Lecture 5: Statistical models of images

Today

- Review Fourier transforms
- Image statistics
- Texture synthesis

The Discrete Fourier transform

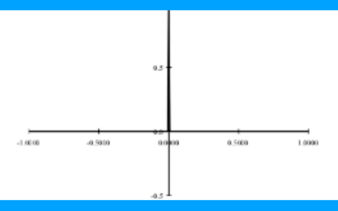
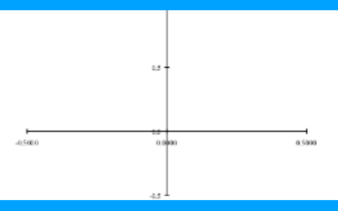
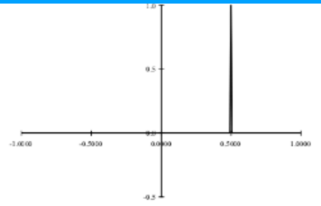
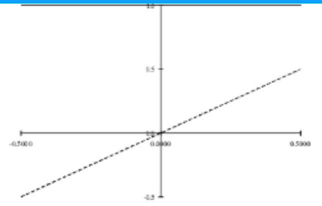
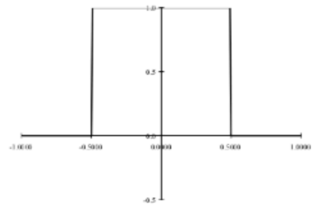
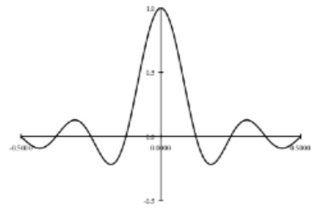
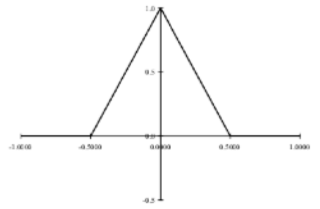
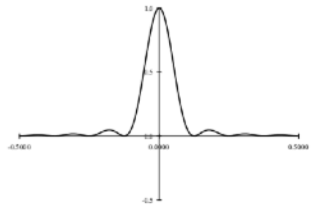
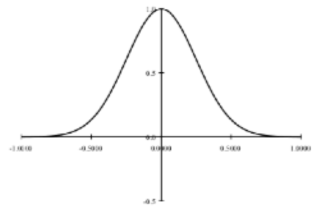
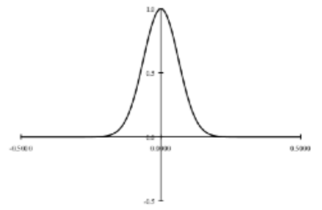
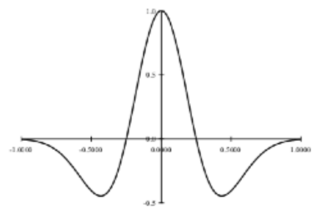
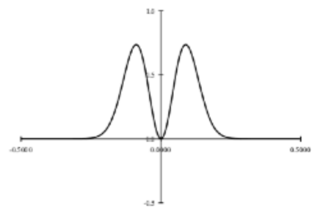
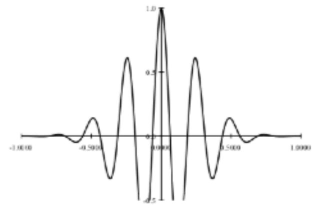
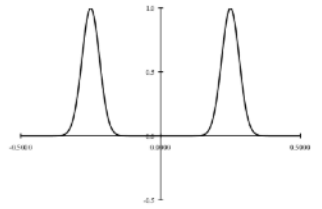
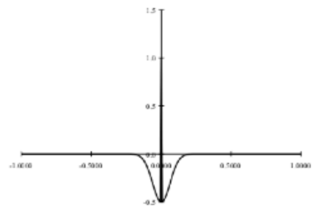
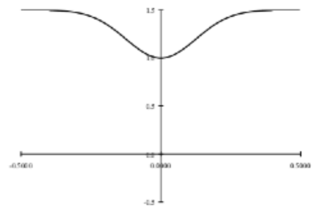
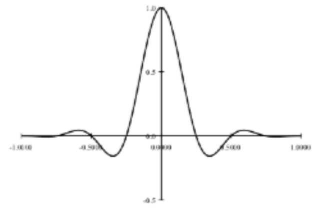
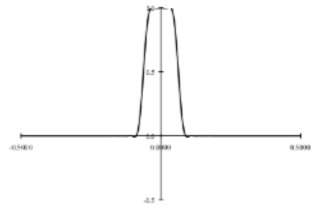
The Discrete Fourier Transform (DFT) transforms a signal *f[n]* into *F[u]* as:

Fourier coefficient
for frequency *u*

1D image

$$\boxed{F[u]} = \sum_{n=0}^{N-1} \boxed{f[n]} \, \boxed{\exp\left(-2\pi j \frac{un}{N}\right)}$$

Examples

Name		Signal		Transform	
impulse		$\delta(x)$	$\Leftrightarrow$	1	
shifted impulse		$\delta(x-u)$	$\Leftrightarrow$	$e^{-j\omega u}$	
box filter		$\mathrm{box}(x/a)$	$\Leftrightarrow$	$a\mathrm{sinc}(a\omega)$	
tent		$\mathrm{tent}(x/a)$	$\Leftrightarrow$	$a\mathrm{sinc}^2(a\omega)$	
Gaussian		$G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega;\sigma^{-1})$	
Laplacian of Gaussian		$(\frac{x^2}{\sigma^4}-\frac{1}{\sigma^2})G(x;\sigma)$	$\Leftrightarrow$	$-\frac{\sqrt{2\pi}}{\sigma}\omega^2 G(\omega;\sigma^{-1})$	
Gabor		$\cos(\omega_0 x)G(x;\sigma)$	$\Leftrightarrow$	$\frac{\sqrt{2\pi}}{\sigma}G(\omega\pm\omega_0;\sigma^{-1})$	
unsharp mask		$(1+\gamma)\delta(x)$ $-\gamma G(x;\sigma)$	$\Leftrightarrow$	$(1+\gamma)-$ $\frac{\sqrt{2\pi}\gamma}{\sigma}G(\omega;\sigma^{-1})$	
windowed sinc		$\mathrm{rcos}(x/(aW))$ $\mathrm{sinc}(x/a)$	$\Leftrightarrow$	(see Figure 3.29)	

From Szeliski 3.4

Examples

```
a = np.eye(8)[0]
np.fft.fft(a)
```

→

```
array([1.+0.j, 1.+0.j,
1.+0.j, 1.+0.j, 1.+0.j,
1.+0.j, 1.+0.j, 1.+0.j])
```

Examples

```
a = np.ones(8)
np.fft.fft(a)
```

→

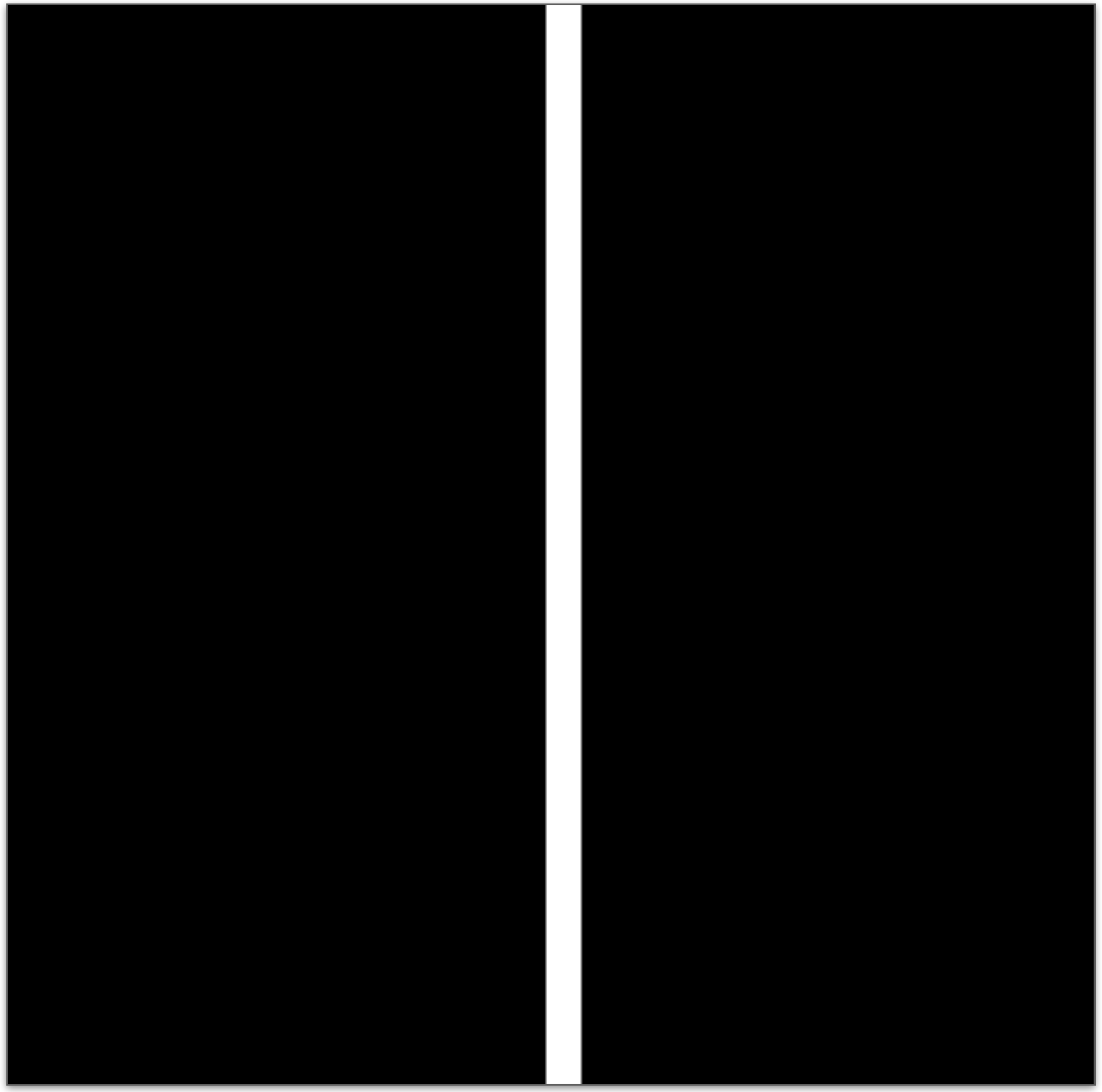
```
array([8.+0.j, 0.+0.j,
0.+0.j, 0.+0.j, 0.+0.j,
0.+0.j, 0.+0.j, 0.+0.j])
```

2D Discrete Fourier Transform

2D Discrete Fourier Transform (DFT) transforms an image *f* [*n*,*m*] into *F* [*u*,*v*] as:

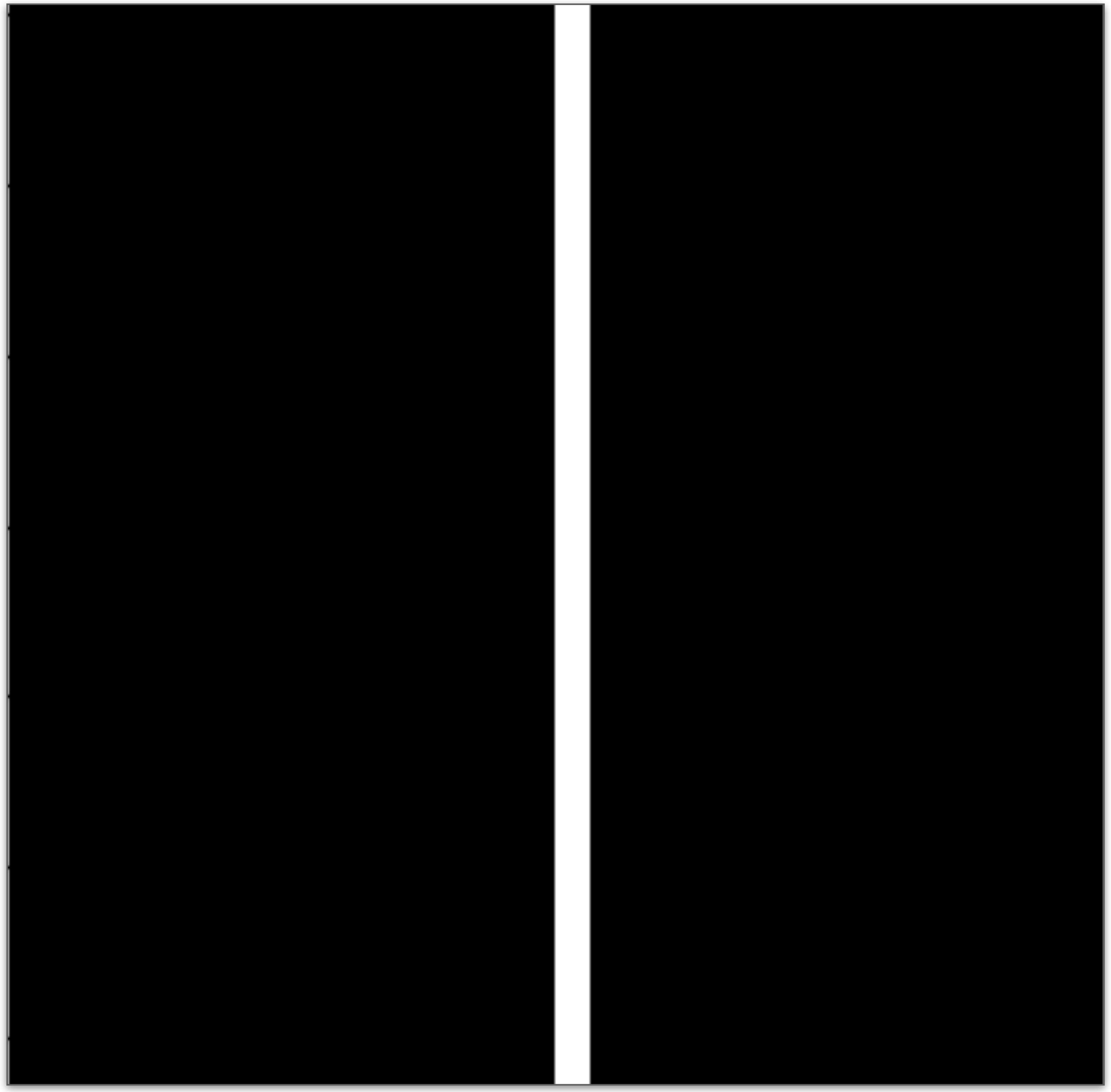
$$F[u,v] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f[n,m] \exp\left(-2\pi j \left(\frac{un}{N} + \frac{vm}{M} \right) \right)$$

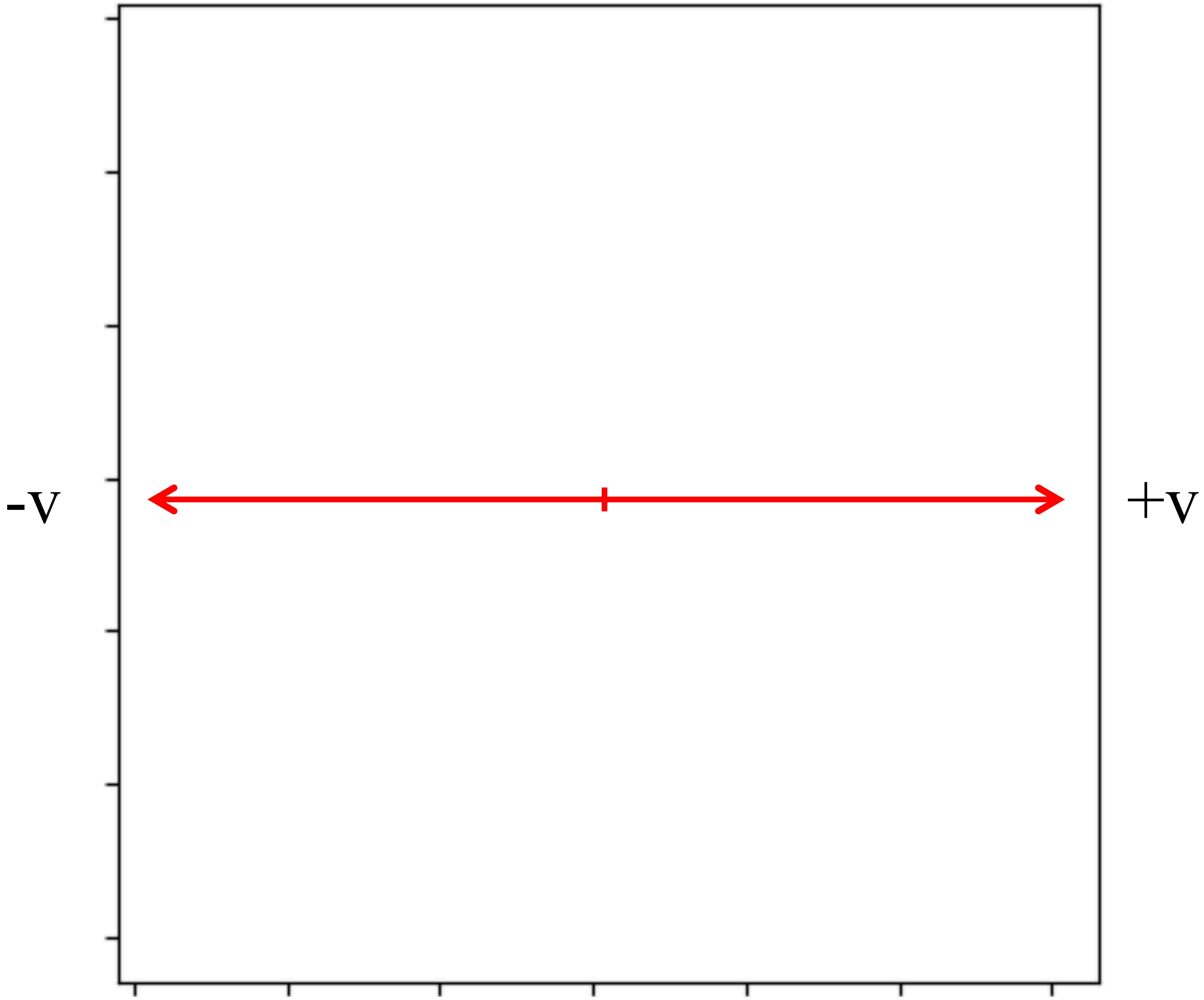
Examples



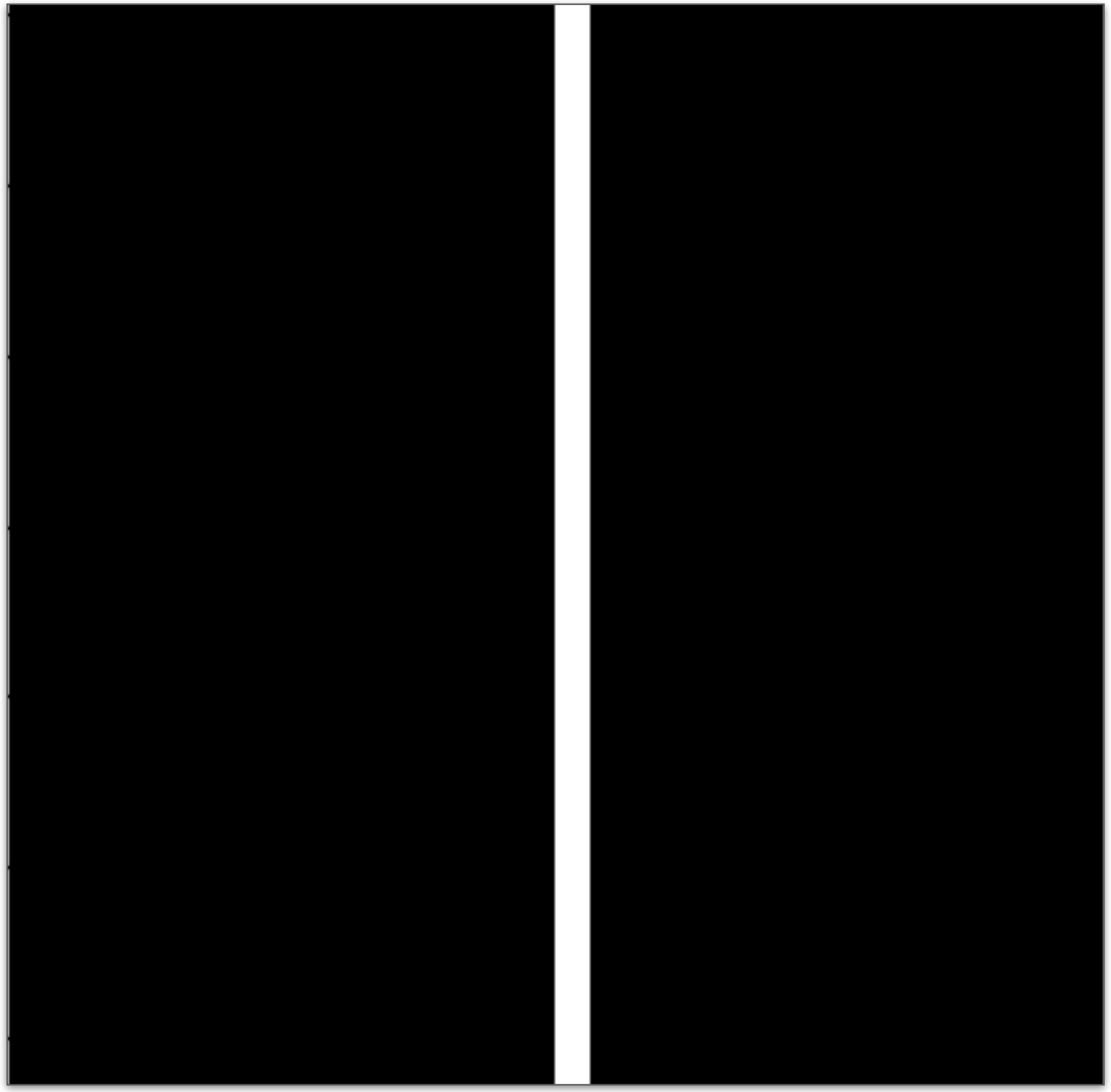
What is the 2D FFT of a line?

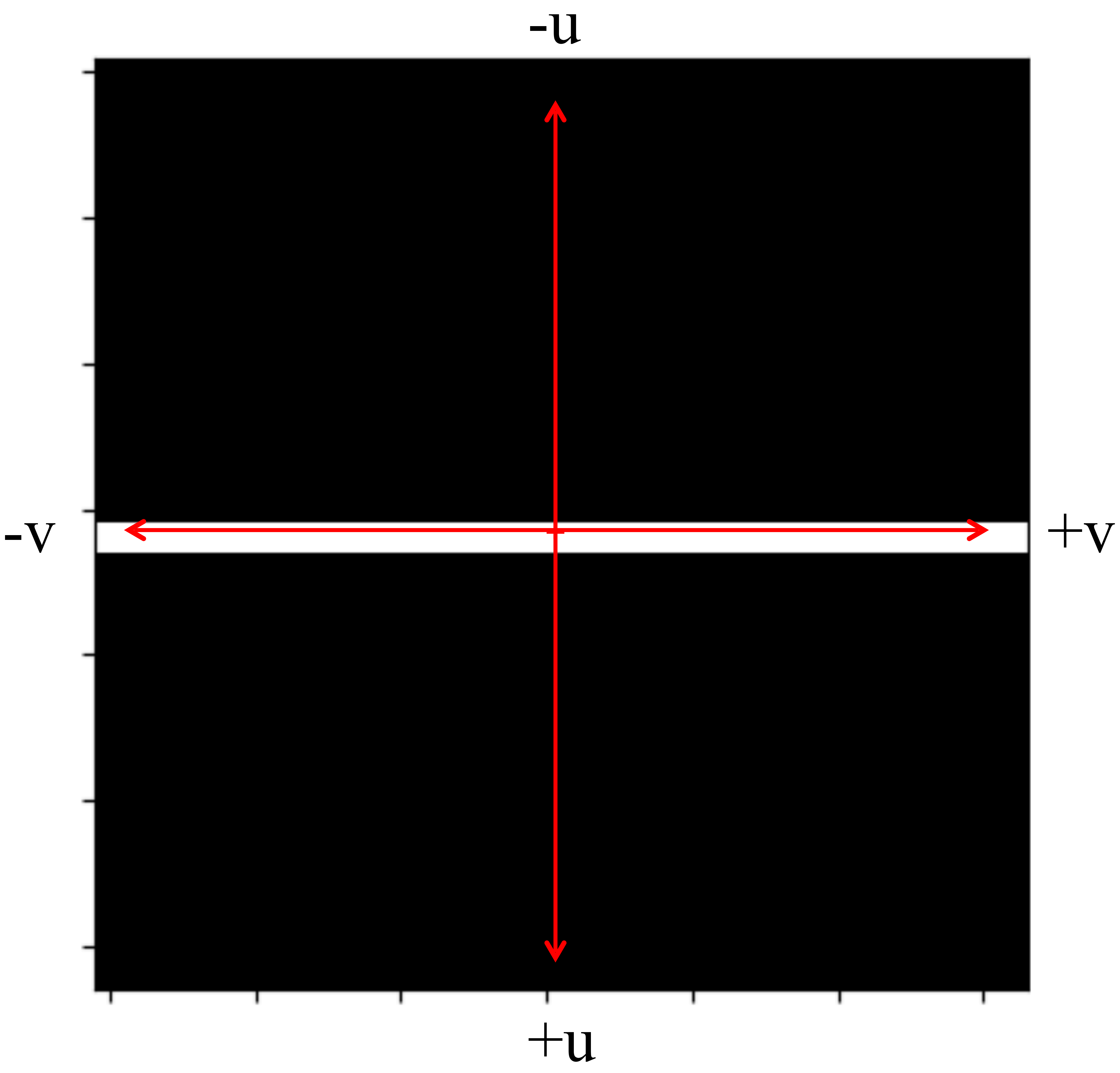
Fourier transform in x direction





Full 2D Fourier transform





What is a “natural” image?

“Natural” image

“Fake” image

Source: Torralba, Freeman, Isola

To appear in: Handbook of Video and Image Processing, 2nd edition
ed. Alan Bovik, ©Academic Press, 2005.

