Using translation in learning-based vision

Instructor - Jack Valmadre

Maths for AI - 2022

AUSTRALIAN INSTITUTE FOR MACHINE LEARNING

THE UNIVERSITY of ADELAIDE

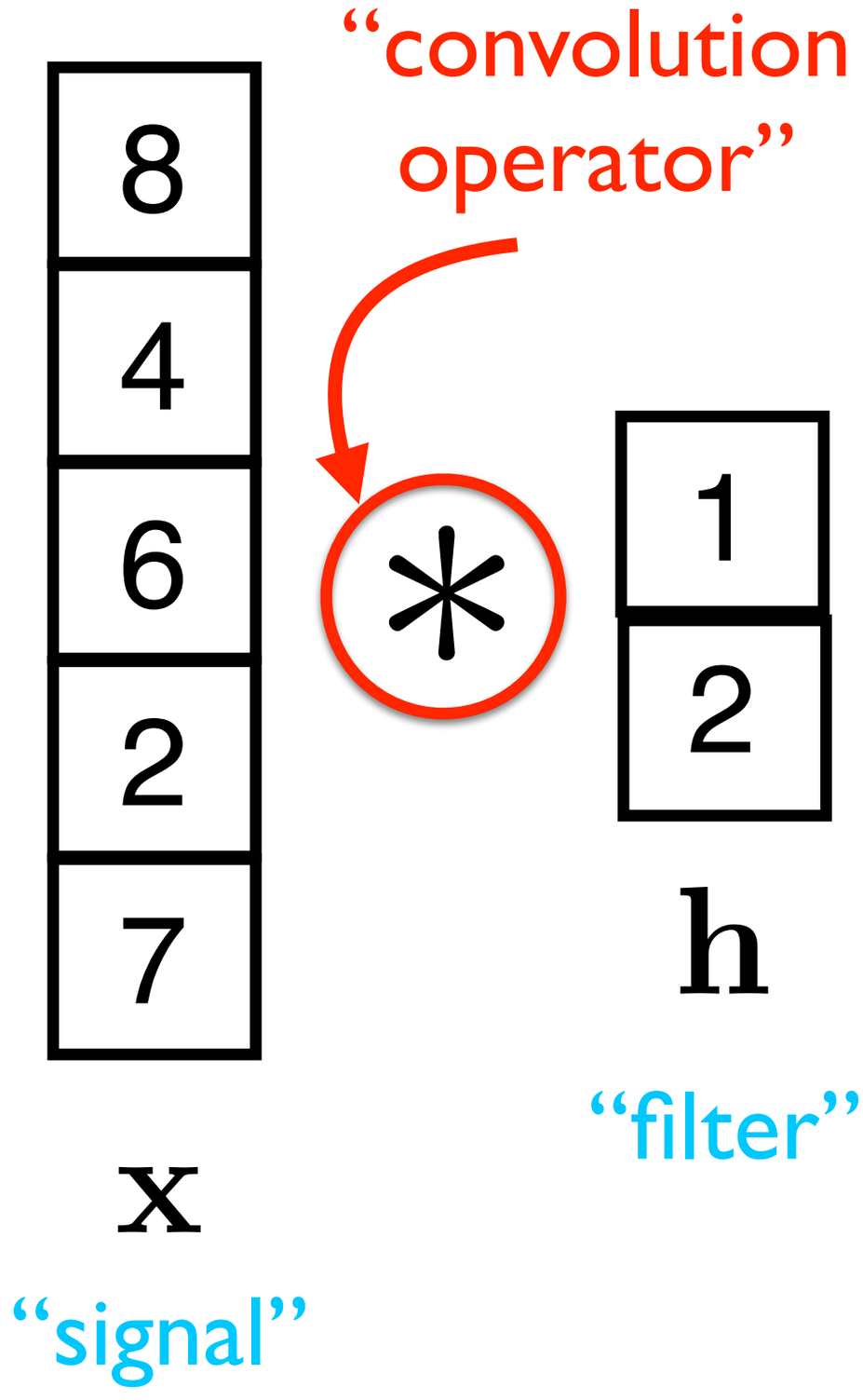
Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks

Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks

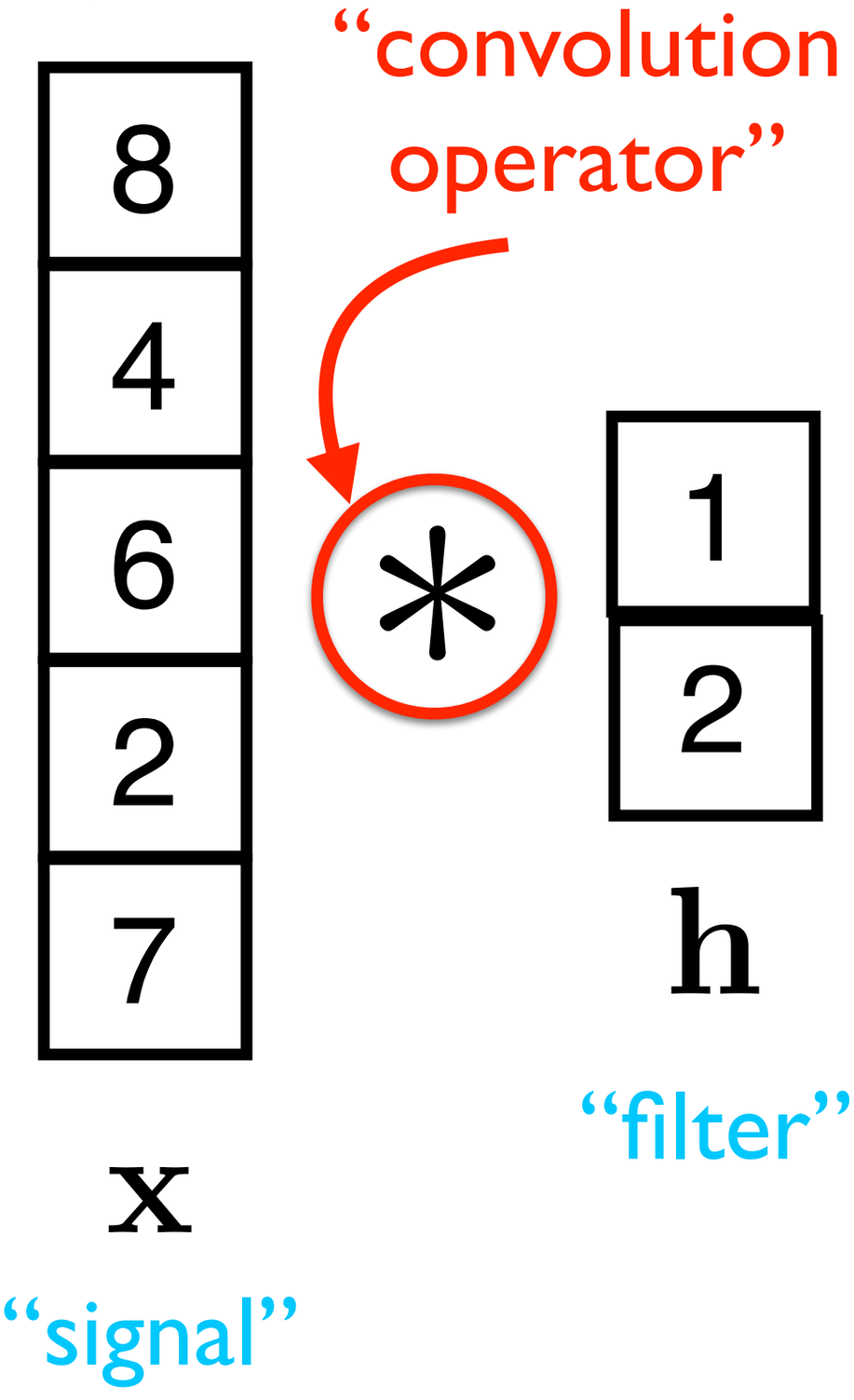
Reminder: Convolution



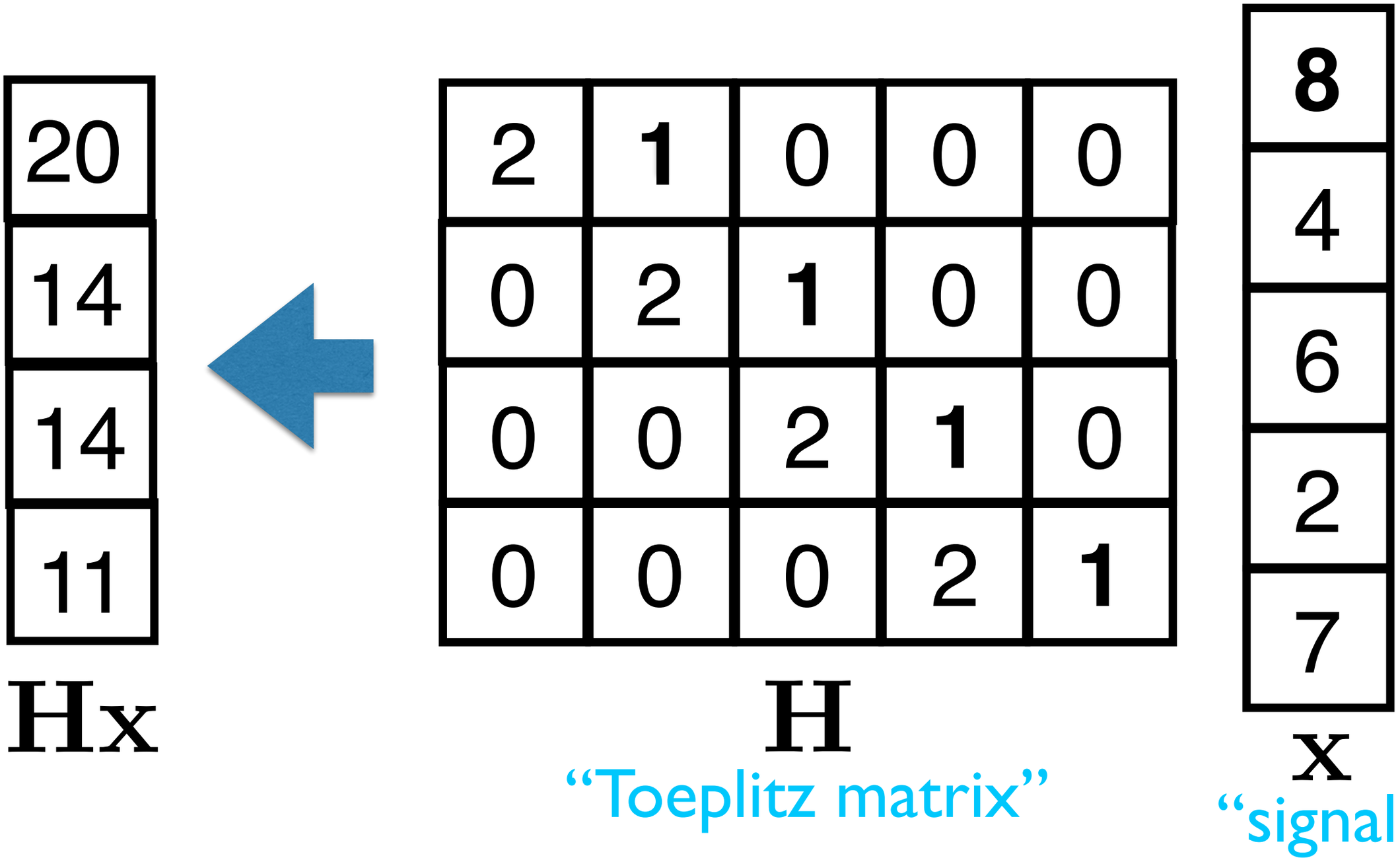
Reminder: Convolution

```
>>> from scipy.signal import \
    convolve as conv

>>> conv(x, h, 'valid')
array([20, 14, 14, 11])
```



Reminder: Convolution



Boundary conditions

- More than one type of convolutional operator:

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

Boundary conditions

- More than one type of convolutional operator:

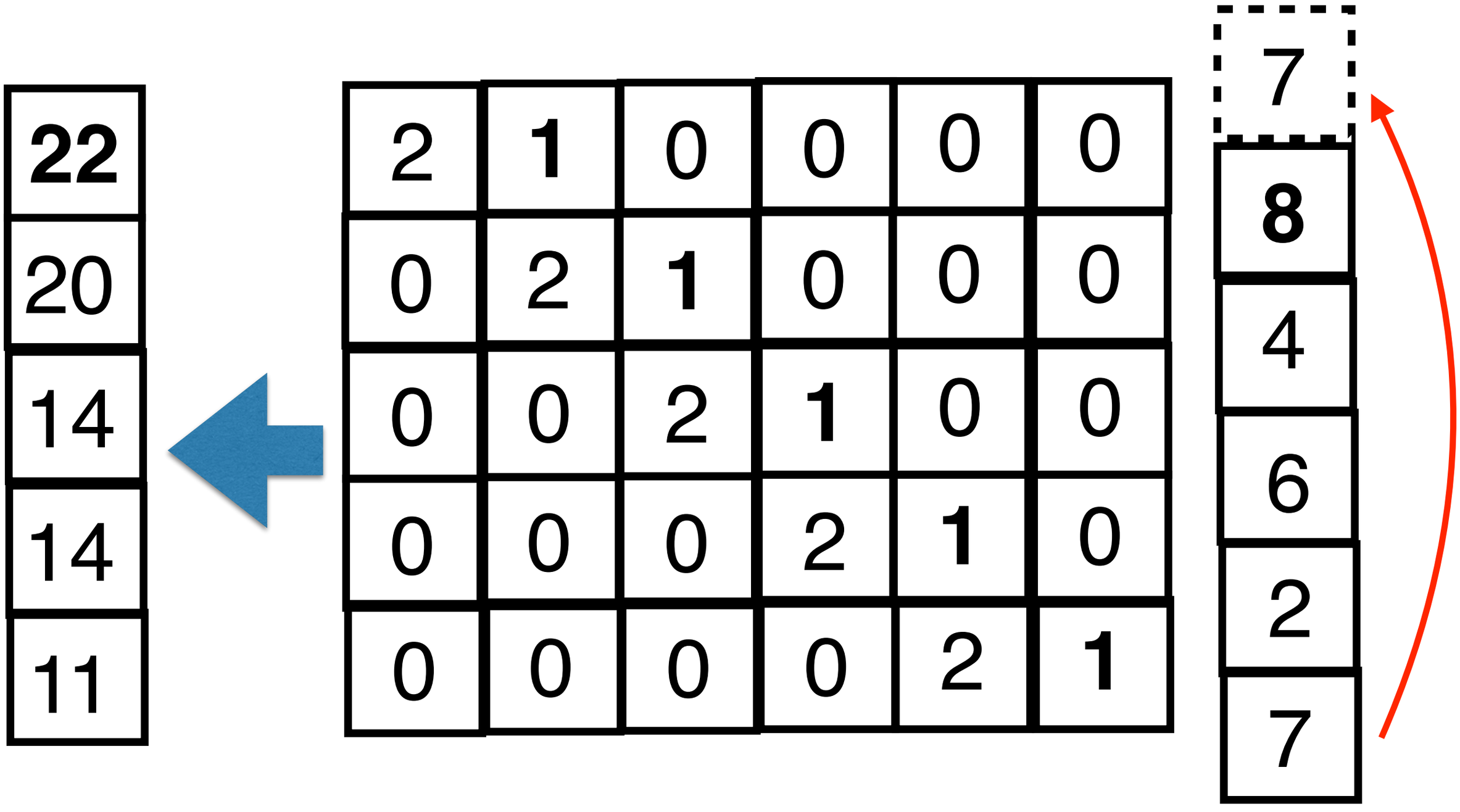
```
>>> conv(x, h, 'valid')
>>> conv(x, h, 'same')
>>> conv(x, h, 'full')
```

- What does H look like for ‘full’ convolution?
- How about ‘same’?

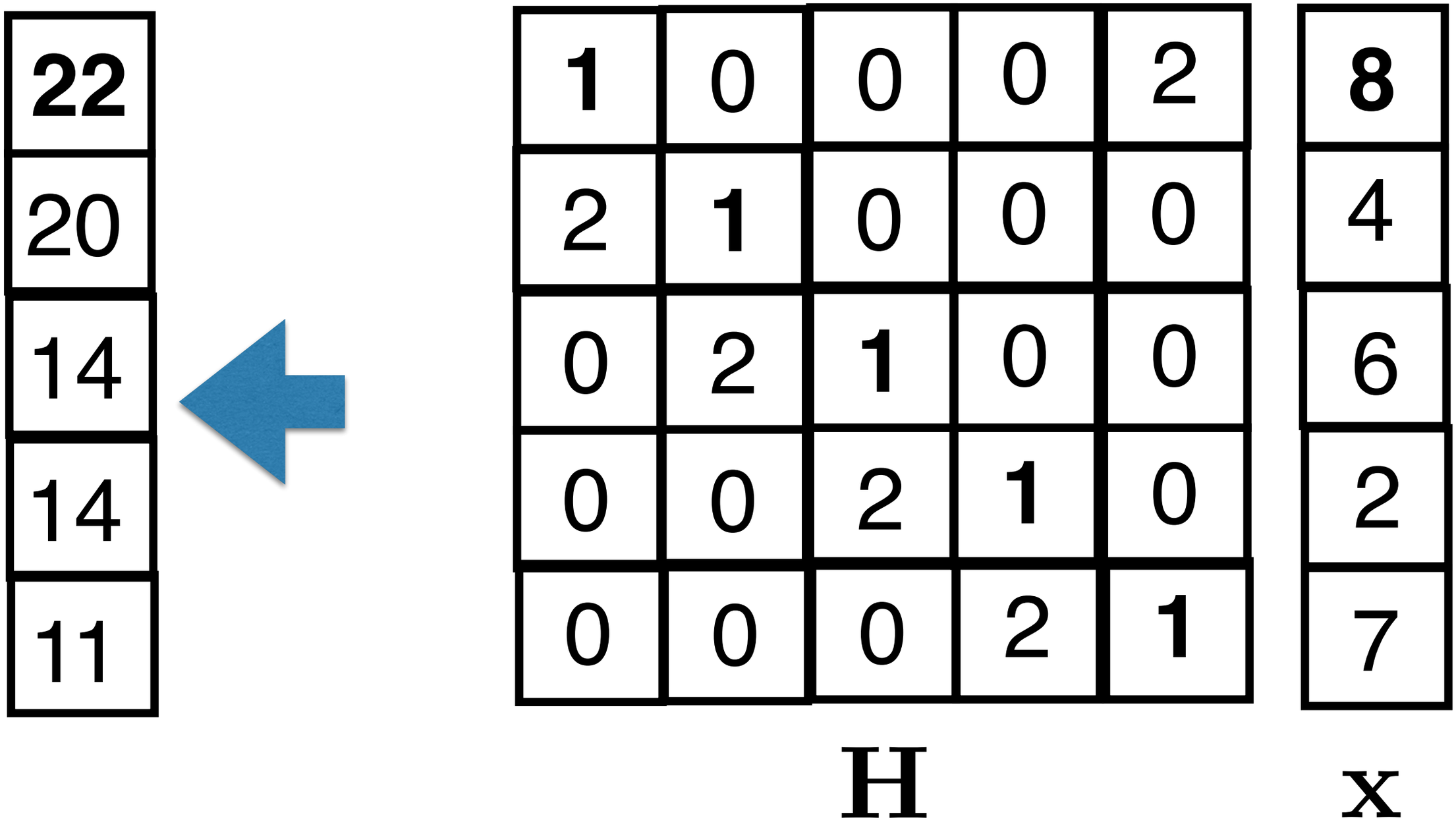
'valid'

2	1	0	0	0
0	2	1	0	0
0	0	2	1	0
0	0	0	2	1

Circular convolution

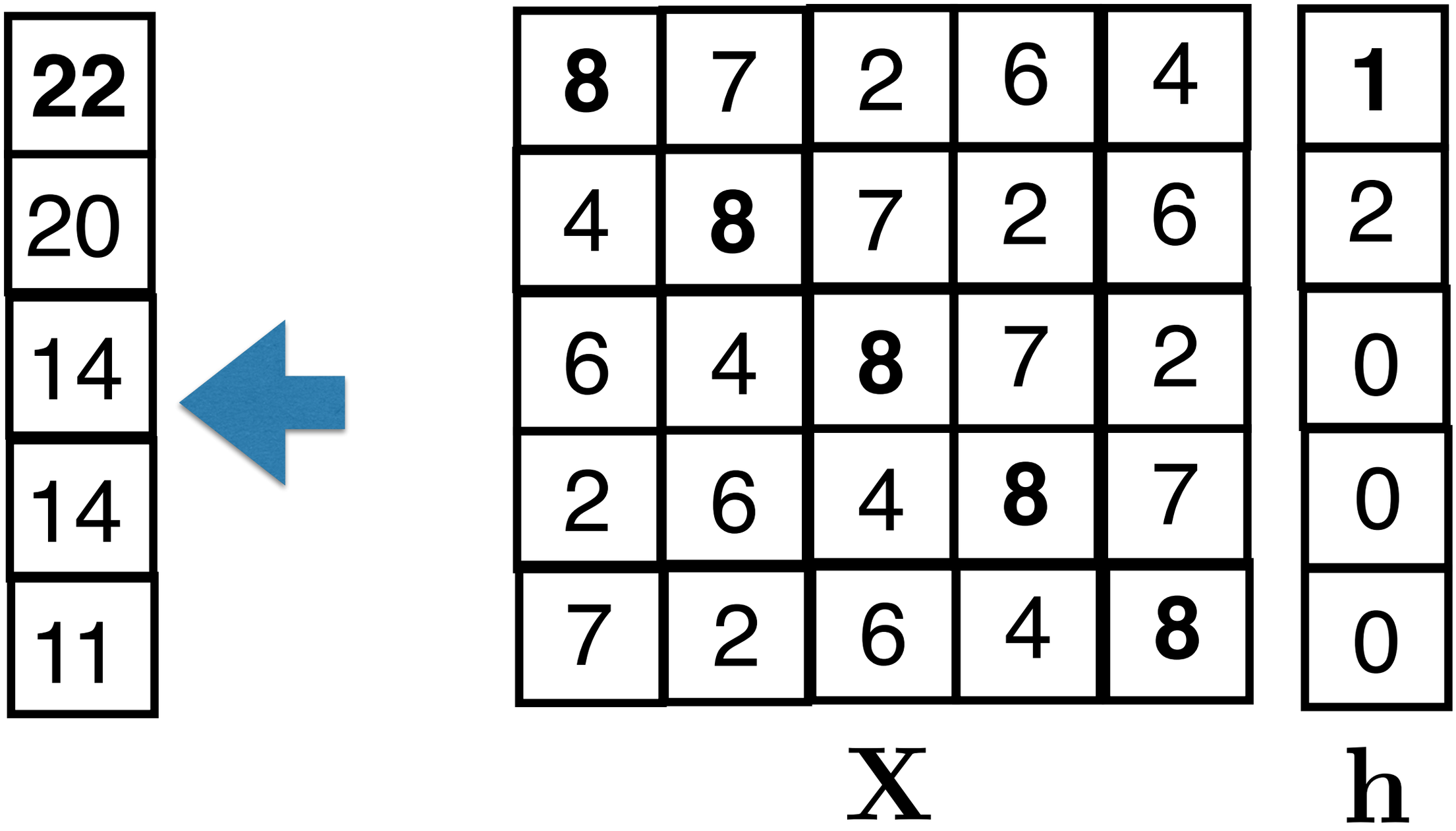


Circular convolution



"circulant matrix"

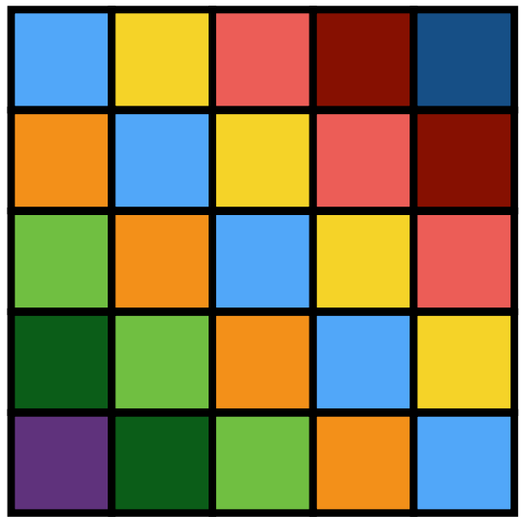
Circular convolution



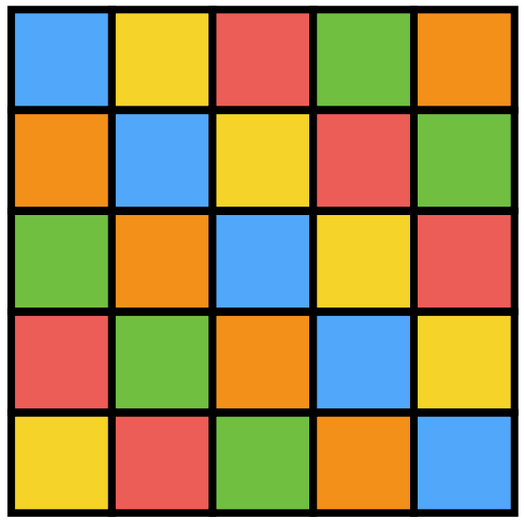
"circulant matrix"

Toeplitz and circulant matrices

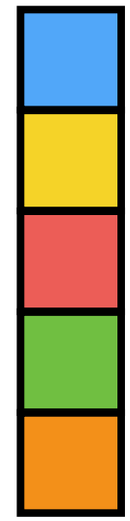
$(D \times D)$



$$X_{ij} = x[j - i]$$



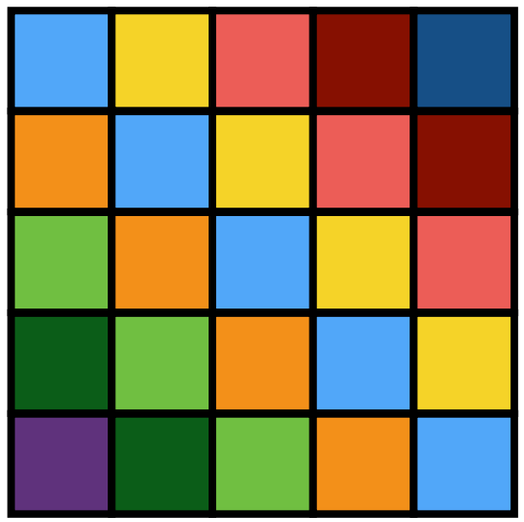
$$X_{ij} = x[j - i \bmod D]$$

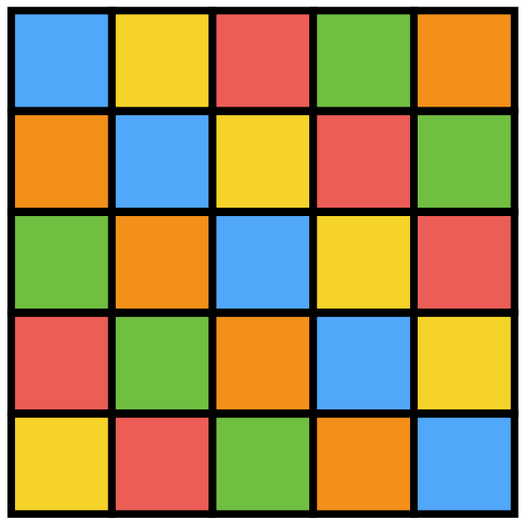


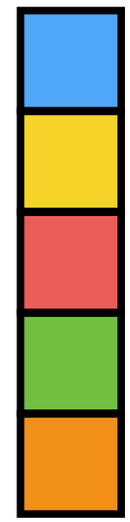
$$\mathbf{x} \in \mathbb{R}^D$$

Toeplitz and circulant matrices

$(D \times D)$







$X_{ij} = x[j - i]$

$X_{ij} = x[j - i \bmod D]$

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

```
>>> X[m,n] = x[mod(n-m, D)]
```

Discrete Fourier Transform

- Fourier transform

$$\hat{x}(\xi) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\xi t} dt$$

Discrete Fourier Transform

- Fourier transform

$$\hat{x}(\xi) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\xi t} dt$$

- Discrete Fourier Transform (DFT)

$$\hat{x}[k] = \sum_{t=0}^{D-1} x[t]\, e^{-i2\pi \frac{kt}{D}}$$

Discrete Fourier Transform

- Fourier transform

$$\hat{x}(\xi) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\xi t} dt$$

- Discrete Fourier Transform (DFT)

$$\hat{x}[k] = \sum_{t=0}^{D-1} x[t]\, e^{-i2\pi \frac{kt}{D}}$$

$$\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$$

$$\mathbf{F} = \left(e^{-i2\pi \frac{kt}{D}} \right)_{k,t}$$

Discrete Fourier Transform

- Fourier transform

$$\hat{x}(\xi) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi\xi t} dt$$

- Discrete Fourier Transform (DFT)

$$\hat{x}[k] = \sum_{t=0}^{D-1} x[t]\, e^{-i2\pi \frac{kt}{D}}$$

$$\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$$

$$\mathbf{F} = \left(e^{-i2\pi \frac{kt}{D}} \right)_{k,t}$$

Just a linear transform!

