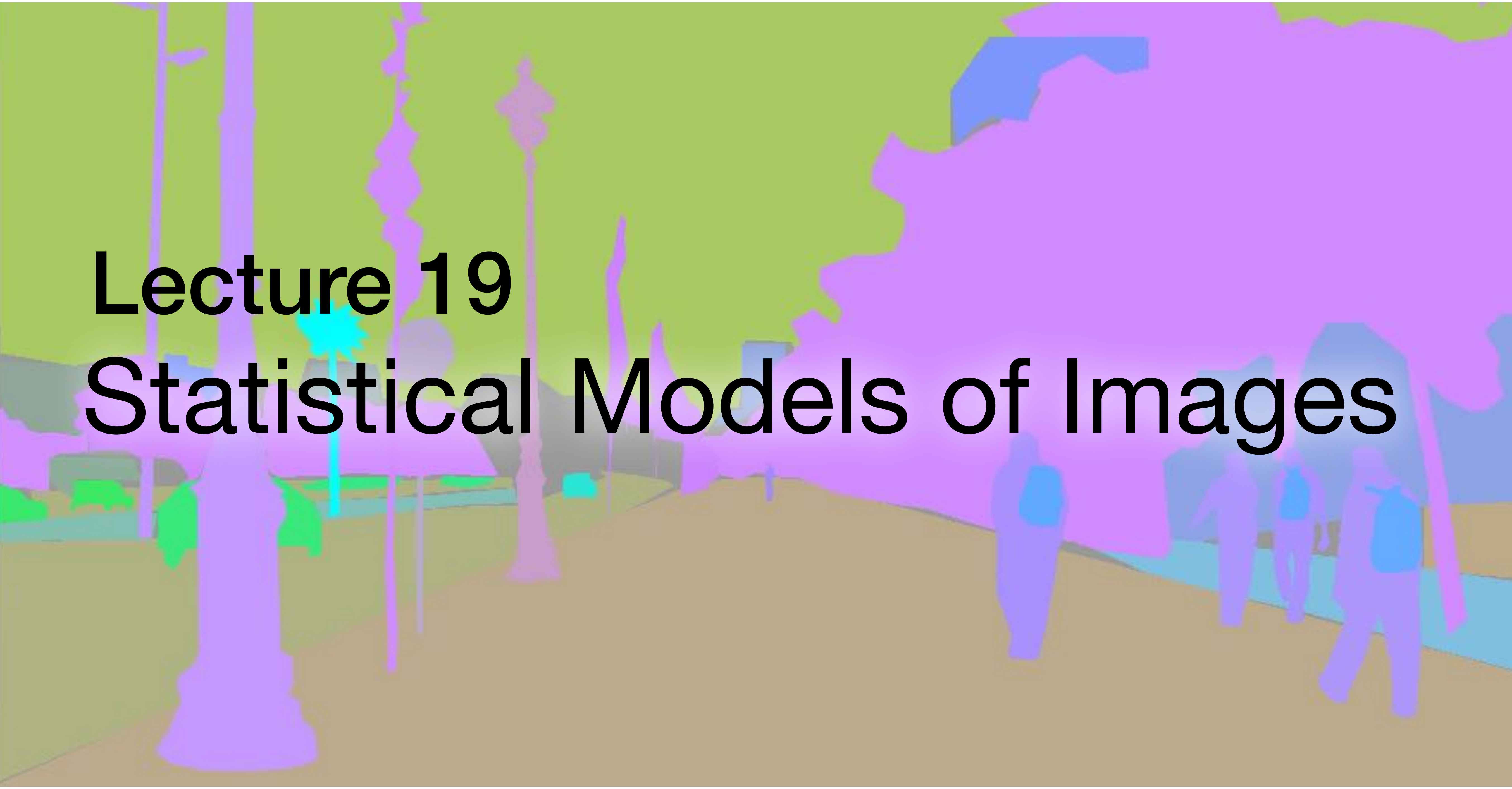


Lecture 19
Statistical Models of Images

6.3800/6.3801 Advances in Computer Vision

spring 2023

Freeman, Sitzmann, Konakovic Lukovic

April 20, 2023

The visual system seems to be tuned to a particular subset of all possible images:

Demo inspired from D. Field

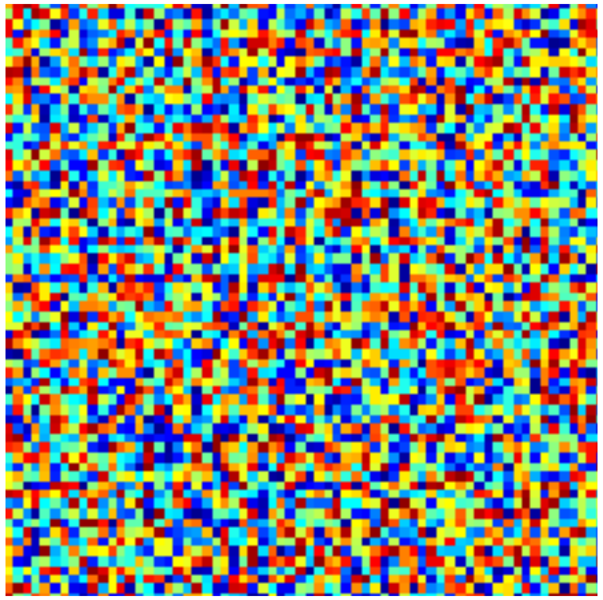
Remember each of these images

Was this one of them?