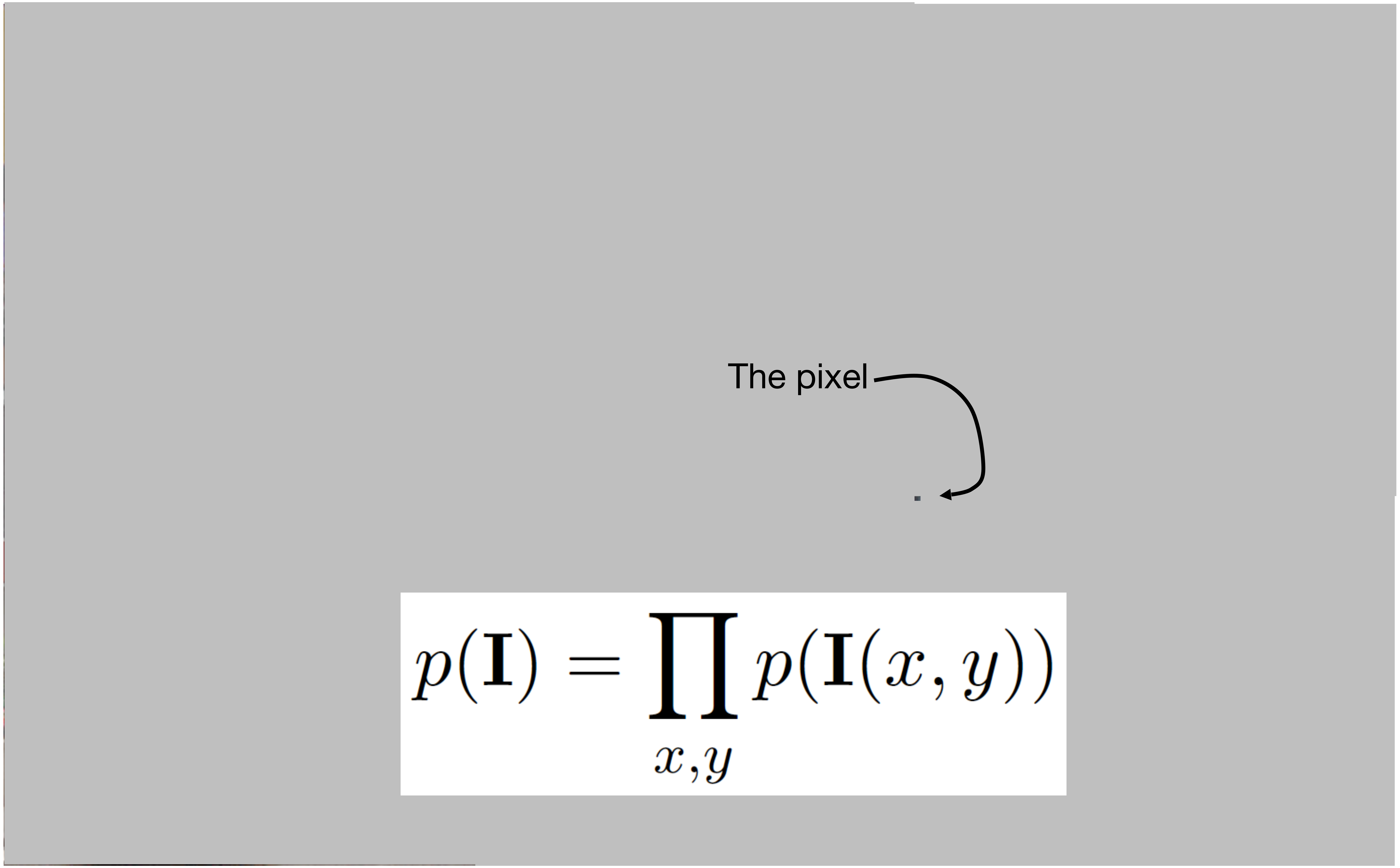
4.7 Statistical Modeling of Photographic Images

Eero P. Simoncelli
New York University

January 18, 2005

Source: Torralba, Freeman, Isola

Statistical modeling of images



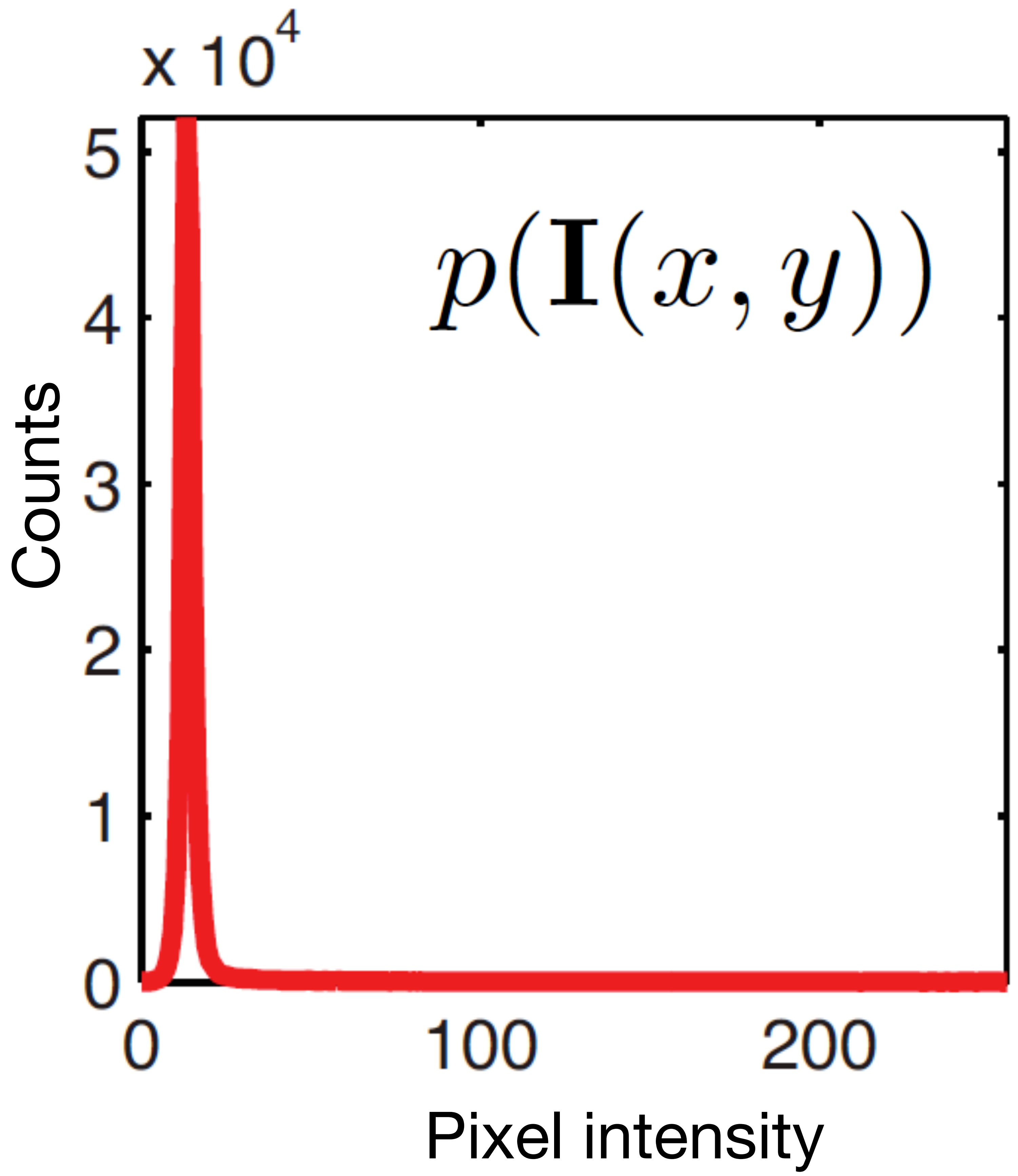
$$p(\mathbf{I}) = \prod_{x,y} p(\mathbf{I}(x,y))$$

Statistical modeling of images

$$p(\mathbf{I}) = \prod_{x,y} p(\mathbf{I}(x, y))$$

Assumptions:
- Independence: All pixels are independent.
- Stationarity: The distribution of pixel intensities does not depend on image location.

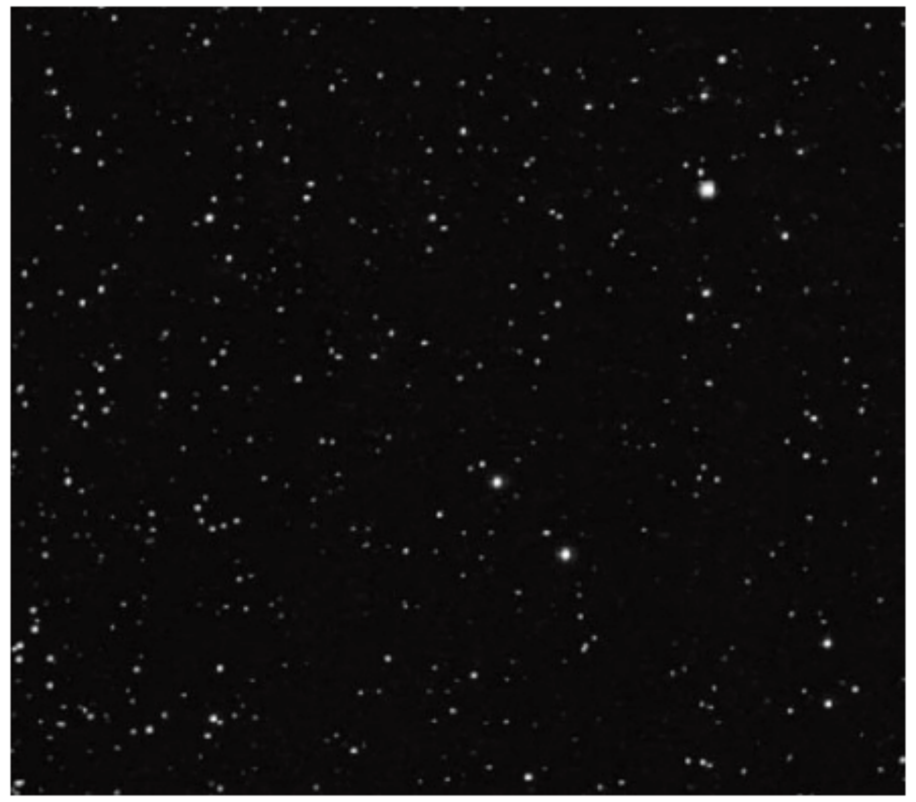
Fitting the model

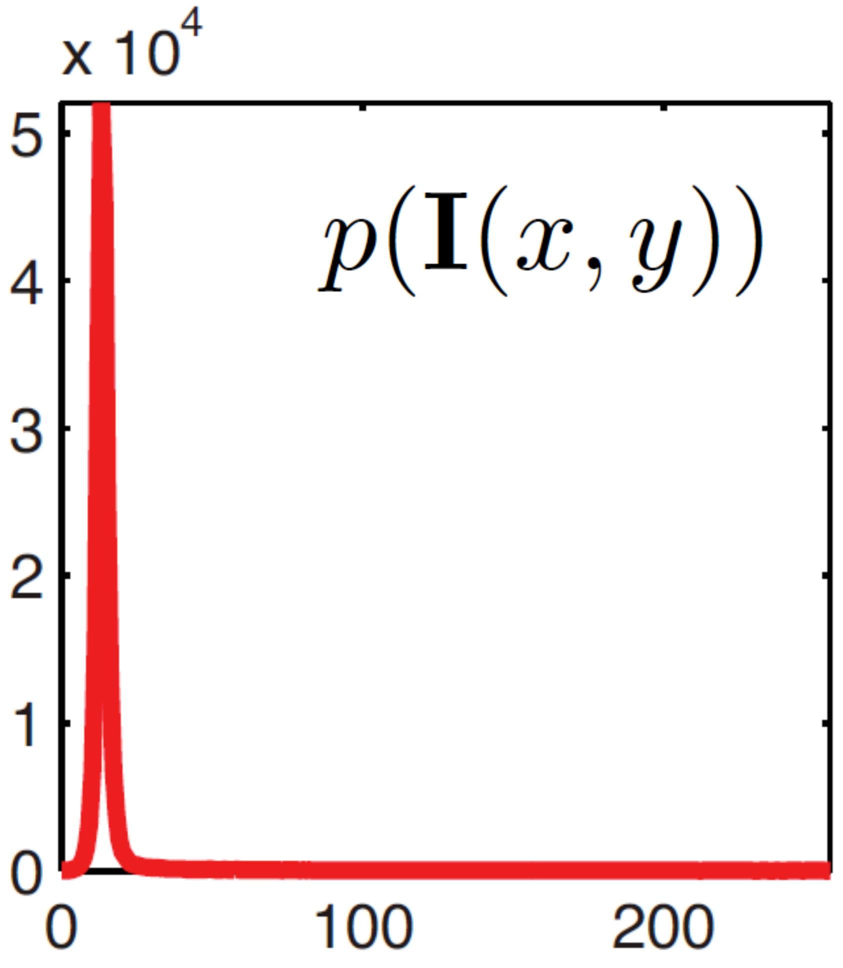


Source: Torralba, Freeman, Isola

Sampling new images

$$p(\mathbf{I}) = \prod_{x,y} p(\mathbf{I}(x,y))$$



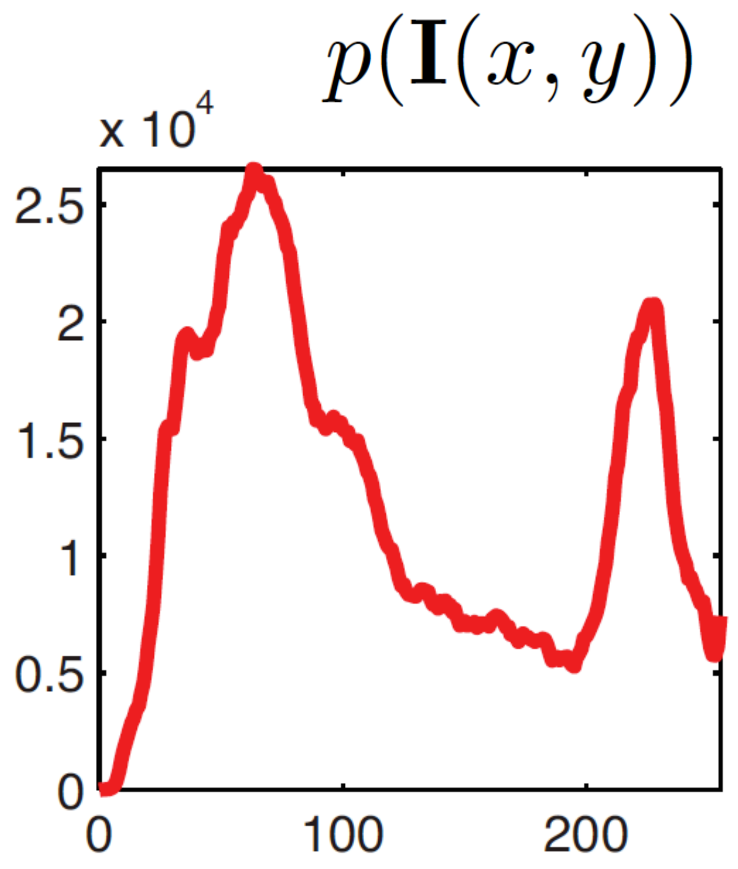


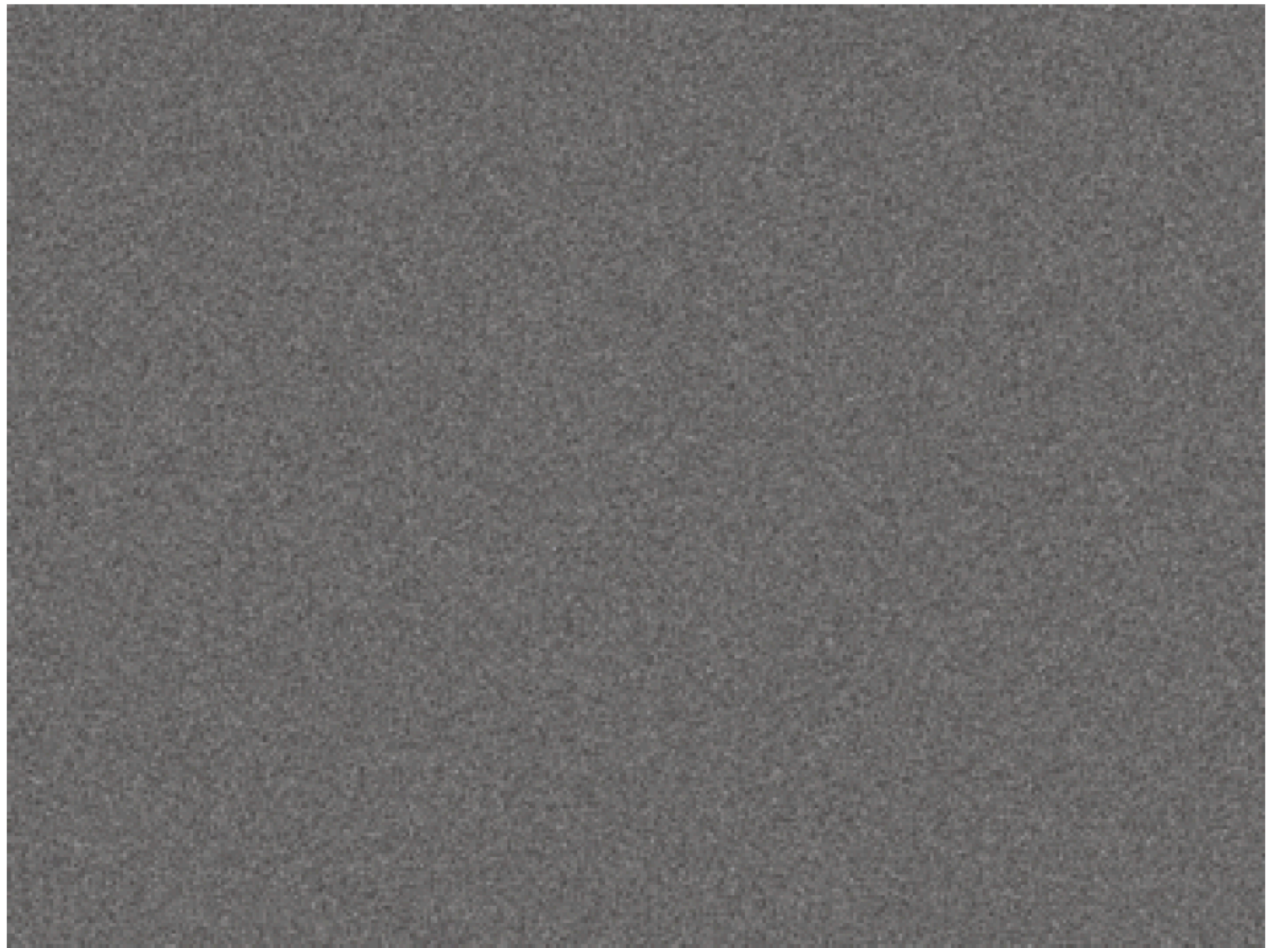
Sample

Source: Torralba, Freeman, Isola

Sampling new images

$$p(\mathbf{I}) = \prod_{x,y} p(\mathbf{I}(x,y))$$



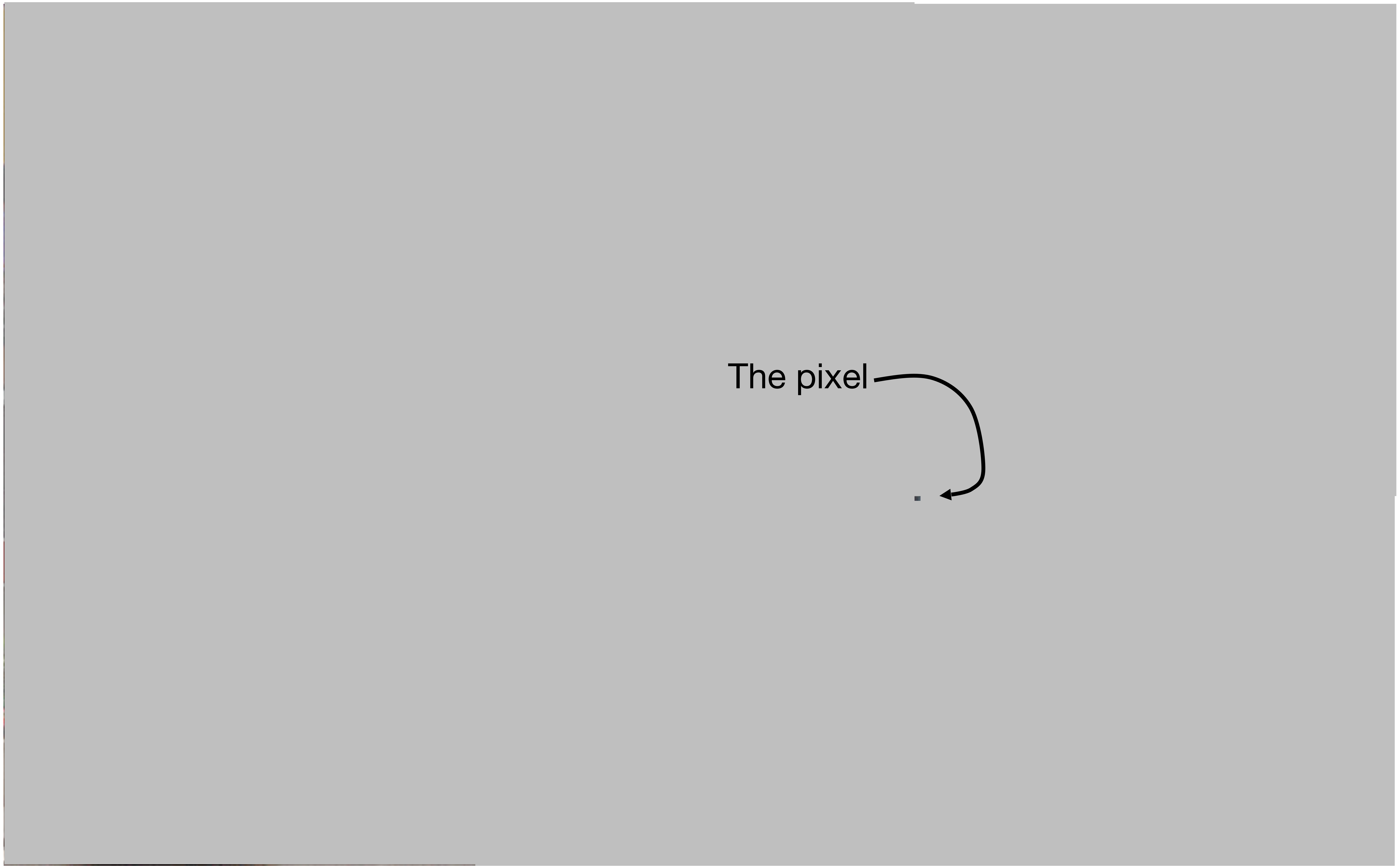


Sample

Doesn’t work very well!

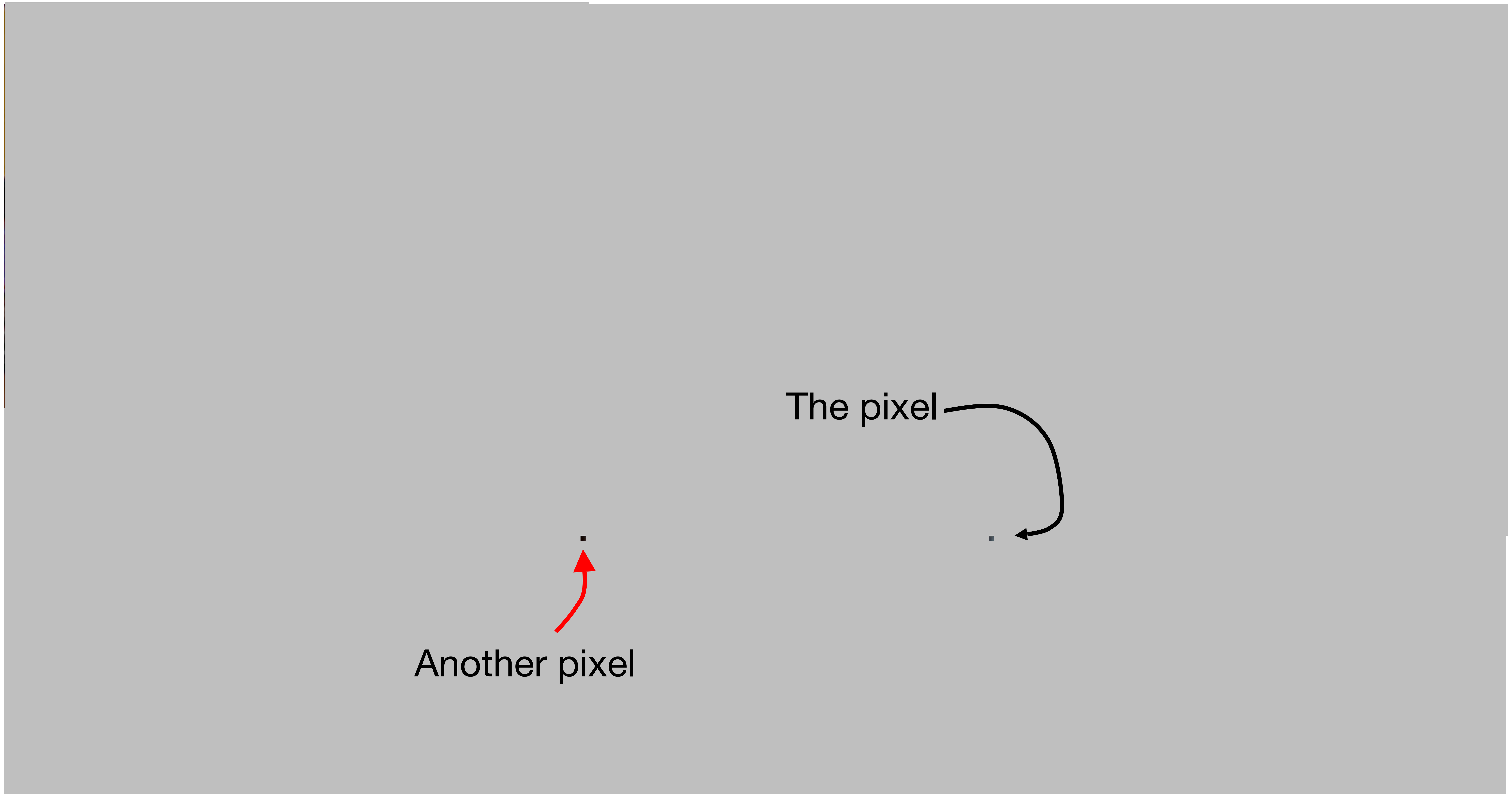
Source: Torralba, Freeman, Isola

Statistical modeling of images



Source: Torralba, Freeman, Isola

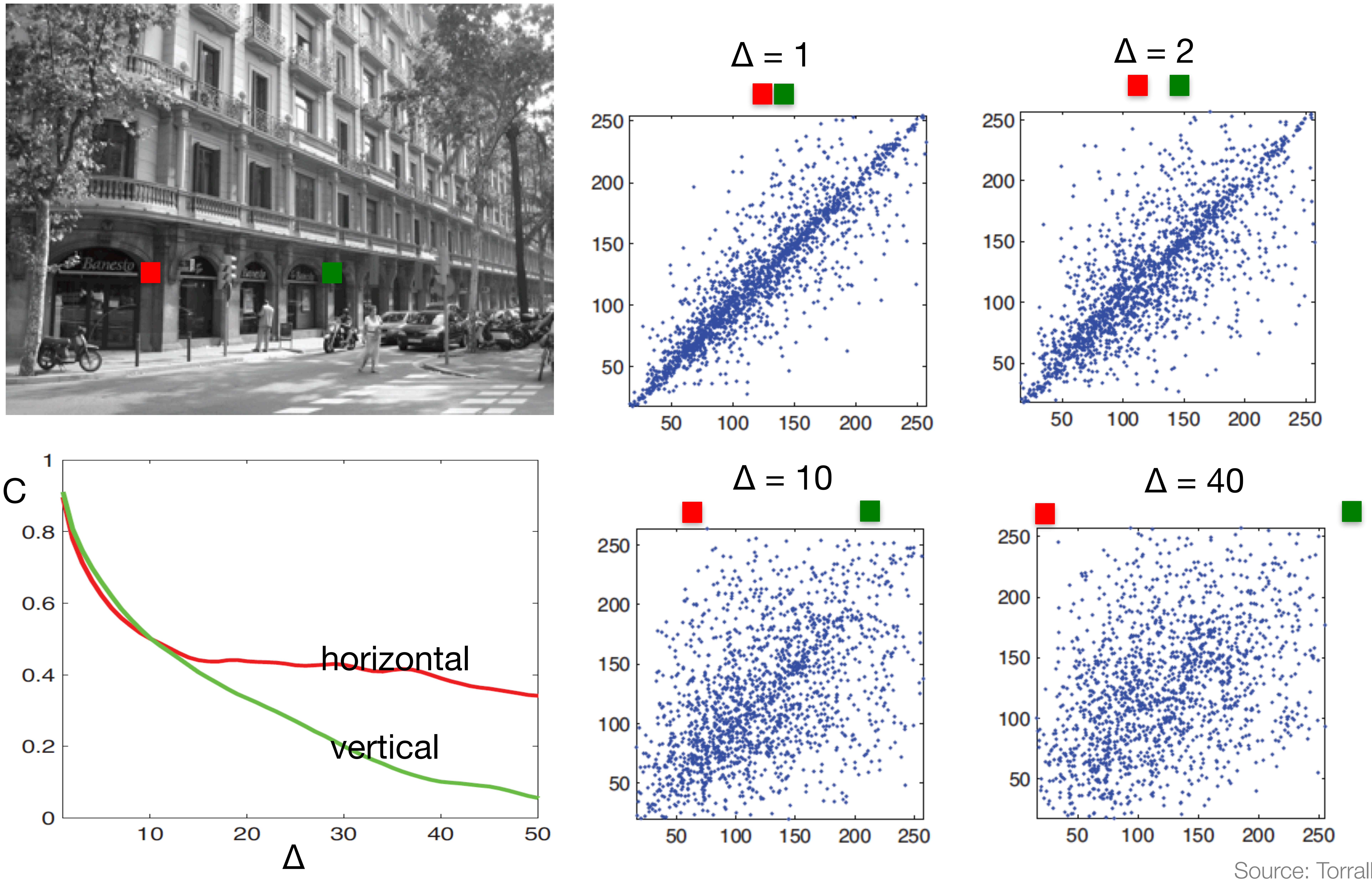
Statistical modeling of images



$$C(\Delta x, \Delta y) = \rho\left[\mathbf{I}(x + \Delta x, y + \Delta y), \mathbf{I}(x, y)\right]$$

Source: Torralba, Freeman, Isola

$$C(\Delta x, \Delta y) = \rho\left[\mathbf{I}(x + \Delta x, y + \Delta y), \mathbf{I}(x, y)\right]$$



Source: Torralba, Freeman, Isola

Gaussian model

We want a distribution that captures the correlation structure typical of natural images.