Convolution theorem

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

	Time	Frequency	Conv. thm.
Fourier transform	continuous; x(t)	continuous; X(ξ)	“linear”
Fourier series	continuous; x(t)	discrete; X_k	periodic
DTFT	discrete; x[t]	continuous; X(ξ)	“linear”
DFT	discrete; x[t]	discrete; X[k]	periodic

Isomorphism

Convolution

$$y = x * a$$

Time reversal $(\mathrm{mod}\ D)$

$$y = Jx$$
$$Jx[t] \triangleq x[-t]$$

Real

$$\mathrm{Im}\{x\} = 0$$

Product

$$\hat{y} = \hat{x} \circ \hat{a}$$

Conjugate

$$\hat{y} = \hat{x}^*$$

Conj. symm. $(\mathrm{mod}\ D)$

$$\hat{x}[t] = \hat{x}^*[-t]$$

Note: DFT is complex

27.0
4.93
1.57
1.57
4.93

+ i

00.0
0.50
5.57
-5.57
-0.50

$\hat{x}$

Fx

8
4
6
2
7

x

Example: Time reversal

27.0		00.0		8
4.93		0.50		7
1.57	− i	5.57		2
1.57		-5.57		6
4.93		-0.50		4

$\mathbf{Fx}$

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

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

Isomorphism

Cross-correlation

$$y = x \otimes a = Jx * a$$

Conjugate product

$$\hat{y} = \hat{x}^* \circ \hat{a}$$

Isomorphism

Cross-correlation

$$y = x \otimes a = Jx * a$$

Conjugate product

$$\hat{y} = \hat{x}^* \circ \hat{a}$$

Question: How is x⊗a related to a⊗x?

Isomorphism

Cross-correlation

$$y = x \otimes a = Jx * a$$

Conjugate product

$$\hat{y} = \hat{x}^* \circ \hat{a}$$

Question: How is x⊗a related to a⊗x?

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

Isomorphism

Cross-correlation

$$y = x \otimes a = Jx * a$$

Conjugate product

$$\hat{y} = \hat{x}^* \circ \hat{a}$$

Question: How is x⊗a related to a⊗x?

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

Isomorphism

Cross-correlation

$$y = x \otimes a = Jx * a$$

Conjugate product

$$\hat{y} = \hat{x}^* \circ \hat{a}$$

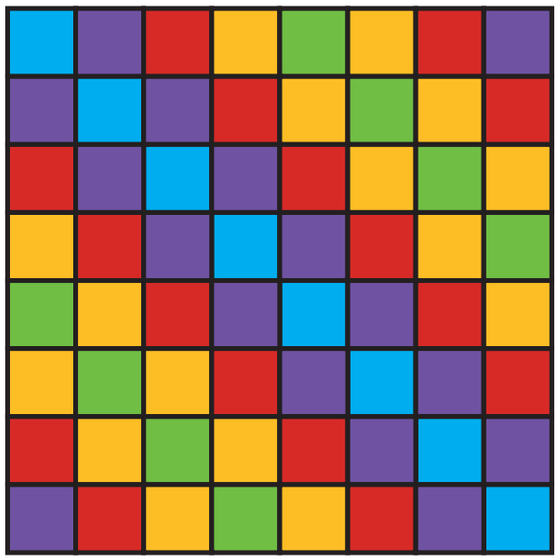
Question: How is x⊗a related to a⊗x?

$$\mathcal{F}(a \otimes x) = \hat{a}^* \circ \hat{x} = (\hat{a} \circ \hat{x}^*)^* = \mathcal{F}(J(x \otimes a))$$

Diagonalising circulant matrices

$$\mathbf{A}\mathbf{x} = \mathbf{a} * \mathbf{x} = \mathbf{F}^{-1}\mathbf{F}(\mathbf{a} * \mathbf{x})$$

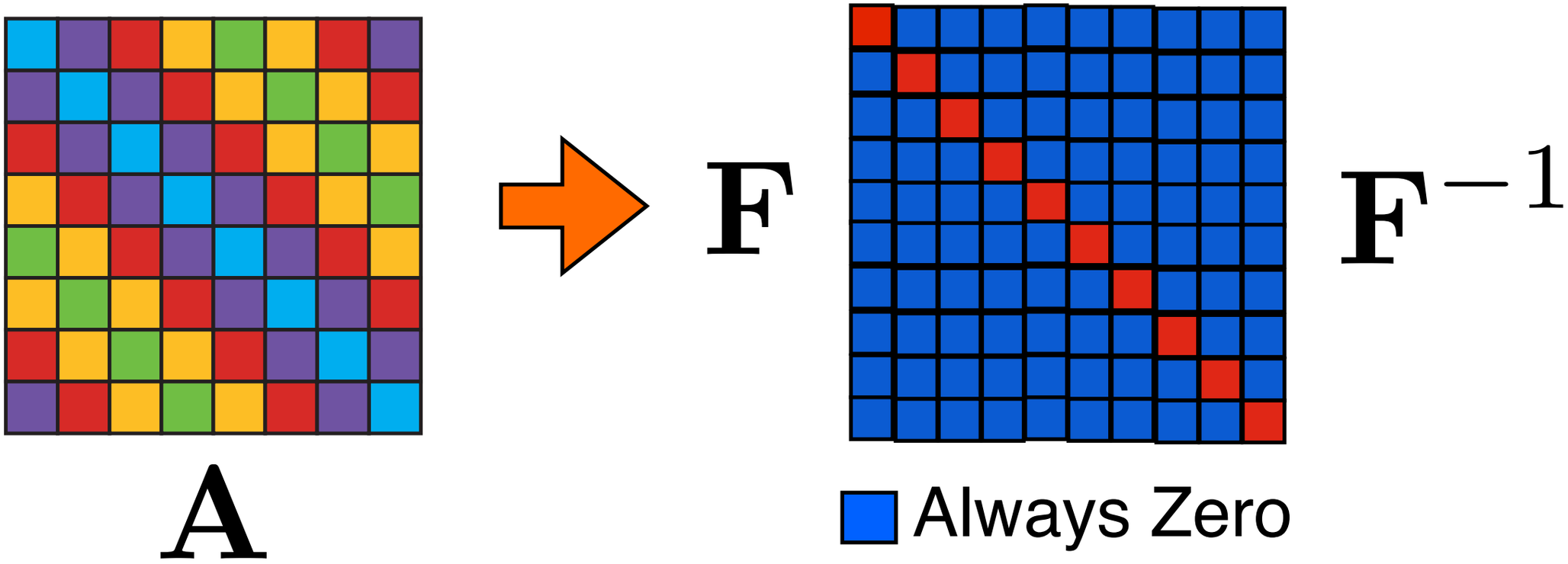
$$= \mathbf{F}^{-1}(\hat{\mathbf{a}} \circ \hat{\mathbf{x}}) = \mathbf{F}^{-1}\,\mathrm{diag}(\hat{\mathbf{a}})\mathbf{F}\mathbf{x}$$



$\mathbf{A}$

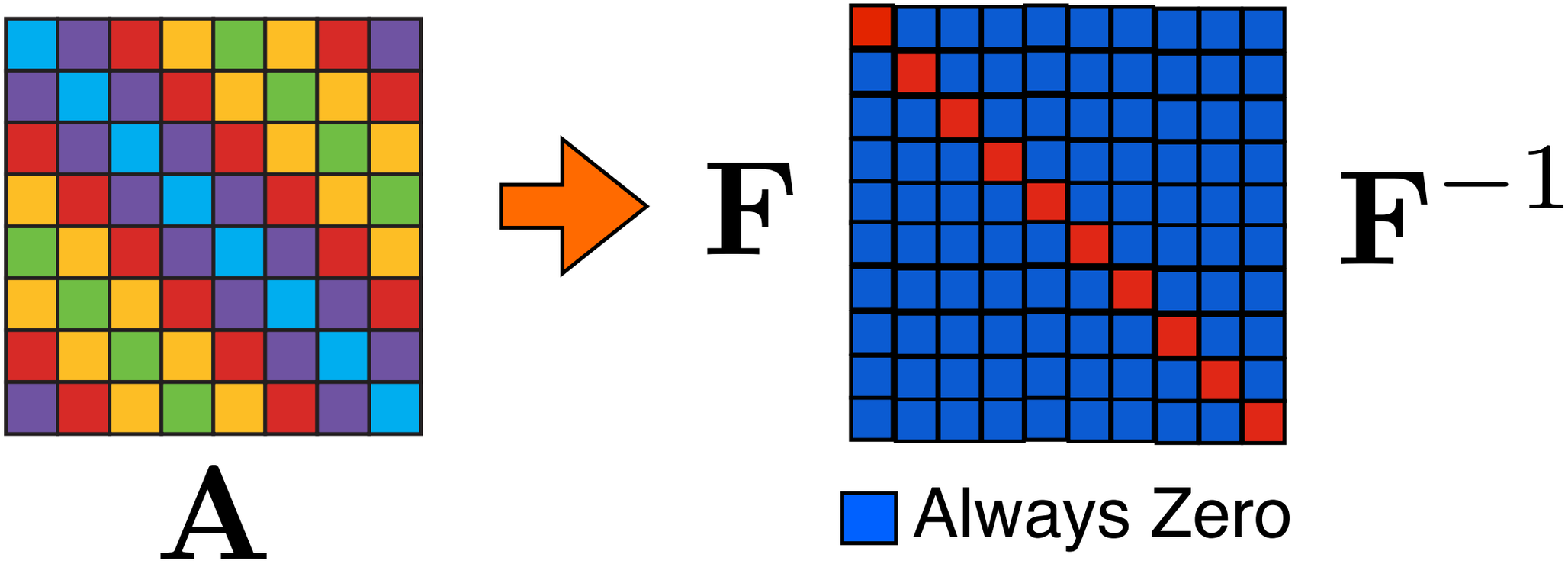
Diagonalising circulant matrices

$$\mathbf{A}\mathbf{x} = \mathbf{a} * \mathbf{x} = \mathbf{F}^{-1}\mathbf{F}(\mathbf{a} * \mathbf{x})$$
$$= \mathbf{F}^{-1}(\hat{\mathbf{a}} \circ \hat{\mathbf{x}}) = \mathbf{F}^{-1}\,\mathrm{diag}(\hat{\mathbf{a}})\mathbf{F}\mathbf{x}$$



Diagonalising circulant matrices

$$\mathbf{A}\mathbf{x} = \mathbf{a} * \mathbf{x} = \mathbf{F}^{-1}\mathbf{F}(\mathbf{a} * \mathbf{x})$$
$$= \mathbf{F}^{-1}(\hat{\mathbf{a}} \circ \hat{\mathbf{x}}) = \mathbf{F}^{-1}\,\mathrm{diag}(\hat{\mathbf{a}})\mathbf{F}\mathbf{x}$$



All circulant matrices have the *same eigenvectors*!
Eigen*values* given by Fourier transform

Diagonalising circulant matrices

Question: What if A is real and symmetric?
Shouldn't its eigendecomposition be real?

Diagonalising circulant matrices

Question: What if A is real and symmetric? Shouldn't its eigendecomposition be real?

$$\mathbf{A} = \mathbf{A}^T \iff a[t] = a[-t]$$

Diagonalising circulant matrices

Question: What if A is real and symmetric? Shouldn't its eigendecomposition be real?

$$\mathbf{A} = \mathbf{A}^T \iff a[t] = a[-t]$$

$$\iff \hat{\mathbf{a}}^* = \hat{\mathbf{a}}$$

Diagonalising circulant matrices

Question: What if A is real and symmetric? Shouldn't its eigendecomposition be real?

$$\mathbf{A} = \mathbf{A}^T \iff a[t] = a[-t]$$

$$\iff \hat{\mathbf{a}}^* = \hat{\mathbf{a}} \quad \text{real (eigenvalues)}$$

Diagonalising circulant matrices

Question: What if A is real and symmetric? Shouldn't its eigendecomposition be real?

$$\mathbf{A} = \mathbf{A}^T \iff a[t] = a[-t]$$

$$\iff \hat{\mathbf{a}}^* = \hat{\mathbf{a}} \quad \text{real (eigenvalues)}$$

$$\mathbf{a} \text{ real} \iff \hat{a}[k] = \hat{a}^*[-k]$$

Diagonalising circulant matrices

Question: What if A is real and symmetric? Shouldn't its eigendecomposition be real?

$$\mathbf{A} = \mathbf{A}^T \iff a[t] = a[-t]$$

$$\iff \hat{\mathbf{a}}^* = \hat{\mathbf{a}} \quad \text{real (eigenvalues)}$$

$$\mathbf{a} \text{ real} \iff \hat{a}[k] = \hat{a}^*[-k] = \hat{a}[-k]$$

Diagonalising circulant matrices

Question: What if A is real and symmetric?
Shouldn't its eigendecomposition be real?

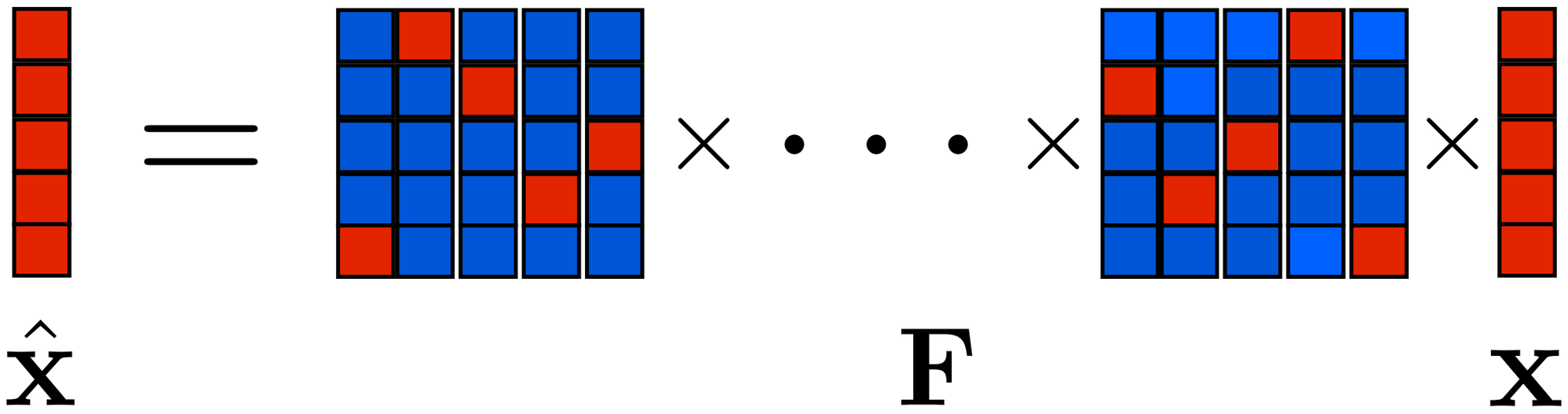
$$\mathbf{A} = \mathbf{A}^T \iff a[t] = a[-t]$$

$$\iff \hat{\mathbf{a}}^* = \hat{\mathbf{a}} \quad \text{real (eigenvalues)}$$

$$\mathbf{a} \text{ real} \iff \hat{a}[k] = \hat{a}^*[-k] \quad = \hat{a}[-k]$$

Eigenvalues k $\notin$ {0, D/2} have *multiplicity two*
$\Rightarrow$ Eigenvectors are not unique

Fast Fourier Transform



Carl Friedrich Gauss

Not Always Zero

Always Zero

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

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

FFT as a linear transform

- Using fftmtx.py (see link),

```
>>> F = fftmtx(5)
>>> x = randn(5); h = randn(5);
```

- Apply the FFT in the naive and fast methods.

```
>>> xf_a = dot(F,x)
>>> xf_b = fft(x)
```

- You should see that `xf_a` and `xf_b` are the same.

FFT as a linear transform

- Using fftmtx.py (see link),

```
>>> F = fftmtx(5)
>>> x = randn(5); h = randn(5);
```

- Apply the FFT in the naive and fast methods.

```
>>> xf_a = dot(F,x)
>>> xf_b = fft(x)
```

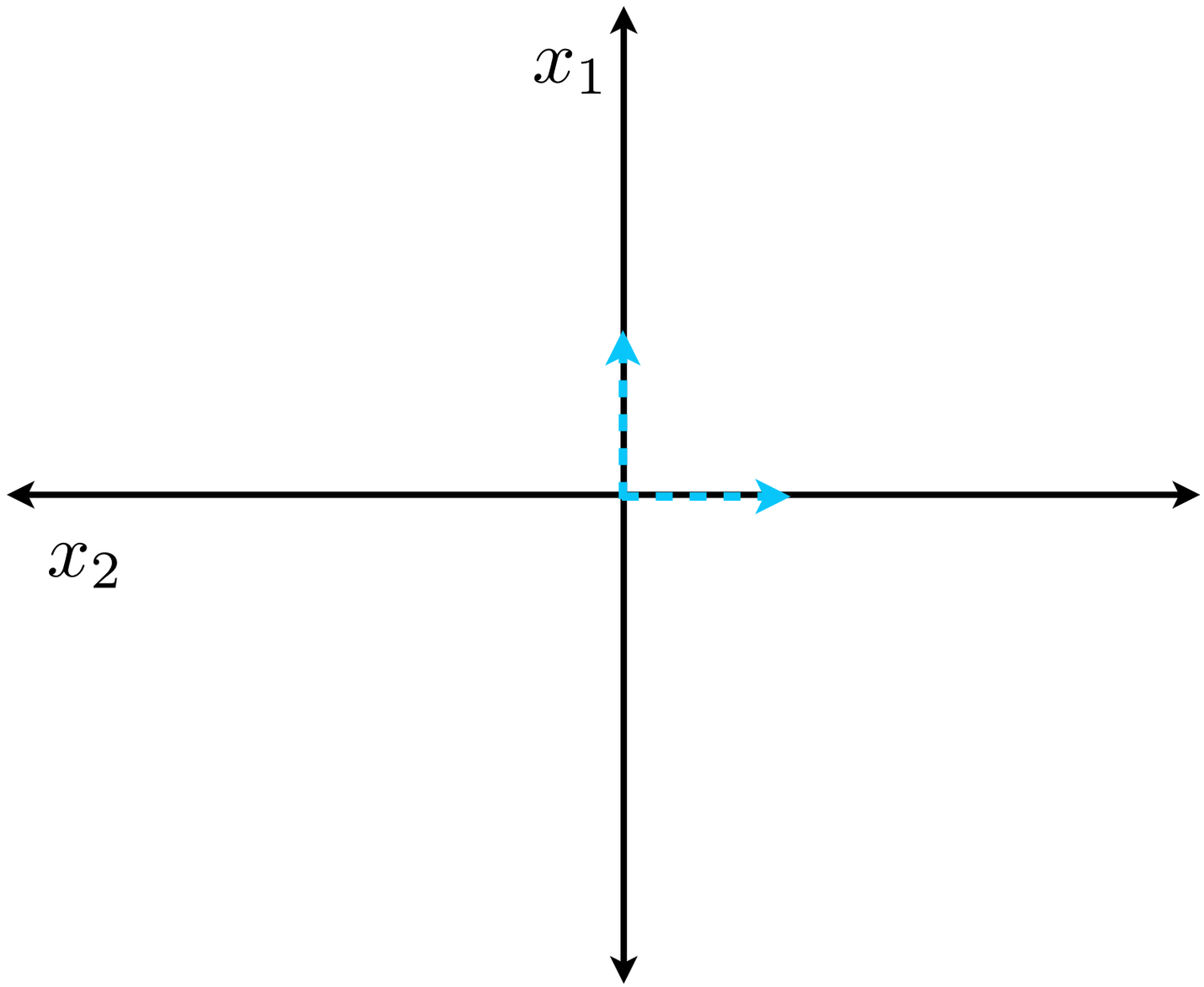
- You should see that `xf_a` and `xf_b` are the same.

Question: what does `F.H @ F` look like?

