

# Correlation Filters

Instructor - Simon Lucey

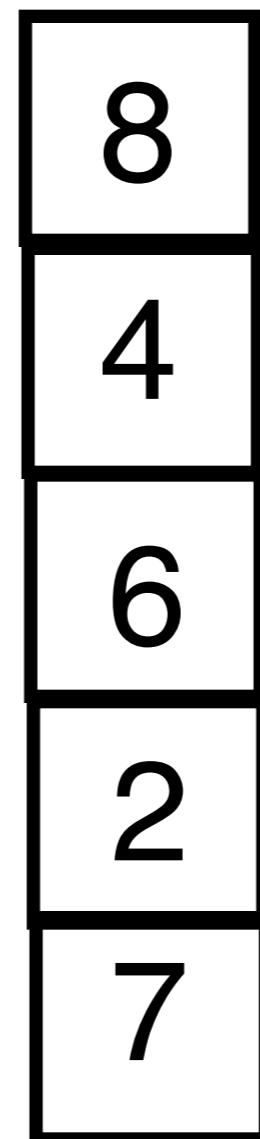
**16-423 - Designing Computer Vision Apps**

# Today

---

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

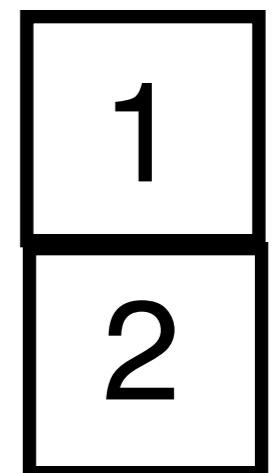
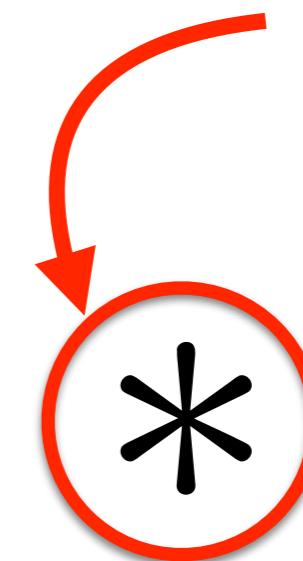
# Convolution



$x$

“signal”

“convolution  
operator”

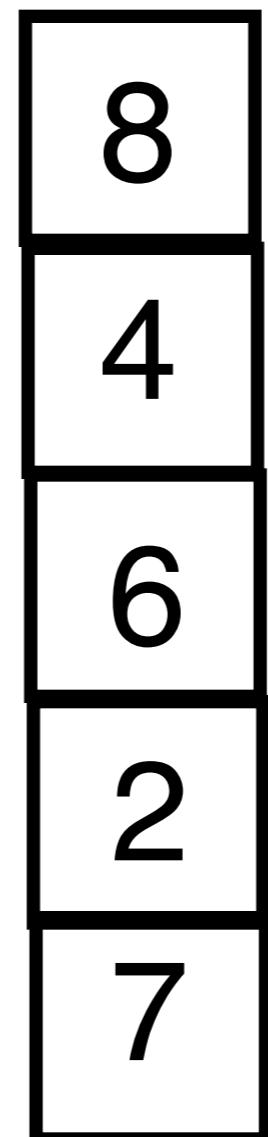


$h$

“filter”

# Convolution

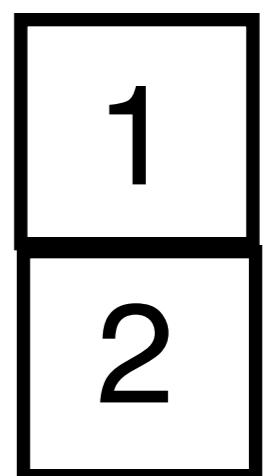
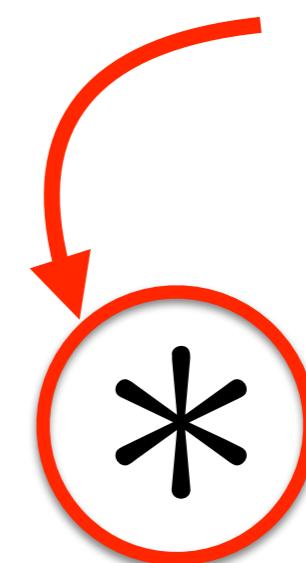
```
>> conv(x,h,'valid')  
ans =  
20  
14  
14  
11
```



**$x$**

“signal”

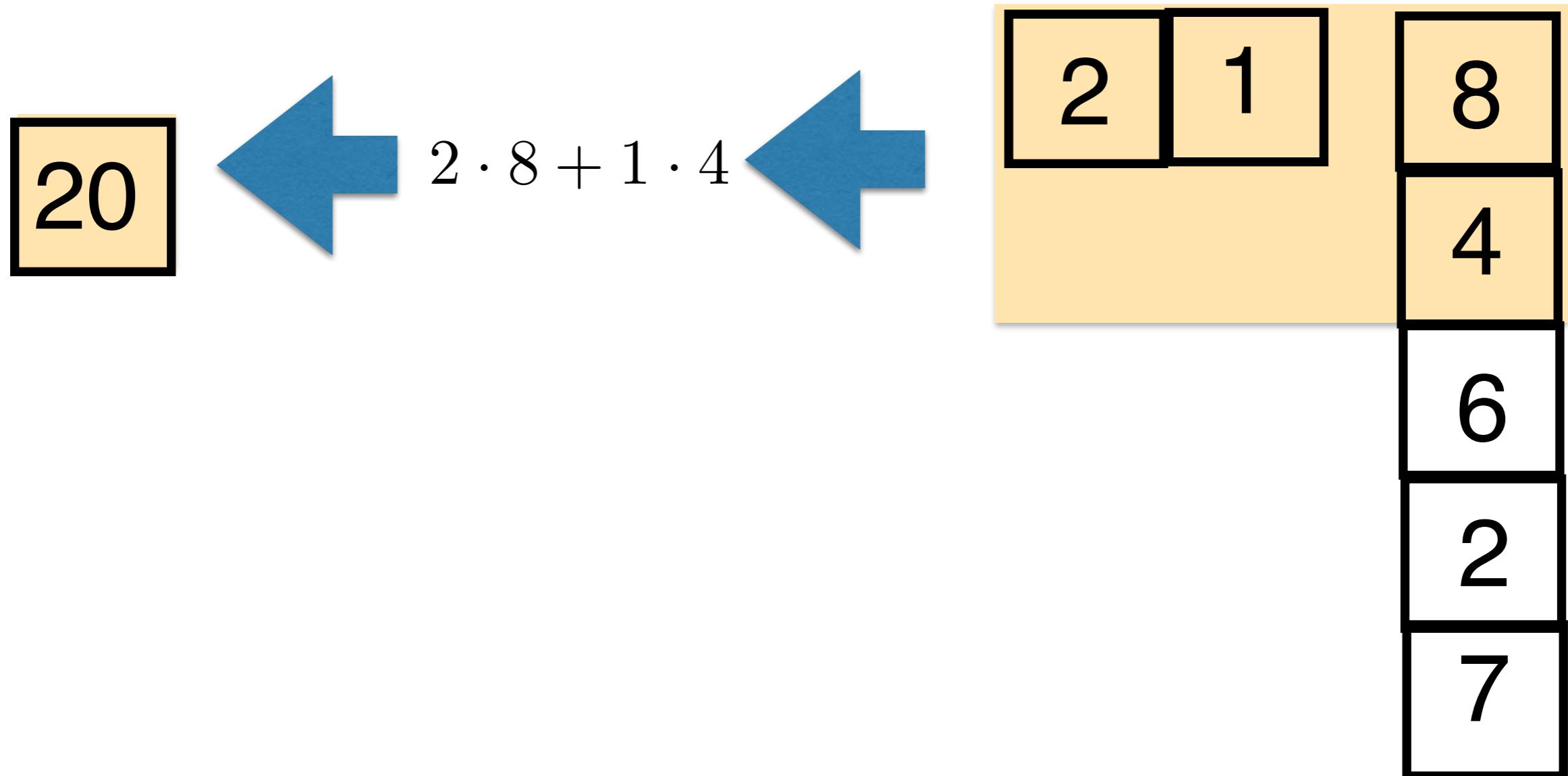
“convolution  
operator”



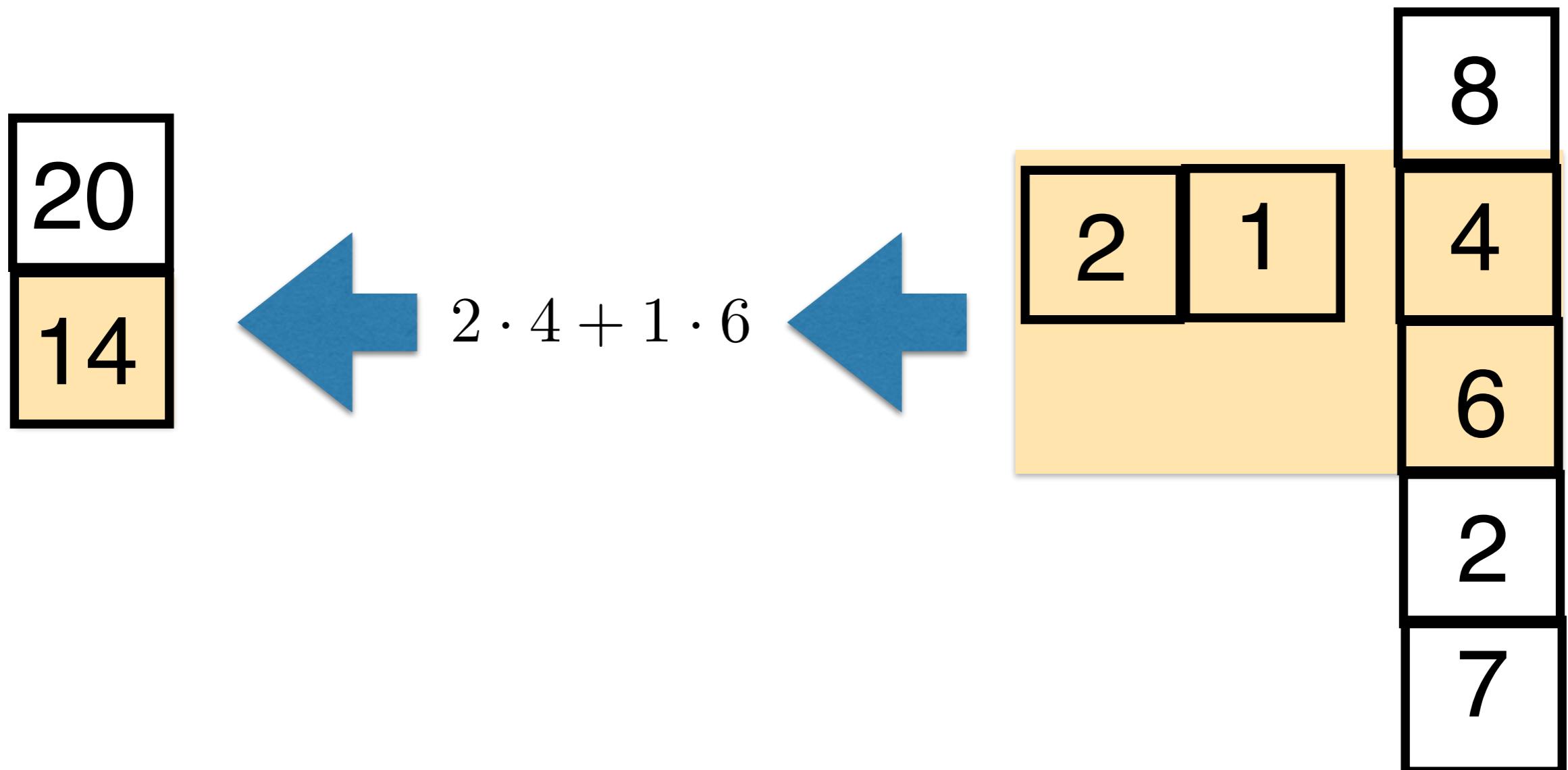
**$h$**

“filter”