Let **C** be the covariance matrix of the image:

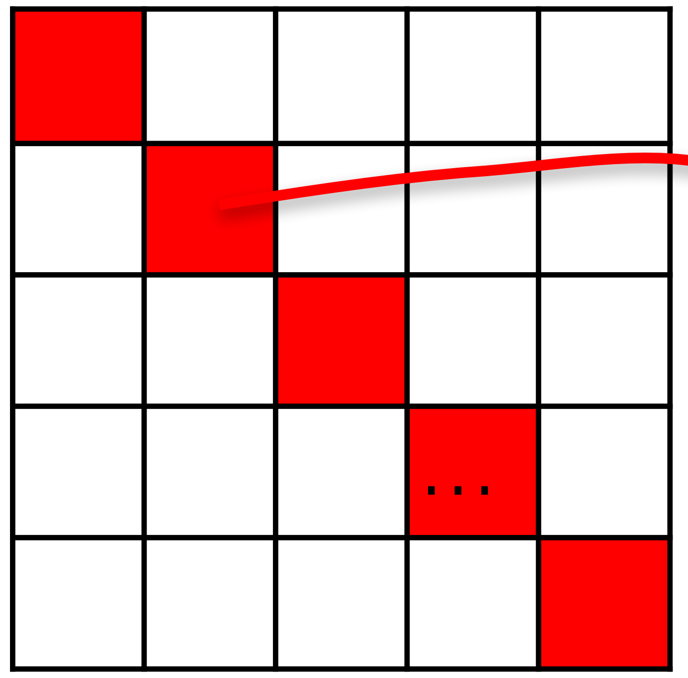
$$p(\mathbf{I}) = \exp\left(-\frac{1}{2}\mathbf{I}^T\mathbf{C}^{-1}\mathbf{I}\right) \qquad C = \begin{bmatrix} c_0 & c_1 & c_2 & & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & c_2 & & \vdots \\ & c_{n-1} & c_0 & c_1 & \ddots & \\ \vdots & \ddots & \ddots & \ddots & & c_2 \\ & & & & & c_1 \\ c_1 & \cdots & & c_{n-1} & & c_0 \end{bmatrix}$$

Stationarity assumption: Symmetrical circulant matrix

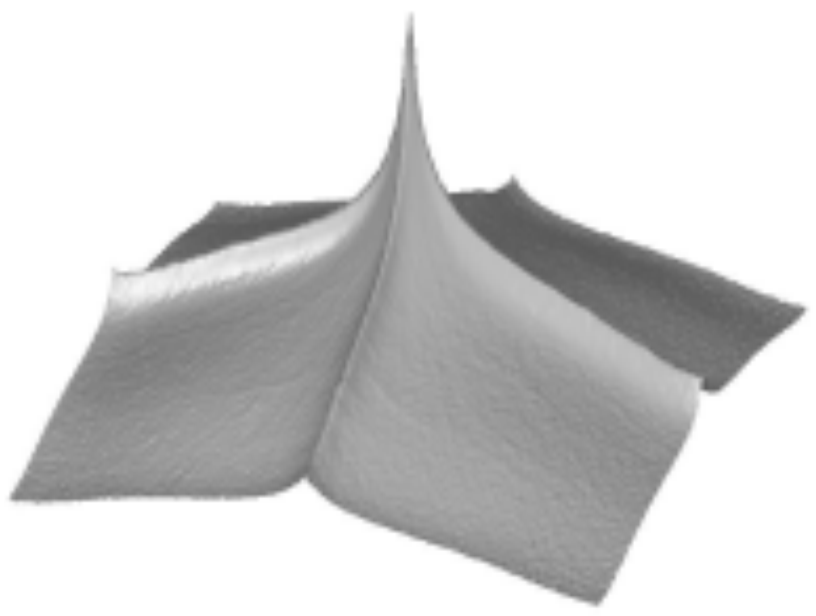
Diagonalization of circulant matrices: C = EDE^T

The eigenvectors are the Fourier basis

The eigenvalues are the squared magnitude of the Fourier coefficients

D= 

$$|\hat{\mathbf{I}}(v)|^2 \simeq \frac{1}{|v|^{2\alpha}}$$



Source: Torralba, Freeman, Isola

A remarkable property of natural images

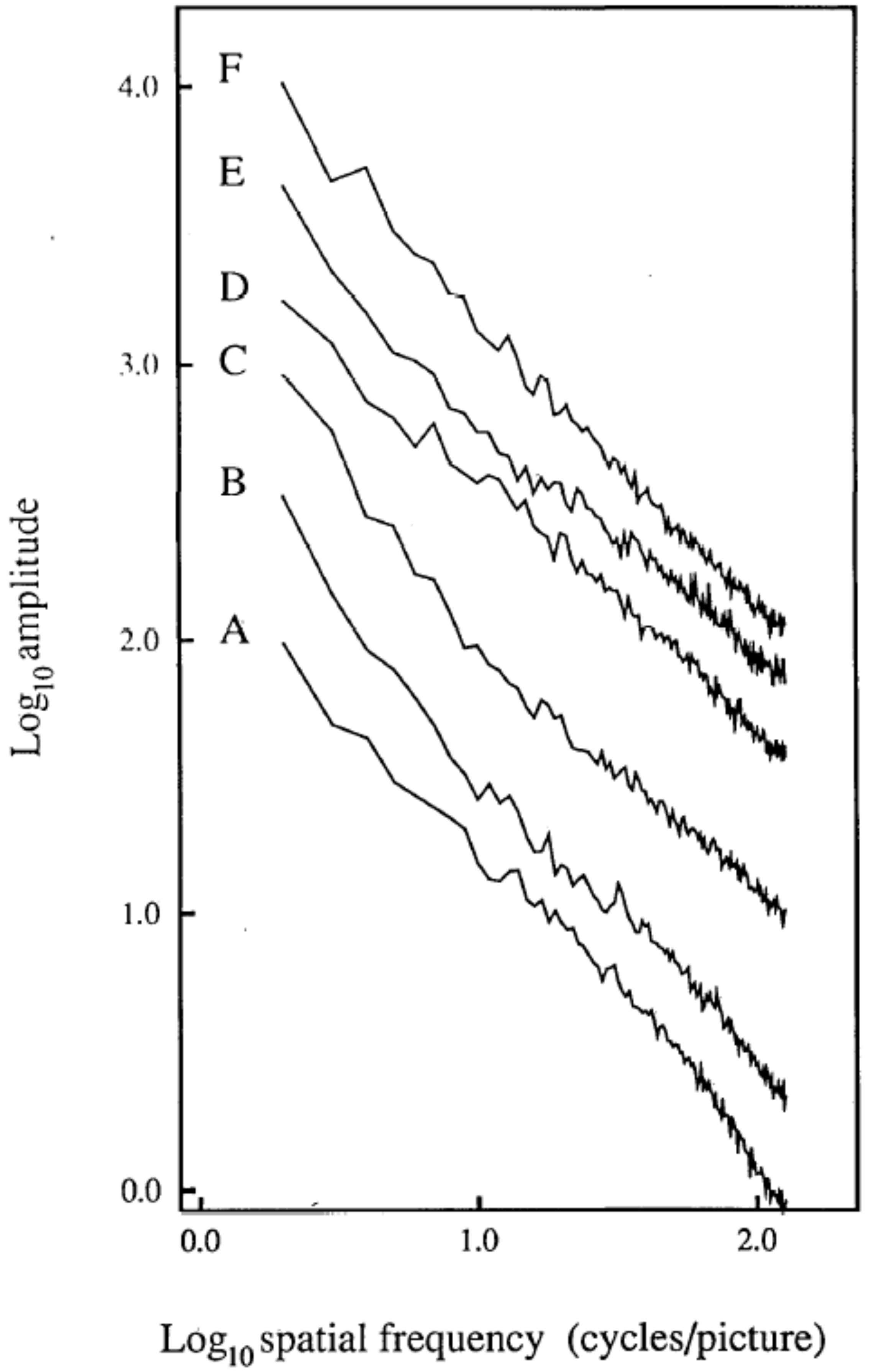
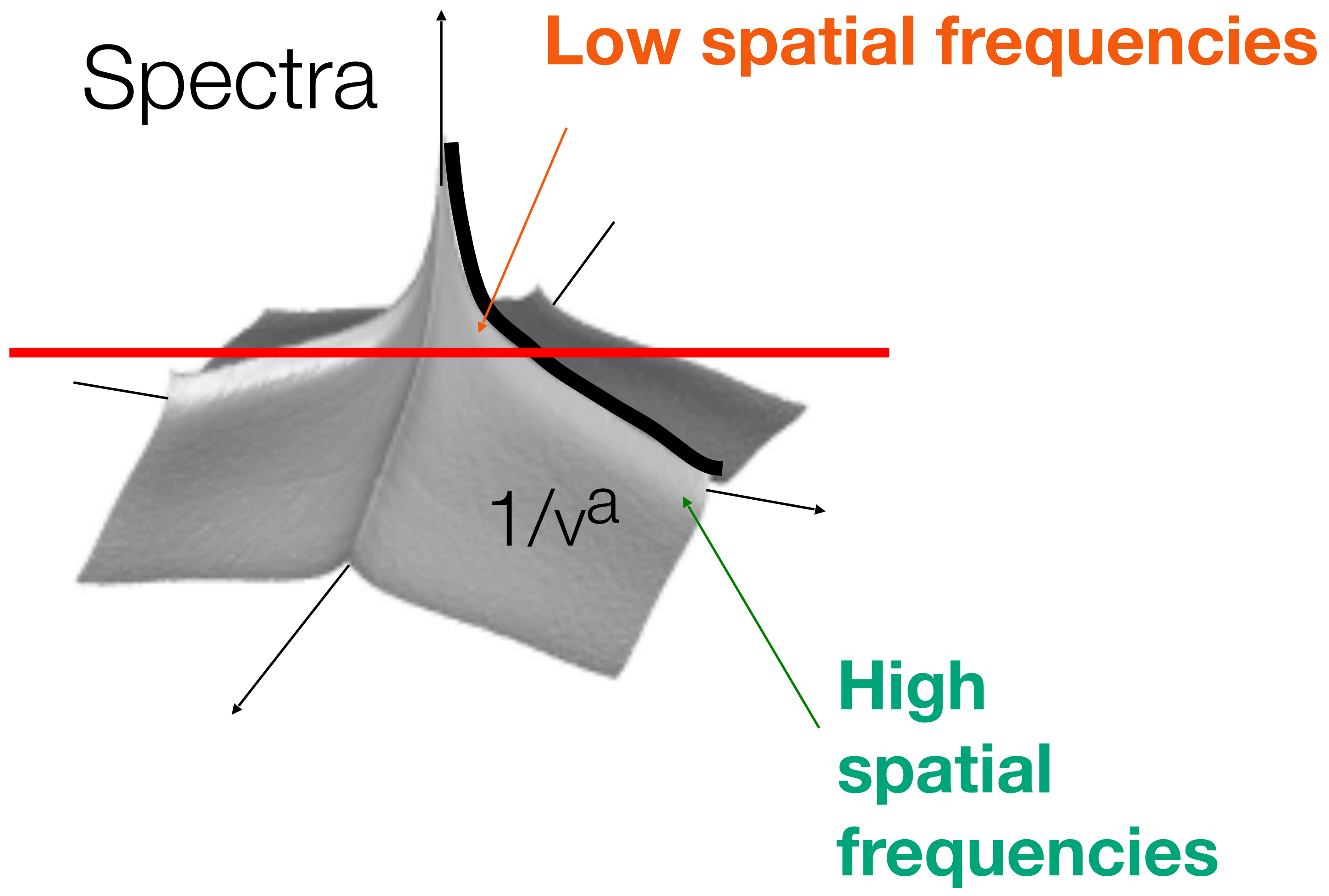


Fig. 8. Amplitude spectra for the six images A–F, averaged across all orientations. The spectra have been shifted up for clarity. On these log–log coordinates the spectra fall off by a factor of roughly $1/f$ (a slope of –1). Therefore the power spectra fall off as $1/f^2$.

Power spectra
fall off as

$$|\hat{\mathbf{I}}(v)| \simeq \frac{1}{|v|^{\alpha}}$$

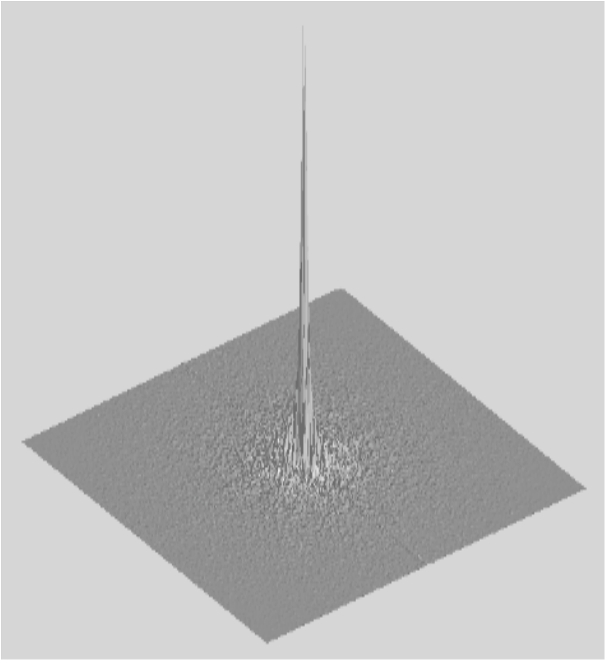


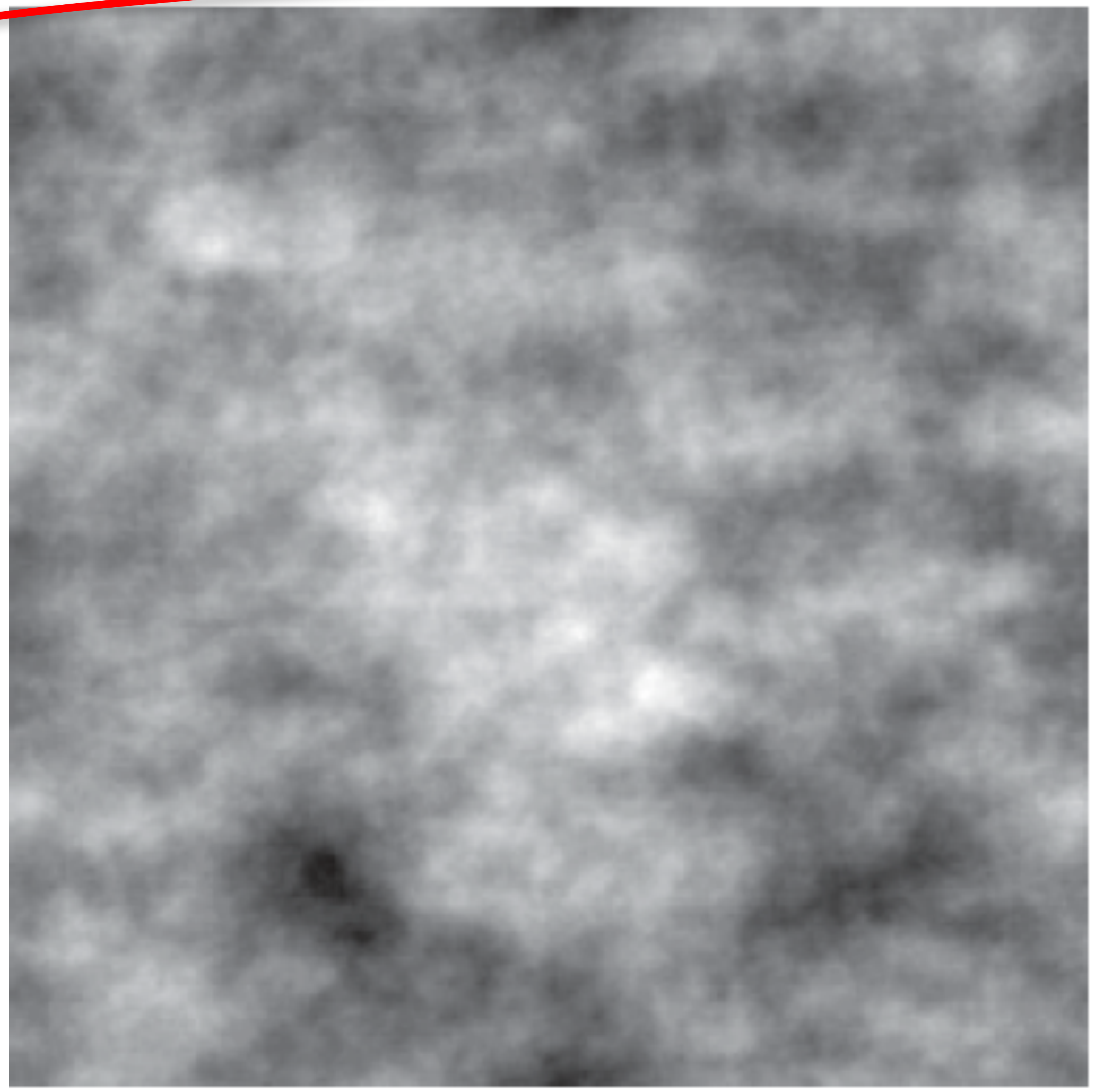
D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," J. Opt. Soc. Am. A **4**, 2379- (1987)

Source: Torralba, Freeman, Isola

Sampling new images

$$p(\mathbf{I}) = \exp\left(-\frac{1}{2}\mathbf{I}^T\mathbf{C}^{-1}\mathbf{I}\right)$$



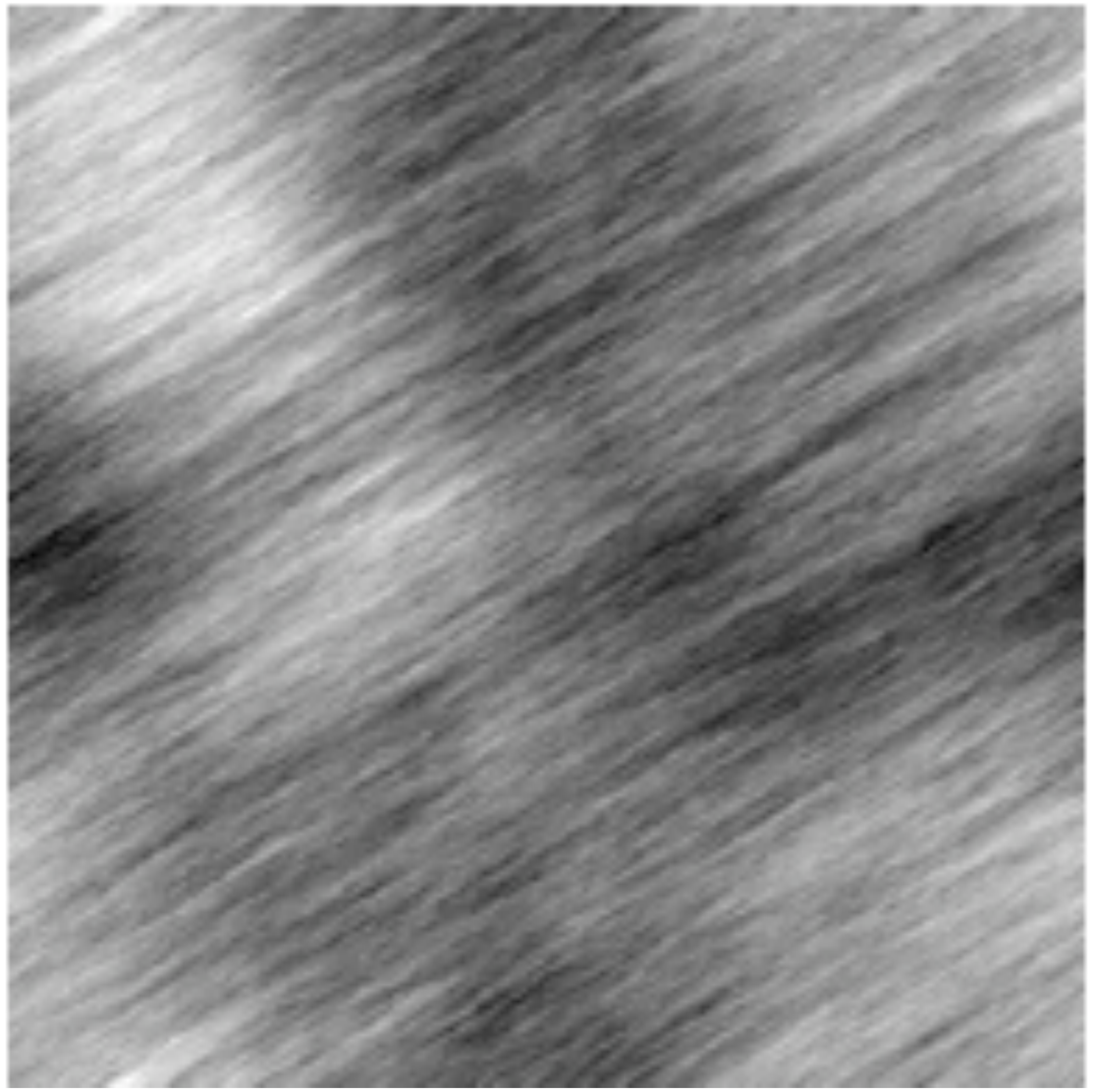


Sample

Fit the Gaussian image model to each of the images in the top row, then draw another random sample (with random phase), you get the bottom row.

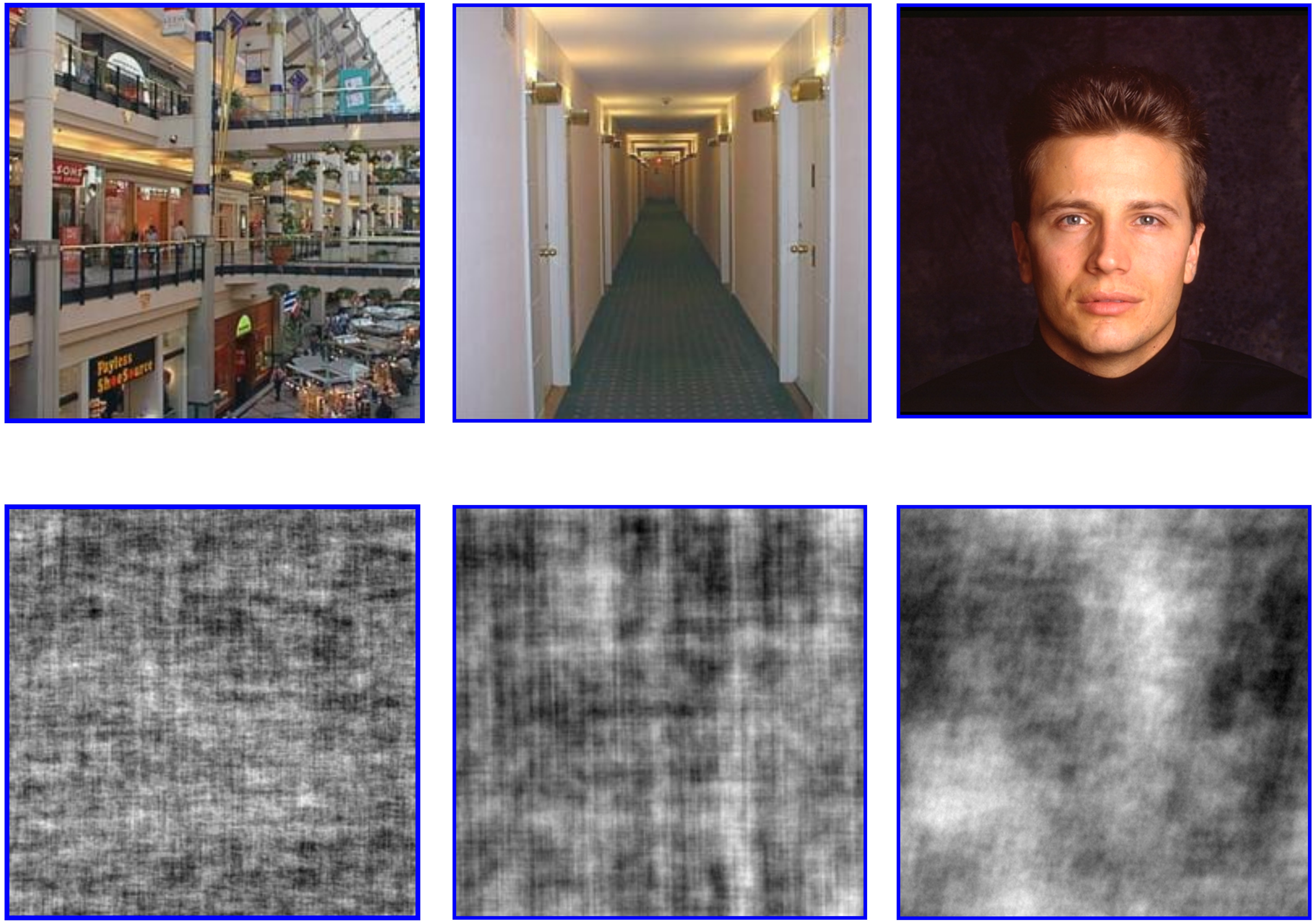
Source: Torralba, Freeman, Isola

Sampling new images



Source: Torralba, Freeman, Isola

Doesn't always work well!