Parseval's Theorem

Antoine Parseval

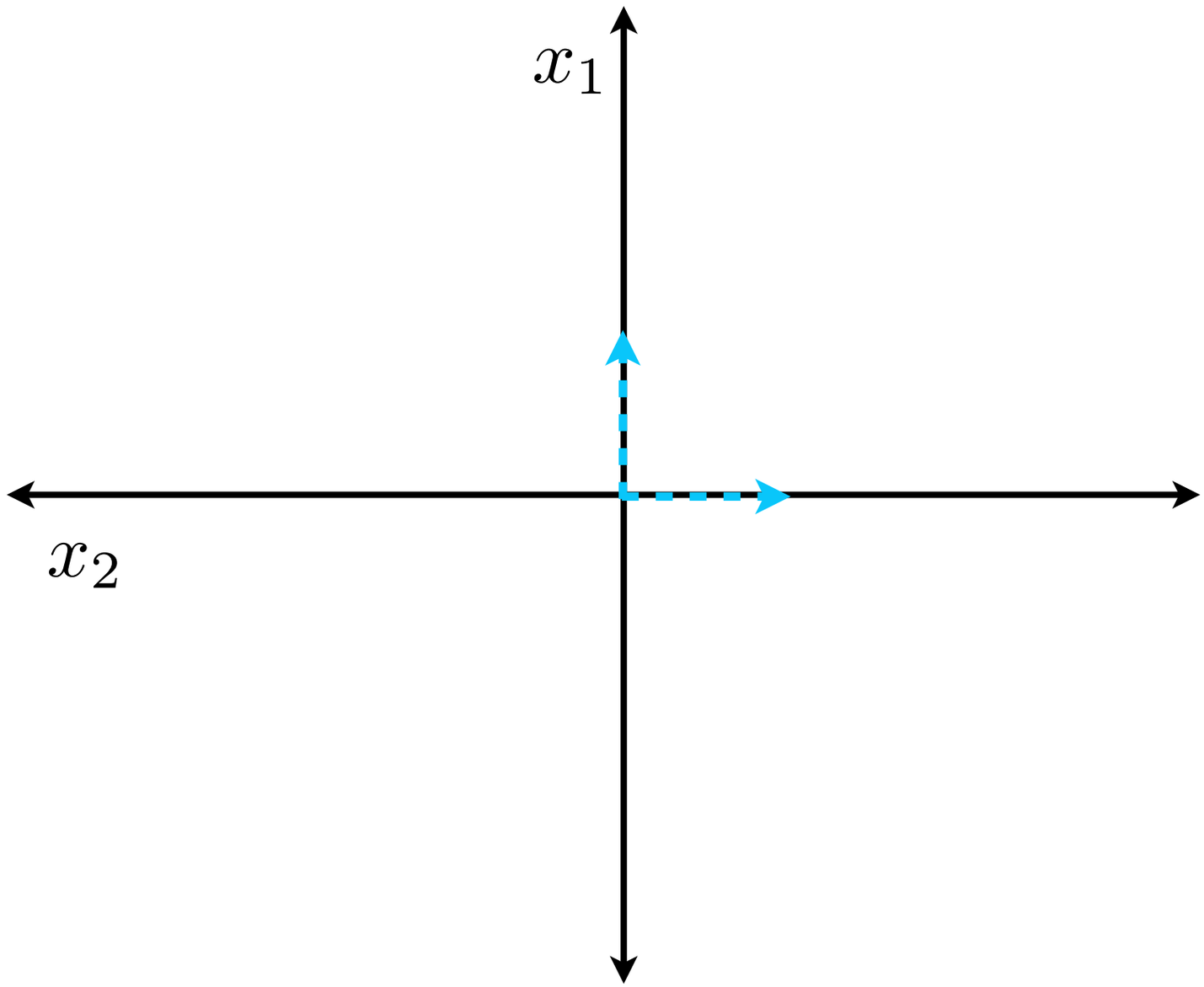
$$||\mathbf{x}||_2^2 \propto ||\hat{\mathbf{x}}||_2^2$$



Parseval's Theorem

Antoine Parseval

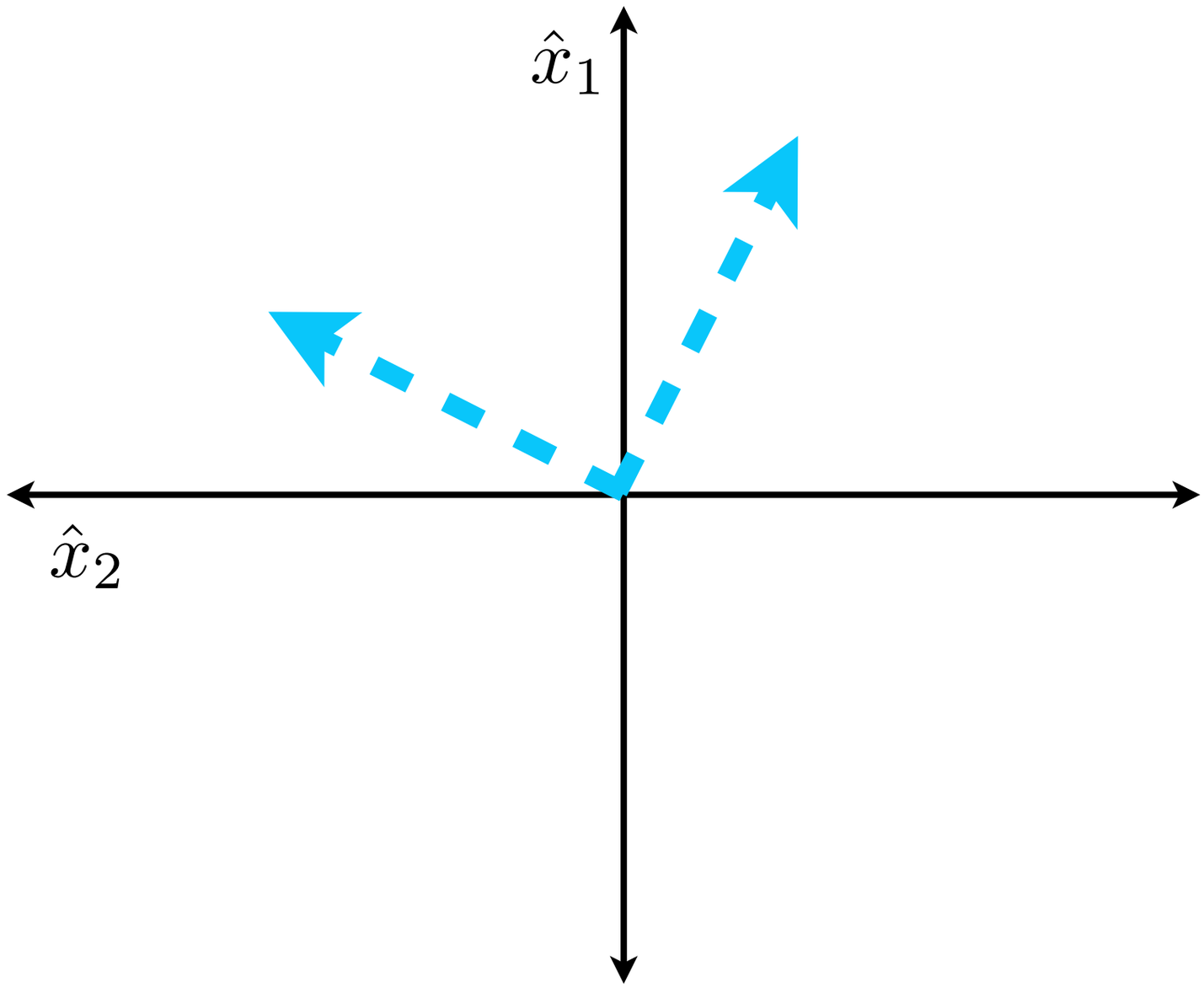
$$||\mathbf{x}||_2^2 = \frac{1}{D}||\hat{\mathbf{x}}||_2^2$$



Parseval's Theorem

Antoine Parseval

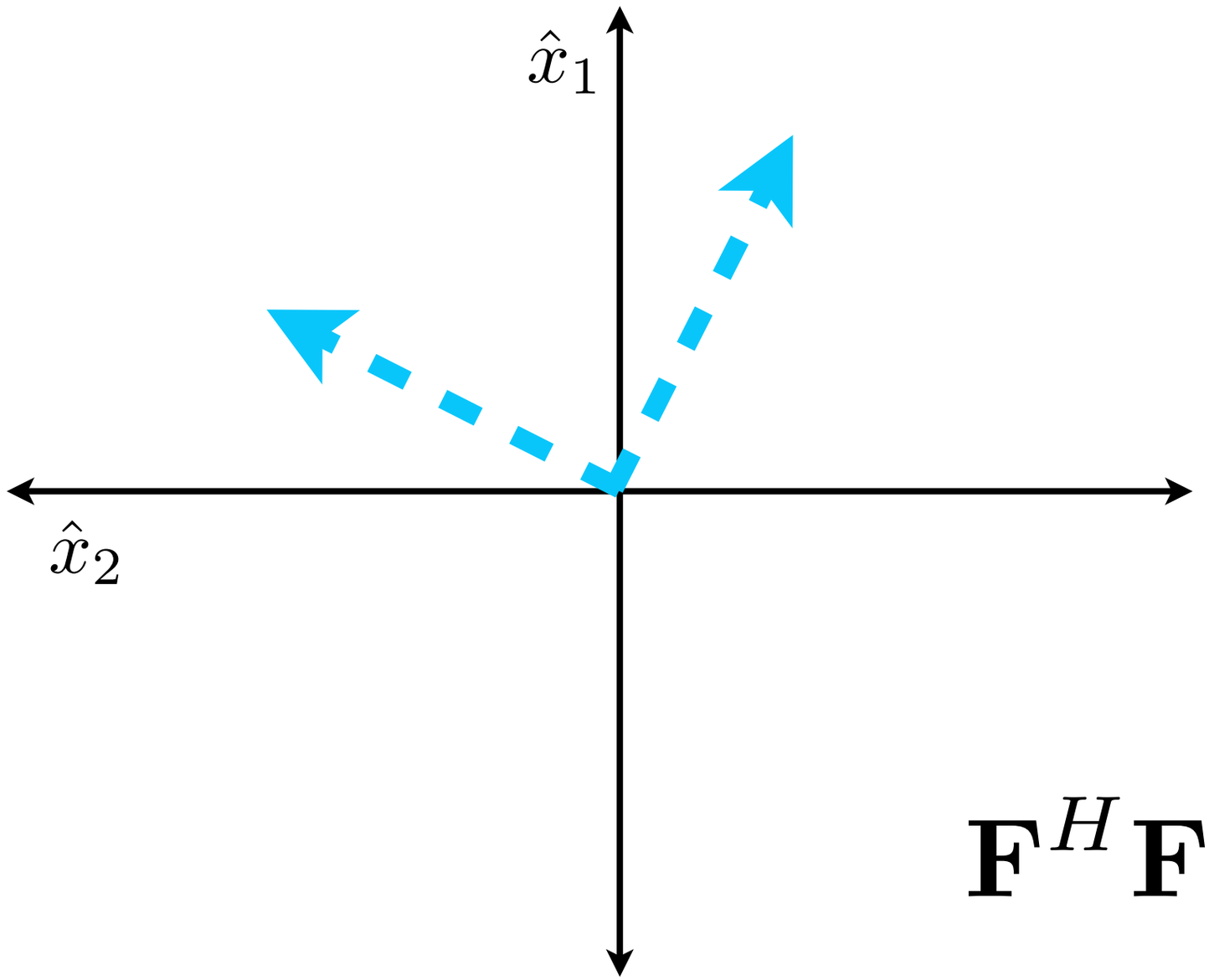
$$||\mathbf{x}||_2^2 = \frac{1}{D}||\hat{\mathbf{x}}||_2^2$$



Parseval's Theorem

Antoine Parseval

$$||\mathbf{x}||_2^2 = \frac{1}{D}||\hat{\mathbf{x}}||_2^2$$



$$\mathbf{F}^H\mathbf{F} = D\mathbf{I}$$

N-dim DFT

- Everything above applies for N-dim Fourier transform and N-dim convolution

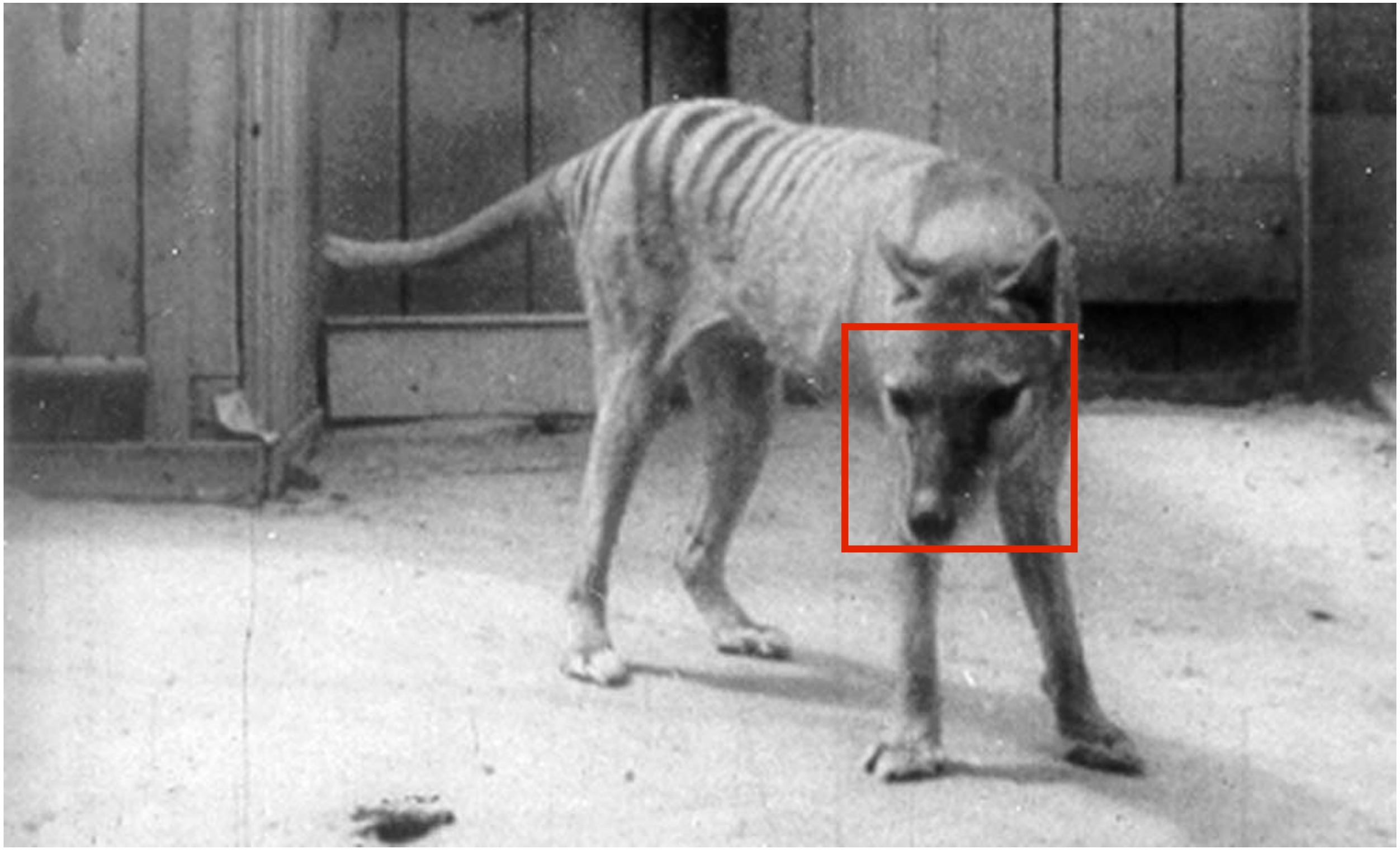
- Treat signals as real vectors in $\mathbb{R}^{[D_1]\times[D_2]\times\cdots}$

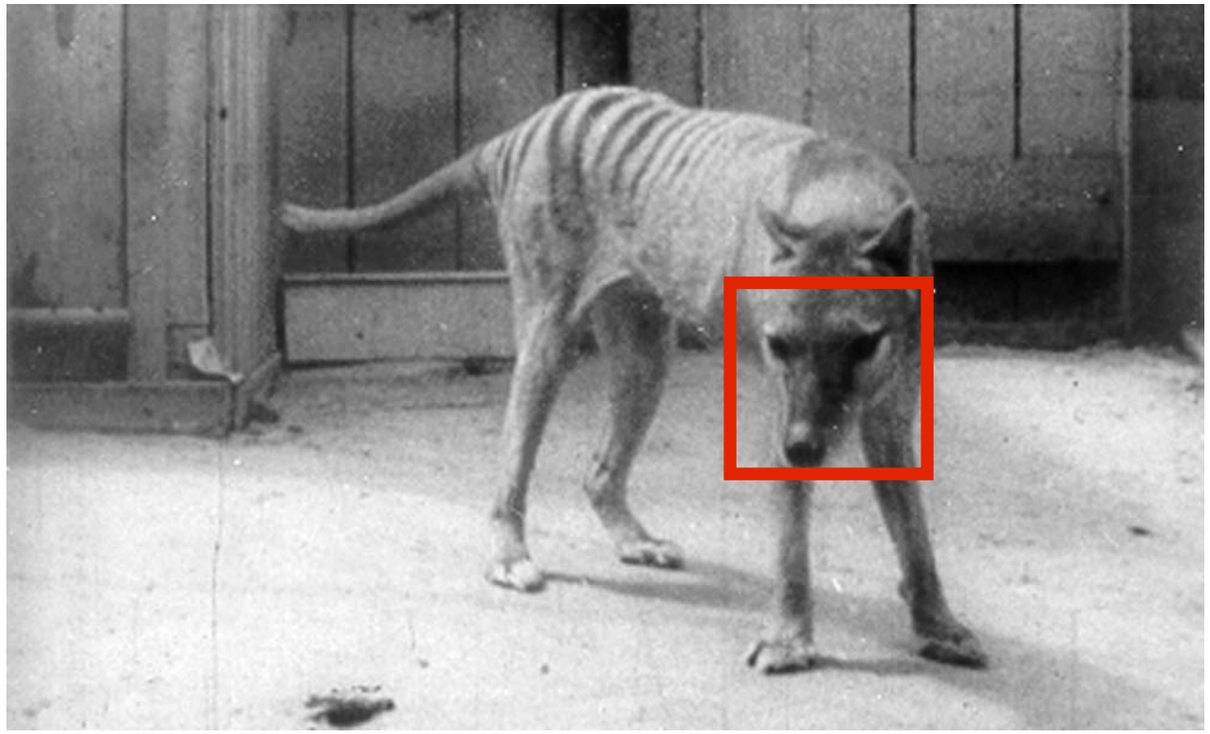
$$[D] = \{0, 1, \ldots, D-1\}$$

- N-dim FFT obtained independent 1-dim FFTs

Today

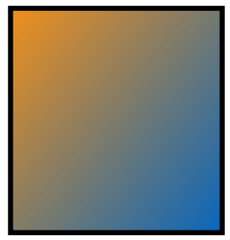
- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks





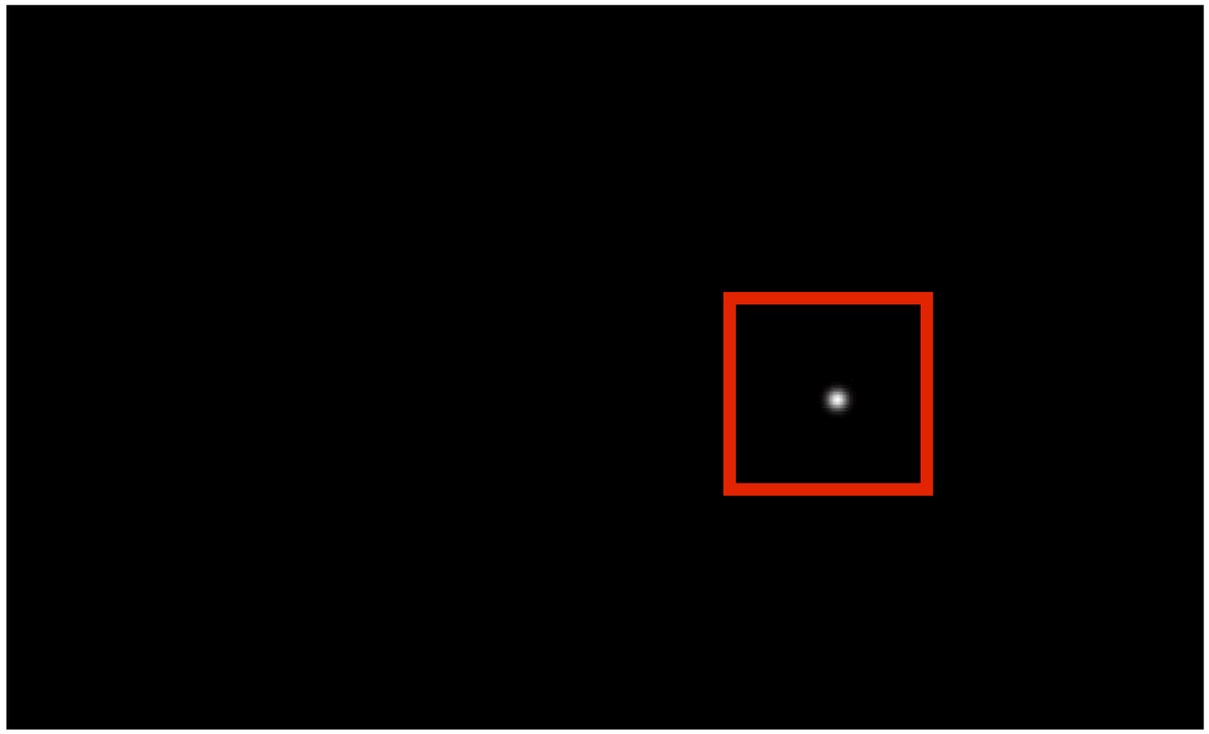
“known signal” $\mathbf{x}$

$\otimes$

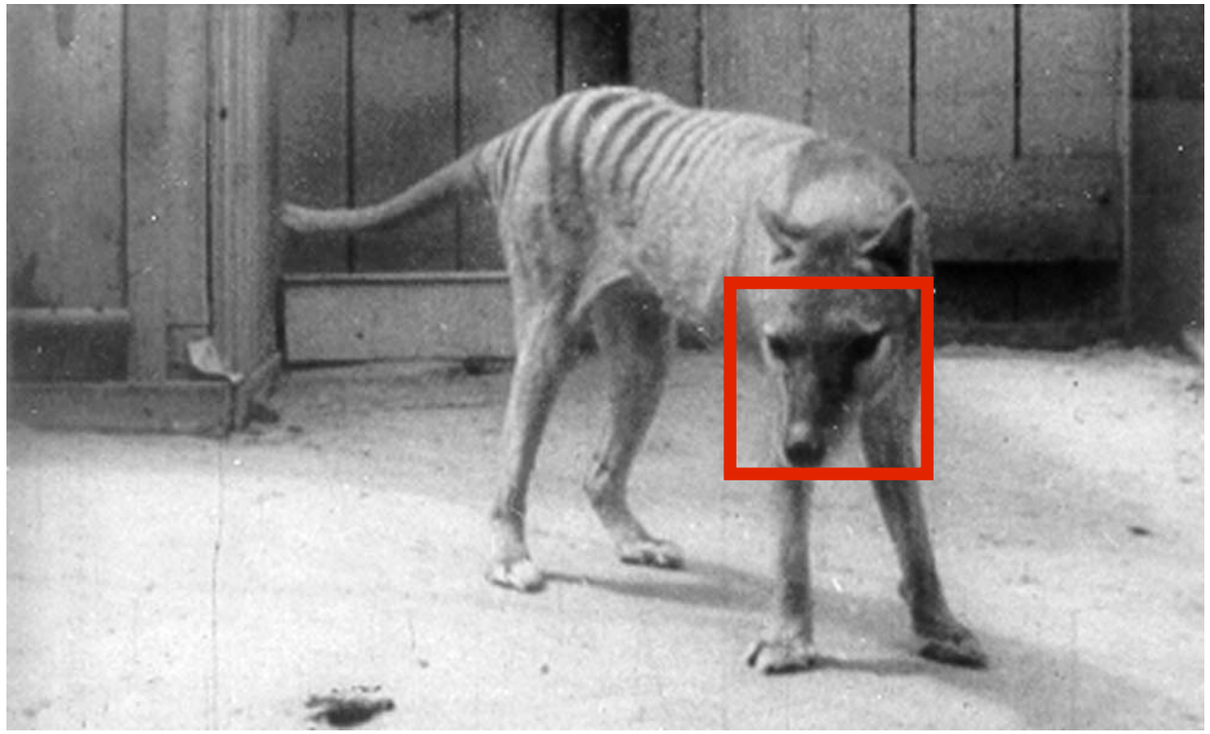


$\mathbf{g}$

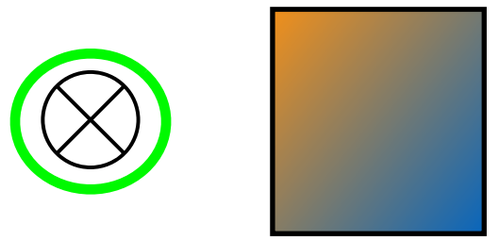
“unknown filter”



“known response” $\mathbf{y}$



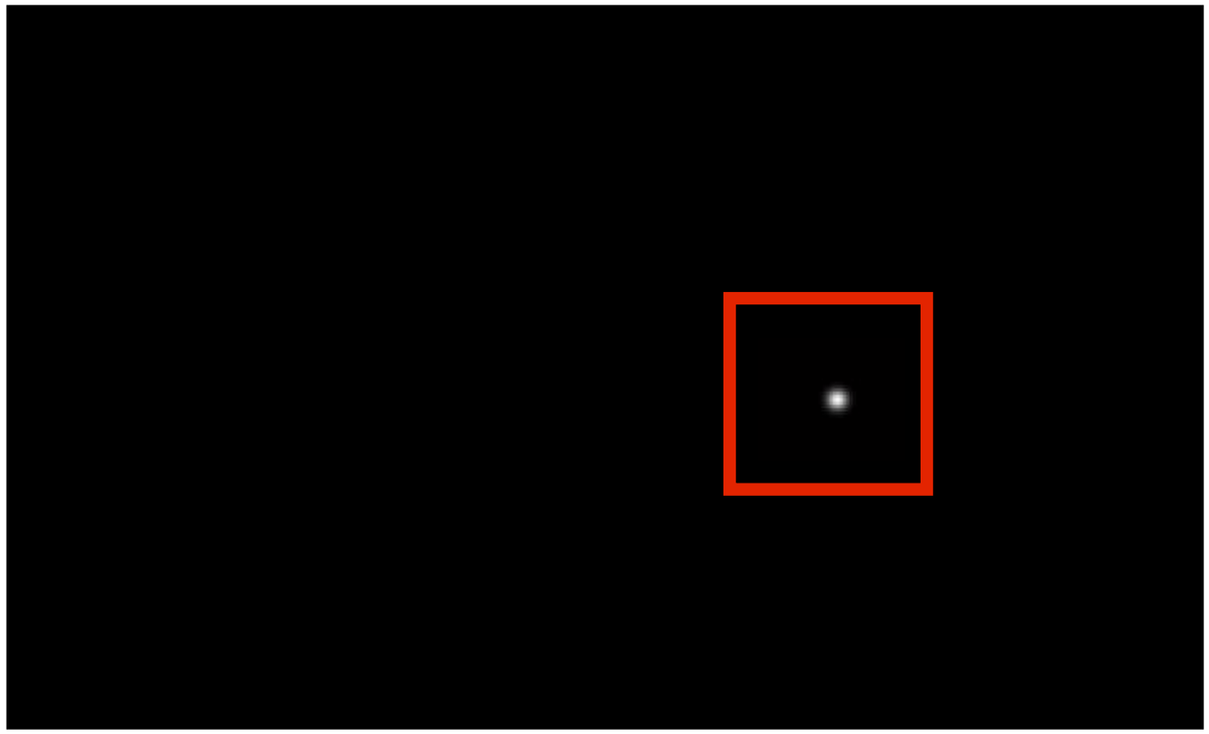
“known signal” $\mathbf{x}$



“unknown filter”