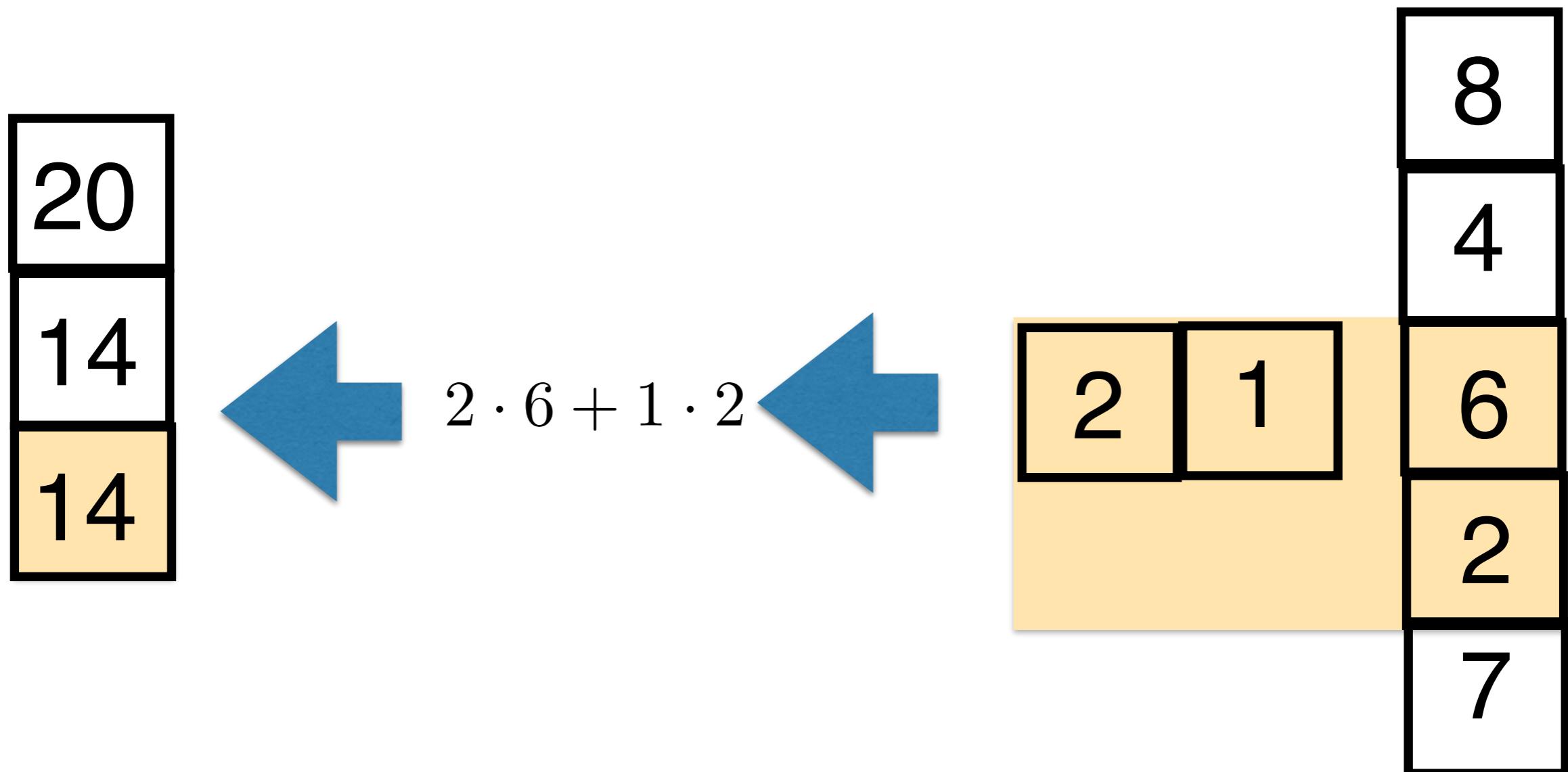
# Convolution



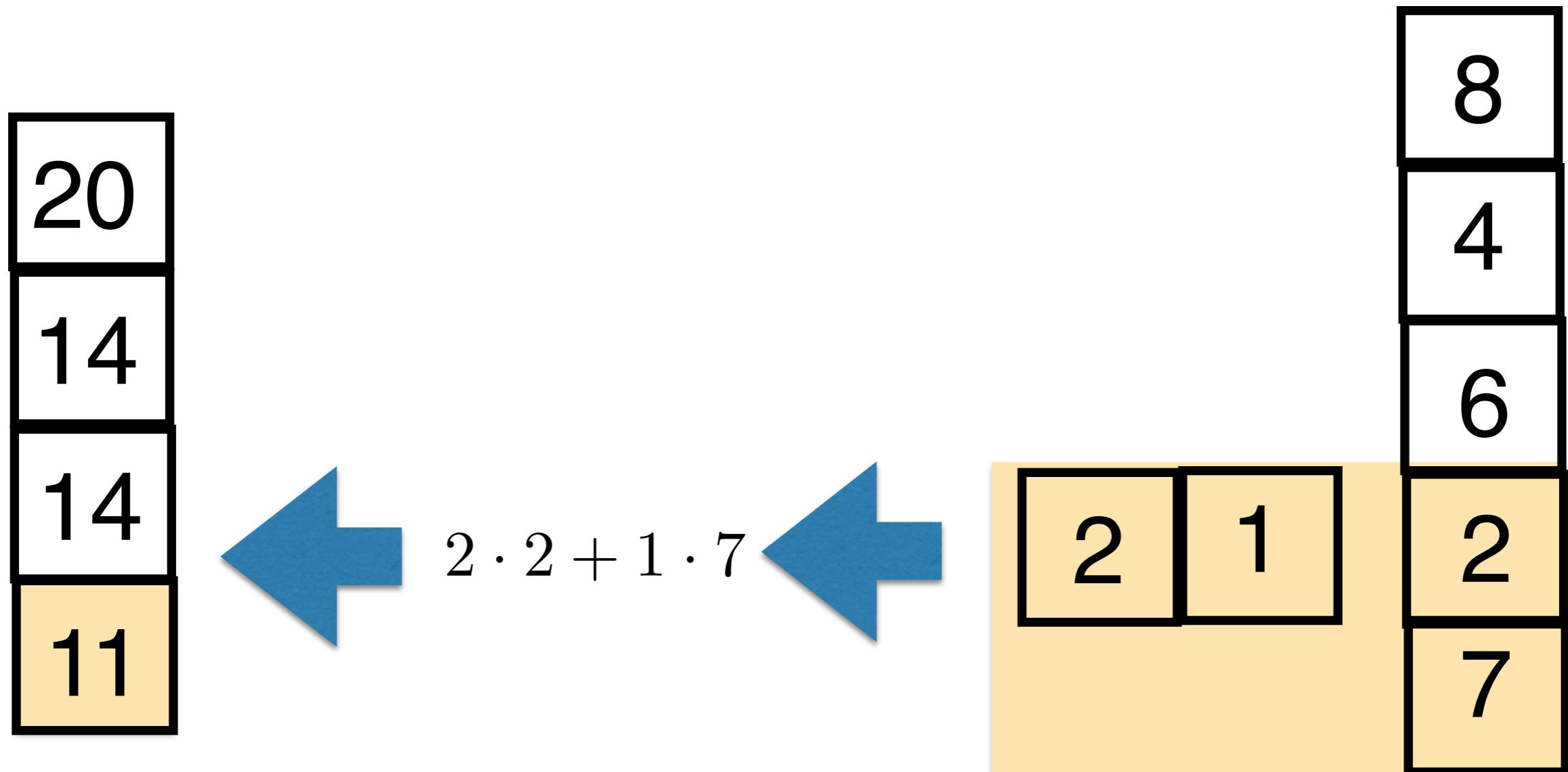
# Convolution



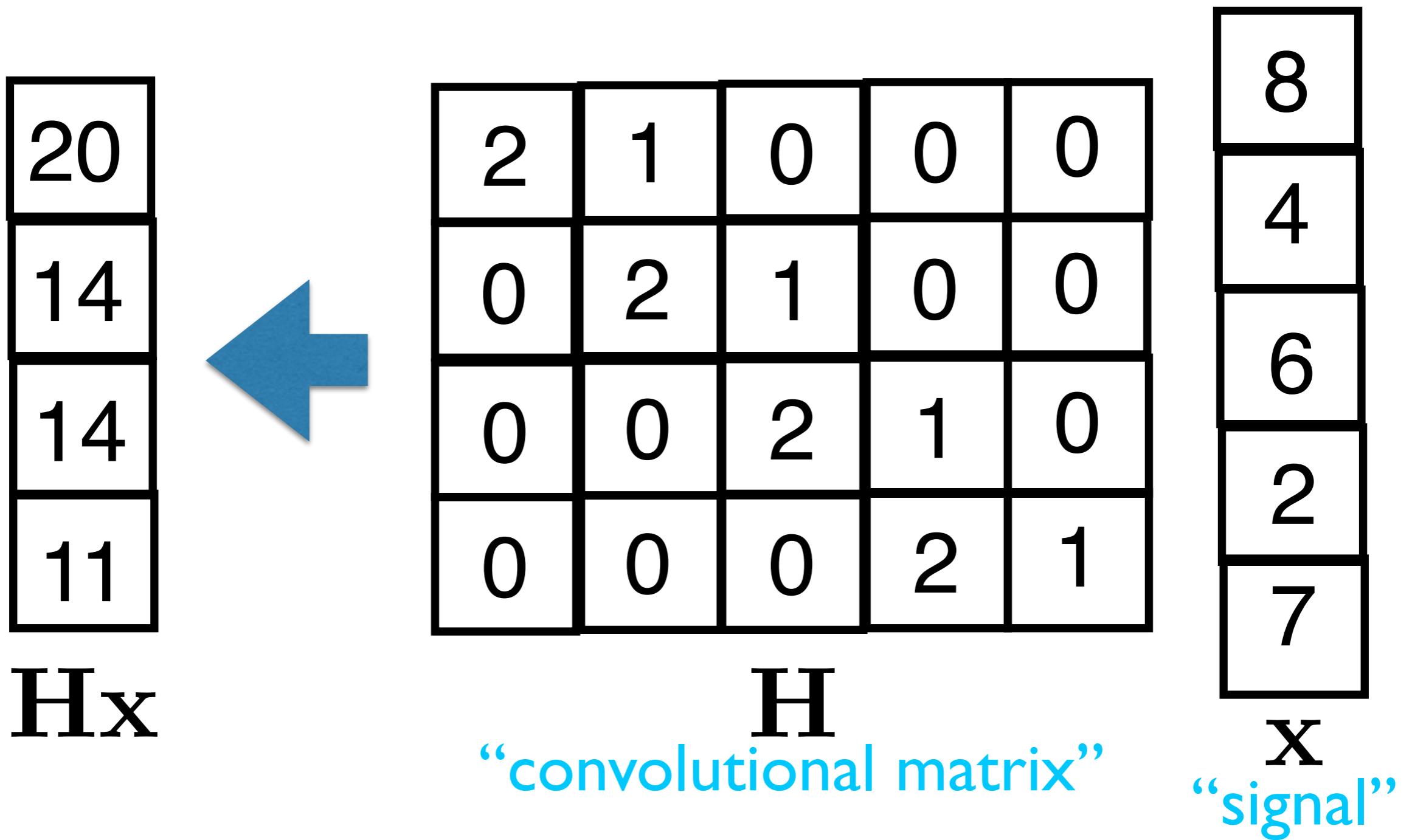
# Convolution



# Convolution



# Convolution



# Types of Convolution

---

- More than just one type of convolutional operator:-

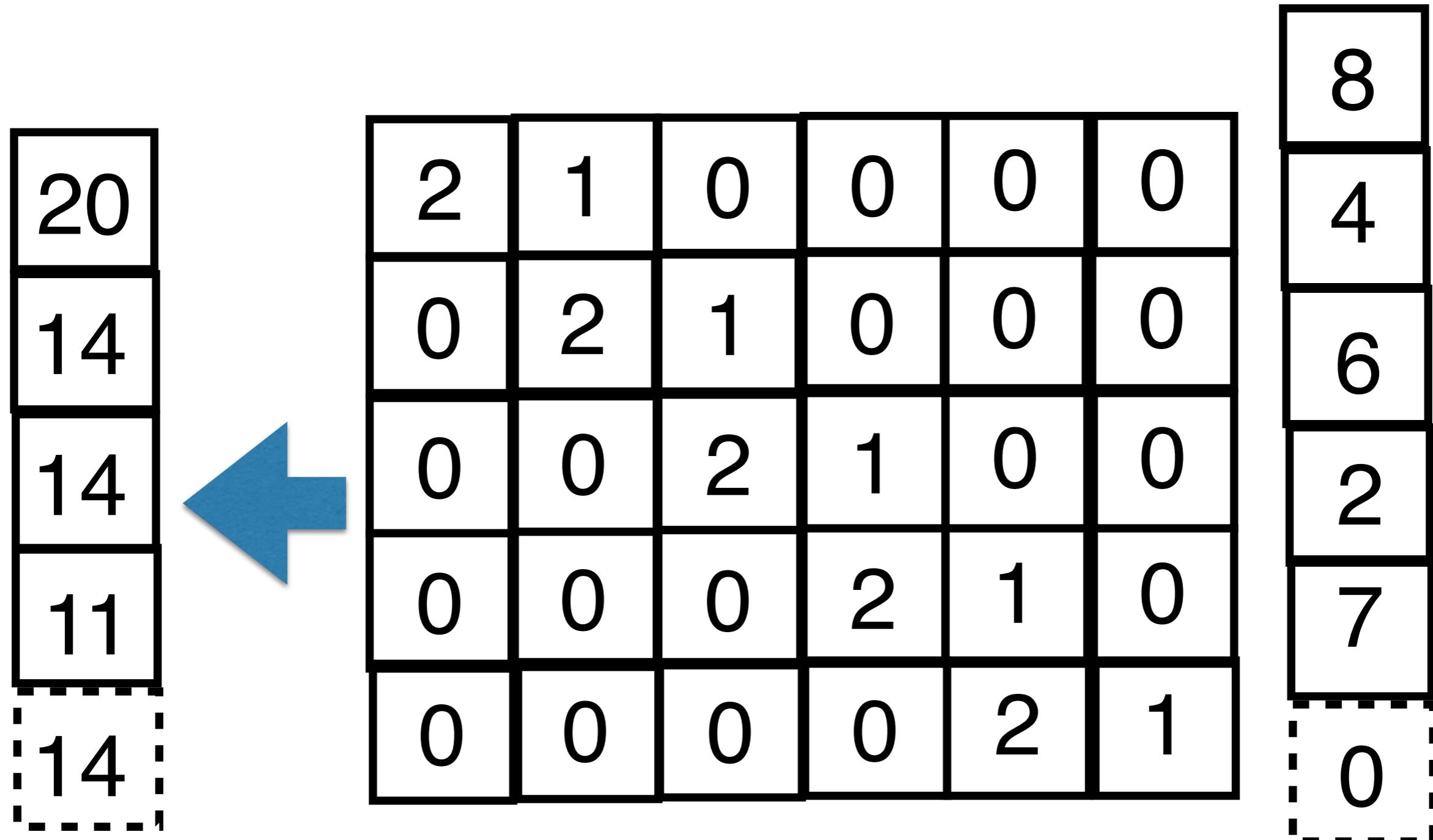
- “Valid” convolution

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

- “Same” convolution

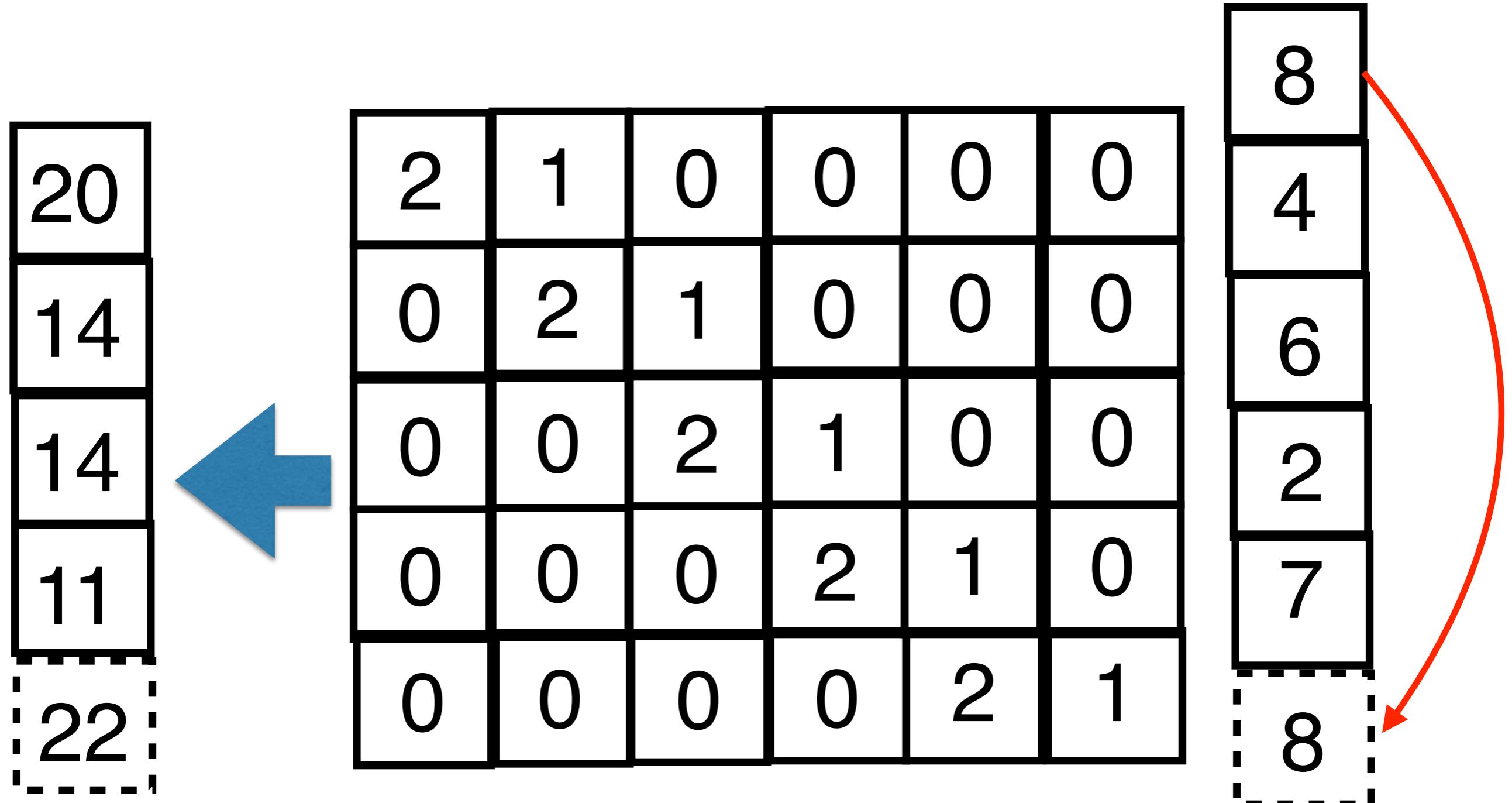
```
>> conv(x,h,'same')
```

# Zero-Padded Convolution

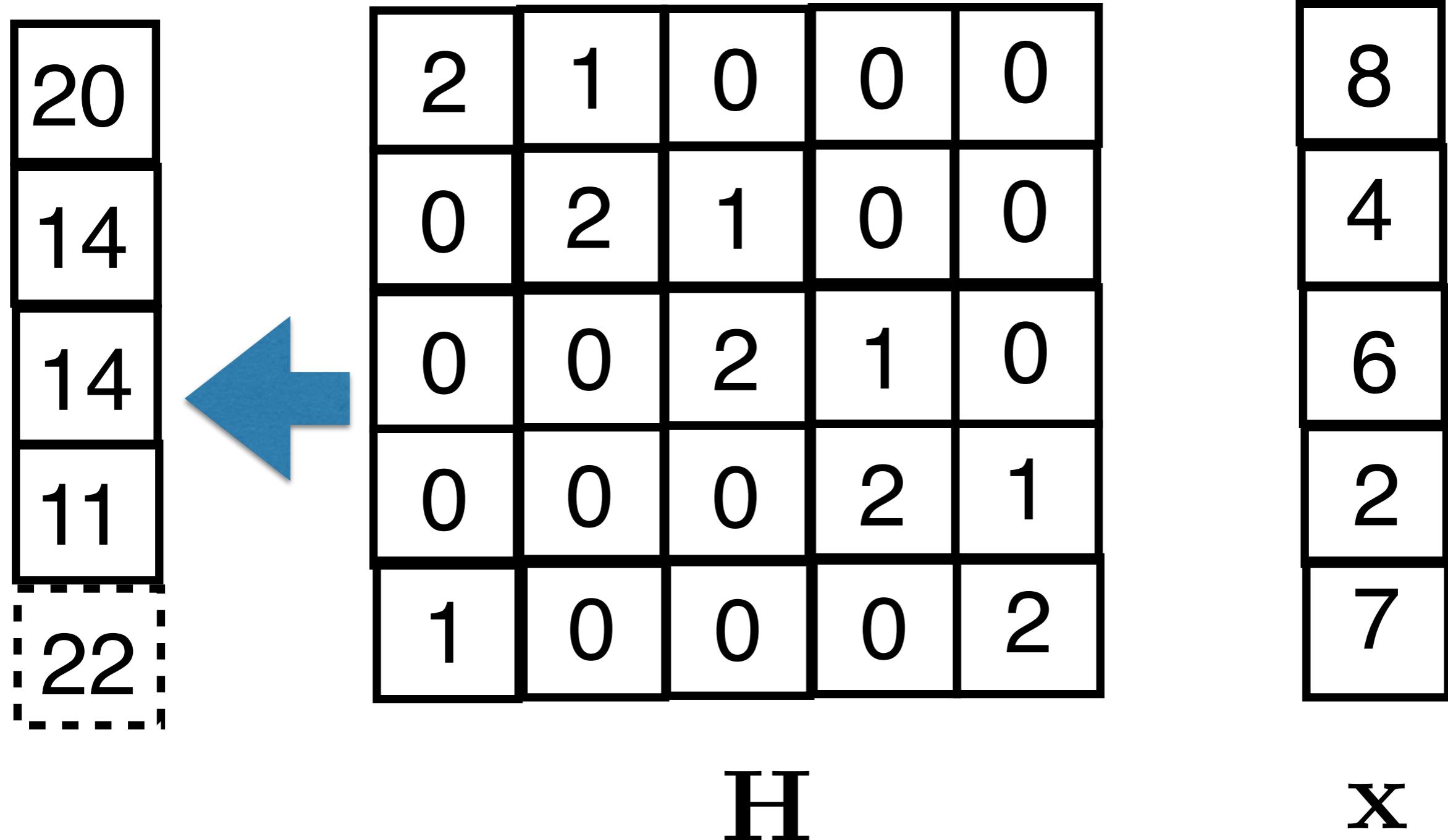


```
>> conv(x, h, 'same')
```

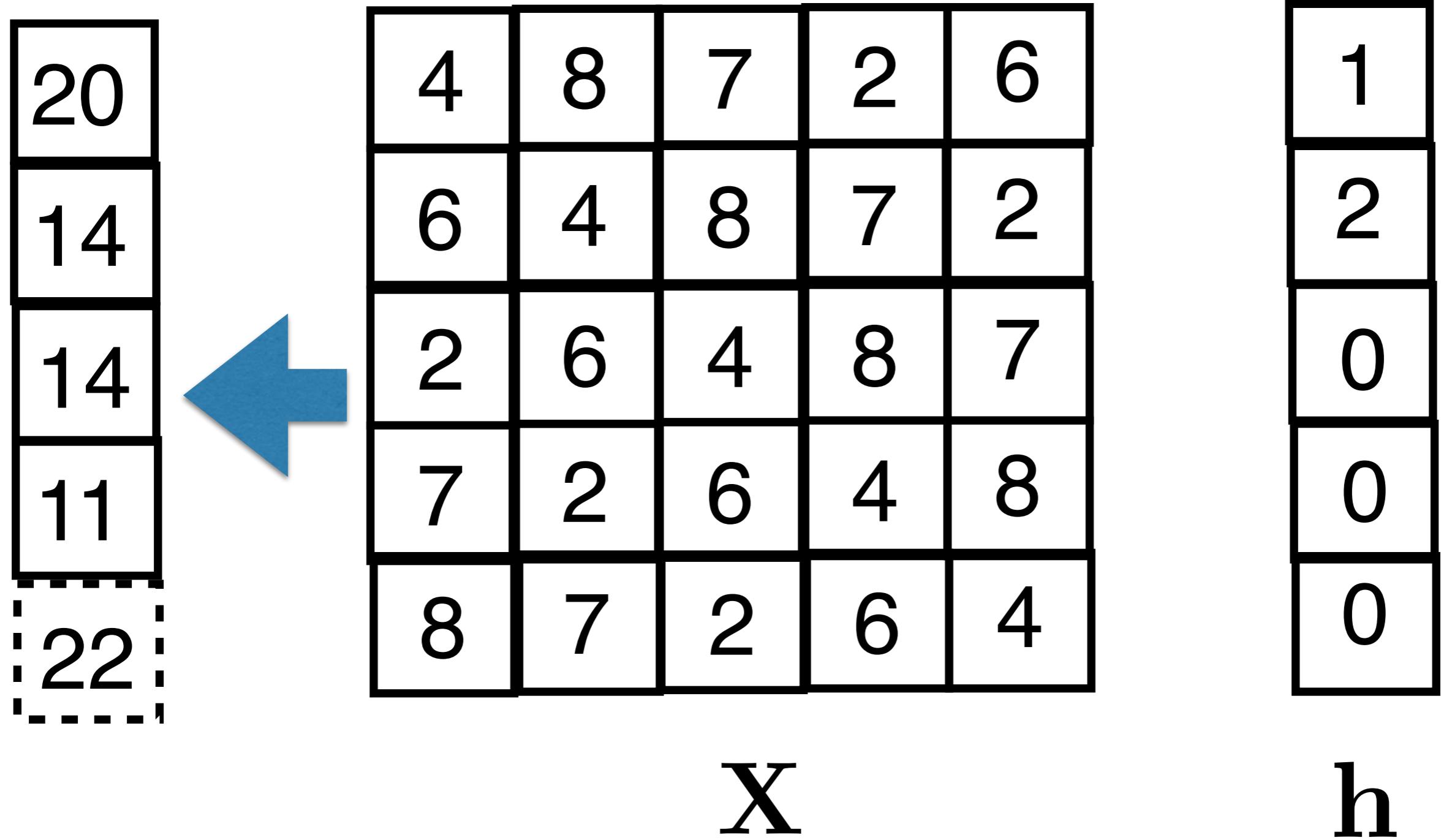
# Circular Convolution



# Circular Convolution

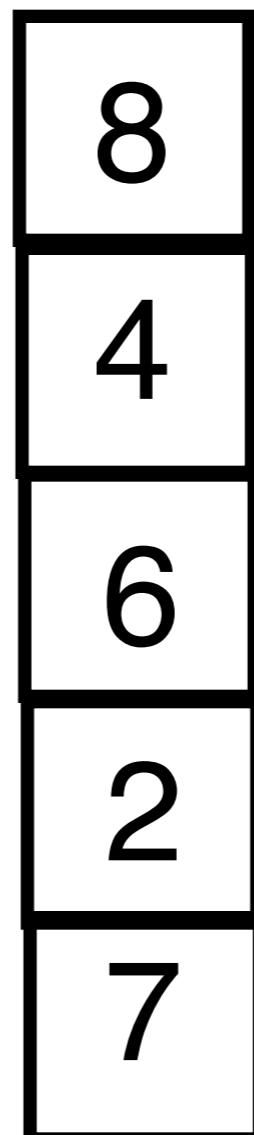


# Circular Convolution



# Correlation

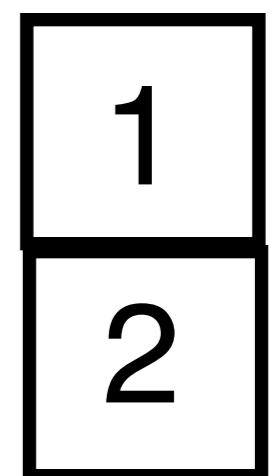
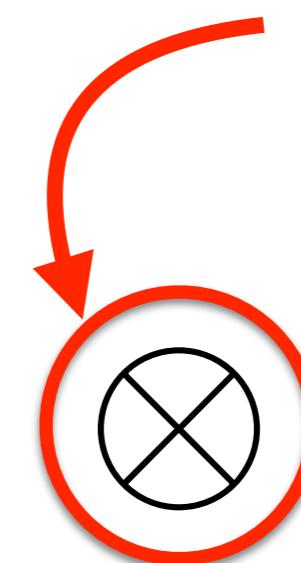
```
>> conv(x, flipud(h),  
... 'same')  
ans =  
  
16  
16  
10  
16  
7
```



**$x$**

“signal”

“correlation  
operator”

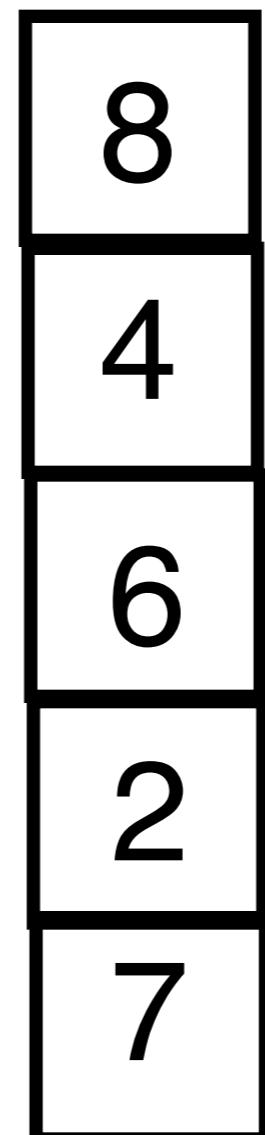


**$h$**

“filter”

# Correlation

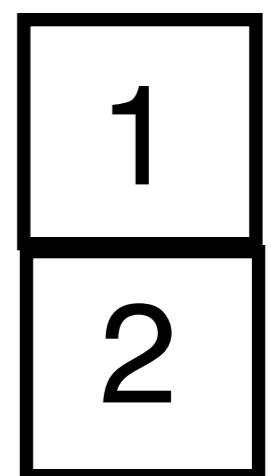
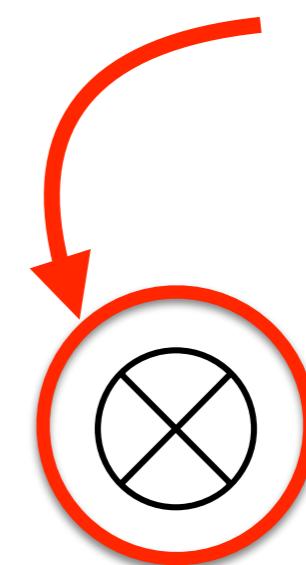
```
>> imfilter(x, h)  
ans =  
16  
16  
16  
10  
16  
7
```



**x**

“signal”