Remember each of these images

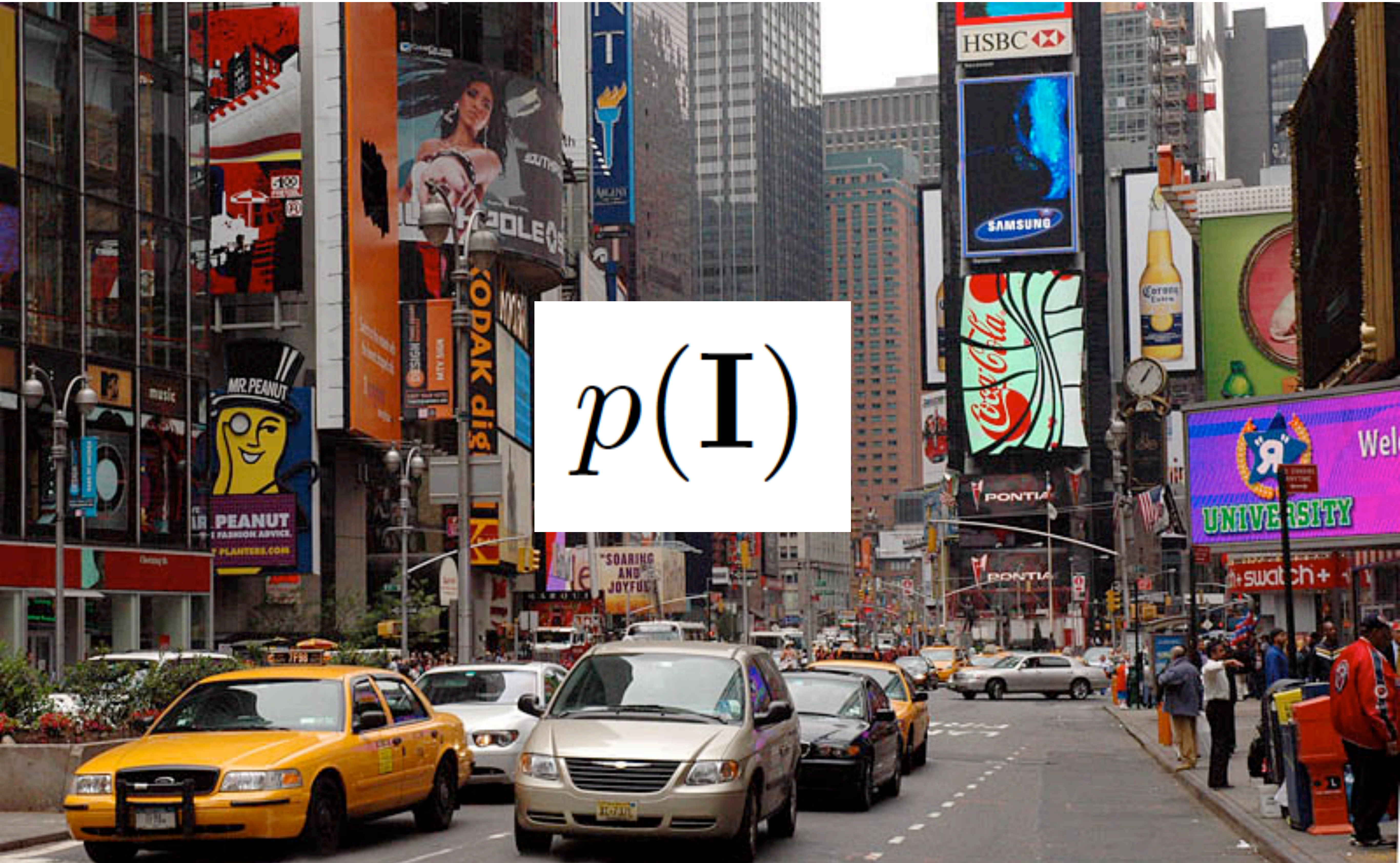
Test 2

Was this one of them?



The visual system is tuned to process structures typically found in the world.