$\mathbf{g}$

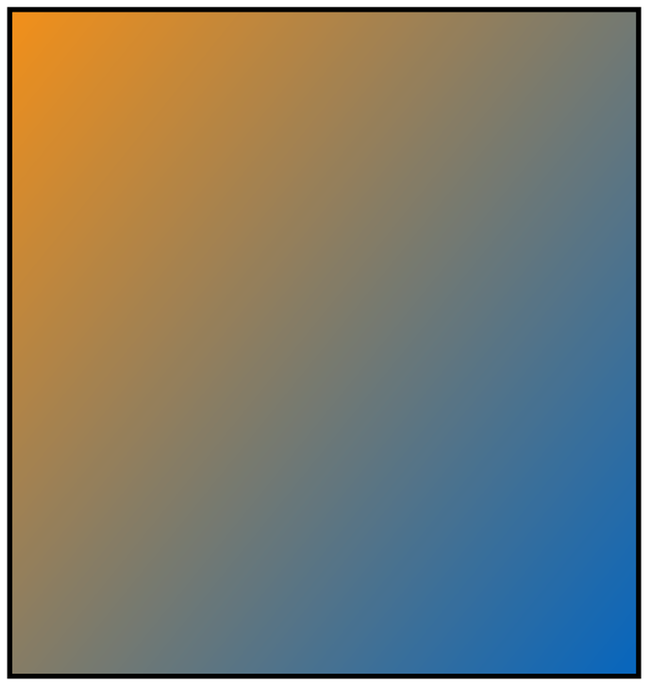
“correlation”



“known response” $\mathbf{y}$

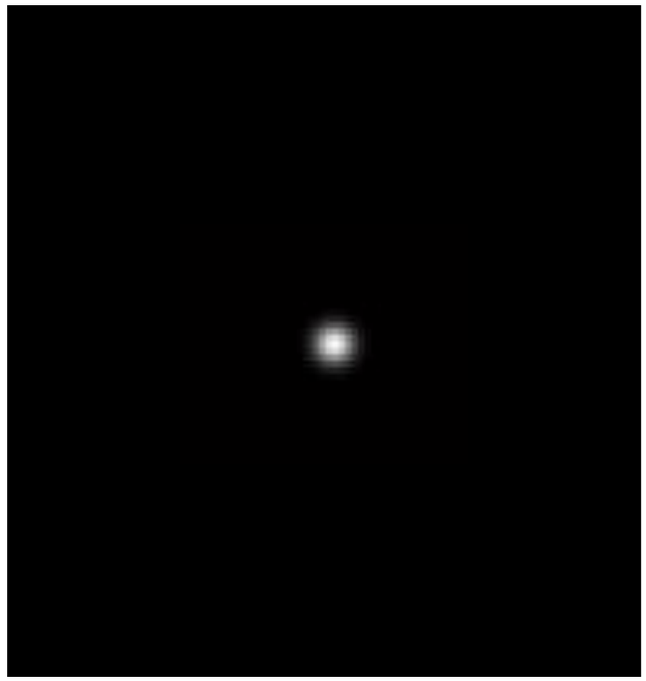
“known signal” $\mathbf{x}$

$\otimes$



“unknown filter” $\mathbf{g}$

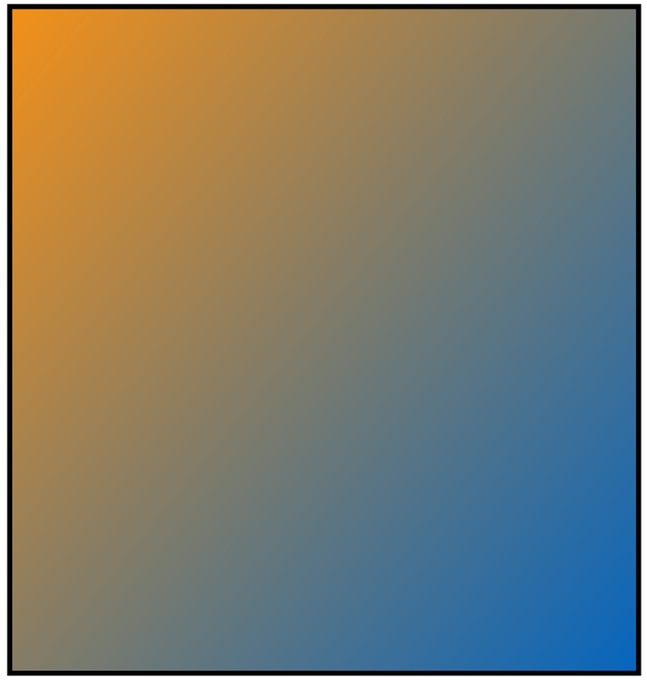
“circular correlation”



“known response” $\mathbf{y}$

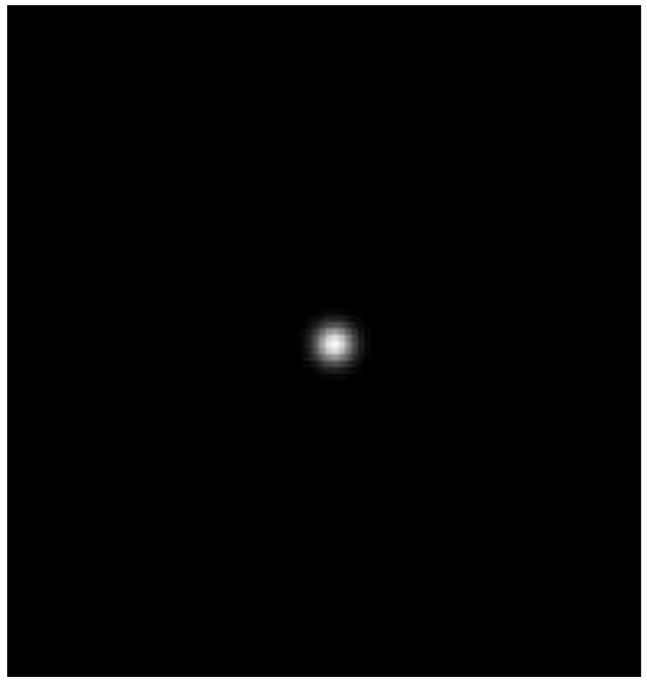
“known signal” $\mathbf{x}$

$\otimes$



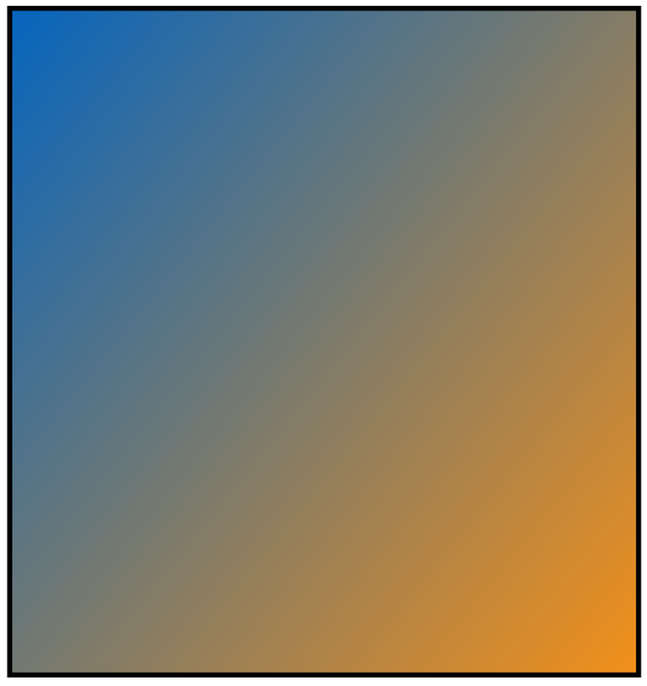
“unknown filter”

$\mathbf{g}$



“known response” $\mathbf{y}$

*

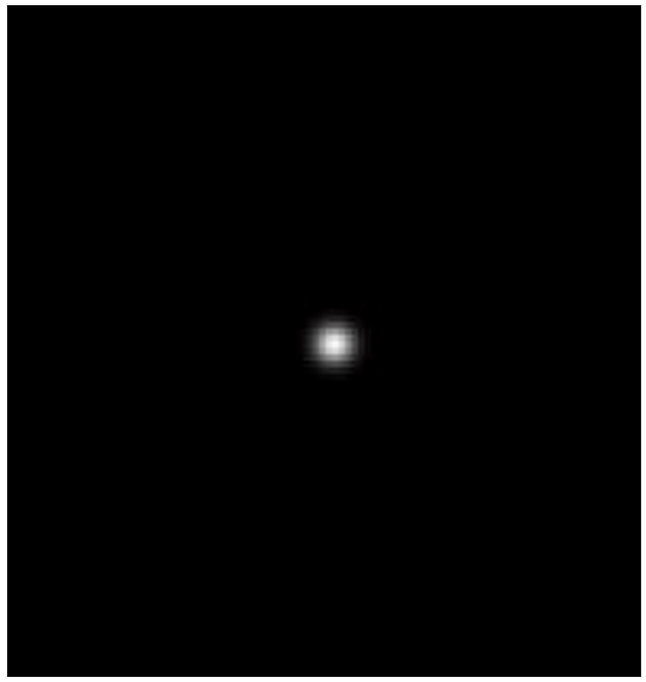


"unknown filter"

h

"known signal" **x**

```
g = flipud(fliplr(h))
```

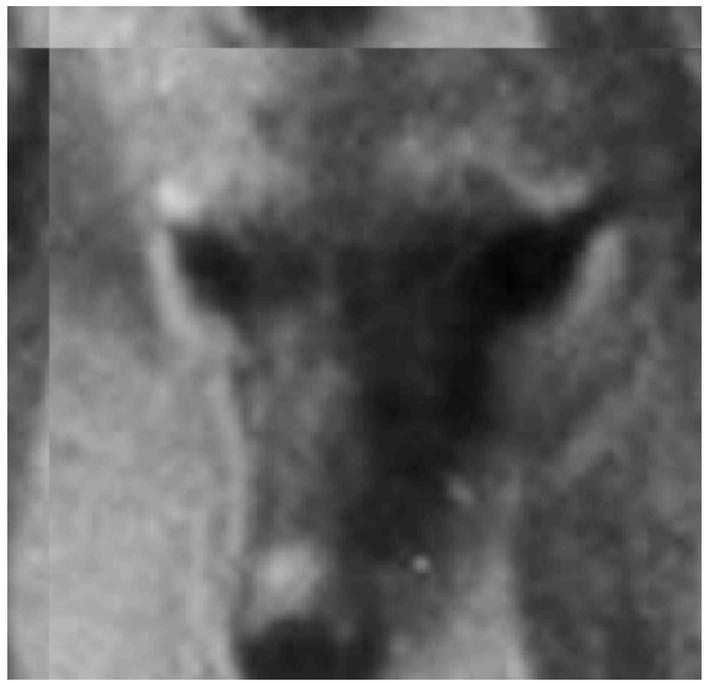


"known response" **y**

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

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

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



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

```
>>> xshift = circshift(x,(20,20))
```

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

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

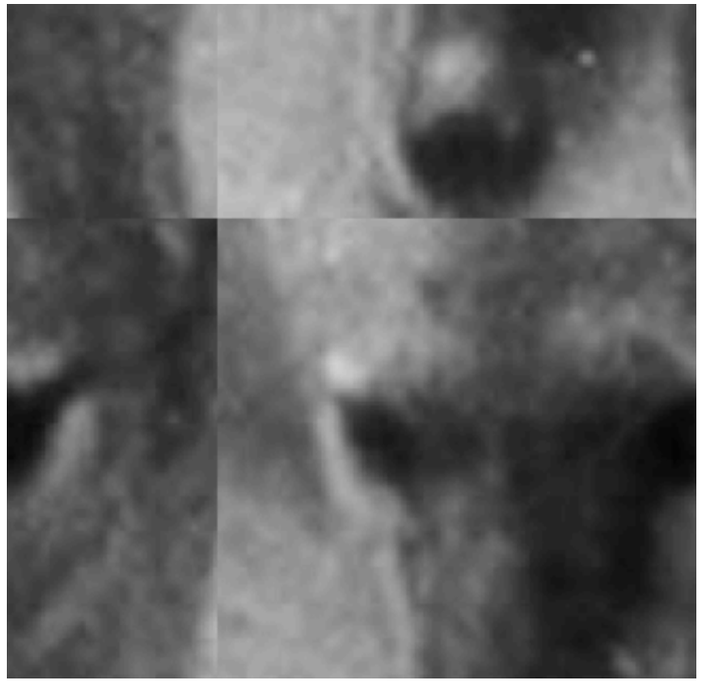
$\mathbf{x}[-20, -20]$

```
>>> xshift = circshift(x,(-20,-20))
```

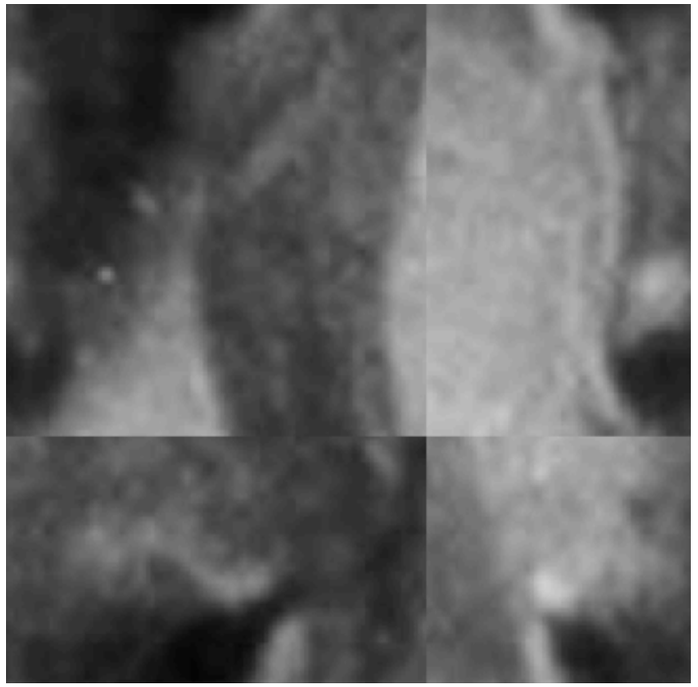
$\mathbf{x}[0, 0]$

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

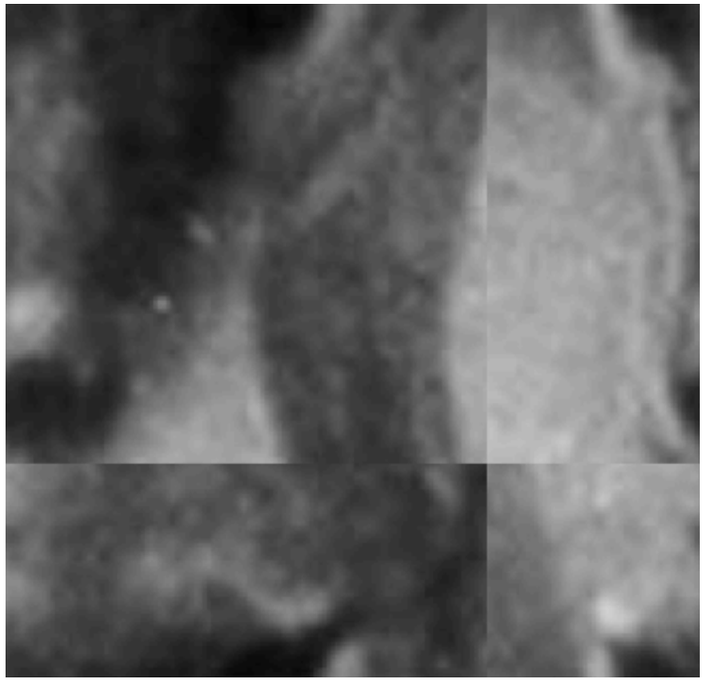
$\mathbf{x}[-20, -20]$



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



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

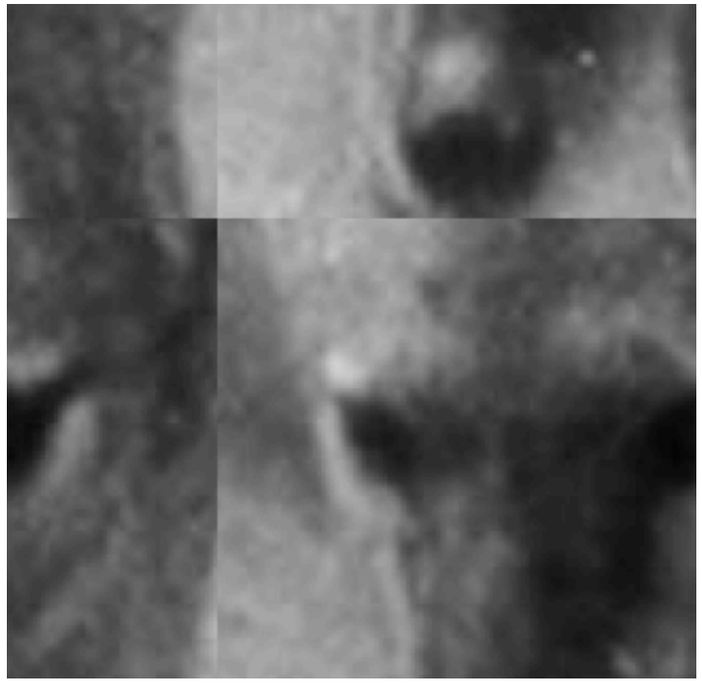


$\mathbf{x}[200, 200]$

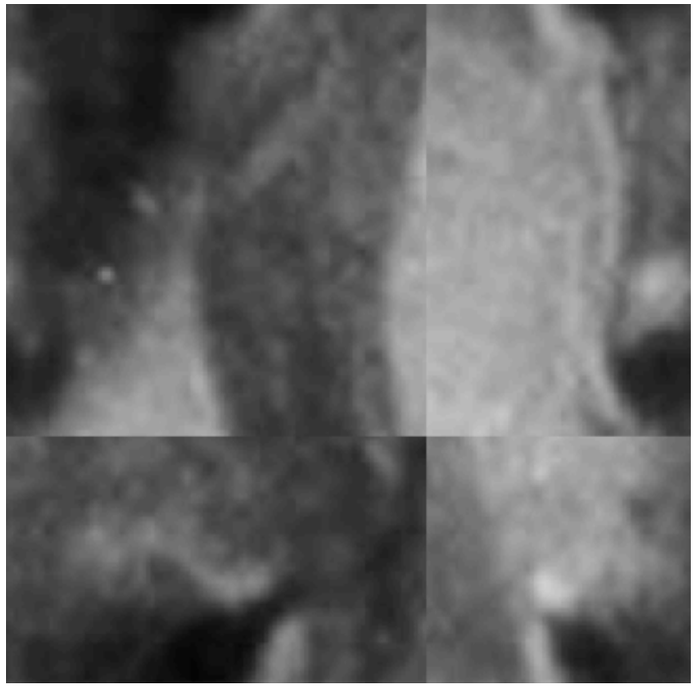
$\mathbf{x}[0, 0]$

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

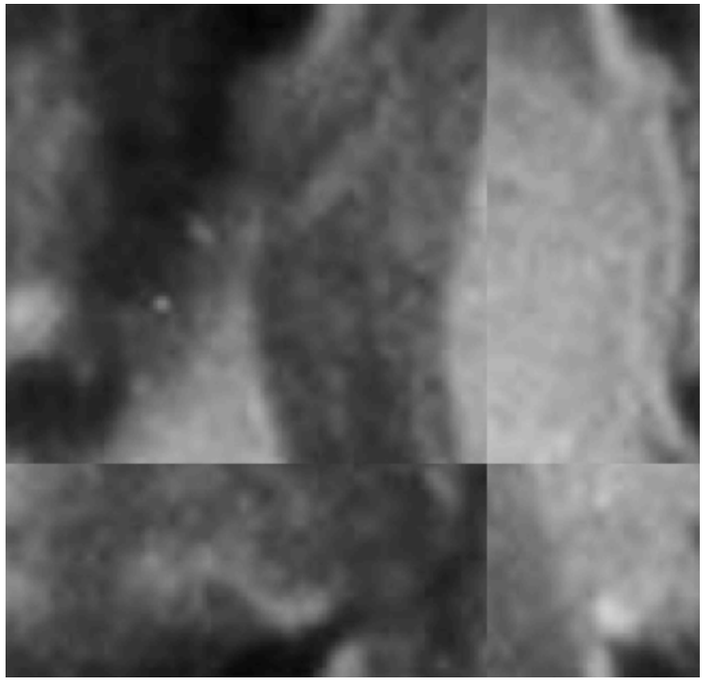
$\mathbf{x}[-20, -20]$



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



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



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

```
def circshift(im,shift):
  return roll(roll(im,shift[0],axis=0),shift[1],axis=1)
```

Linear Least Squares Discriminant

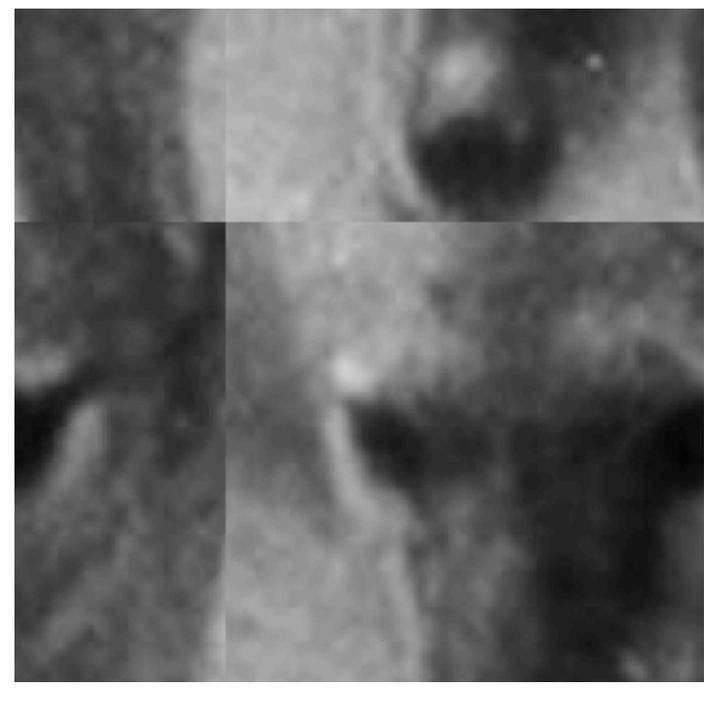
- One can view a correlation filters in the spatial domain as a linear least squares discriminant.
- Made popular by Bolme et al., referred to in literature as a **M**inimum **O**utput **S**um of **S**quared **E**rror (MOSSE) filter.

$$\arg\min_{\mathbf{h}} \frac{1}{2} \sum_{\tau \in \mathcal{C}} |y_\tau - \mathbf{x}[\tau]^T \mathbf{h}|^2$$

Linear Least Squares Discriminant

- One can view a correlation filters in the spatial domain as a linear least squares discriminant.
- Made popular by Bolme et al., referred to in literature as a **M**inimum **O**utput **S**um of **S**quared **E**rror (MOSSE) filter.

$$\arg\min_{\mathbf{h}} \frac{1}{2} \sum_{\tau \in \mathcal{C}} |y_\tau - \mathbf{x}[\tau]^T \mathbf{h}|^2$$

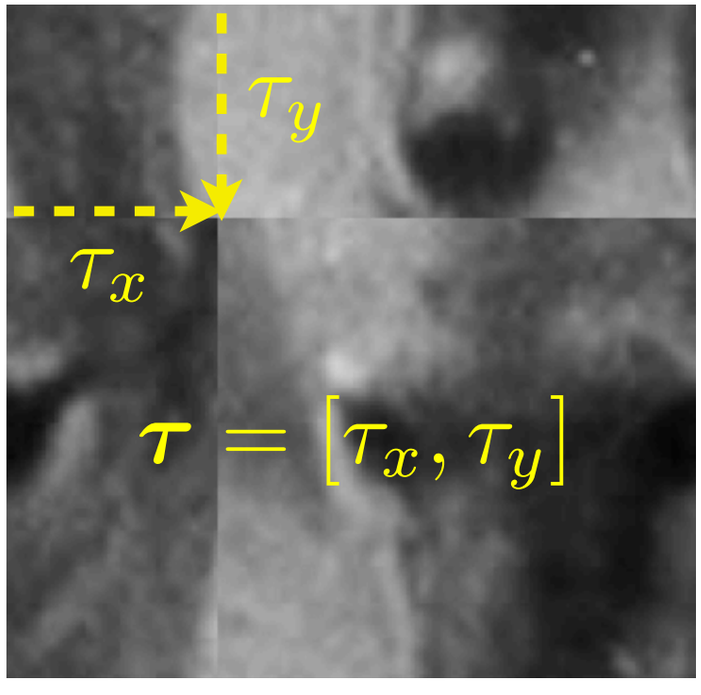
$\mathbf{x}[\boldsymbol{\tau}]$ 

Linear Least Squares Discriminant

- One can view a correlation filters in the spatial domain as a linear least squares discriminant.
- Made popular by Bolme et al., referred to in literature as a **M**inimum **O**utput **S**um of **S**quared **E**rror (MOSSE) filter.

