Correlation Filters

Instructor - Simon Lucey

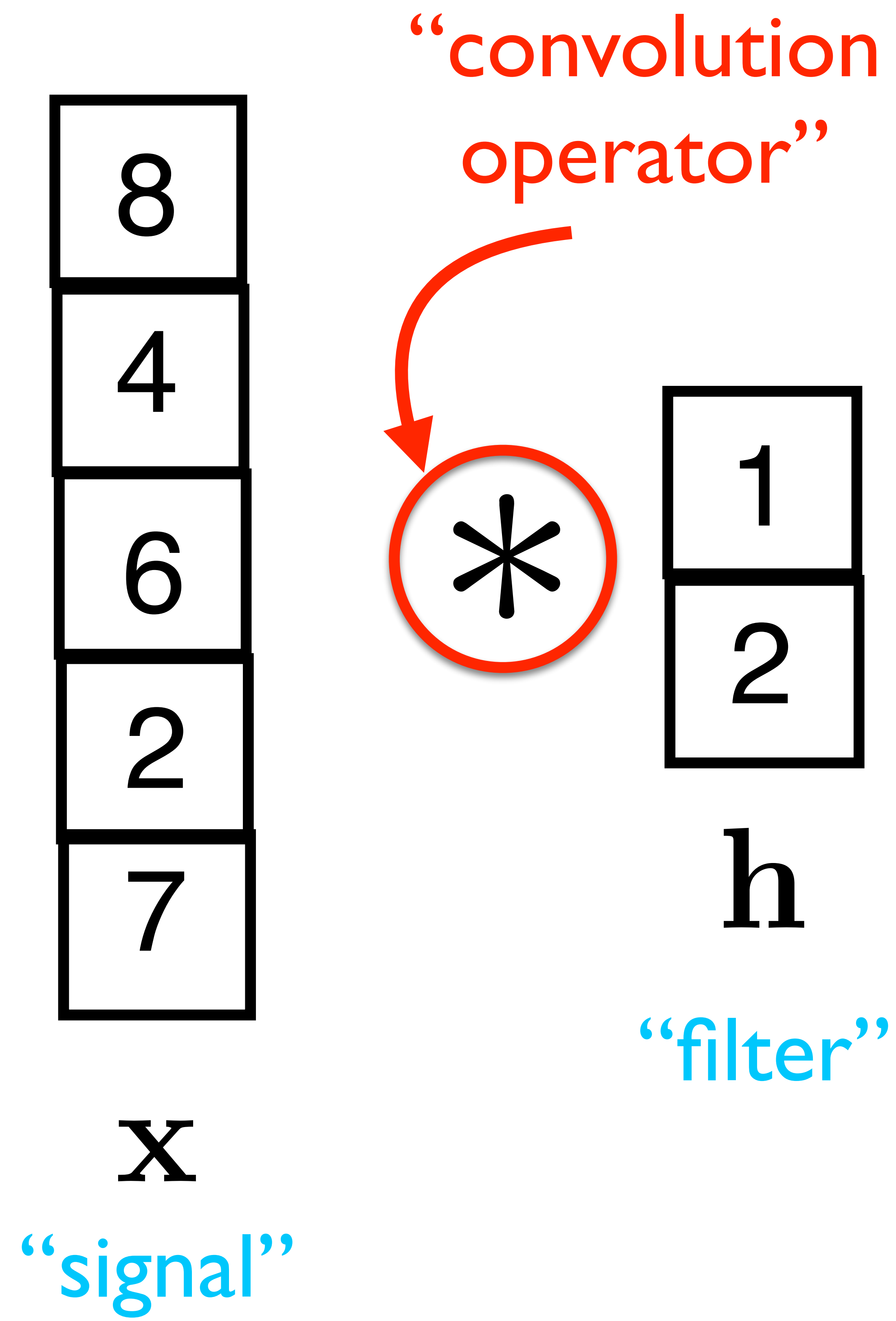
16-423 - Designing Computer Vision Apps

Carnegie Mellon
THE ROBOTICS INSTITUTE

Today

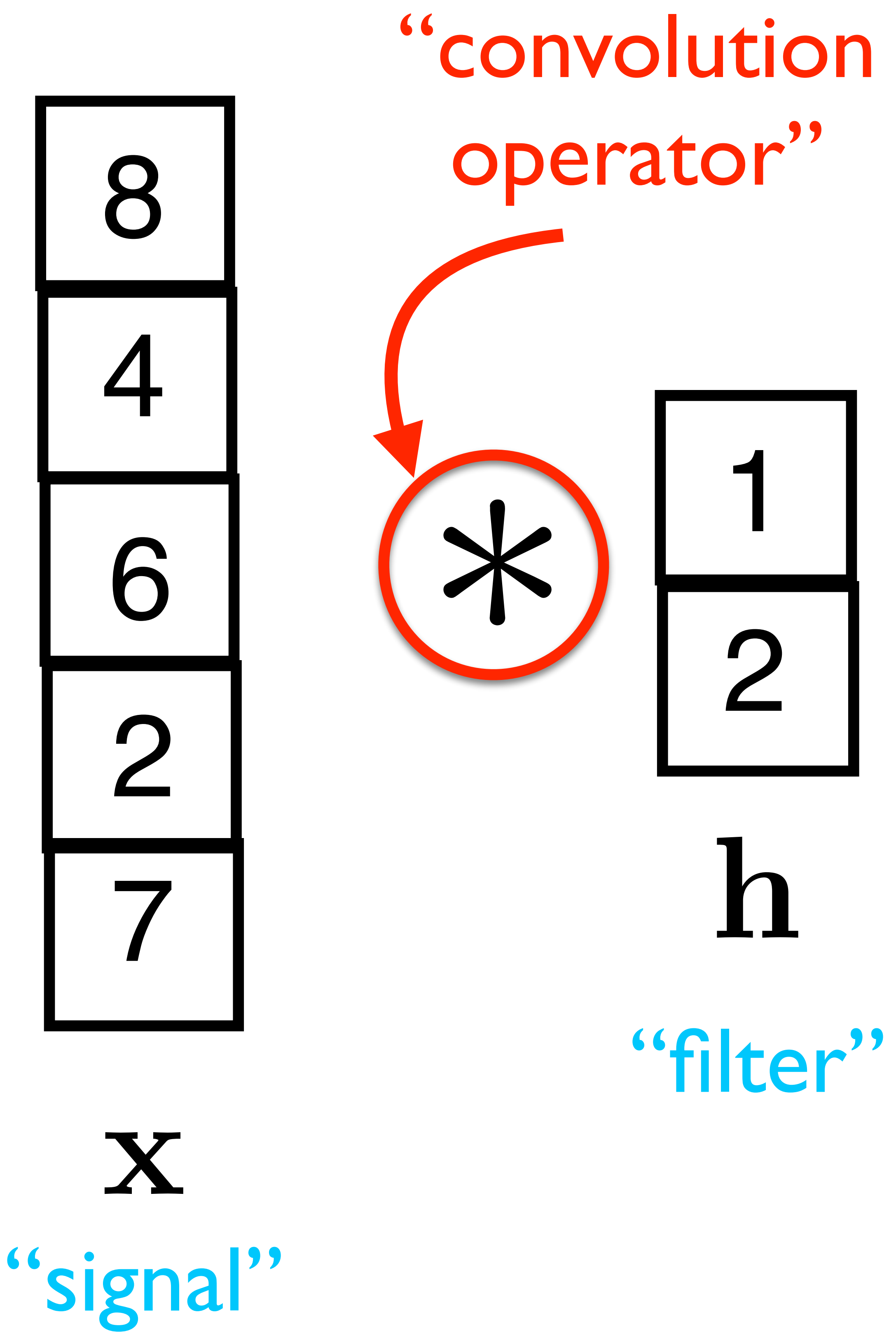
- Types of Convolution
- Fast Fourier Transform (FFT)
- The Correlation Filter

Convolution

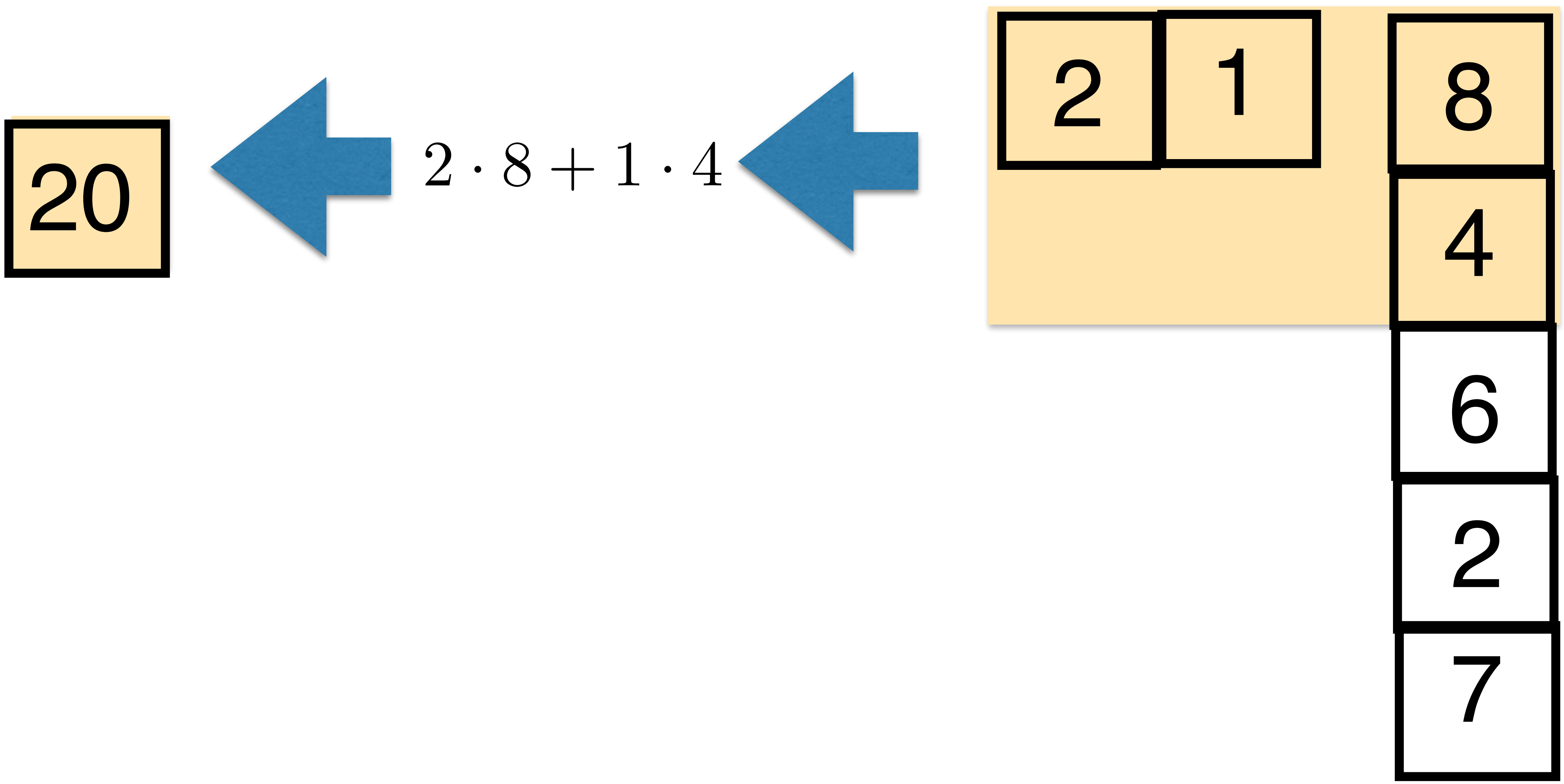


Convolution

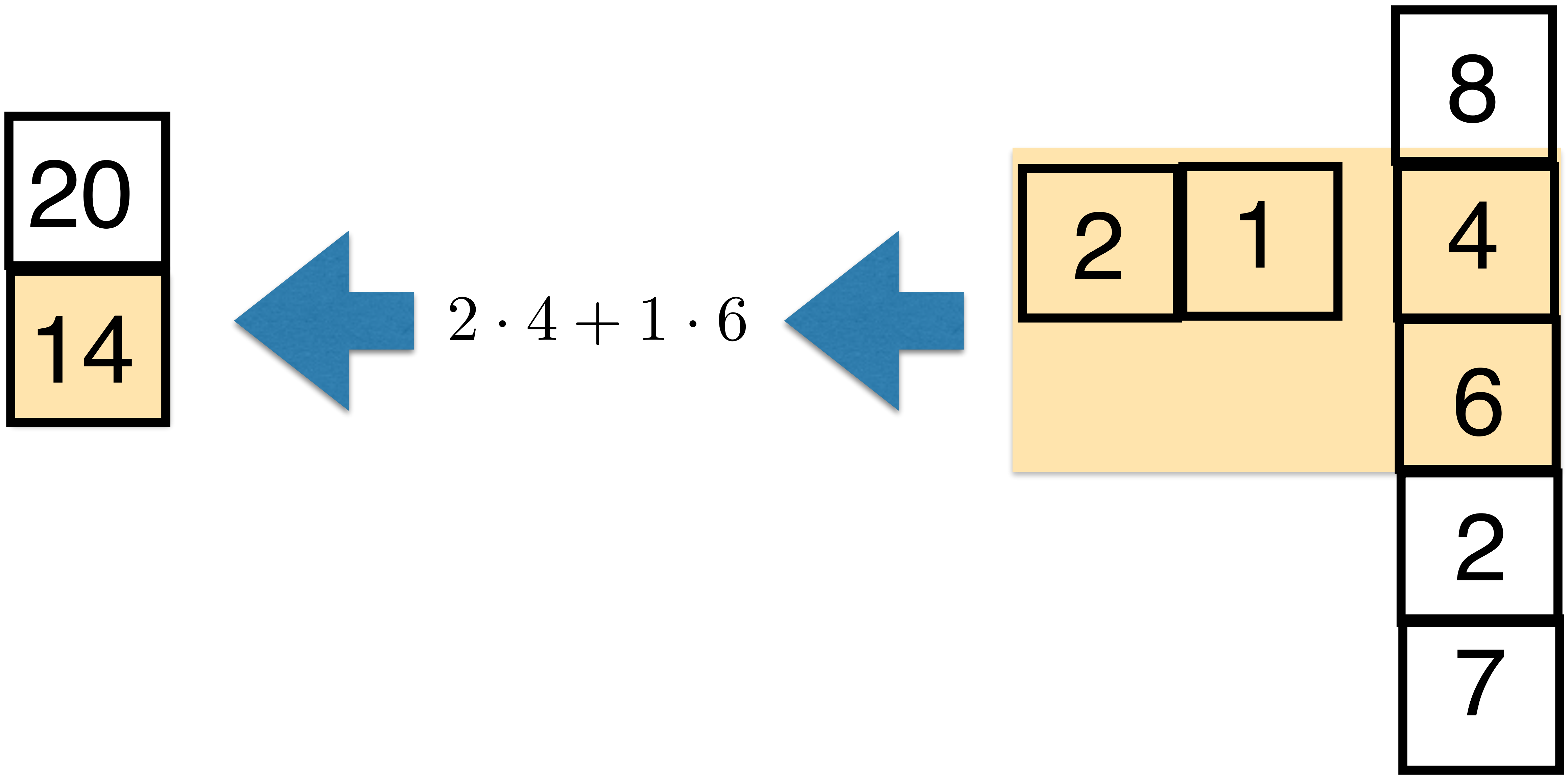
```
>> conv(x,h,'valid')
ans =

    20
    14
    14
    11
```



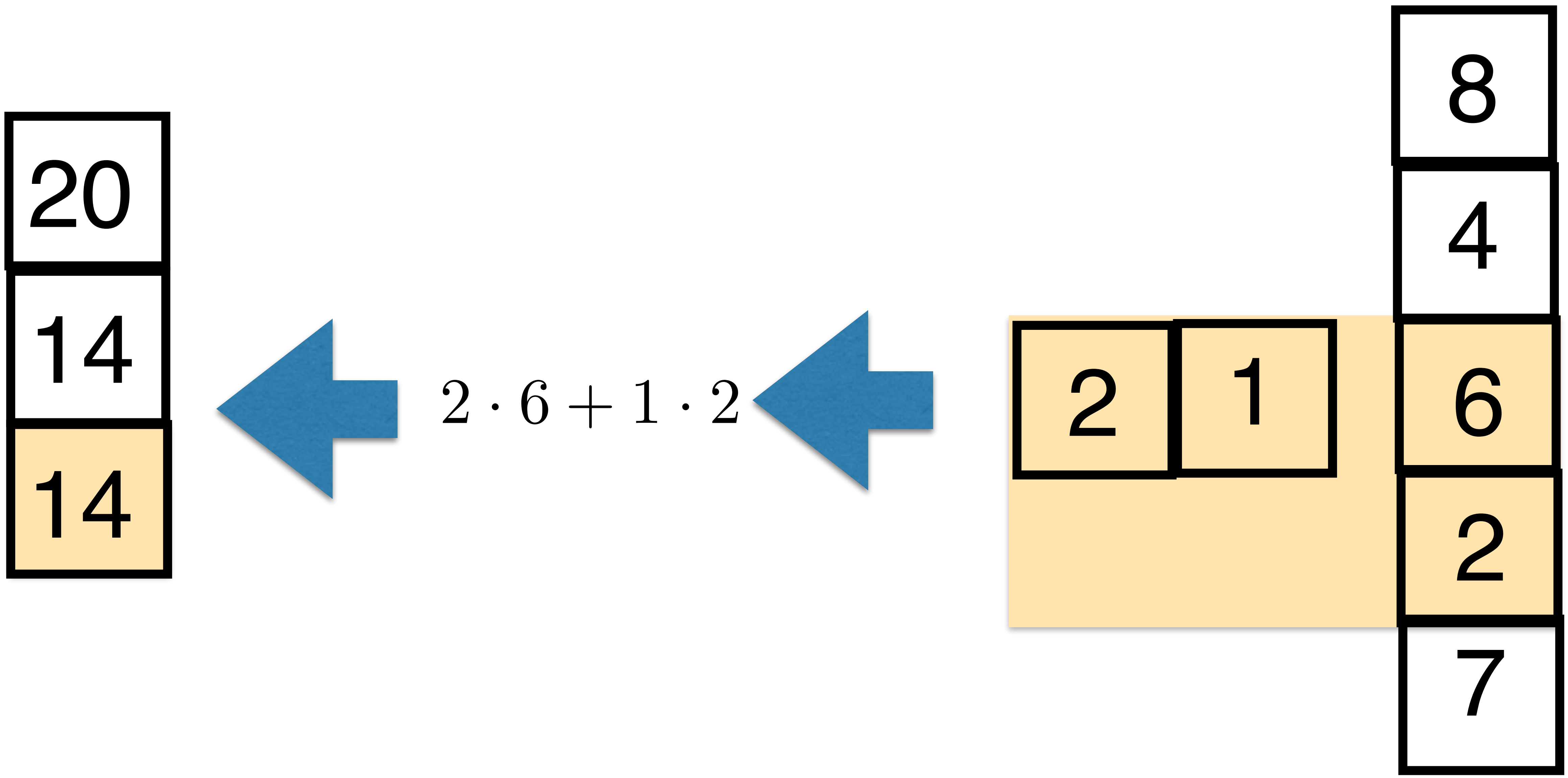
Convolution



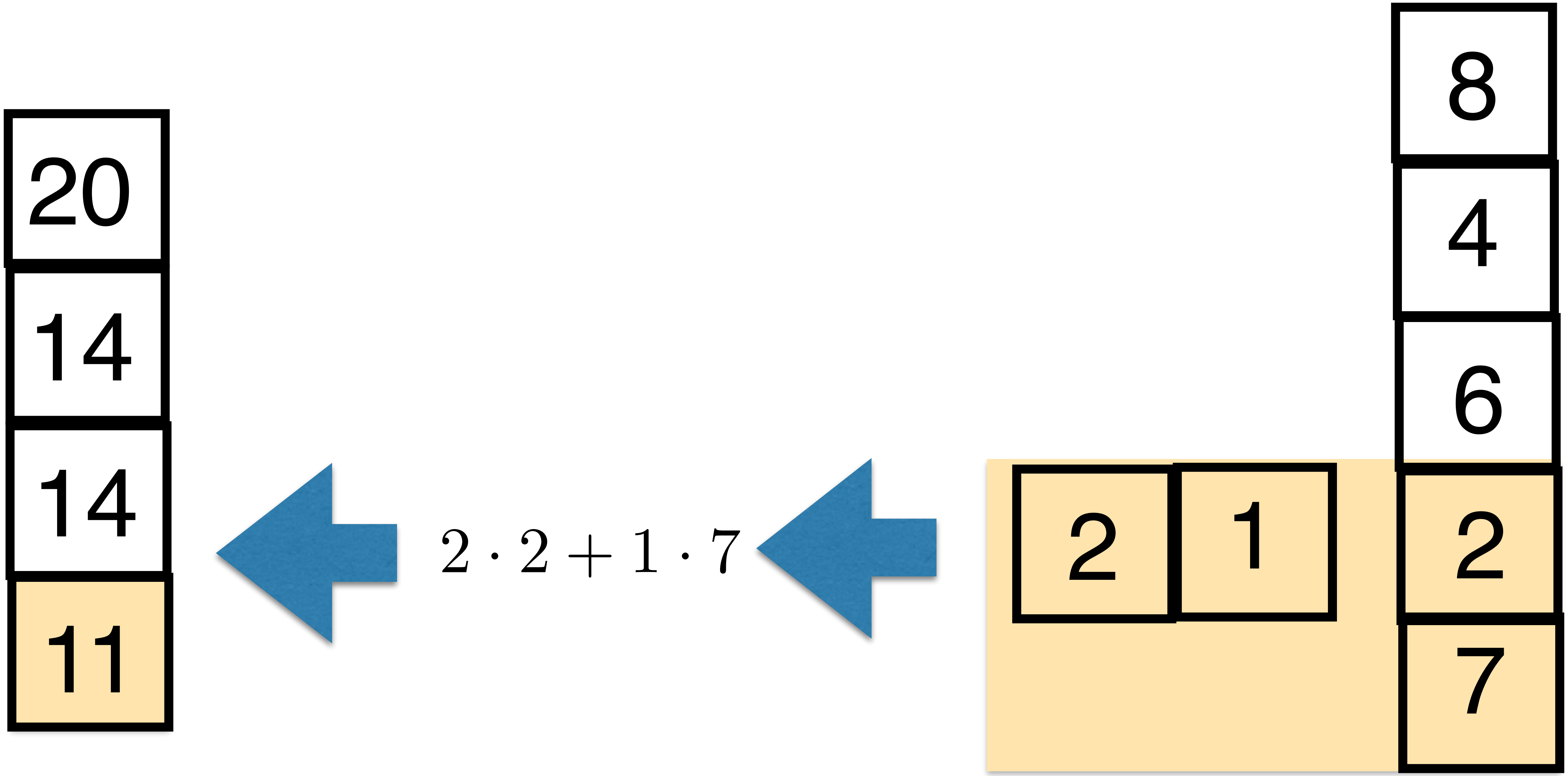
Convolution



Convolution



Convolution



Convolution

$$
\underbrace{\begin{bmatrix} 20 \\ 14 \\ 14 \\ 11 \end{bmatrix}}_{\mathbf{Hx}}
\leftarrow
\underbrace{\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 \end{bmatrix}}_{\mathbf{H}}
\underbrace{\begin{bmatrix} 8 \\ 4 \\ 6 \\ 2 \\ 7 \end{bmatrix}}_{\mathbf{x}}
$$

$\mathbf{H}$ "convolutional matrix"

$\mathbf{x}$ "signal"