Statistical modeling of images



$$p(\mathbf{I})$$

Visual Worlds

Visual Worlds

Visual Worlds

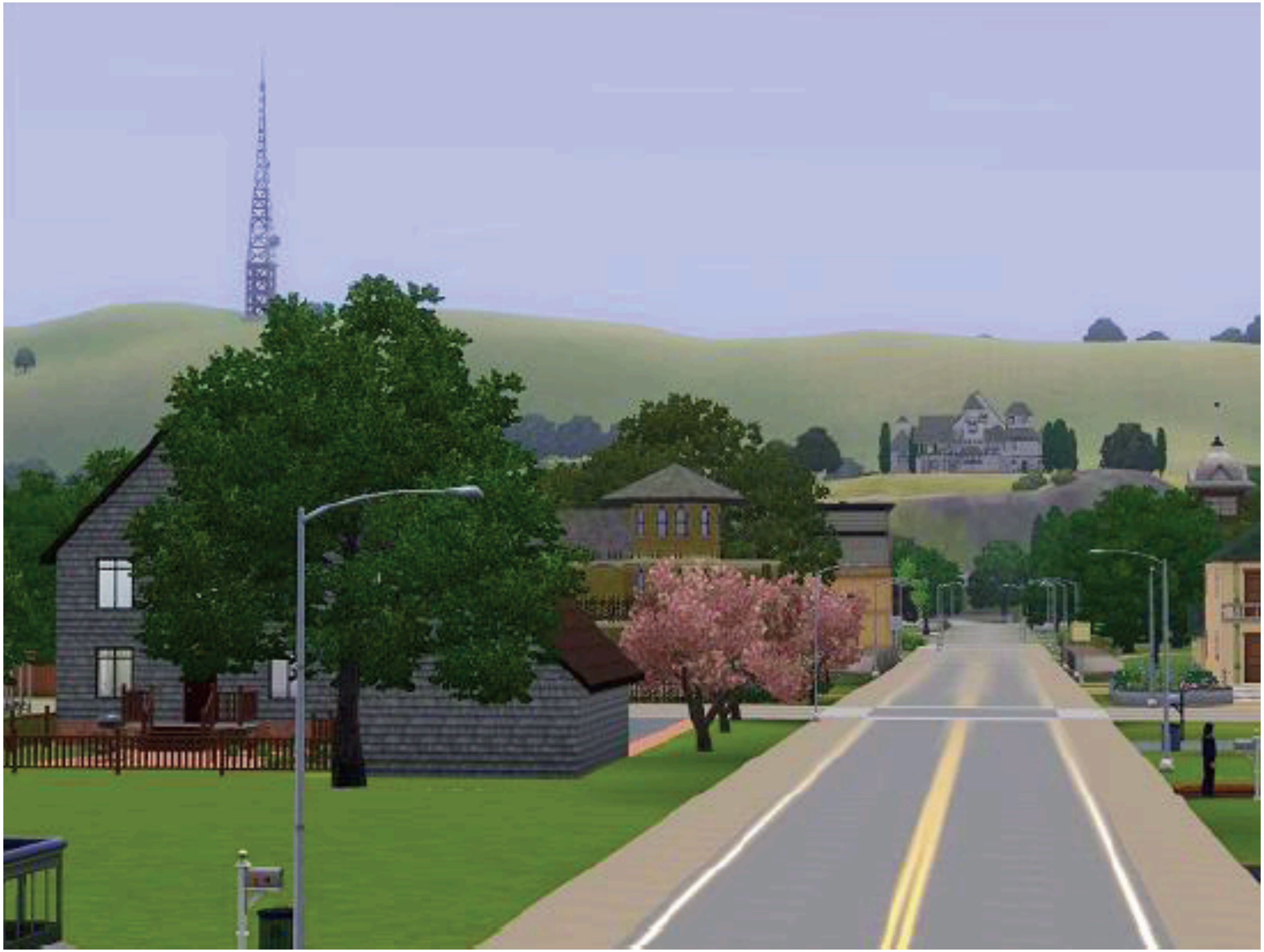
Visual Worlds

Visual Worlds

Visual Worlds

Visual Worlds

Visual Worlds



Visual Worlds

Statistical Image Models

- Gaussian image model
  - image synthesis
  - Wiener filter denoising
- Kurtotic wavelets model
  - image synthesis
  - Bayesian denoising
- Non-parametric MRF model
  - image synthesis (Efros and Leung texture model)
  - Non-local means denoising

Statistical Image Models

- Gaussian image model
  - image synthesis
  - Wiener filter denoising
- Kurtotic wavelets model
  - image synthesis
  - Bayesian denoising
- Non-parametric MRF model
  - image synthesis (Efros and Leung texture model)
  - Non-local means denoising