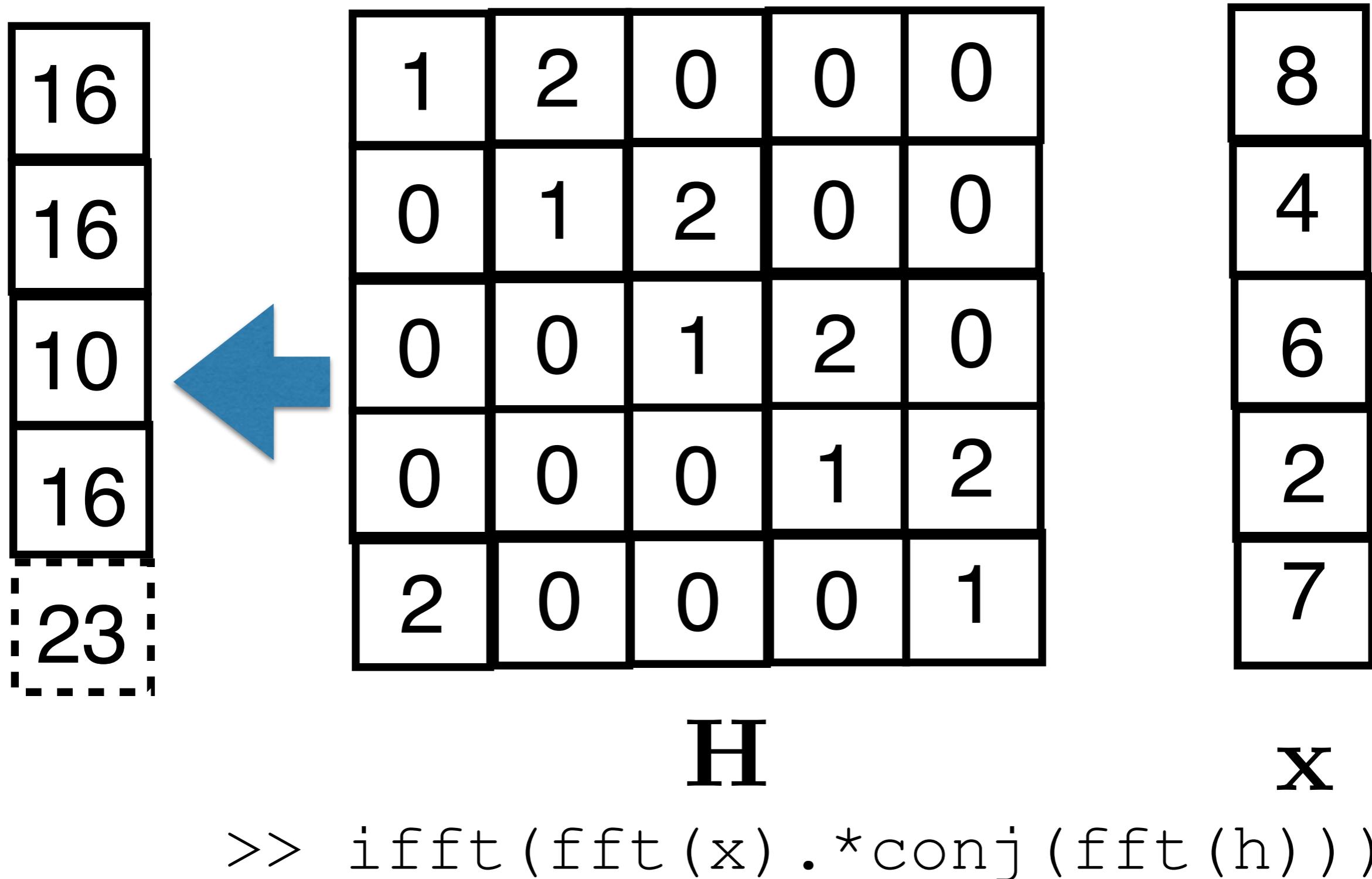
“correlation  
operator”



**h**

“filter”

# Circular Correlation



# Correlation vs. Convolution

---

- Convolution is preferred mathematically as it is associative,

$$(x * h) * h = x * (h * h)$$

- Correlation is not associative,

$$(x \otimes h) \otimes h \neq x \otimes (h \otimes h)$$

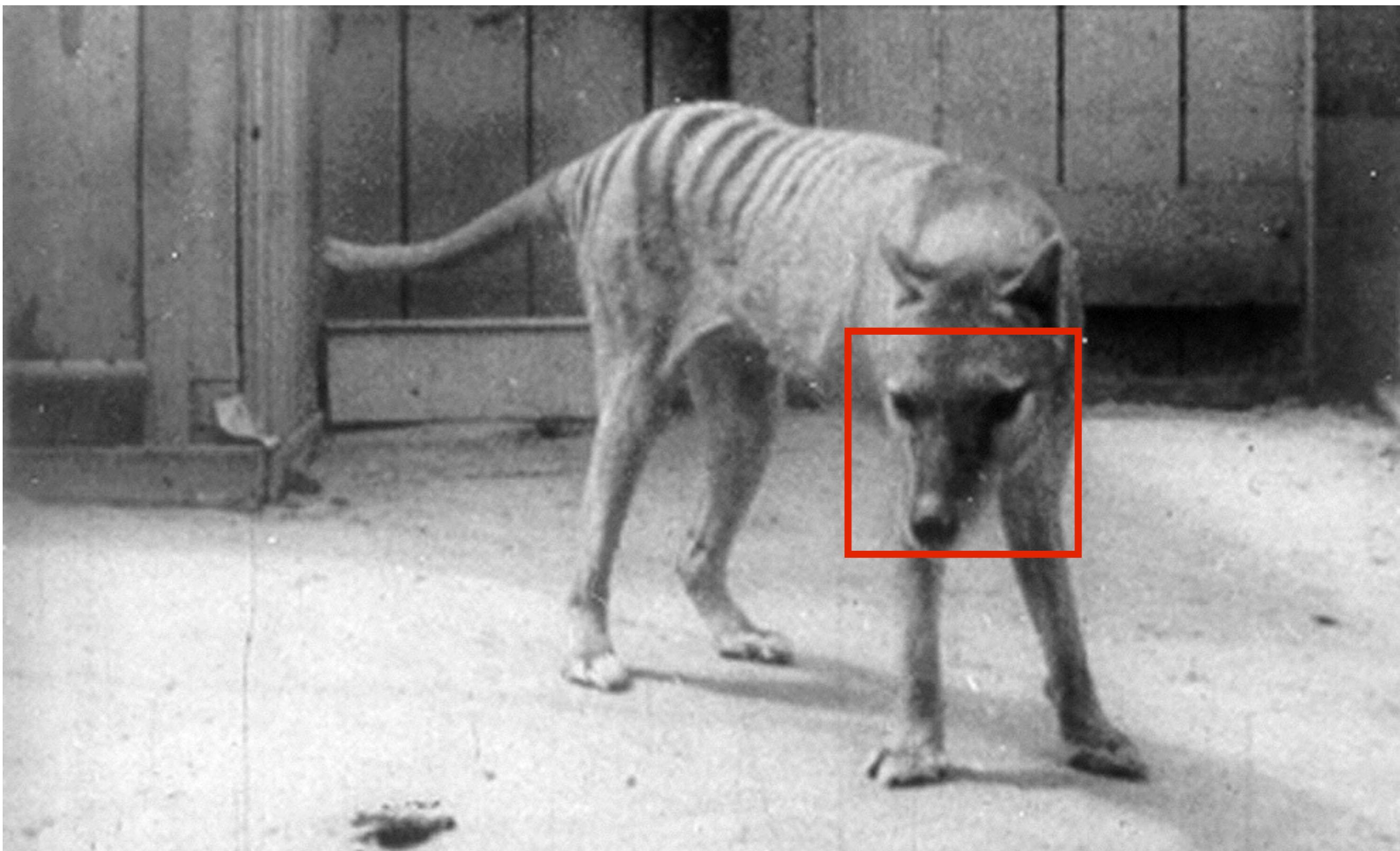
- Correlation preferred, however, for signal matching/detection.

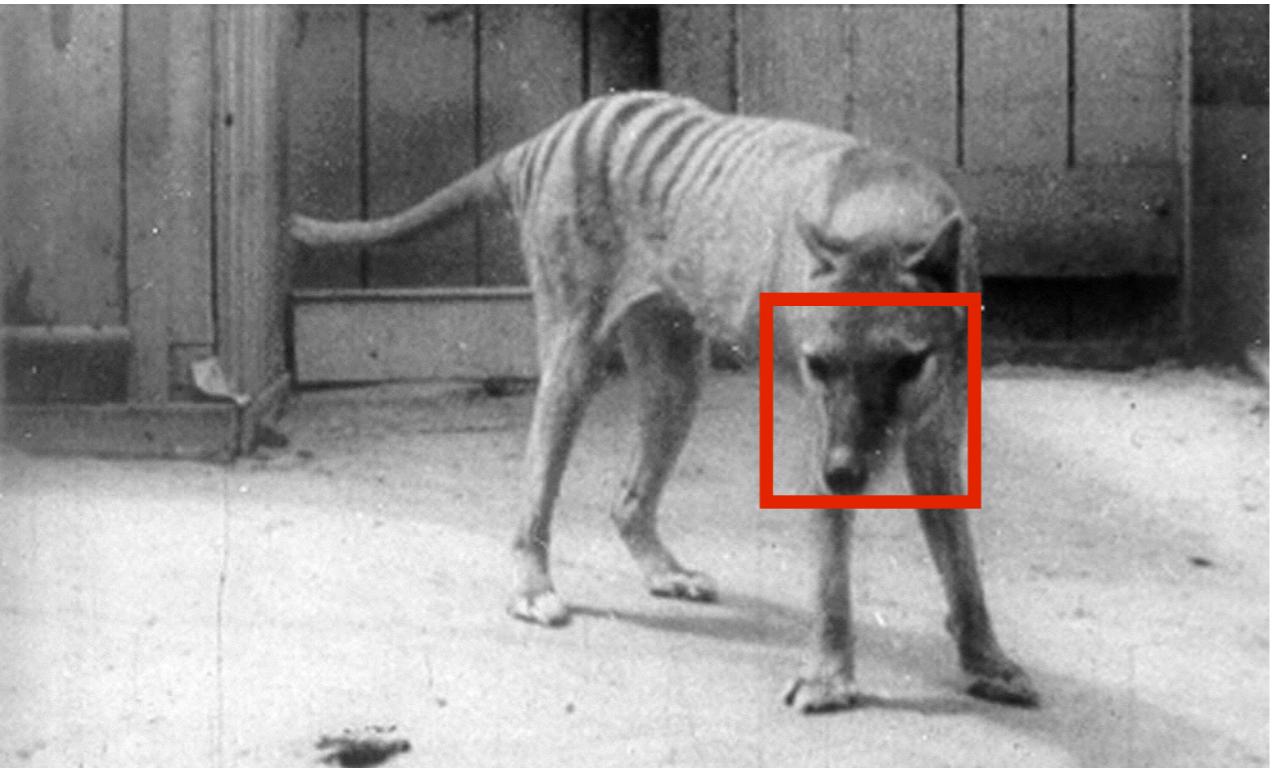
# Today

---

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

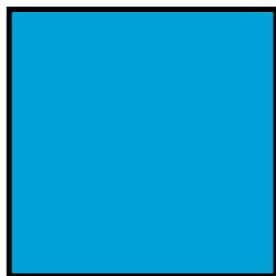






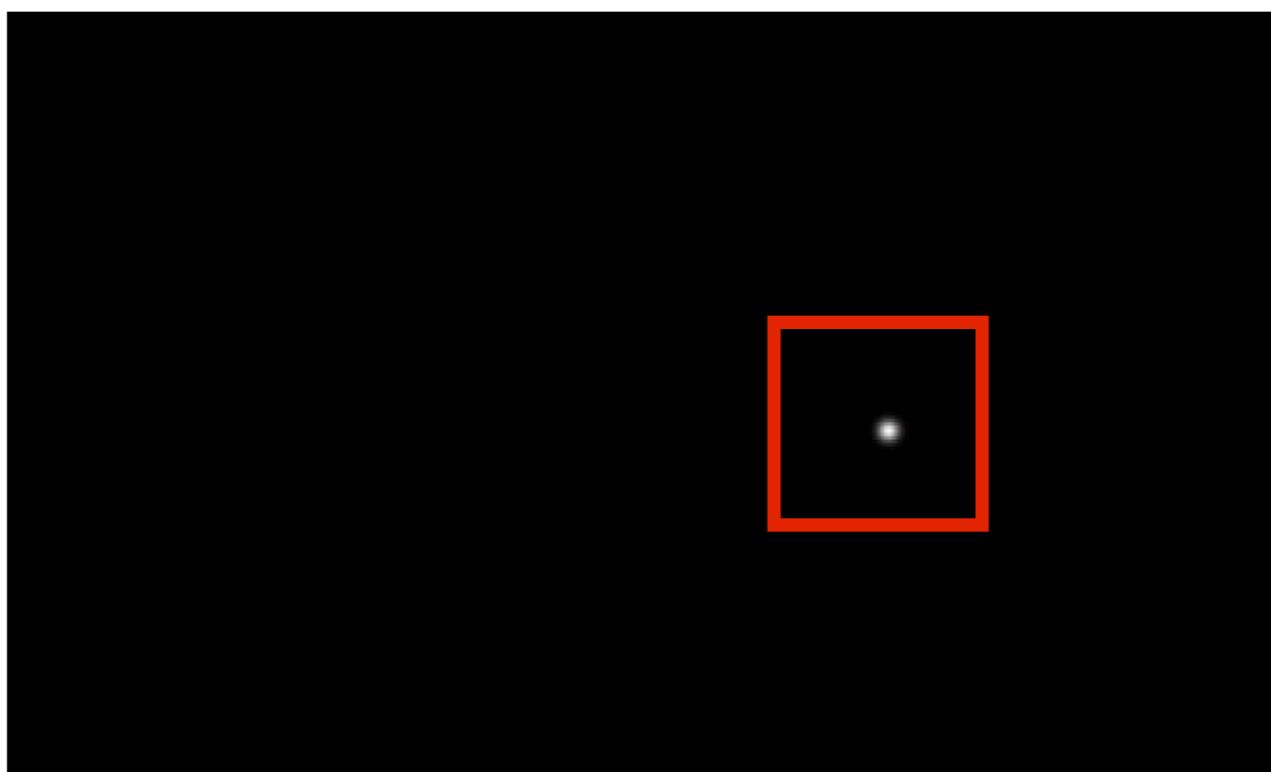
“known signal”  $\mathbf{X}$

$*$



$\mathbf{h}$

“unknown  
filter”