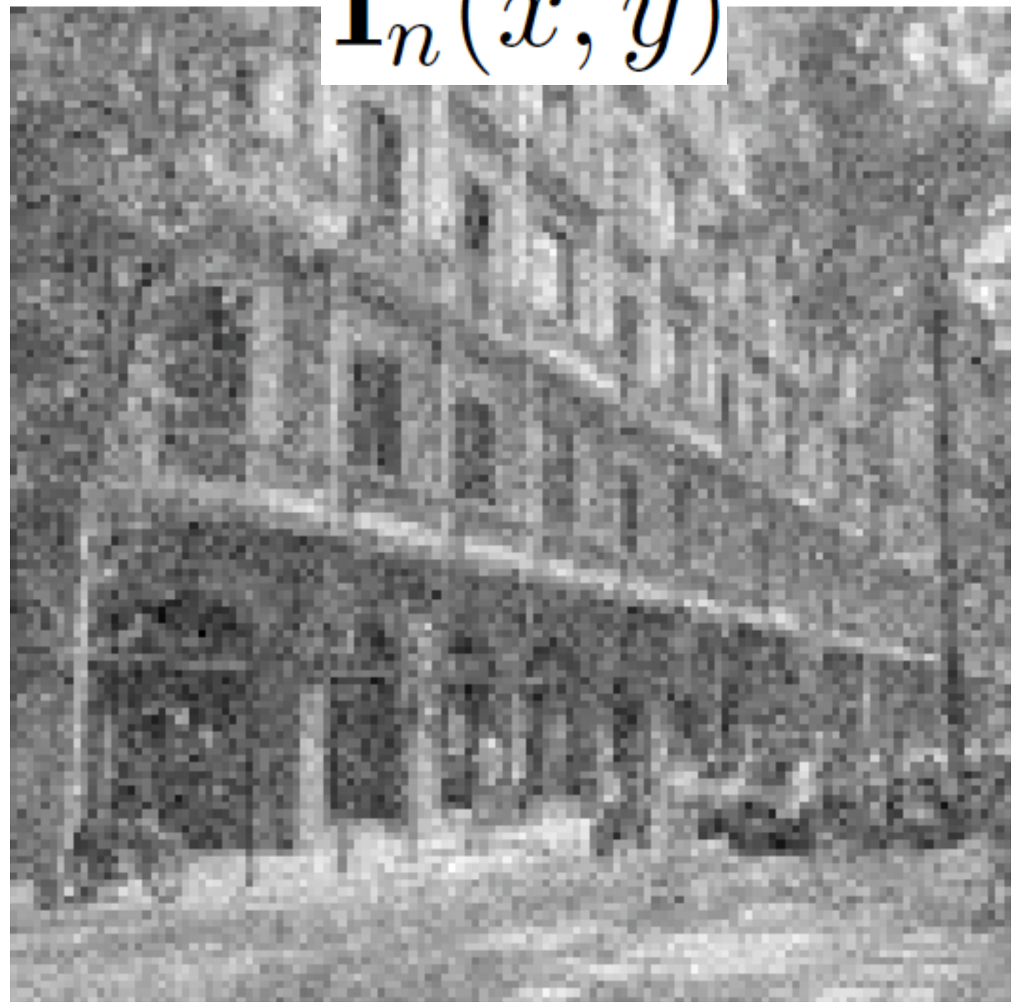


Source: Torralba, Freeman, Isola

Denoising

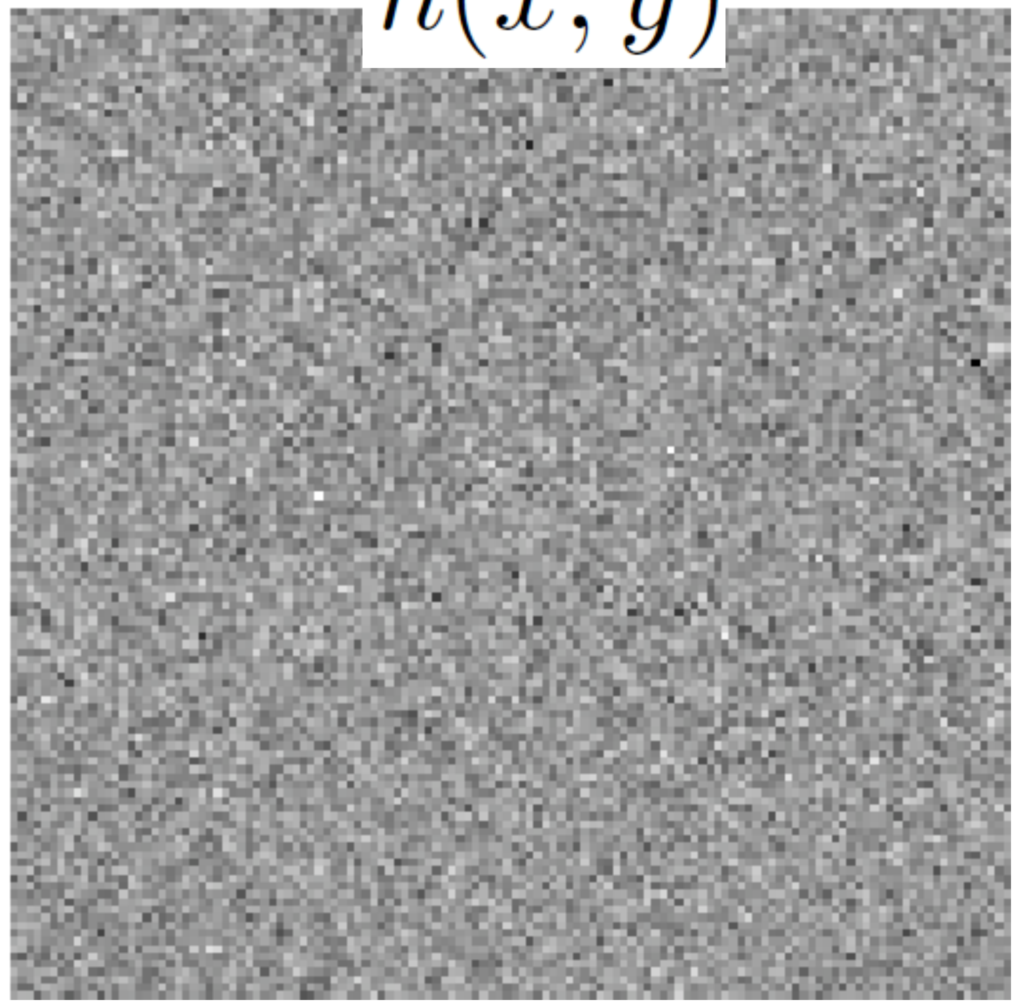
Decomposition of a noisy image

$\mathbf{I}_n(x, y)$



$=$

$n(x, y)$



$+$

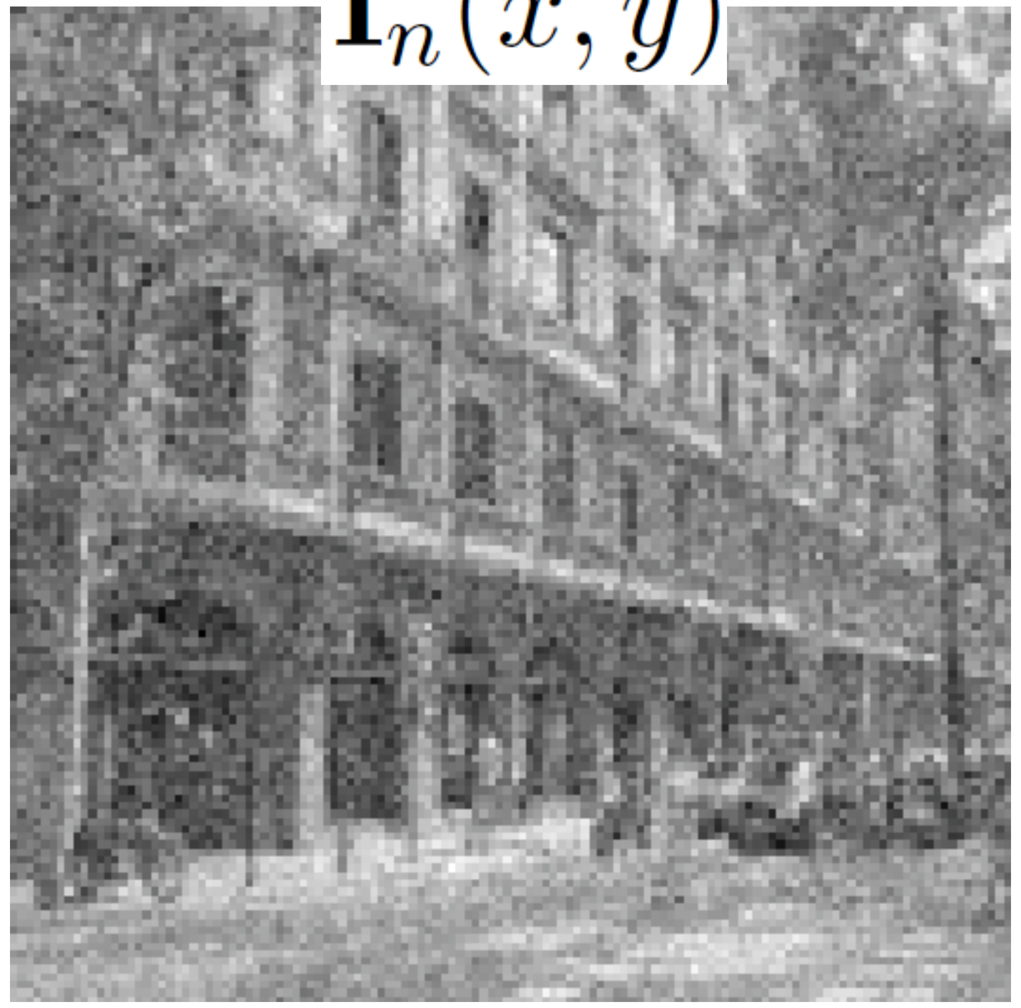
$\mathbf{I}(x, y)$

Source: Torralba, Freeman, Isola

Denoising

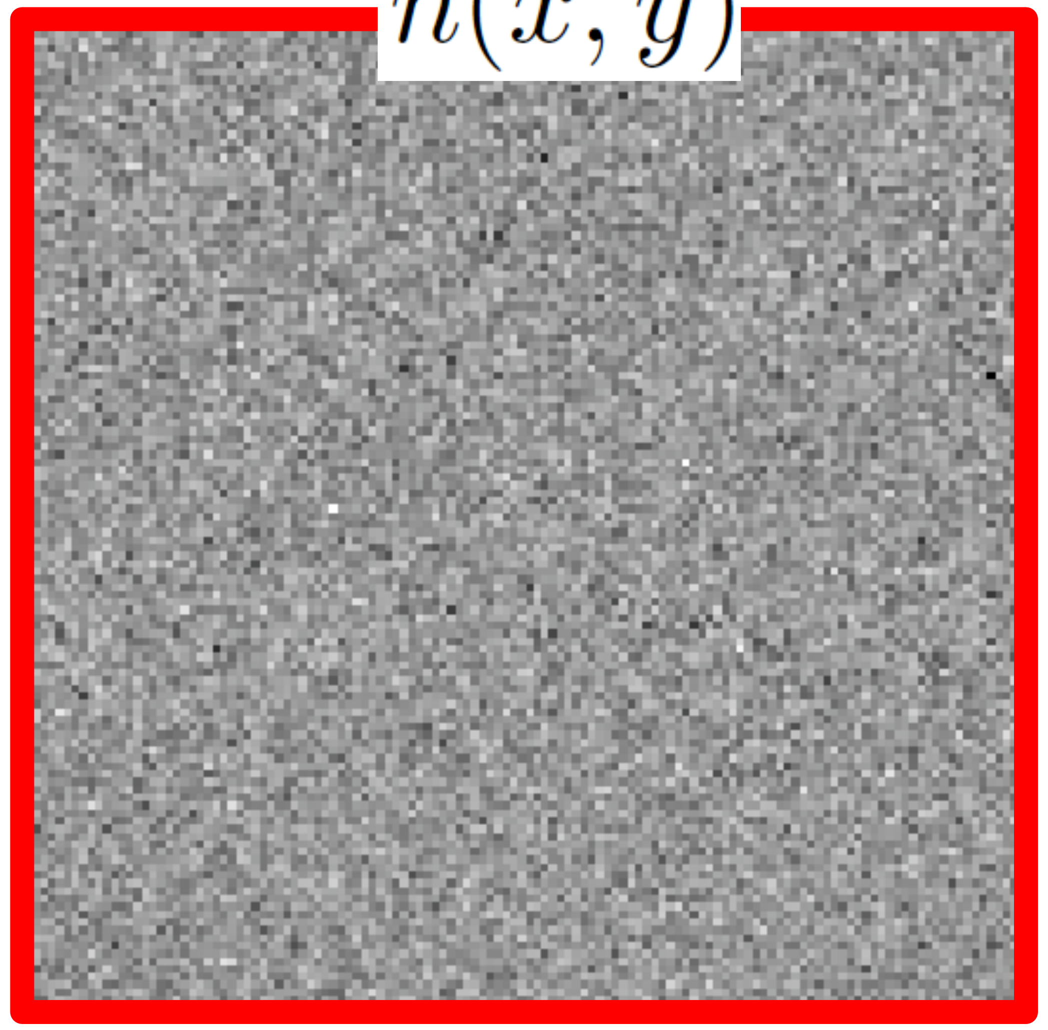
Decomposition of a noisy image

$\mathbf{I}_n(x, y)$



$=$

$n(x, y)$



White Gaussian noise: $N(0, \sigma_n^2)$

$+$

$\mathbf{I}(x, y)$

Natural image

Find I(x,y) that maximizes the posterior (maximum a posteriory, MAP):

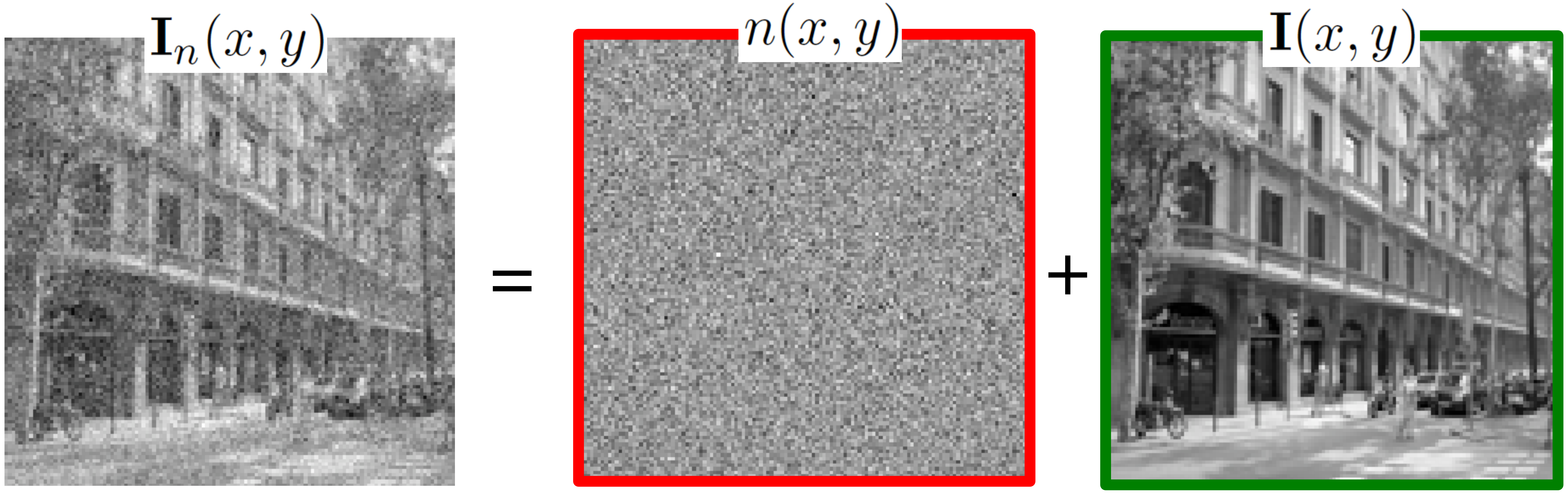
$$\max_{\mathbf{I}} p(\mathbf{I}|\mathbf{I}_n) = \max_{\mathbf{I}} \underbrace{\boxed{p(\mathbf{I}_n|\mathbf{I})}}_{\text{likelihood}} \times \underbrace{\boxed{p(\mathbf{I})}}_{\text{prior}}$$

(Bayes' theorem)

Source: Torralba, Freeman, Isola

Denoising

Decomposition of a noisy image



White Gaussian noise: $N(0, \sigma_n^2)$ Natural image

Find I(x,y) that maximizes the posterior (maximum a posteriory, MAP):

$$\max_{\mathbf{I}} p(\mathbf{I}|\mathbf{I}_n) = \max_{\mathbf{I}} \underset{\text{likelihood}}{\boxed{p(\mathbf{I}_n|\mathbf{I})}} \times \underset{\text{prior}}{\boxed{p(\mathbf{I})}}$$

$$= \max_{\mathbf{I}} \boxed{\exp(-\left|\mathbf{I}_n - \mathbf{I}\right|^2 / \sigma_n^2)} \times \boxed{\exp\left(-\frac{1}{2}\mathbf{I}^T\mathbf{C}^{-1}\mathbf{I}\right)}$$

Source: Torralba, Freeman, Isola

Denoising

$$\max_{\mathbf{I}} p(\mathbf{I}|\mathbf{I}_n) = \max_{\mathbf{I}} \boxed{p(\mathbf{I}_n|\mathbf{I})} \times \boxed{p(\mathbf{I})}$$

likelihood

prior

$$= \max_{\mathbf{I}} \boxed{\exp(-\left|\mathbf{I}_n - \mathbf{I}\right|^2 / \sigma_n^2)} \times \boxed{\exp\left(-\frac{1}{2}\mathbf{I}^T \mathbf{C}^{-1}\mathbf{I}\right)}$$

The solution is:

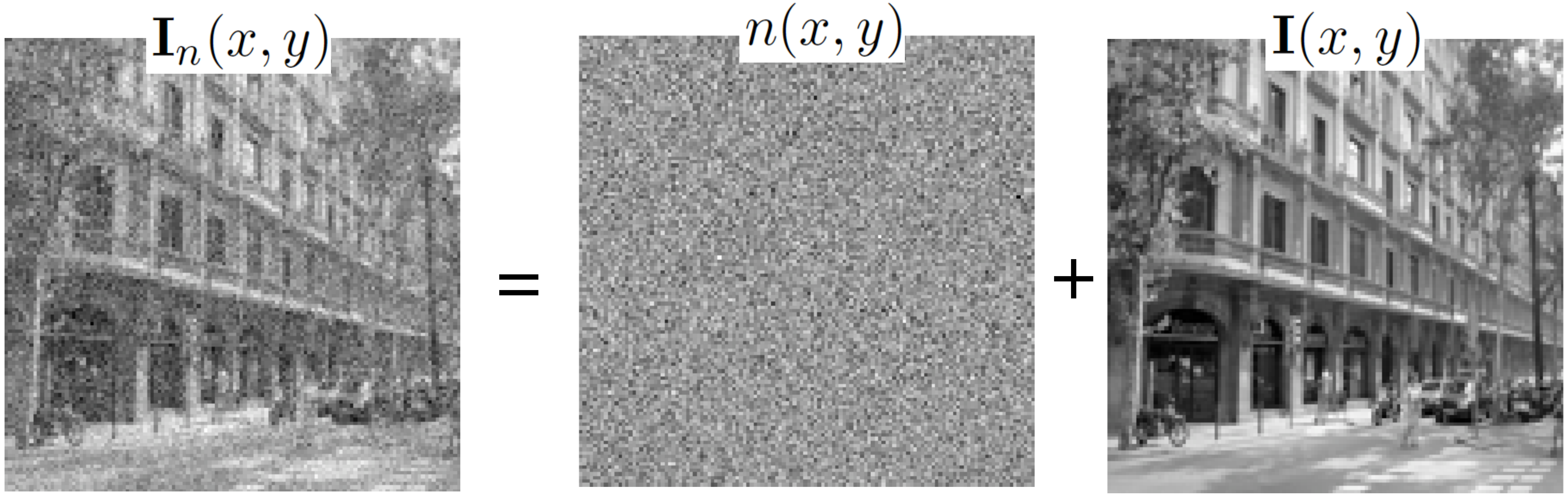
$$\mathbf{I} = \mathbf{C}\left(\mathbf{C} + \sigma_n^2 \mathbb{I}\right)^{-1} \mathbf{I}_n$$ (note this is a linear operation)

This can also be written in the Fourier domain, for constant A:

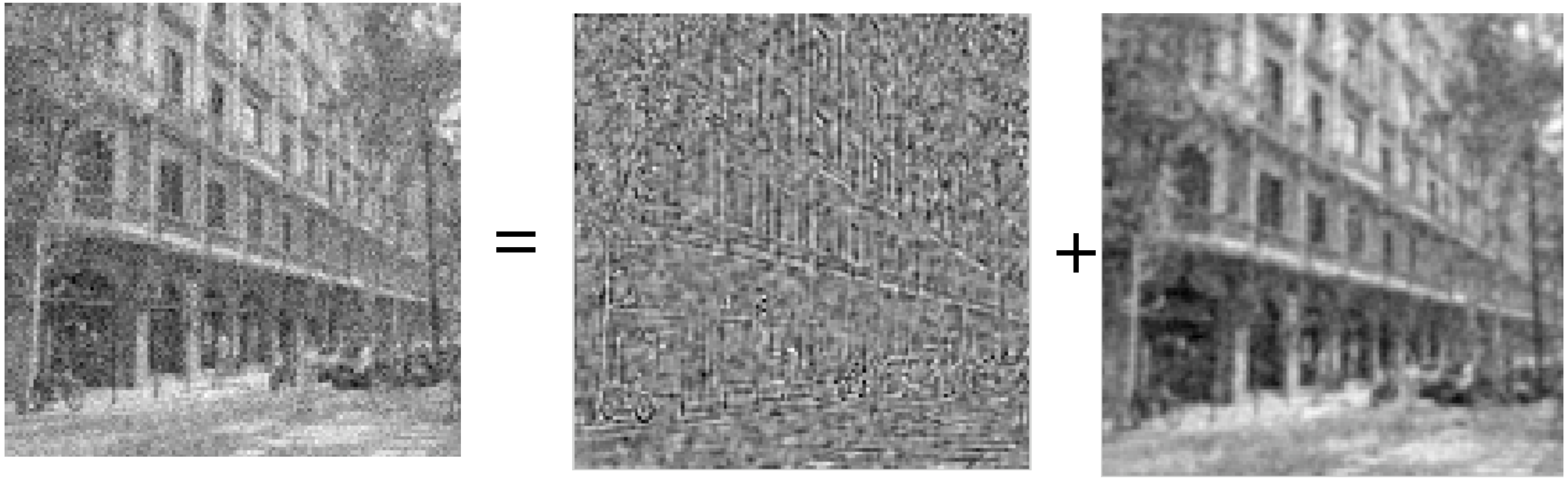
$$\widetilde{\mathbf{I}}(v) = \frac{A/|v|^{2\alpha}}{A/|v|^{2\alpha} + \sigma_n^2} \widetilde{\mathbf{I}}_n(v)$$

Source: Torralba, Freeman, Isola

The truth:

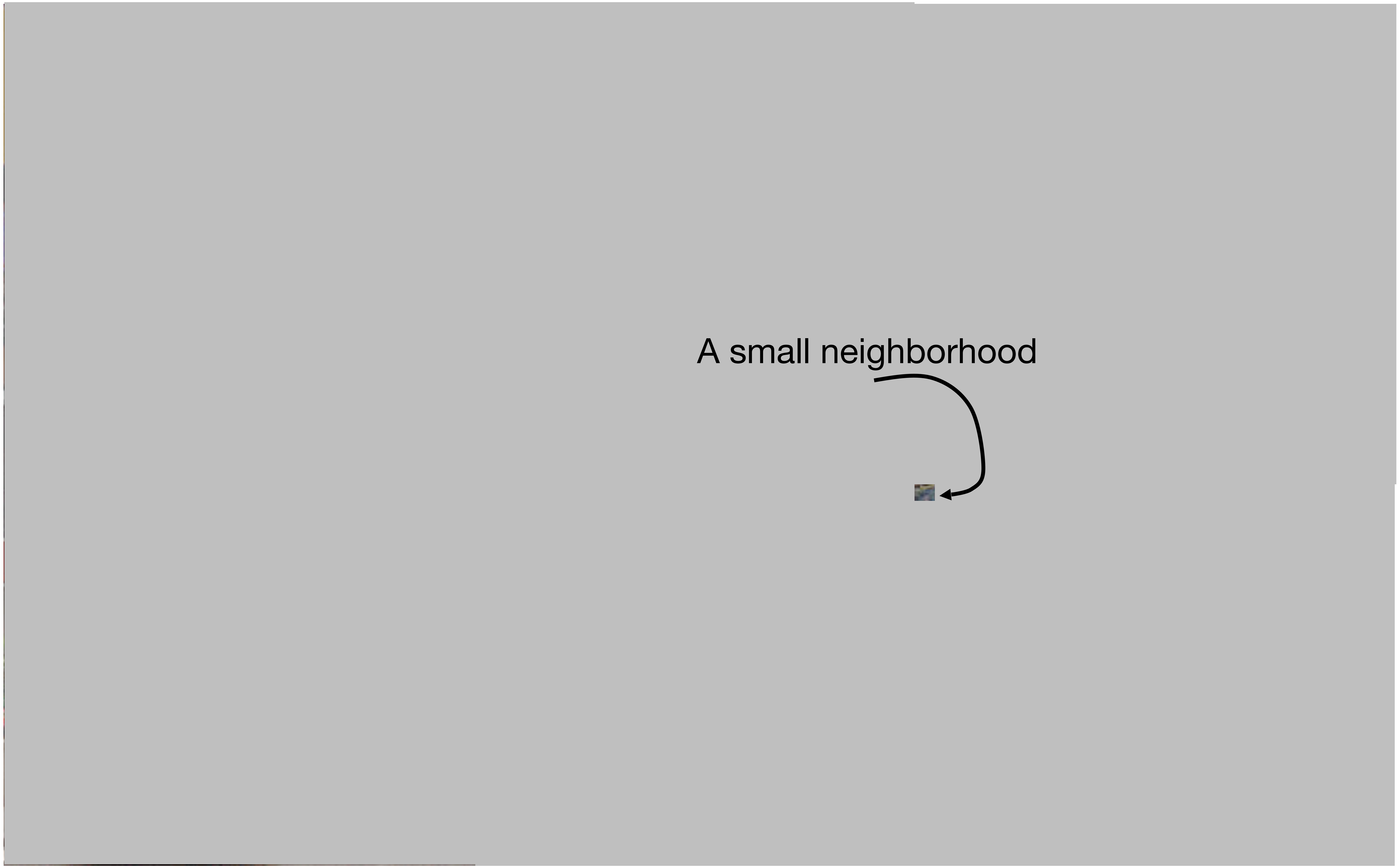


The estimated decomposition:



Source: Torralba, Freeman, Isola

Statistical modeling of images



Source: Torralba, Freeman, Isola

[-1 1]

g[m,n]

$\otimes$

[-1, 1]

h[m,n]

=

f[m,n]

Source: Torralba, Freeman, Isola

$[-1\ 1]^T$

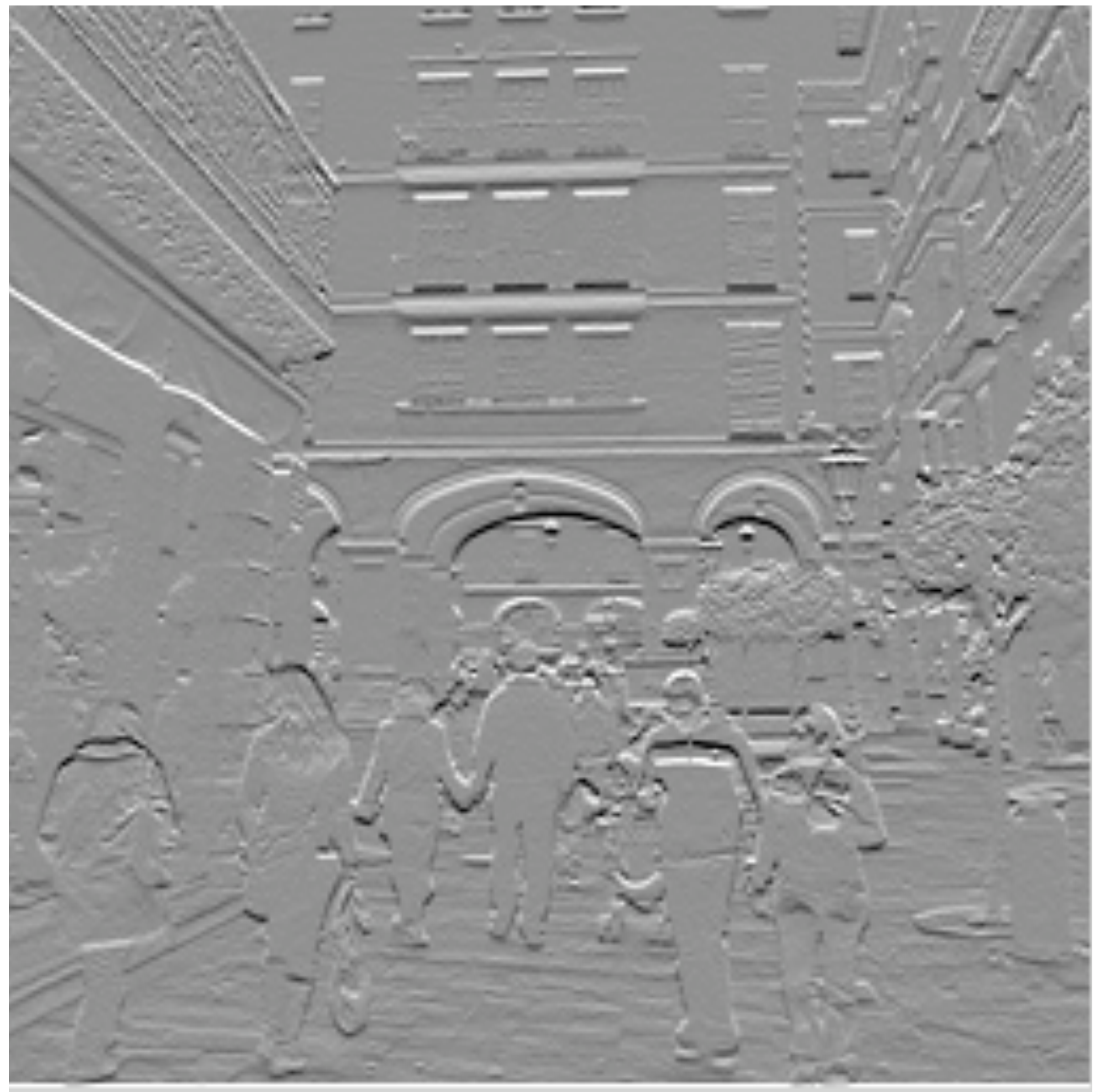
g[m,n]

$\otimes$

$[-1, 1]^T$

h[m,n]

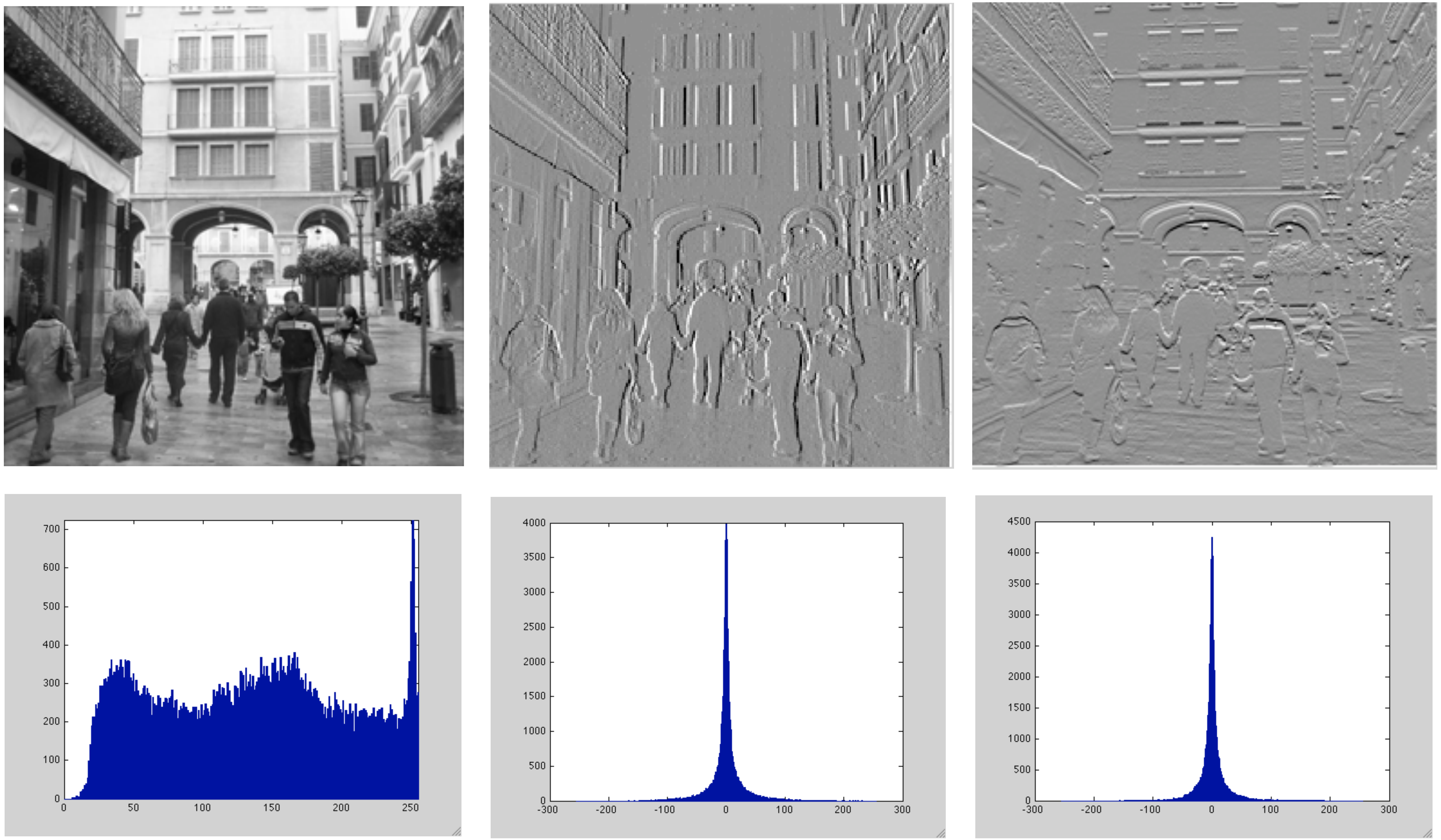
=



f[m,n]

Source: Torralba, Freeman, Isola

Observation: Sparse filter response



Source: Torralba, Freeman, Isola

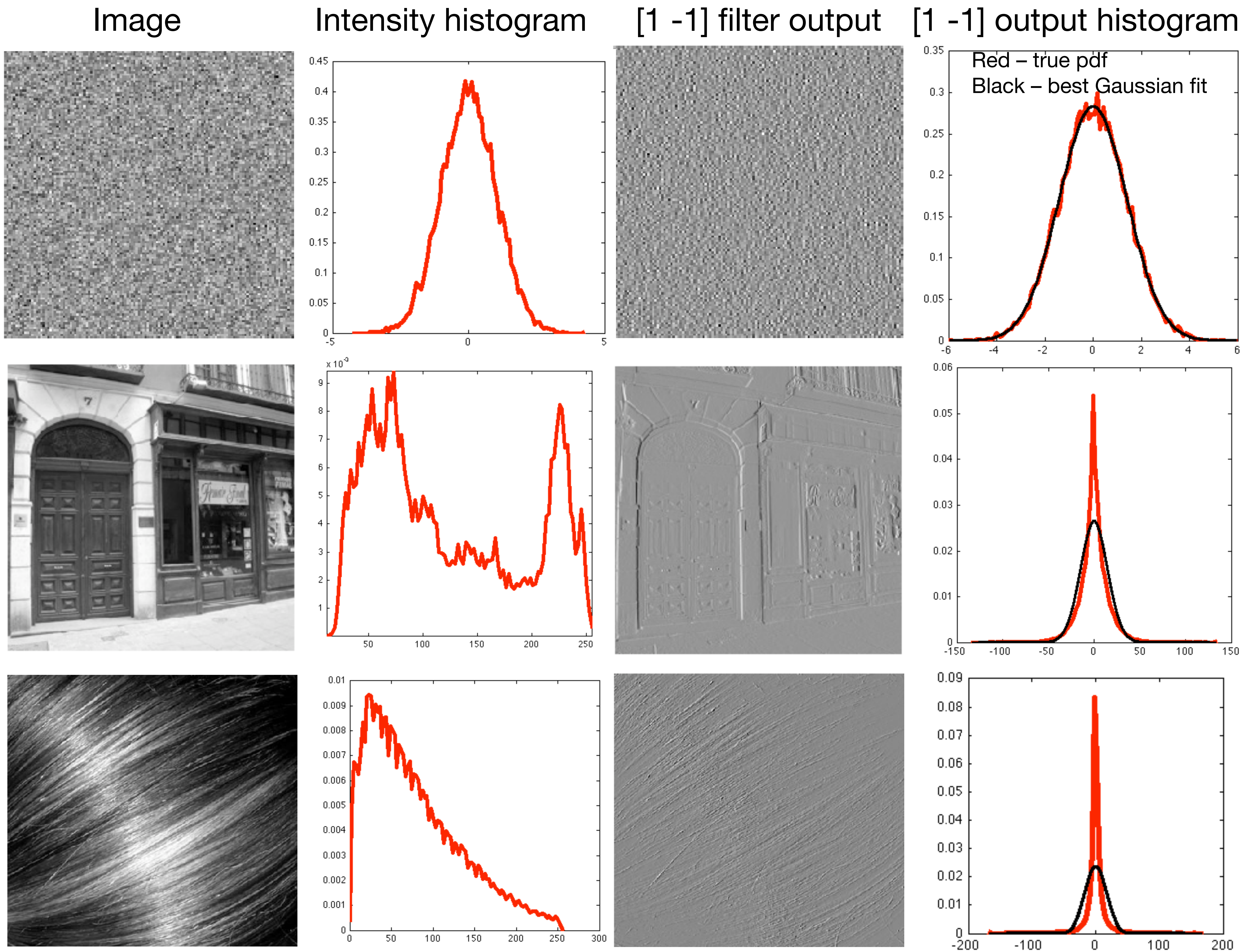
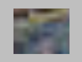


Image
Intensity histogram
[1 -1] filter output
[1 -1] output histogram
Red – true pdf
Black – best Gaussian fit

A model based on filter outputs

A small neighborhood



$$p(\mathbf{I}) = \prod_{k} \prod_{x,y} p(h_k(x,y))$$

All pixels and all outputs are independent

Filter outputs

Source: Torralba, Freeman, Isola

Sampling images

Gaussian model

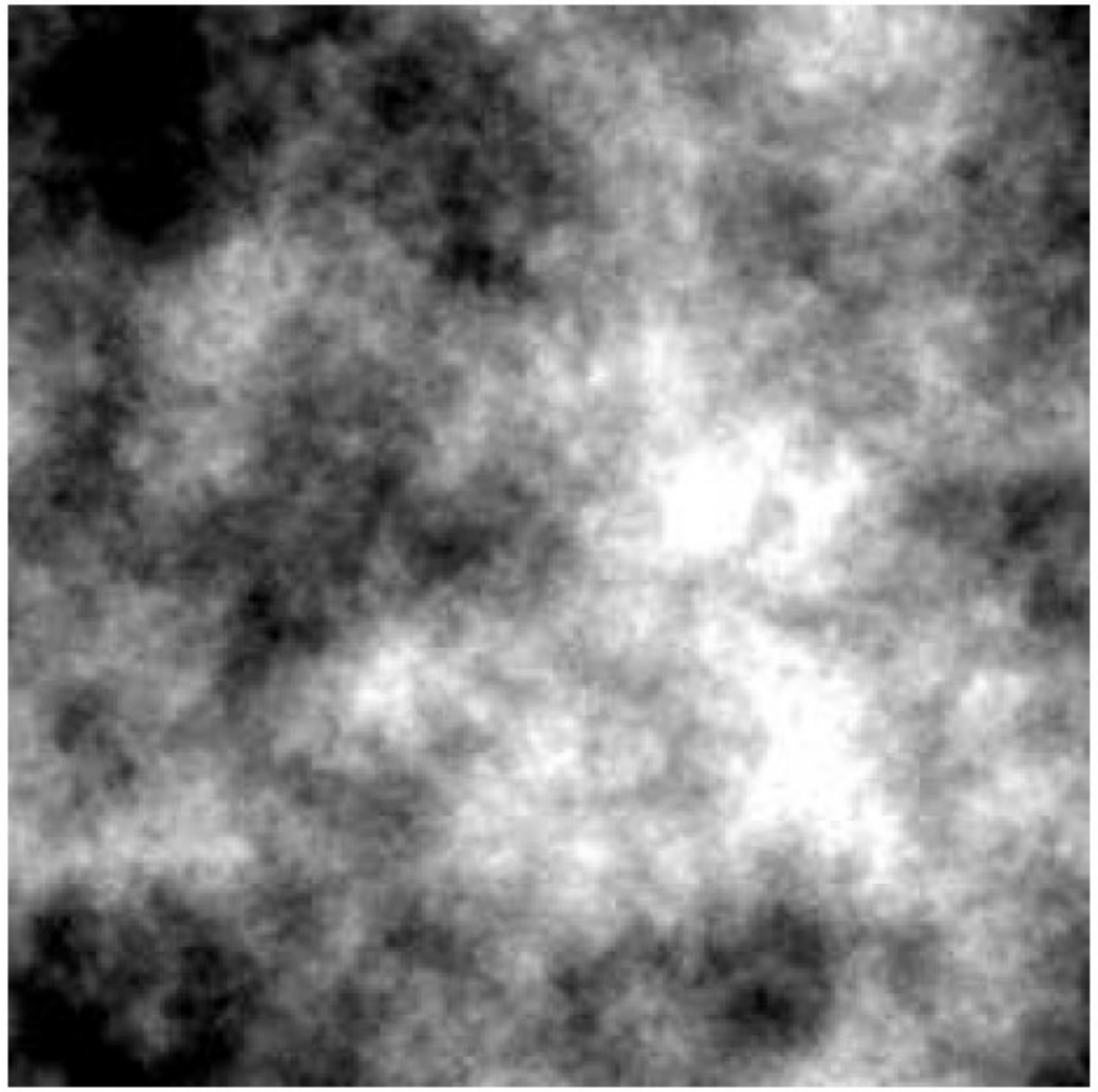


Fig. 3. Example image randomly drawn from the Gaussian spectral model, with $\gamma = 2.0$.

Wavelet marginal model

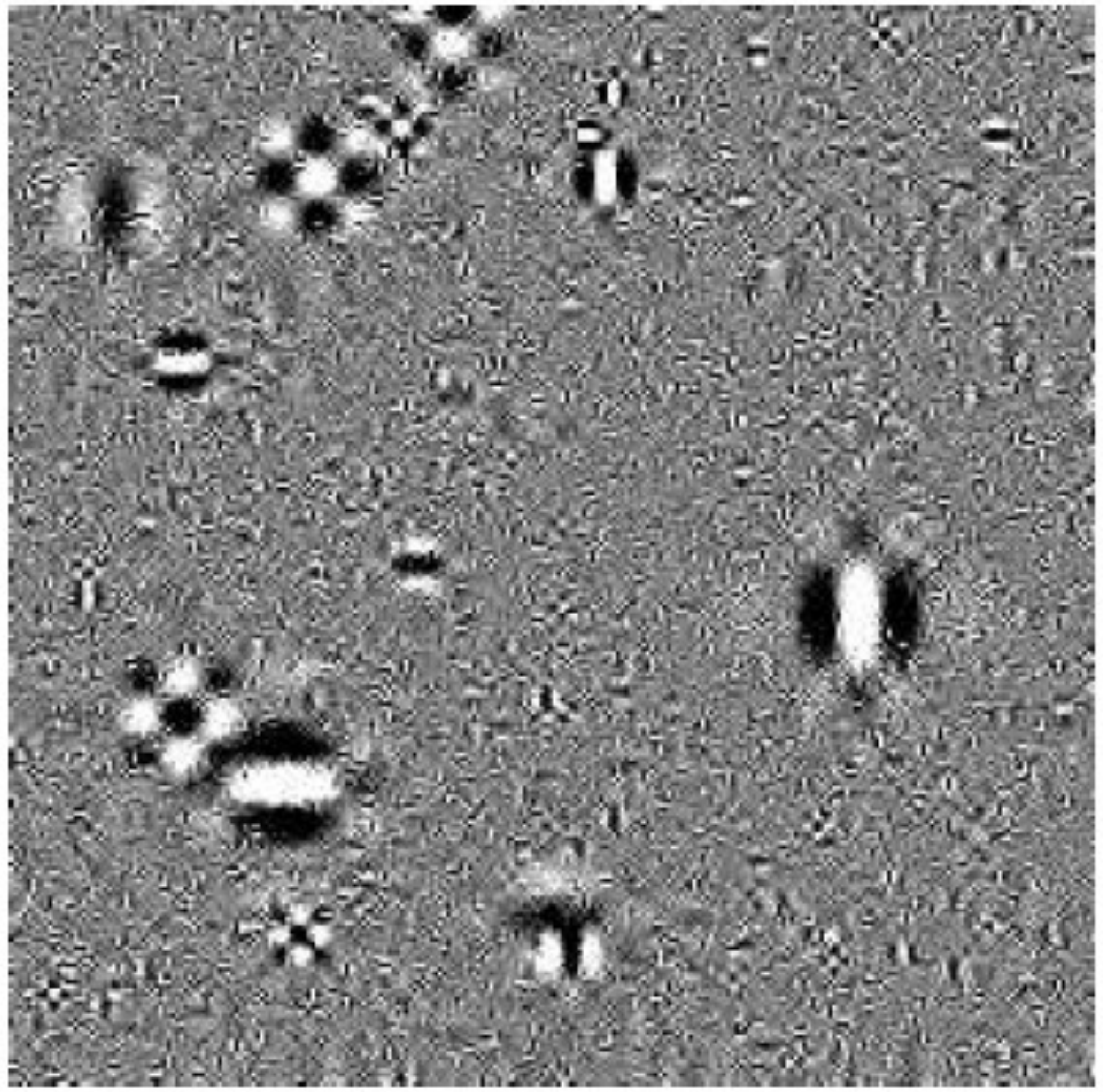


Fig. 6. A sample image drawn from the wavelet marginal model, with subband density parameters chosen to fit the image of Fig. 7.

Source: Torralba, Freeman, Isola

Taking a picture…

What the camera give us…

How do we correct this?

Slides R. Fergus

Deblurring

Slides R. Fergus

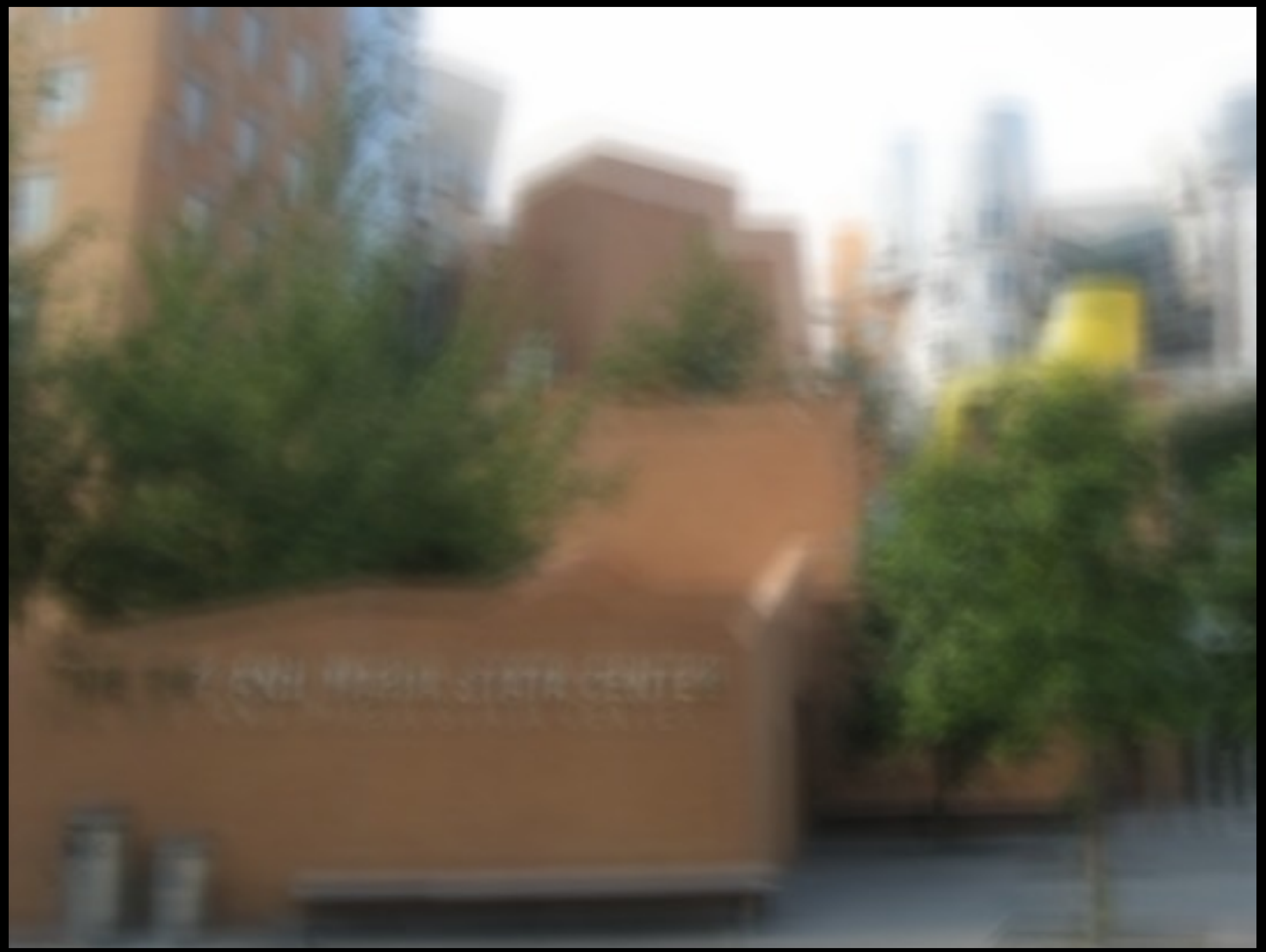
Deblurring

Slides R. Fergus

Deblurring

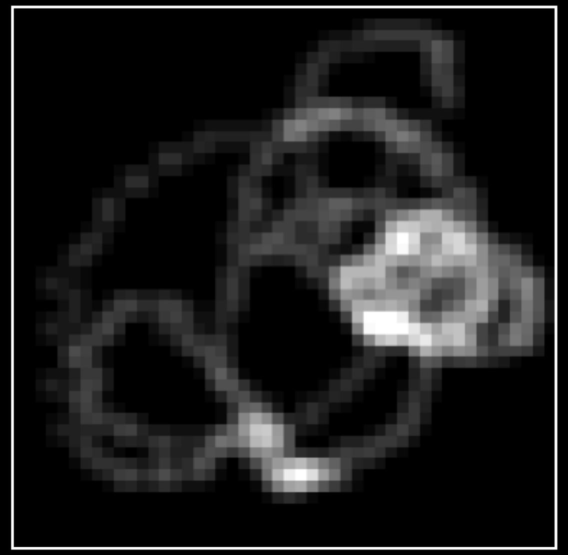
Slides R. Fergus

Image formation process



=

$\otimes$



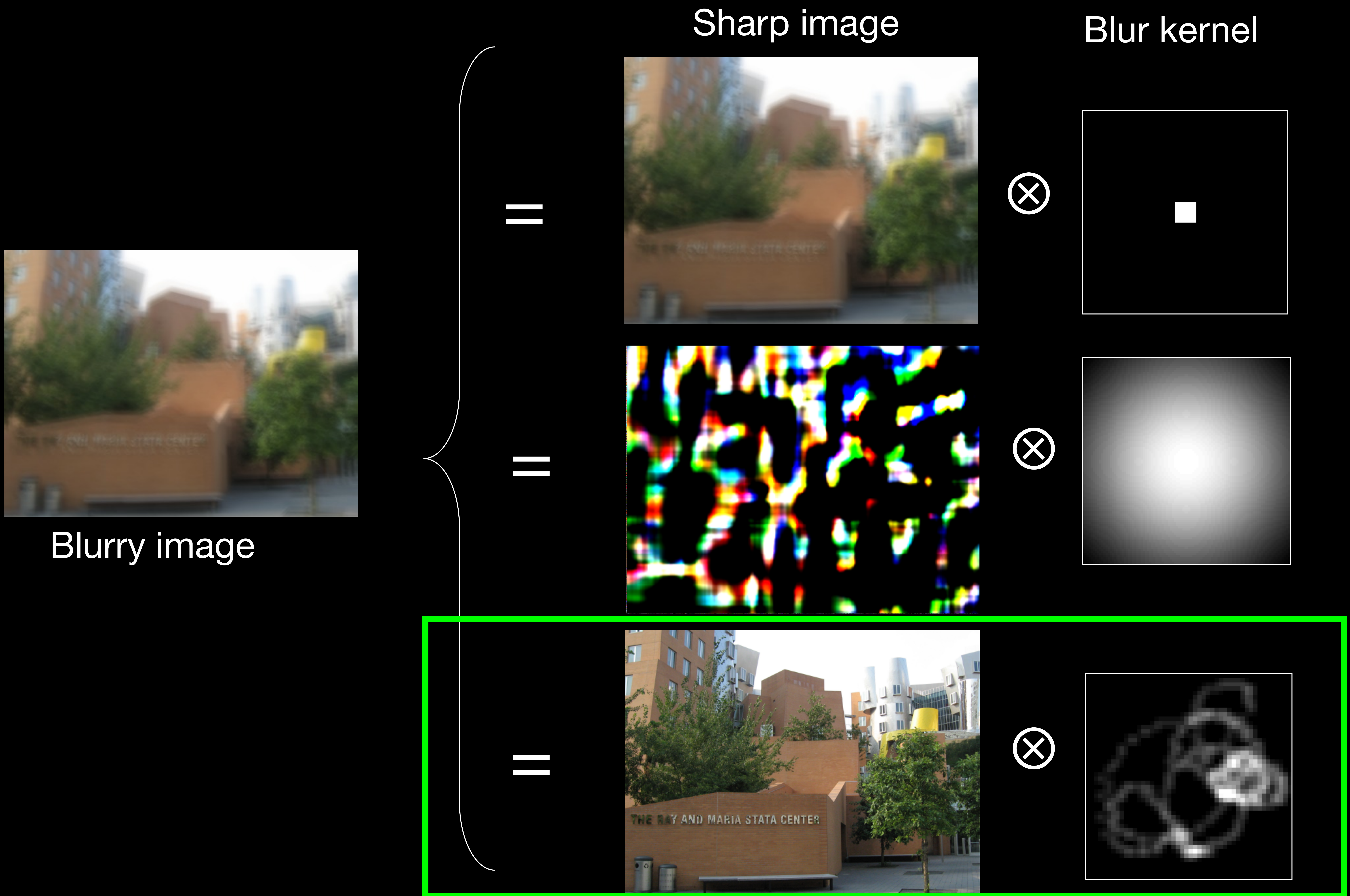
Blur kernel

$\underbrace{\text{Blurry image}}_{\text{Input to algorithm}}$ = $\underbrace{\text{Sharp image}}_{\text{Desired output}}$ $\otimes$ Blur kernel