$$\arg\min_{\mathbf{h}} \frac{1}{2} \sum_{\tau \in \mathcal{C}} |y_\tau - \mathbf{x}[\tau]^T \mathbf{h}|^2$$

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



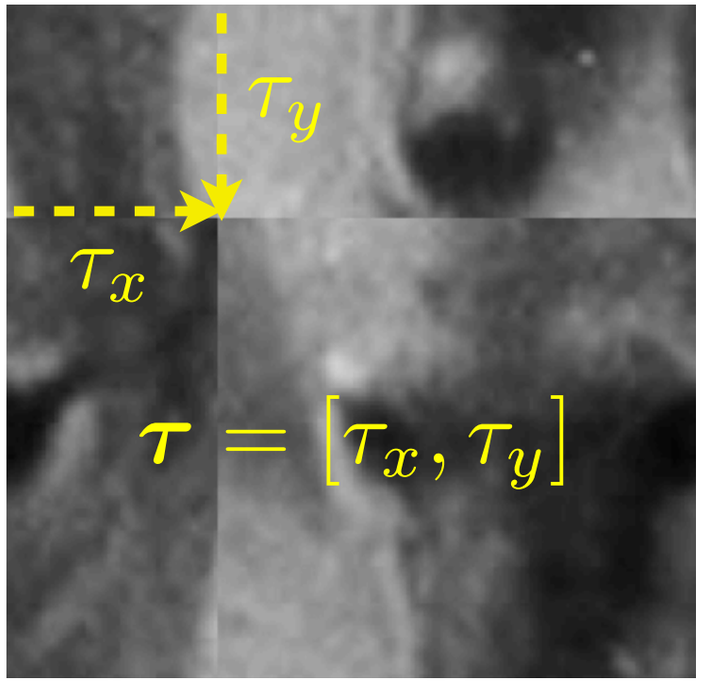
Linear Least Squares Discriminant

- One can view a correlation filters in the spatial domain as a linear least squares discriminant.
- Made popular by Bolme et al., referred to in literature as a **M**inimum **O**utput **S**um of **S**quared **E**rror (MOSSE) filter.

$$\arg\min_{\mathbf{h}} \frac{1}{2} \sum_{\tau \in \mathcal{C}} |y_\tau - \mathbf{x}[\tau]^T \mathbf{h}|^2$$

"set of all shifts"

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



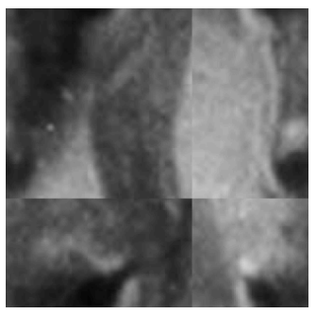
Ridge Regression

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

"data term"

"regulariser"
or "prior"

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

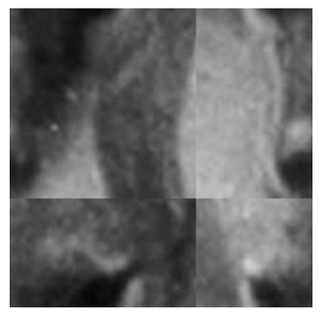
$\mathbf{x}[\boldsymbol{\tau}]$			• • •	
$y_{\boldsymbol{\tau}}$	1	0	• • •	0

$$\mathbf{X} = [\mathbf{x}[\boldsymbol{\tau_1}], \ldots, \mathbf{x}[\boldsymbol{\tau}_D]] \qquad \mathbf{y} = [y_{\boldsymbol{\tau}_1}, \ldots, y_{\boldsymbol{\tau}_D}]^T$$

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

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

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

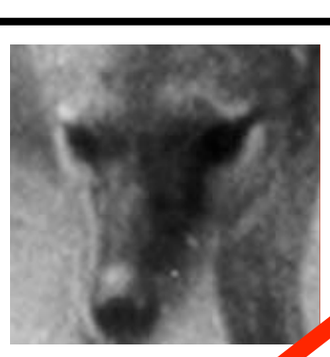
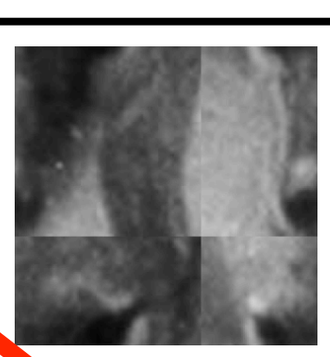
$\mathbf{x}[\boldsymbol{\tau}]$			• • •	
$y_{\boldsymbol{\tau}}$	1	0	• • •	0

$$\mathbf{X} = [\mathbf{x}[\boldsymbol{\tau_1}], \ldots, \mathbf{x}[\boldsymbol{\tau}_D]] \qquad \mathbf{y} = [y_{\boldsymbol{\tau}_1}, \ldots, y_{\boldsymbol{\tau}_D}]^T$$

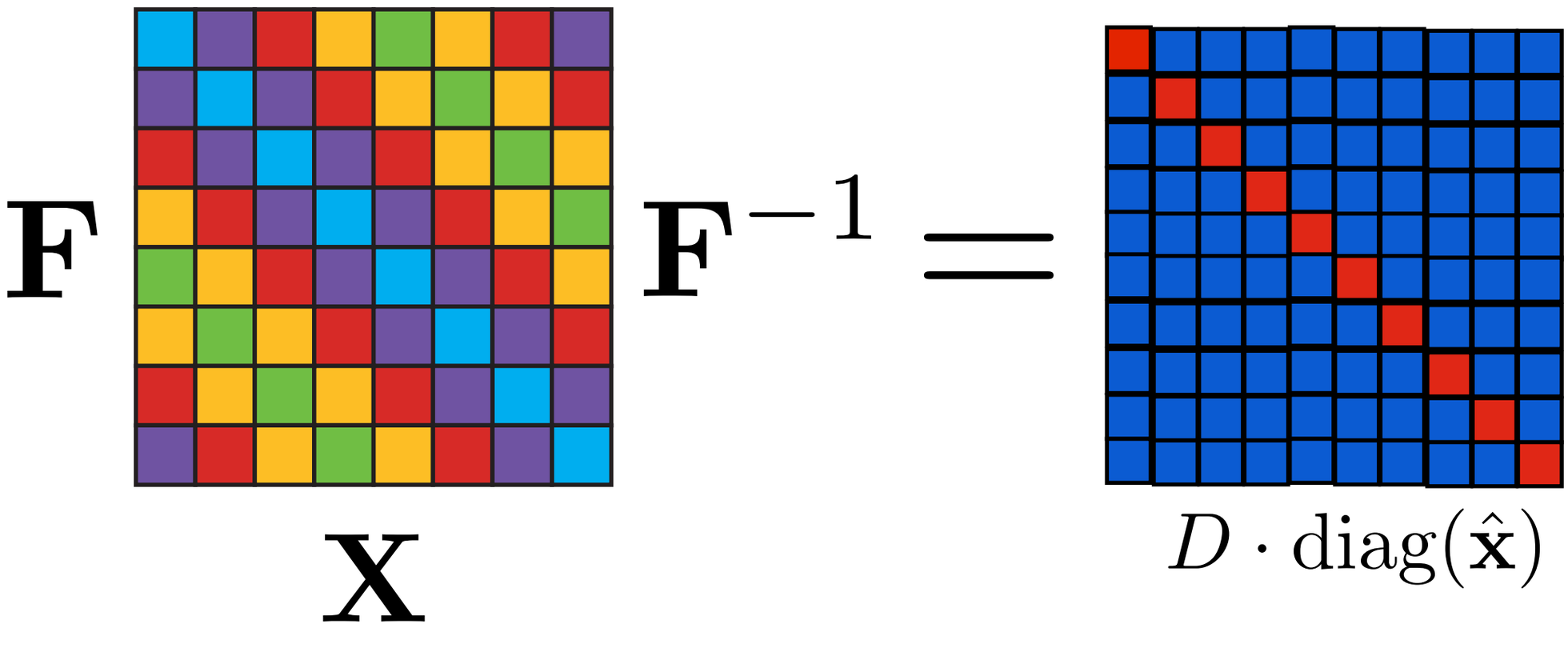
$$E(\mathbf{h}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2$$

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

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

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

$$\mathbf{X} = [\mathbf{x}[\boldsymbol{\tau}_1], \ldots, \mathbf{x}[\boldsymbol{\tau}_D]] \qquad \mathbf{y} = [y_{\boldsymbol{\tau}_1}, \ldots, y_{\boldsymbol{\tau}_D}]^T$$



$\mathbf{F} \leftarrow$ Fourier Transform

Not Always Zero

Always Zero

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{h} = \mathbf{X}^T\mathbf{y}$$

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{h} = \mathbf{X}^T\mathbf{y}$$

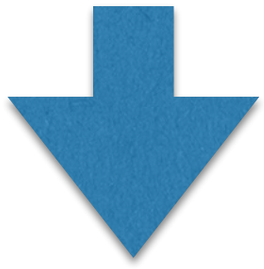
"express as convolution"

$$(\mathbf{x} \otimes \mathbf{x} + \lambda\boldsymbol{\delta}) * \mathbf{h} = \mathbf{x} \otimes \mathbf{y}$$

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{h} = \mathbf{X}^T\mathbf{y}$$

"express as convolution"

$$(\mathbf{x} \otimes \mathbf{x} + \lambda\boldsymbol{\delta}) * \mathbf{h} = \mathbf{x} \otimes \mathbf{y}$$



"diagonalise with DFT"

$$(\hat{\mathbf{x}}^* \circ \hat{\mathbf{x}} + \lambda\mathbf{1}) \circ \hat{\mathbf{h}} = \hat{\mathbf{x}}^* \circ \hat{\mathbf{y}}$$

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{h} = \mathbf{X}^T\mathbf{y}$$

"express as convolution"

$$(\mathbf{x} \otimes \mathbf{x} + \lambda\boldsymbol{\delta}) * \mathbf{h} = \mathbf{x} \otimes \mathbf{y}$$

"diagonalise with DFT"

$$(\hat{\mathbf{x}}^* \circ \hat{\mathbf{x}} + \lambda\mathbf{1}) \circ \hat{\mathbf{h}} = \hat{\mathbf{x}}^* \circ \hat{\mathbf{y}}$$

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

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

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

```
>>> 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)
```

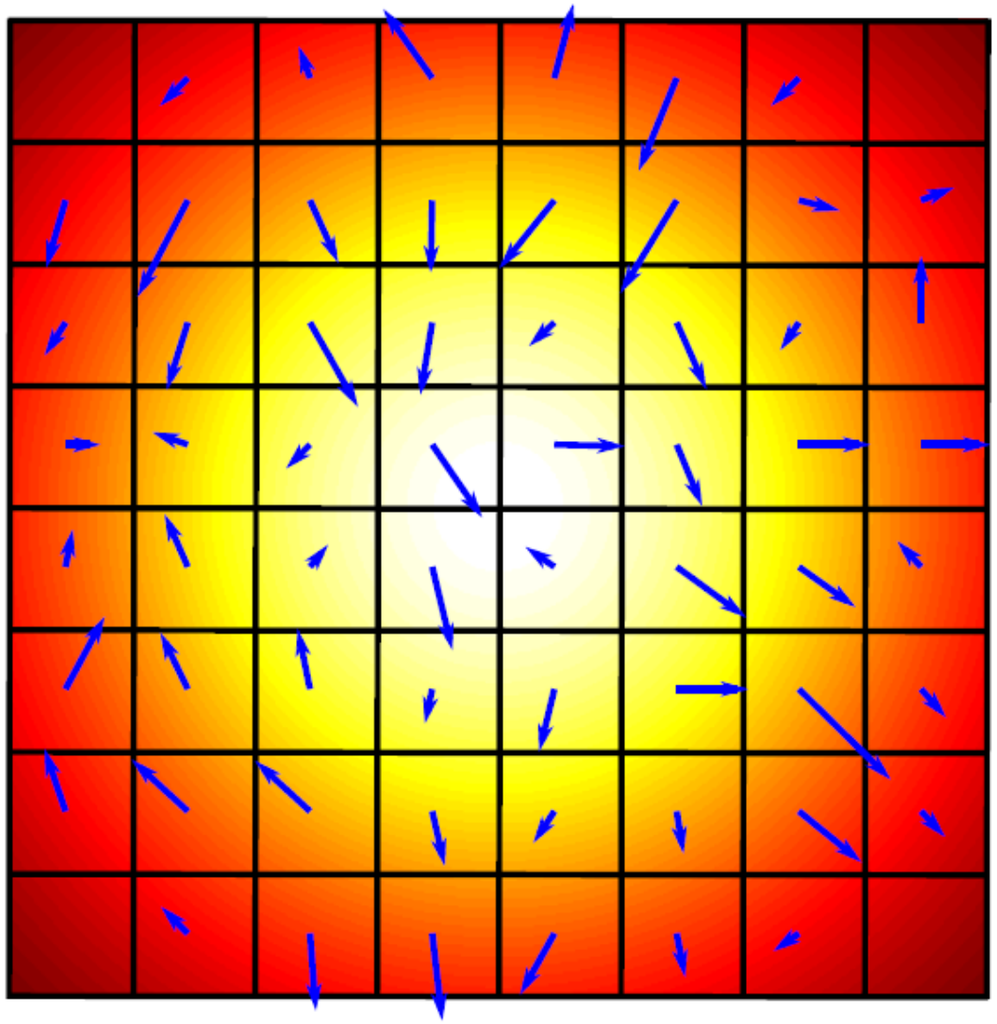
$$\hat{\mathbf{h}} = \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda \mathbf{1})$$

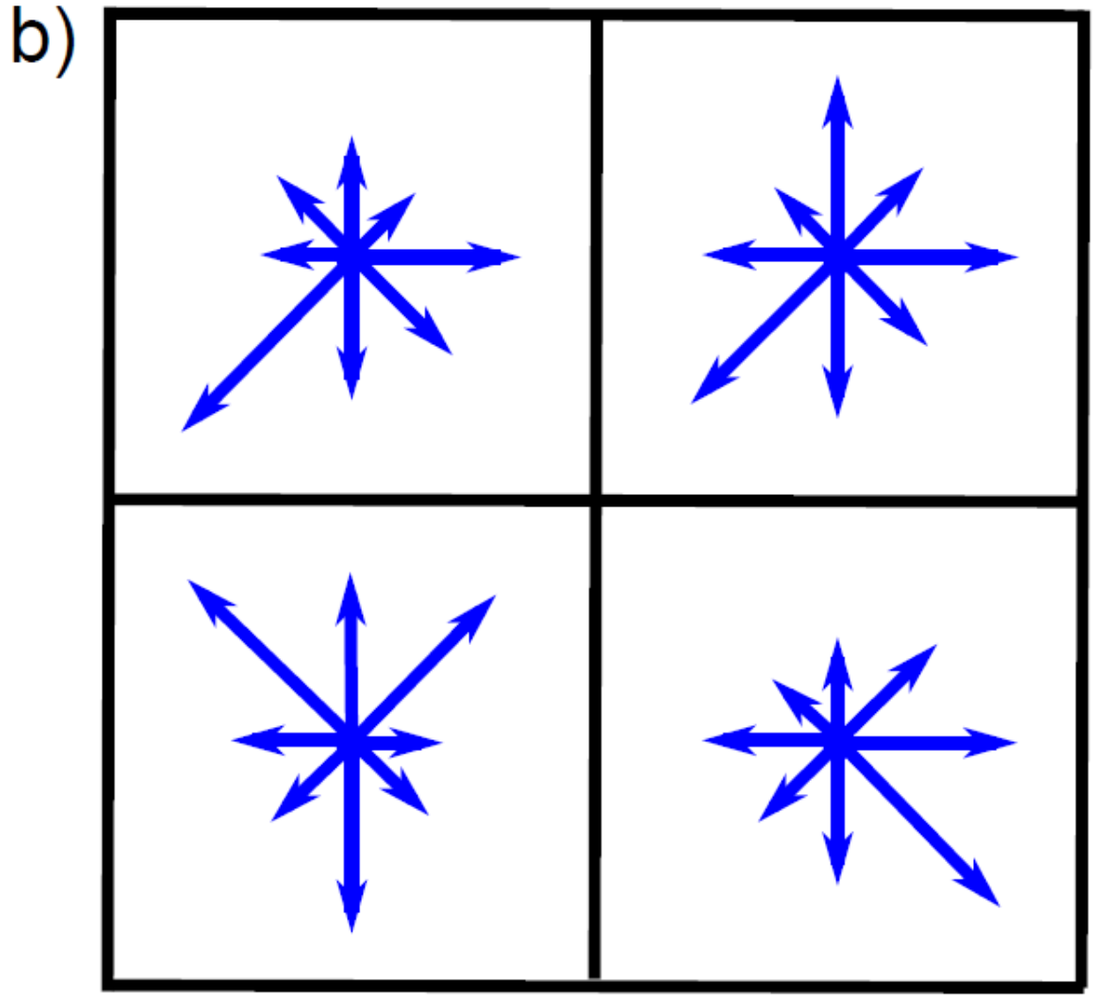
```
>>> xf = fft2(x)
>>> yf = fft2(y)
>>> sxx = xf*conj(xf)
>>> sxy = xf*conj(yf)
>>> hf = sxy/(sxx + 1e-3)
```

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

Robust Representations



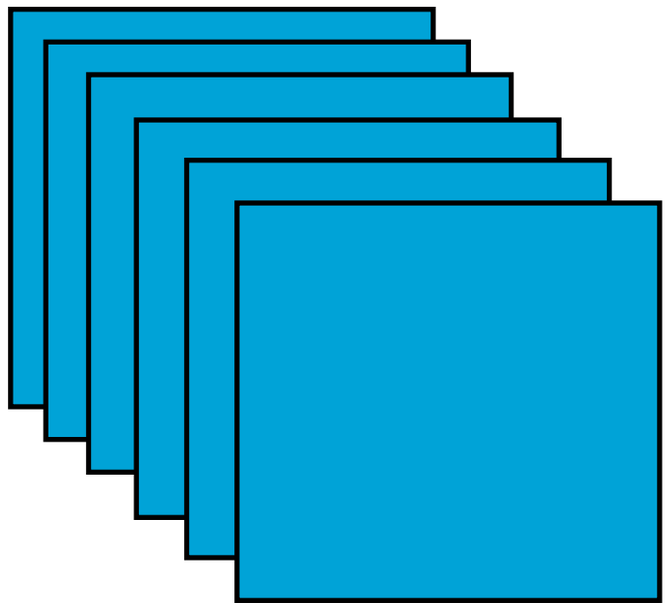


1. Compute image gradients

2. Pool into local histograms
3. Concatenate histograms
4. Normalize histograms

$$\psi \rightarrow \mathbb{R}^N : \mathbb{R}^{N \times K}$$

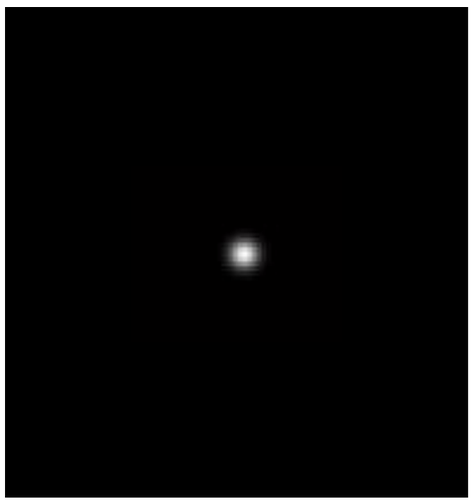
*



"unknown filter"

$\mathbf{h}$

"known signal" $\mathbf{x}$



"known response" $\mathbf{y}$

Kiani, Sim and Lucey ICCV 2013.

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

K = no. of channels

Kiani, Sim and Lucey ICCV 2013.

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

Kiani, Sim and Lucey ICCV 2013.

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

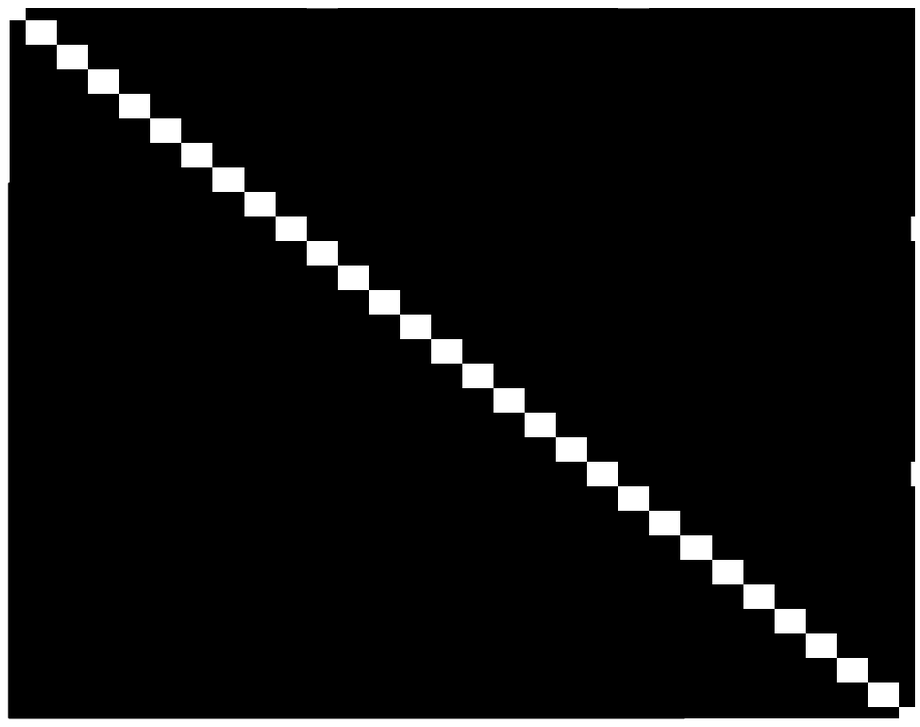
$$(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1} \quad \Rightarrow \quad \mathcal{O}(K^3 D^3)$$

$K = \text{no. of channels}$
$D = \text{number of samples in } \mathbf{x}$

Kiani, Sim and Lucey ICCV 2013.

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

$$\hat{\mathbf{X}}\hat{\mathbf{X}}^T + \lambda\mathbf{I} =$$



Kiani, Sim and Lucey ICCV 2013.