“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]);
```



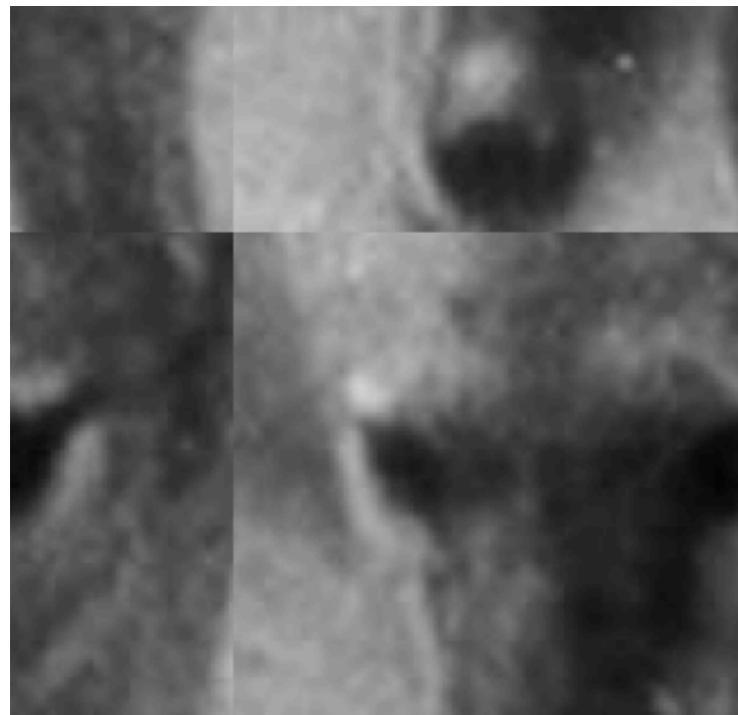
$\mathbf{x}[0, 0]$



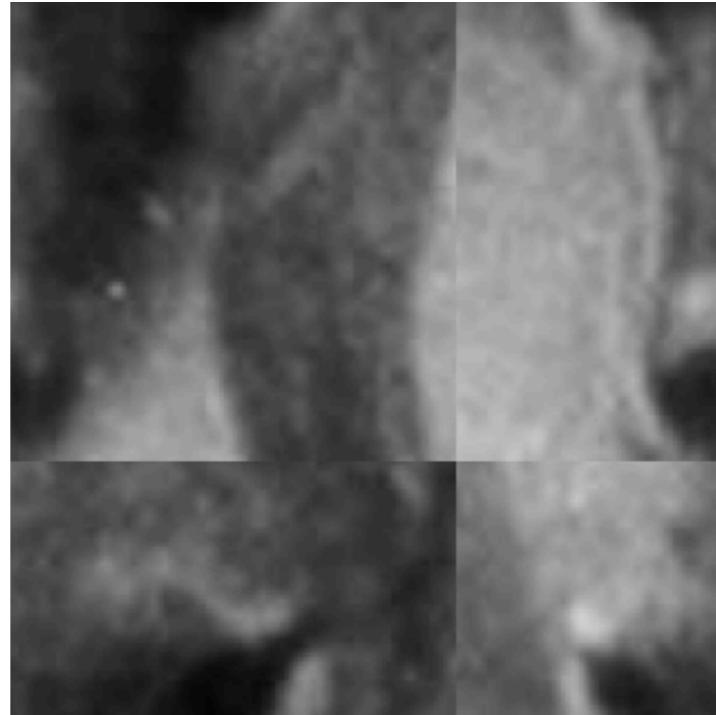
$\mathbf{x}[20, 20]$



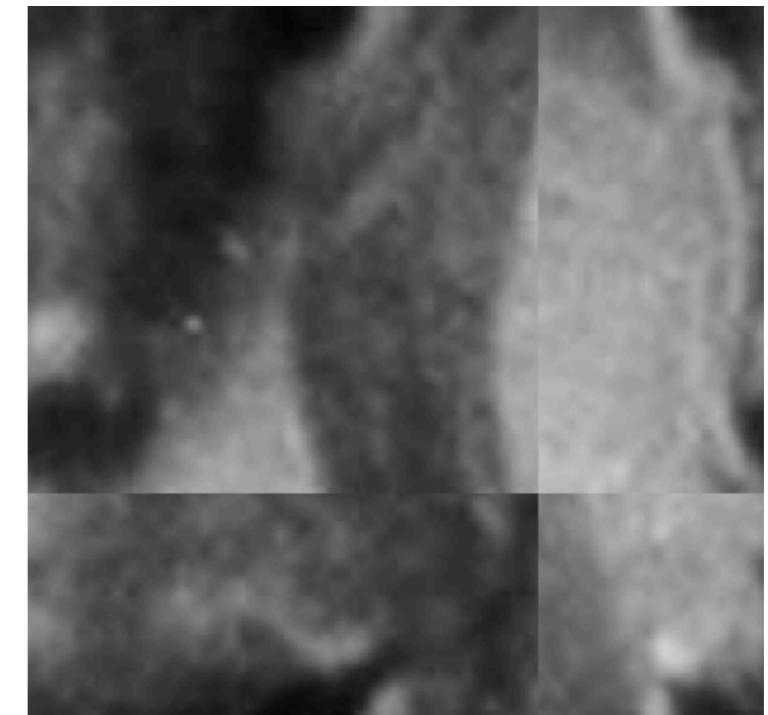
$\mathbf{x}[-20, -20]$



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

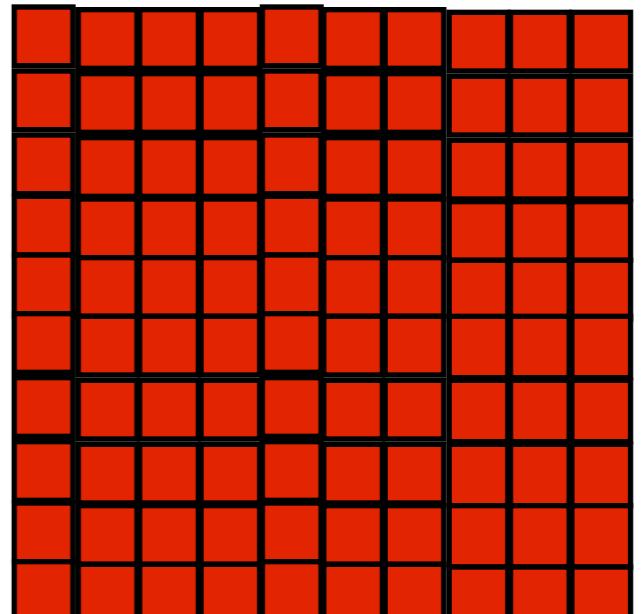


$\mathbf{x}[-100, -100]$



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

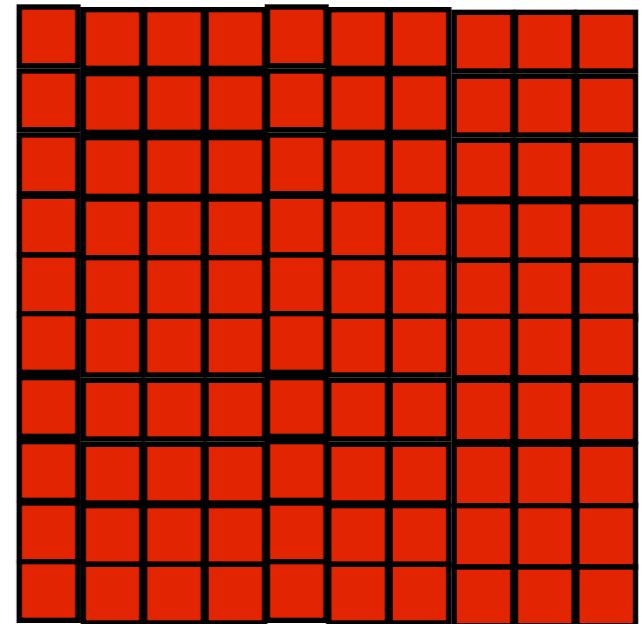
$$\mathbf{S} = \sum_{\tau \in \mathcal{C}} \mathbf{x}[\tau] \mathbf{x}[\tau]^T =$$



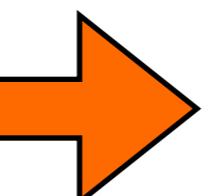
$$(D\times D)$$

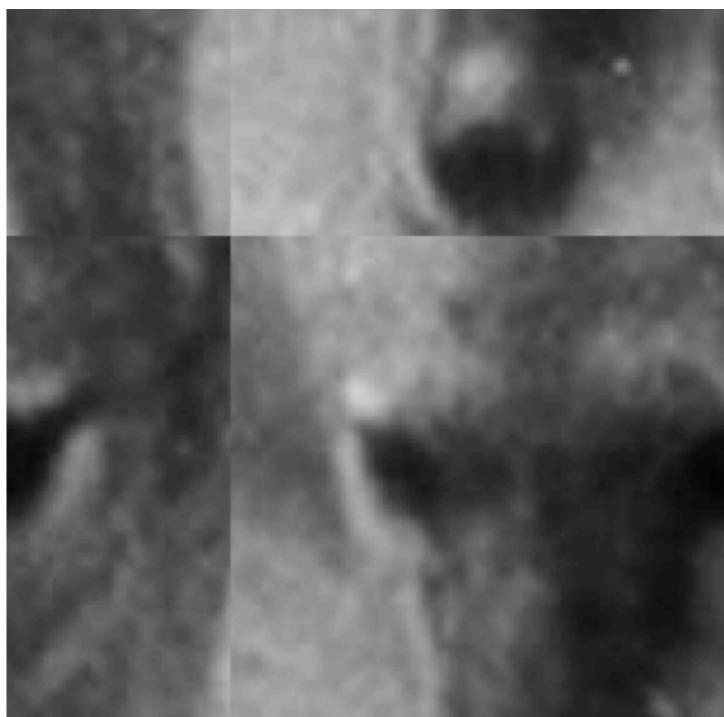
$$S = \sum_{\tau \in C} \mathbf{x}[\tau] \mathbf{x}[\tau]^T =$$

“set of all  
circular shifts”



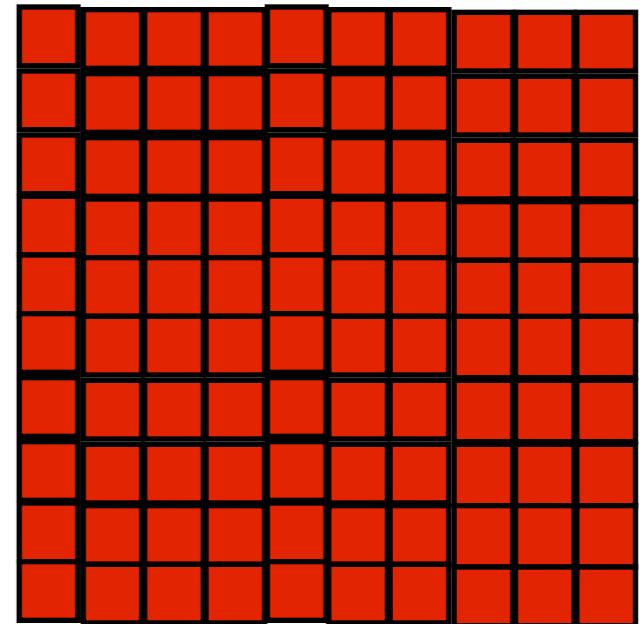
$(D \times D)$

$\mathbf{x}[\tau]$  



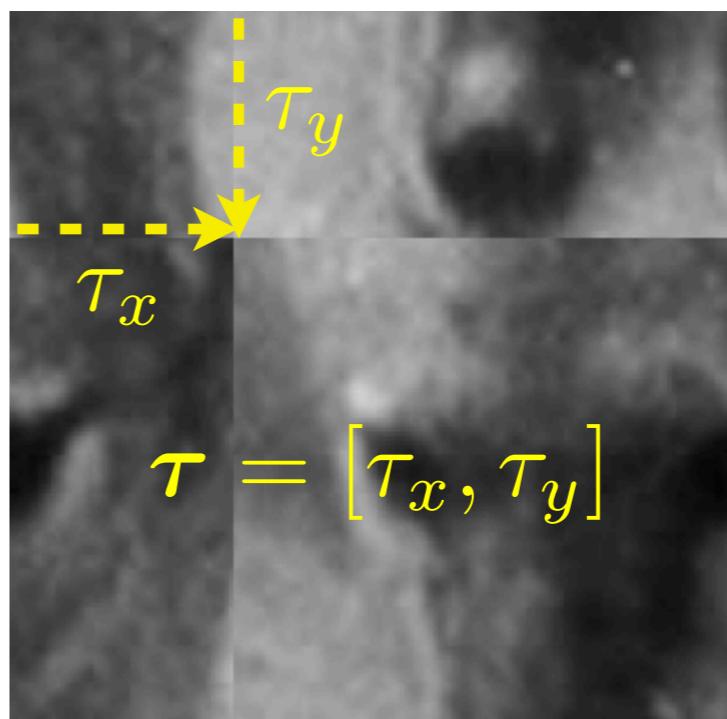
$$S = \sum_{\tau \in C} \mathbf{x}[\tau] \mathbf{x}[\tau]^T =$$

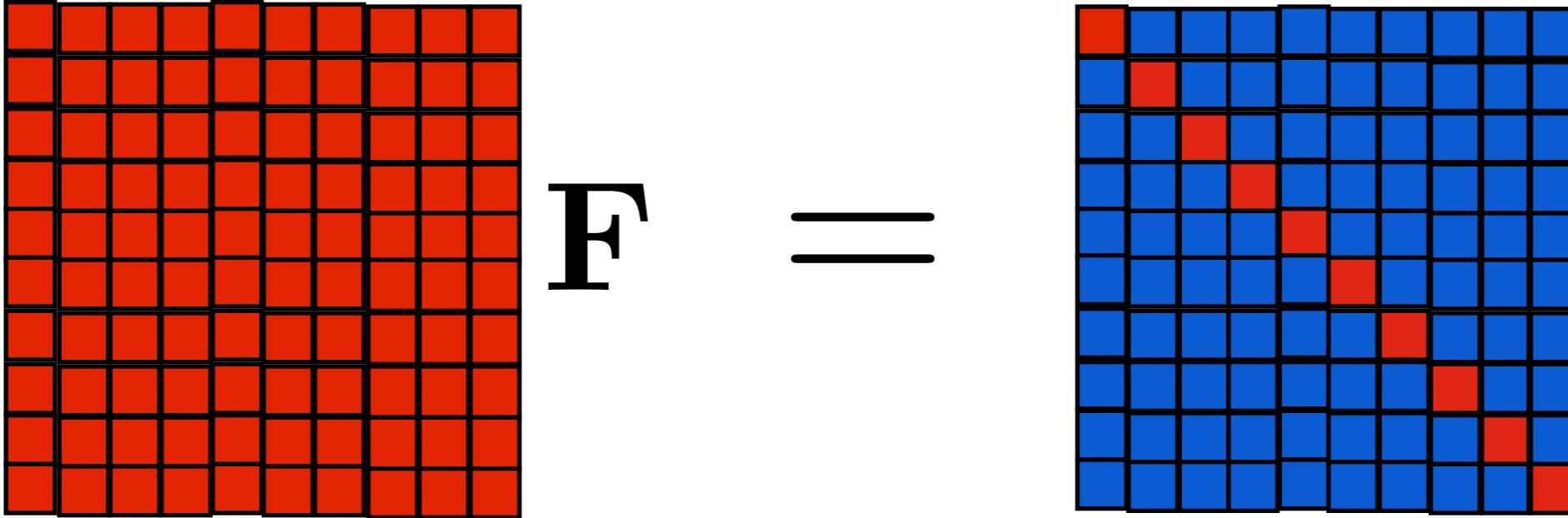
“set of all  
circular shifts”



$$(D \times D)$$

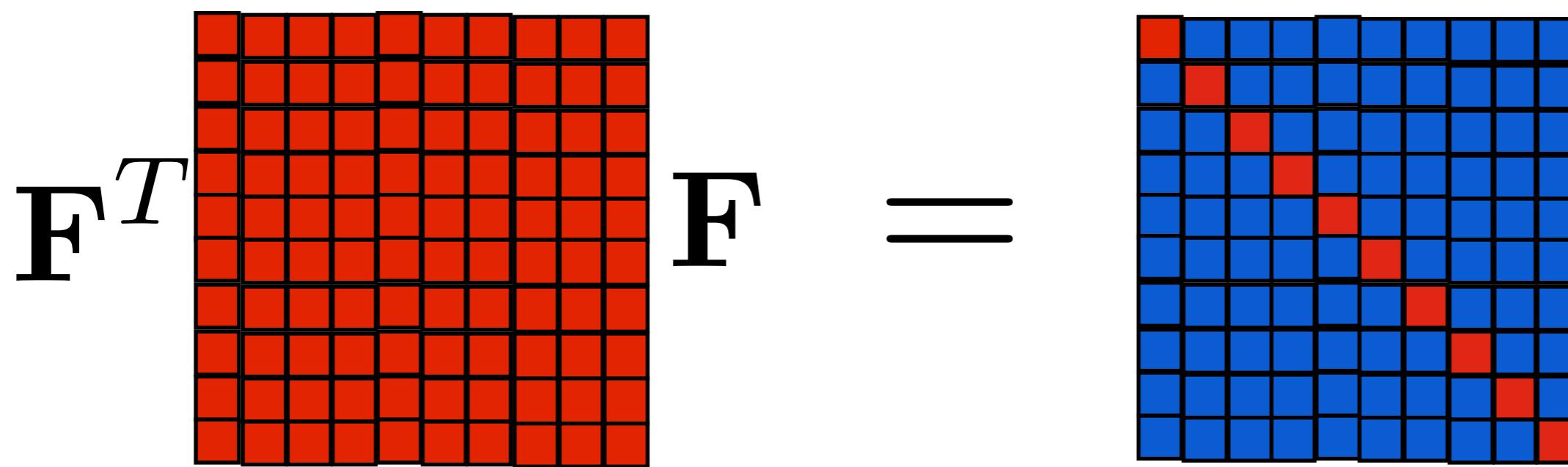
$$\mathbf{x}[\tau] \rightarrow$$



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


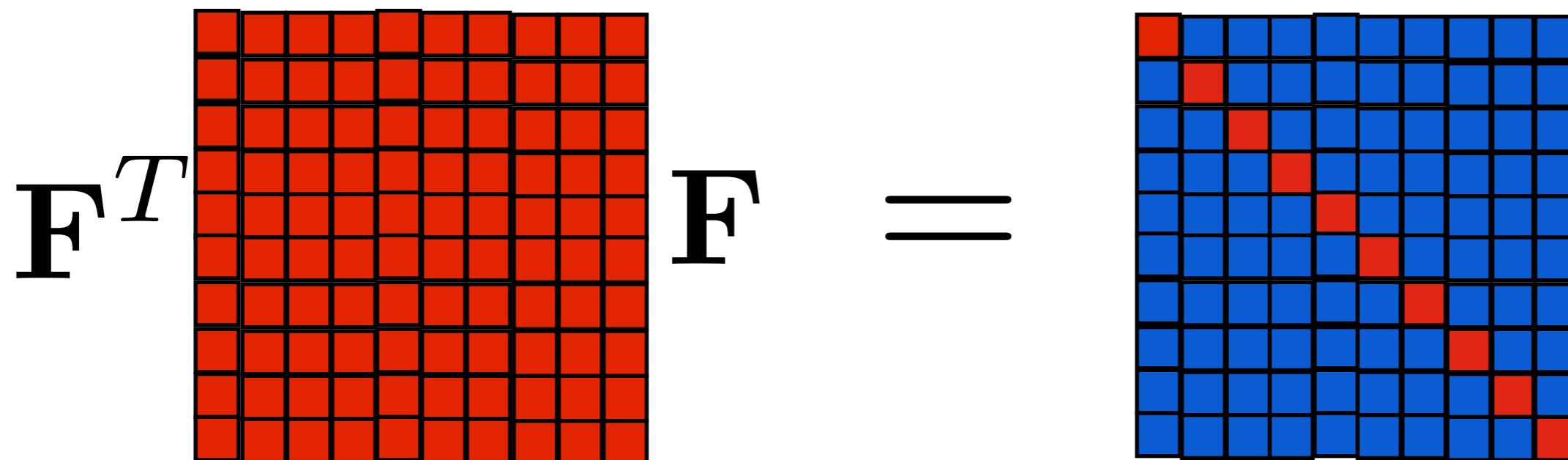
$\mathbf{F} \leftarrow \text{eigenvectors of } \mathbf{S}$

■ Not Always Zero ■ Always Zero



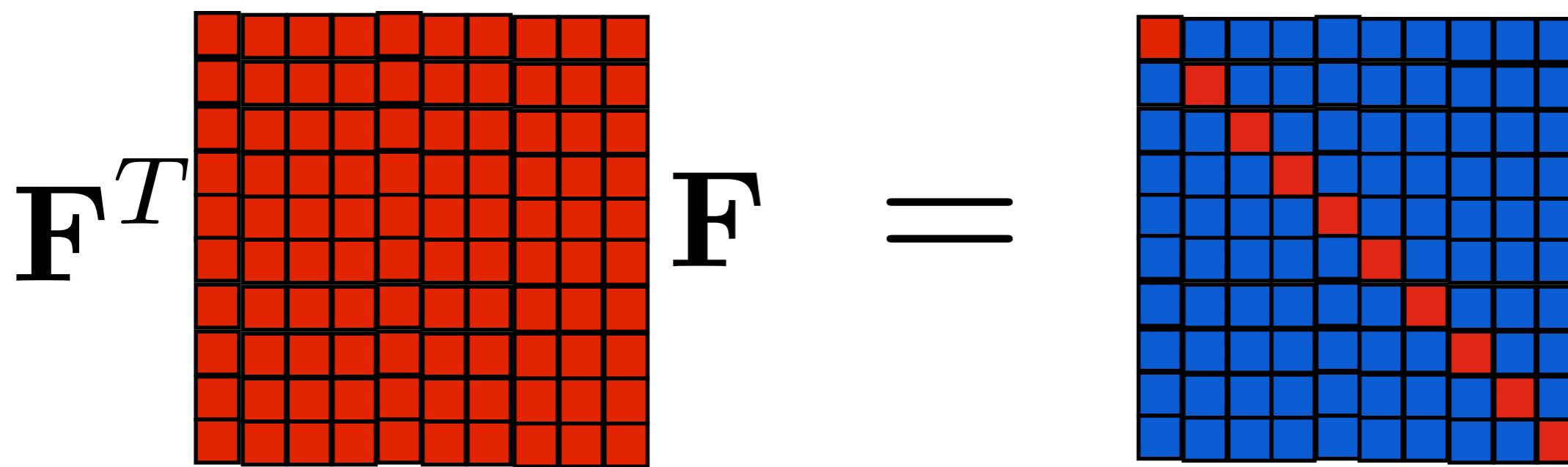
$\mathbf{F} \leftarrow$  Fourier Transform

■ Not Always Zero ■ Always Zero



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

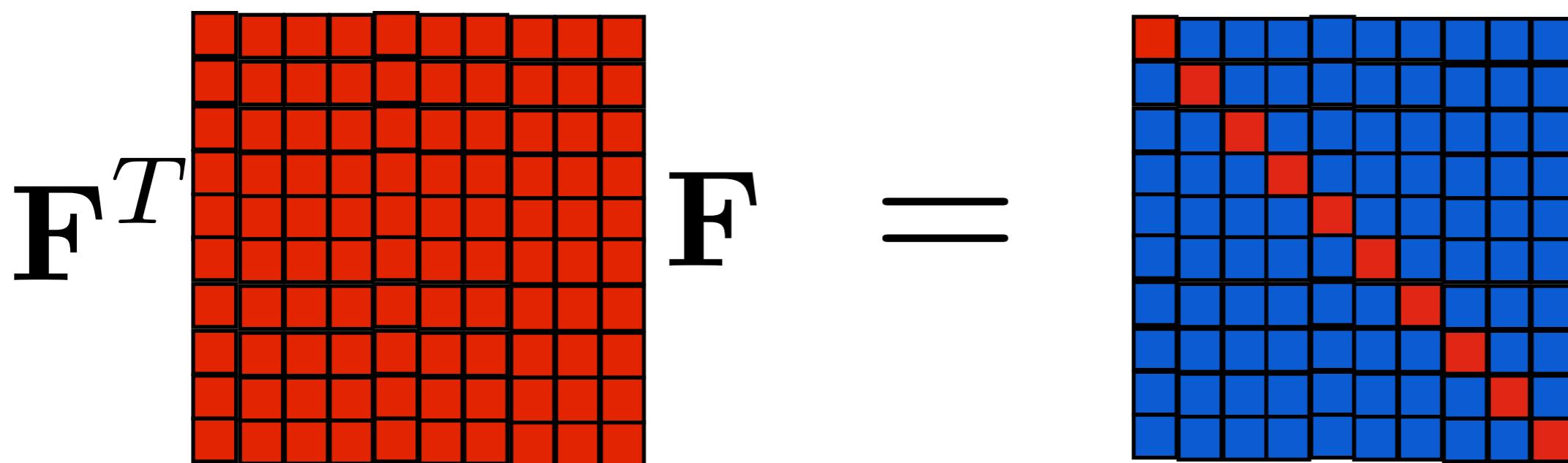
■ Not Always Zero ■ Always Zero



$\mathbf{x}[\tau]$



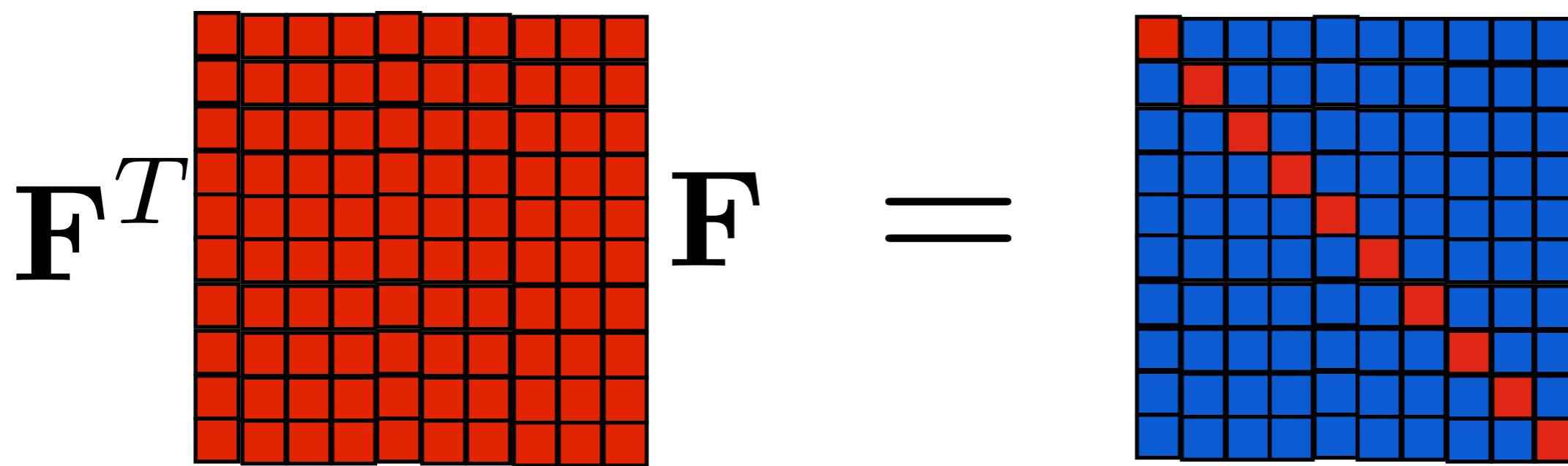
Not Always Zero    Always Zero

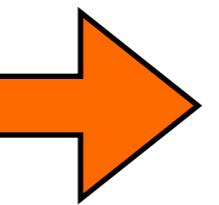


$\mathbf{x}[\tau]$



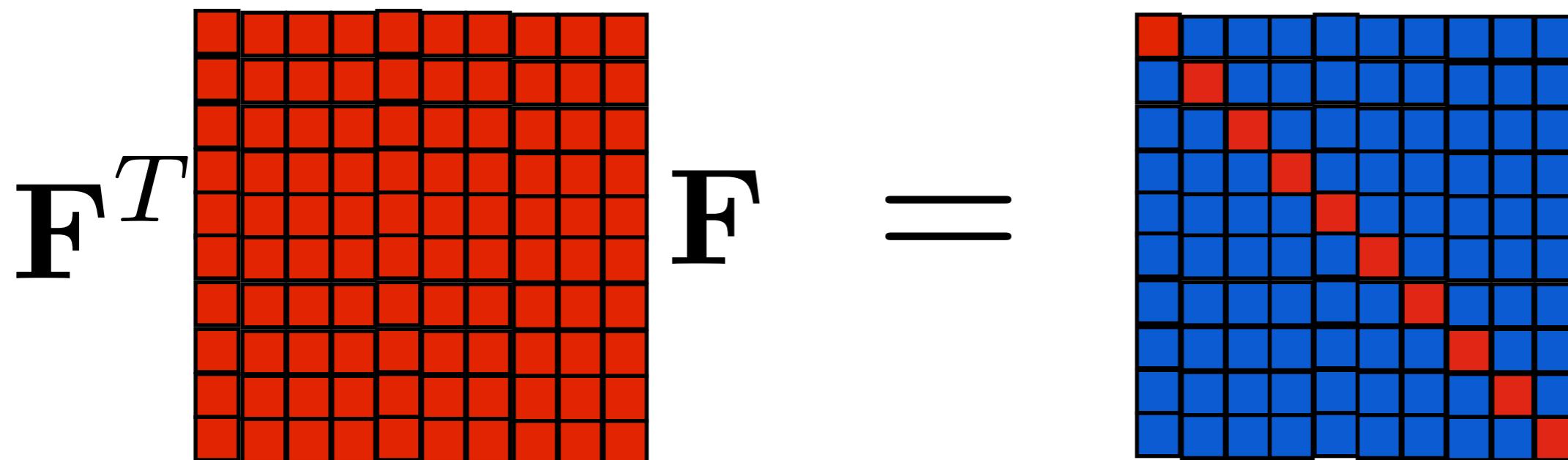
Not Always Zero    Always Zero



$\mathbf{x}[\tau]$  



 Not Always Zero     Always Zero



$\mathbf{F} \leftarrow$  Fourier Transform

■ Not Always Zero ■ Always Zero



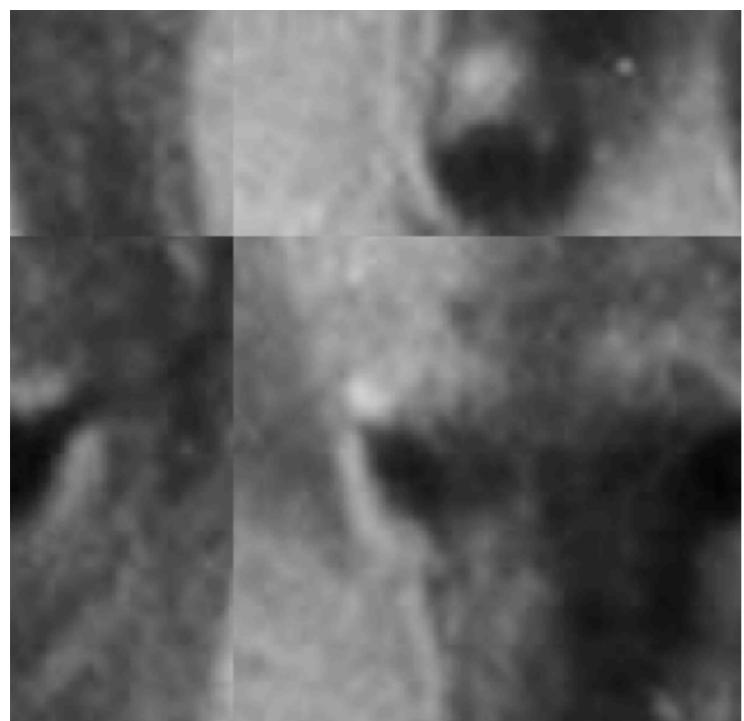
$\mathbf{x}[0, 0]$



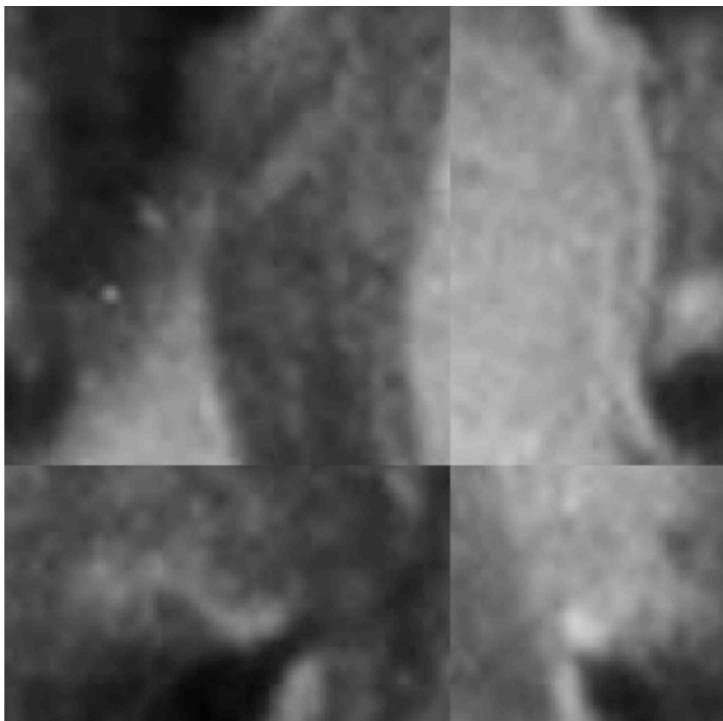
$\mathbf{x}[20, 20]$



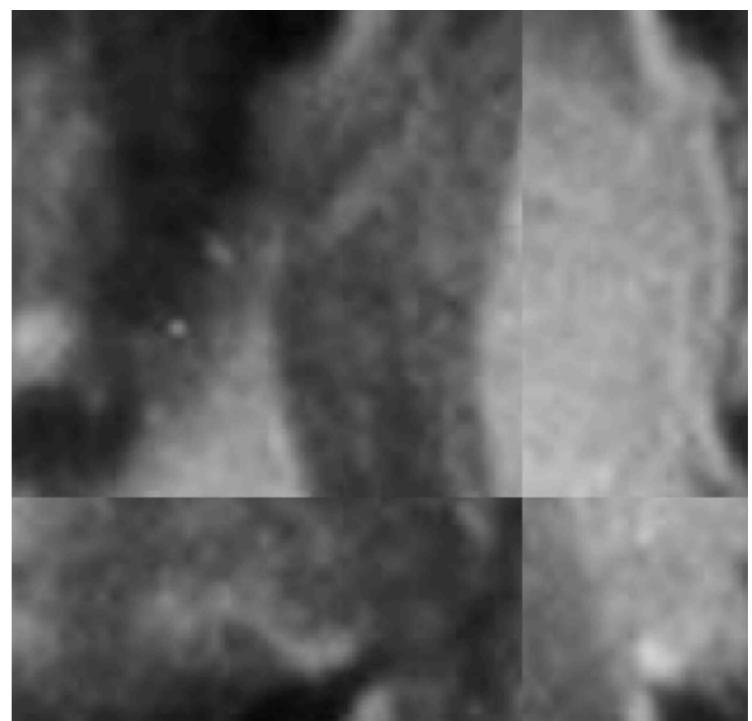
$\mathbf{x}[-20, -20]$



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



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



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



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



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



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



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



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

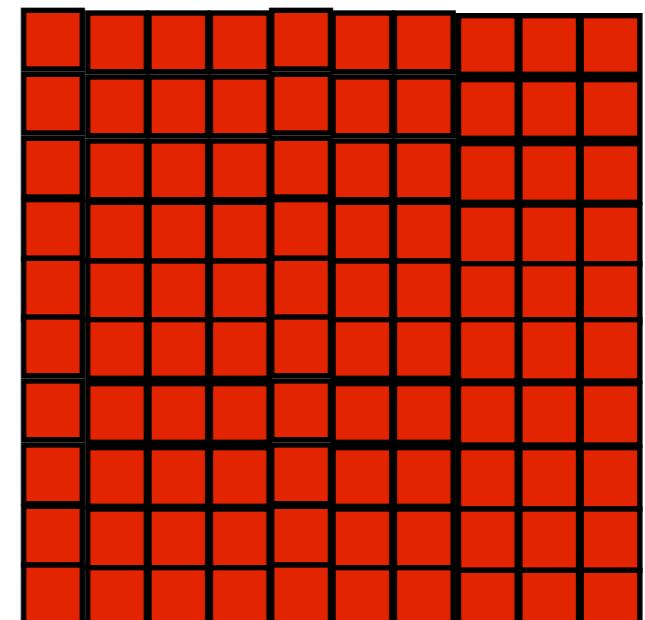


$\mathbf{x}[-20, -20]$