Types of Convolution

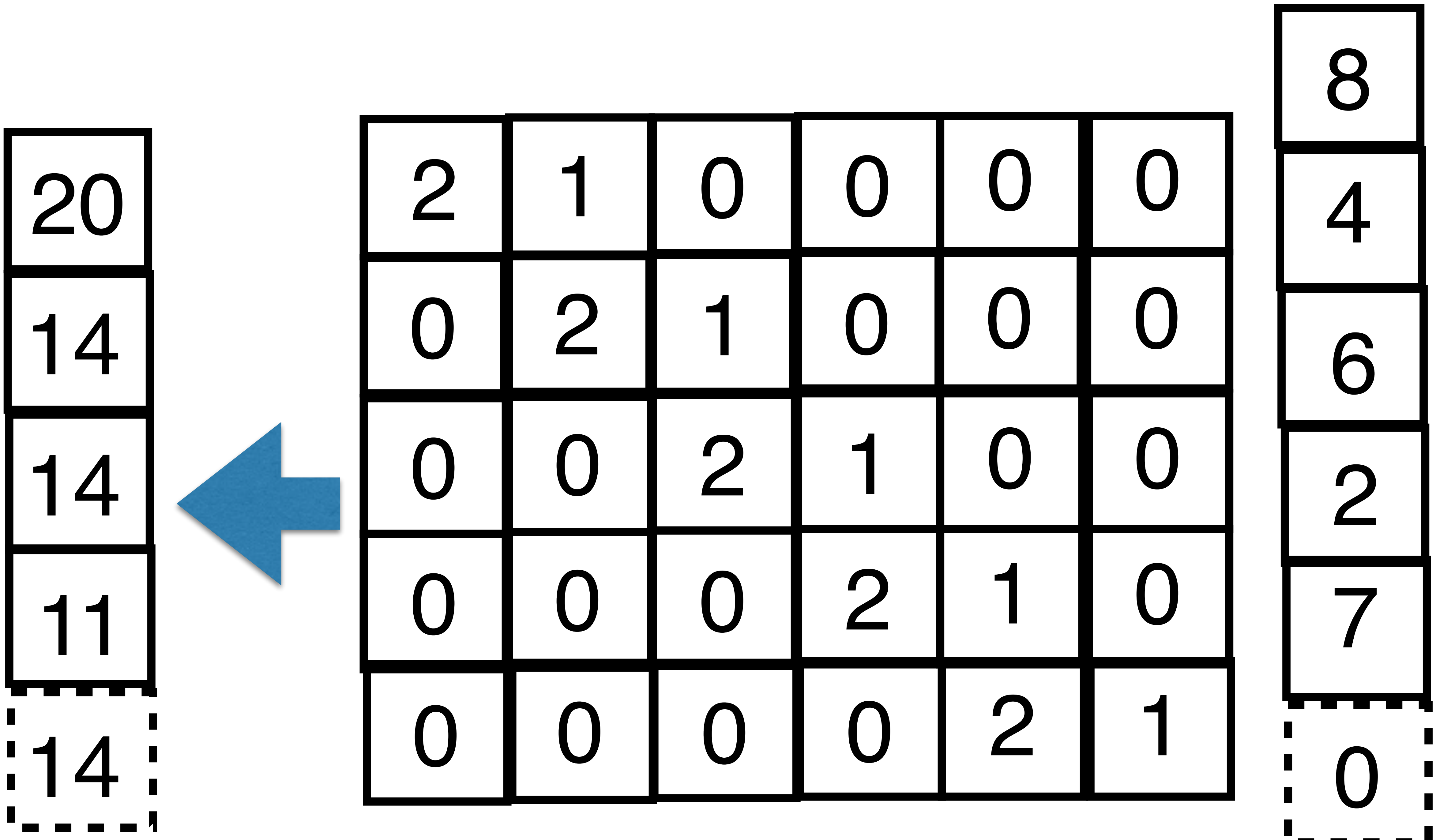
- More than just one type of convolutional operator:-
  - “Valid” convolution

```
>> conv(x,h,'valid')
```

  - “Same” convolution

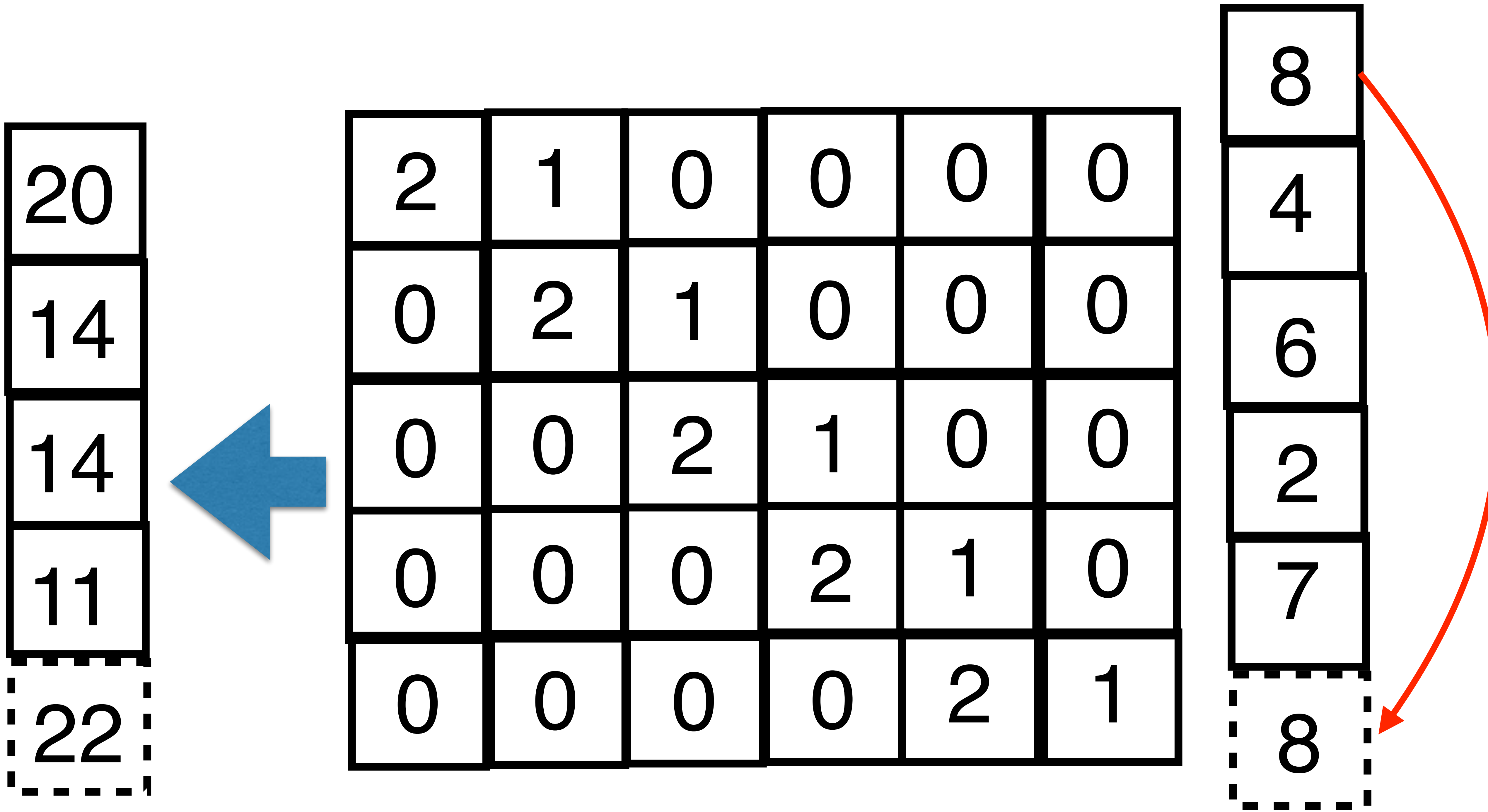
```
>> conv(x,h,'same')
```

Zero-Padded Convolution

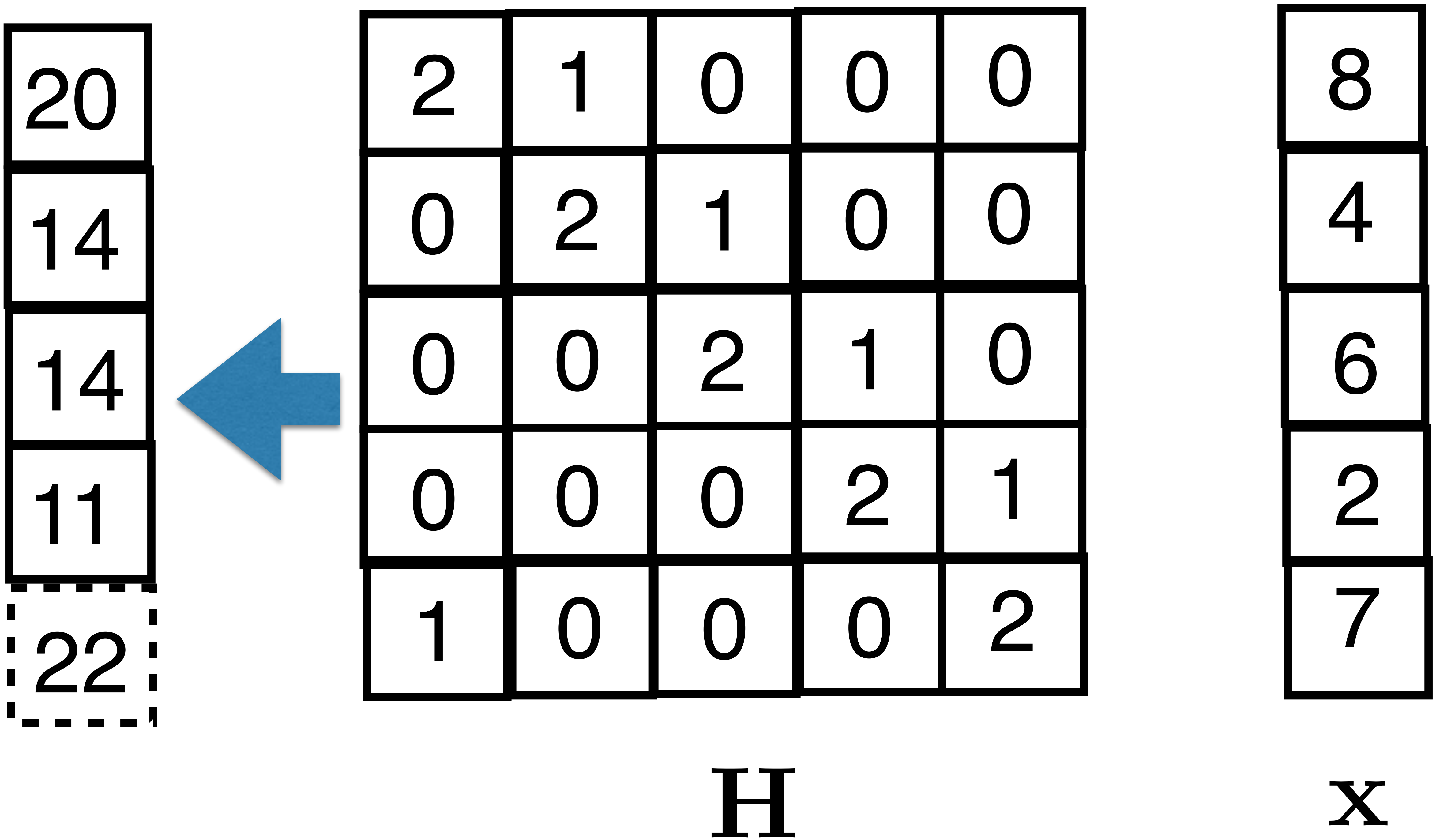



```
>> conv(x,h,'same')
```

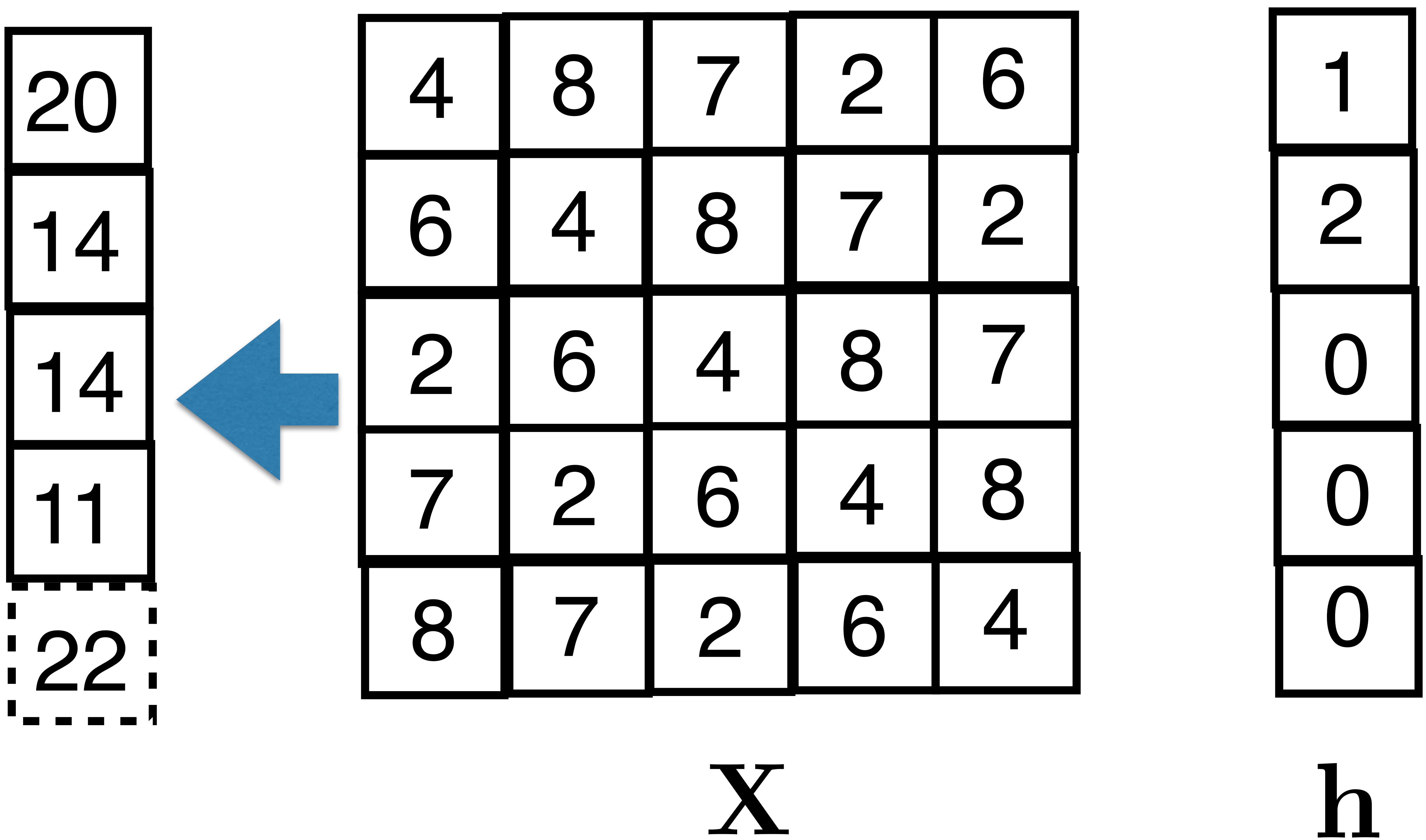
Circular Convolution



Circular Convolution

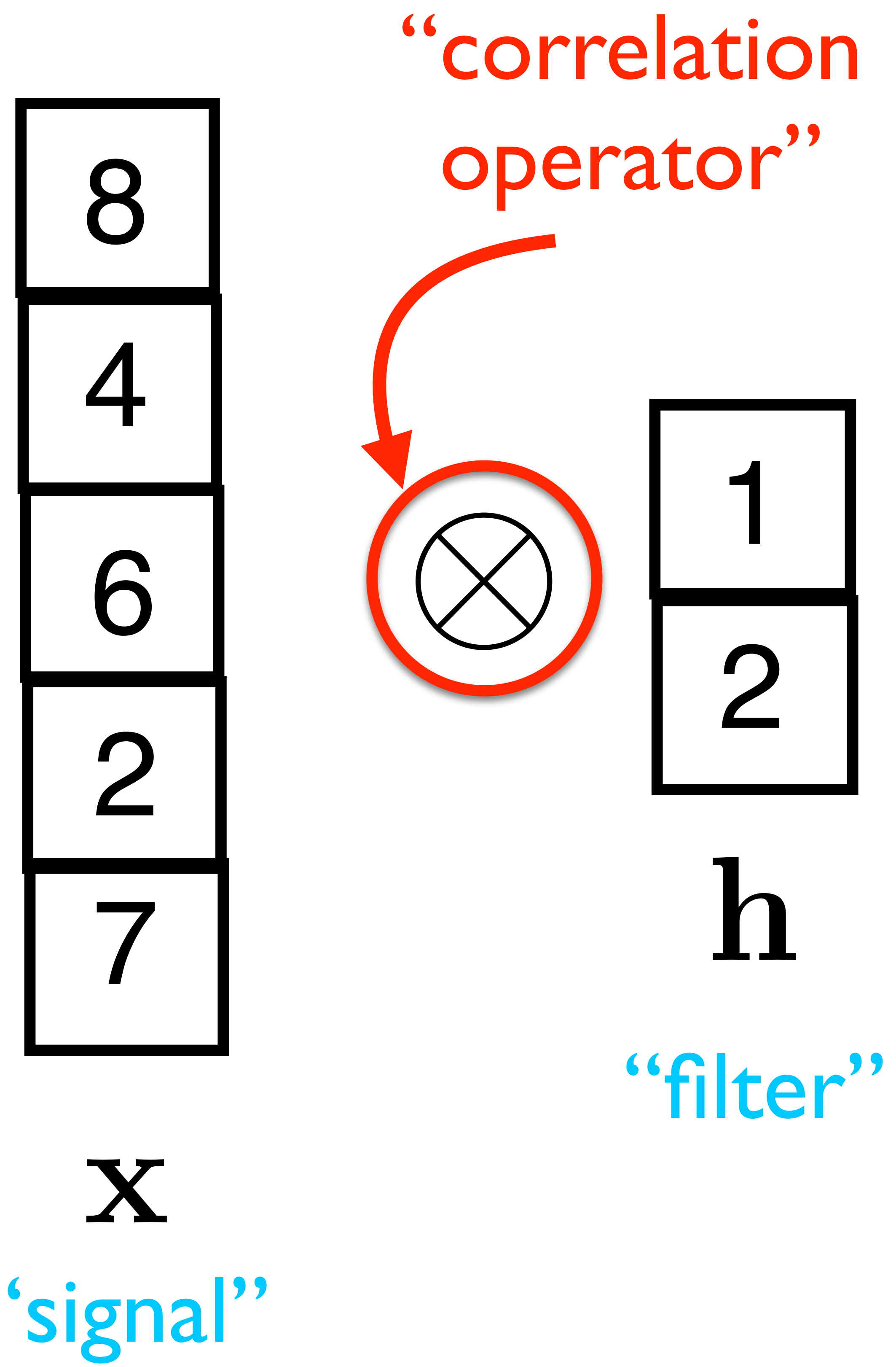


Circular Convolution



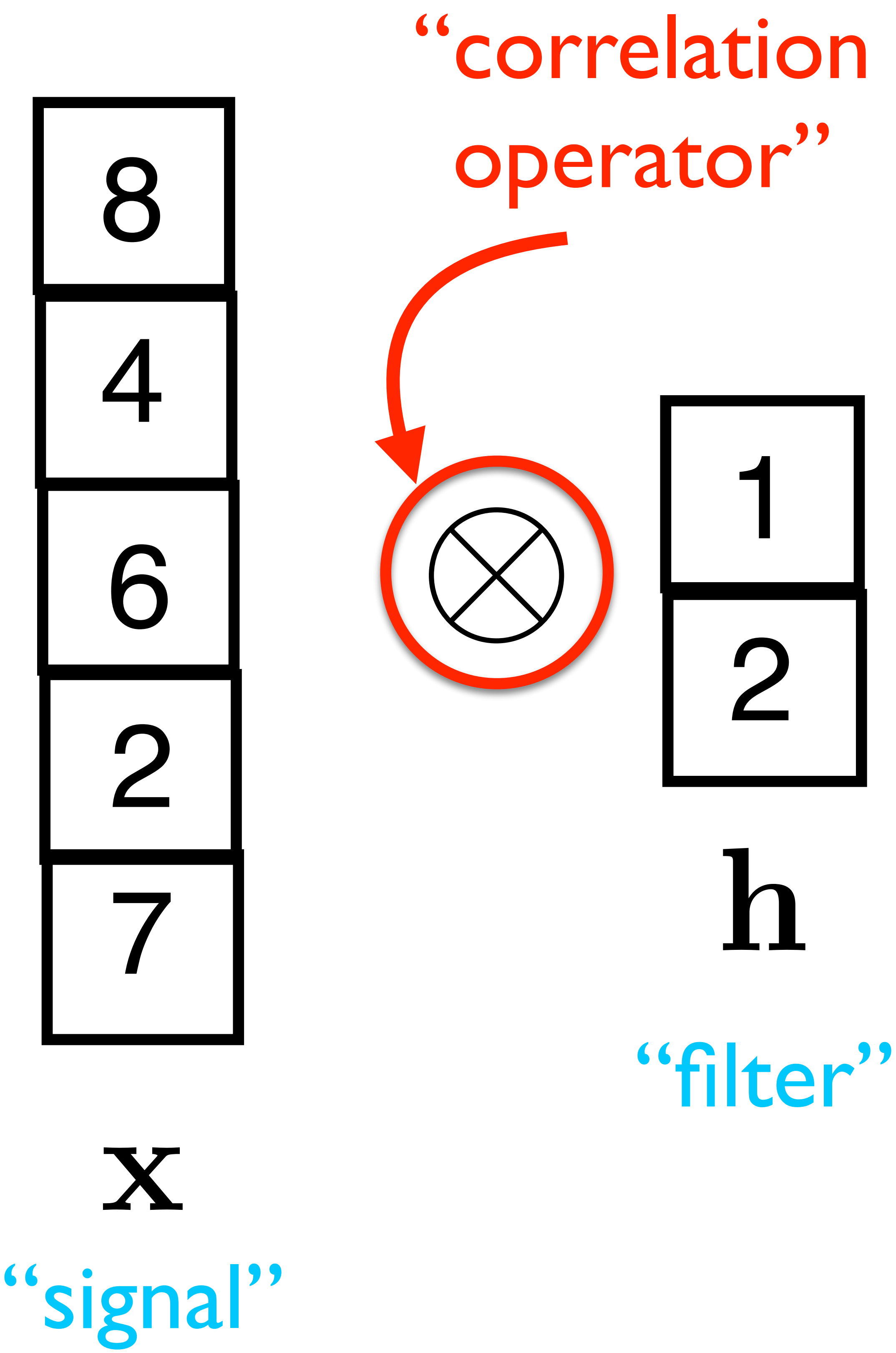
Correlation

```
>> conv(x,flipud(h),
...'same')
ans =

    16
    16
    10
    16
     7
```

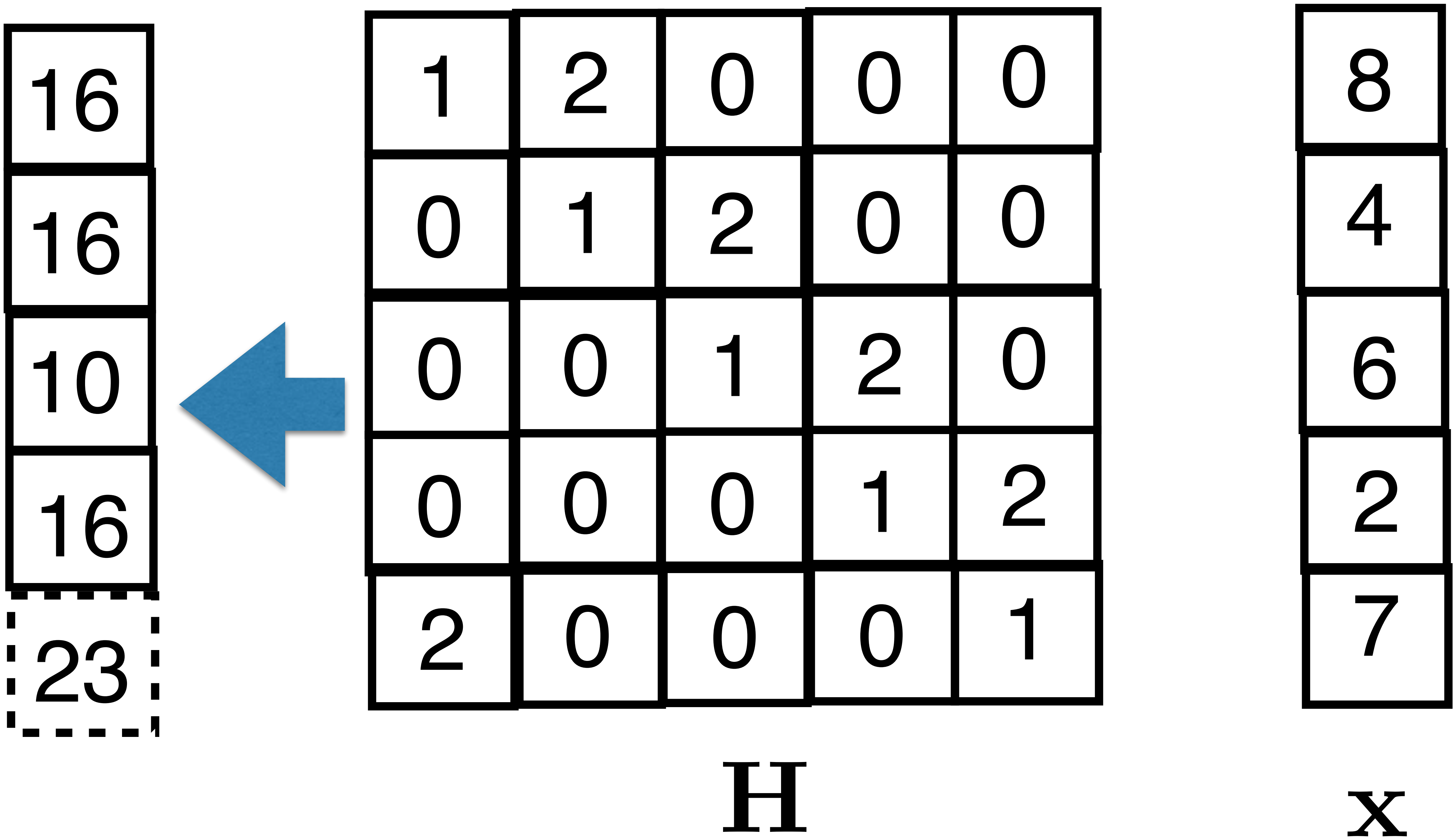


Correlation

```
>> imfilter(x,h)
ans =

    16
    16
    10
    16
     7
```



Circular Correlation




```
>> ifft(fft(x).*conj(fft(h)))
```

Correlation vs. Convolution

- Convolution is preferred mathematically as it is associative,

$$(\mathbf{x} * \mathbf{h}) * \mathbf{h} = \mathbf{x} * (\mathbf{h} * \mathbf{h})$$