Statistical modeling of images

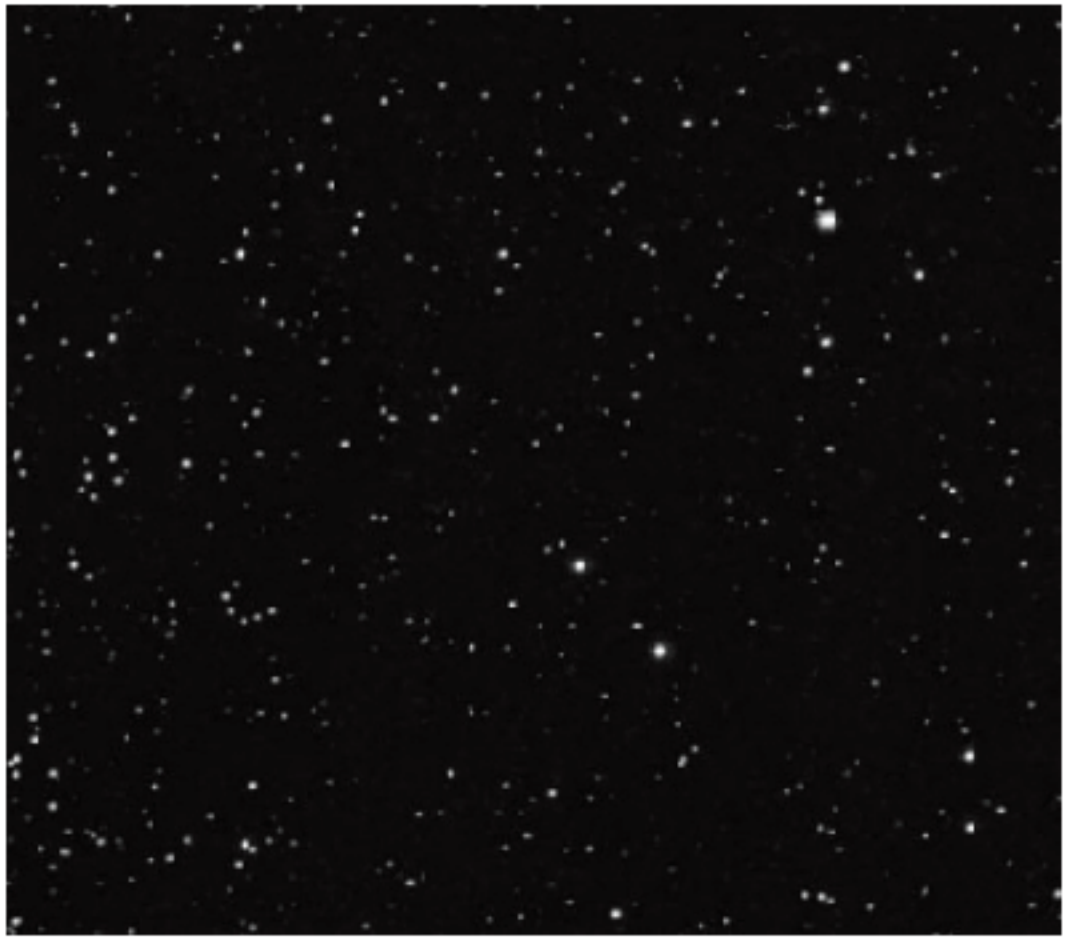
0th image model: independent pixels

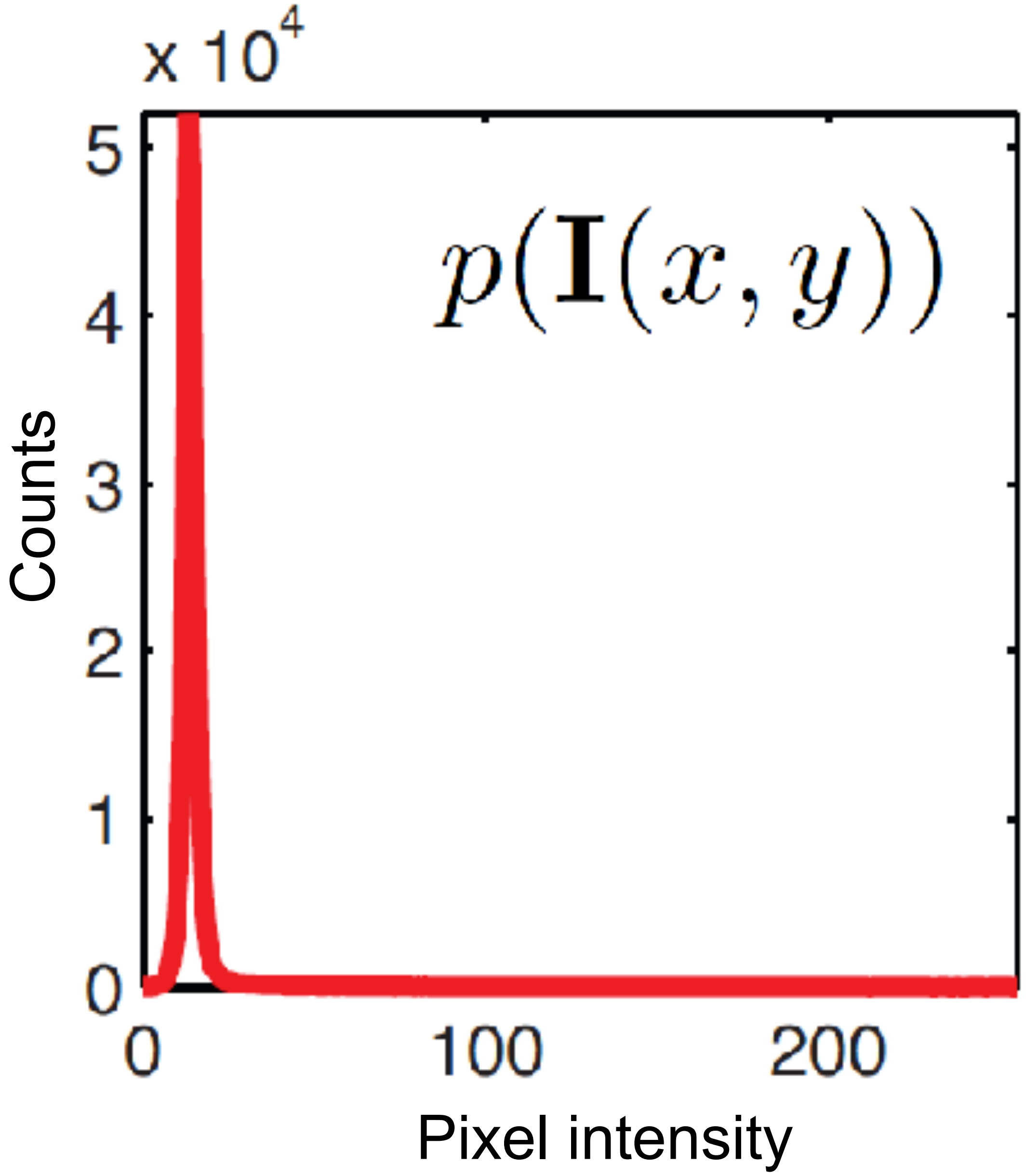
$$p(\mathbf{I}) = \prod_{x,y} p(\mathbf{I}(x,y))$$

