ?$$

Convolution in 2D - examples

Original

$*$

•0	•0	•0
•0	•1	•0
•0	•0	•0

$=$

Filtered
(no change)

Convolution in 2D - examples

Original

$$* \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad = \quad ?$$

Convolution in 2D - examples

Original

$*$

0	0	0
0	0	1
0	0	0

$=$

Shifted right
By 1 pixel

Convolution in 2D - examples

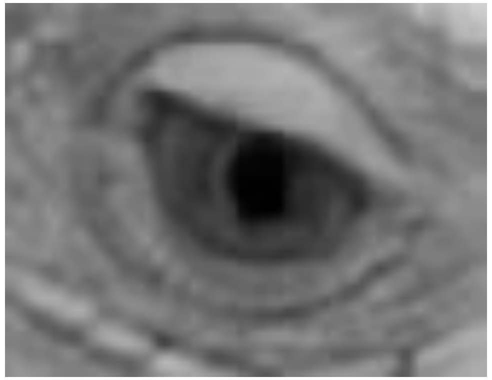
Original

$$* \frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} = ?$$

Convolution in 2D - examples

Original

$$\ast \; \frac{1}{9} \begin{bmatrix} \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \\ \bullet 1 & \bullet 1 & \bullet 1 \end{bmatrix} =$$



Blur (with a box filter)

Convolution in 2D - examples

Original

$$ * \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) = ? $$

(Note that filter sums to 1)

$$ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \overbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}^{\text{“details of the image”}} $$

What does blurring take away?

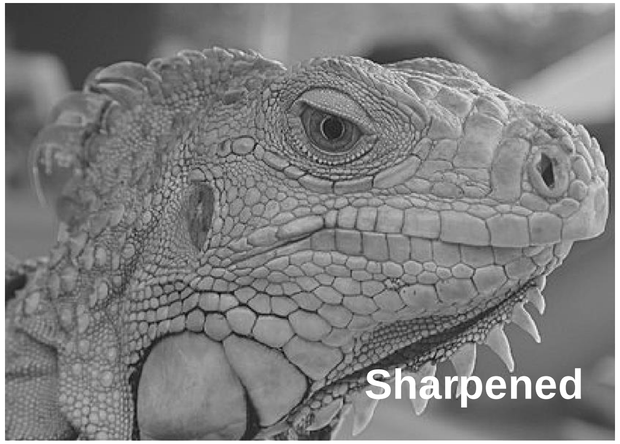
−

=

- Let’s add it back:

+

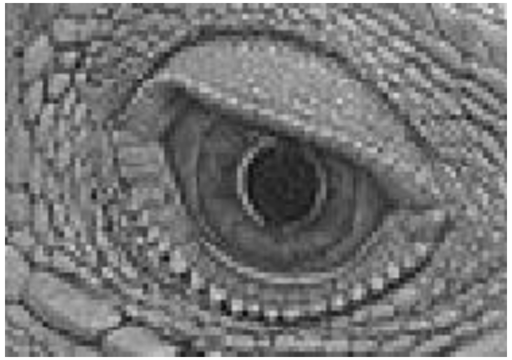
=



Convolution in 2D – Sharpening filter

Original

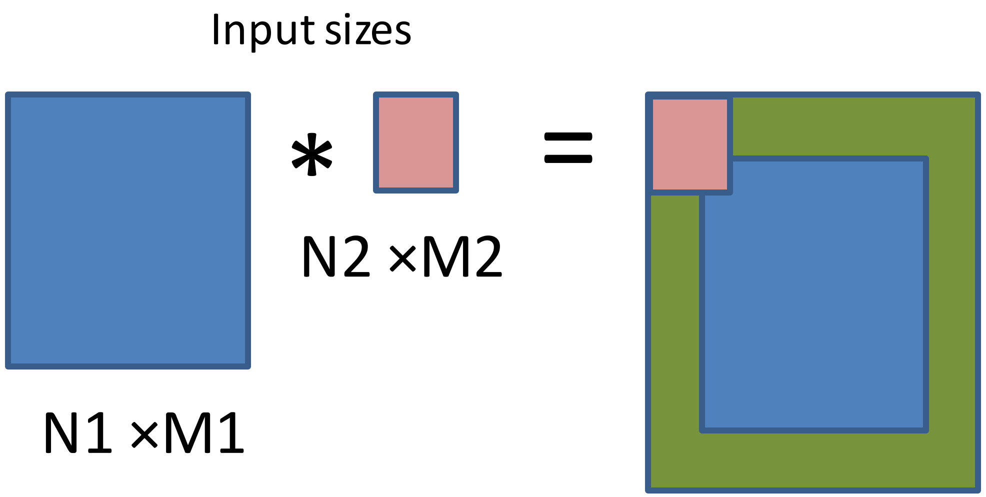
$$* \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) =$$