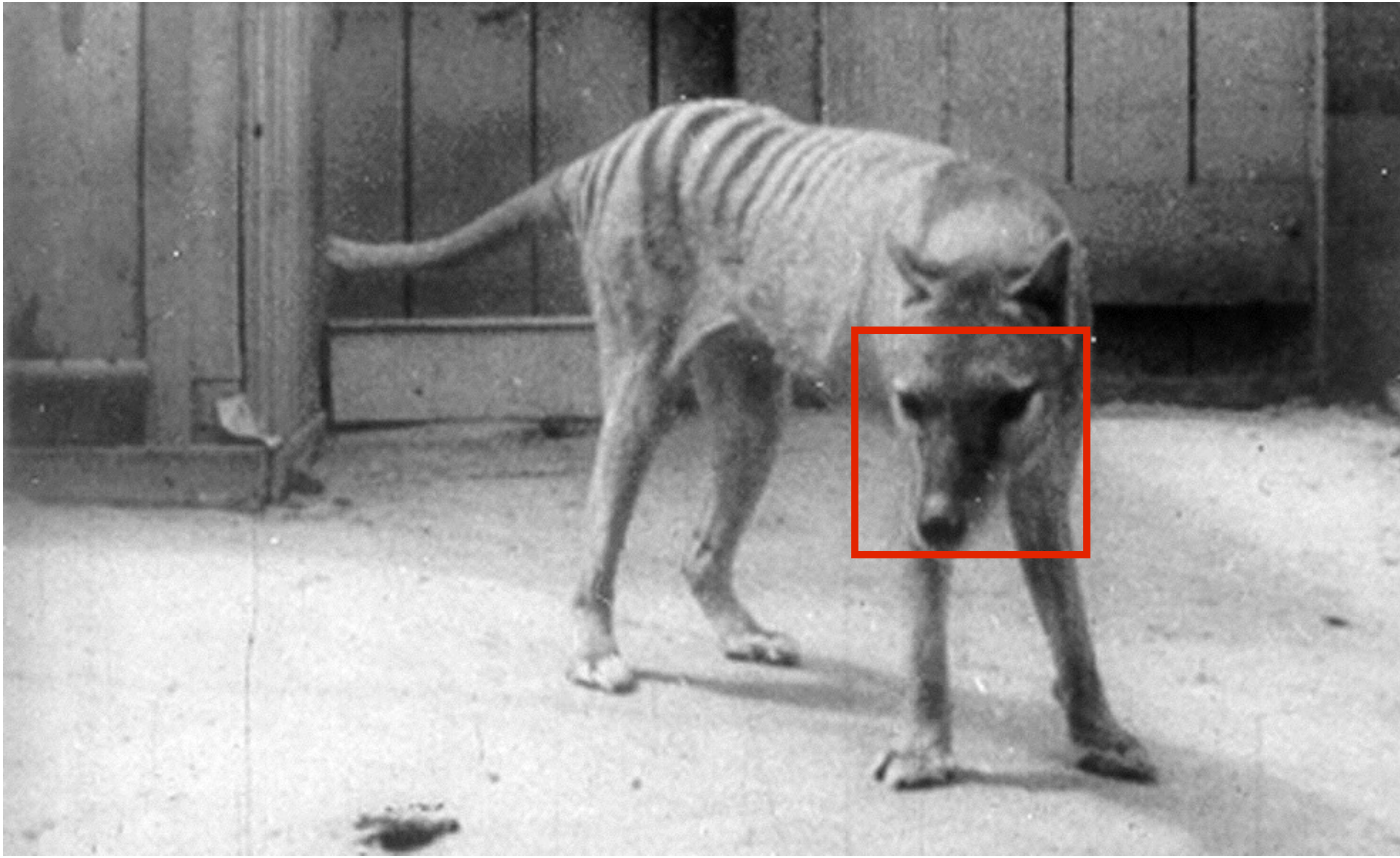
- Correlation is not associative,

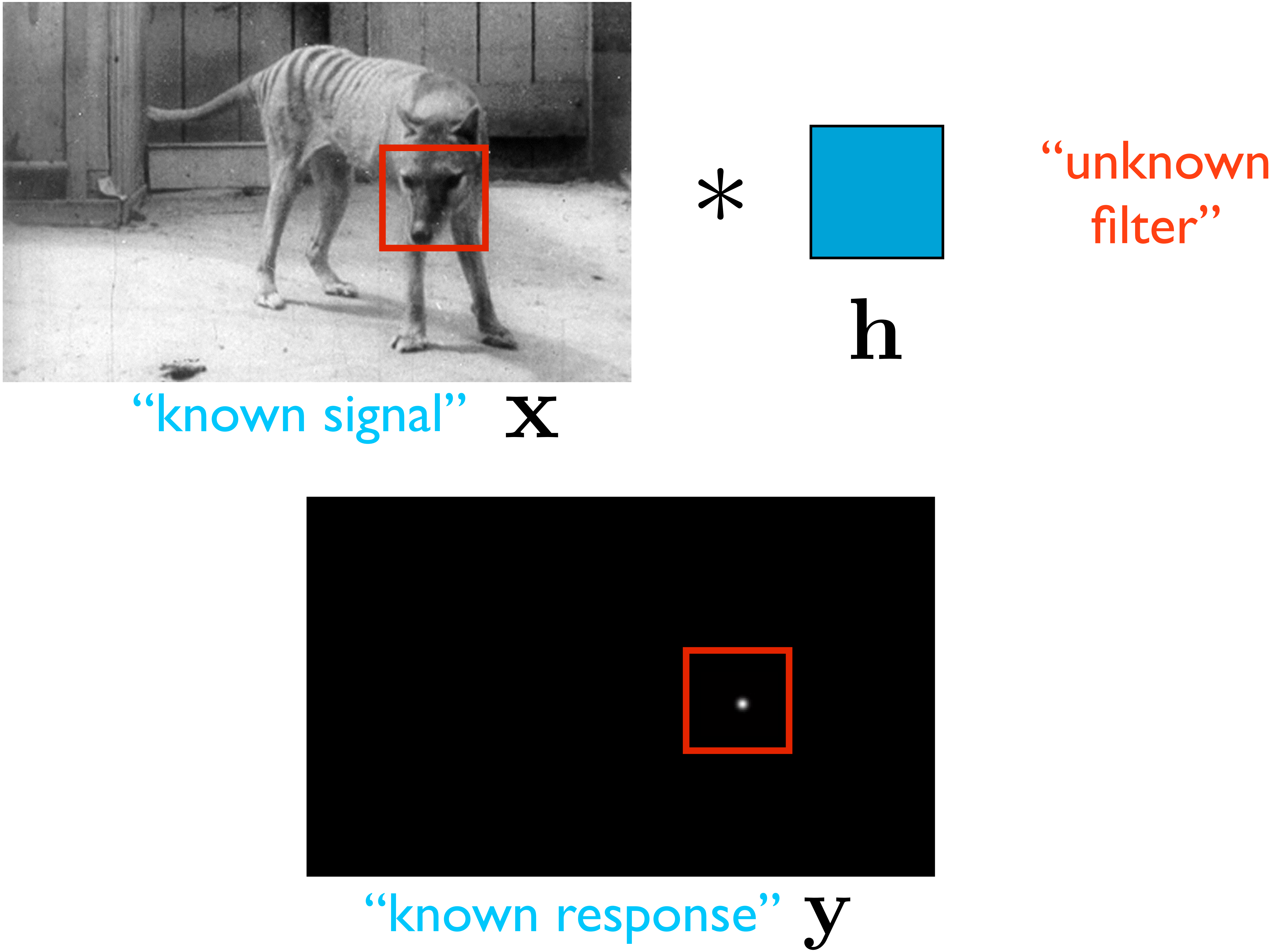
$$(\mathbf{x} \otimes \mathbf{h}) \otimes \mathbf{h} \neq \mathbf{x} \otimes (\mathbf{h} \otimes \mathbf{h})$$

- Correlation preferred, however, for signal matching/ detection.

Today

- Types of Convolution
- Fast Fourier Transform (FFT)
- The Correlation Filter



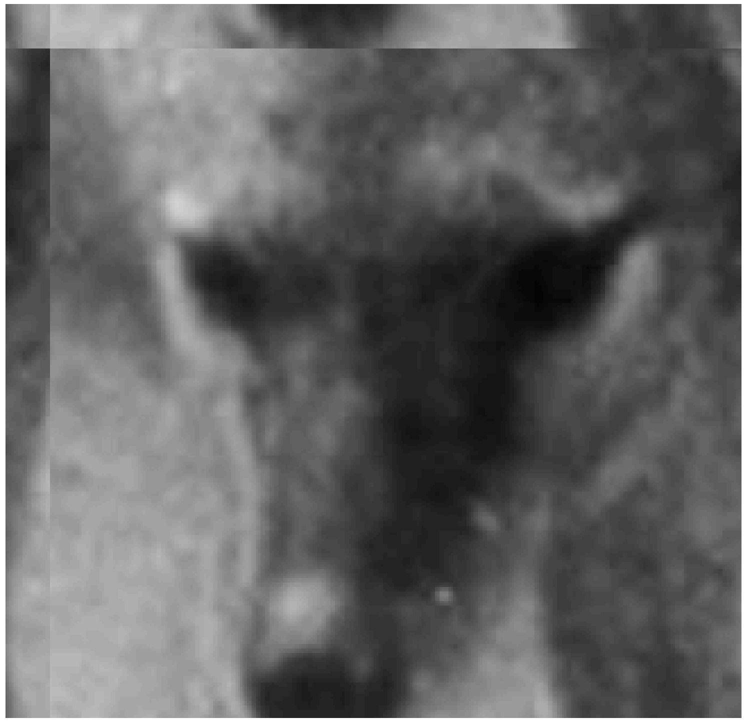


*
"unknown filter"
$\mathbf{h}$
"known signal" $\mathbf{x}$
"known response" $\mathbf{y}$

$\mathbf{x} \in \mathcal{R}^D$

$\mathbf{x}[0,0]$

$\mathbf{x}[0,0]$



$\mathbf{x}[20,20]$

```
>> xshift = circshift(x,[20,20]);
```

$\mathbf{x}[0,0]$

$\mathbf{x}[20,20]$

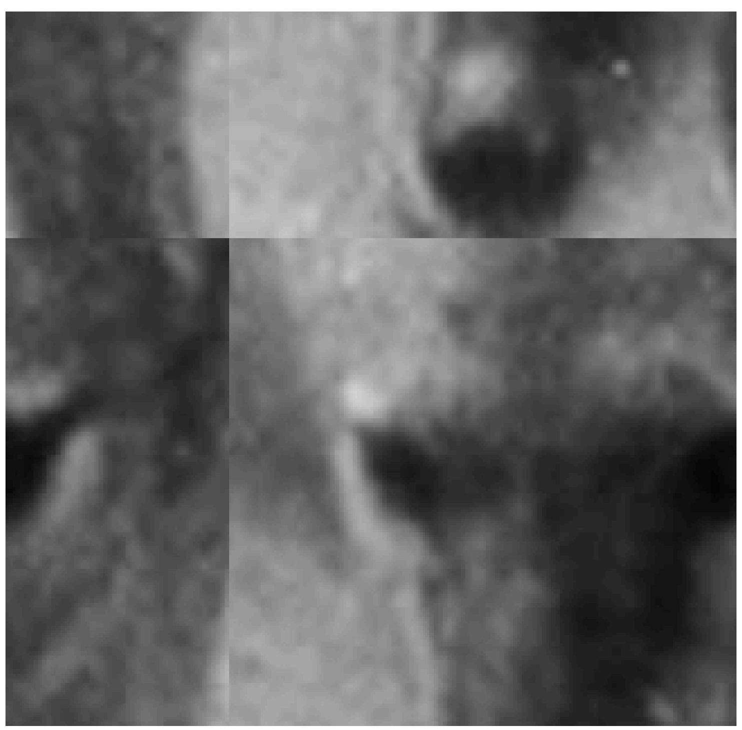
$\mathbf{x}[-20,-20]$

```
>> xshift = circshift(x,[-20,-20]);
```

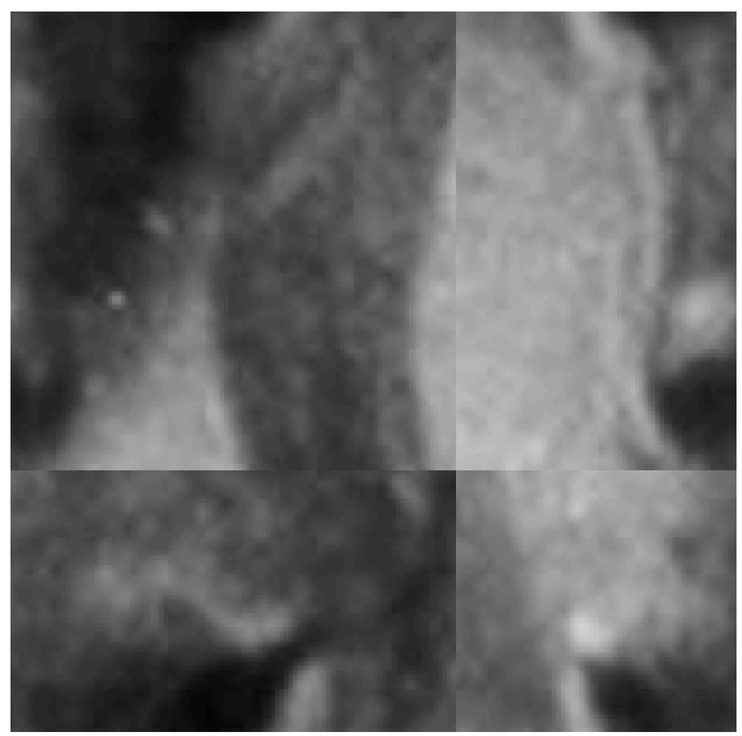
$\mathbf{x}[0,0]$

$\mathbf{x}[20,20]$

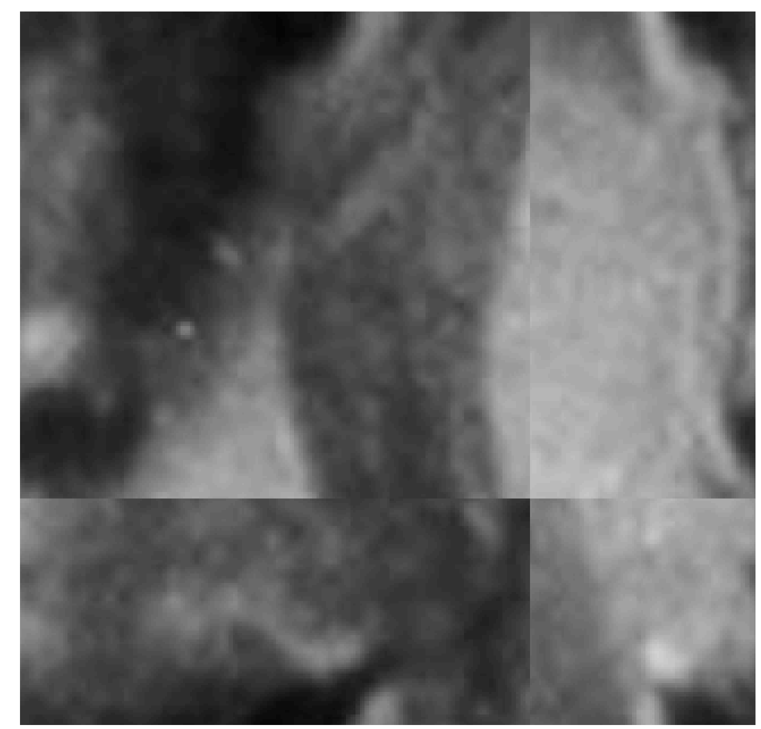
$\mathbf{x}[-20,-20]$



$\mathbf{x}[100,100]$

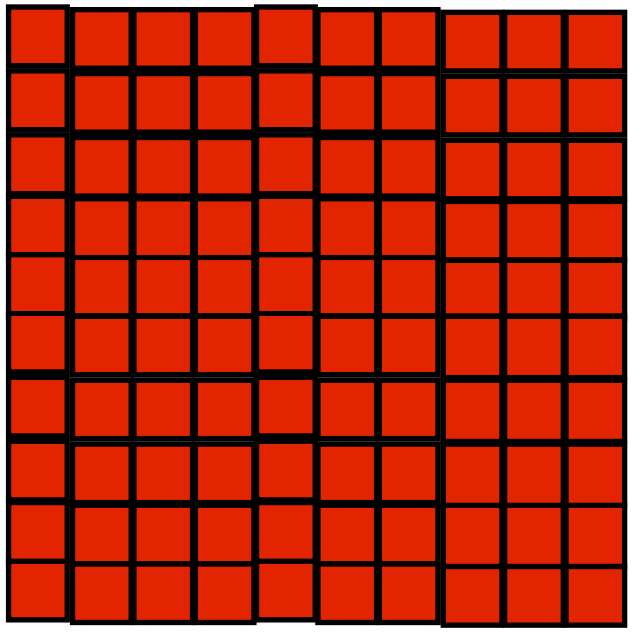


$\mathbf{x}[-100,-100]$



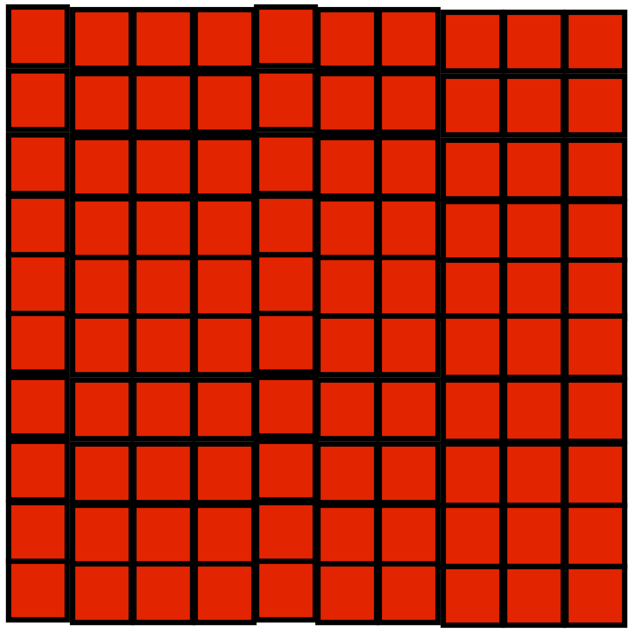
$\mathbf{x}[200,200]$

$$\mathbf{S} = \sum_{\boldsymbol{\tau} \in \mathcal{C}} \mathbf{x}[\boldsymbol{\tau}]\mathbf{x}[\boldsymbol{\tau}]^T =$$



$(D \times D)$

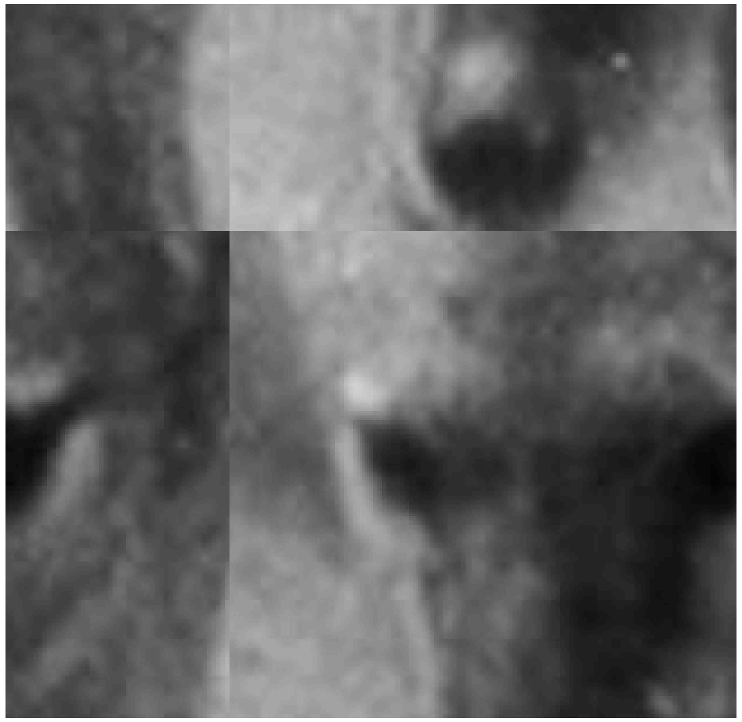
$$\mathbf{S} = \sum_{\boldsymbol{\tau} \in \mathcal{C}} \mathbf{x}[\boldsymbol{\tau}]\mathbf{x}[\boldsymbol{\tau}]^T =$$



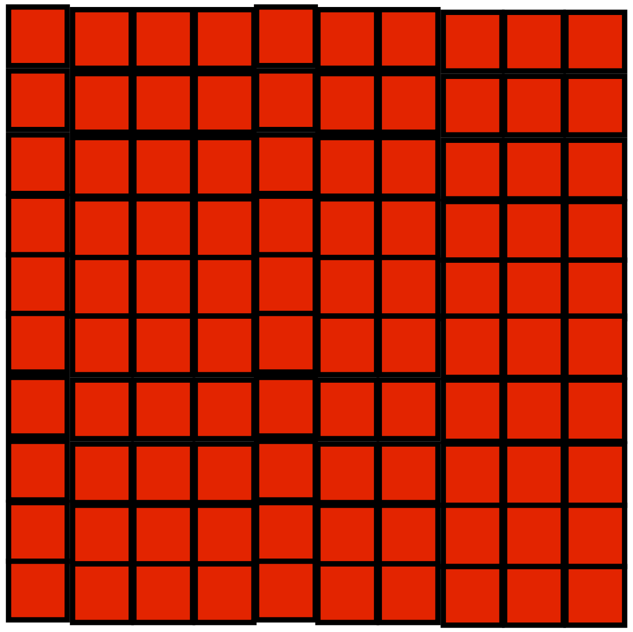
$(D \times D)$

"set of all circular shifts"

$\mathbf{x}[\boldsymbol{\tau}]$



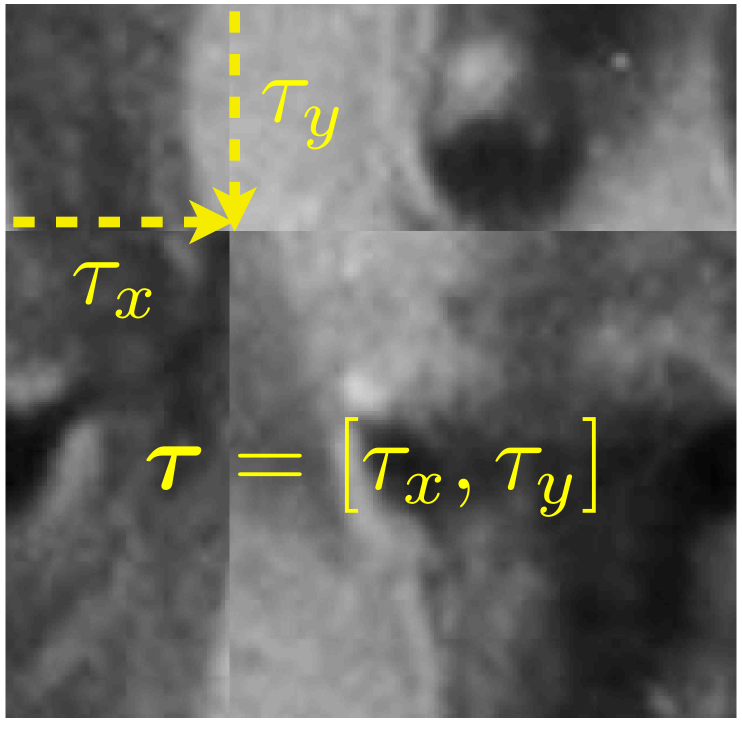
$$\mathbf{S} = \sum_{\boldsymbol{\tau} \in \mathcal{C}} \mathbf{x}[\boldsymbol{\tau}]\mathbf{x}[\boldsymbol{\tau}]^T =$$