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

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

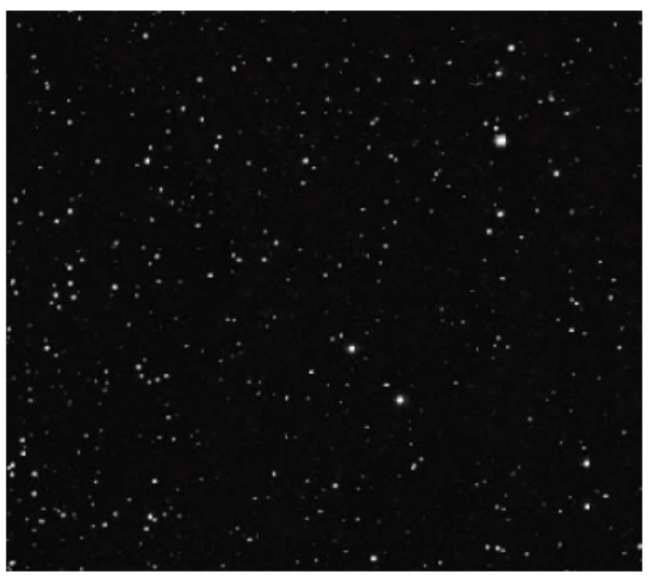
Fitting the model

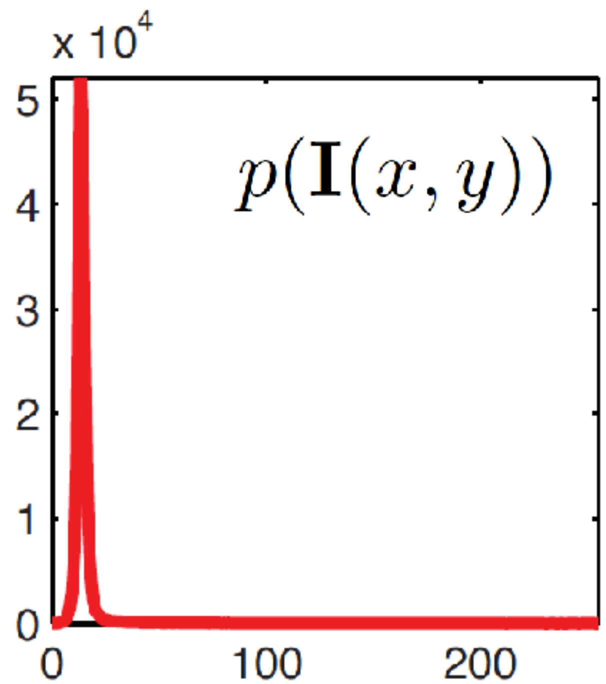




Sampling new images