Sharpening filter: Accentuates differences with local average

Implementation detail: Image support and edge effect

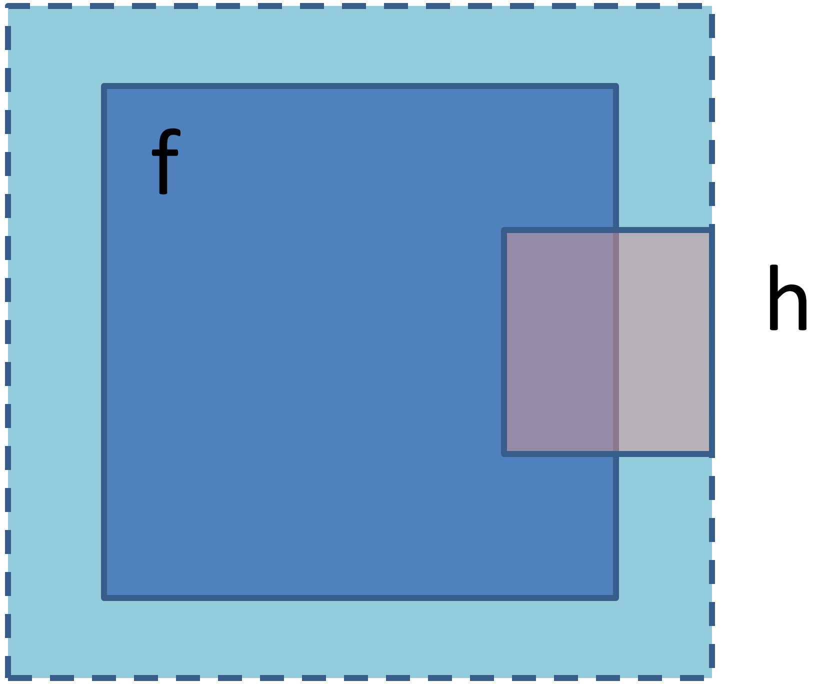
- A computer will only convolve **finite support signals.**
  - That is: images that are zero for n, m outside some rectangular region
- numpy's convolution performs 2D Discrete convolution of finite-support signals.



$(N1 + N2 - 1) \times (M1 + M2 - 1)$

Implementation detail: Image support and edge effect

- A computer will only convolve **finite support signals.**
- What happens at the edge?



- zero “padding”
- edge value replication
- mirror extension
- more (beyond the scope of this class)

What we will learn today?

- Convolution
- Correlation

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

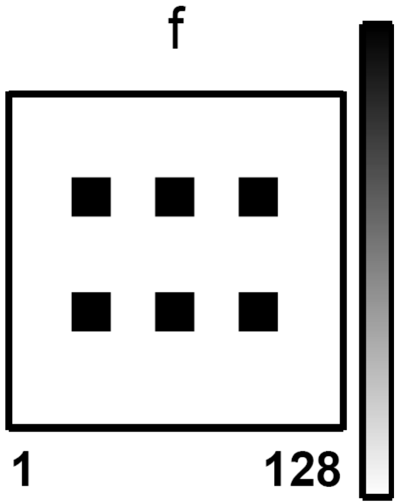
(Cross) correlation – symbol: $**$

Cross correlation of two 2D signals *f[n,m]* and *h[n,m]*

$$f[n,m] ** h[n,m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k,l] \cdot h[n+k, m+l]$$

- Equivalent to a convolution without the flip
- Use it to measure 'similarity' between f and h.