Convolution operator

Model is approximation

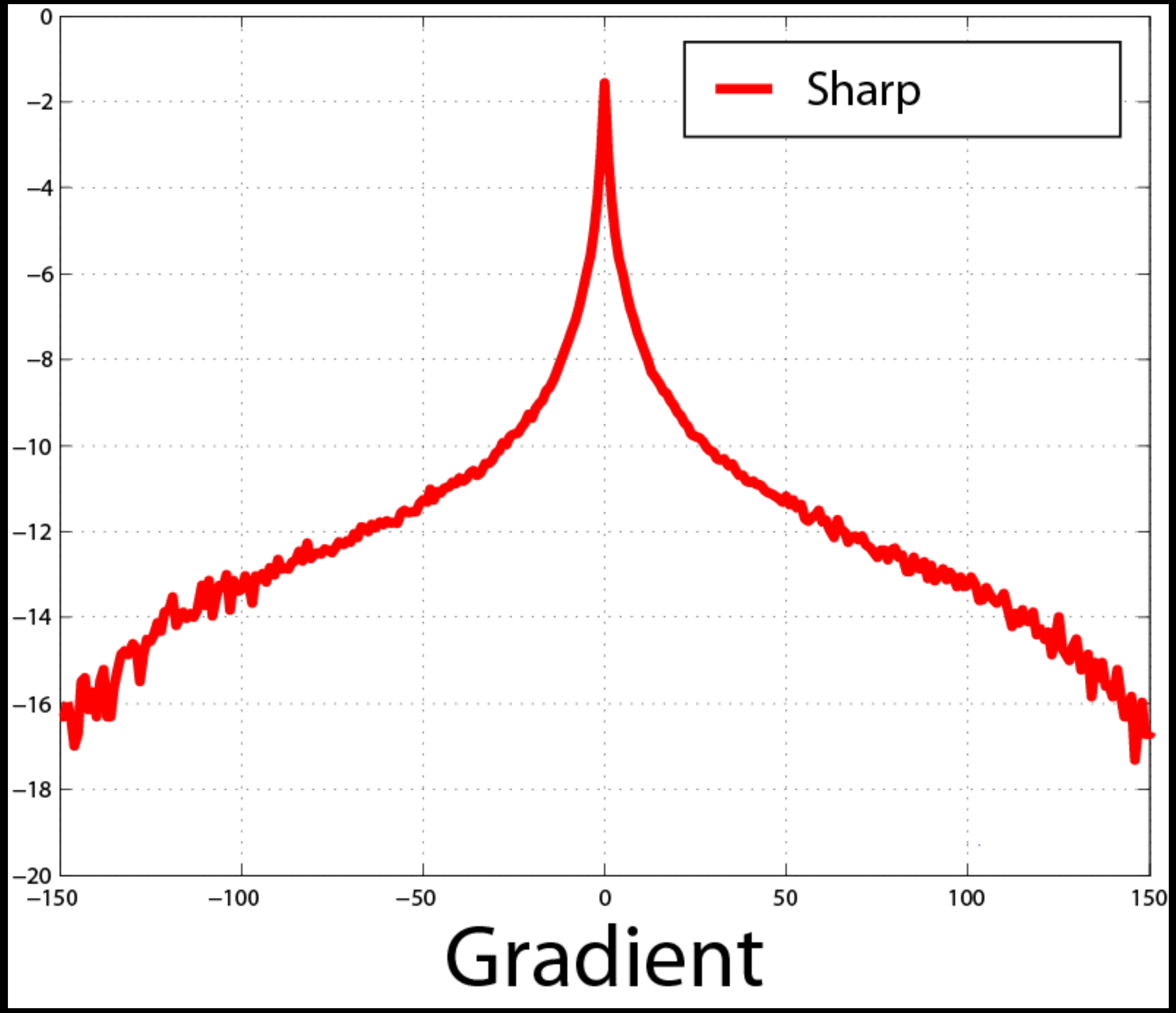
Multiple possible solutions



Natural image statistics

Characteristic distribution with heavy tails

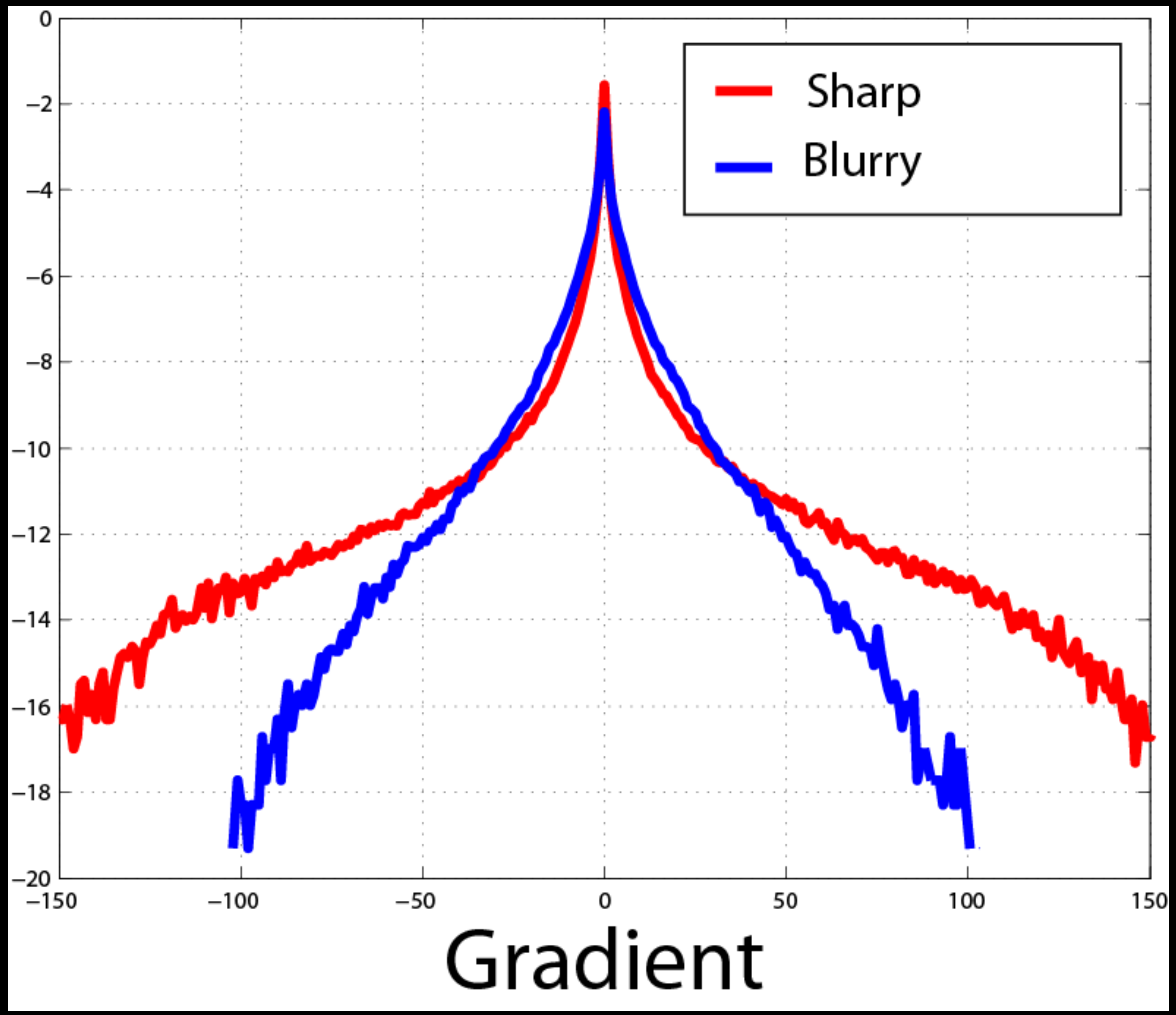
Histogram of image gradients



Slides R. Fergus

Blury images have different statistics

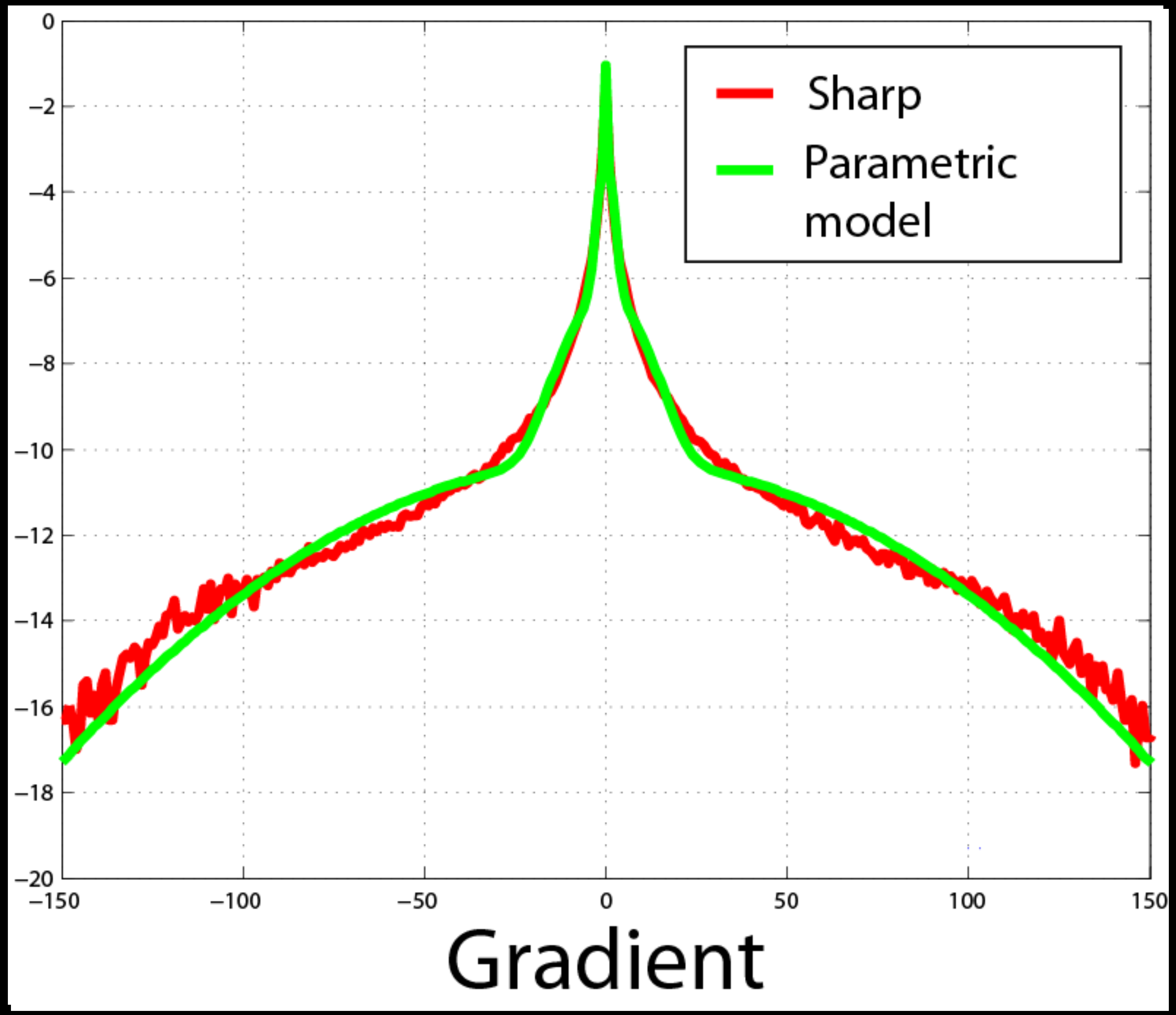
Histogram of image gradients



Slides R. Fergus

Parametric distribution

Histogram of image gradients



Use parametric model of sharp image statistics

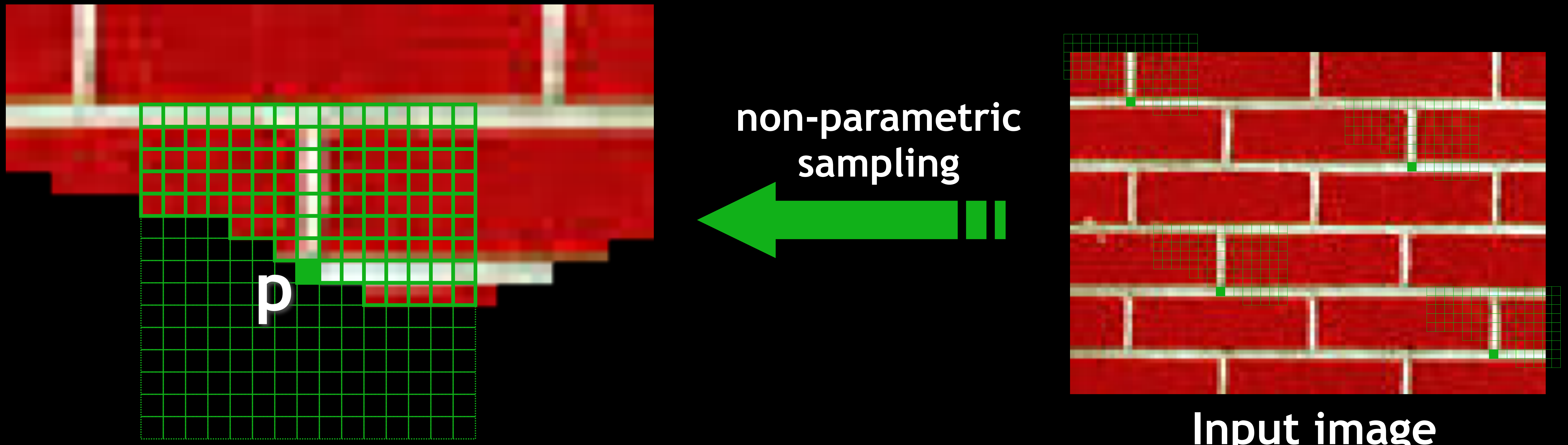
Slides R. Fergus

Nonparametric image models

Texture Synthesis

Source: A. Efros

Efros & Leung Algorithm



Synthesizing a pixel

Input image

- Assuming Markov property, compute P(**p**|N(**p**))
  - Building explicit probability tables infeasible
  - Instead, we *search the input image* for all similar neighborhoods — that's our pdf for **p**
  - To sample from this pdf, just pick one match at random

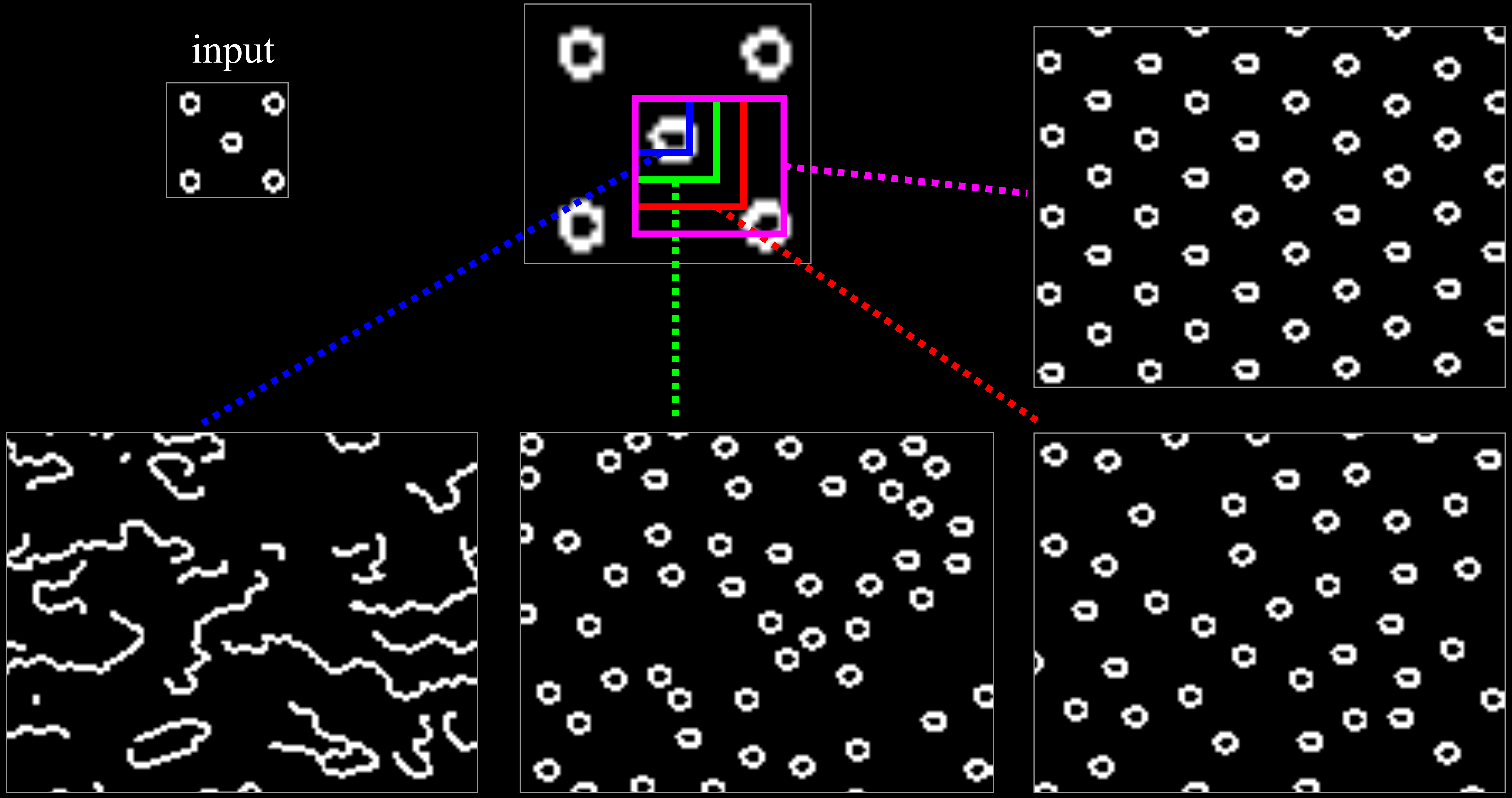
Source: A. Efros

Some Details

- Growing is in "onion skin" order
  - Within each "layer", pixels with most neighbors are synthesized first
  - If no close match can be found, the pixel is not synthesized until the end
- Using *Gaussian-weighted* SSD is very important
  - to make sure the new pixel agrees with its closest neighbors
  - Approximates reduction to a smaller neighborhood window if data is too sparse