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

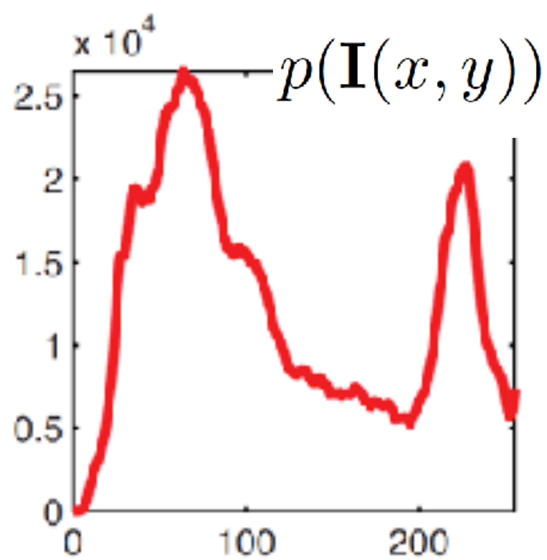


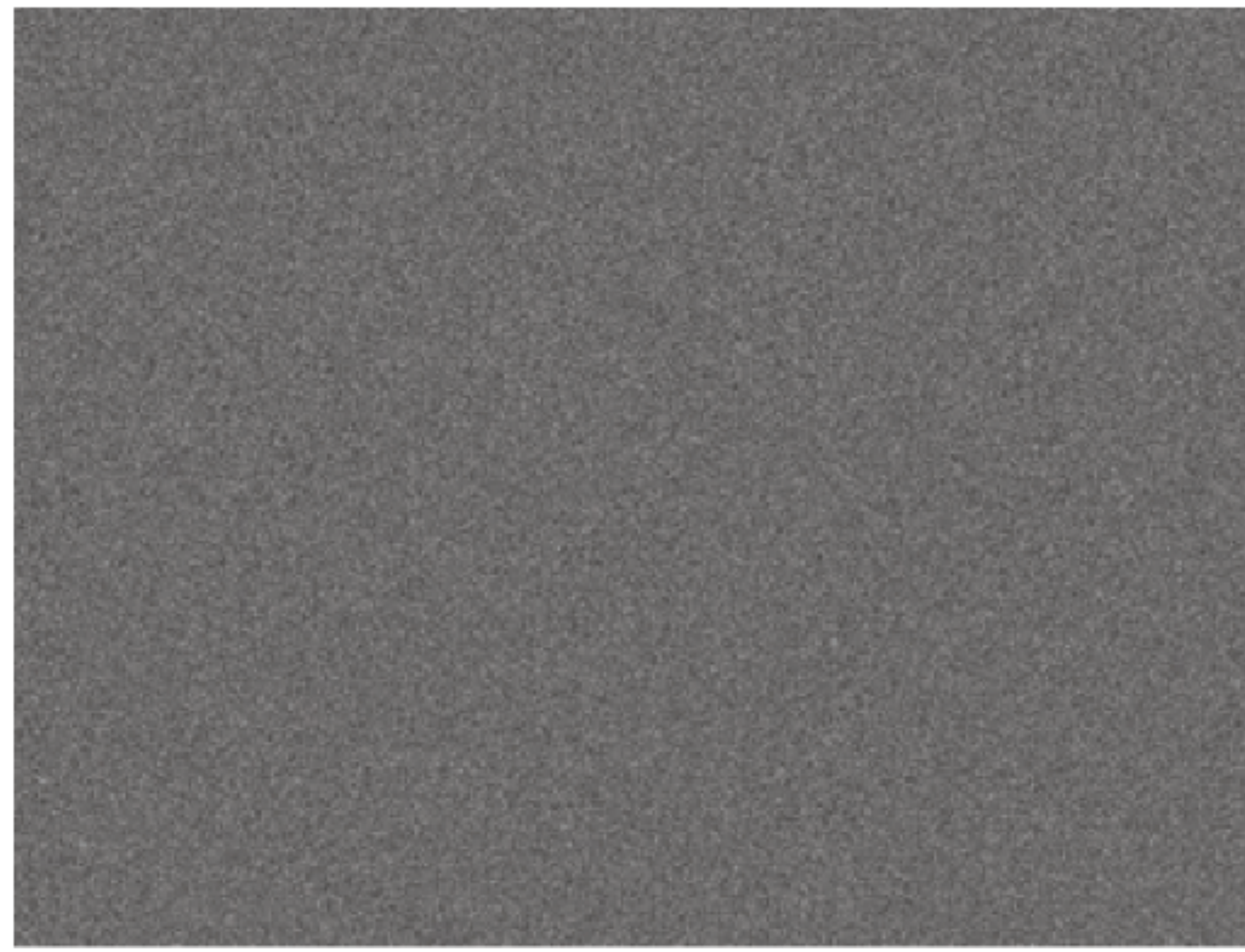


Sample

Sampling new images

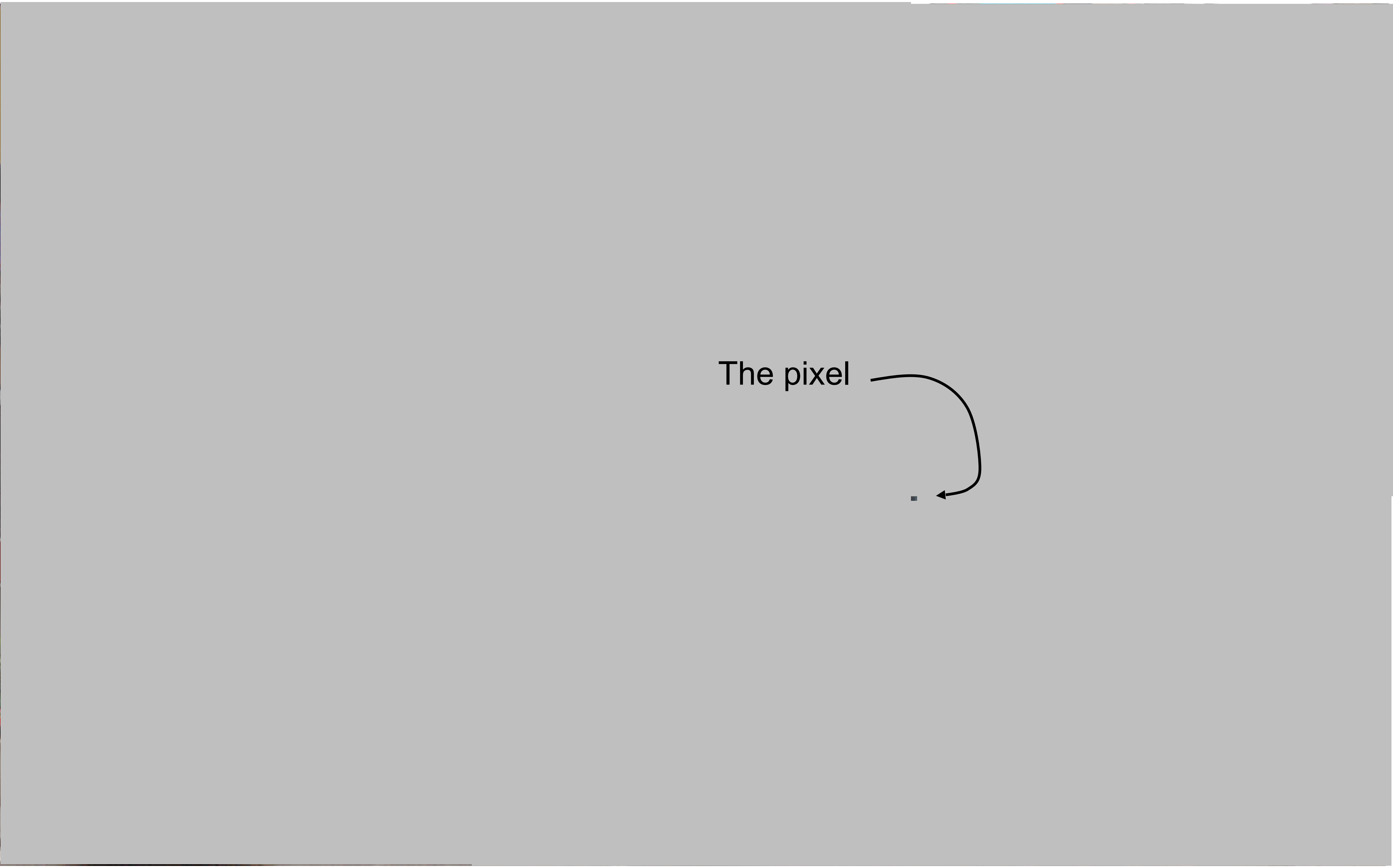
$$p(\mathbf{I}) = \prod_{x,y} p(\mathbf{I}(x,y))$$