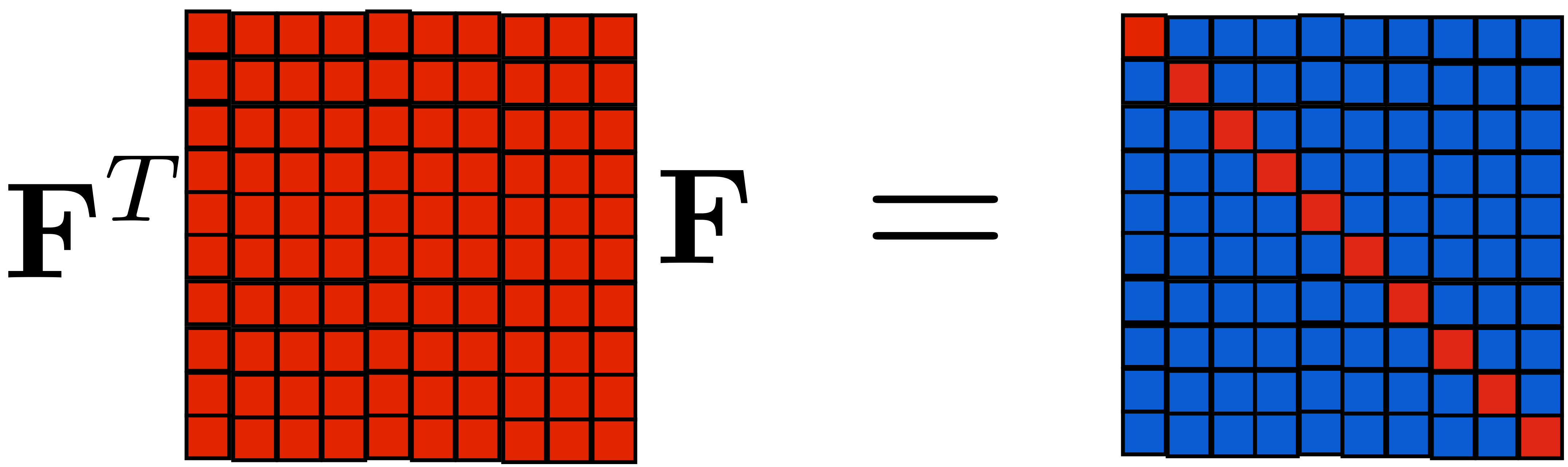


$(D \times D)$

"set of all circular shifts"

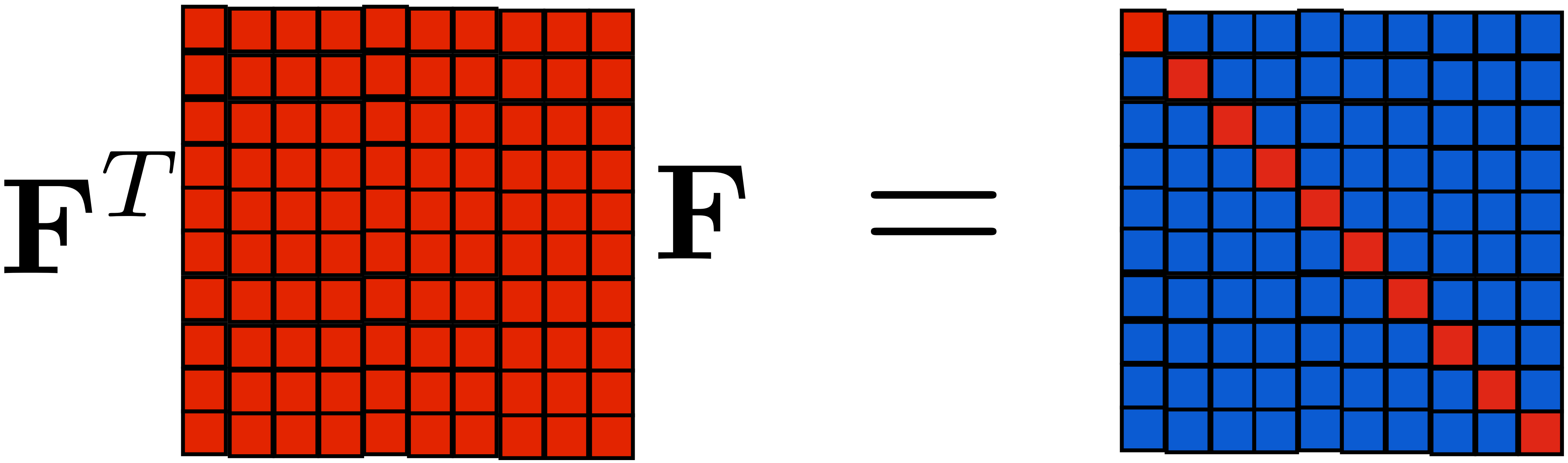
$\mathbf{x}[\boldsymbol{\tau}]$





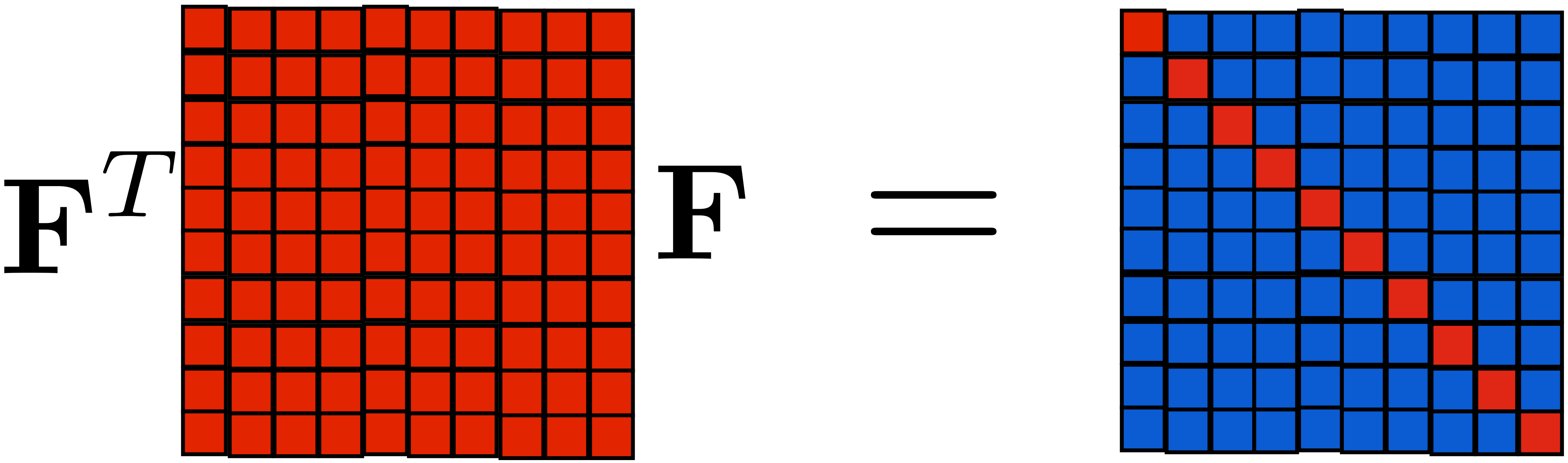
$$\mathbf{F} \leftarrow \text{eigenvectors of } \mathbf{S}$$

Not Always Zero   Always Zero



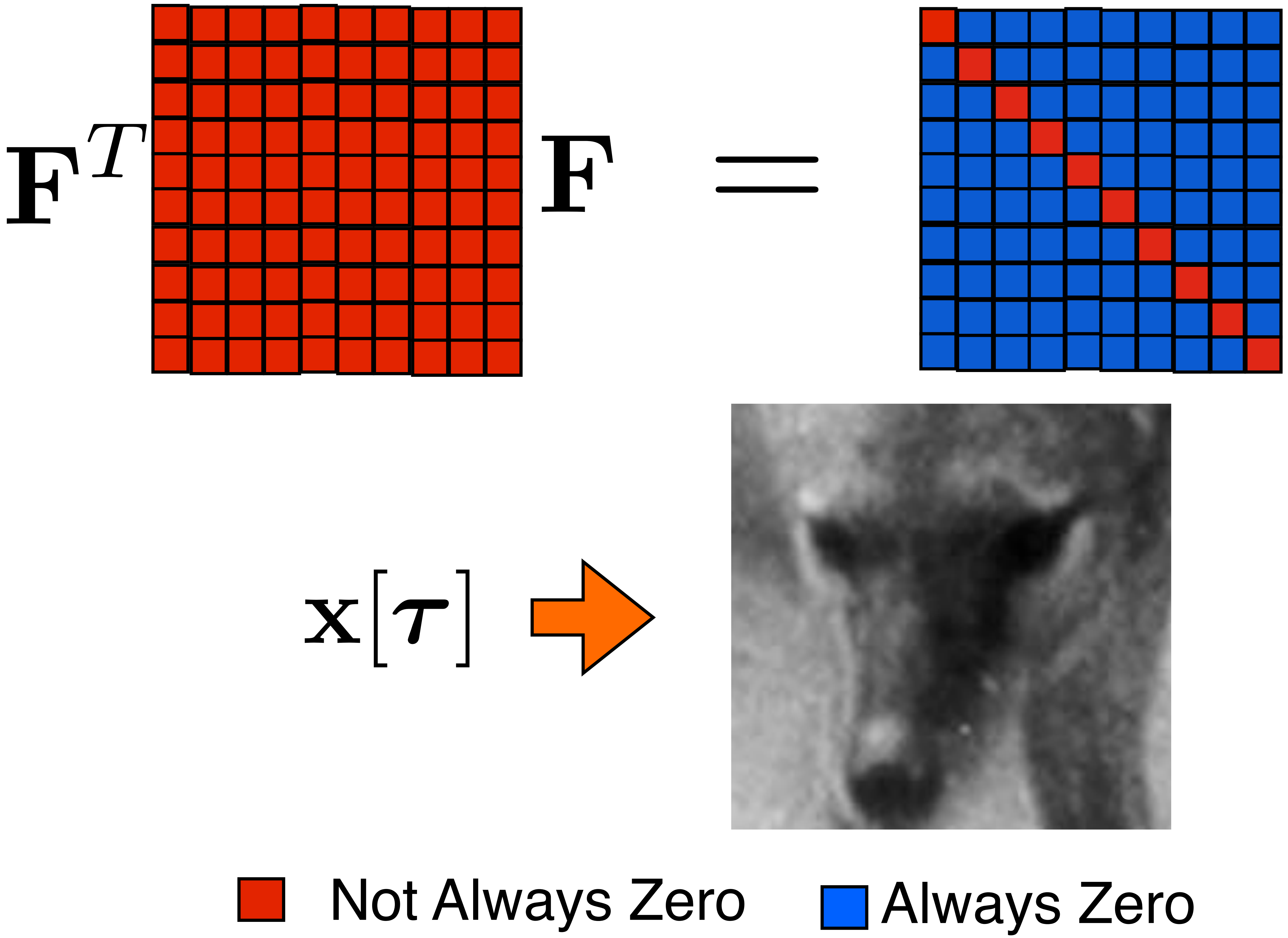
$$\mathbf{F} \leftarrow \text{Fourier Transform}$$

Not Always Zero   Always Zero

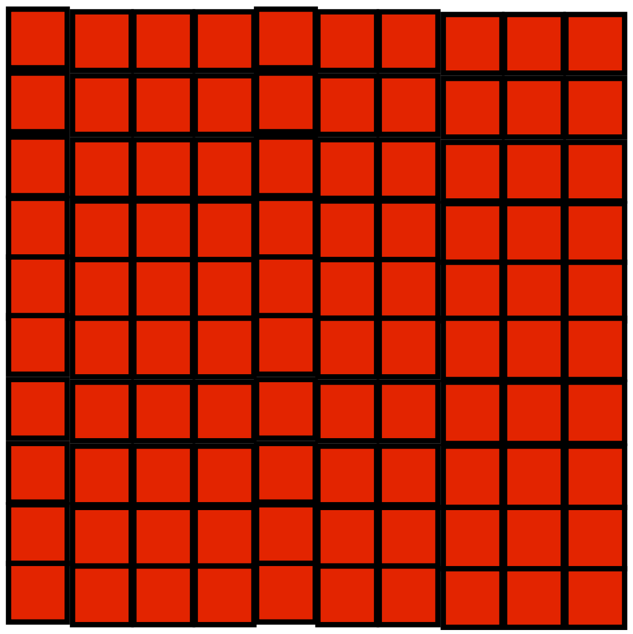
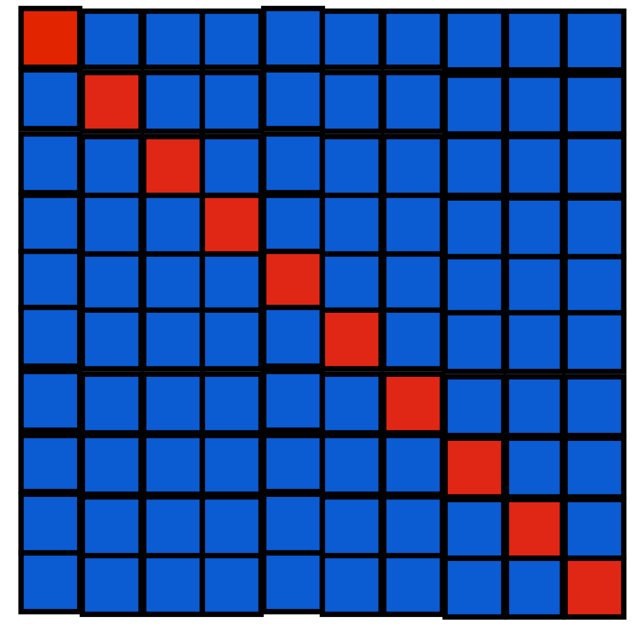


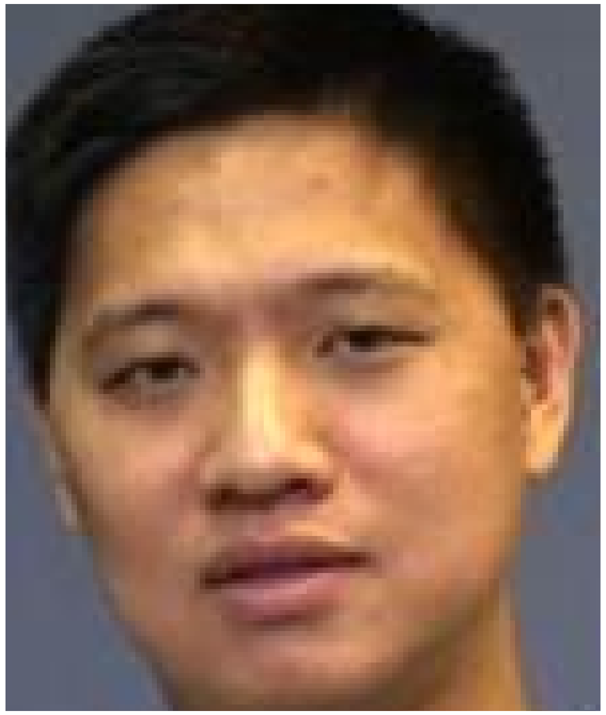
$$\mathbf{F} \leftarrow \text{eigenvectors of } \mathbf{S}$$

Not Always Zero   Always Zero

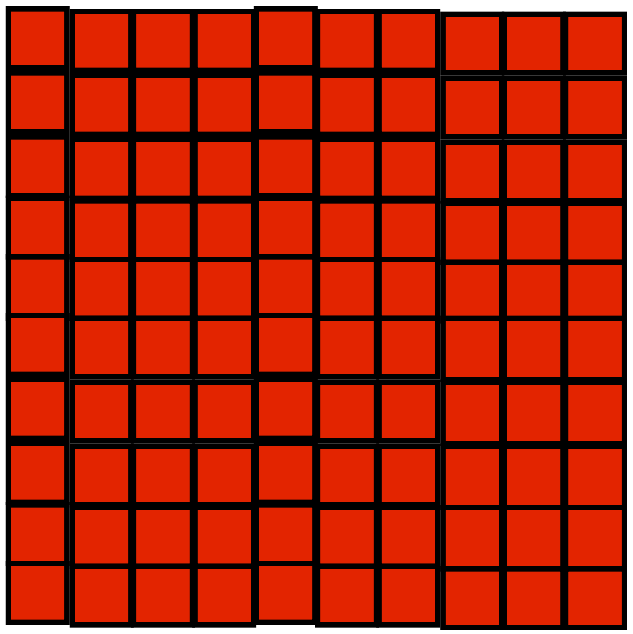
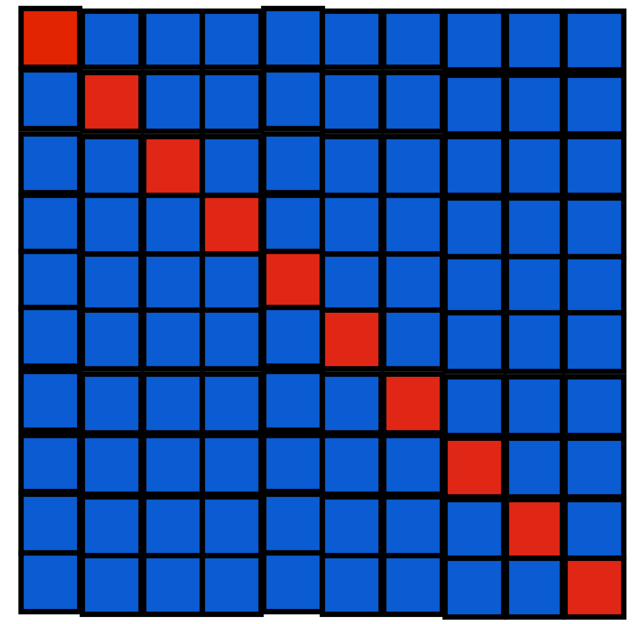


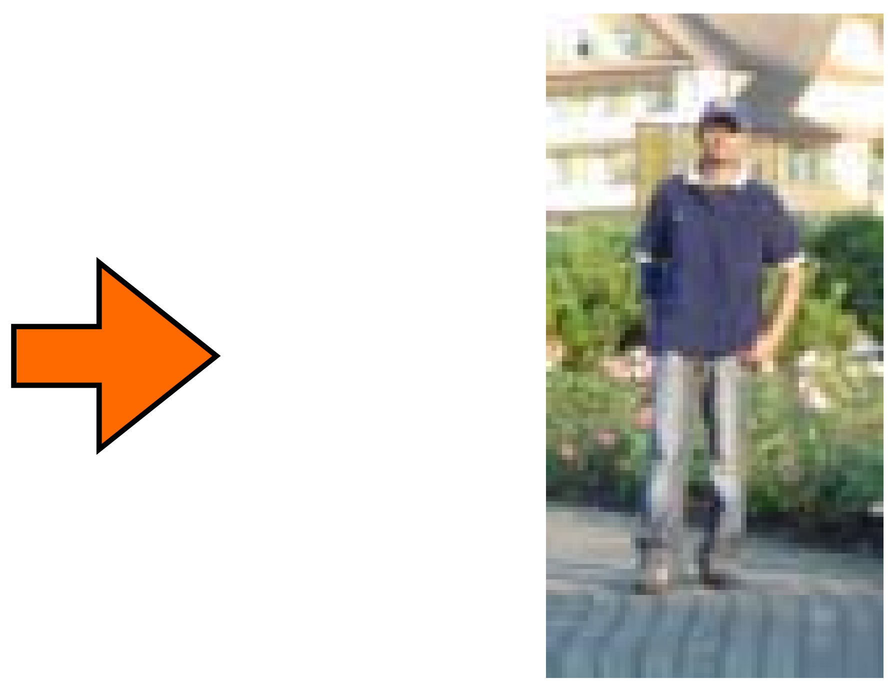
$\mathbf{F}^T$
$\mathbf{F}$
$=$
$\mathbf{x}[\boldsymbol{\tau}]$
Not Always Zero
Always Zero

$\mathbf{F}^T$  $\mathbf{F}$ $=$ 

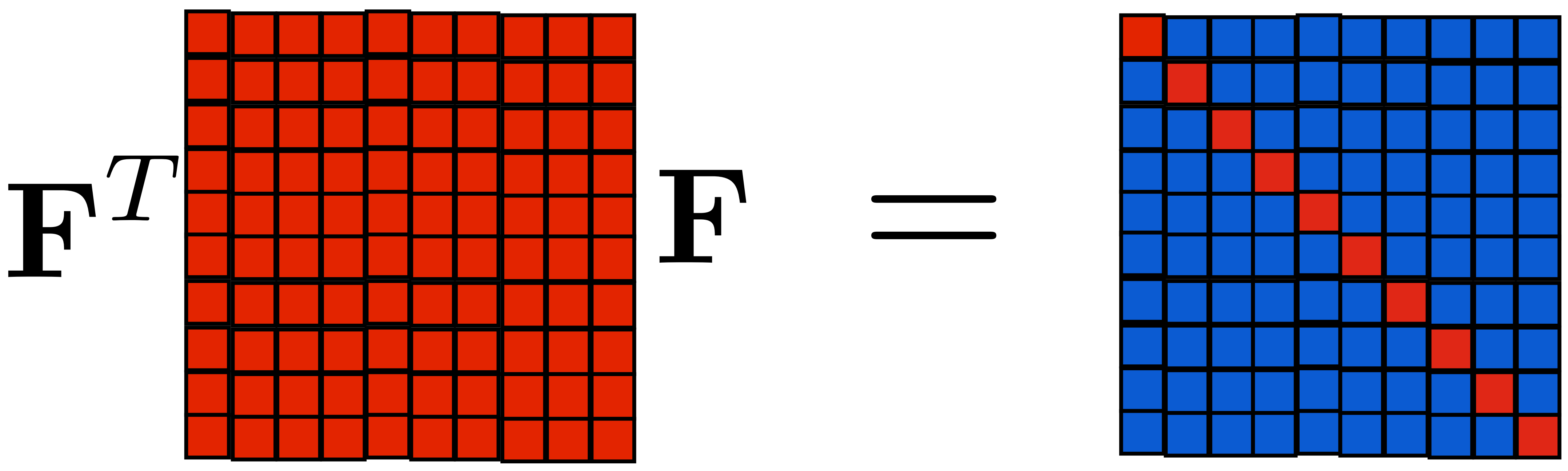
$\mathbf{x}[\boldsymbol{\tau}]$ → 

Not Always Zero   Always Zero

$\mathbf{F}^T$  $\mathbf{F}$ $=$ 

$\mathbf{x}[\boldsymbol{\tau}]$ 

Not Always Zero (red) Always Zero (blue)



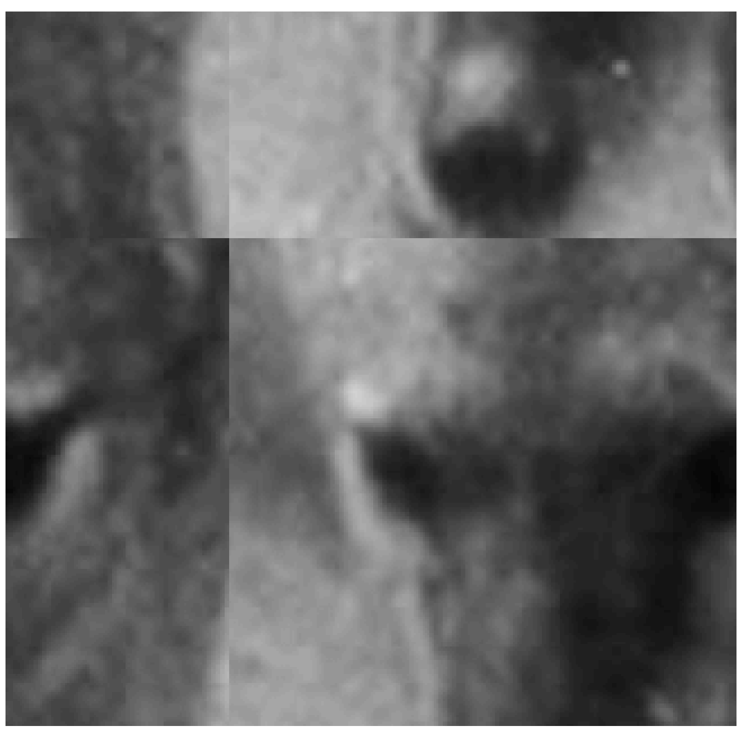
$$\mathbf{F} \leftarrow \text{Fourier Transform}$$