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

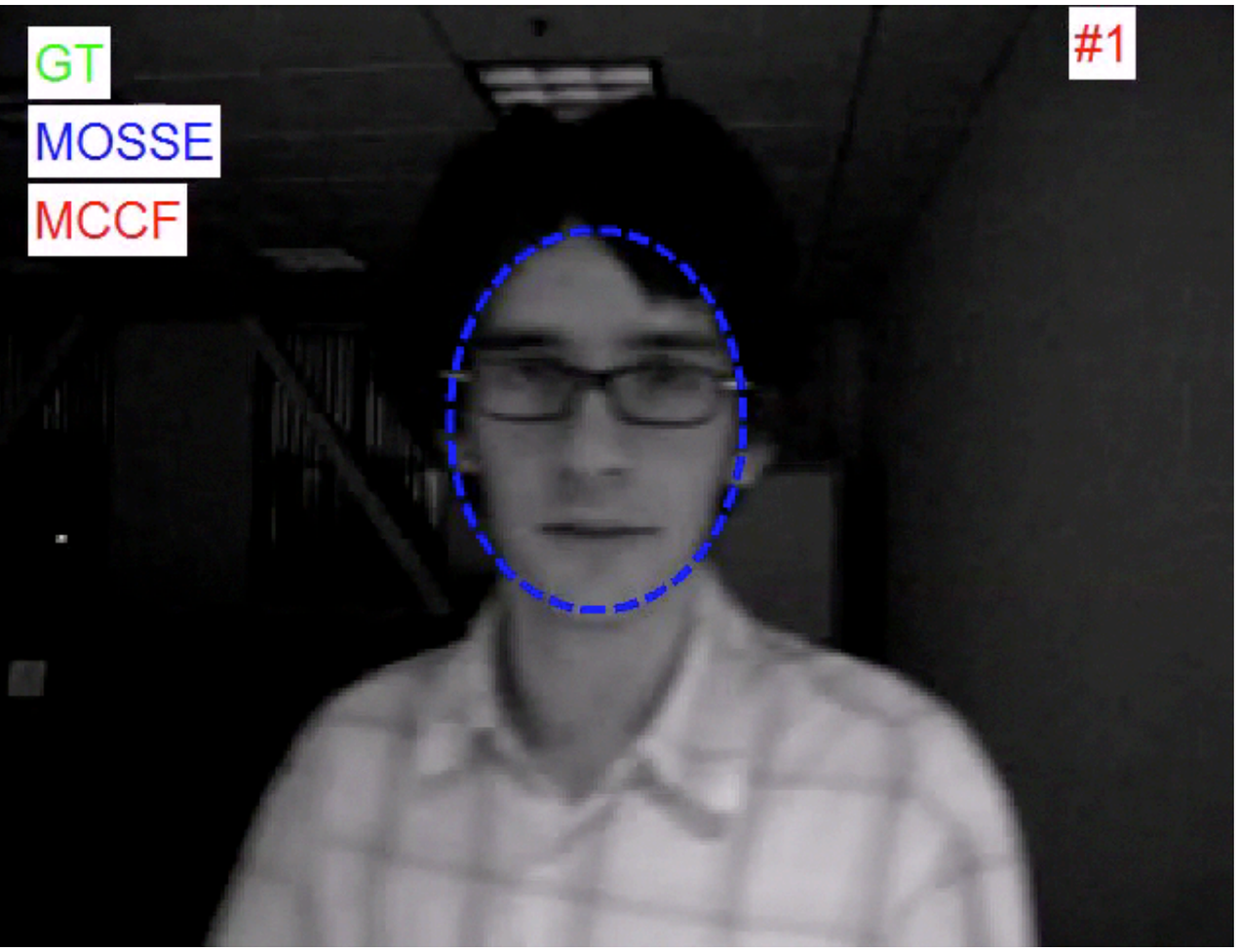
$$(\hat{\mathbf{X}}\hat{\mathbf{X}}^T + \lambda\mathbf{I})^{-1} \Rightarrow \mathcal{O}(K^3 D)$$

$K = \text{no. of channels}$
$D = \text{number of samples in } \mathbf{x}$

where,

$$\hat{\mathbf{X}} = [\text{diag}(\hat{\mathbf{x}}_1), \ldots, \text{diag}(\hat{\mathbf{x}}_K)]$$

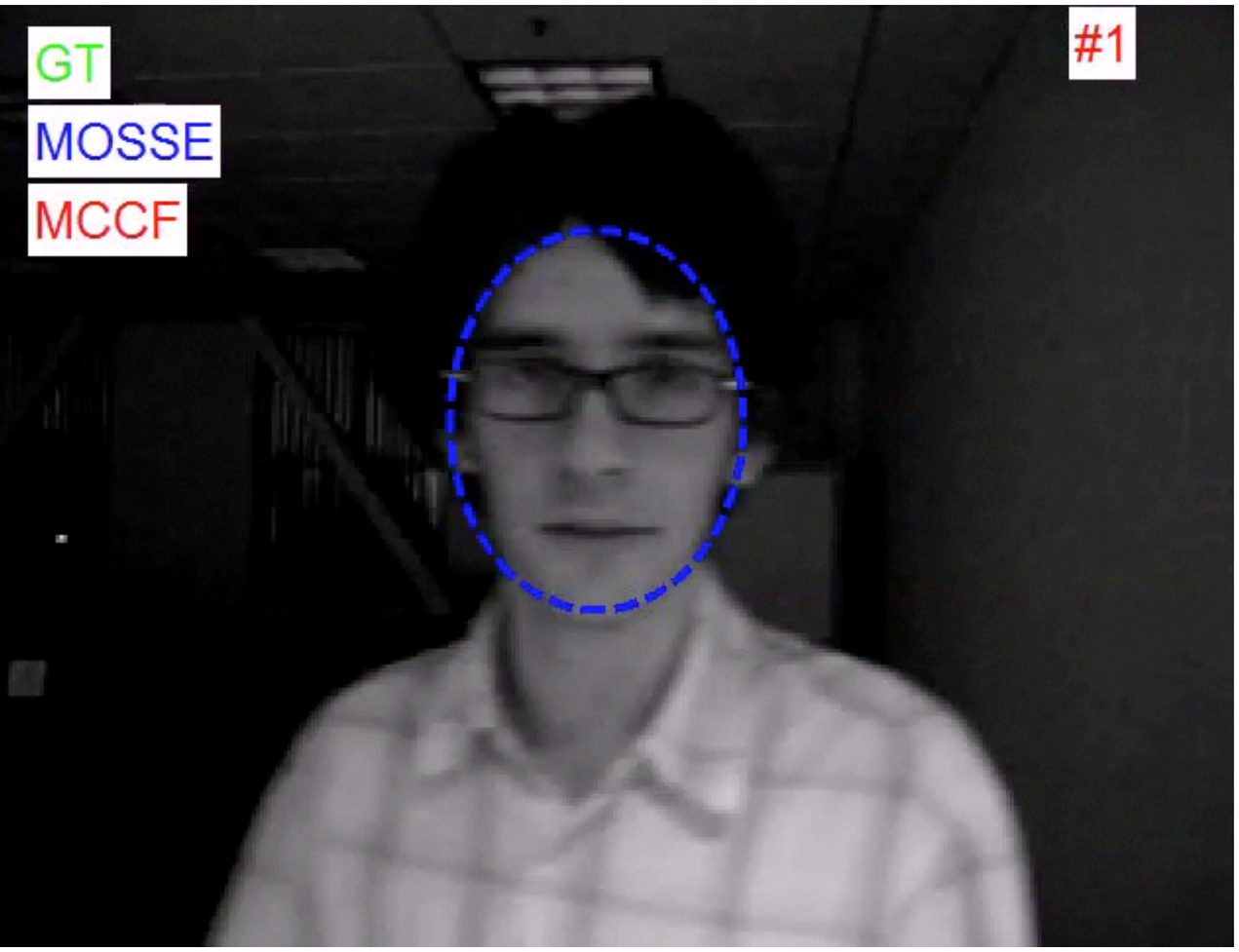
Kiani, Sim and Lucey ICCV 2013.



MOSSE = Single-Channel CF

MCCF = Multi-Channel CF

Kiani, Sim and Lucey ICCV 2013.

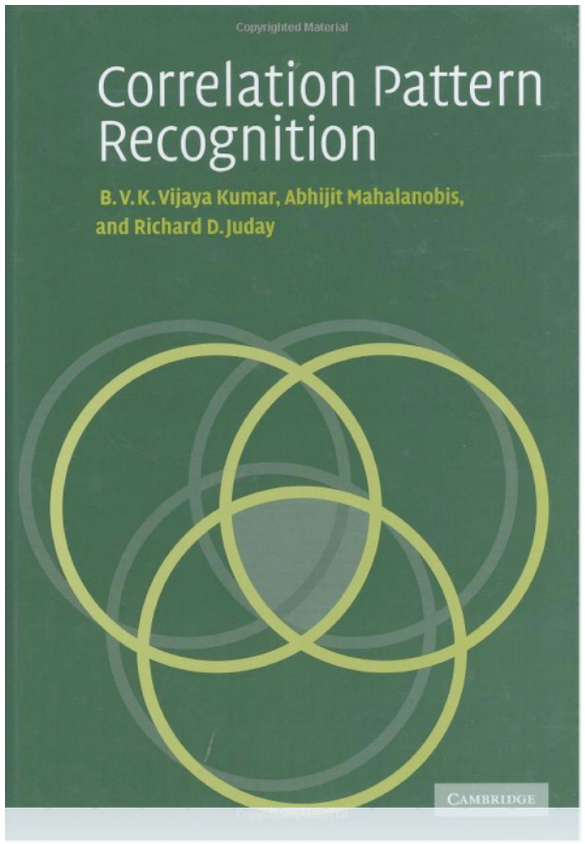


MOSSE = Single-Channel CF

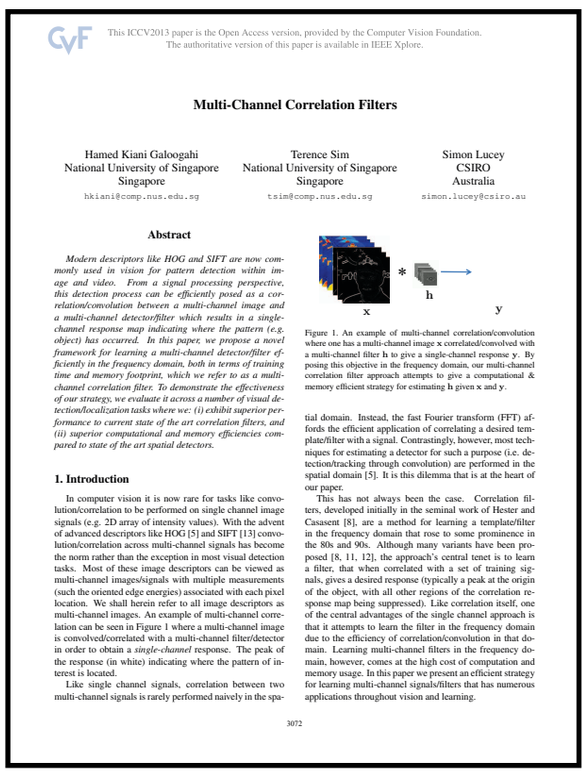
MCCF = Multi-Channel CF

Kiani, Sim and Lucey 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.
- Henriques, Caseiro, Martins, & Batista, "High-Speed Tracking with Kernelized Correlation Filters", PAMI 2014.



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 detection process can be efficiently posed as a 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, which we refer to as a multi-channel correlation filter. 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.

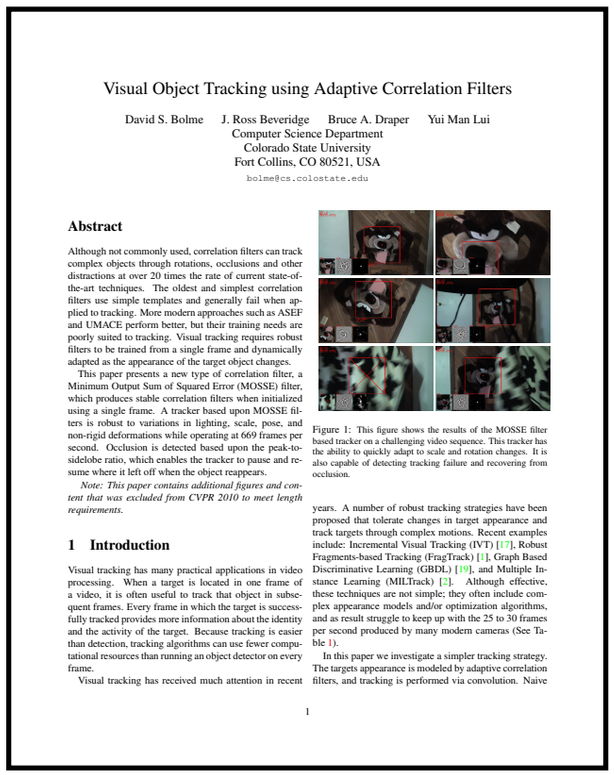
1. Introduction

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 [13] convolution/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 can be seen in Figure 1 where a multi-channel image is convolved/correlated with a multi-channel filter/detector in order to obtain a *single-channel* response. The peak of the response (in white) indicating where the pattern of interest is located.

Like single channel signals, correlation between two multi-channel signals is rarely performed naively in the spatial domain. Instead, the fast Fourier transform (FFT) affords the efficient application of correlating a desired template/filter with a signal. Contrastingly, however, most techniques for estimating a detector for such a purpose (i.e. detection/tracking through convolution) are performed in the spatial domain [5]. It is this dilemma that is at the heart of our paper.

Figure 1. An example of multi-channel correlation/convolution where one has a multi-channel image **x** correlated/convolved with a multi-channel filter **h** to give a single-channel response **y**. By posing this objective in the frequency domain, our multi-channel correlation filter approach attempts to give a computational & memory efficient strategy for estimating **h** given **x** and **y**.

This has not always been the case. Correlation filters, developed initially in the seminal work of Hester and Casasent [8], are a method for learning a template/filter in the frequency domain that rose to some prominence 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 attempts to learn the filter in the frequency domain due 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.



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 presents a new type of correlation filter, a Minimum Output Sum of Squared Error (MOSSE) filter, which produces 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

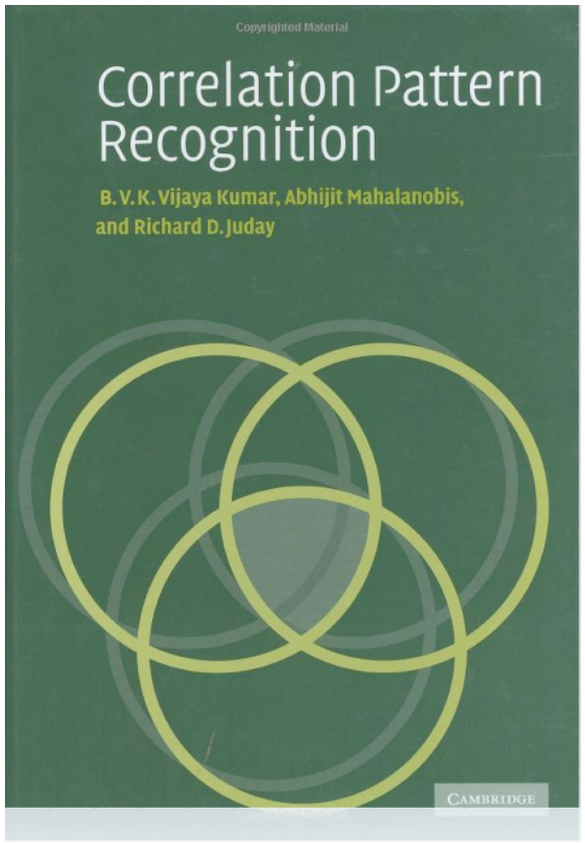
Visual tracking has many practical applications in video processing. When a target is located in one frame of a video, it is often useful to track that object in subsequent frames. Every frame in which the target is successfully tracked provides more information about the identity and the activity of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

Visual tracking has received much attention in recent years. A number of robust tracking strategies have been proposed that tolerate changes in target appearance and track targets through complex motions. Recent examples include: Incremental Visual Tracking (IVT) [17], Robust Fragments-based Tracking (FragTrack) [1], Graph Based Discriminative Learning (GBDL) [19], and Multiple Instance Learning (MILTrack) [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).

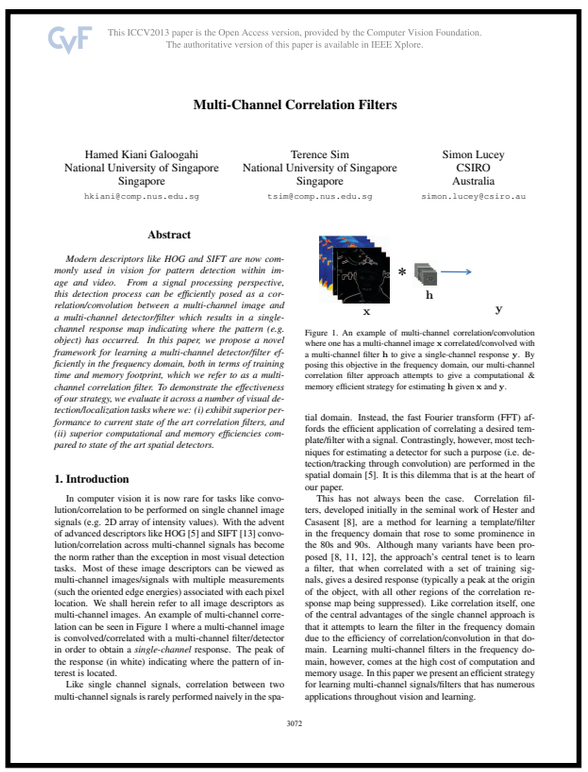
Figure 1: This figure shows the results of the MOSSE filter based tracker on a challenging video sequence. This tracker has the ability to quickly adapt to scale and rotation changes. It is also capable of detecting tracking failure and recovering from occlusion.

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

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.
- Henriques, Caseiro, Martins, & Batista, "High-Speed Tracking with Kernelized Correlation Filters", PAMI 2014.



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 detection process can be efficiently posed as a 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, which we refer to as a multi-channel correlation filter. 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.

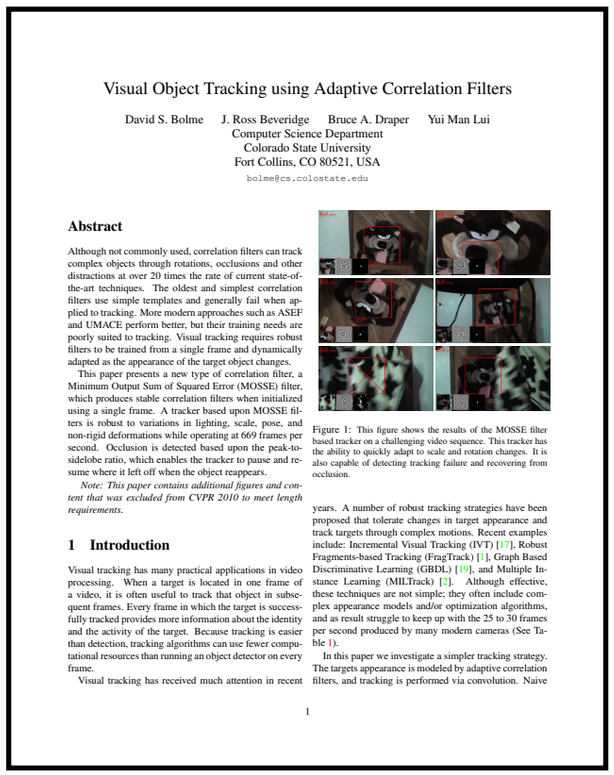
1. Introduction

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 [13] convolution/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 can be seen in Figure 1 where a multi-channel image is convolved/correlated with a multi-channel filter/detector in order to obtain a *single-channel* response. The peak of the response (in white) indicating where the pattern of interest is located.

Like single channel signals, correlation between two multi-channel signals is rarely performed naively in the spatial domain. Instead, the fast Fourier transform (FFT) affords the efficient application of correlating a desired template/filter with a signal. Contrastingly, however, most techniques for estimating a detector for such a purpose (i.e. detection/tracking through convolution) are performed in the spatial domain [5]. It is this dilemma that is at the heart of our paper.

Figure 1. An example of multi-channel correlation/convolution where one has a multi-channel image **x** correlated/convolved with a multi-channel filter **h** to give a single-channel response **y**. By posing this objective in the frequency domain, our multi-channel correlation filter approach attempts to give a computational & memory efficient strategy for estimating **h** given **x** and **y**.

This has not always been the case. Correlation filters, developed initially in the seminal work of Hester and Casasent [8], are a method for learning a template/filter in the frequency domain that rose to some prominence 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 attempts to learn the filter in the frequency domain due 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.



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 presents a new type of correlation filter, a Minimum Output Sum of Squared Error (MOSSE) filter, which produces 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.

Visual tracking has received much attention in recent years. A number of robust tracking strategies have been proposed that tolerate changes in target appearance and track targets through complex motions. Recent examples include: Incremental Visual Tracking (IVT) [17], Robust Fragments-based Tracking (FragTrack) [1], Graph Based Discriminative Learning (GBDL) [19], and Multiple Instance Learning (MILTrack) [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).

Figure 1: This figure shows the results of the MOSSE filter based tracker on a challenging video sequence. This tracker has the ability to quickly adapt to scale and rotation changes. It is also capable of detecting tracking failure and recovering from occlusion.

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

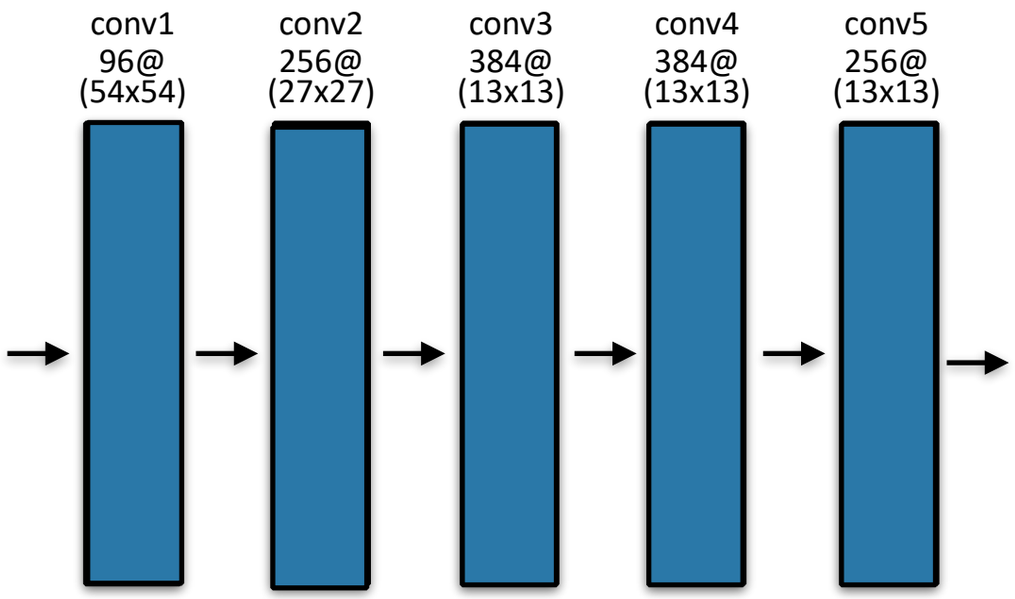
Question: Potential drawbacks of training with circular shifts?

Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks

Deep Tracking Methods

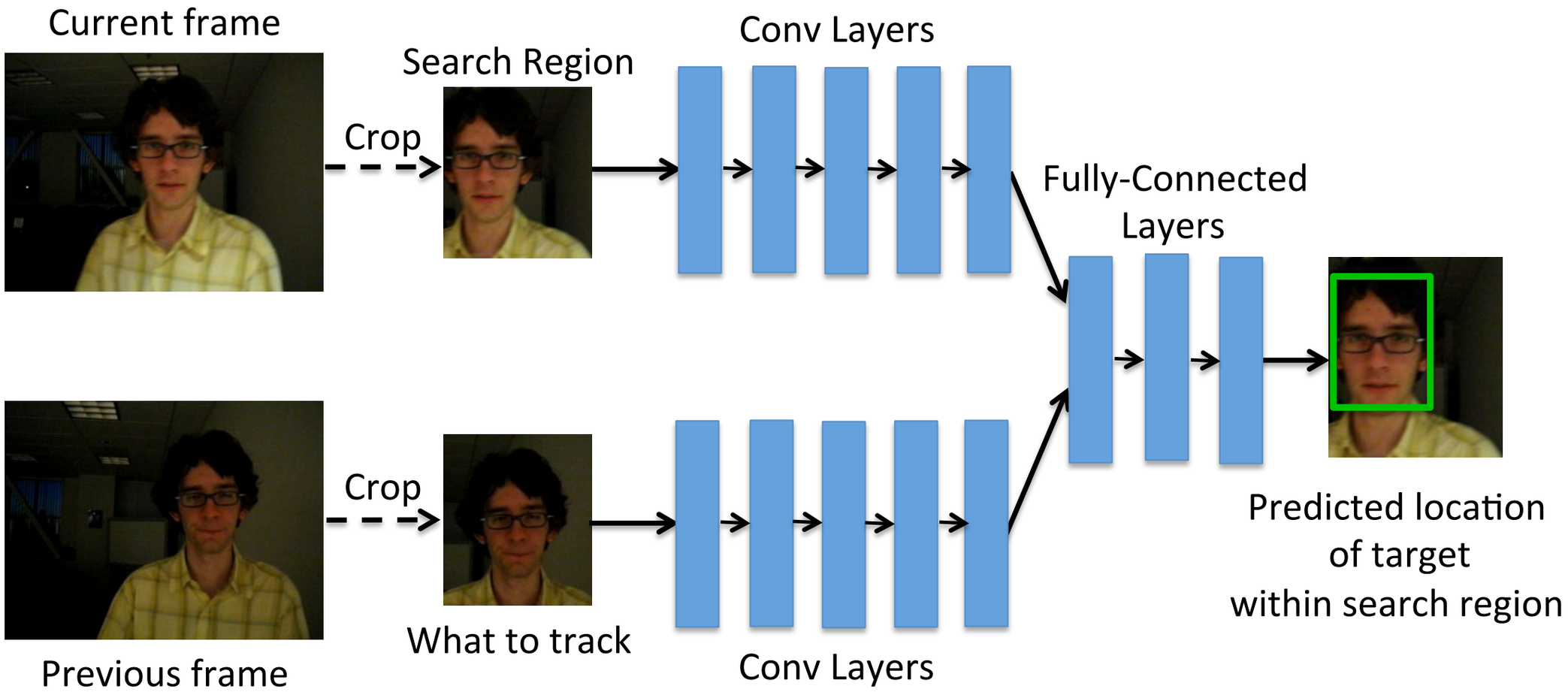
- Recently, there have been works that have tried to explore the employment of tracking using deep learning features.
- As efficiency is key, strategy is to learn from a large ensemble of labeled offline videos.
- Of particular interest are two papers,
  1. Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.
  2. Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, arXiv 2016.



Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks

Deep Regression Networks



Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

Deep Regression Networks

Previous video frame centered on object

Current video frame, shifted, with ground-truth bounding box

Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

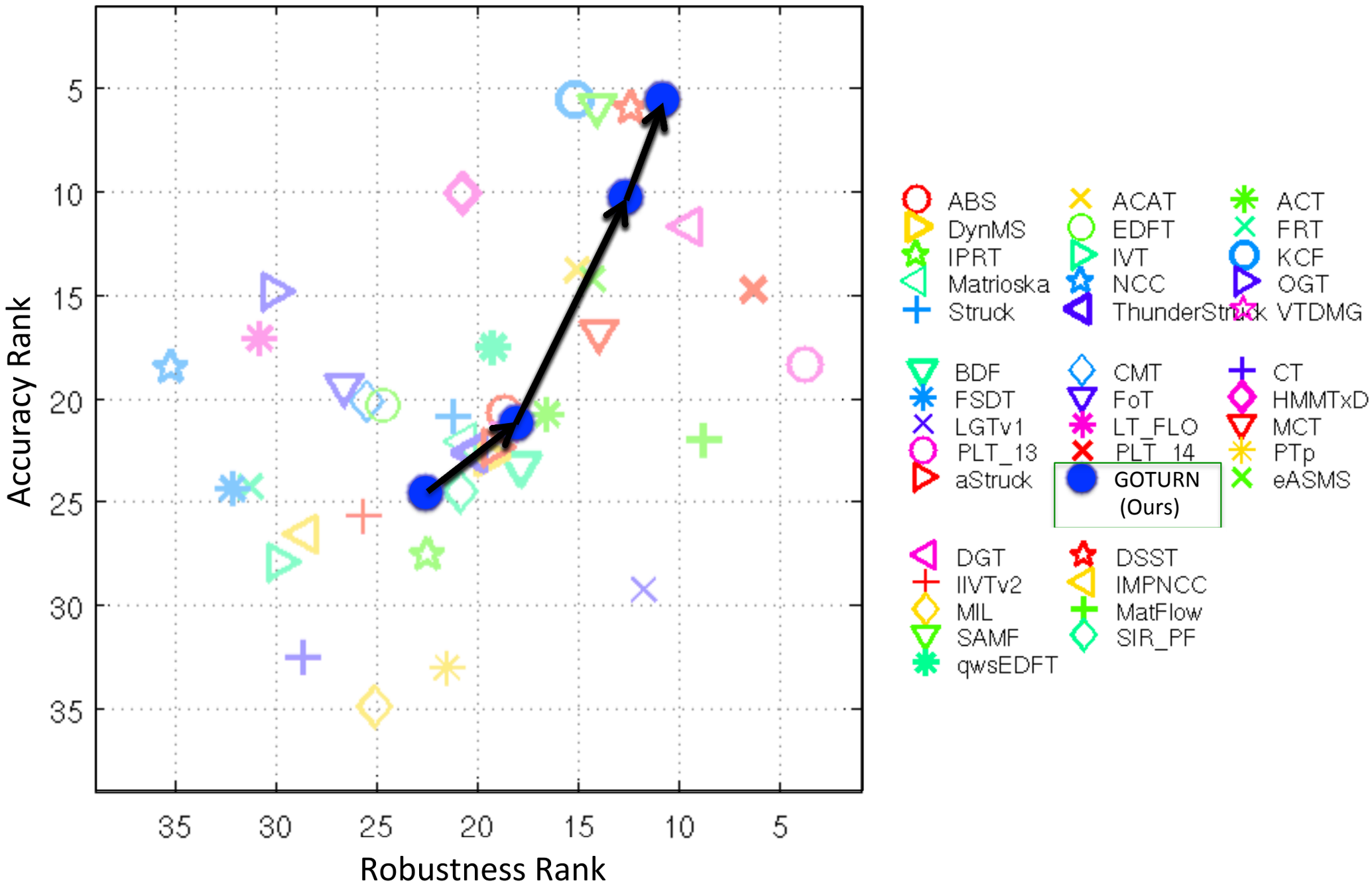
Deep Regression Networks

Image centered on object

Shifted image with ground-truth bounding box

Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

Deep Regression Networks



Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

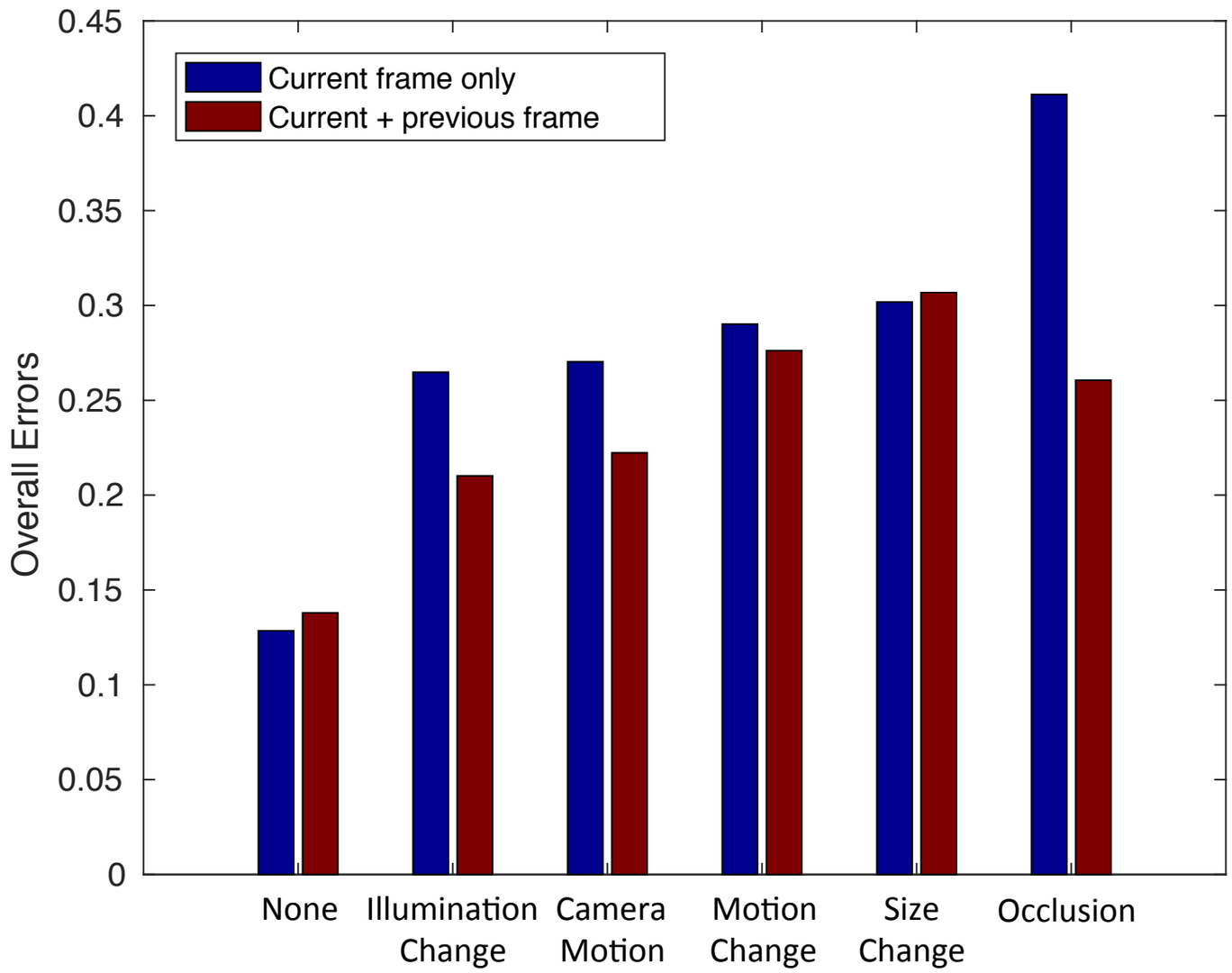
How does it work?

- Two hypotheses,
  1. The network compares the previous frame to the current frame to find the target object in the current frame.
  2. The network acts as a local generic “object detector” and simply locates the nearest “object.”

Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

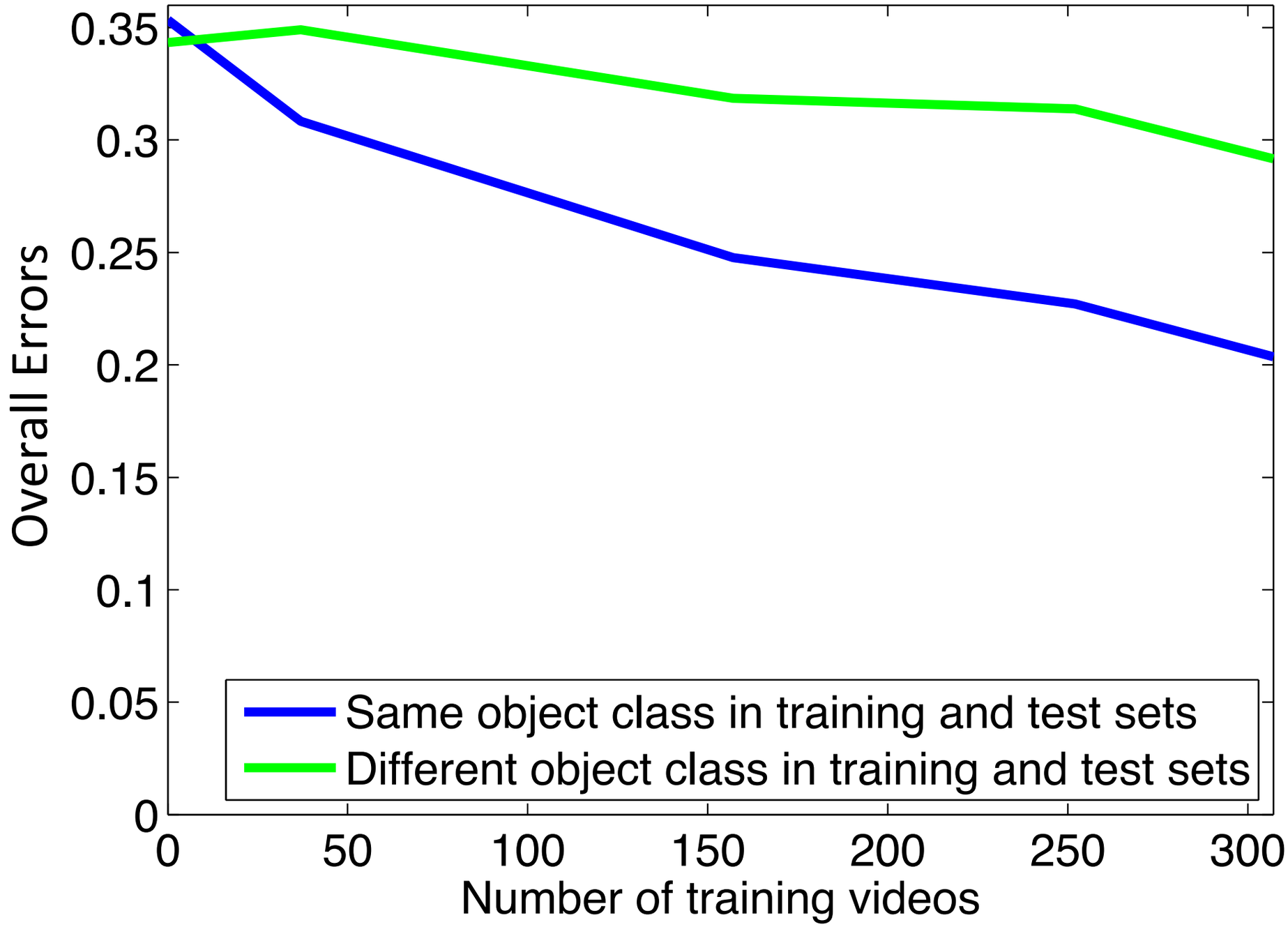
How does it work?

- Two hypotheses,
  1. The network compares the previous frame to the current frame to find the target object in the current frame.
  2. The network acts as a local generic “object detector” and simply locates the nearest “object.”



Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

Out-of-distribution generalisation

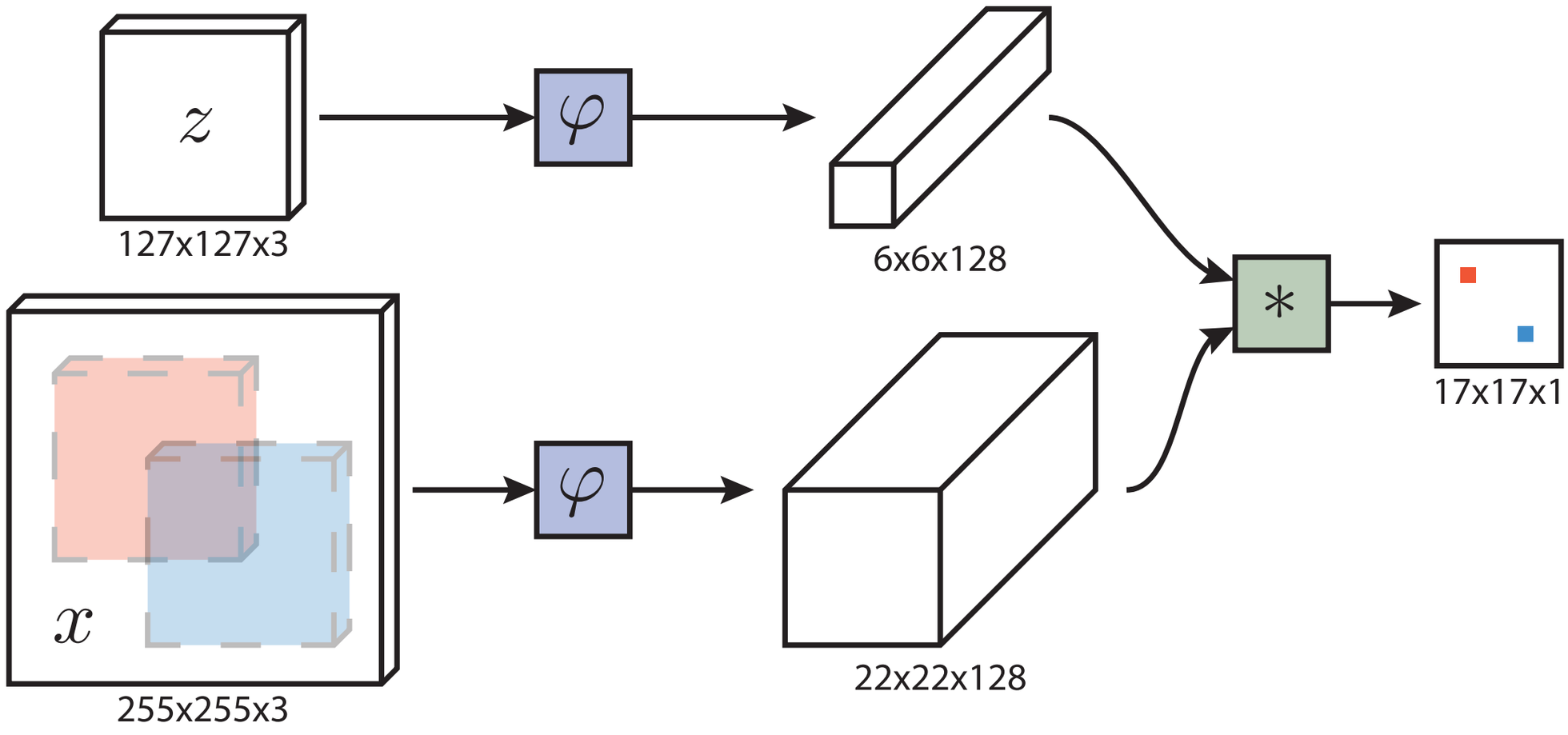


Held, Thrun, Savarese “Learning to Track at 100 FPS with Deep Regression Networks”, ECCV 2016.

Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks

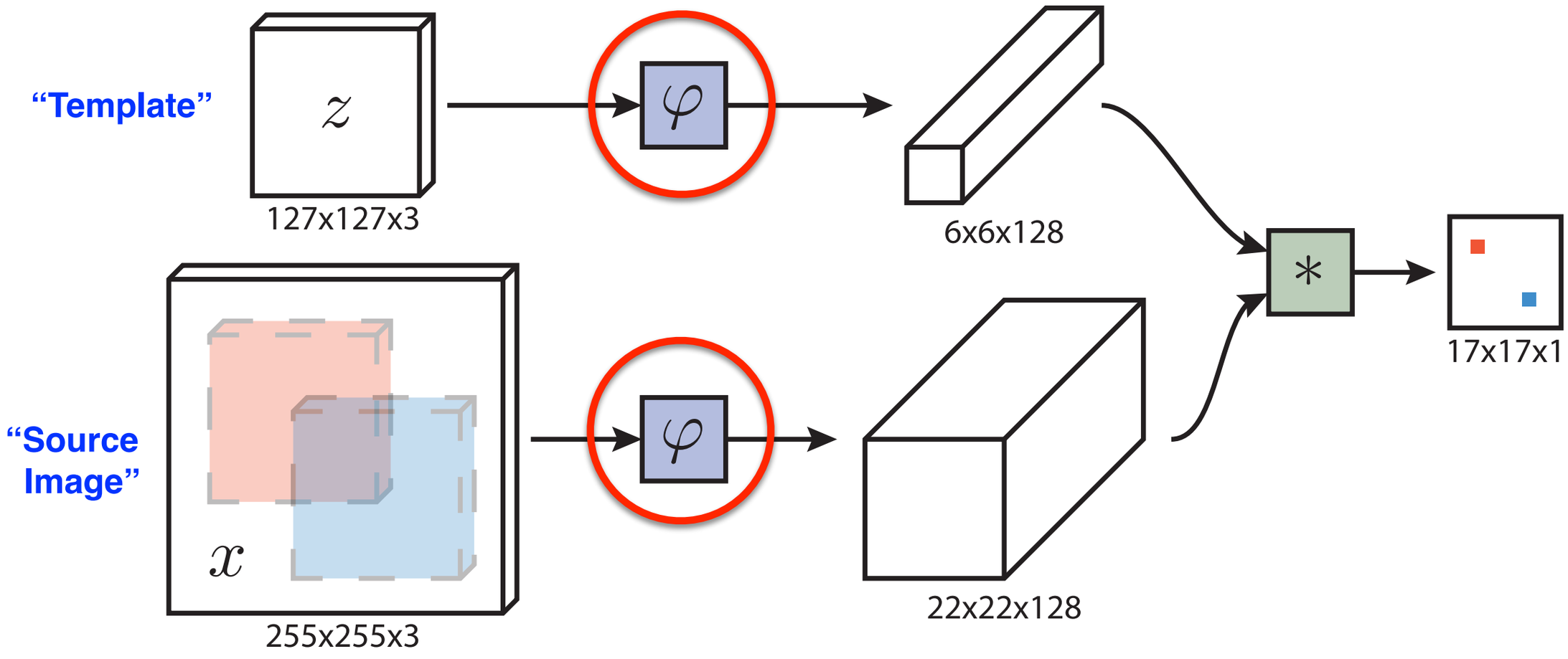
Fully Convolutional Siamese Networks



Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

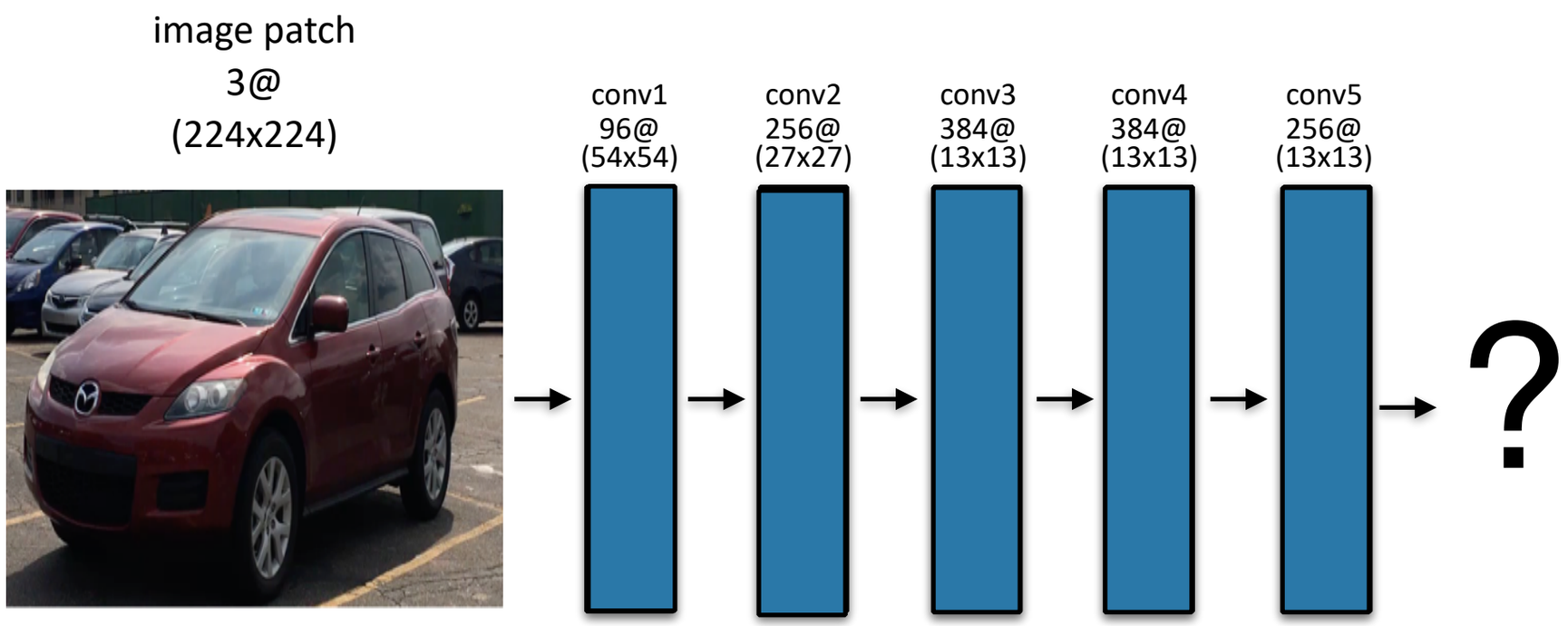
Fully Convolutional Siamese Networks

"Siamese" as they apply an identical transformation to both inputs



Bertinetto, Valmadre, Henriques, Vedaldi, Torr "Fully-Convolutional Siamese Networks for Object Tracking", ArXiv 2016.

Fully Convolutional



Feature transform commutes with displacement operator

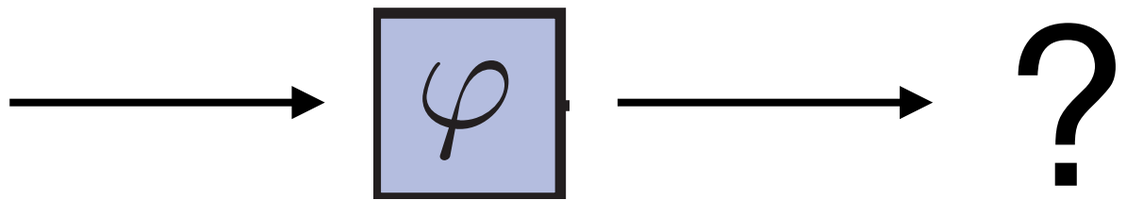
$$L_\tau x[t] = x[t - \tau]$$ (possibly with stride s ≠ 1)

$$L_\tau \varphi(x) = \varphi(L_{s\tau} x)$$

Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

Fully Convolutional

image patch
3@
(224x224)