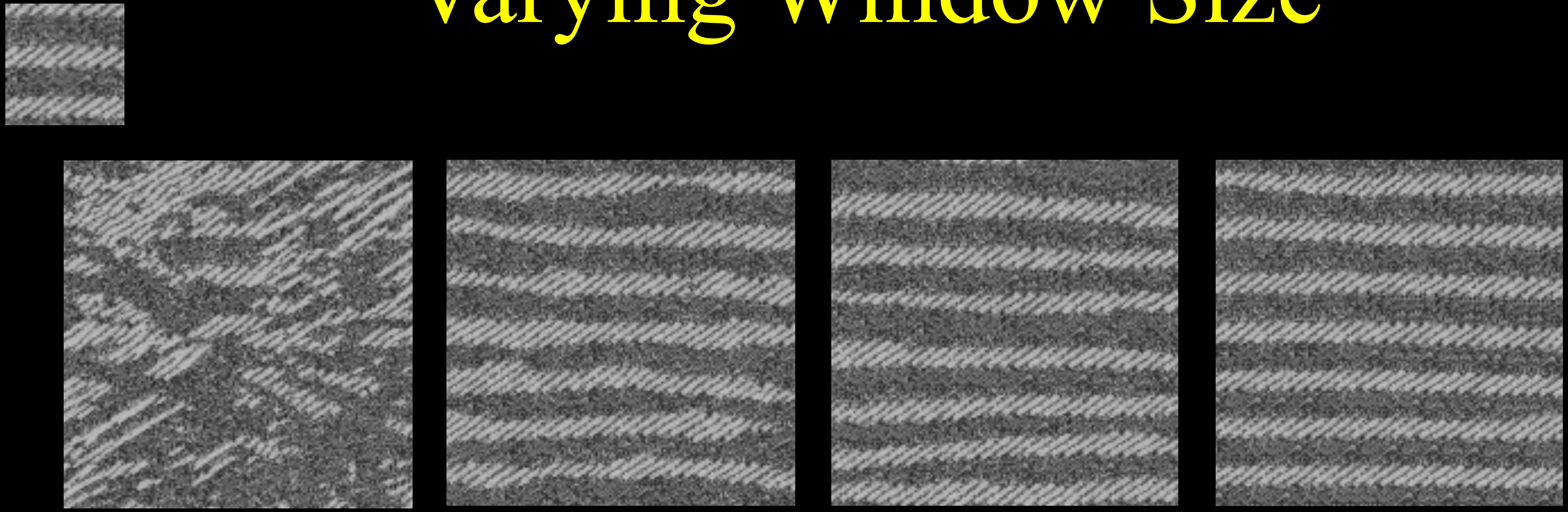
Source: A. Efros

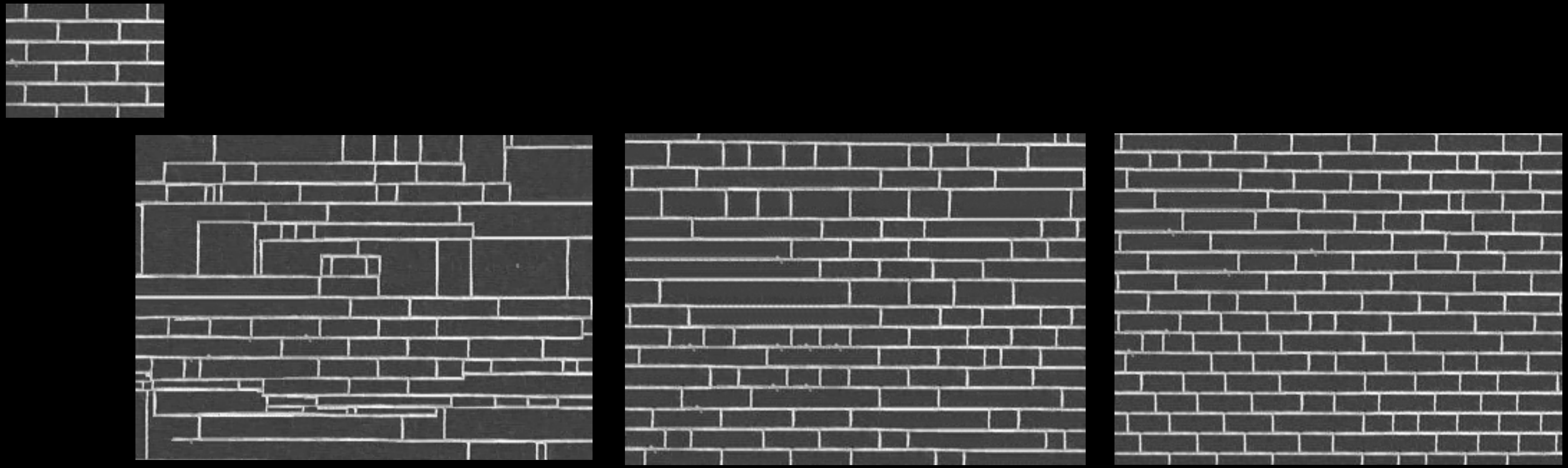
Neighborhood Window



Source: A. Efros

Varying Window Size



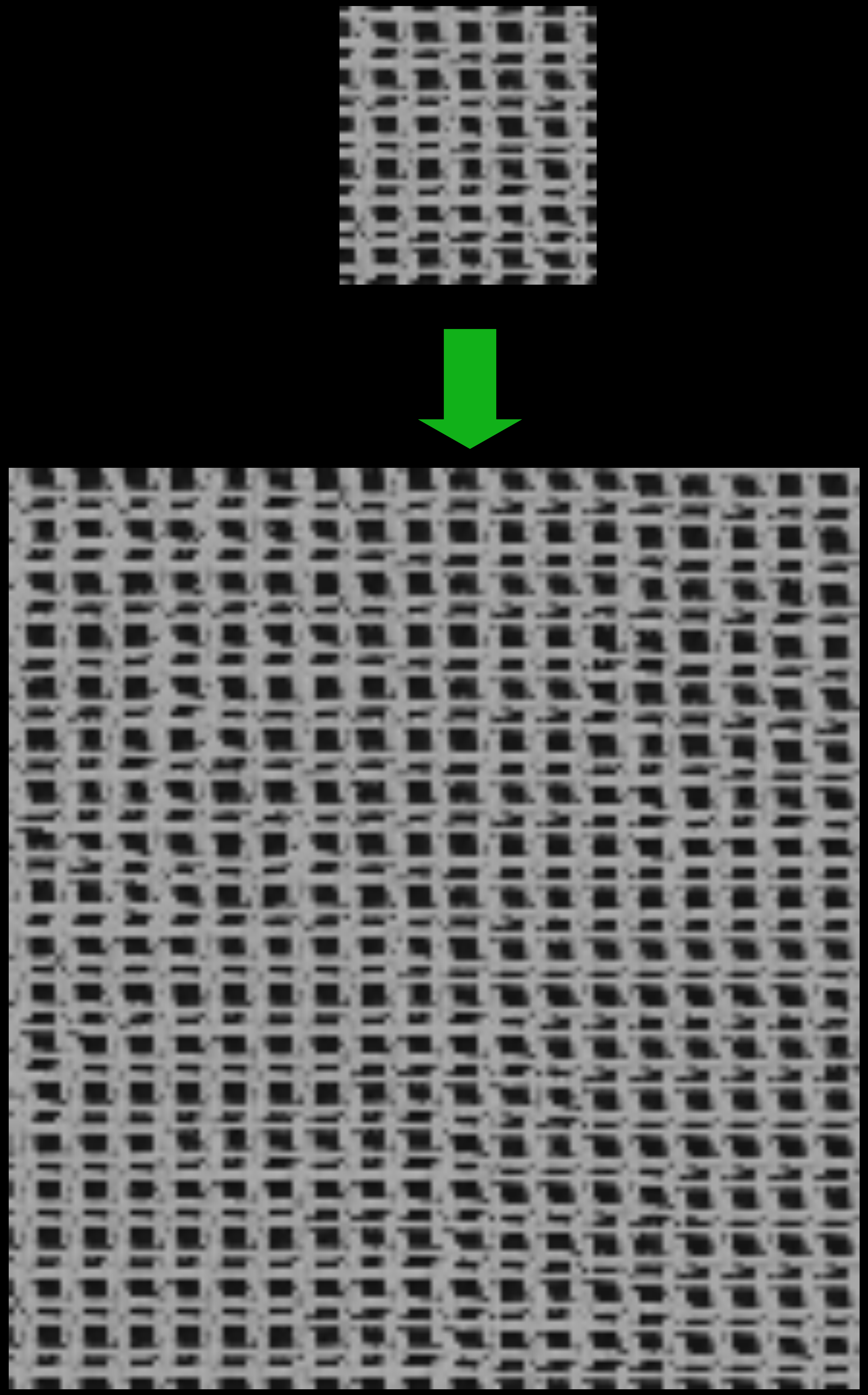


Increasing window size

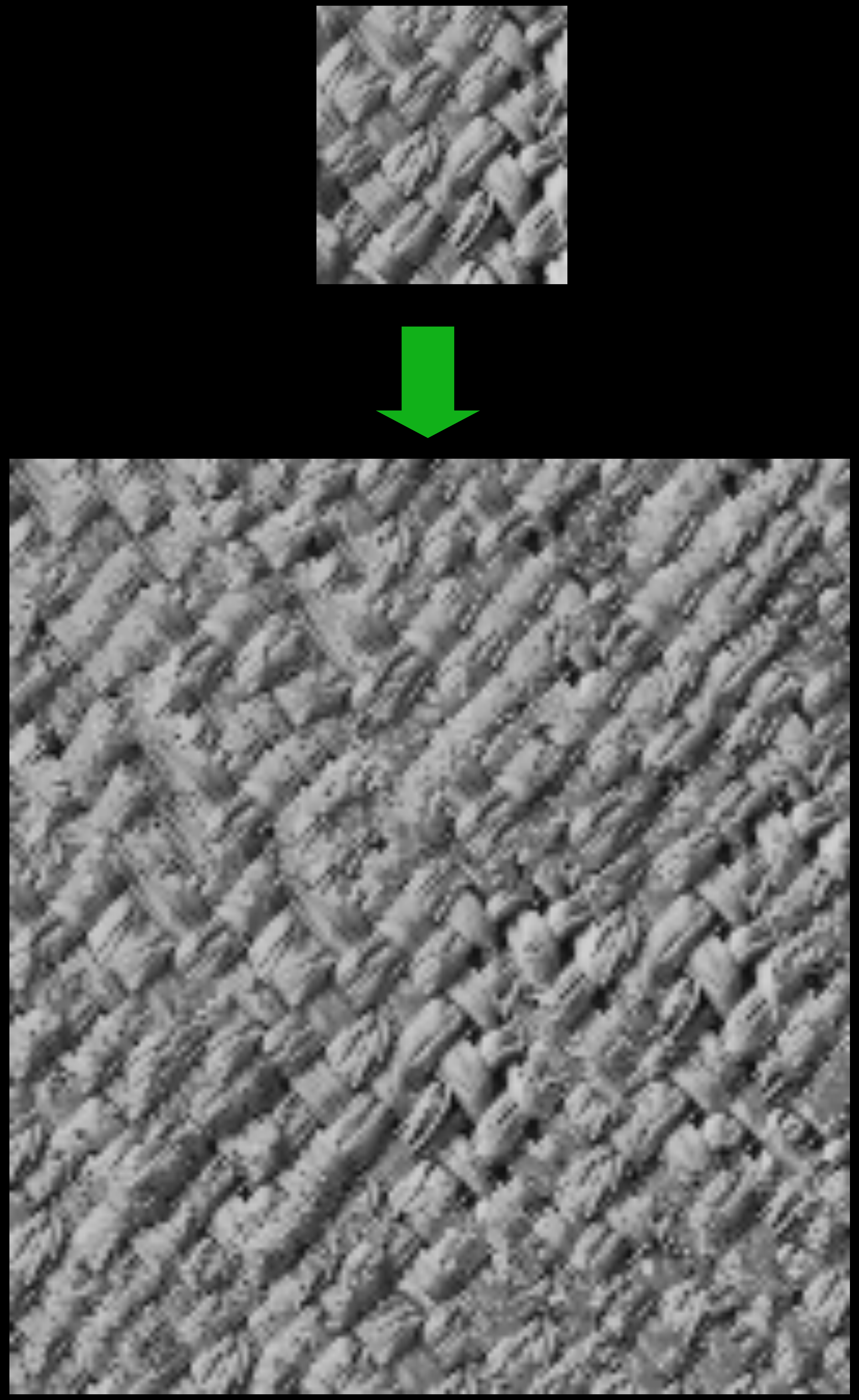
Source: A. Efros

Synthesis Results

french canvas



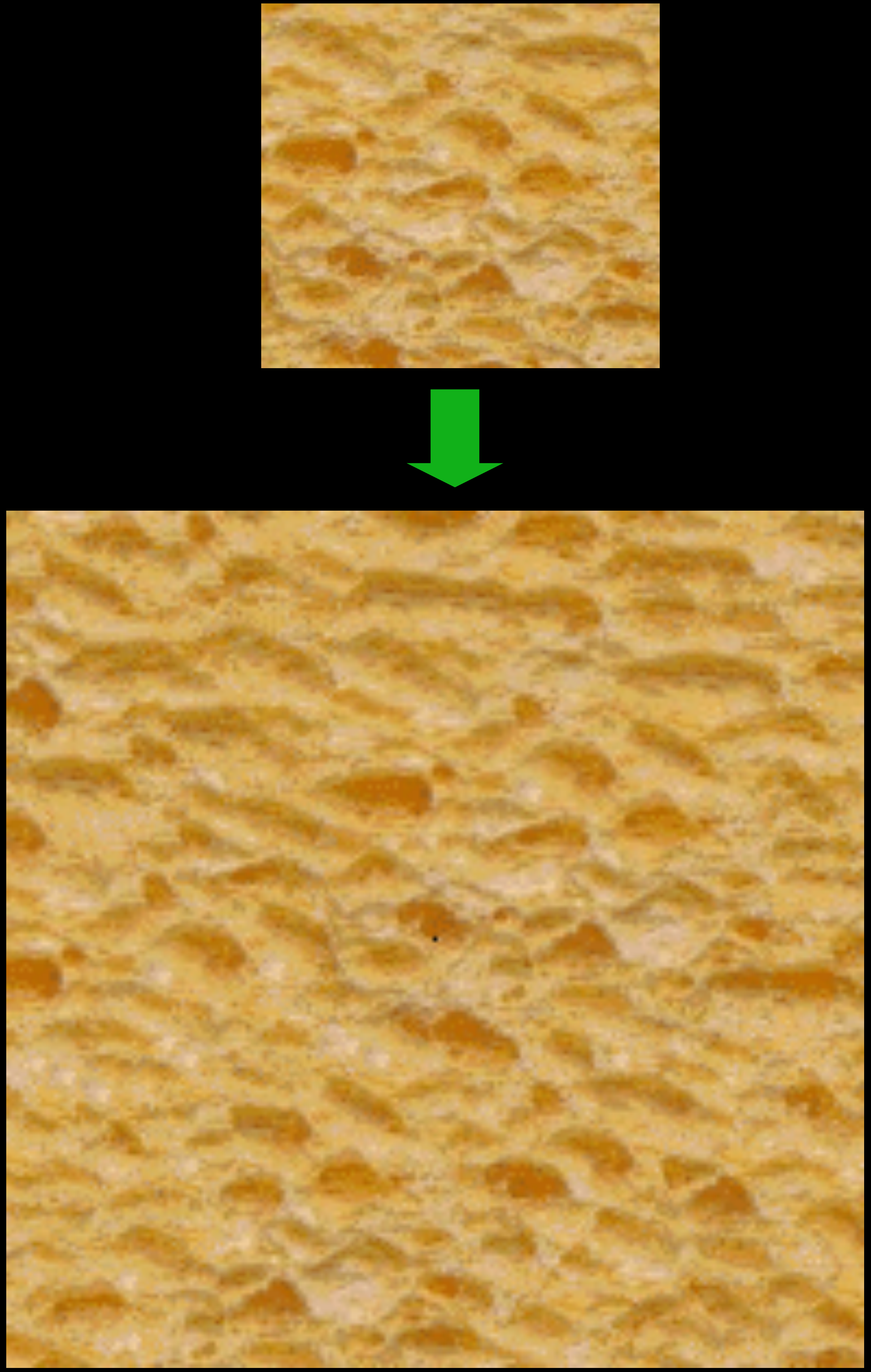
rafia weave



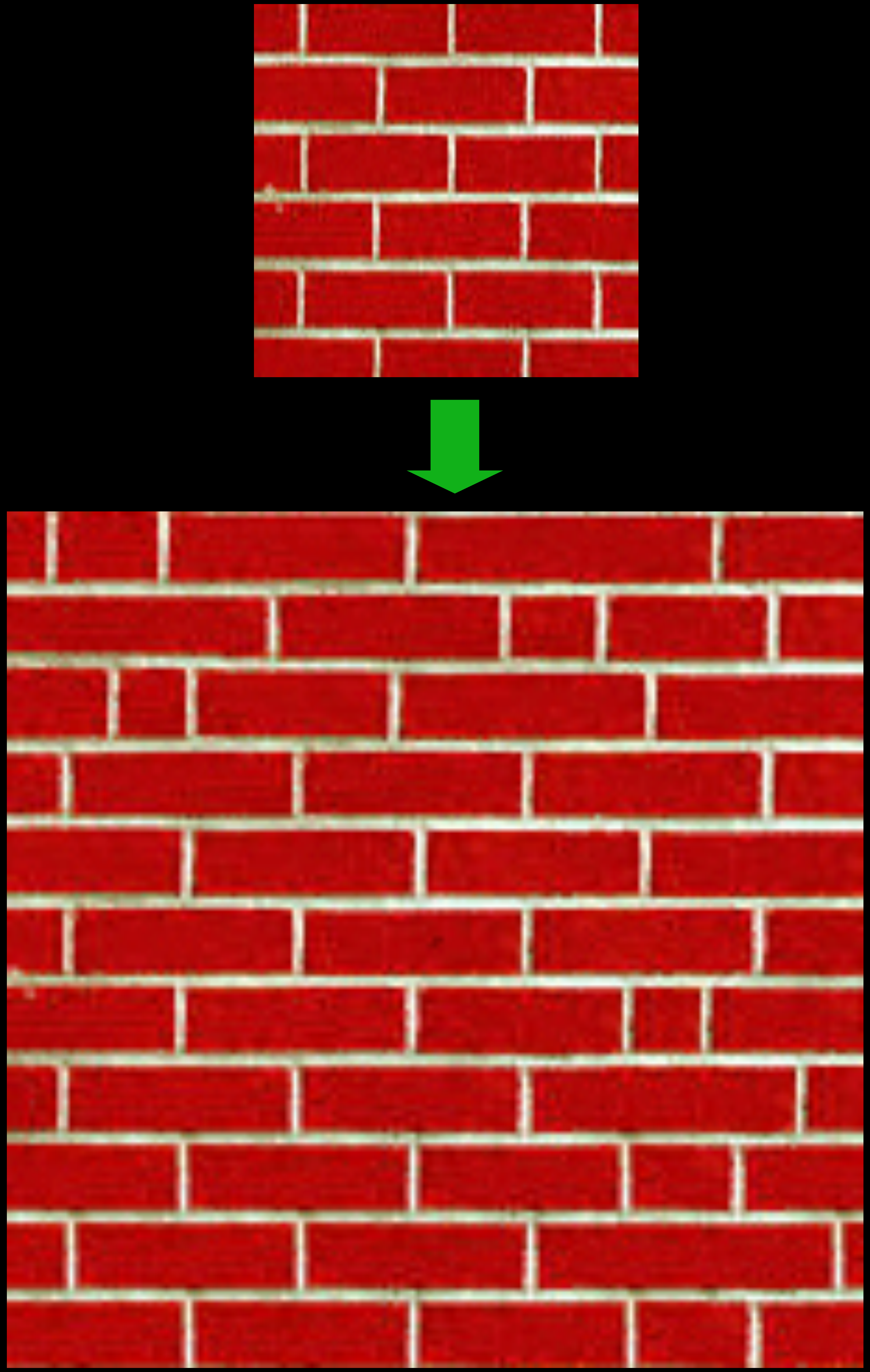
Source: A. Efros

More Results

white bread

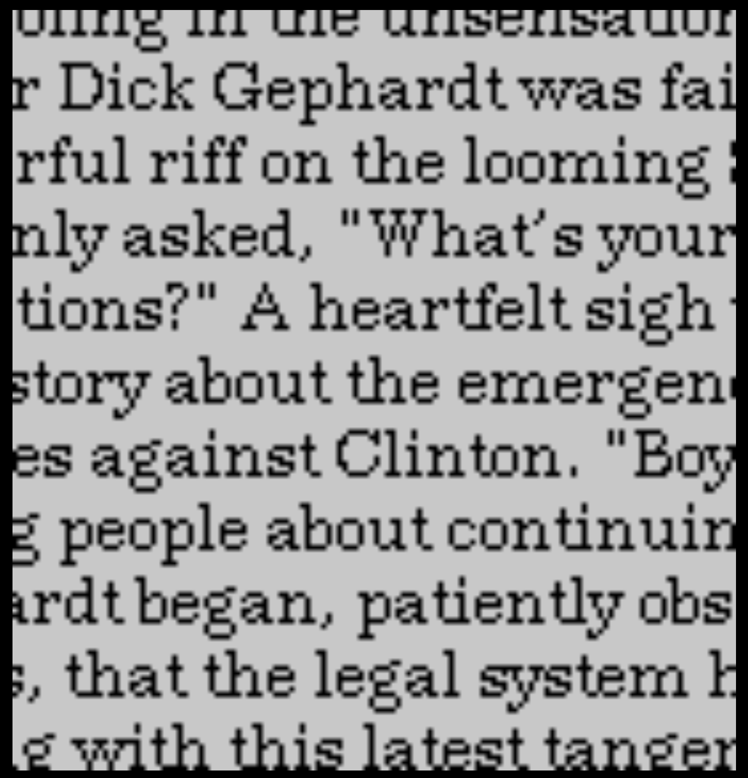


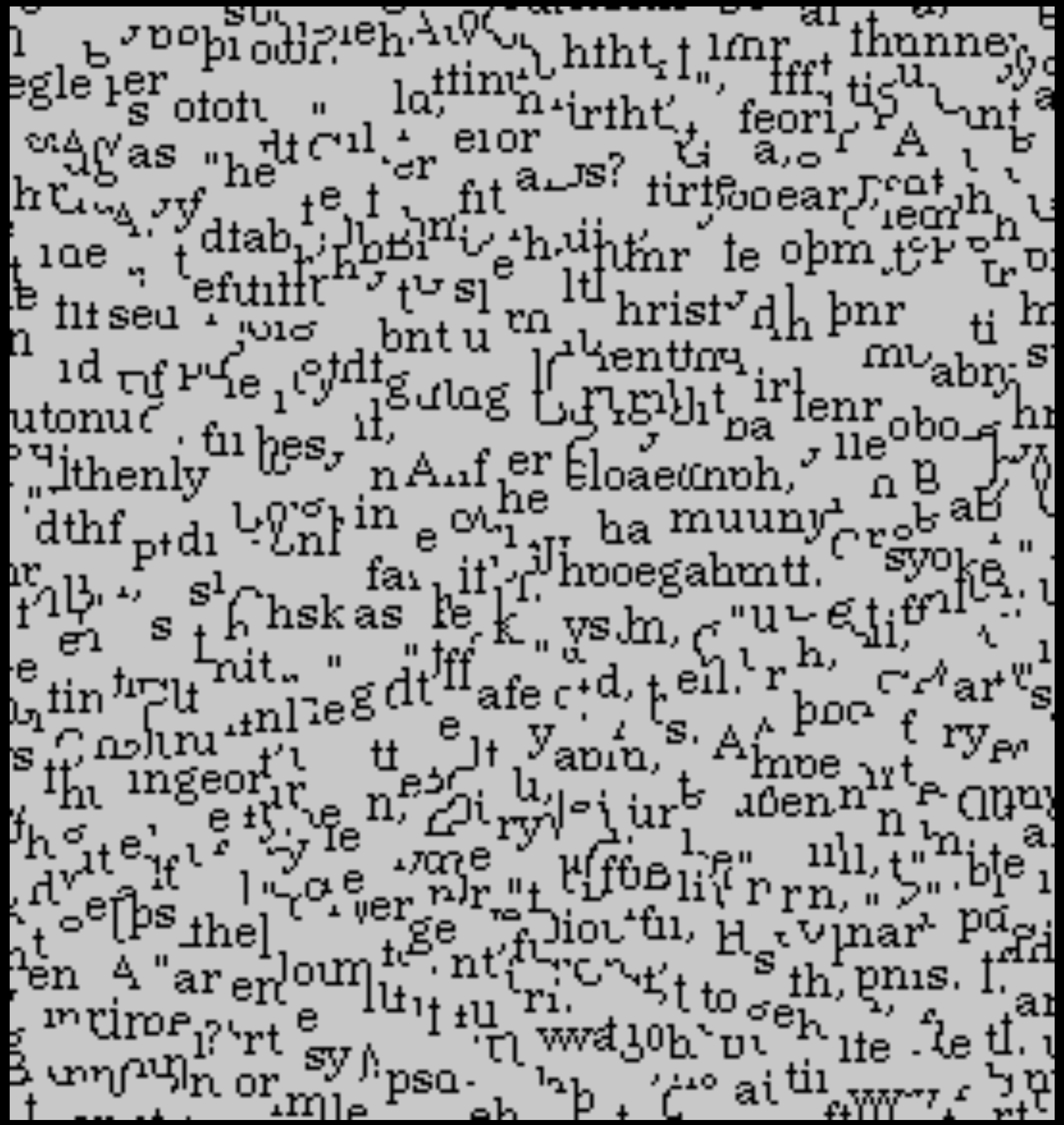
brick wall

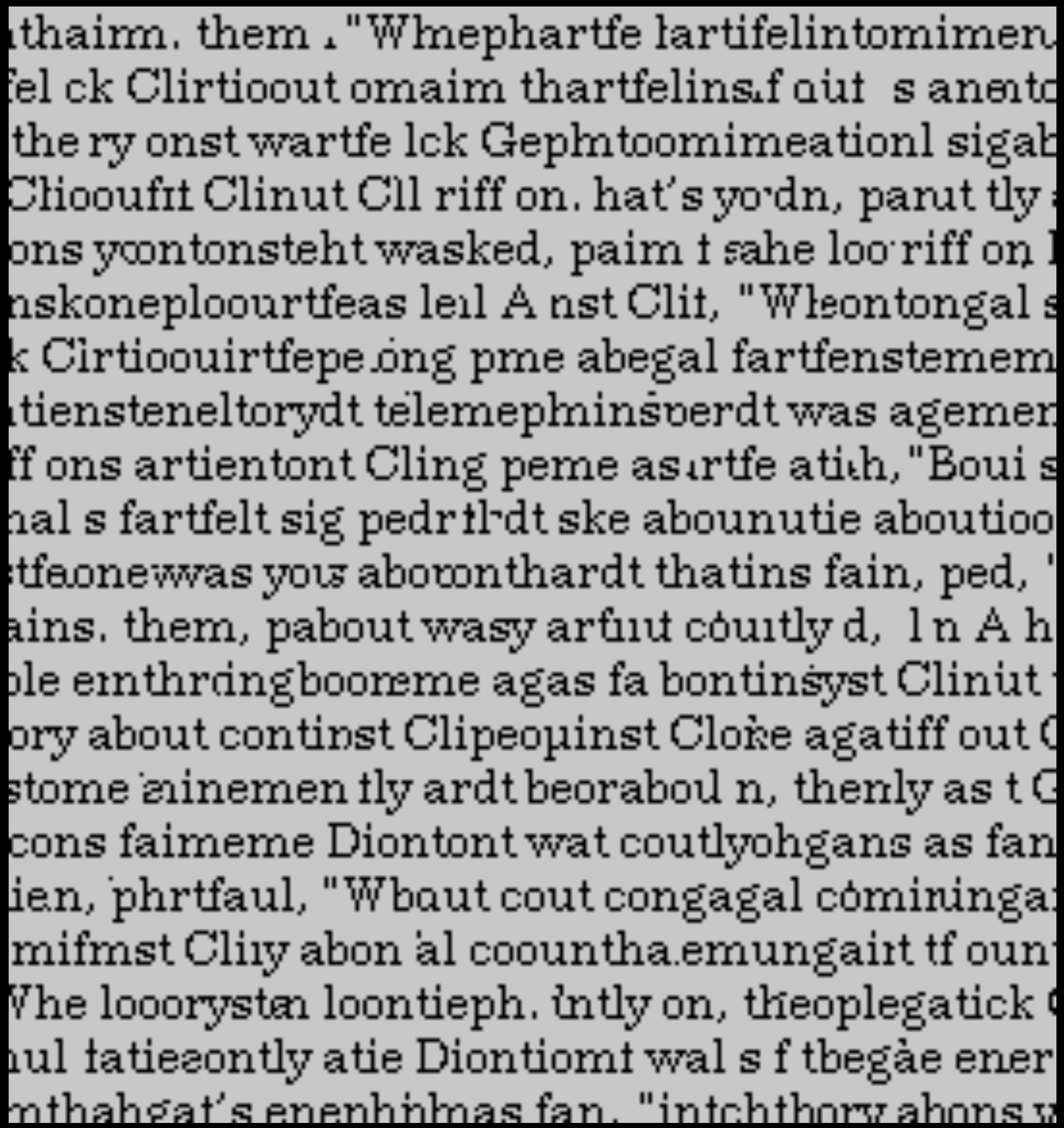


Source: A. Efros

Homage to Shannon

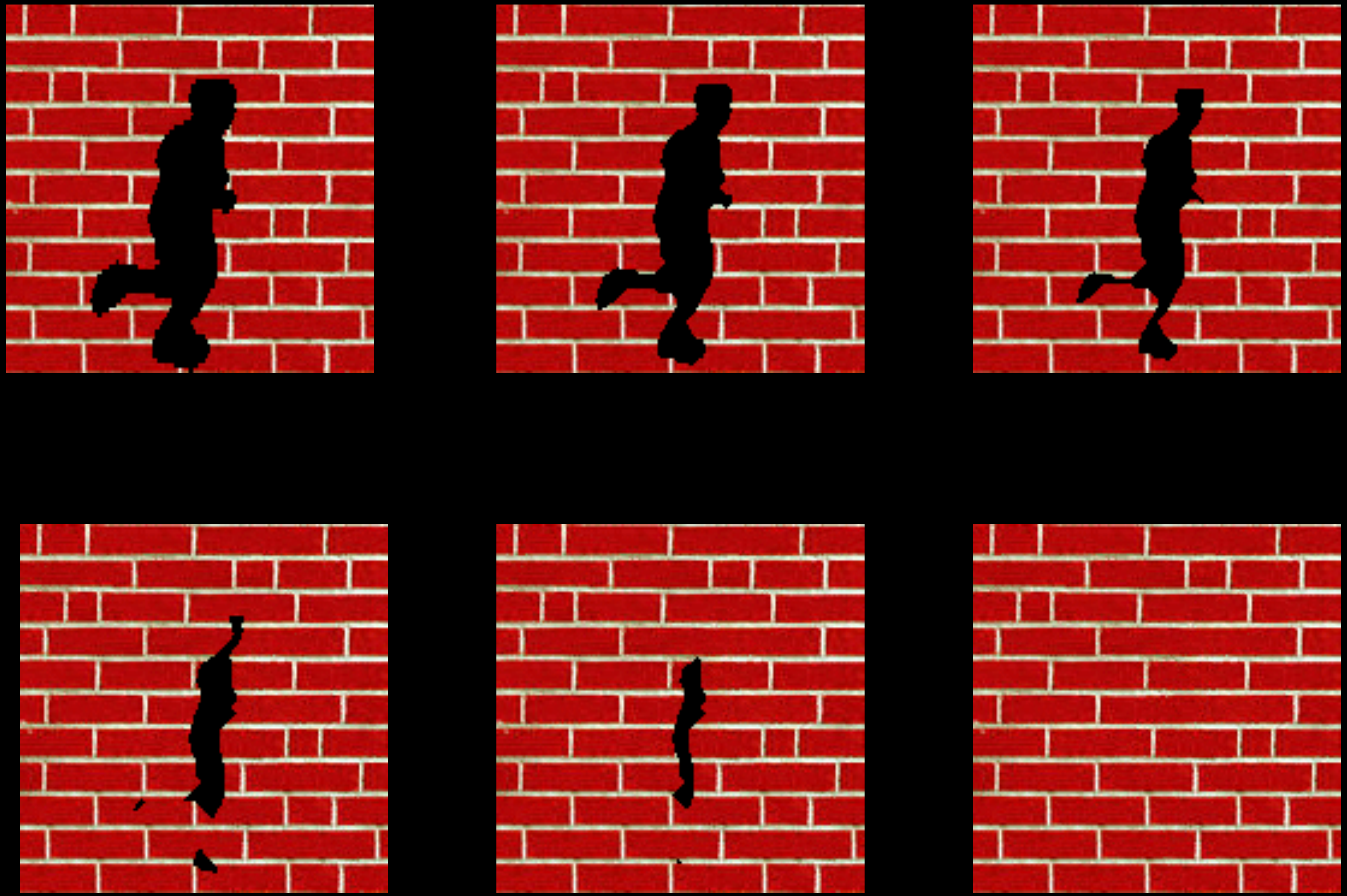






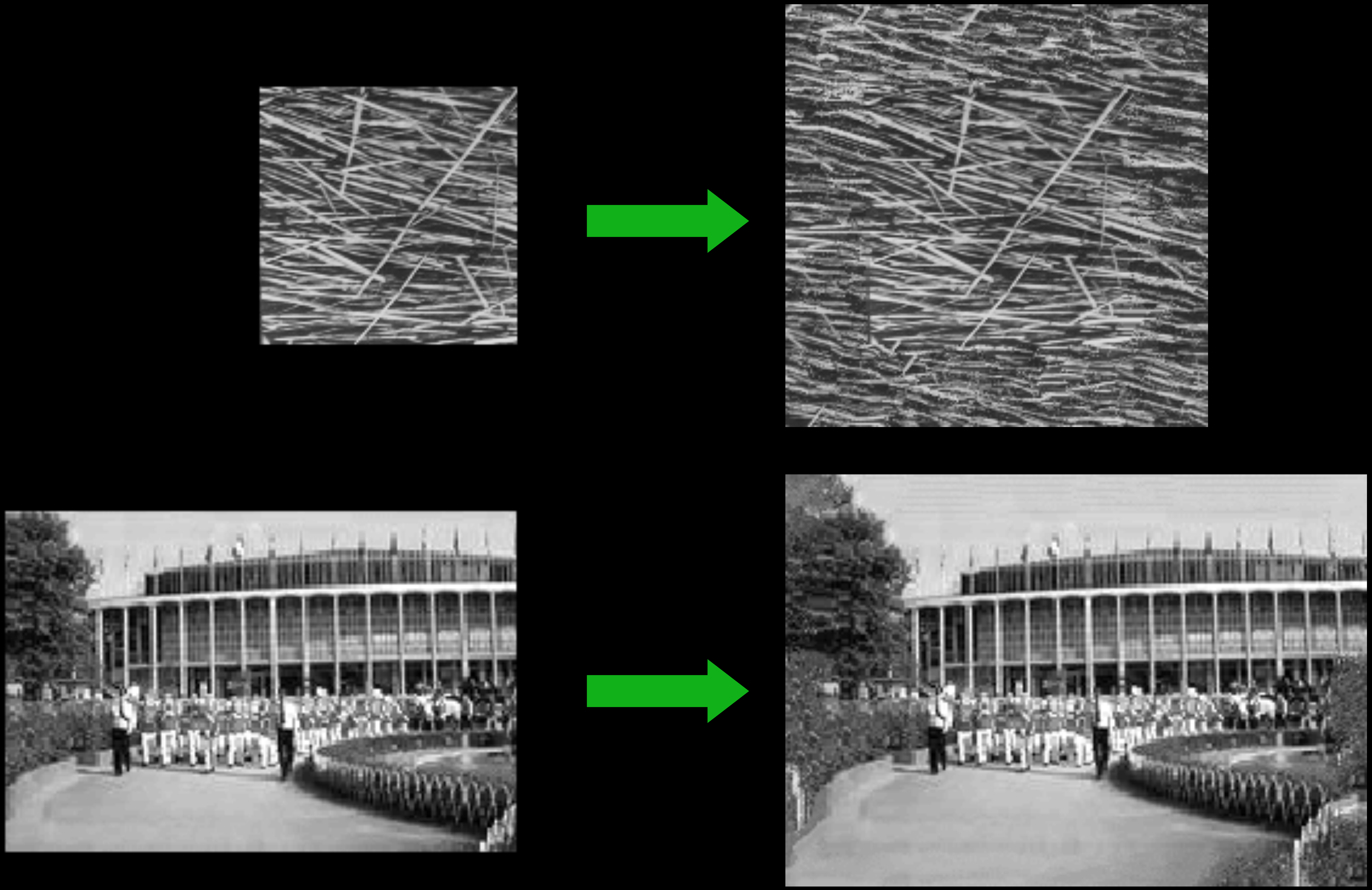
Source: A. Efros

Hole Filling



Source: A. Efros

Extrapolation

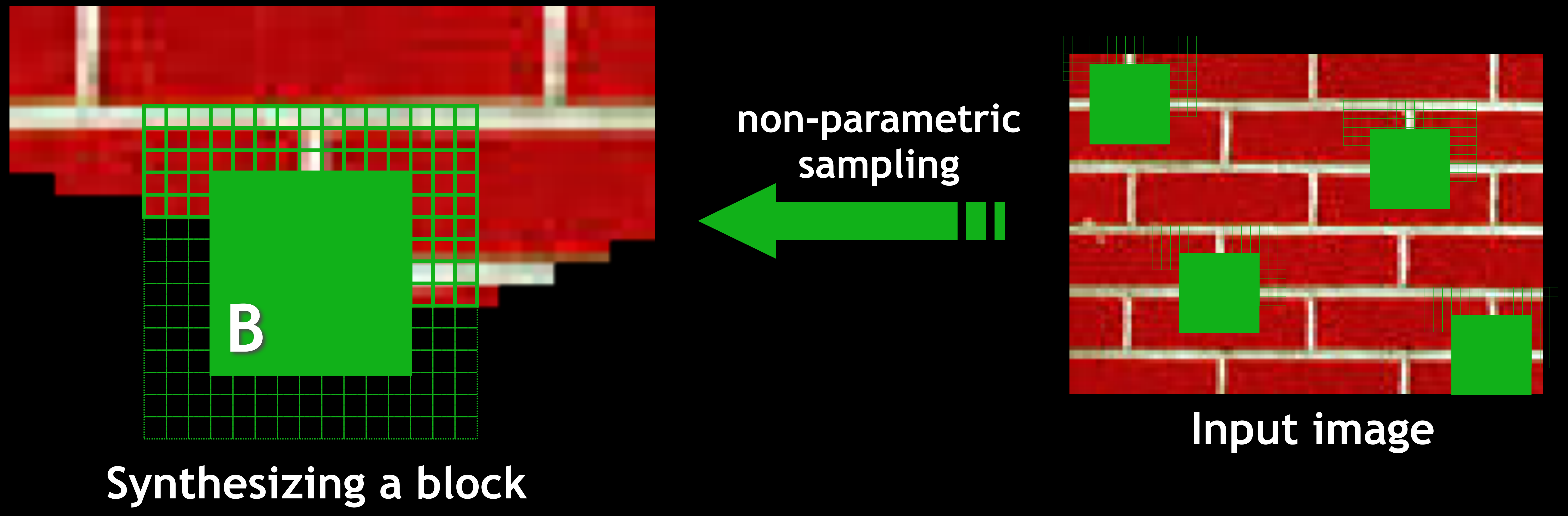


Source: A. Efros

Summary

- The Efros & Leung algorithm
  - Very simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - …but very slow