$$\mathbf{S} = \sum \mathbf{x}[\tau] \mathbf{x}[\tau]^T =$$

$\tau \in \mathcal{C}'$

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



$(D \times D)$



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



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



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

$$\mathbf{V}^T \mathbf{V} =$$
A diagram illustrating matrix multiplication. On the left, there is a 10x10 grid of red squares labeled  $\mathbf{V}^T$ . To its right is a black equals sign. To the right of the equals sign is another 10x10 grid, which is mostly blue with several red squares scattered across it, representing the result of the multiplication.

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

$$\hat{\mathbf{x}} = \begin{matrix} \text{[Red]} \\ \text{[Blue]} \\ \text{[Red]} \\ \text{[Blue]} \\ \text{[Red]} \end{matrix} = \begin{matrix} \text{[Blue]} & \text{[Red]} & \text{[Blue]} & \text{[Blue]} & \text{[Blue]} \\ \text{[Blue]} & \text{[Blue]} & \text{[Red]} & \text{[Blue]} & \text{[Blue]} \\ \text{[Blue]} & \text{[Blue]} & \text{[Blue]} & \text{[Red]} & \text{[Blue]} \\ \text{[Red]} & \text{[Blue]} & \text{[Blue]} & \text{[Blue]} & \text{[Blue]} \\ \text{[Blue]} & \text{[Blue]} & \text{[Blue]} & \text{[Blue]} & \text{[Red]} \end{matrix} \times \dots \times \begin{matrix} \text{[Blue]} & \text{[Blue]} & \text{[Red]} & \text{[Blue]} \\ \text{[Red]} & \text{[Blue]} & \text{[Blue]} & \text{[Blue]} \\ \text{[Blue]} & \text{[Blue]} & \text{[Blue]} & \text{[Red]} \\ \text{[Blue]} & \text{[Red]} & \text{[Blue]} & \text{[Blue]} \\ \text{[Blue]} & \text{[Blue]} & \text{[Blue]} & \text{[Red]} \end{matrix} \times \begin{matrix} \text{[Red]} \\ \text{[Red]} \\ \text{[Red]} \end{matrix}$$

$\hat{\mathbf{x}}$

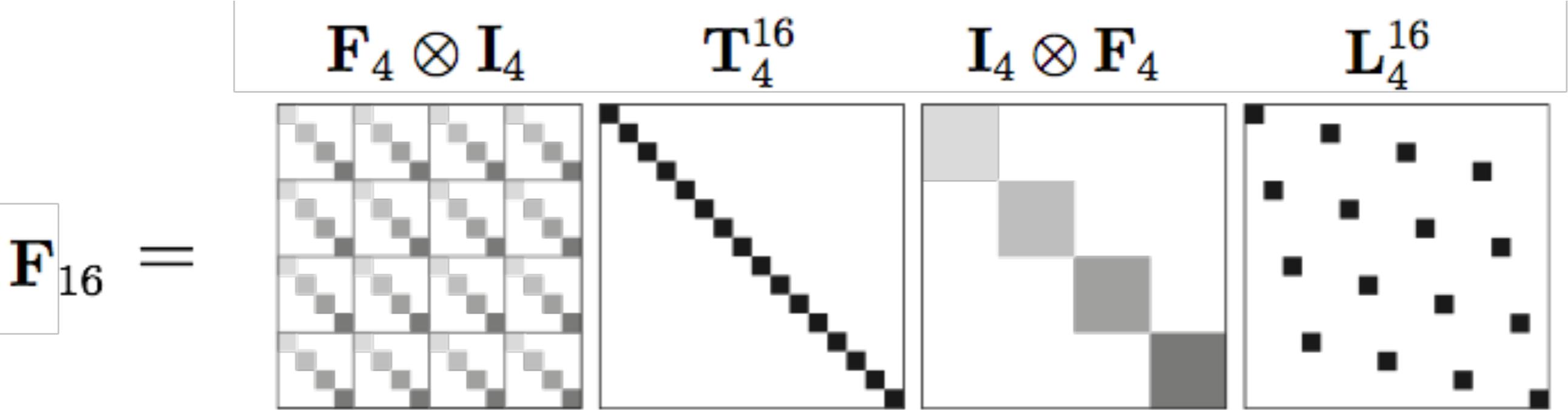
$\mathbf{F} \quad \mathbf{x}$

$$\mathcal{O}(D \log D)$$


Carl Friedrich Gauss

- Not Always Zero
- Always Zero

$\boxed{\mathbf{F}_{16}}$



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

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

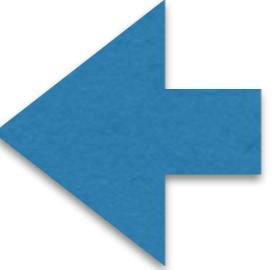
$\mathbf{L}_4^{16} \rightarrow$  permutation matrix

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

# FFT can be Real

27.0
4.93
1.57
5.57
0.50

$F_x$



8
4
6
2
7

$\hat{x}$

$x$

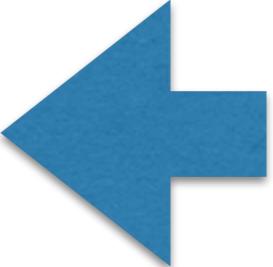
# FFT can be Complex

27.0
4.93
1.57
1.57
4.93

+ i

00.0
0.50
5.57
-5.57
-0.50

$F_x$



8
4
6
2
7

$\hat{x}$

$x$

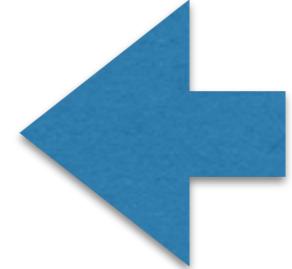
# FFT can be Complex

27.0
4.93
1.57
1.57
4.93

00.0
0.50
5.57
-5.57
-0.50

- i

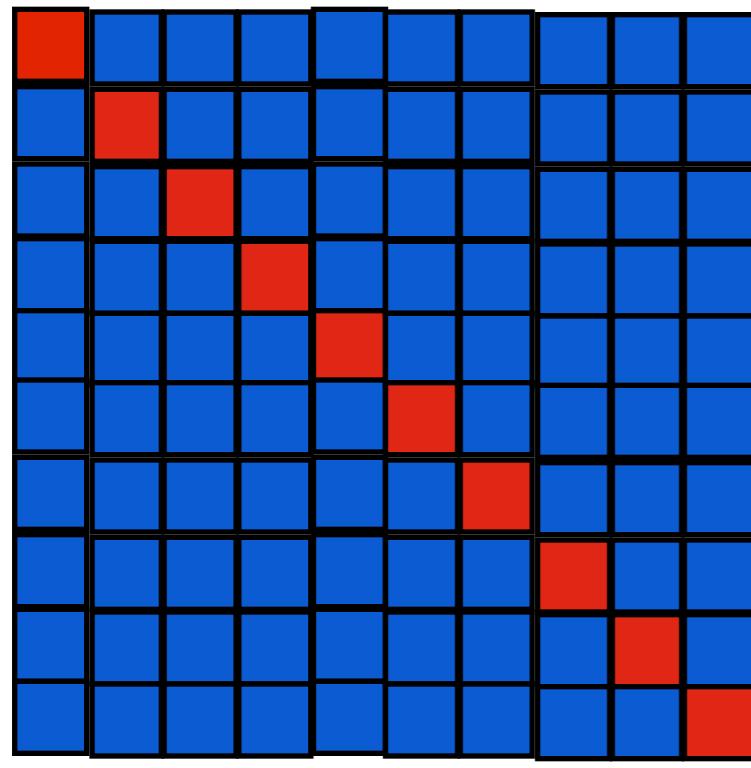
$F_x$



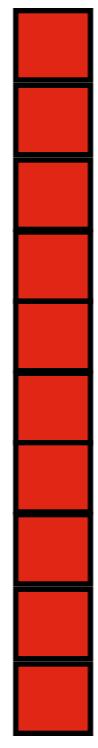