Feature transform commutes with displacement operator

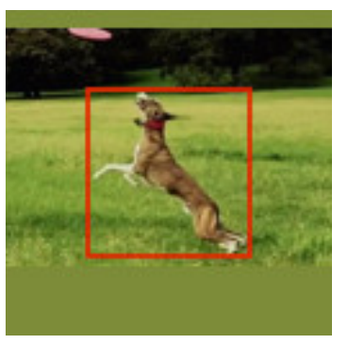
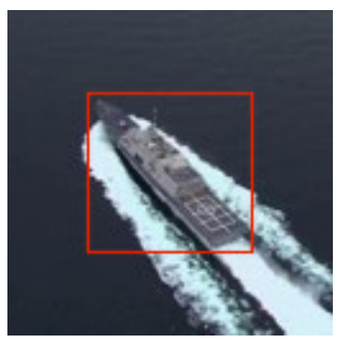
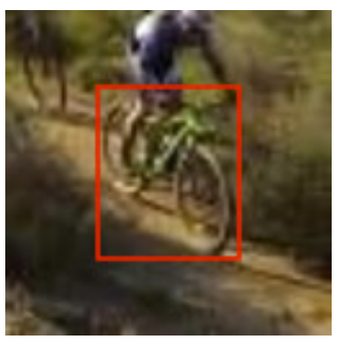
$$L_\tau x[t] = x[t - \tau]$$ (possibly with stride s ≠ 1)

$$L_\tau \varphi(x) = \varphi(L_{s\tau} x)$$

Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

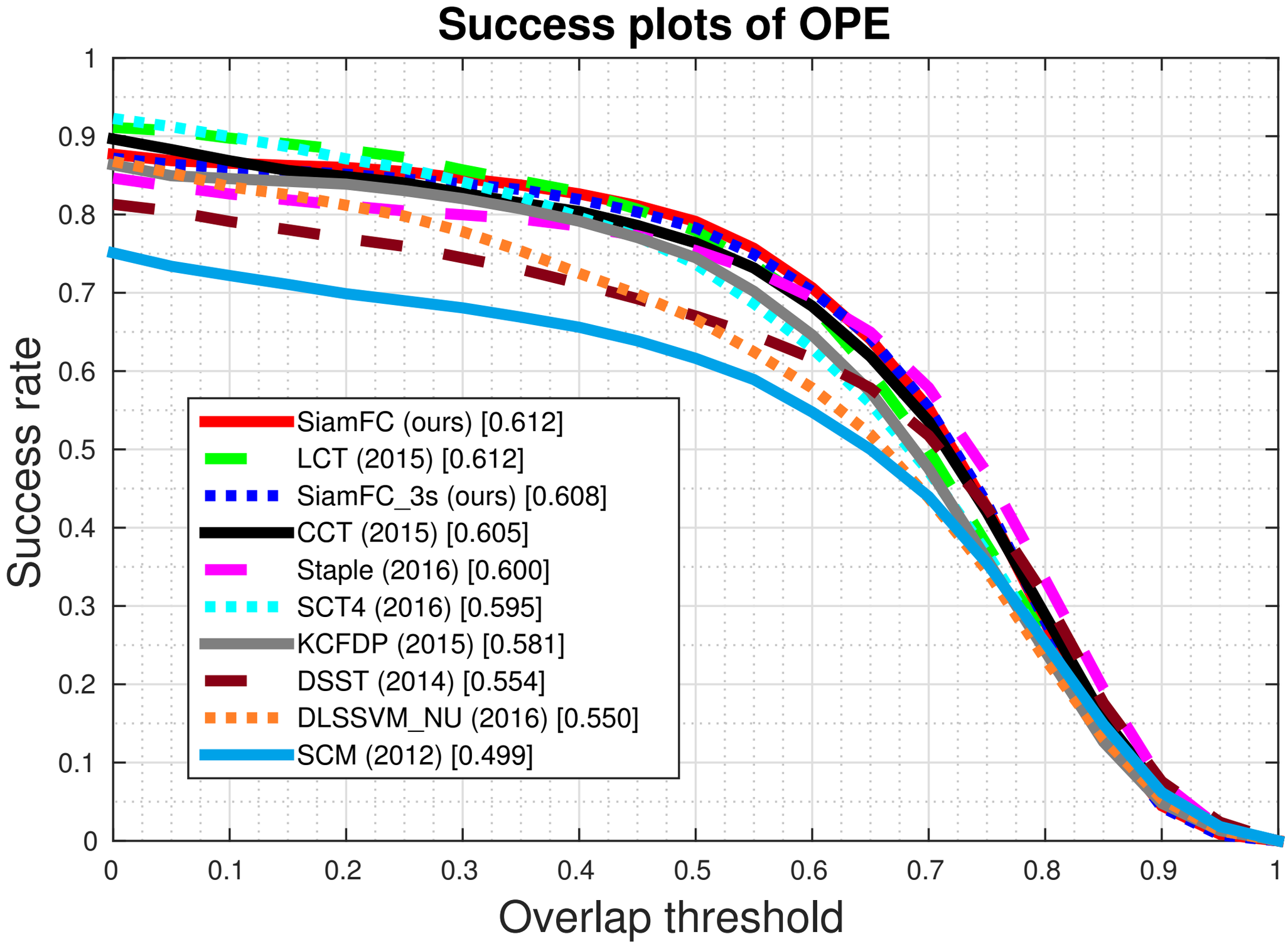
Fully Convolutional Siamese Networks

- Example pairs from training sequences


Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

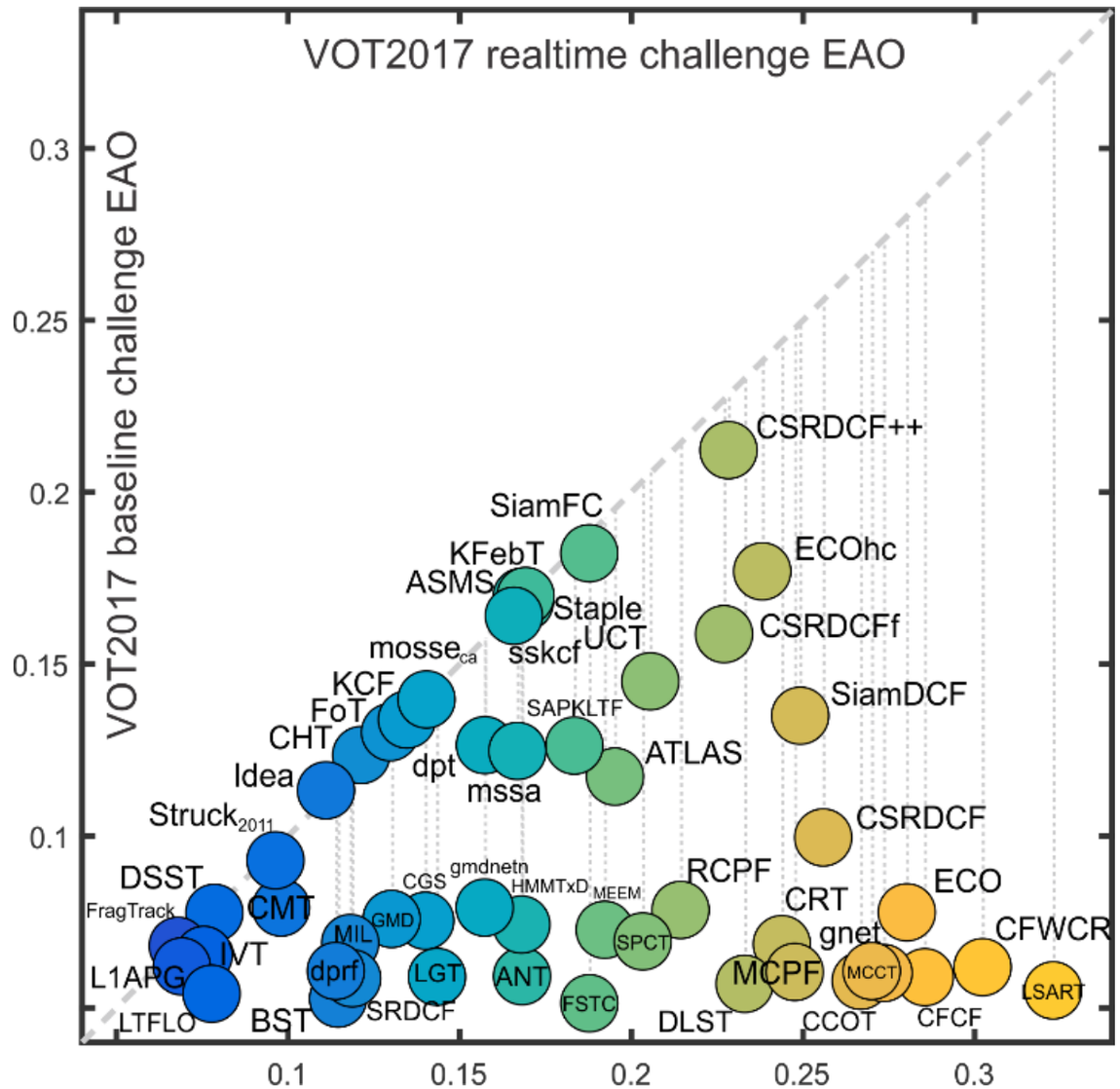
Fully Convolutional Siamese Networks



Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

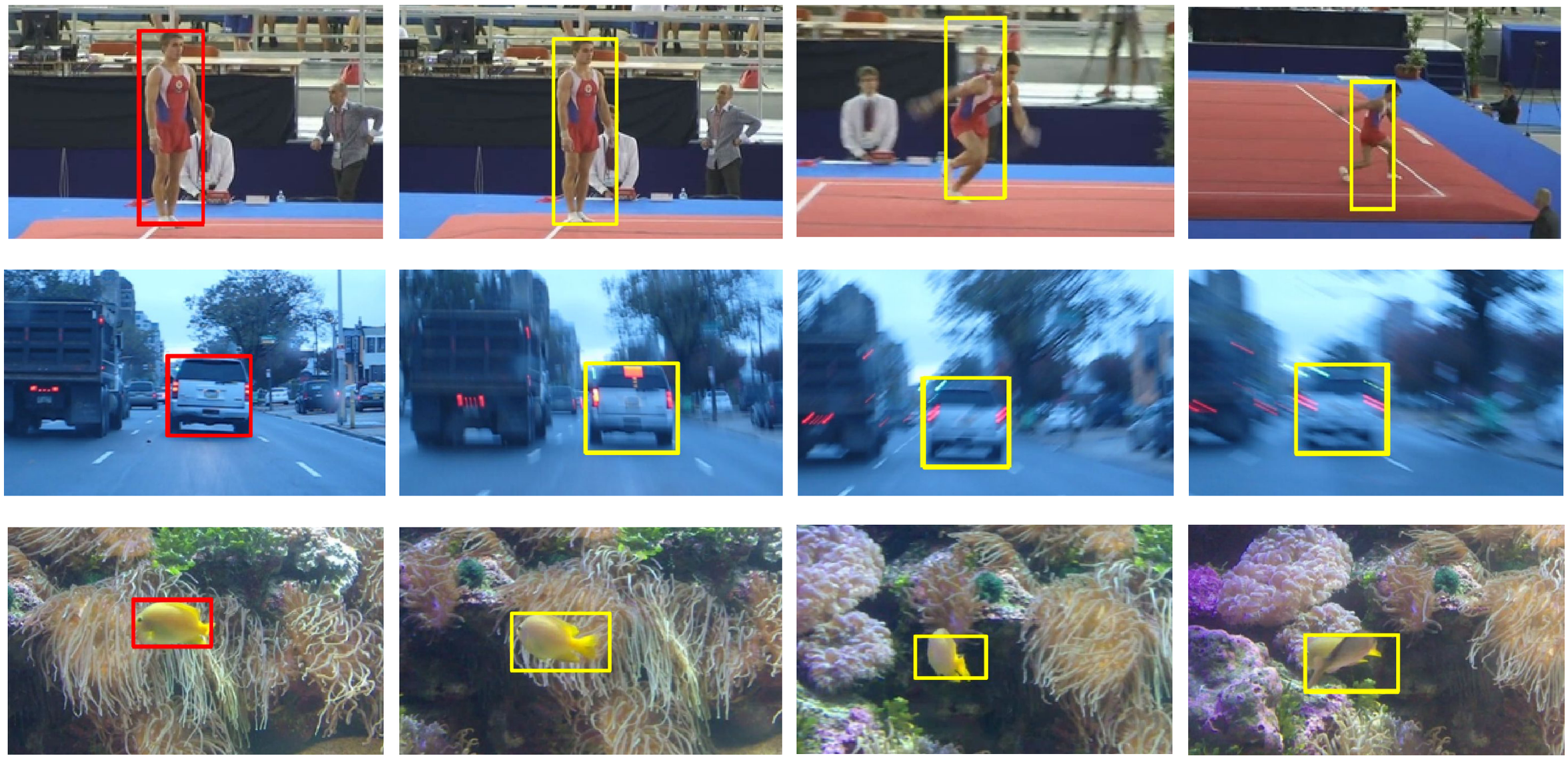
Fully Convolutional Siamese Networks

- Real-time evaluation: frames dropped if tracker is not ready
  - 22nd overall
  - 2nd in real-time



https://www.votchallenge.net/vot2017/results.html

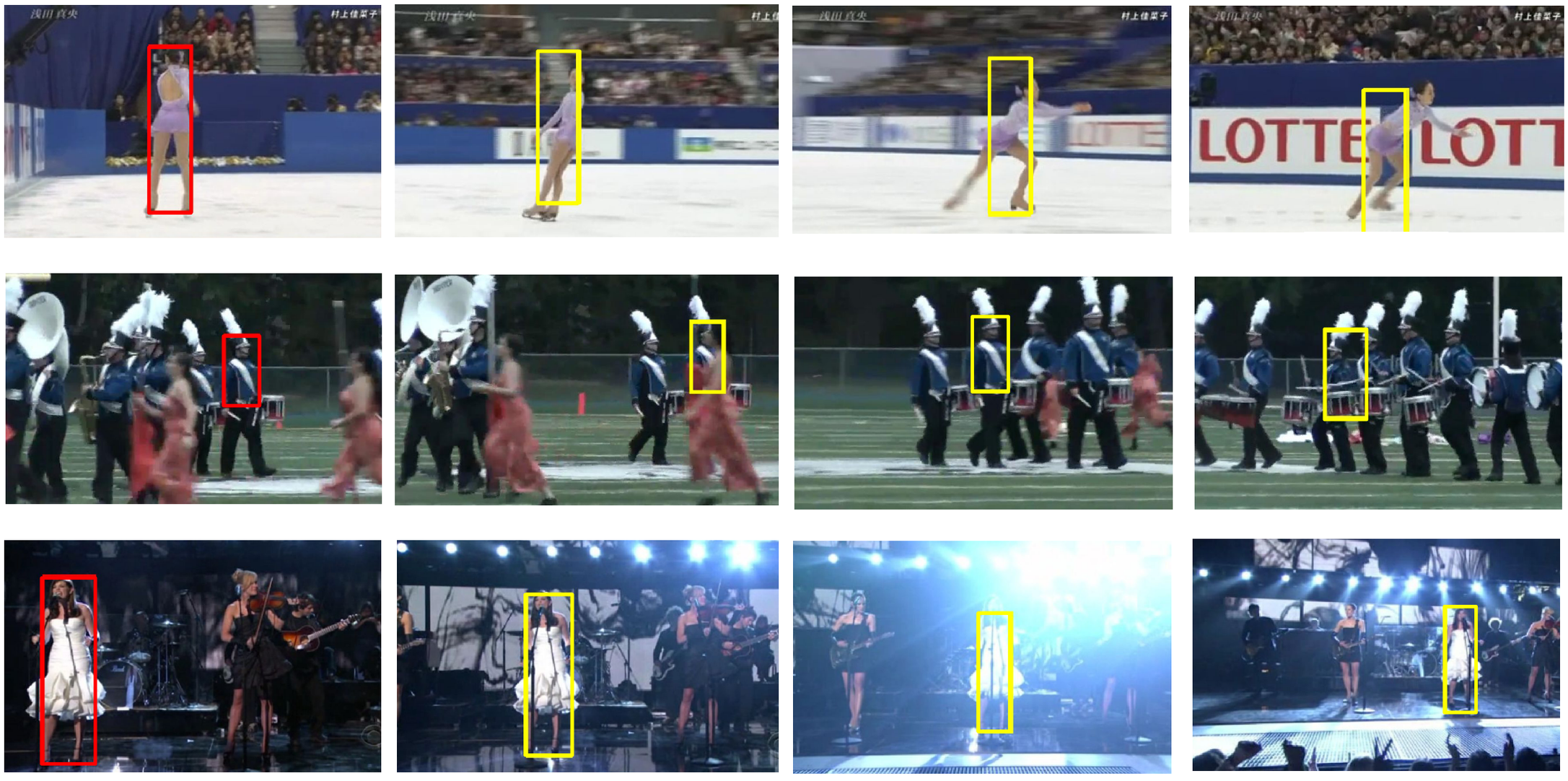
Fully Convolutional Siamese Networks



Frame 1 (init.) Frame 50 Frame 100 Frame 200

Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

Fully Convolutional Siamese Networks



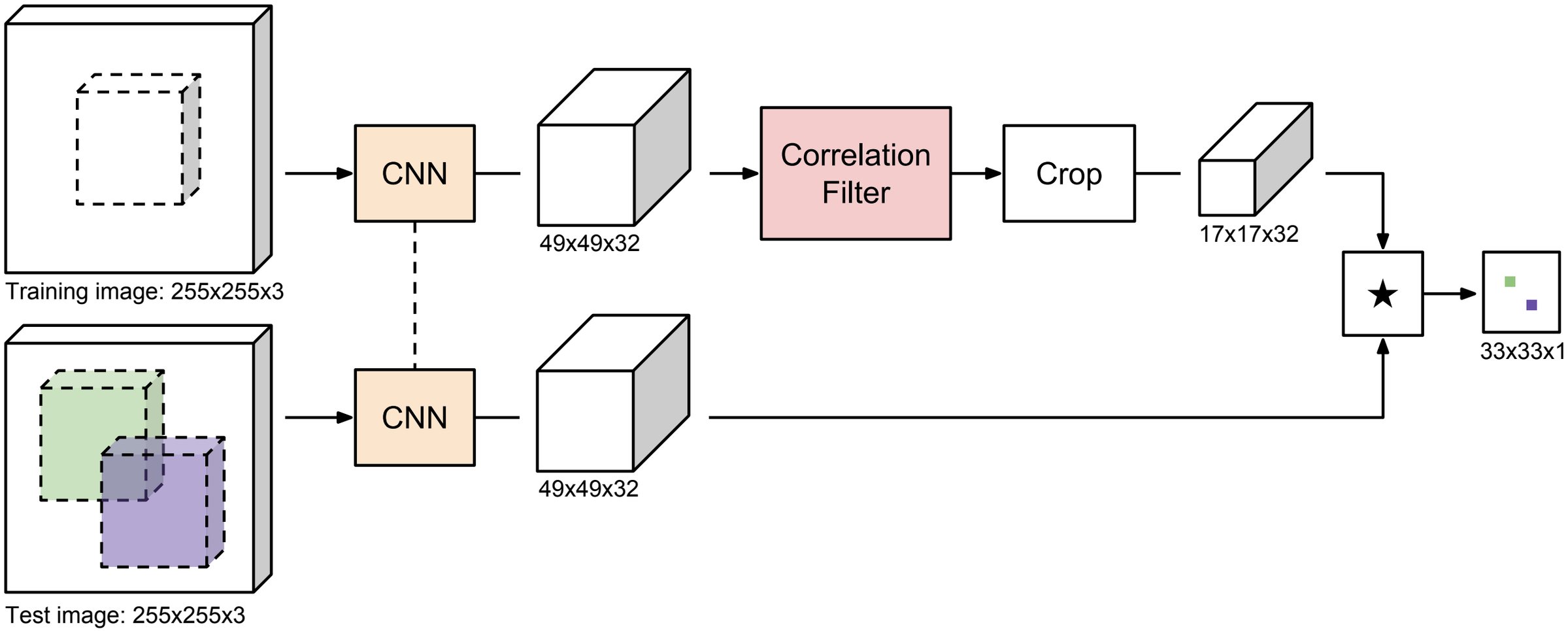
Frame 1 (init.) Frame 50 Frame 100 Frame 200

Bertinetto, Valmadre, Henriques, Vedaldi, Torr “Fully-Convolutional Siamese Networks for Object Tracking”, ArXiv 2016.

Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks

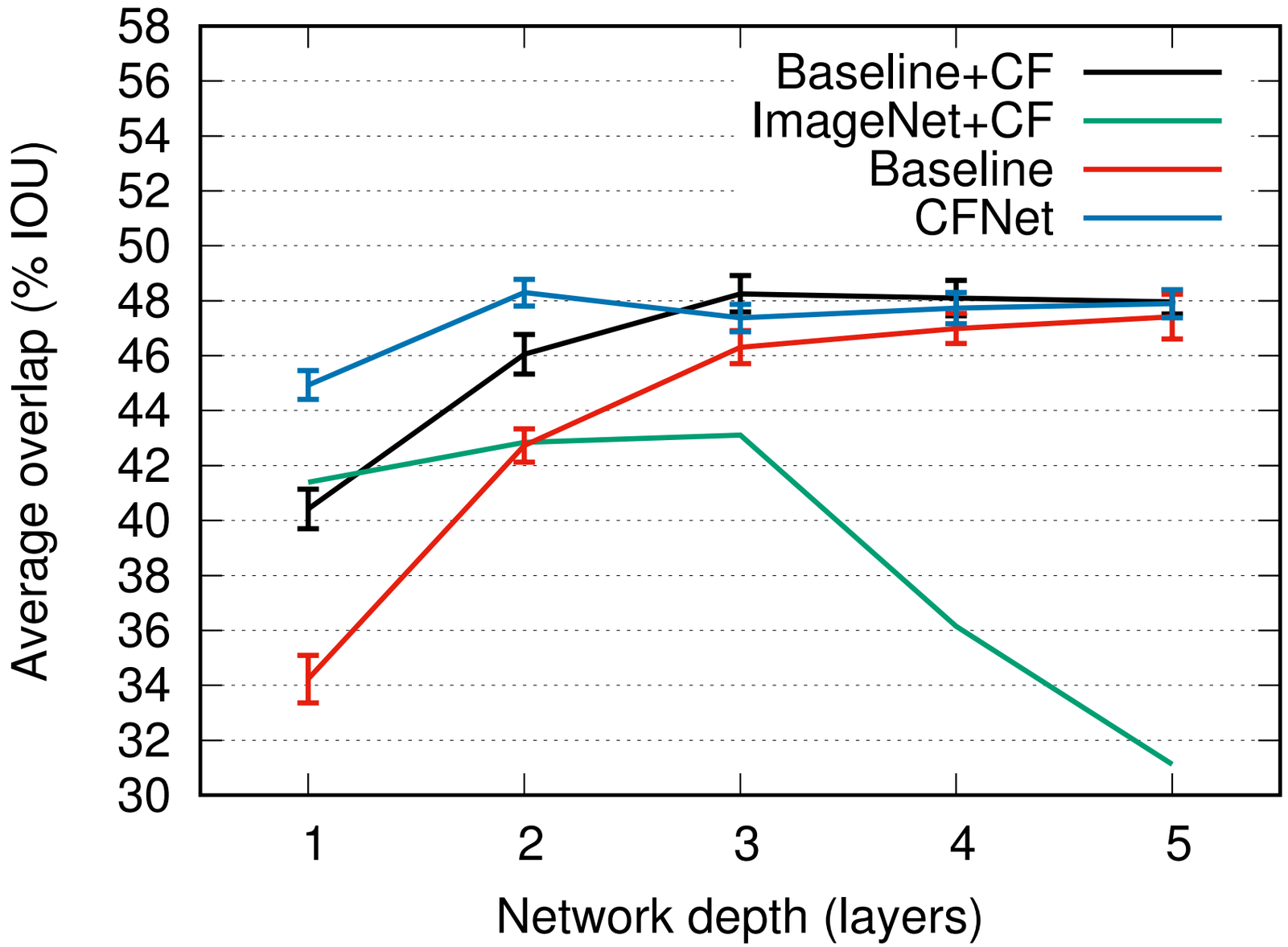
Correlation Filter networks



Valmadre, Bertinetto, Henriques, Vedaldi, Torr “End-to-end representation learning for Correlation Filter based tracking”, CVPR 2017.

Correlation Filter networks

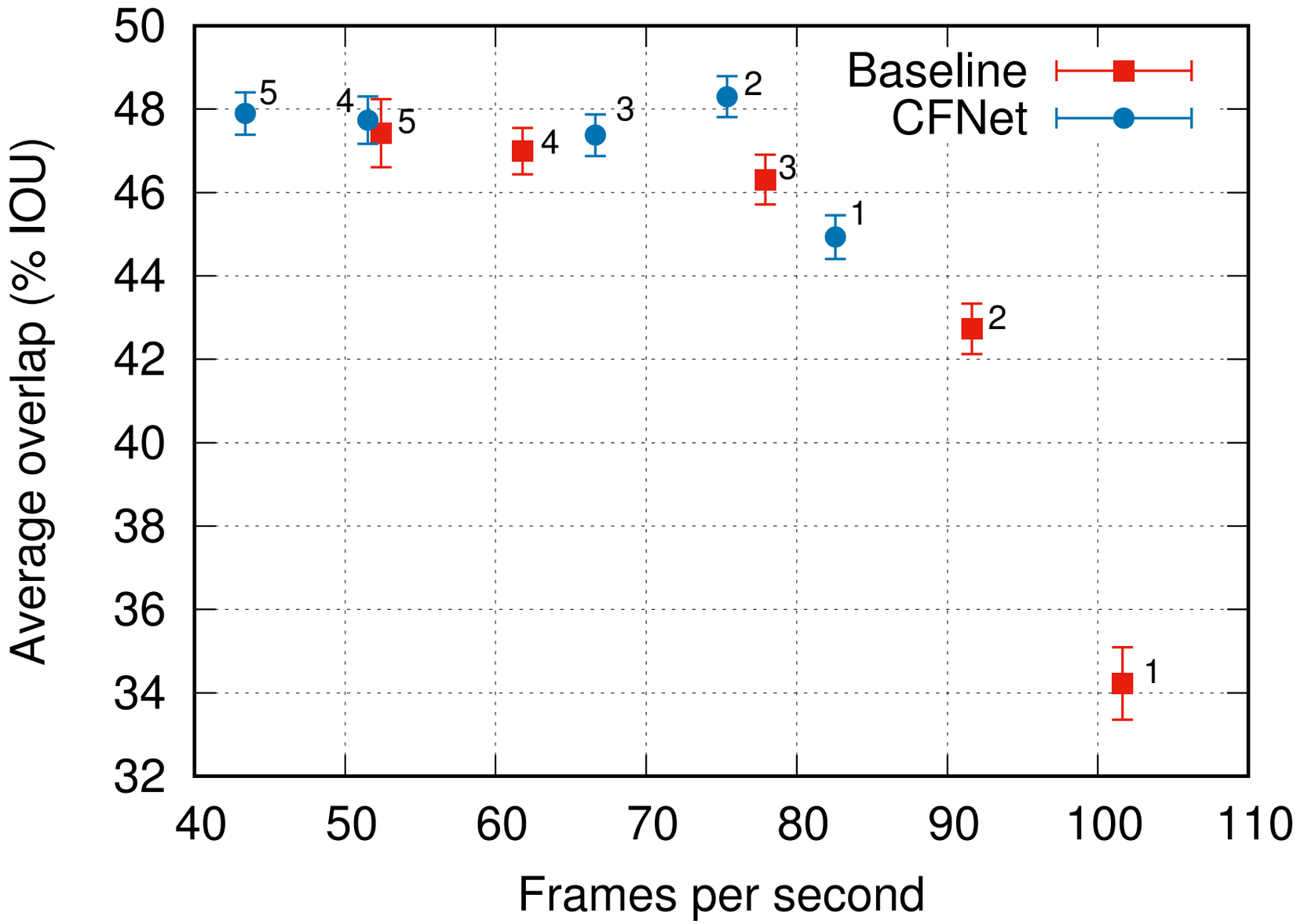
- Does it help to include Correlation Filter during training?



Valmadre, Bertinetto, Henriques, Vedaldi, Torr “End-to-end representation learning for Correlation Filter based tracking”, CVPR 2017.

Correlation Filter networks

- Does it help to include Correlation Filter during training?



Valmadre, Bertinetto, Henriques, Vedaldi, Torr “End-to-end representation learning for Correlation Filter based tracking”, CVPR 2017.

Today

- Convolution, circulant Toeplitz matrices, FFT
- Correlation filter
- Deep object tracking
  - Deep regression networks
  - Fully-convolutional Siamese networks
  - Correlation filter networks