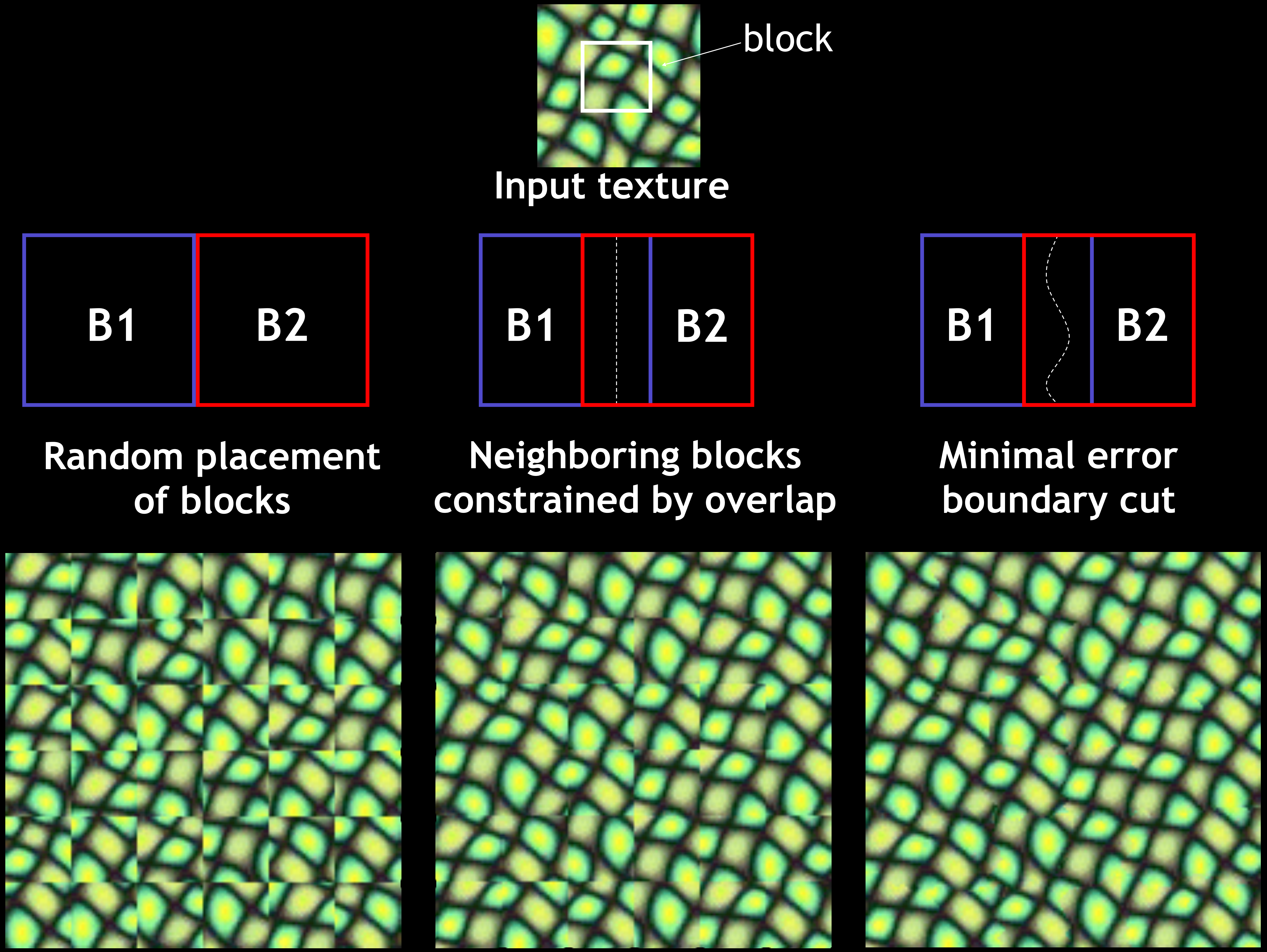
Image Quilting [Efros & Freeman]



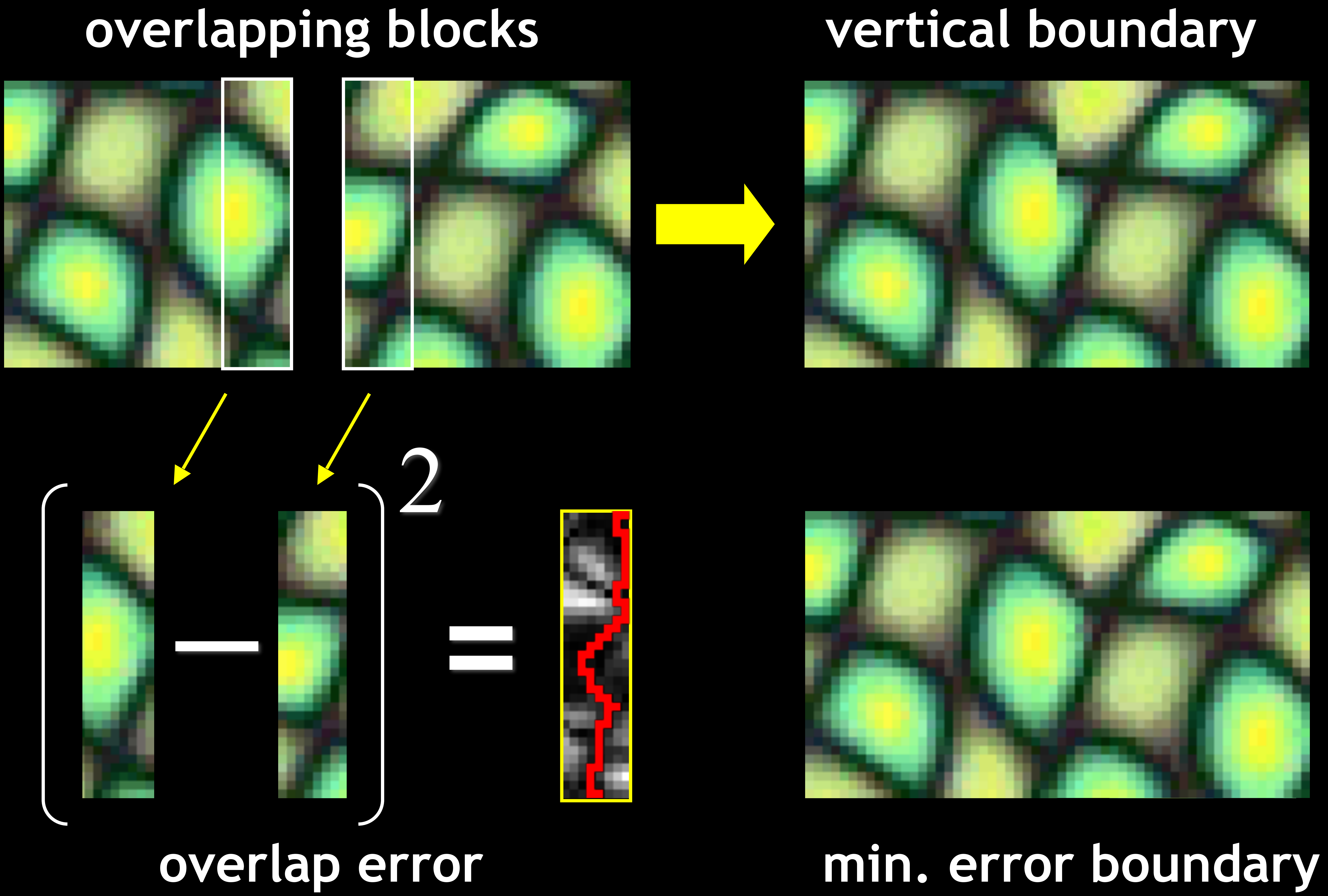
- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

- Exactly the same but now we want P(**B**|N(**B**))
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!


block
Input texture
B1
B2
Random placement of blocks
B1
B2
Neighboring blocks constrained by overlap
B1
B2
Minimal error boundary cut

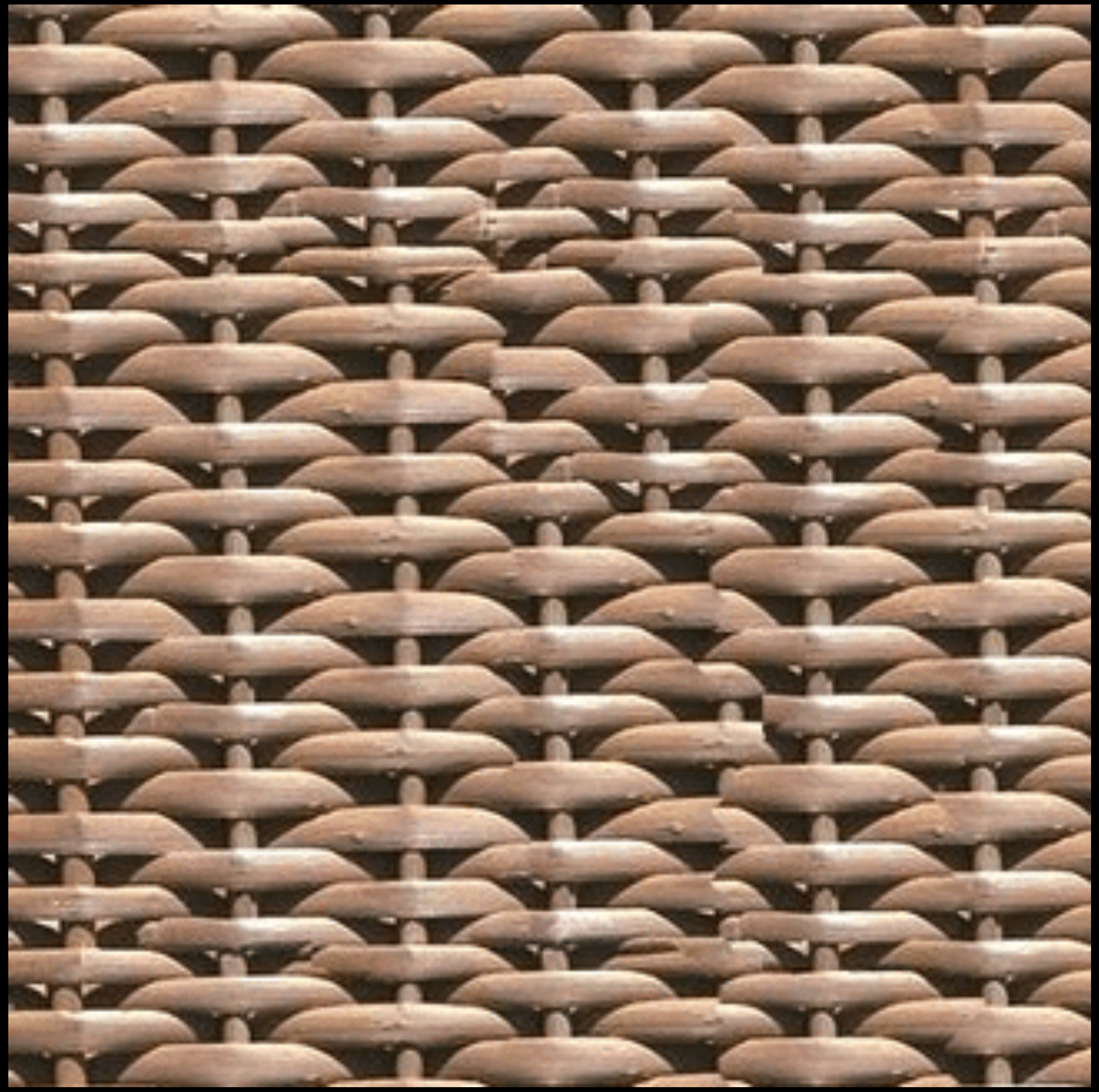
Minimal error boundary



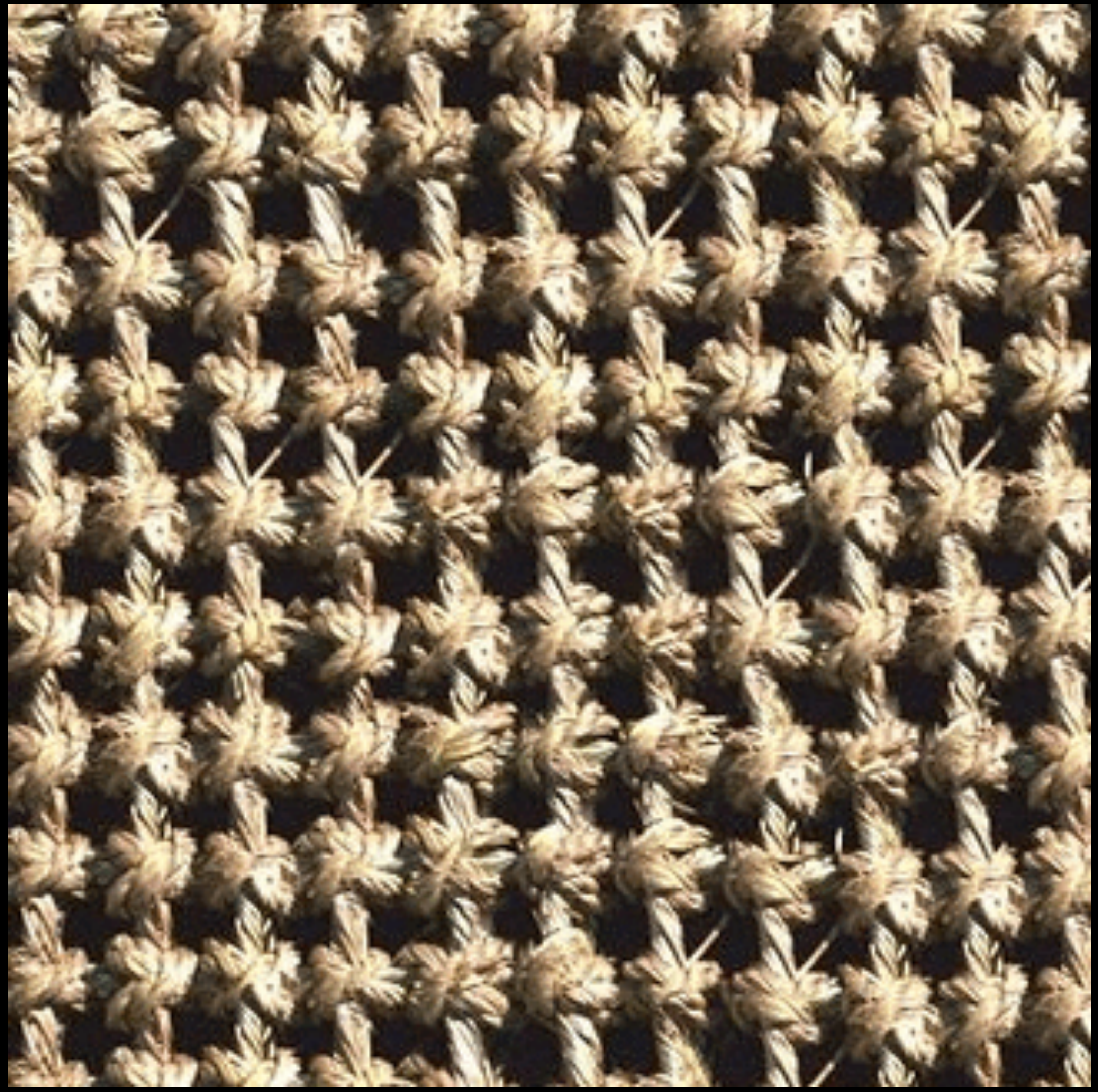
Our Philosophy

- The "Corrupt Professor's Algorithm":
  - Plagiarize as much of the source image as you can
  - Then try to cover up the evidence
- Rationale:
  - Texture blocks are by definition correct samples of texture so problem only connecting them together







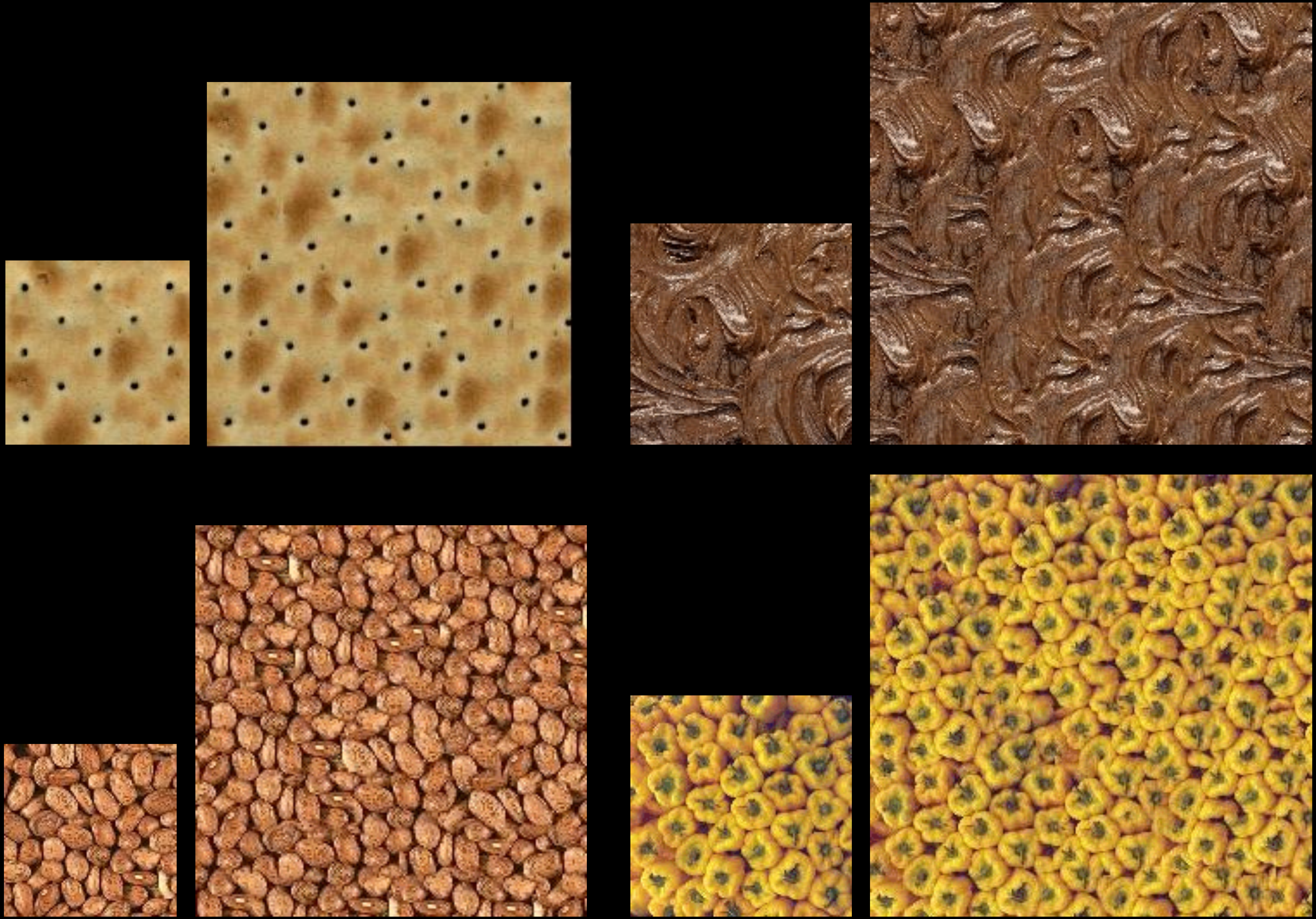





College Inn







Failures
(Chernobyl
Harvest)

Texture Transfer

Constraint

Texture sample