8
7
2
6
4

$\text{conj}(\hat{x})$

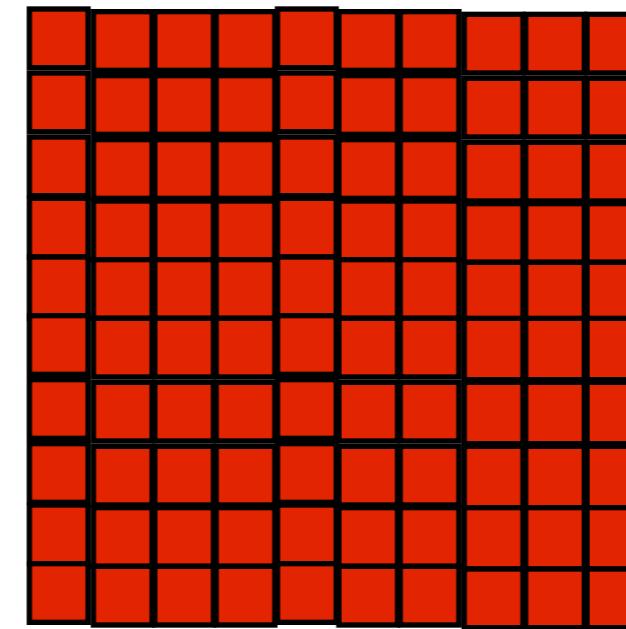
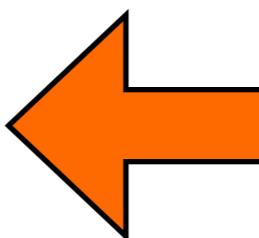
>> flipud(circshift(x, 4))



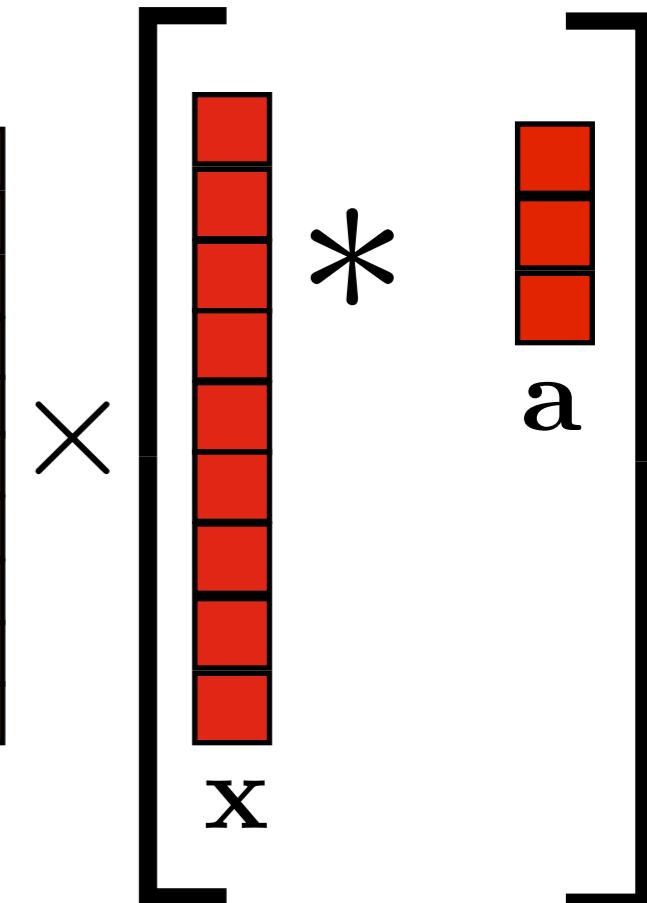
$\text{diag}\{\hat{a}\}$



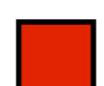
$\hat{x}$



$F$



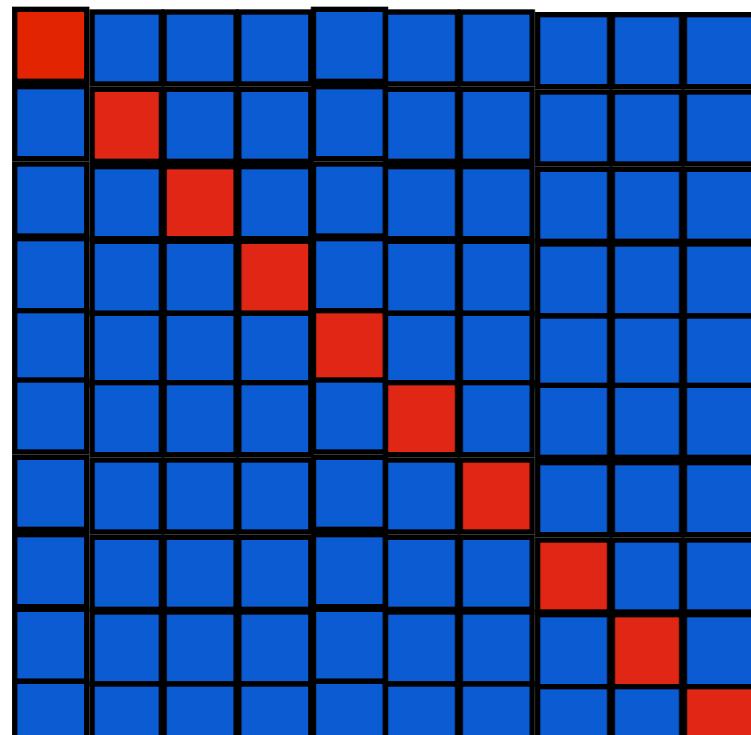
$$\text{diag}\{\hat{a}\}\hat{x} = F(a * x)$$



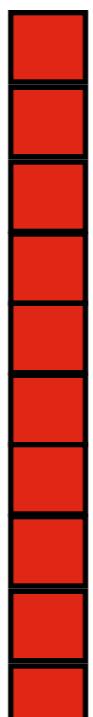
Not Always Zero



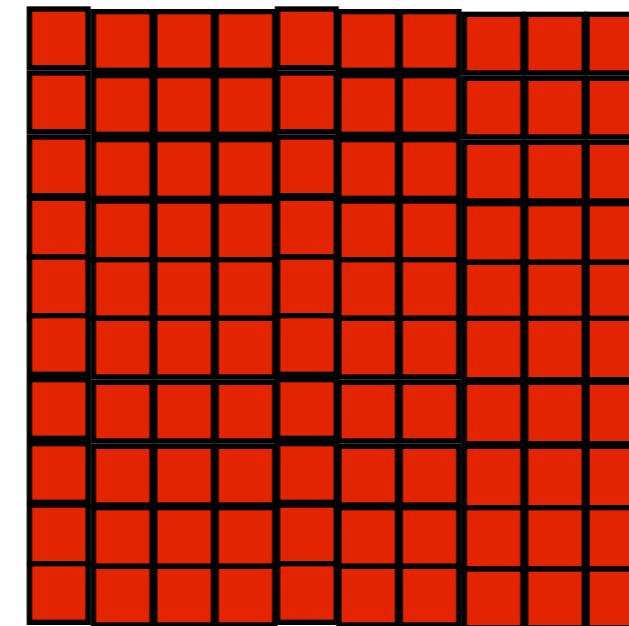
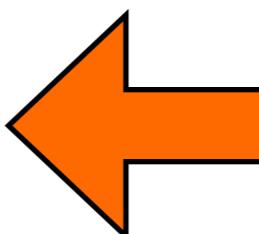
Always Zero



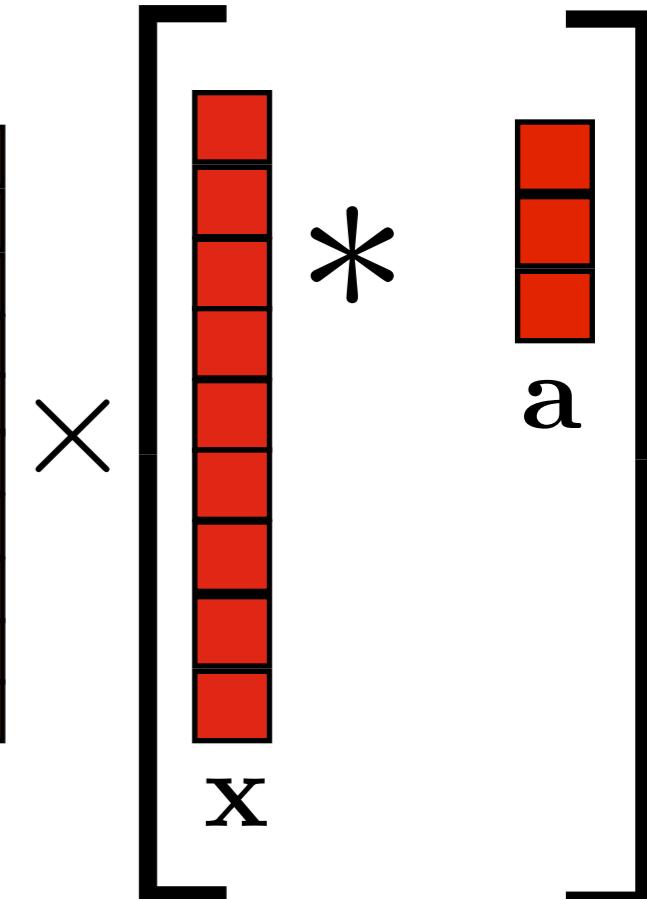
$\text{diag}\{\hat{a}\}$



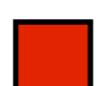
$\hat{x}$



$F$   
Fourier Transform



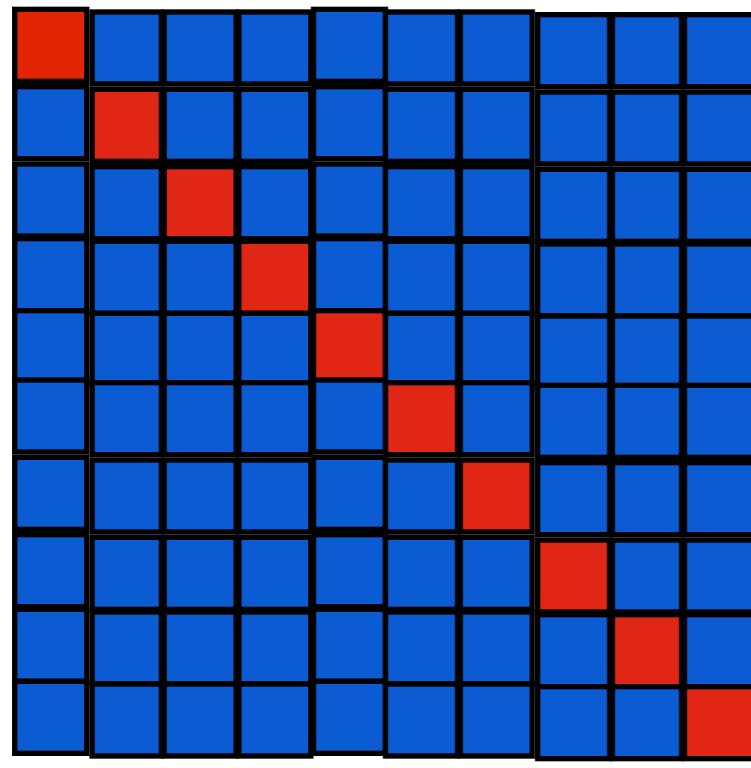
$$\hat{a} \circ \hat{x} = F(a * x)$$



Not Always Zero

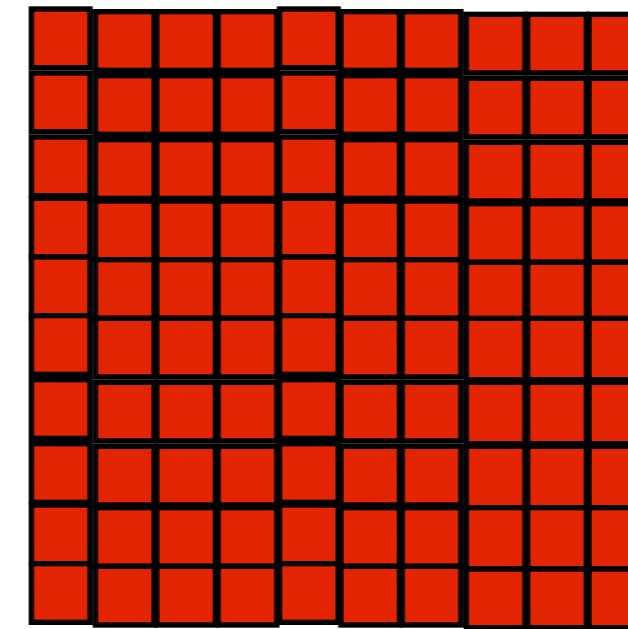
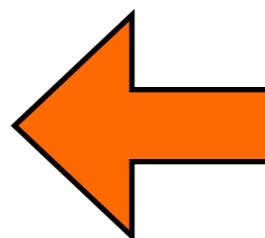


Always Zero

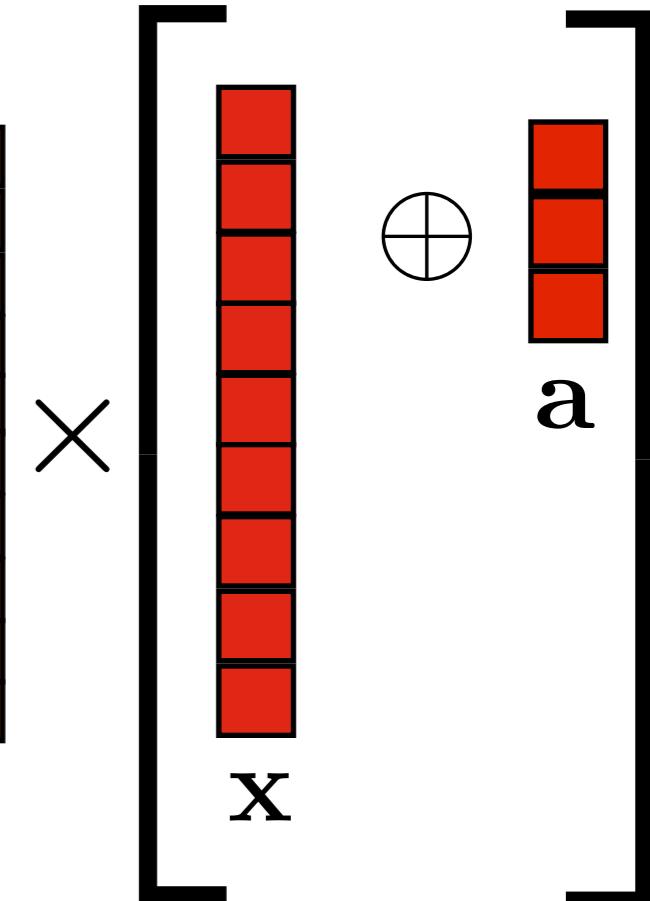


$$\text{diag}\{\hat{\mathbf{a}}\}^T$$

$$\hat{\mathbf{x}}$$



$$\mathbf{F}$$



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



Not Always Zero



Always Zero

# Today

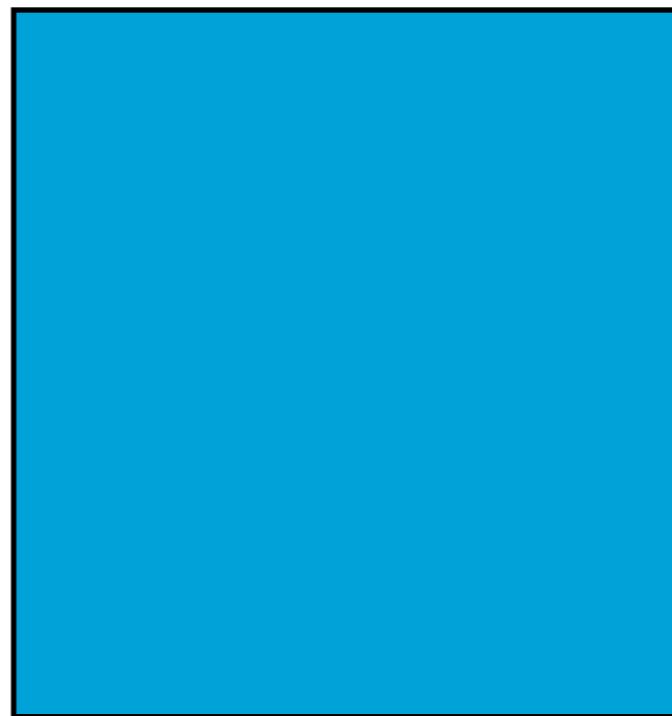
---

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

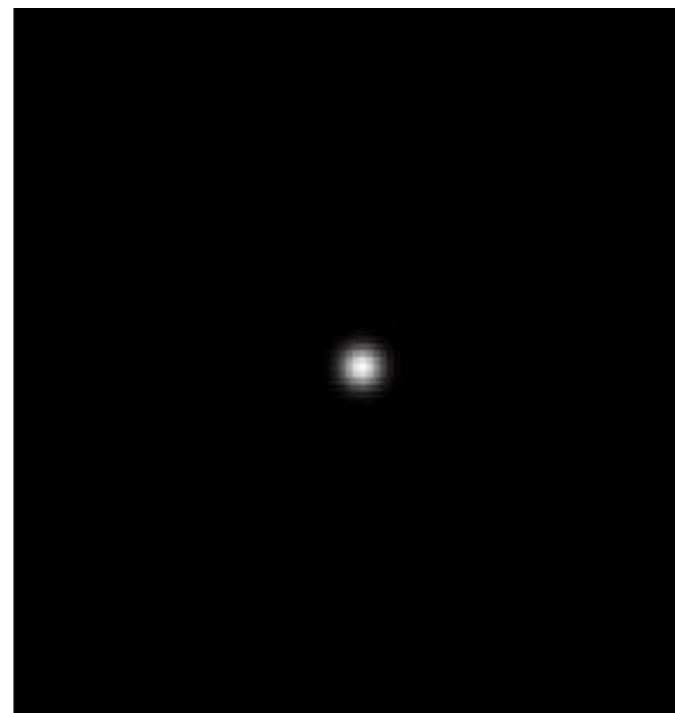


“known signal”  $\mathbf{X}$

\*

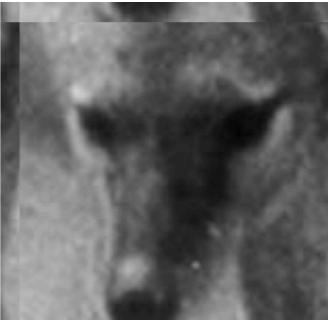
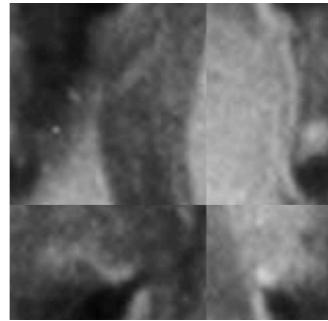


“unknown  
filter”  
 $\mathbf{h}$

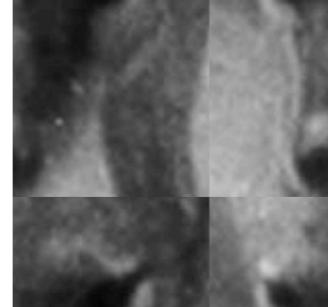


“known response”  $\mathbf{y}$

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

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

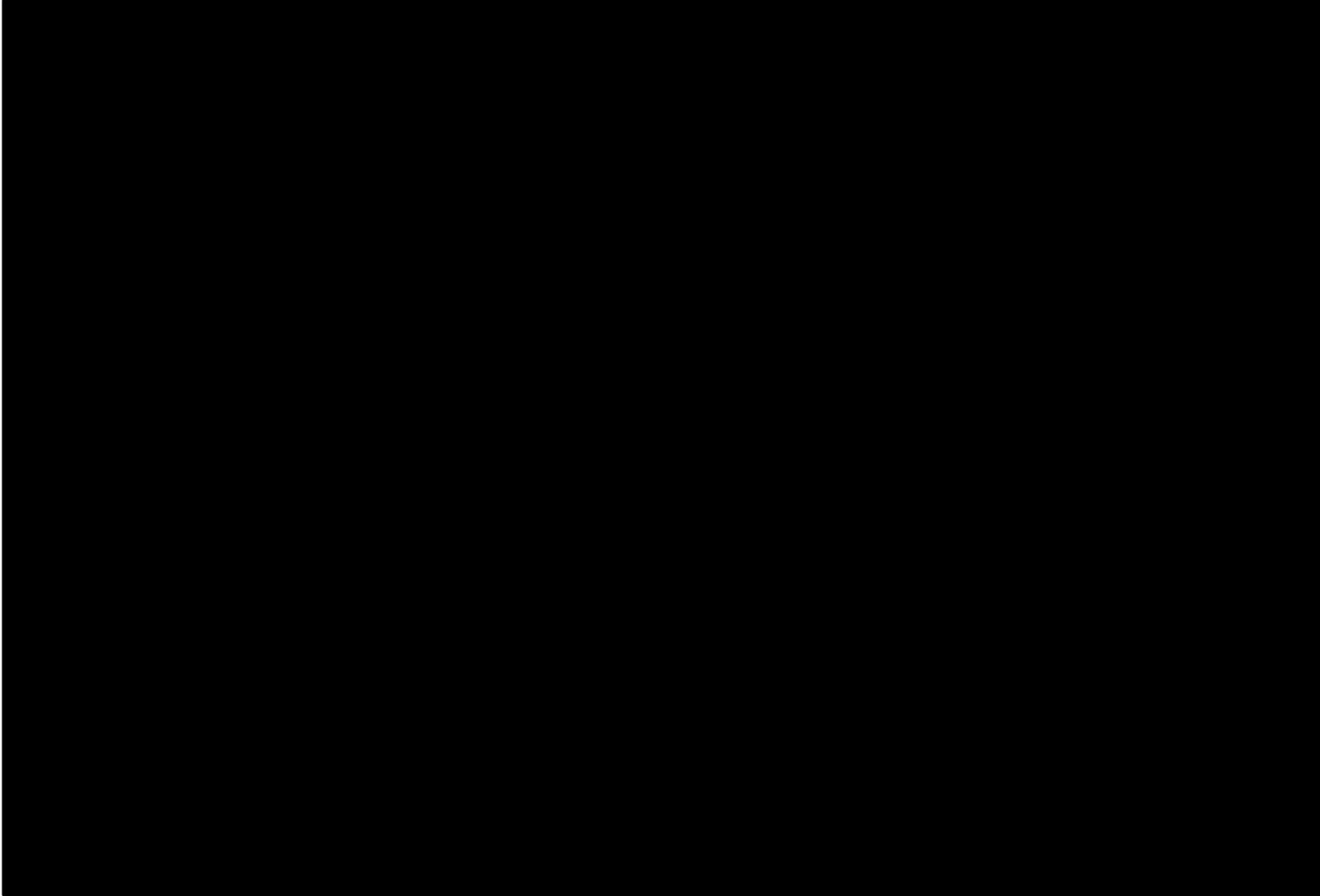
$$E(\mathbf{h}) = \frac{1}{2} \sum_{\tau \in \mathcal{C}} ||y_\tau - \mathbf{x}[\tau]^T \mathbf{h}||_2^2 + \frac{\lambda}{2} ||\mathbf{h}||_2^2$$

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

# Trust Regions

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

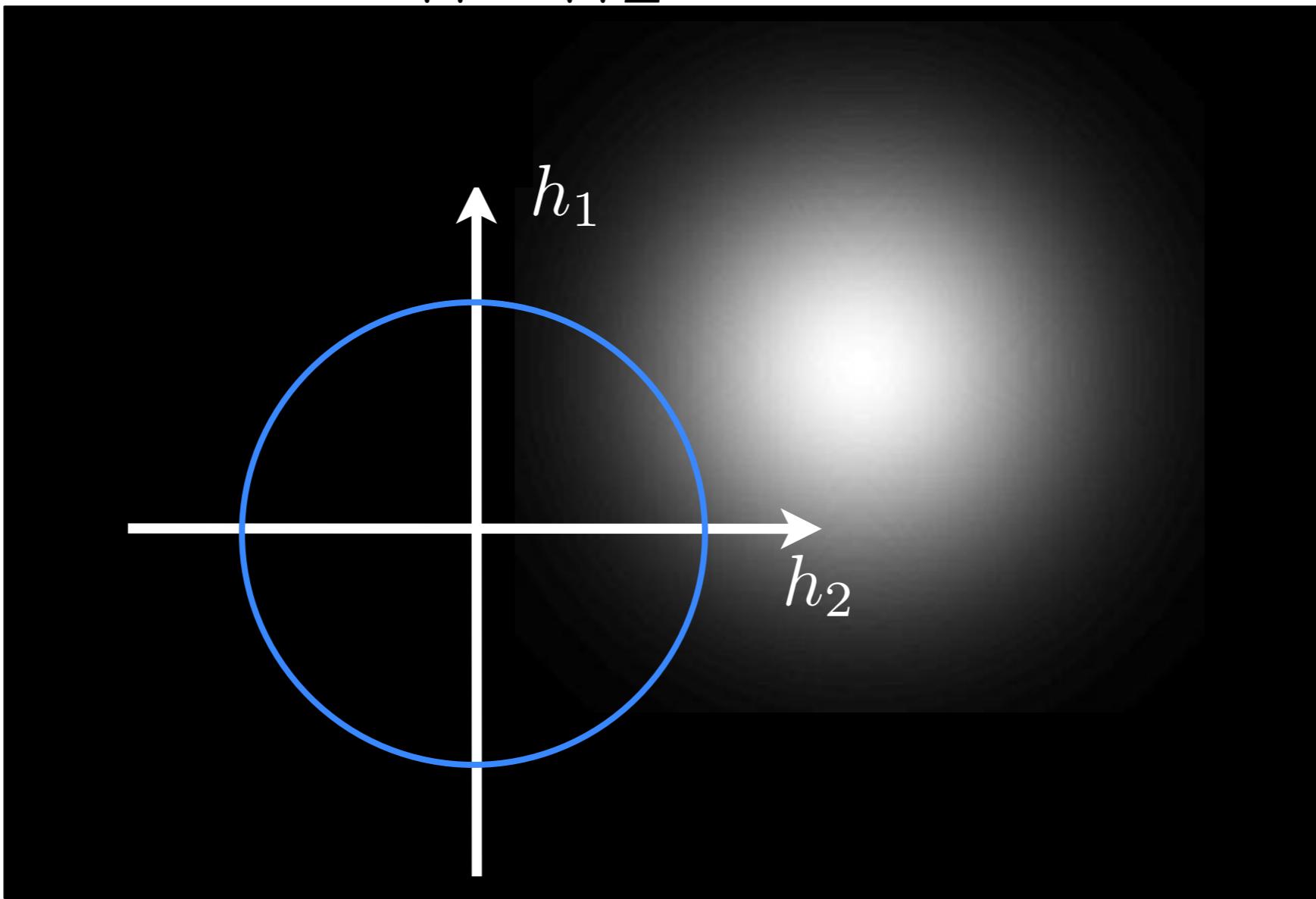
s.t.  $||\mathbf{h}||_2^2 \leq \epsilon$



# Trust Regions

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

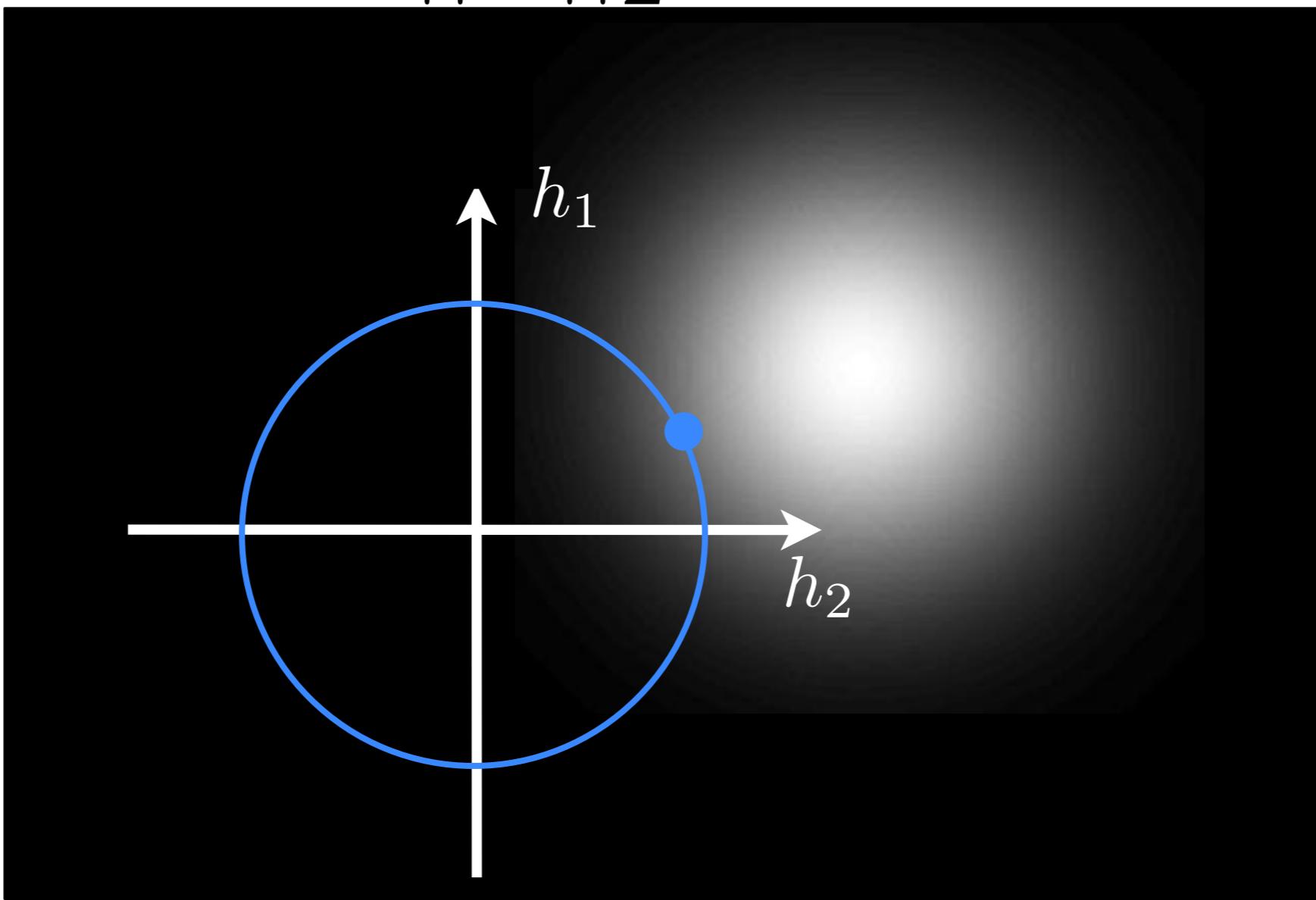
s.t.  $||\mathbf{h}||_2^2 \leq \epsilon$



# Trust Regions

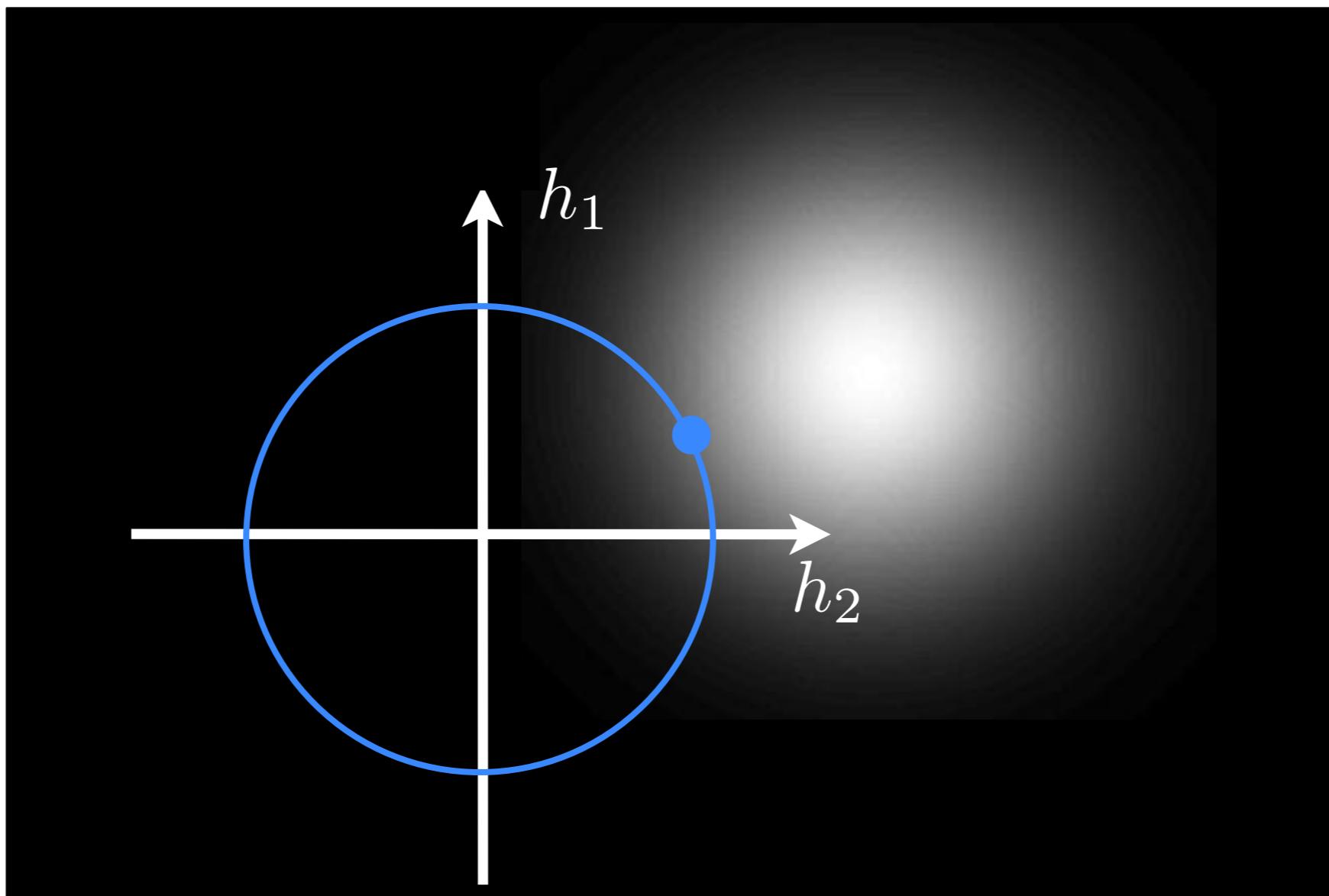
$$E(\mathbf{h}) = \frac{1}{2} \sum_{\tau \in \mathcal{C}} ||y_\tau - \mathbf{x}[\tau]^T \mathbf{h}||_2^2$$

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



# Trust Regions

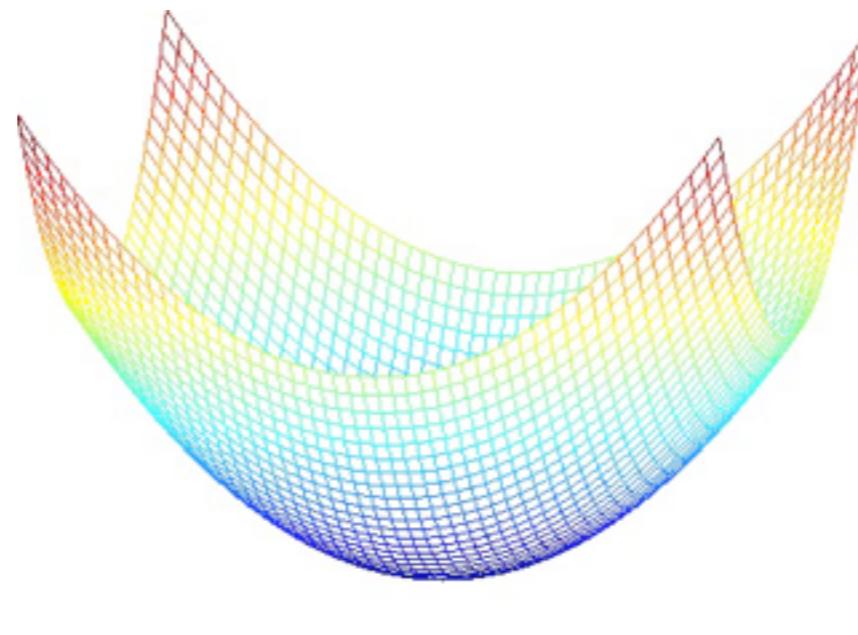
$$E(\mathbf{h}) = \frac{1}{2} \sum_{\tau \in \mathcal{C}} \|y_\tau - \mathbf{x}[\boldsymbol{\tau}]^T \mathbf{h}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2$$



# 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.

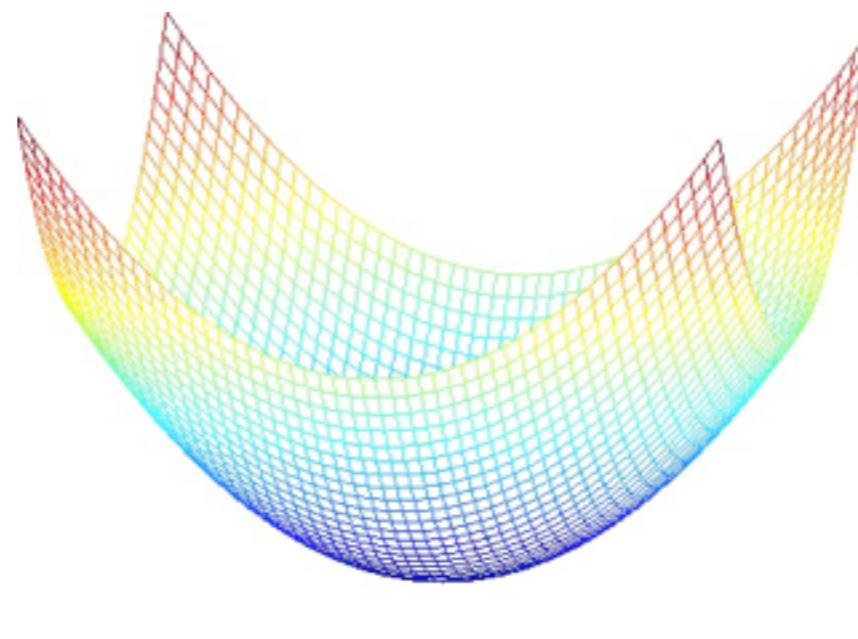