Not Always Zero   Always Zero

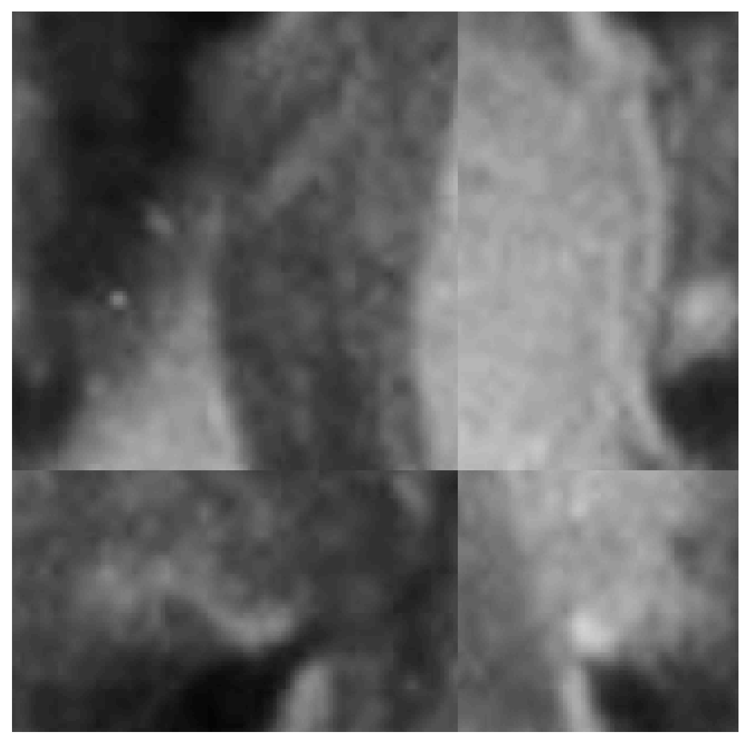
$\mathbf{x}[0,0]$

$\mathbf{x}[20,20]$

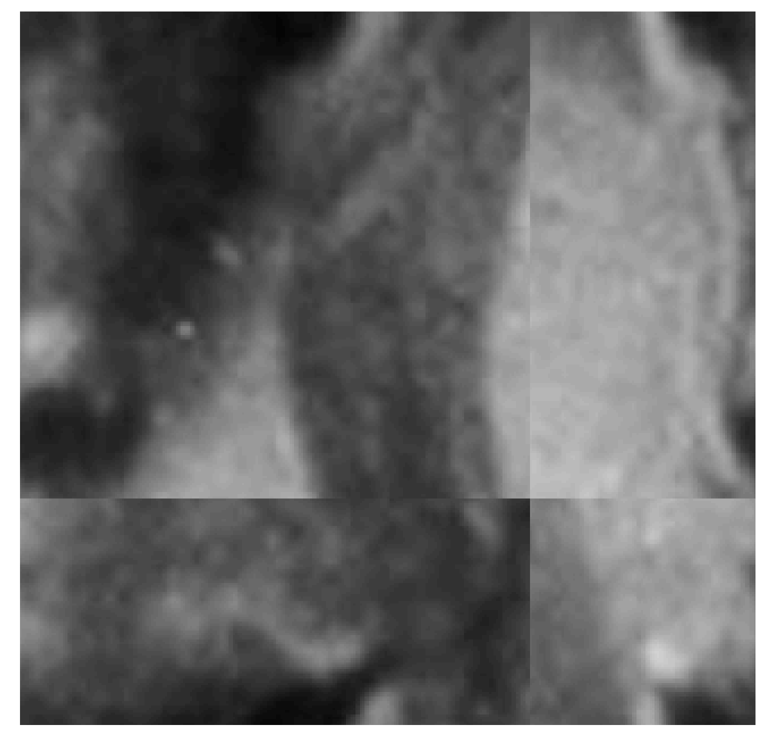
$\mathbf{x}[-20,-20]$



$\mathbf{x}[100,100]$



$\mathbf{x}[-100,-100]$



$\mathbf{x}[200,200]$

$\mathbf{x}[0, 0]$

$\mathbf{x}[20, 20]$

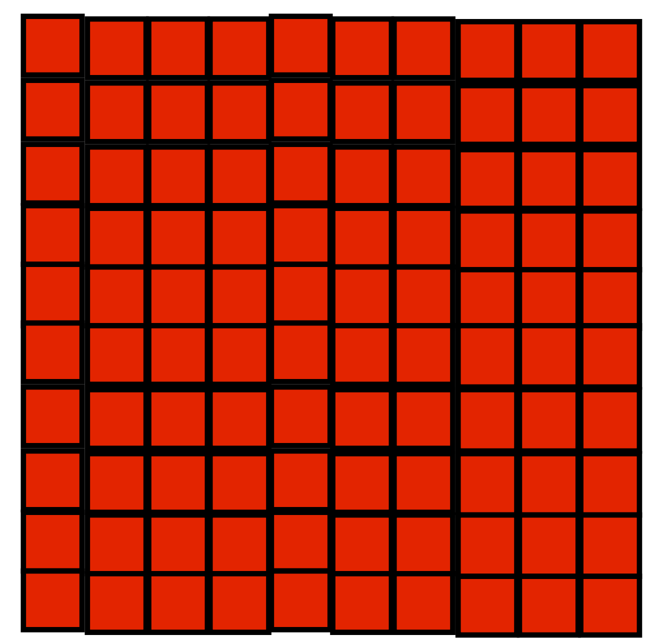
$\mathbf{x}[-20, -20]$

$\mathbf{x}[0, 0]$

$\mathbf{x}[20, 20]$

$\mathbf{x}[-20, -20]$

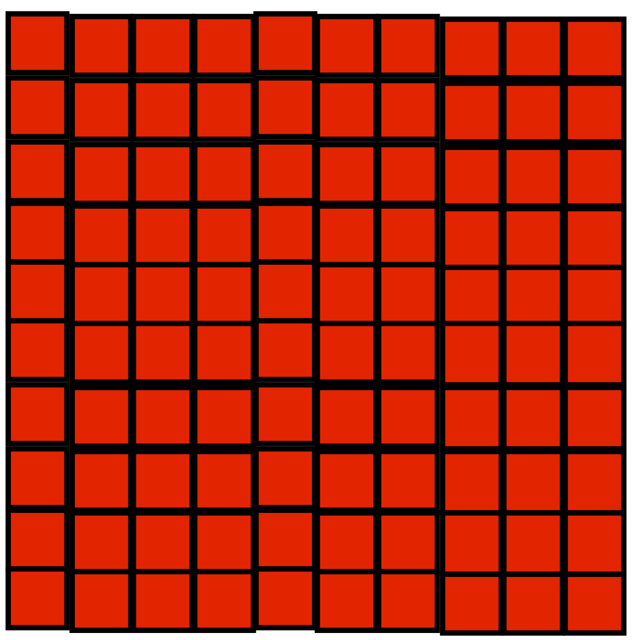
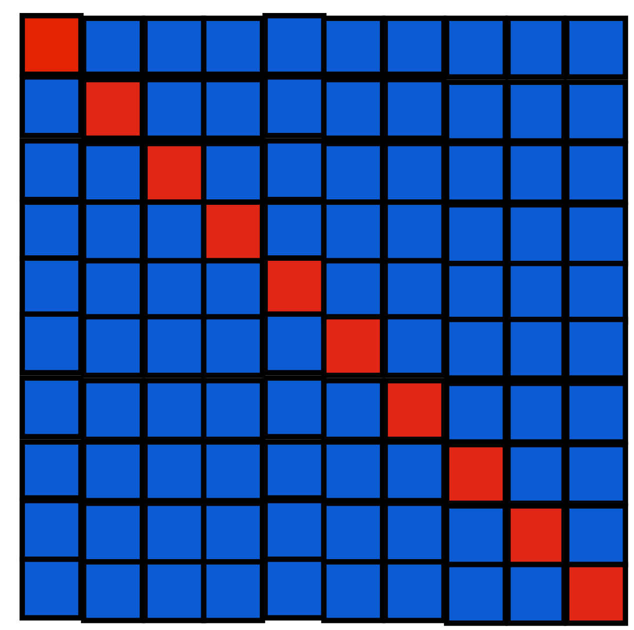
$$\mathbf{S} = \sum_{\boldsymbol{\tau} \in \mathcal{C}'} \mathbf{x}[\boldsymbol{\tau}]\mathbf{x}[\boldsymbol{\tau}]^T =$$



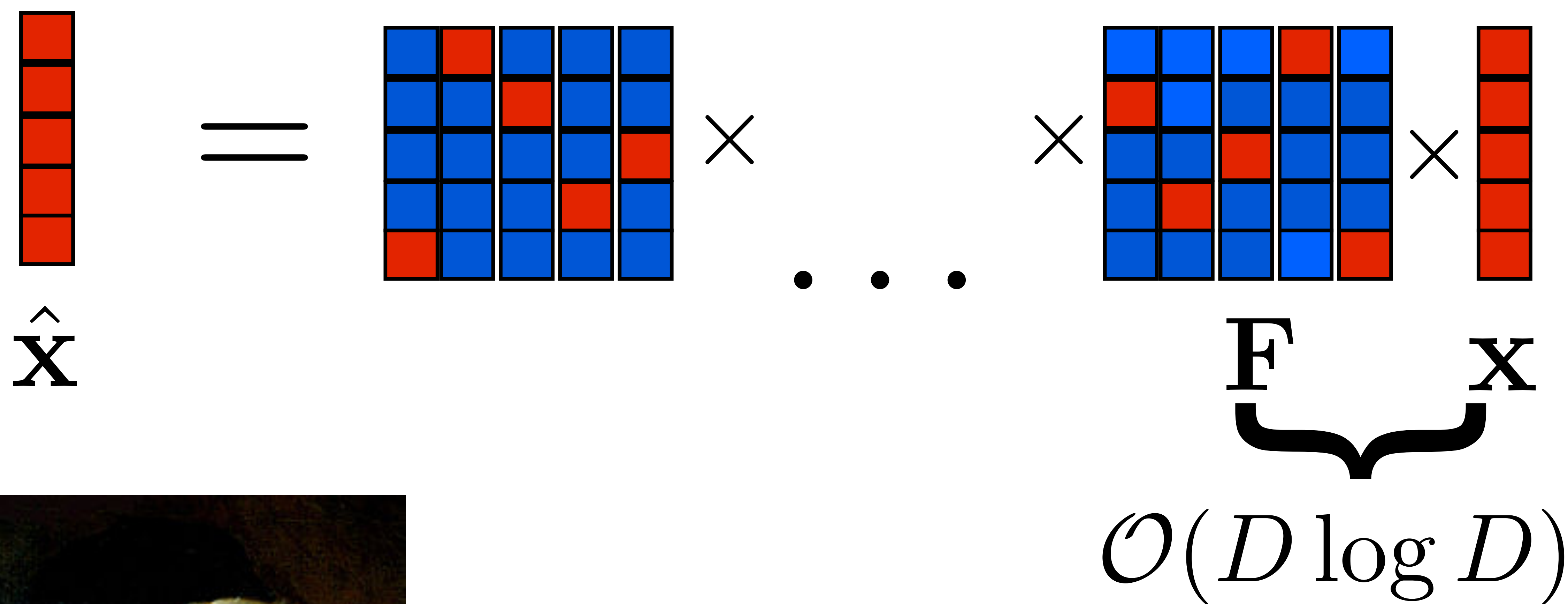
$(D \times D)$

"subset of all circular shifts" $\mathcal{C}' \subseteq \mathcal{C}$

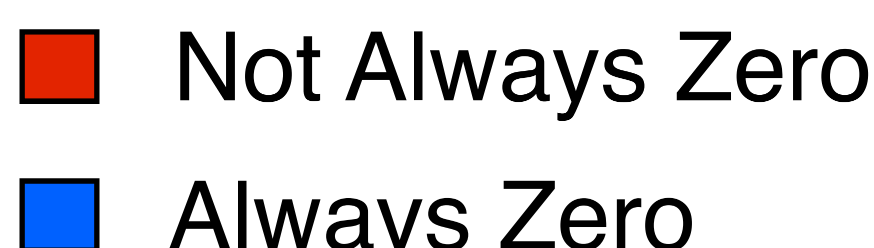
$\mathbf{x}[0,0]$ $\mathbf{x}[20,20]$ $\mathbf{x}[-20,-20]$

$\mathbf{V}^T$  $\mathbf{V}$ $=$ 

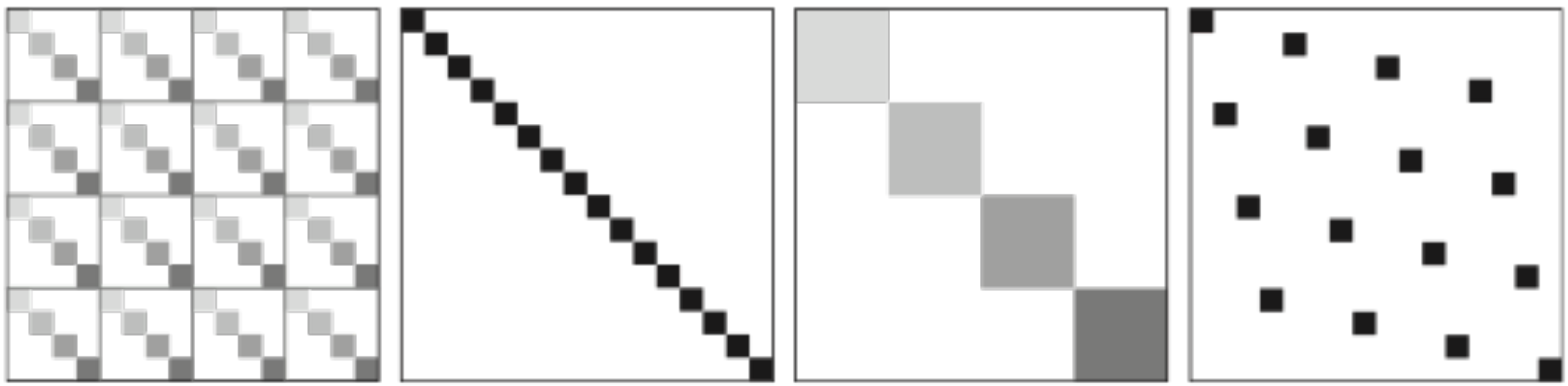
$$\mathbf{V} \neq \mathbf{F}$$




Carl Friedrich Gauss



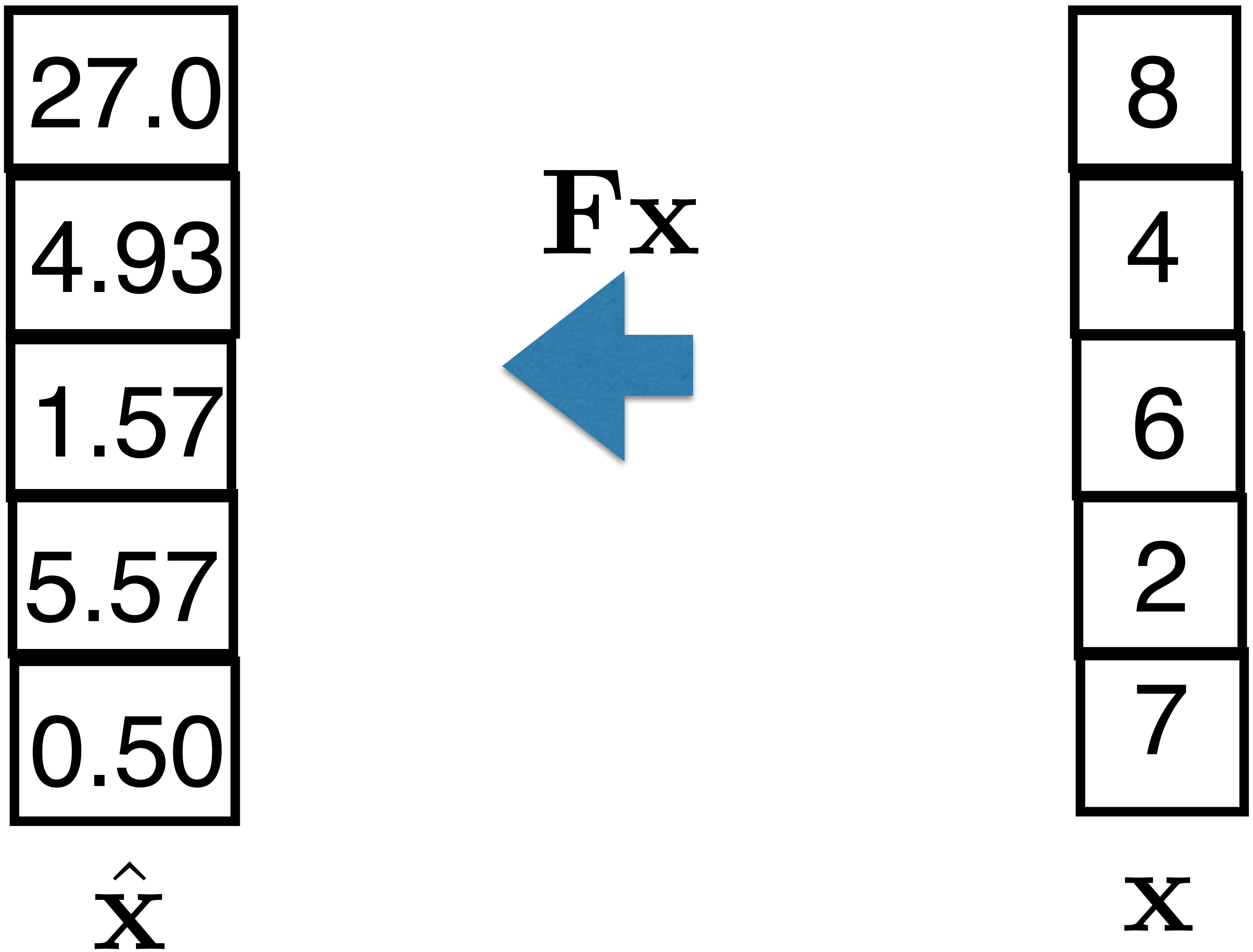
$$\mathbf{F}_{16} = \quad \mathbf{F}_4 \otimes \mathbf{I}_4 \quad \mathbf{T}_4^{16} \quad \mathbf{I}_4 \otimes \mathbf{F}_4 \quad \mathbf{L}_4^{16}$$



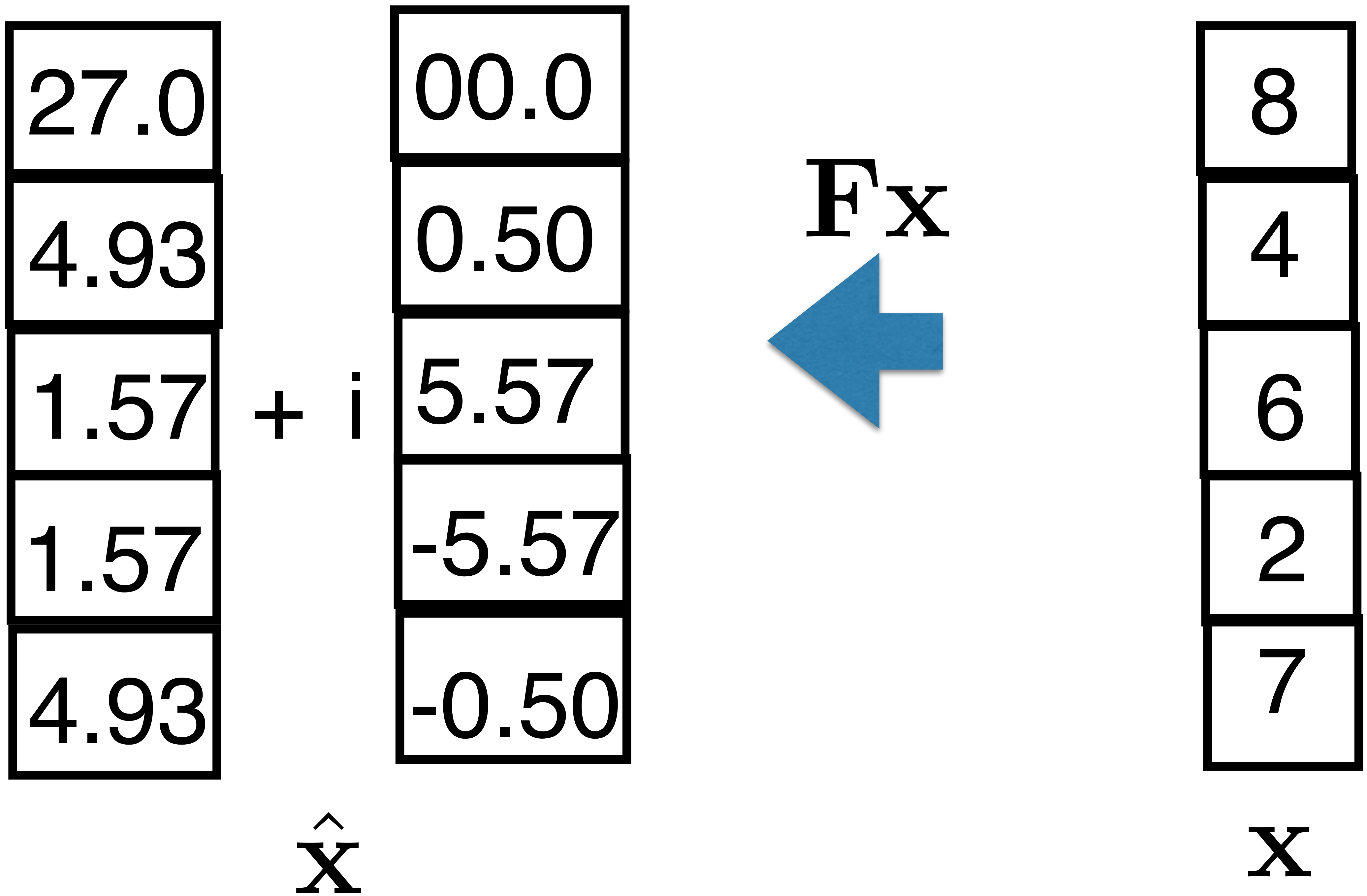
$\mathbf{F}_{16} \rightarrow$ 16 dimensional FFT $\mathbf{F}_4 \rightarrow$ 4 dimensional FFT

$\mathbf{L}_4^{16} \rightarrow$ permutation matrix $\mathbf{T}_4^{16} \rightarrow$ diagonal matrix