(Cross) correlation – example

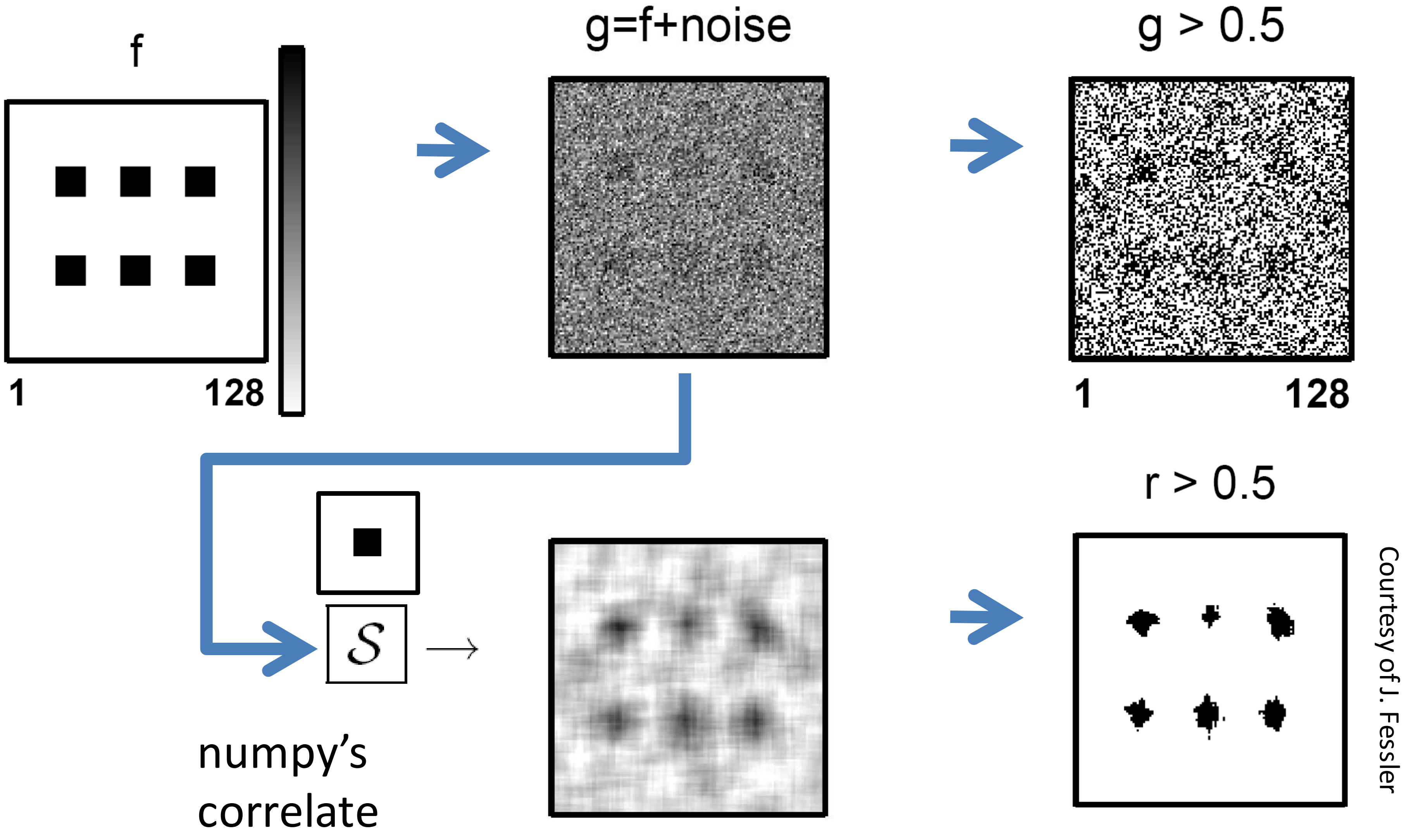


Courtesy of J. Fessler

(Cross) correlation – example

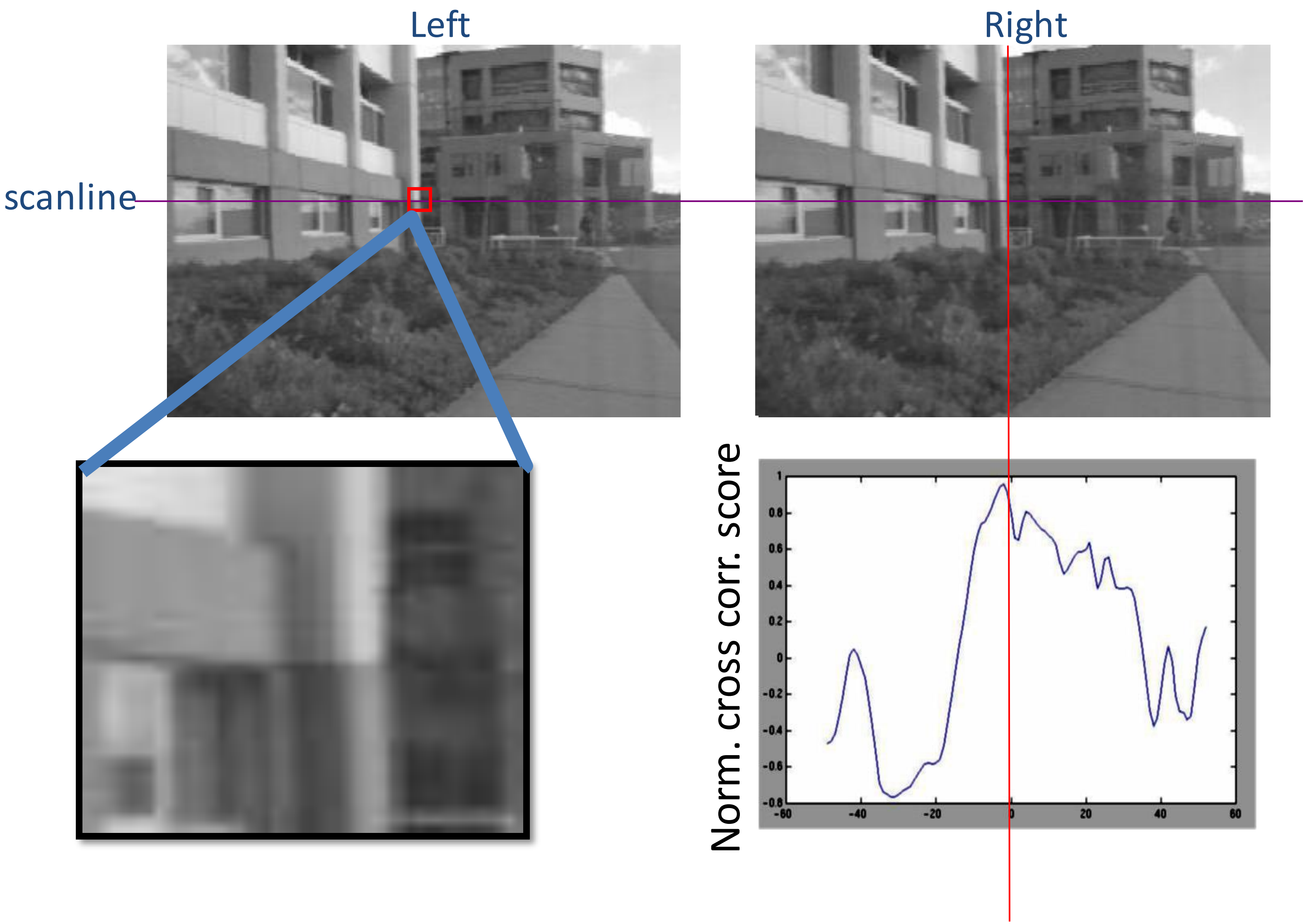
Courtesy of J. Fessler

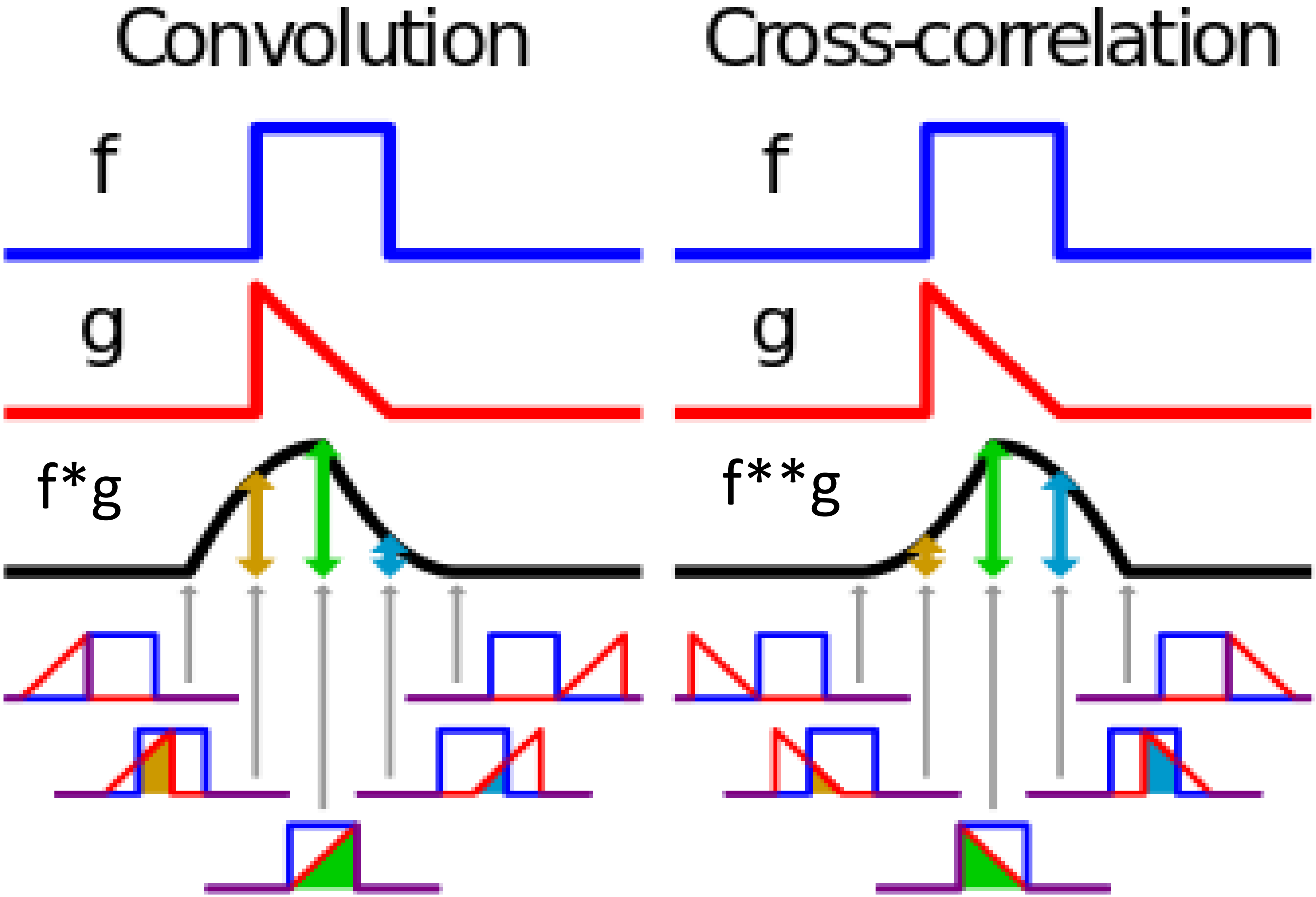
(Cross) correlation – example



Courtesy of J. Fessler

(Cross) correlation – example





Convolution