$$\arg \min_{\mathbf{w}, w_0} \sum_{n=1}^N \|t_i - \mathbf{w}^T \mathbf{x}_i - w_0\|_2^2$$



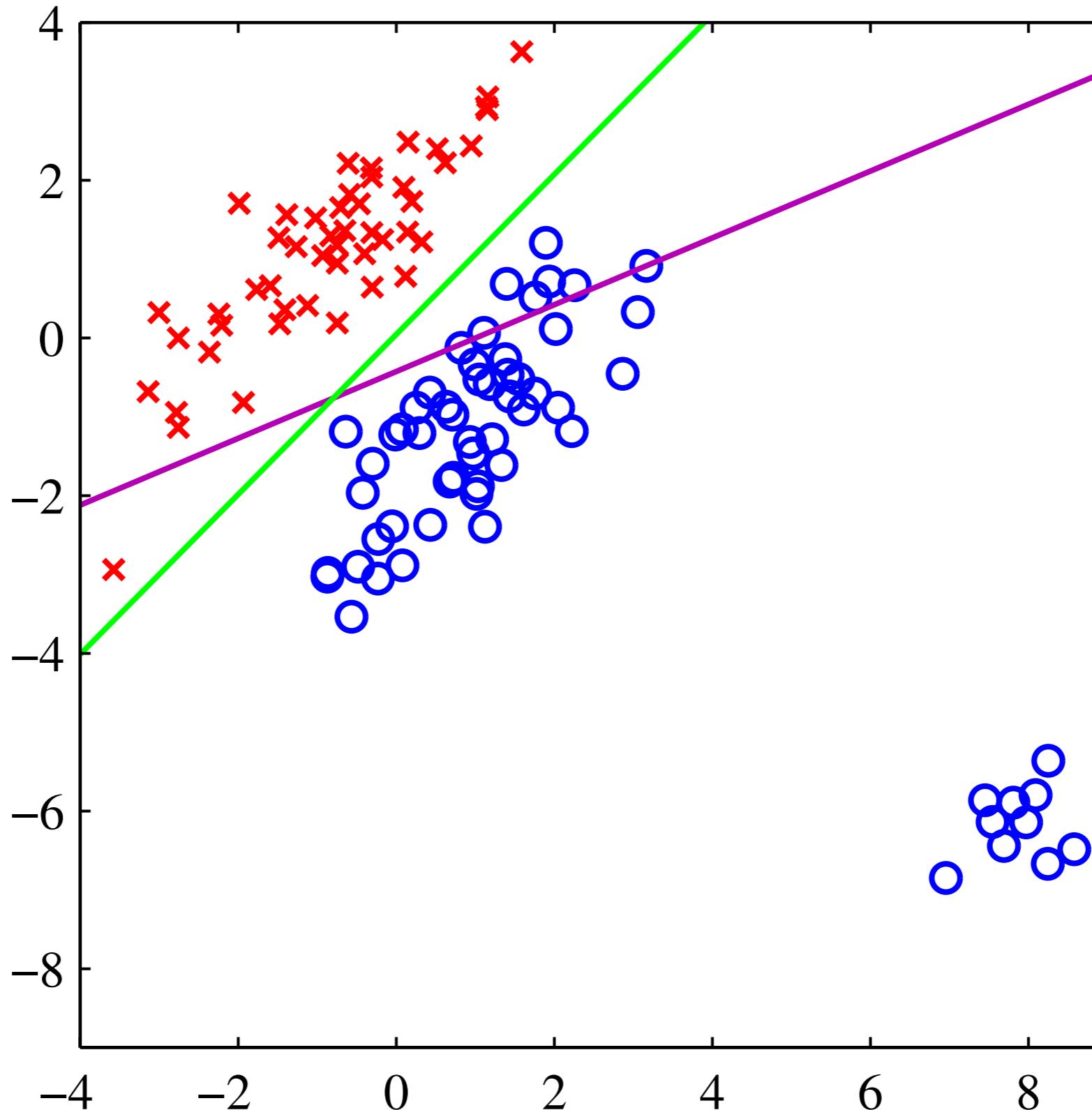
# Linear Least Squares Discriminant

- 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



Detection rate

1

0.8

0.6

0.4

0.2

0

0.1

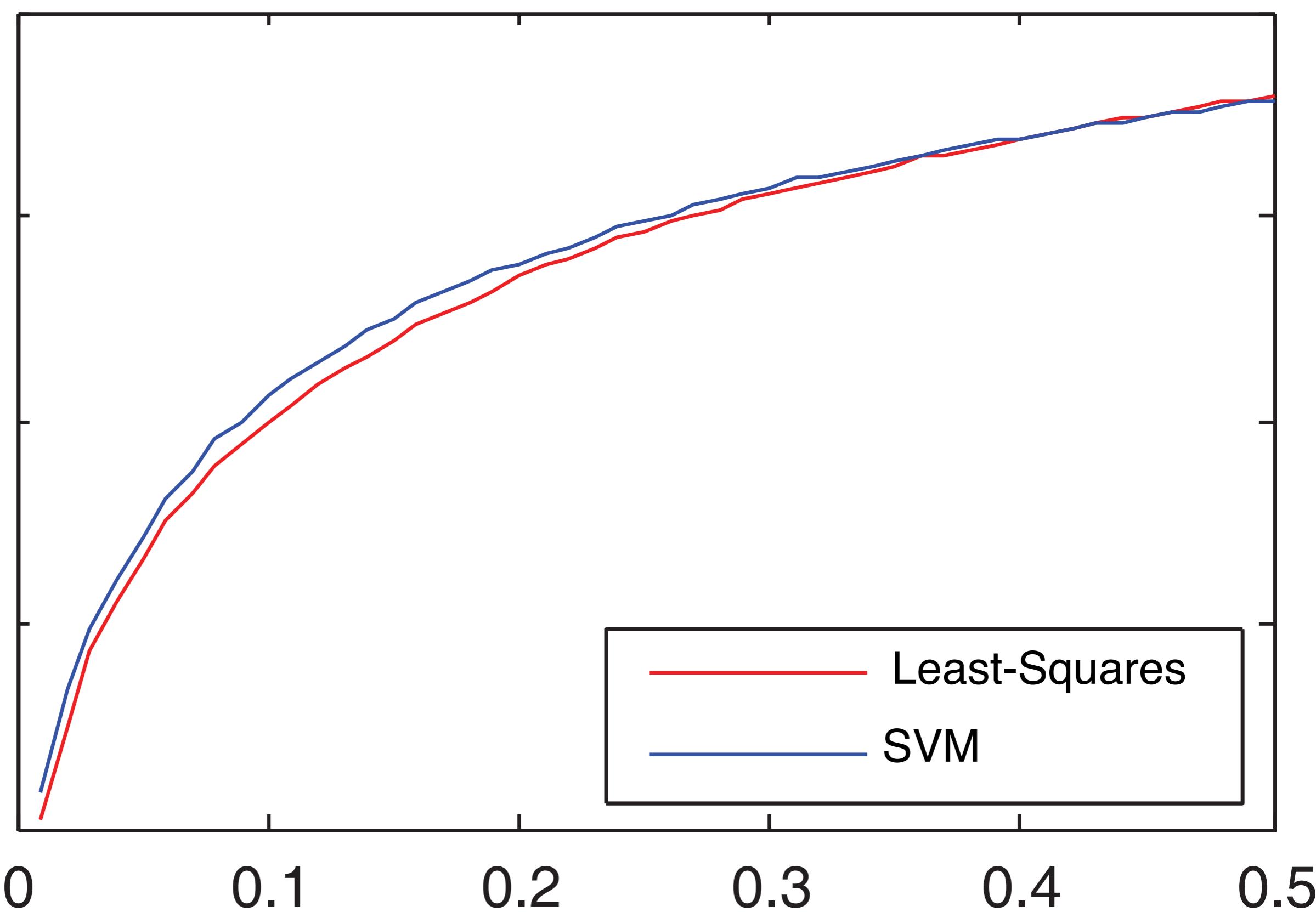
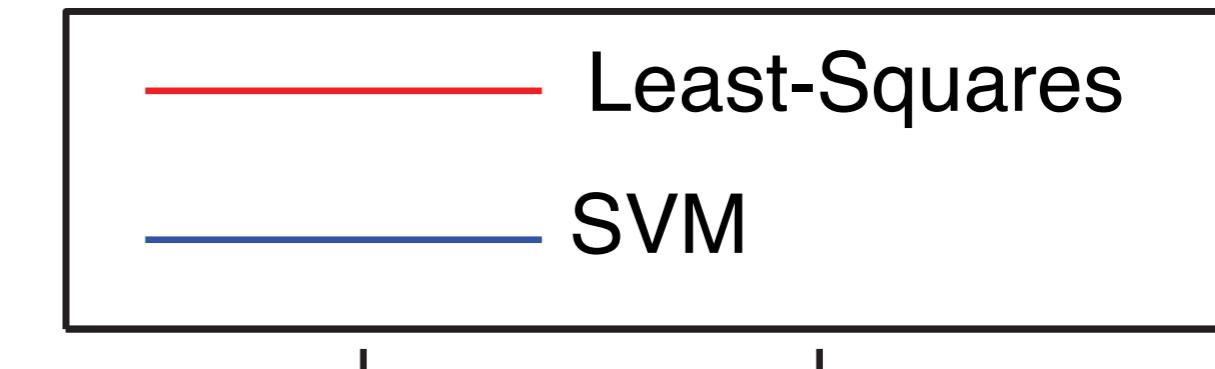
0.2

0.3

0.4

0.5

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



Detection rate

1

0.8

0.6

0.4

0.2

0

0.1

0.2

0.3

0.4

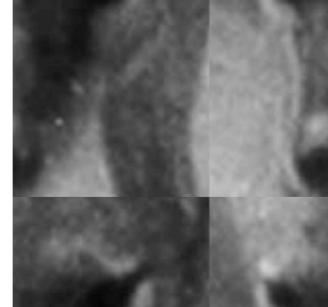
0.5

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

Least-Squares  
SVM



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

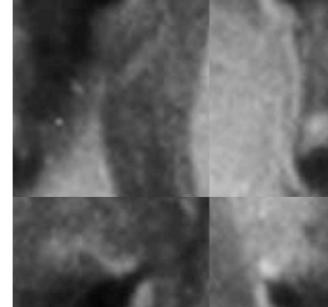
$\mathbf{x}[\tau]$			$\dots$	
$y_\tau$	1	0	$\dots$	0

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

“set of all  
circular shifts”

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

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

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

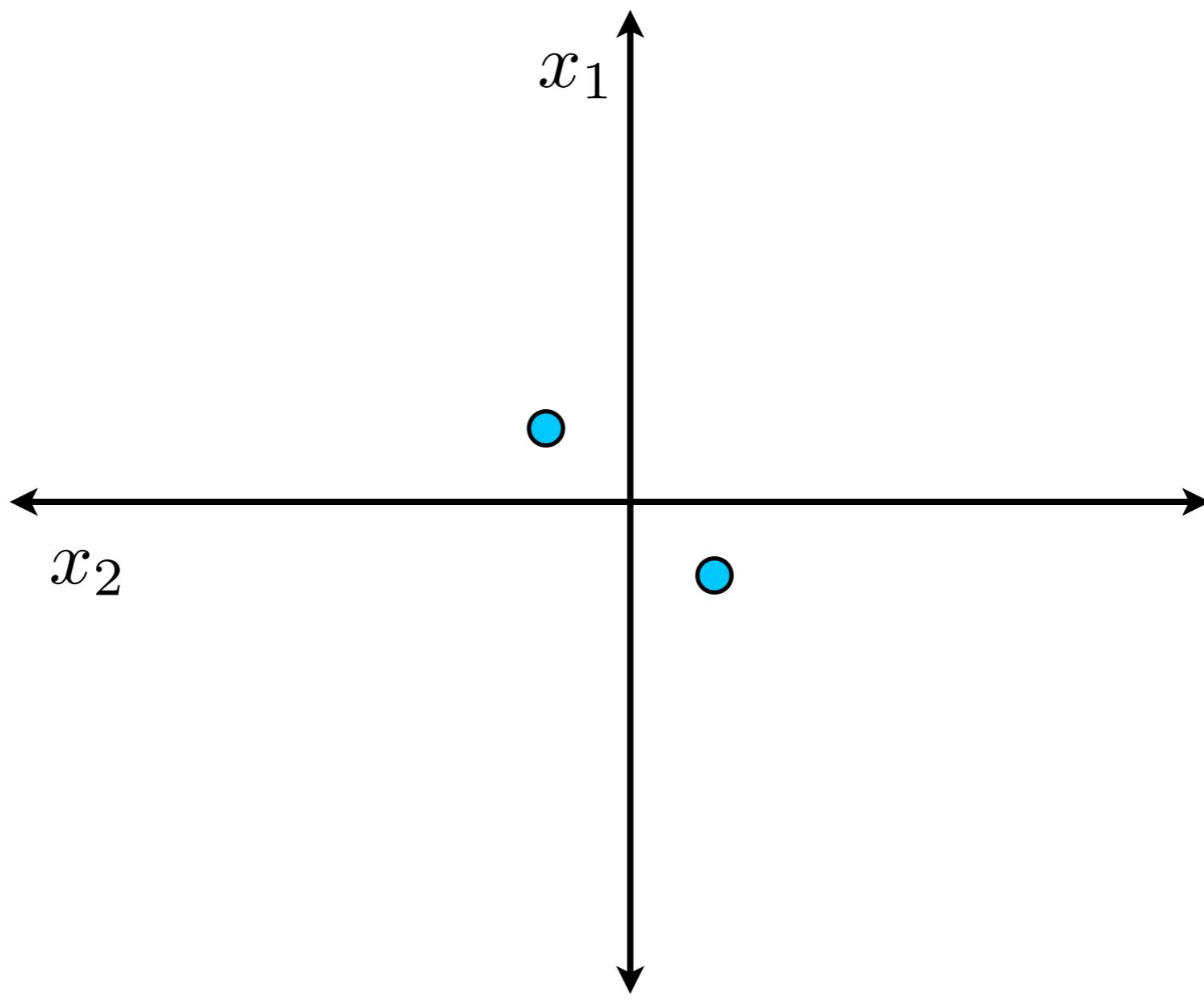
$$E(\mathbf{h}) = \frac{1}{2}||\mathbf{y}-\mathbf{X}\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$

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

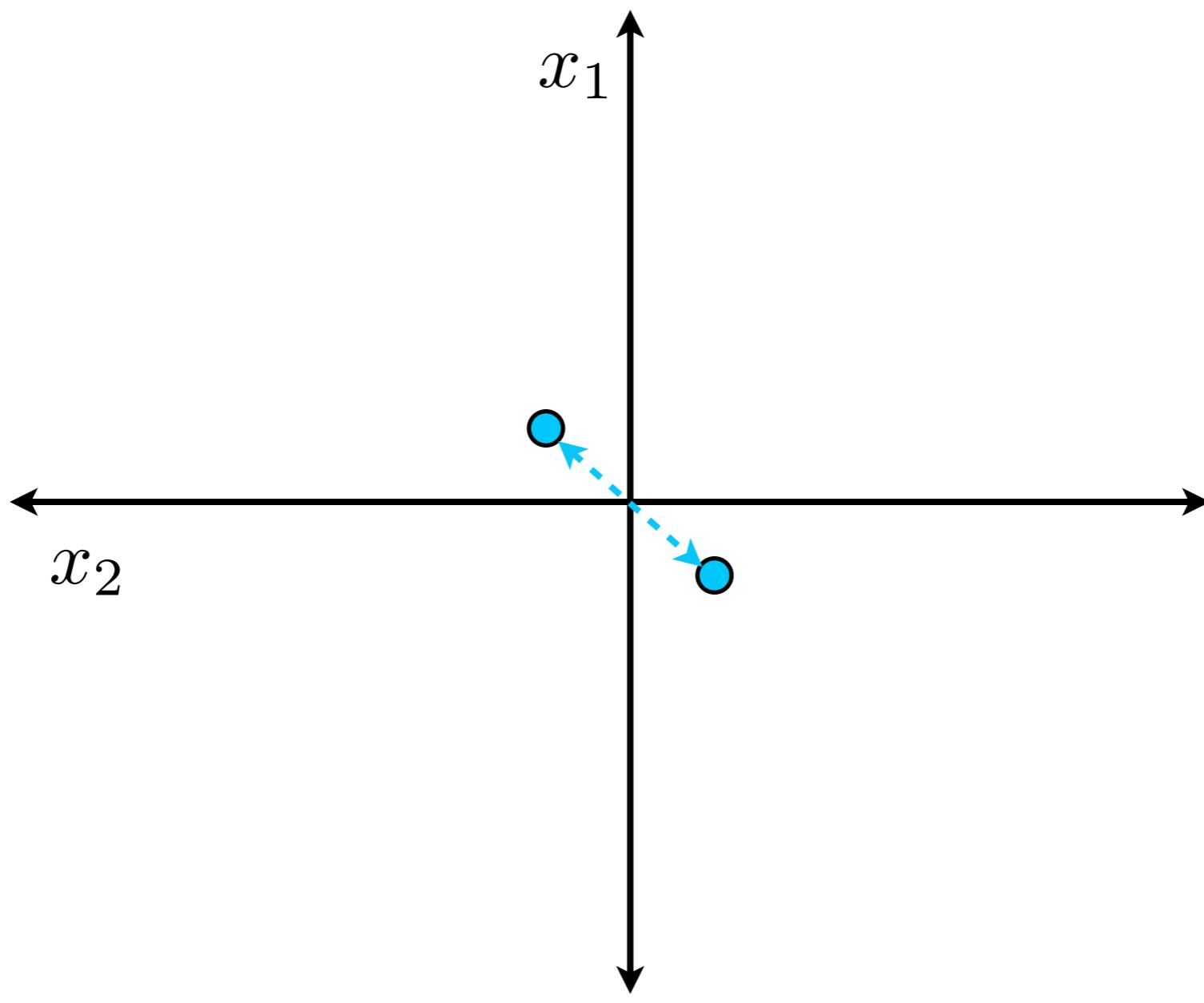
$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \rightarrow \mathcal{O}(D^3)$$

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

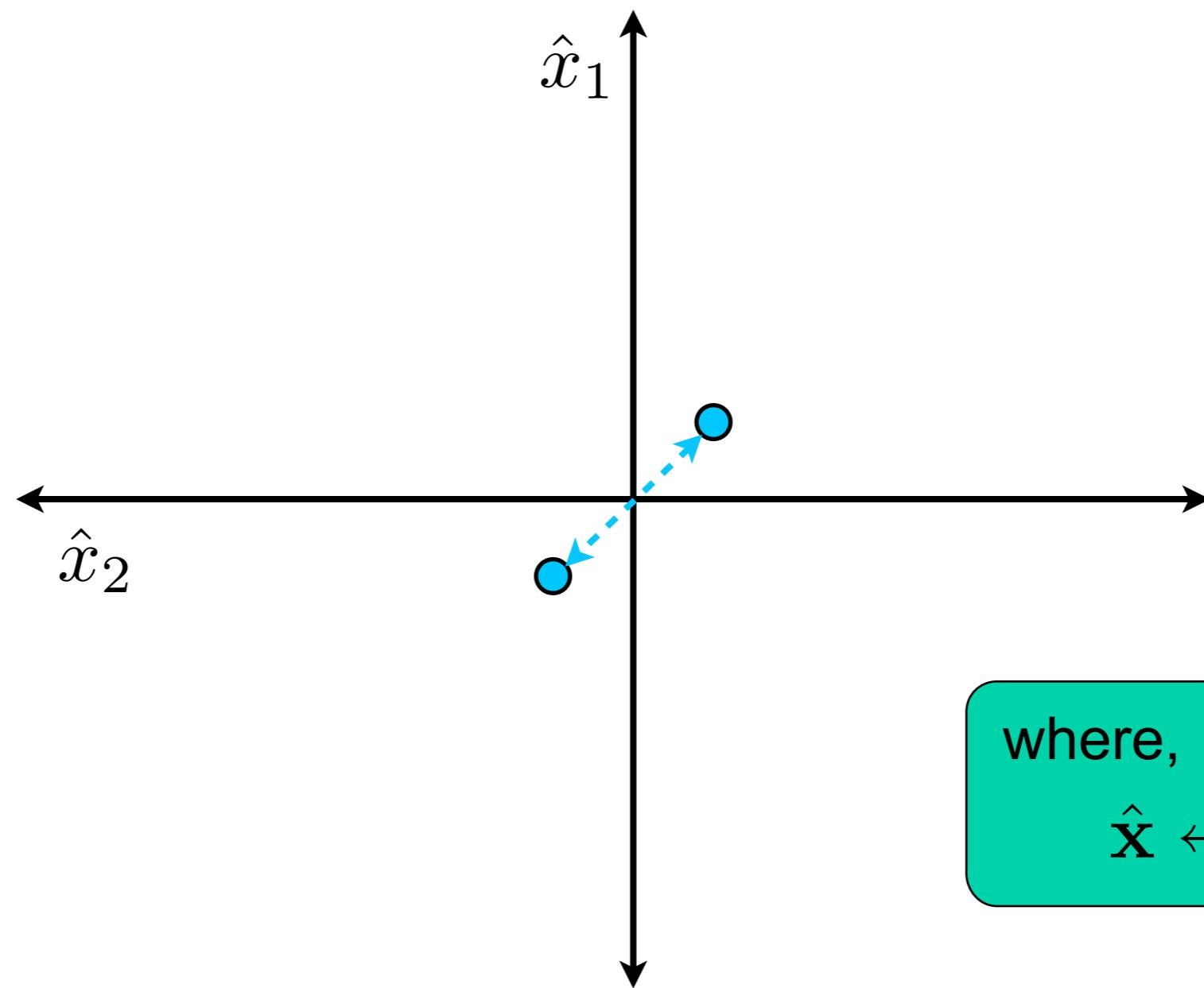
$$E(\mathbf{h}) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{h}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2$$



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



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



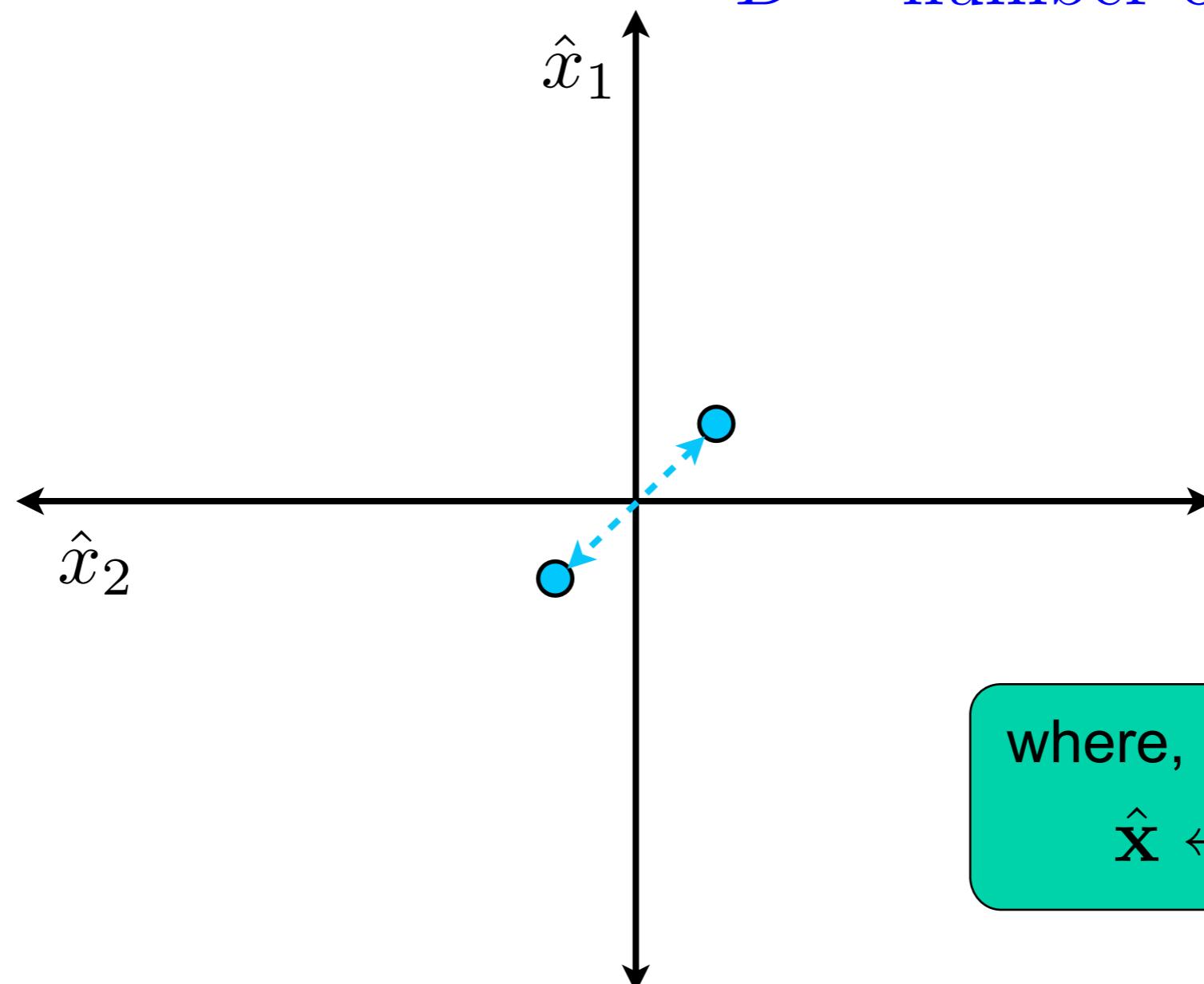
where,

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

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

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

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



where,

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

$$\hat{\mathbf{h}} \; = \; \hat{\mathbf{S}}_{xy}^{-1}\left(\hat{\mathbf{S}}_{xx} + \lambda\mathbf{1}\right)$$

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

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

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

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

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

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

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

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

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