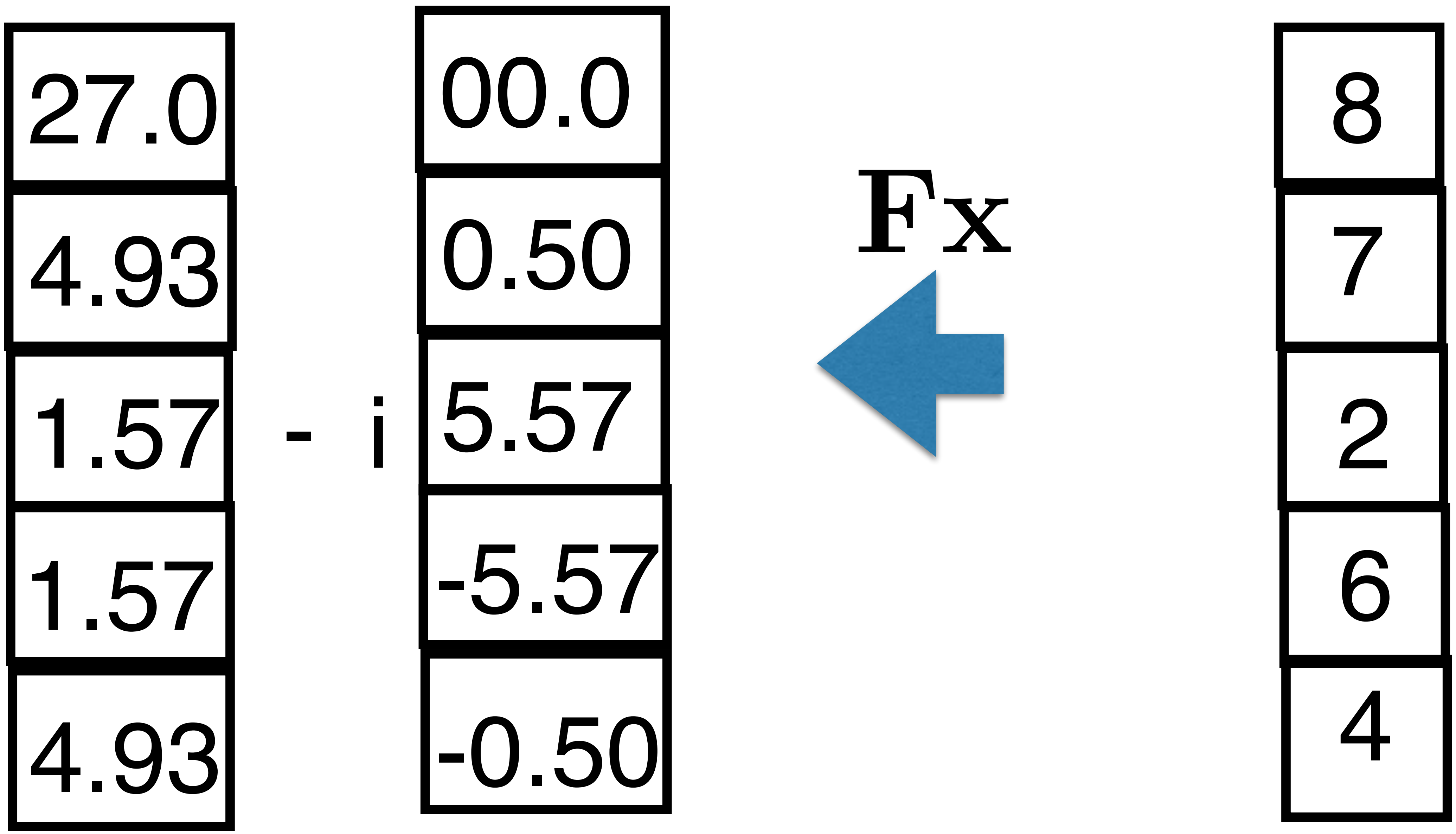
FFT can be Real



FFT can be Complex

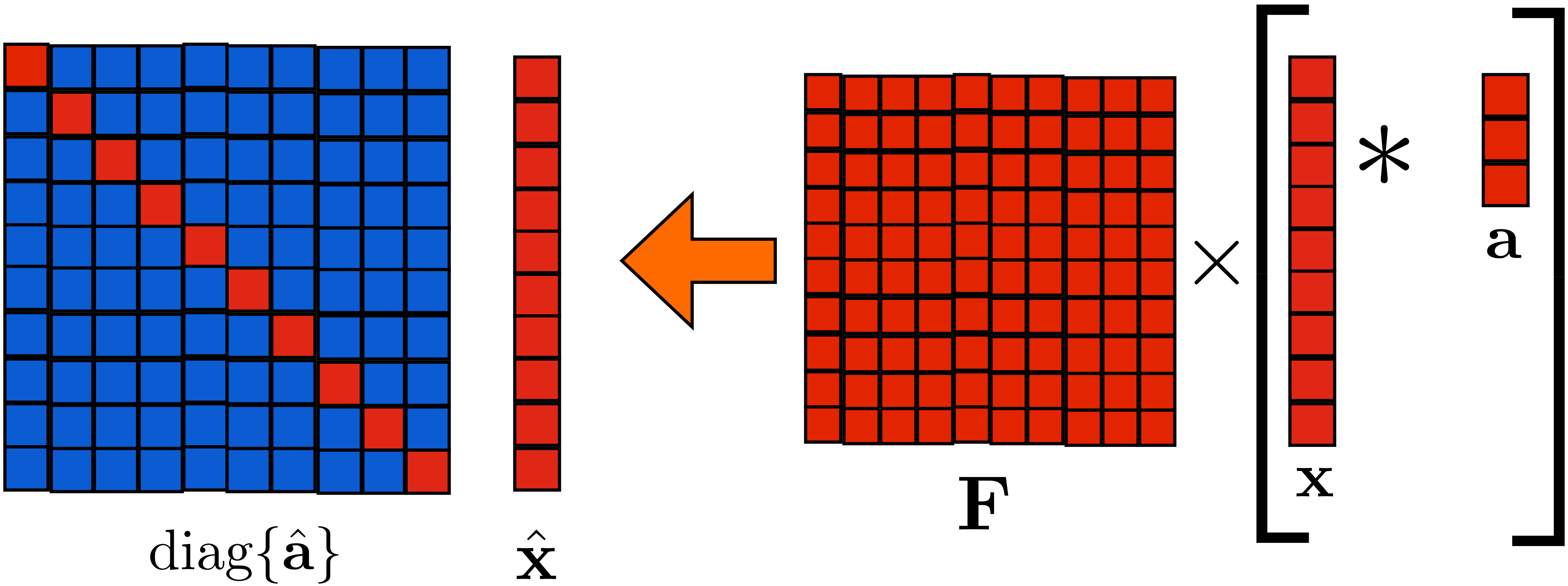


FFT can be Complex

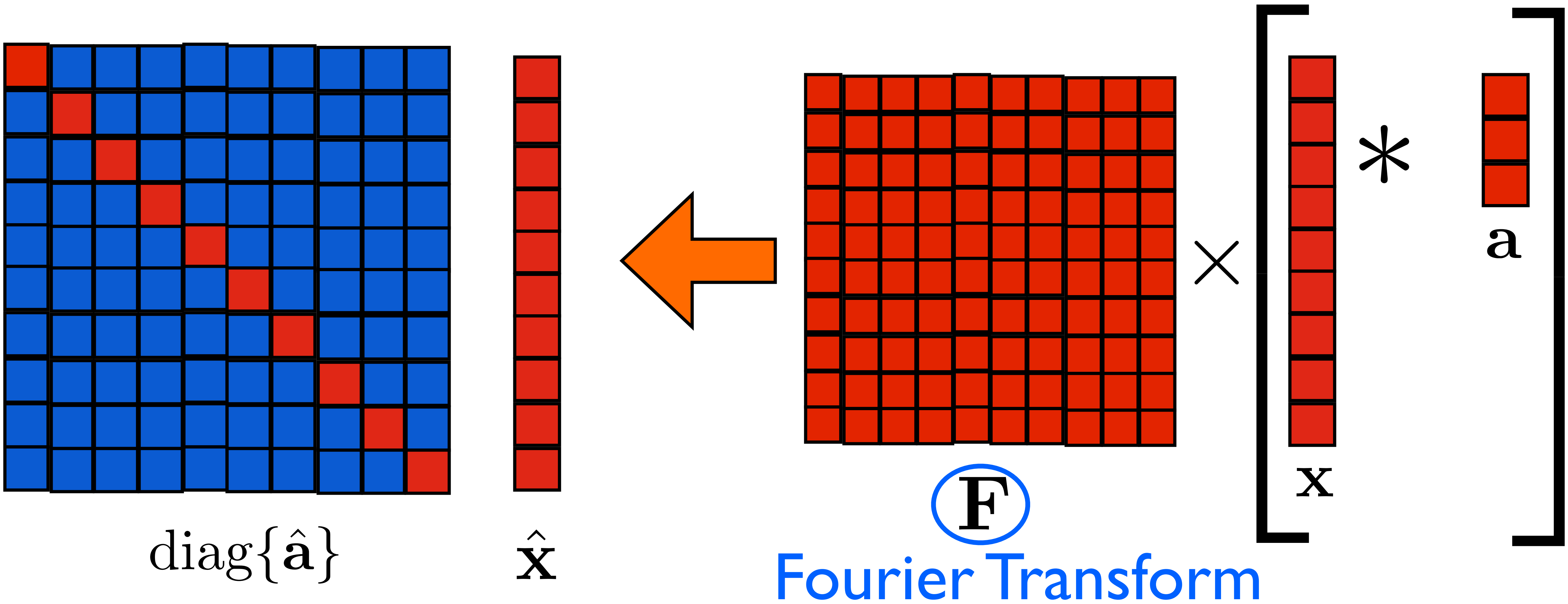


$\mathrm{conj}(\hat{\mathbf{x}})$

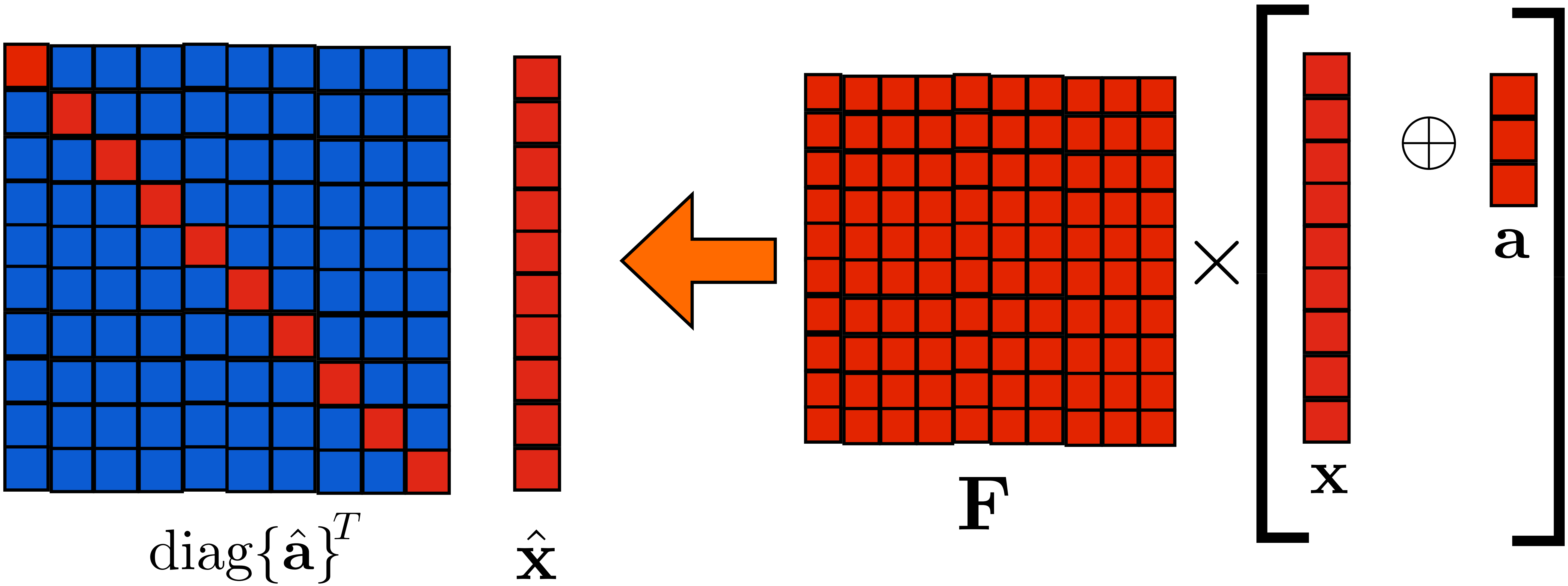
```
>> flipud(circshift(x,4))
```



$$\text{diag}\{\hat{\mathbf{a}}\}\hat{\mathbf{x}} = \mathbf{F}(\mathbf{a} * \mathbf{x})$$



$$\hat{\mathbf{a}} \circ \hat{\mathbf{x}} = \mathbf{F}(\mathbf{a} * \mathbf{x})$$



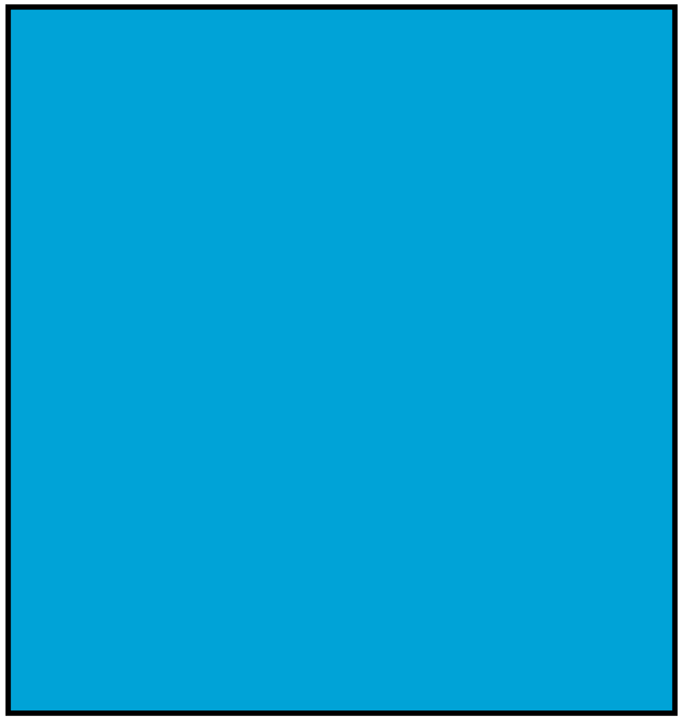
$$\mathrm{conj}\{\hat{\mathbf{a}}\} \circ \hat{\mathbf{x}} = \mathbf{F}(\mathbf{x} \oplus \mathbf{a})$$

Today

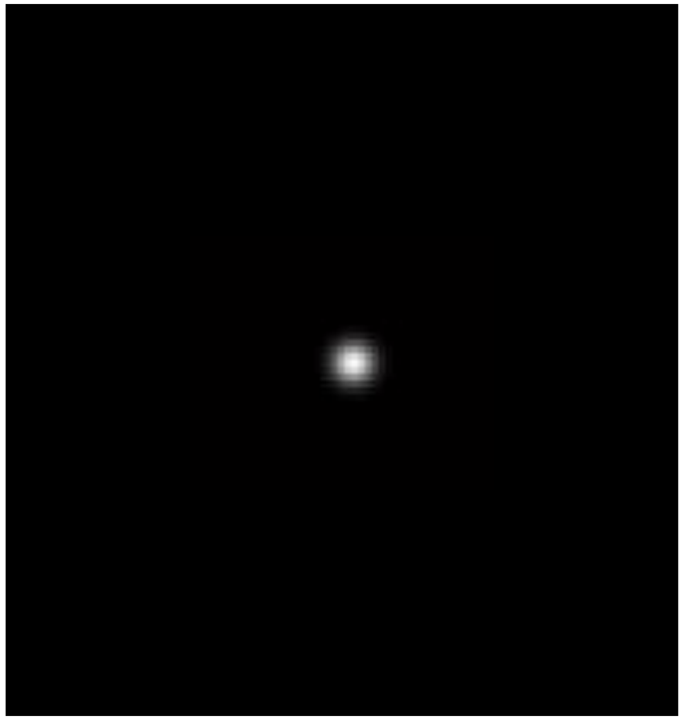
- Types of Convolution
- Fast Fourier Transform (FFT)
- The Correlation Filter

“known signal” $\mathbf{x}$

$*$

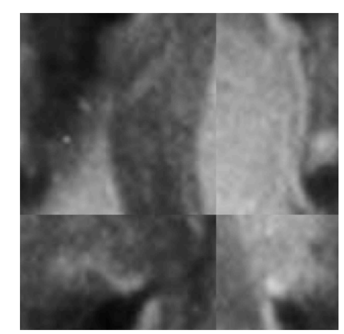


“unknown filter” $\mathbf{h}$

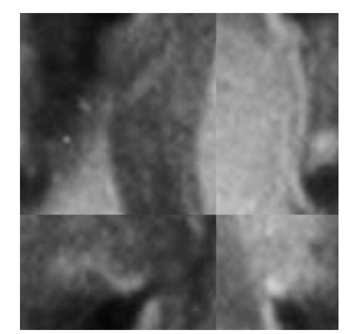


“known response” $\mathbf{y}$

$$E(\mathbf{h}) = \frac{1}{2} \sum_{\boldsymbol{\tau} \in \mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}||_2^2$$

$\mathbf{x}[\boldsymbol{\tau}]$			$\cdots$	
$y_{\boldsymbol{\tau}}$	1	0	$\cdots$	0

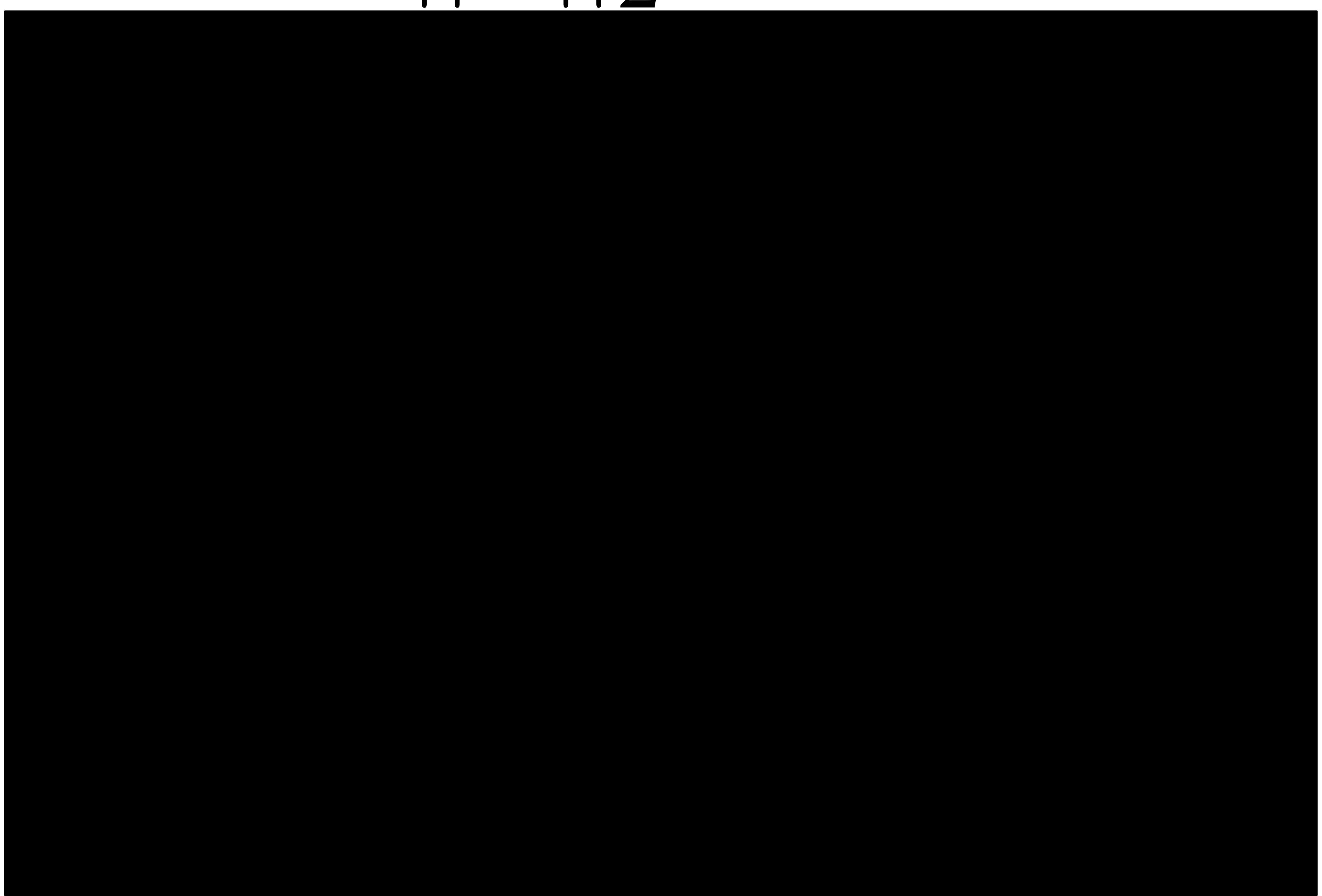
$$E(\mathbf{h}) = \frac{1}{2}\sum_{\boldsymbol{\tau}\in\mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$

$\mathbf{x}[\boldsymbol{\tau}]$			$\cdots$	
$y_{\boldsymbol{\tau}}$	1	0	$\cdots$	0

Trust Regions

$$E(\mathbf{h}) = \frac{1}{2} \sum_{\boldsymbol{\tau} \in \mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}||_2^2$$

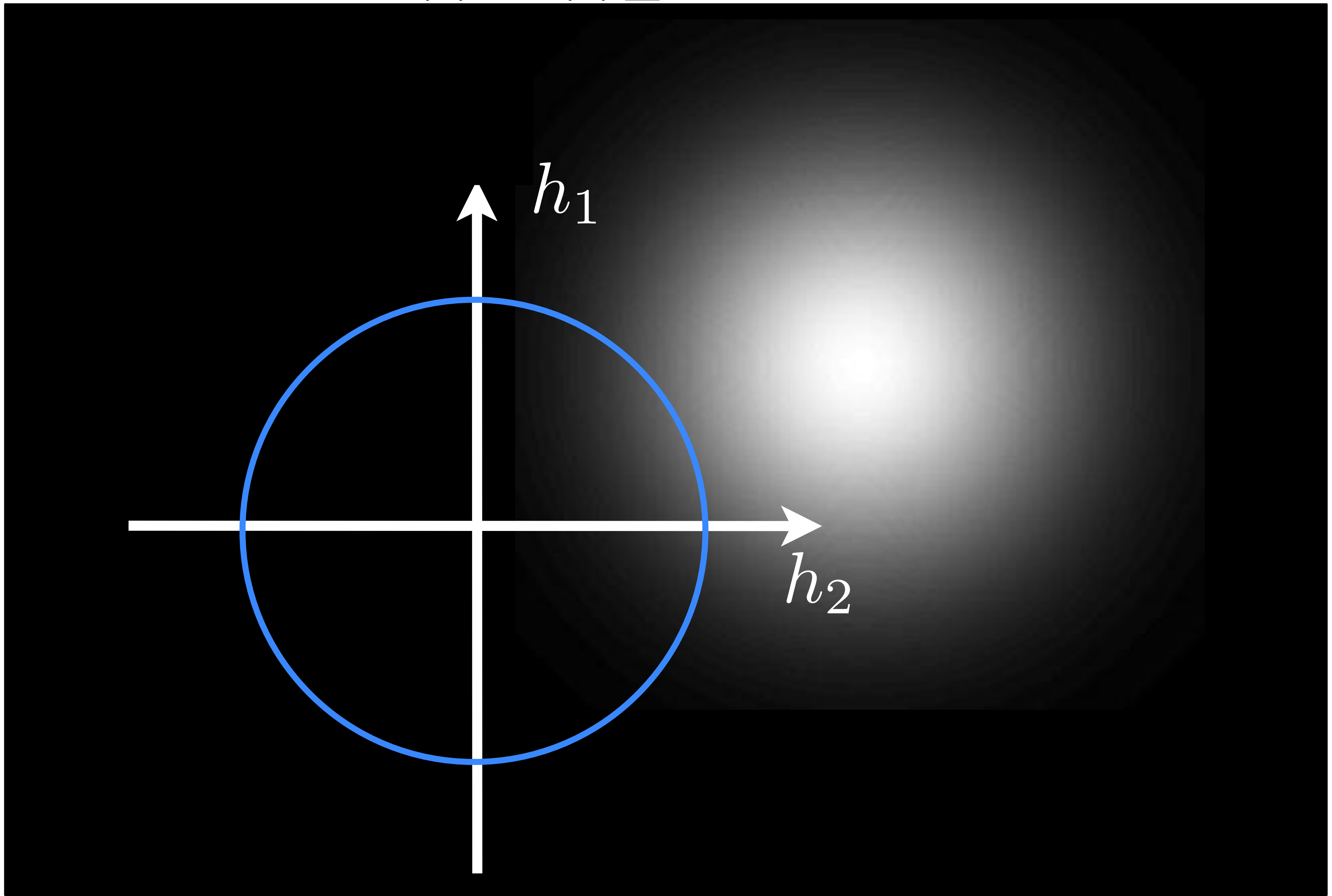
$$\text{s.t. } ||\mathbf{h}||_2^2 \leq \epsilon$$



Trust Regions

$$E(\mathbf{h}) = \frac{1}{2} \sum_{\boldsymbol{\tau} \in \mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}||_2^2$$