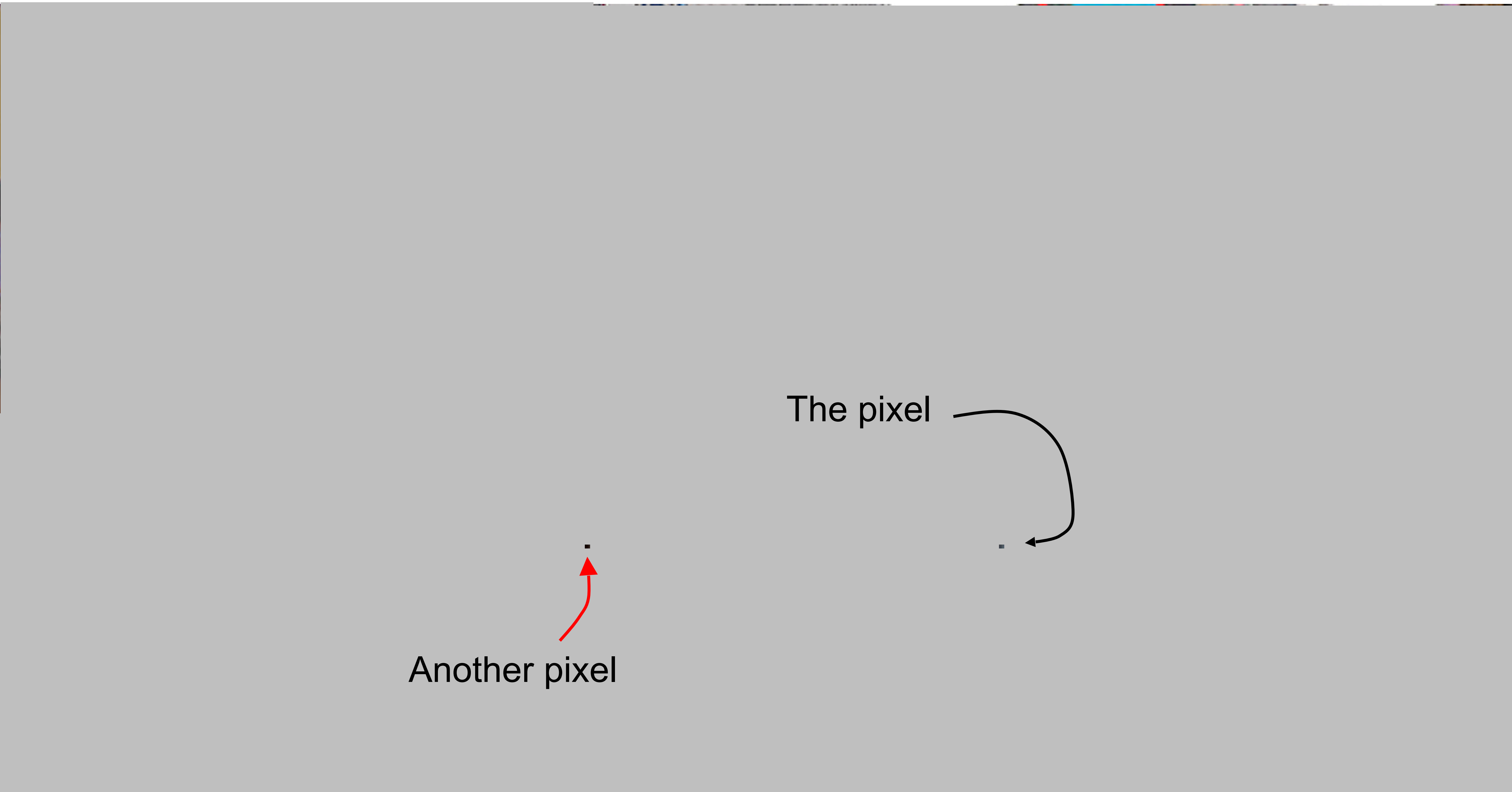


Sample

0th model

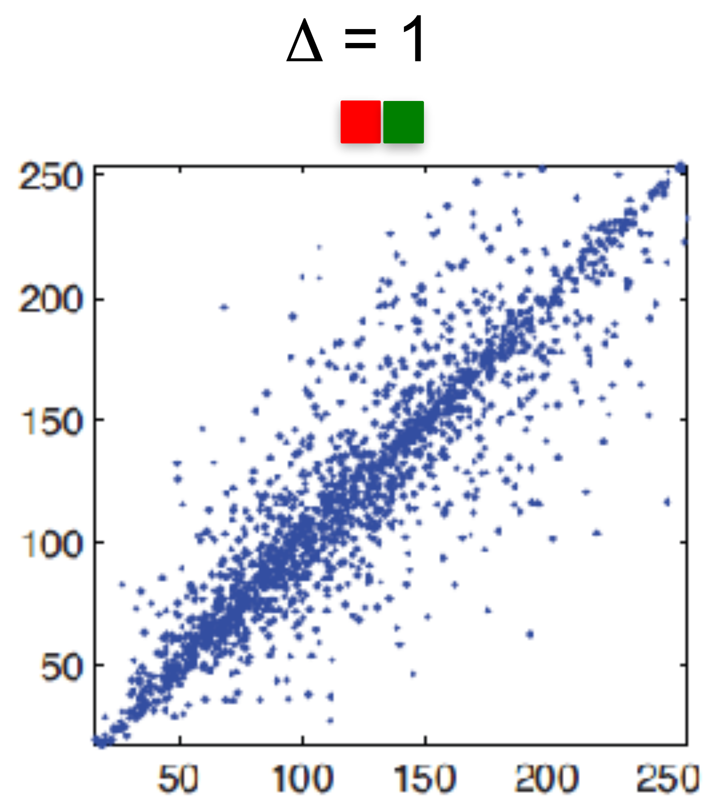