Cross-correlation
f
g
f*g
f
g
f**g

Cross Correlation Application: Vision system for TV remote control

- uses template matching

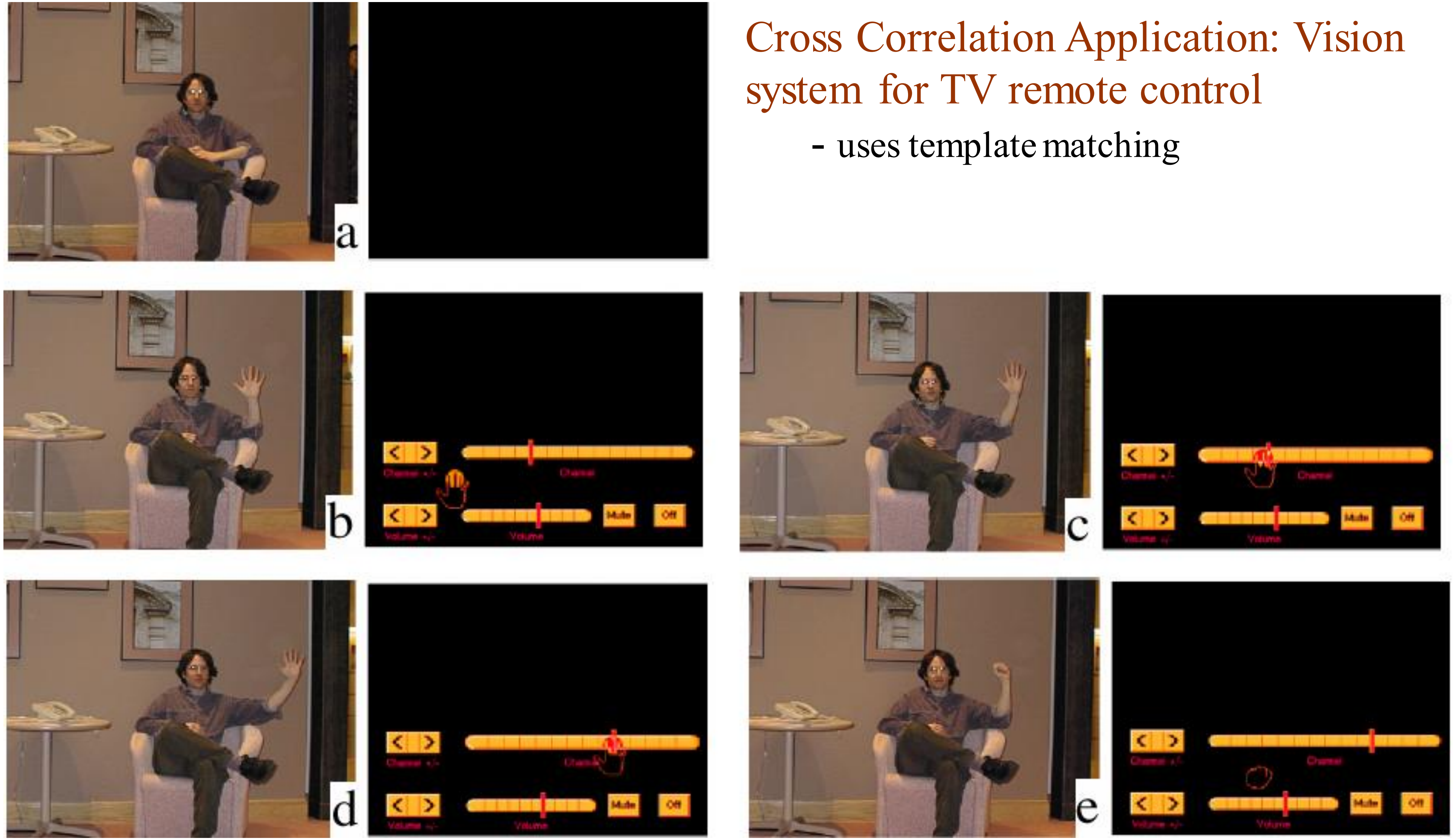


Figure from "Computer Vision for Interactive Computer Graphics," W.Freeman et al, IEEE Computer Graphics and Applications, 1998 copyright 1998, IEEE

Convolution vs. (Cross) Correlation

- When is correlation equivalent to convolution?
- In other words, when is f**g = f*g?

Convolution vs. (Cross) Correlation

- A **<u>convolution</u>** is an integral that expresses the amount of overlap of one function as it is shifted over another function.
  - convolution is a filtering operation

- **<u>Correlation</u>** compares the ***similarity*** *of* ***two*** *sets of* ***data.*** Correlation computes a measure of similarity of two input signals as they are shifted by one another. The correlation result reaches a maximum at the time when the two signals match best.
  - correlation is a measure of relatedness of two signals

Summary

- Convolution
- Correlation