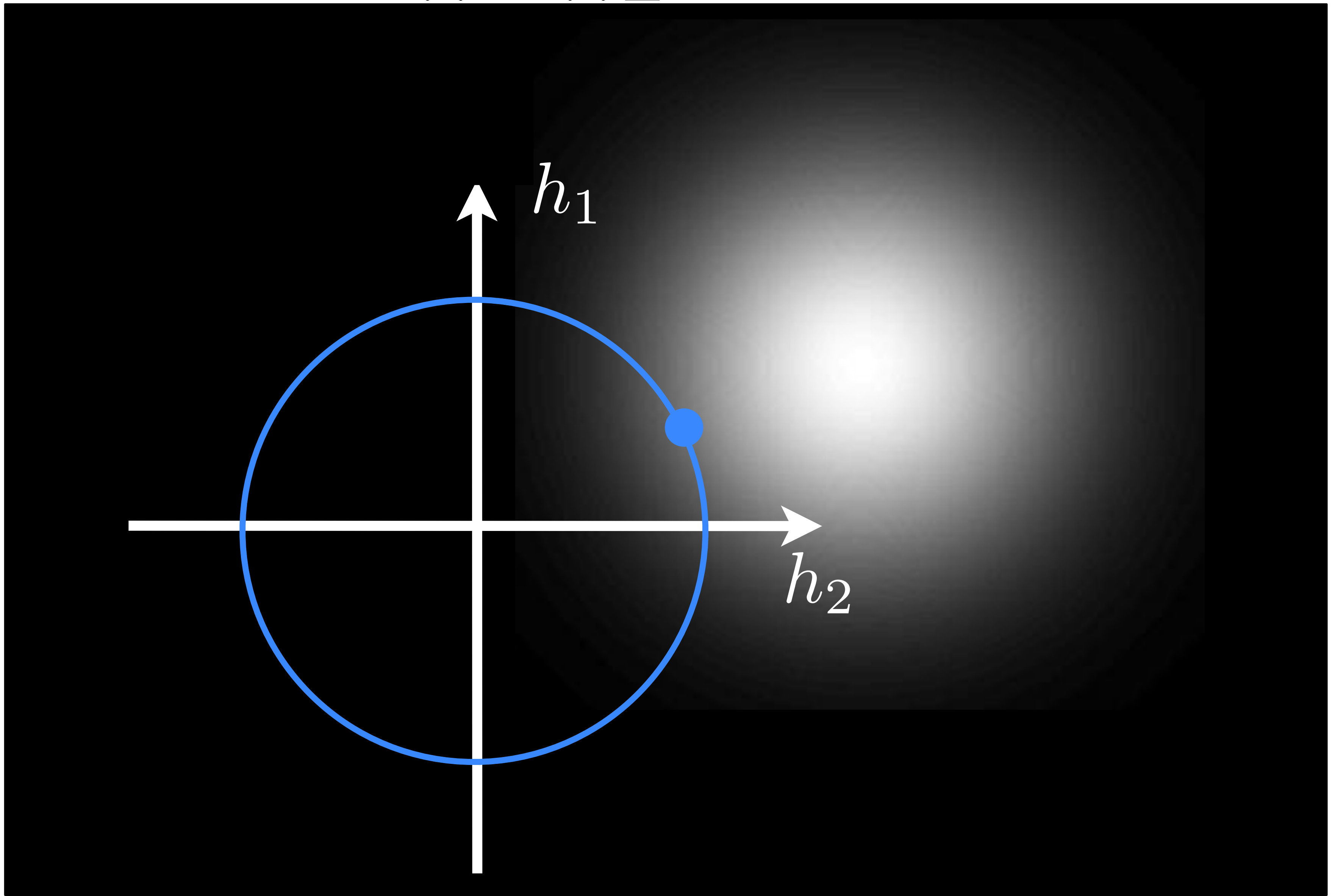
$$\text{s.t. } ||\mathbf{h}||_2^2 \leq \epsilon$$



Trust Regions

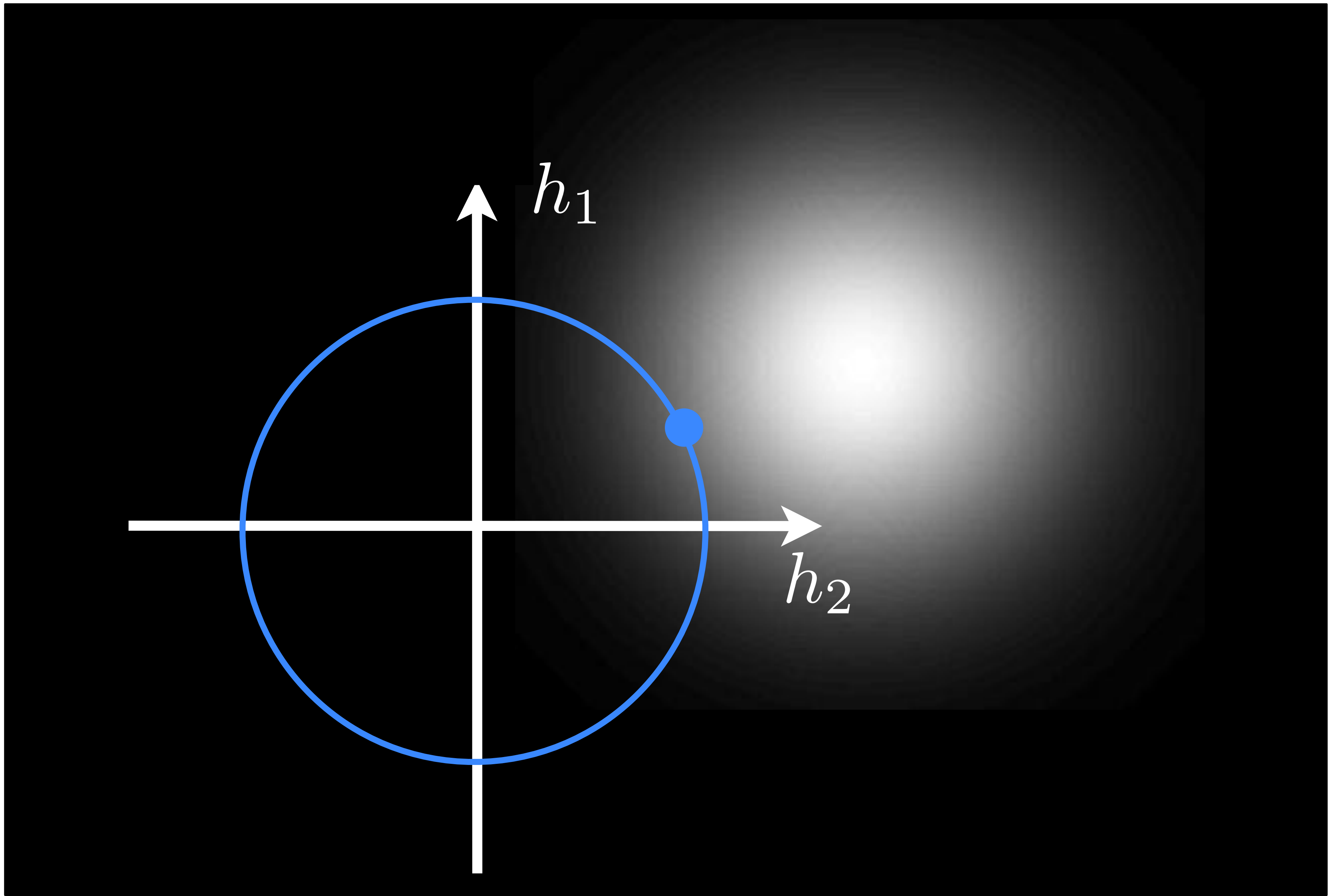
$$E(\mathbf{h}) = \frac{1}{2}\sum_{\boldsymbol{\tau}\in\mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T\mathbf{h}||_2^2$$

$$\text{s.t.} \ \ ||\mathbf{h}||_2^2 \leq \epsilon$$



Trust Regions

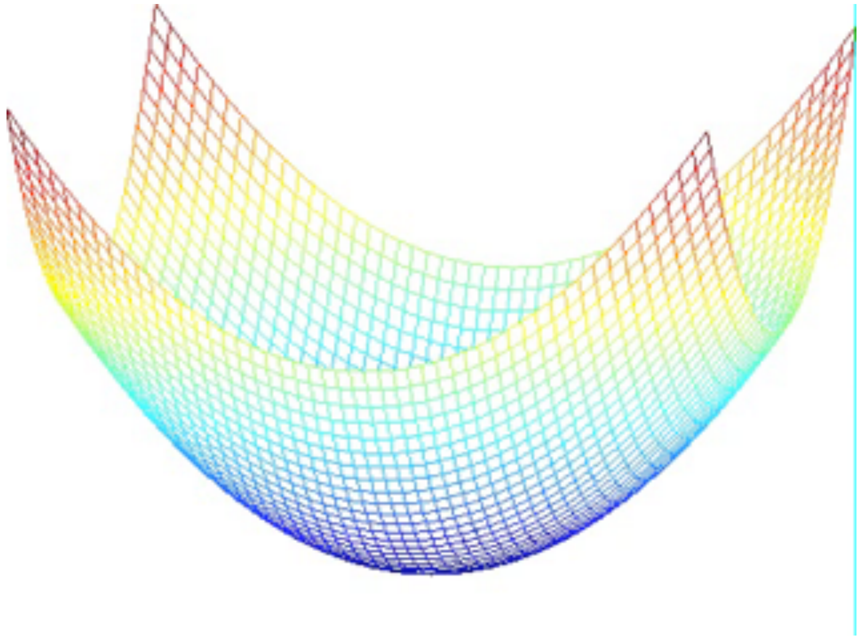
$$E(\mathbf{h}) = \frac{1}{2} \sum_{\boldsymbol{\tau} \in \mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}||_2^2 + \frac{\lambda}{2} ||\mathbf{h}||_2^2$$



Linear Least Squares Discriminant

- One can view a correlation filter in the spatial domain as a linear least squares discriminant.
- Made popular by Bolme et al., referred to in literature as a MOSSE filter.

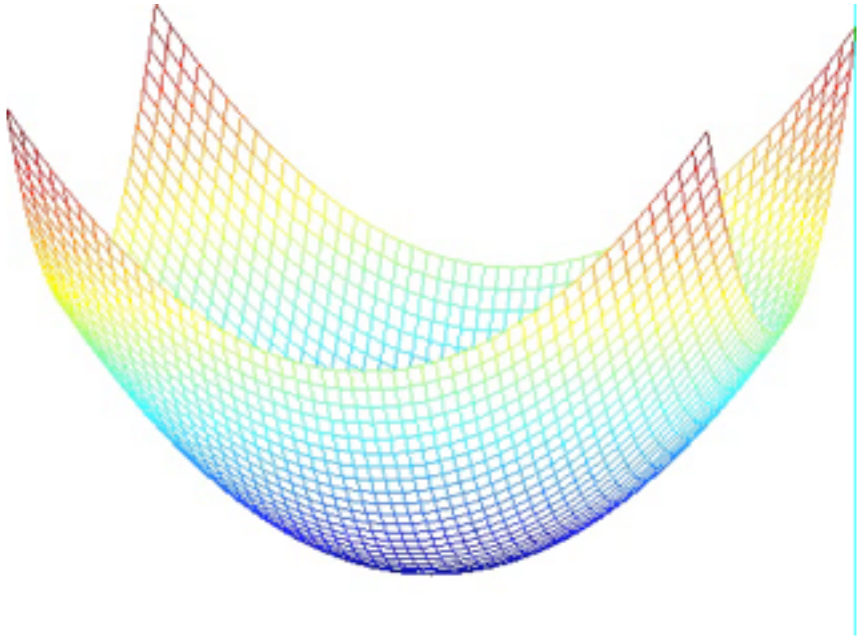
$$\arg\min_{\mathbf{w},w_0} \sum_{n=1}^{N} ||t_i - \mathbf{w}^T\mathbf{x}_i - w_0||_2^2$$



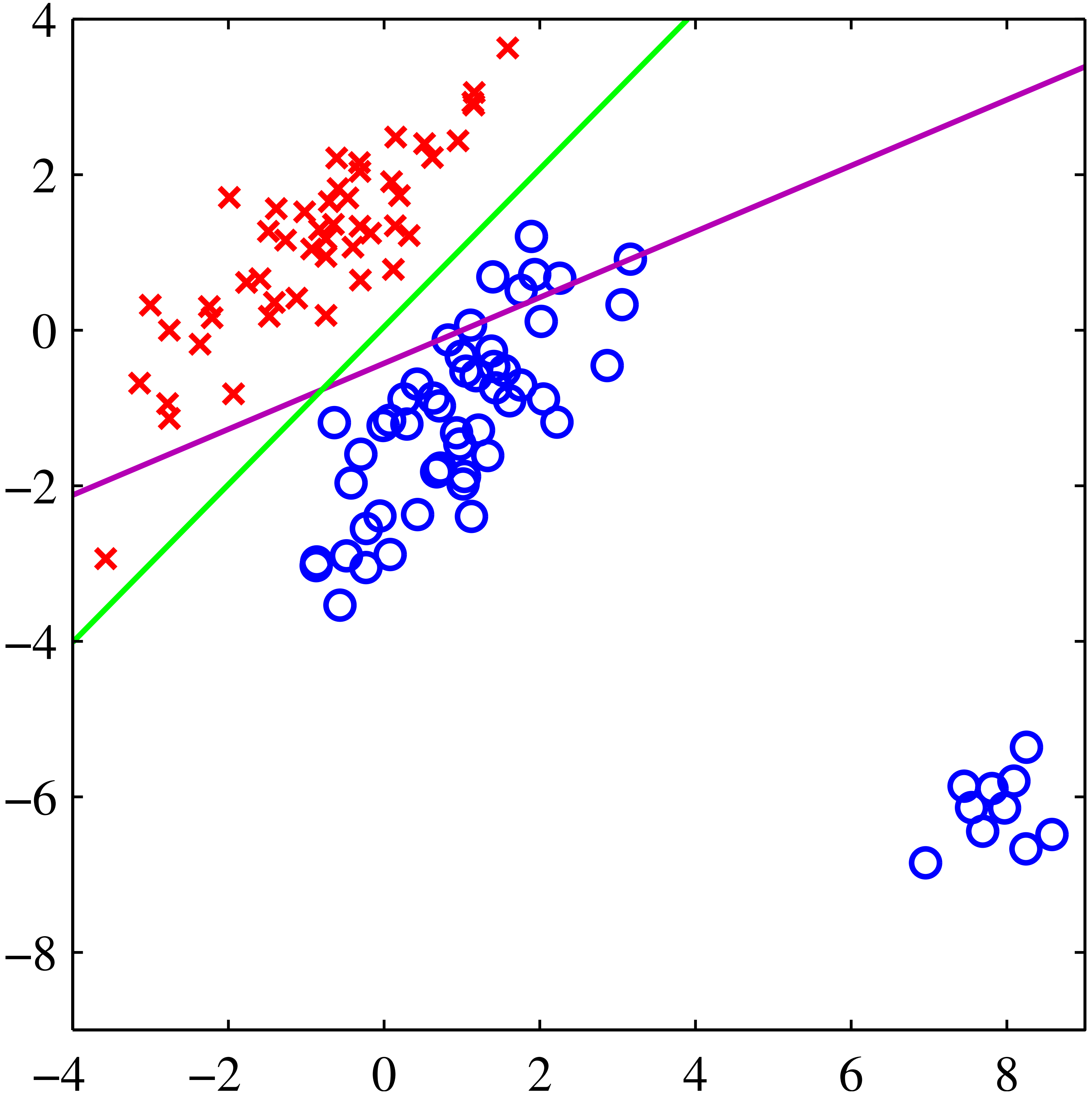
Linear Least Squares Discriminant

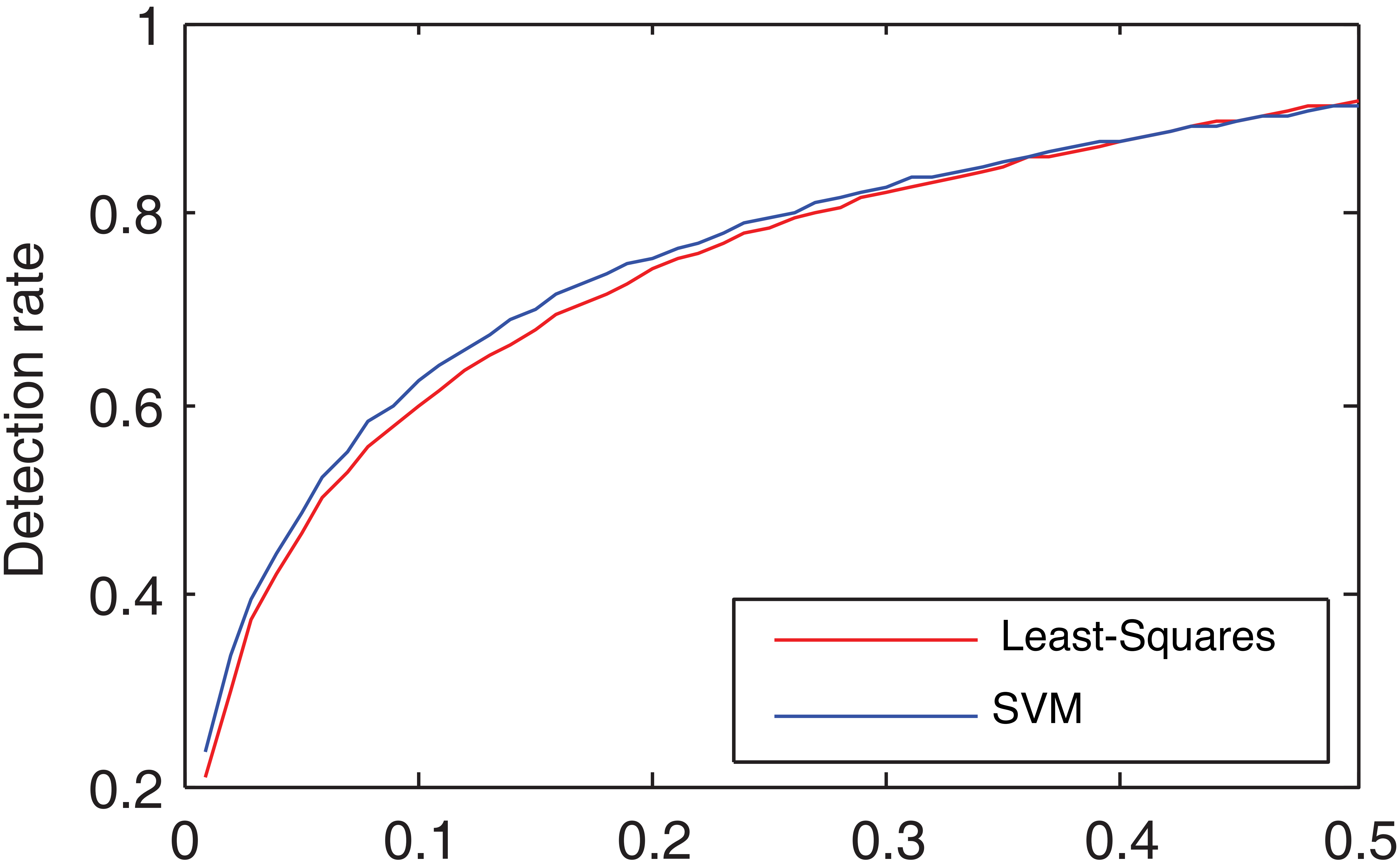
- One can view a correlation filter in the spatial domain as a linear least squares discriminant.
- Made popular by Bolme et al., referred to in literature as a MOSSE filter.

$$\arg\min_{\mathbf{w}, w_0} \sum_{n=1}^{N} ||t_i - \mathbf{w}^T \mathbf{x}_i - w_0||_2^2$$

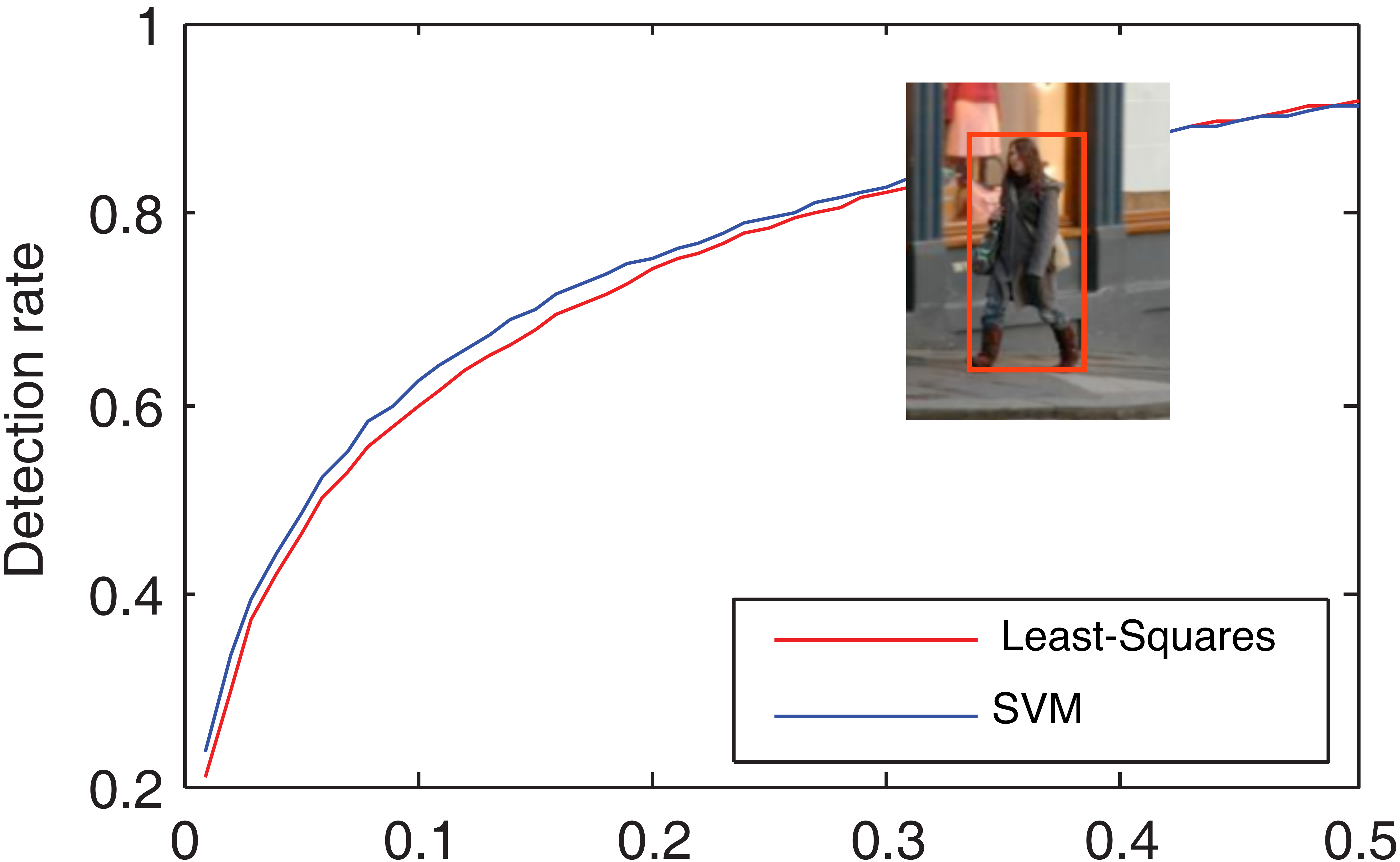


Linear Least Squares Discriminant



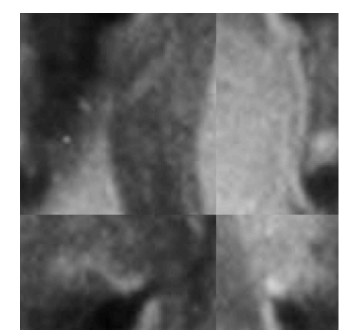


Galoogahi, Sim & Lucey "Multi-Channel Correlation Filters", ICCV 2013.