```
>> 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
<b>MOSSE Filters</b>	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
<b>MOSSE Filters</b>	669fps	2.4Ghz Core 2 Duo

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

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

$N$  = number of training images

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

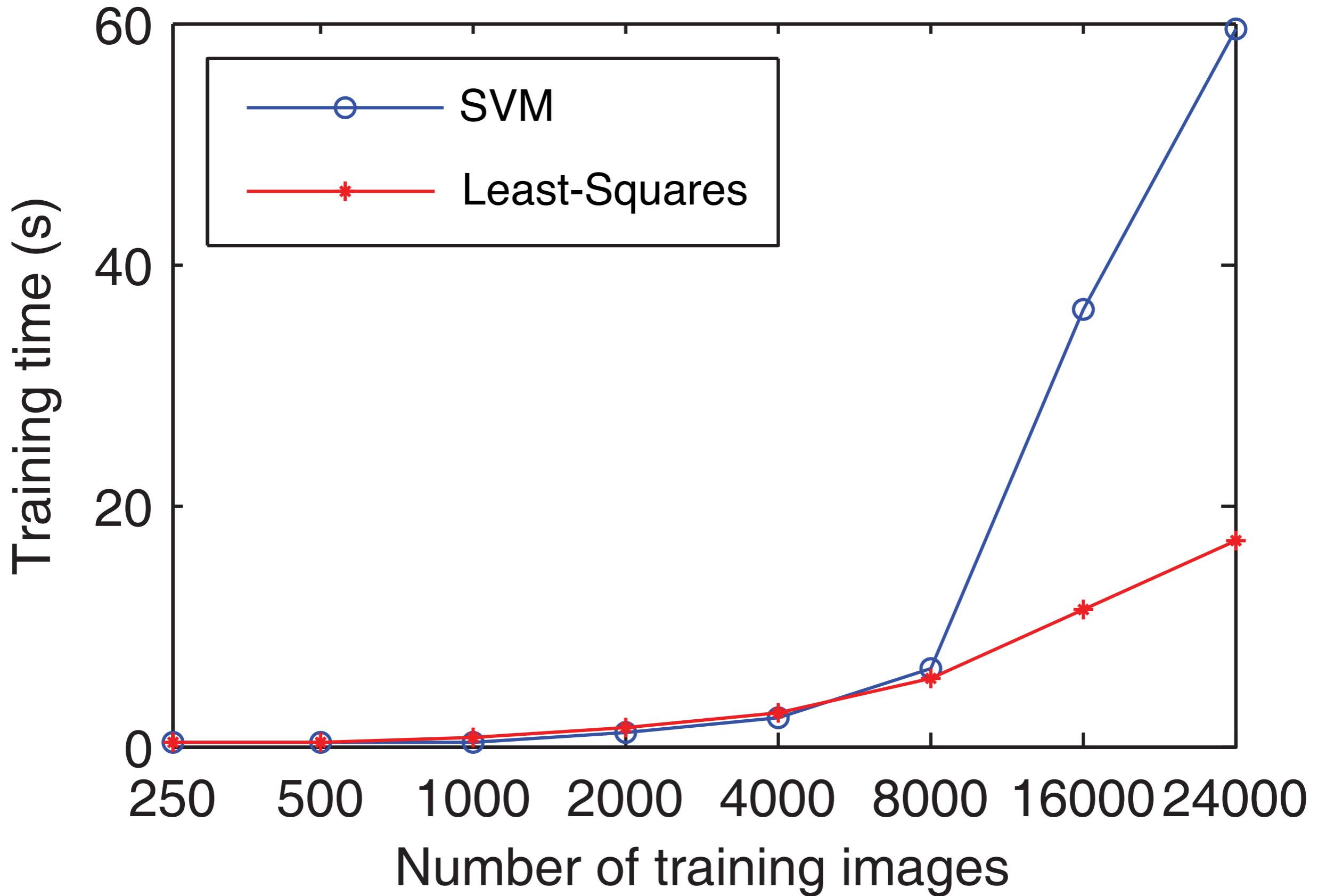
$$\hat{\mathbf{h}} = \hat{\mathbf{S}}_{xy}^{-1} (\hat{\mathbf{S}}_{xx} + \lambda \mathbf{1})$$

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

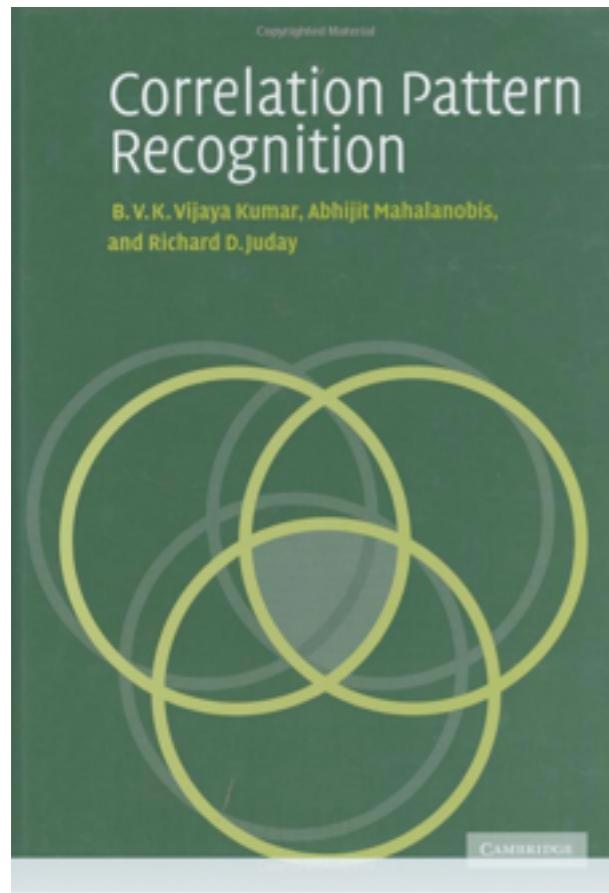
$N$  = number of training images

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

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



# More to read...



**GvF** This ICCV2013 paper is the Open Access version, provided by the Computer Vision Foundation. The authoritative version of this paper is available in IEEE Xplore.

**Multi-Channel Correlation Filters**

Hamed Kiani Galoogahi  
National University of Singapore  
Singapore  
hkiani@comp.nus.edu.sg

Terence Sim  
National University of Singapore  
Singapore  
tsim@comp.nus.edu.sg

Simon Lucey  
CSIRO  
Australia  
simon.lucey@csiro.au

**Abstract**

Modern descriptors like HOG and SIFT are now commonly used in vision for pattern detection within image and video. From a signal processing perspective, this is equivalent to performing a multi-channel correlation/convolution between a multi-channel image and a multi-channel detector/filter which results in a single-channel response map indicating where the pattern (e.g. object) has occurred. In this paper, we propose a novel framework for learning a multi-channel detector/filter efficiently in the frequency domain, both in terms of training time and memory footprint. Such a filter refers to as a multi-channel footprint. To demonstrate the effectiveness of our strategy, we evaluate it across a number of visual detection/localization tasks where we: (i) exhibit superior performance to current state of the art correlation filters, and (ii) superior computational and memory efficiencies compared to state of the art spatial detectors.

**1. Introduction**

In computer vision it is now rare for tasks like convolution/correlation to be performed on single channel image signals (e.g. 2D array of intensity values). With the advent of advanced descriptors like HOG [5] and SIFT [3] multi-channel auto-correlation across multi-channel signals has become the norm rather than the exception in most visual detection tasks. Most of these image descriptors can be viewed as multi-channel images/signals with multiple measurements (such the oriented edge energies) associated with each pixel location. We shall herein refer to all image descriptors as multi-channel images. An example of multi-channel correlation has been shown in Fig. 1. A multi-channel image  $x$  is convolved/convolved with a multi-channel filter  $h$  to give a single-channel response  $y$ .

The has not always been the case. Correlation filters, developed initially in the seminal work of Hester and Casenave [8], are a method for learning a template/filter in the frequency domain. These were first introduced in the 80s and 90s. Although many variants have been proposed [8, 11, 12], the approach's central tenet is to learn a filter, that when correlated with a set of training signals, gives a desired response (typically a peak at the origin of the object, with all other regions of the correlation response map being suppressed). Like correlation itself, one of the central advantages of the single channel approach is that it is computationally efficient. This is due primarily to the efficiency of correlation/convolution in that domain. Learning multi-channel filters in the frequency domain, however, comes at the high cost of computation and memory usage. In this paper we present an efficient strategy for learning multi-channel signals/filters that has numerous applications throughout vision and learning.

Like single channel signals, correlation between two multi-channel signals is rarely performed naively in the space-

3072

- 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.

**Visual Object Tracking using Adaptive Correlation Filters**

David S. Bolme J. Ross Beveridge Bruce A. Draper Yui Man Lui  
Computer Science Department Colorado State University Fort Collins, CO 80521, USA  
bolme@cs.colostate.edu

**Abstract**

Although not commonly used, correlation filters can track complex objects through rotations, occlusions and other distractions at over 20 times the rate of current state-of-the-art techniques. The oldest and simplest correlation filters use simple templates and generally fail when applied to tracking. More modern approaches such as ASEF and UMACE perform better, but their training needs are poorly suited to tracking. Visual tracking requires robust filters to be trained from a single frame and dynamically adapted as the appearance of the target object changes.

This paper proposes a new type of correlation filter, a Minimum Output Sum of Squared Errors (MOSSE) filter, which provides stable correlation filters when initialized using a single frame. A tracker based upon MOSSE filters is robust to variations in lighting, scale, pose, and non-rigid deformations while operating at 669 frames per second. Occlusion is detected based upon the peak-to-sidelobe ratio, which enables the tracker to pause and resume where it left off when the object reappears.

*Note: This paper contains additional figures and content that was excluded from CVPR 2010 to meet length requirements.*

**1 Introduction**

Visual tracking has many practical applications in video processing. When a target is located in one frame of a video, it is often useful to track that object in subsequent frames. Every frame in which the target is successfully tracked provides more information about the identity and the activity of the target. Because tracking is easier than detection, tracking algorithms can use fewer computational resources than running an object detector on every frame.

In this paper we investigate a simpler tracking strategy. The targets appearance is modeled by adaptive correlation filters, and tracking is performed via convolution. Naive

years. A number of robust tracking strategies have been proposed that tolerate changes in target appearance and track targets through occlusions. Some examples include Incremental Visual Tracking (IVT) [17], Robust Feature-based Tracking (Fractrak) [1], Gradient Based Discriminative Learning (GDL) [19], and Multiple Instance Learning (MILITrack) [2]. Although effective, these techniques are not simple; they often include complex appearance models and/or optimization algorithms, and as result struggle to keep up with the 25 to 30 frames per second produced by many modern cameras (See Table 1).