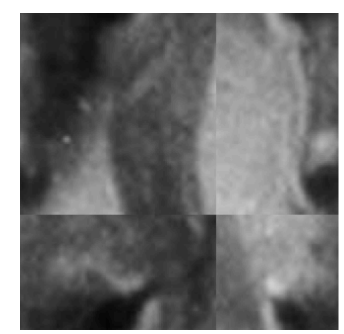
Galoogahi, Sim & Lucey "Multi-Channel Correlation Filters", ICCV 2013.

$$E(\mathbf{h}) = \frac{1}{2} \sum_{\boldsymbol{\tau} \in \mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}||_2^2 + \frac{\lambda}{2} ||\mathbf{h}||_2^2$$

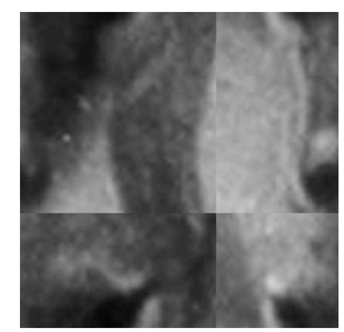
$\mathbf{x}[\boldsymbol{\tau}]$			$\cdots$	
$y_{\boldsymbol{\tau}}$	1	0	$\cdots$	0

$$E(\mathbf{h}) = \frac{1}{2} \sum_{\boldsymbol{\tau} \in \mathcal{C}} ||y_{\boldsymbol{\tau}} - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}||_2^2 + \frac{\lambda}{2} ||\mathbf{h}||_2^2$$

"set of all circular shifts"

$\mathbf{x}[\boldsymbol{\tau}]$			$\cdots$	
$y_{\boldsymbol{\tau}}$	1	0	$\cdots$	0

$$E(\mathbf{h}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x} * \mathbf{h}\|_2^2 + \frac{\lambda}{2}\|\mathbf{h}\|_2^2$$

$\mathbf{x}[\boldsymbol{\tau}]$			$\cdots$	
$y_{\boldsymbol{\tau}}$	1	0	$\cdots$	0

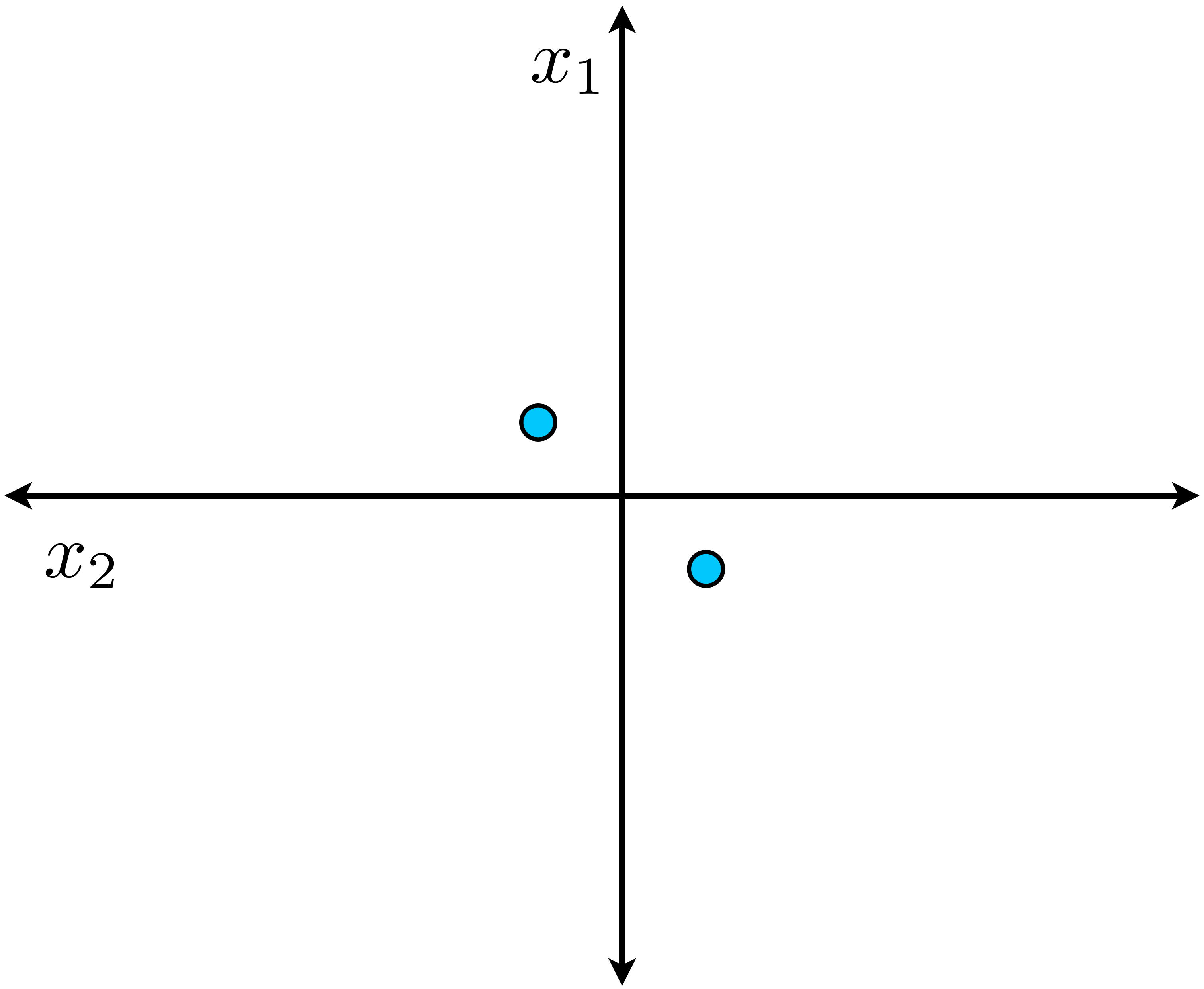
$$E(\mathbf{h}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$

$$E(\mathbf{h}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$

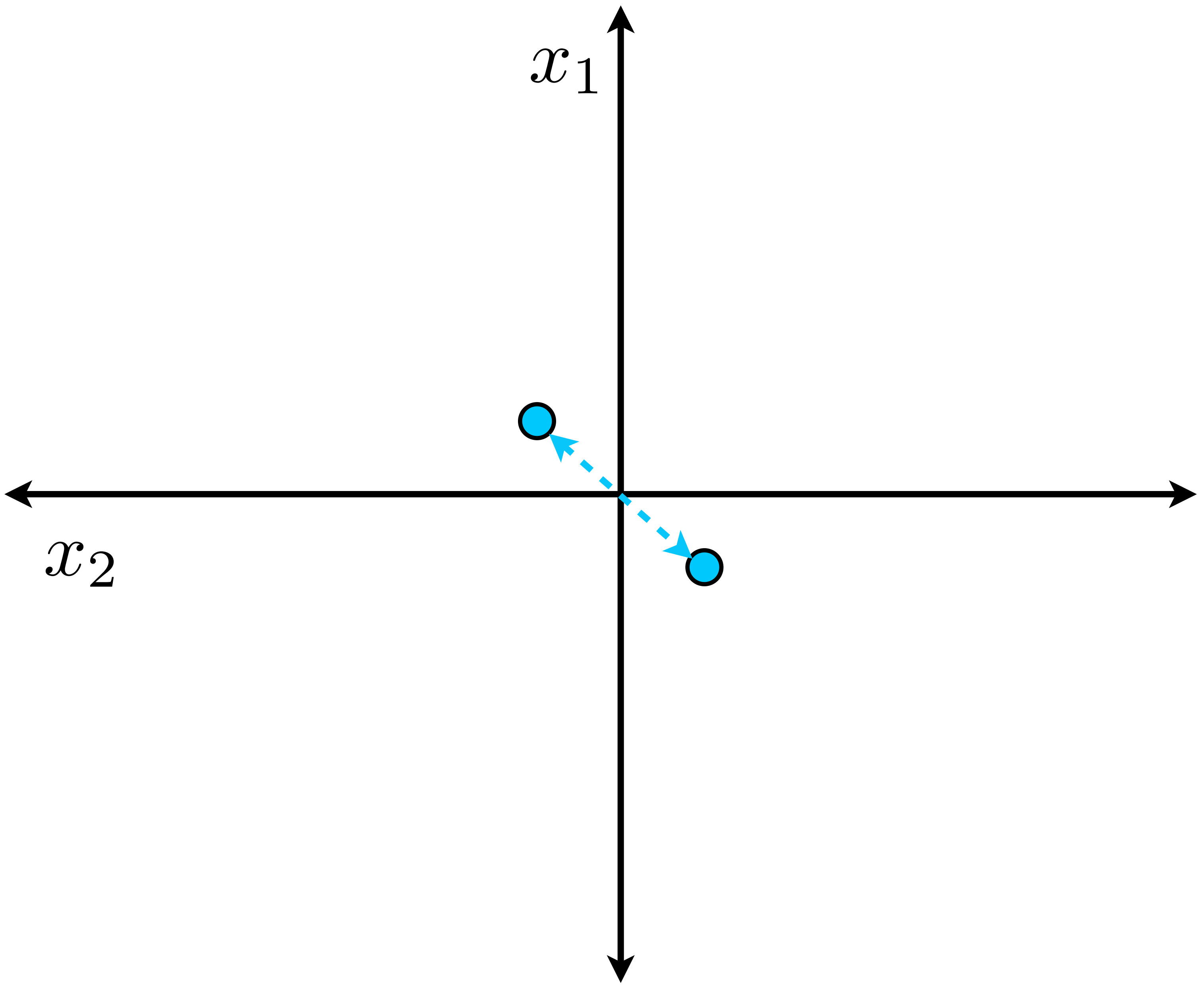
$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} \qquad \mathcal{O}(D^3)$$

$D = \text{number of samples in } \mathbf{x}$

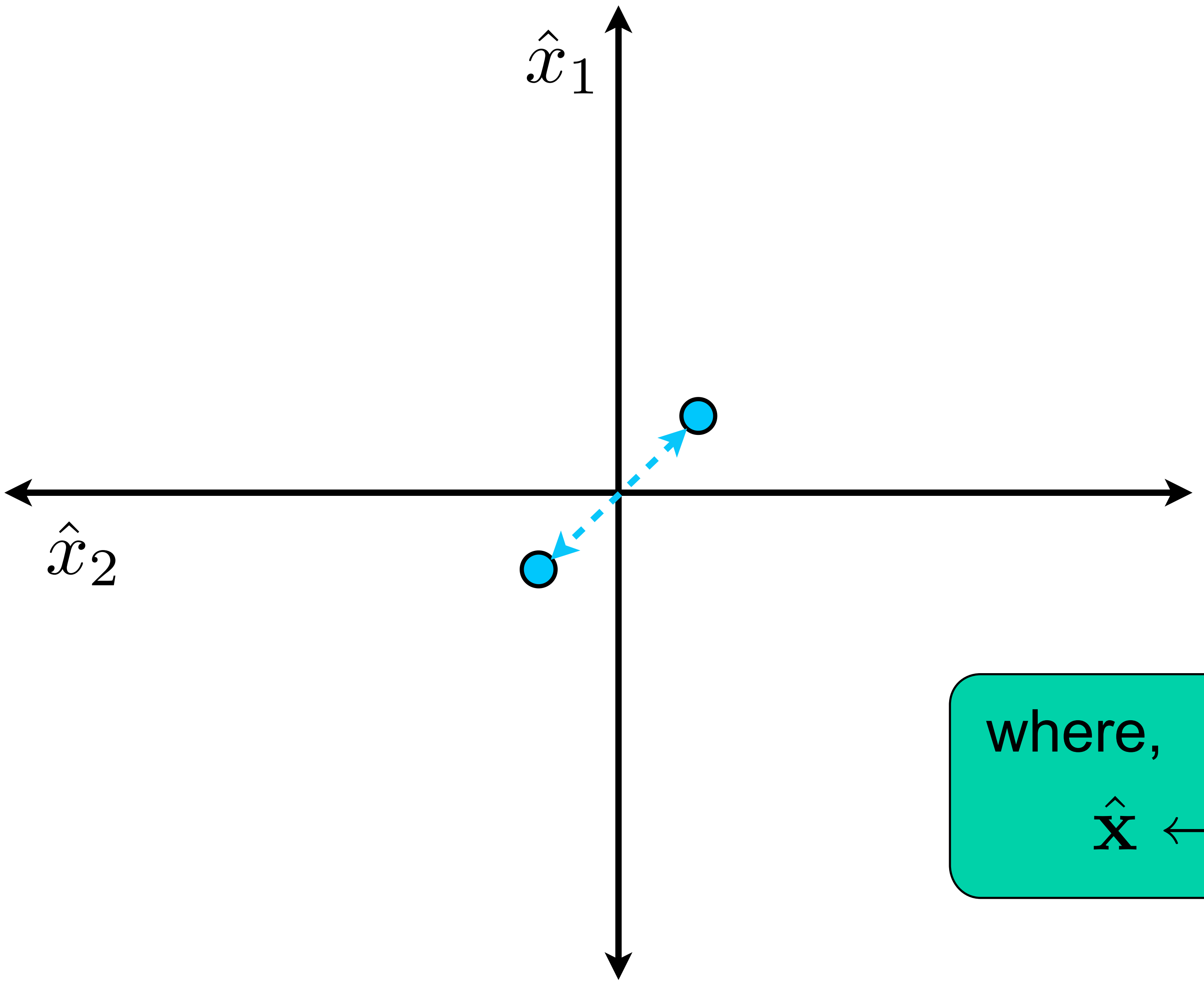
$$E(\mathbf{h}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$



$$E(\mathbf{h}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$



$$E(\hat{\mathbf{h}}) = \frac{1}{2}||\hat{\mathbf{y}} - \mathrm{diag}\{\hat{\mathbf{x}}\}\hat{\mathbf{h}}||_2^2 + \frac{\lambda}{2}||\hat{\mathbf{h}}||_2^2$$



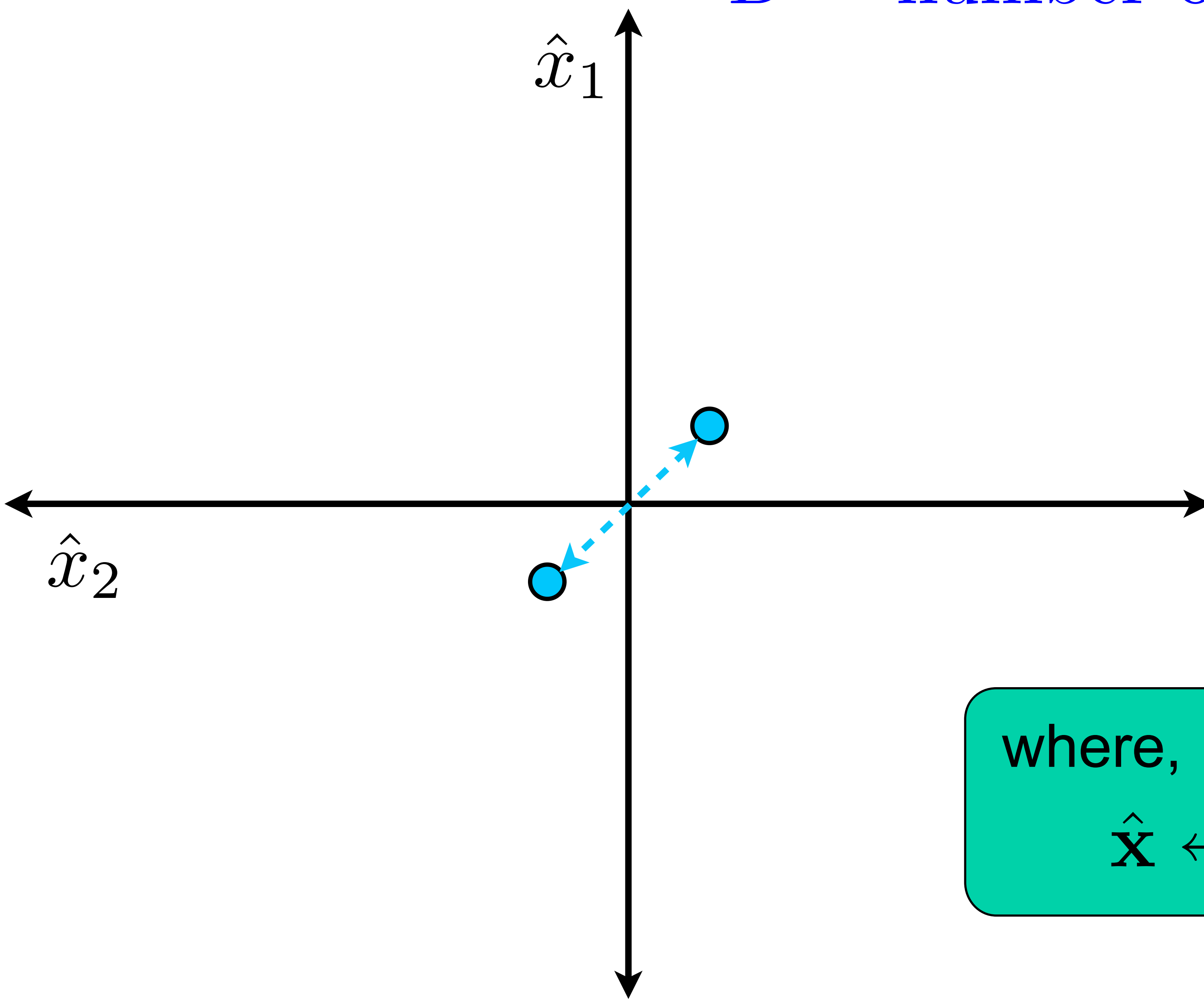
where,

$$\hat{\mathbf{x}} \leftarrow \mathcal{F}\{\mathbf{x}\}$$

$$E(\hat{\mathbf{h}}) = \frac{1}{2}||\hat{\mathbf{y}} - \text{diag}\{\hat{\mathbf{x}}\}\hat{\mathbf{h}}||_2^2 + \frac{\lambda}{2}||\hat{\mathbf{h}}||_2^2$$

$$(\text{diag}(\hat{\mathbf{x}})^T \text{diag}(\hat{\mathbf{x}}) + \lambda \mathbf{I})^{-1} \Rightarrow \mathcal{O}(D \log D)$$

$D =$ number of samples in $\mathbf{x}$



where,

$$\hat{\mathbf{x}} \leftarrow \mathcal{F}\{\mathbf{x}\}$$

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

```
>> xf = fft2(x);
```

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

```
>> xf = fft2(x);
>> yf = fft2(y);
```

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

```
>> xf = fft2(x);
>> yf = fft2(y);
>> sxx = xf.*conj(xf);
```

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

```
>> xf = fft2(x);
>> yf = fft2(y);
>> sxx = xf.*conj(xf);
>> sxy = xf.*conj(yf);
```

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

```
>> xf = fft2(x);
>> yf = fft2(y);
>> sxx = xf.*conj(xf);
>> sxy = xf.*conj(yf);
>> hf = sxy./(sxx + 1e-3);
```

Algorithm	Frame Rate	CPU
FragTrack[1]	realtime	Unknown
GBDL[19]	realtime	3.4 Ghz Pent. 4
IVT [17]	7.5fps	2.8Ghz CPU
MILTrack[2]	25 fps	Core 2 Quad
MOSSE Filters	669fps	2.4Ghz Core 2 Duo

Algorithm	Frame Rate	CPU
FragTrack[1]	realtime	Unknown
GBDL[19]	realtime	3.4 Ghz Pent. 4
IVT [17]	7.5fps	2.8Ghz CPU
MILTrack[2]	25 fps	Core 2 Quad
MOSSE Filters	669fps	2.4Ghz Core 2 Duo

$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

$$\hat{\mathbf{s}}_{xx} = \sum_{i=1}^{N} \hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{x}}_i) \quad \& \quad \hat{\mathbf{s}}_{xy} = \sum_{i=1}^{N} \hat{\mathbf{y}}_i \circ \text{conj}(\hat{\mathbf{x}}_i)$$

N = number of training images

memory efficiency $\leftarrow \mathcal{O}(D)$

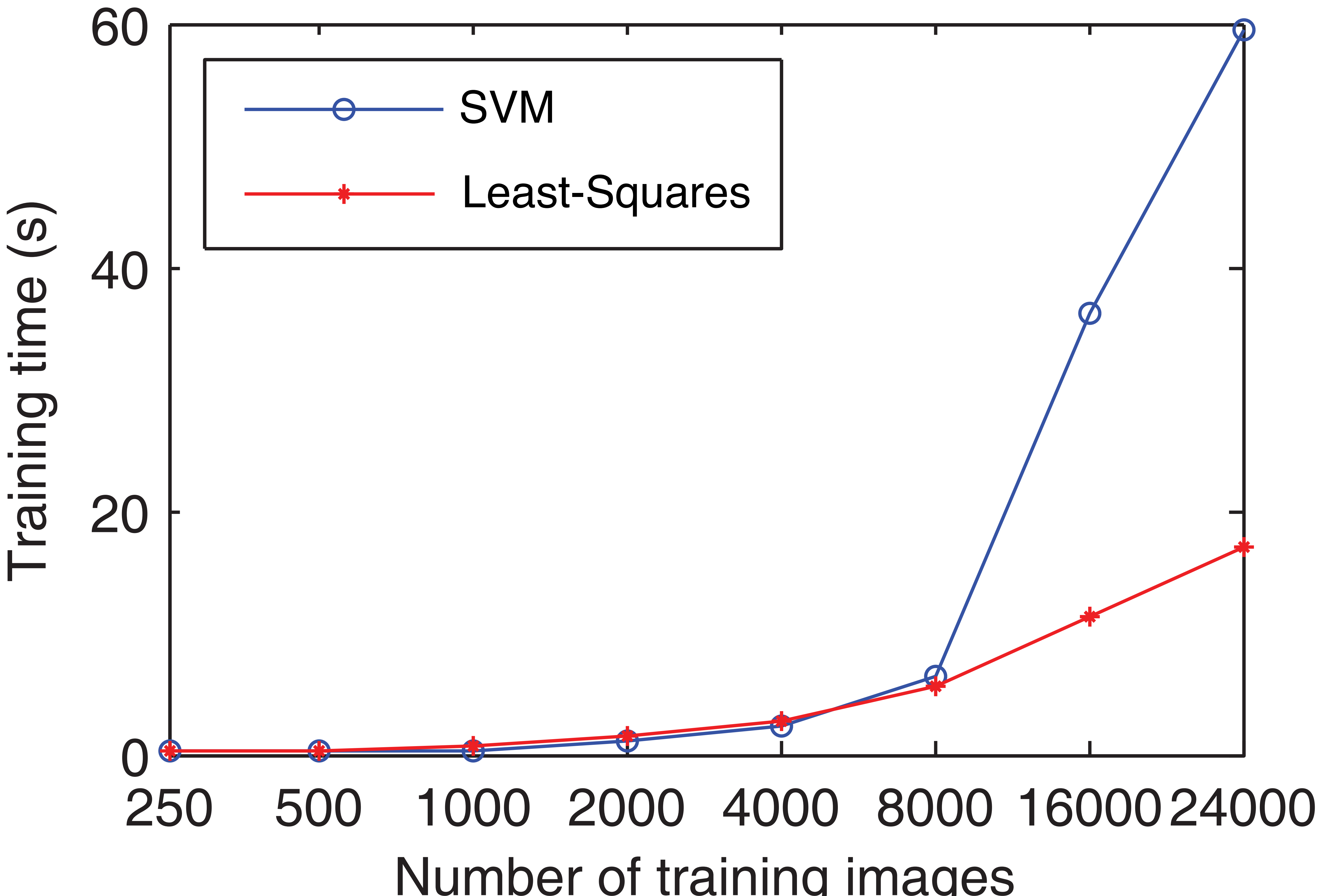
$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

$$\hat{\mathbf{s}}_{xx} = \sum_{i=1}^{N} \hat{\mathbf{x}}_i \circ \mathbf{conj}(\hat{\mathbf{x}}_i) \quad \& \quad \hat{\mathbf{s}}_{xy} = \sum_{i=1}^{N} \hat{\mathbf{y}}_i \circ \mathbf{conj}(\hat{\mathbf{x}}_i)$$

N = number of training images

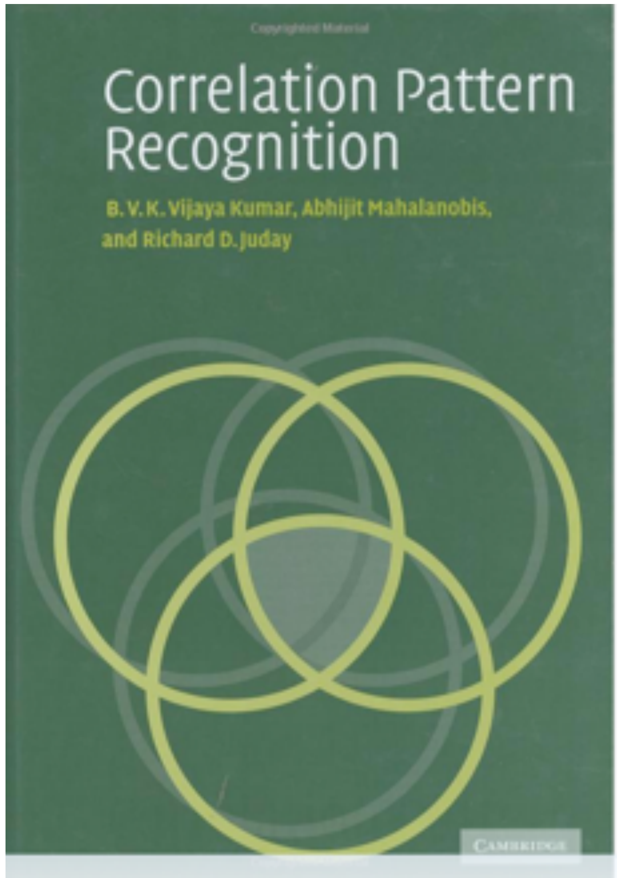
memory efficiency $\leftarrow \mathcal{O}(D)$

SVM memory efficiency $\leftarrow \mathcal{O}(ND)$



Galoogahi, Sim & Lucey "Multi-Channel Correlation Filters", ICCV 2013.

More to read…



- Vijaya Kumar, Mahalanobis, & Juday "Correlation Pattern Recognition", 2010.
- Bolme, Beveridge, Draper & Lui, "Visual Object Tracking using Adaptive Correlation Filters", CVPR 2010.
- Galoogahi, Sim & Lucey "Multi-Channel Correlation Filters", ICCV 2013.