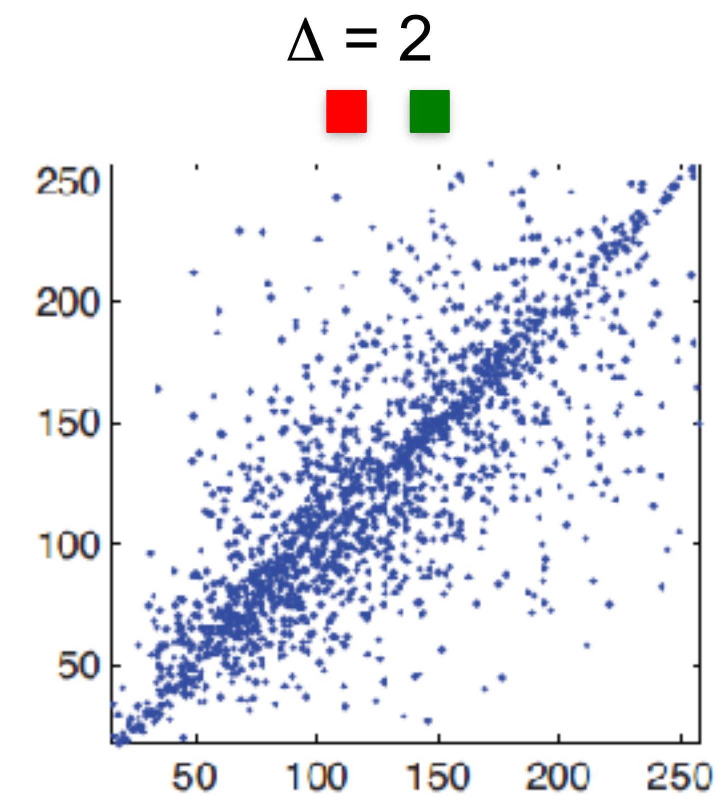
1st model: include pixel correlations

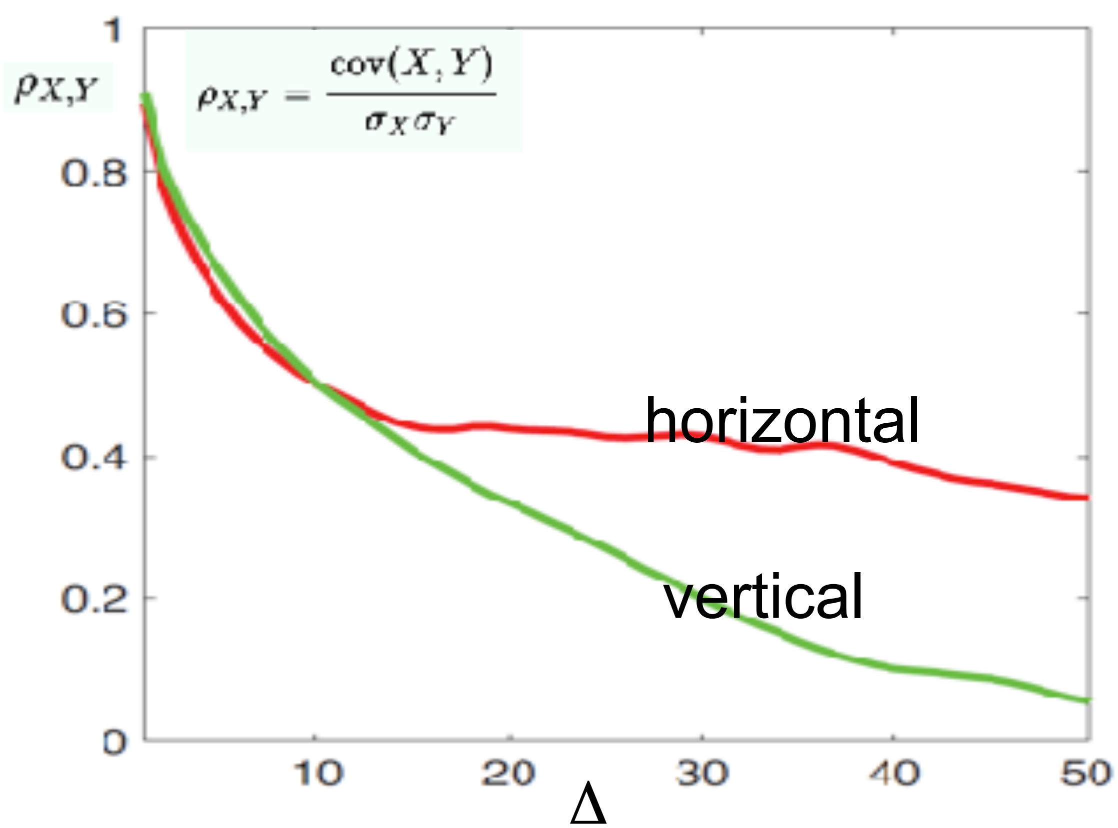


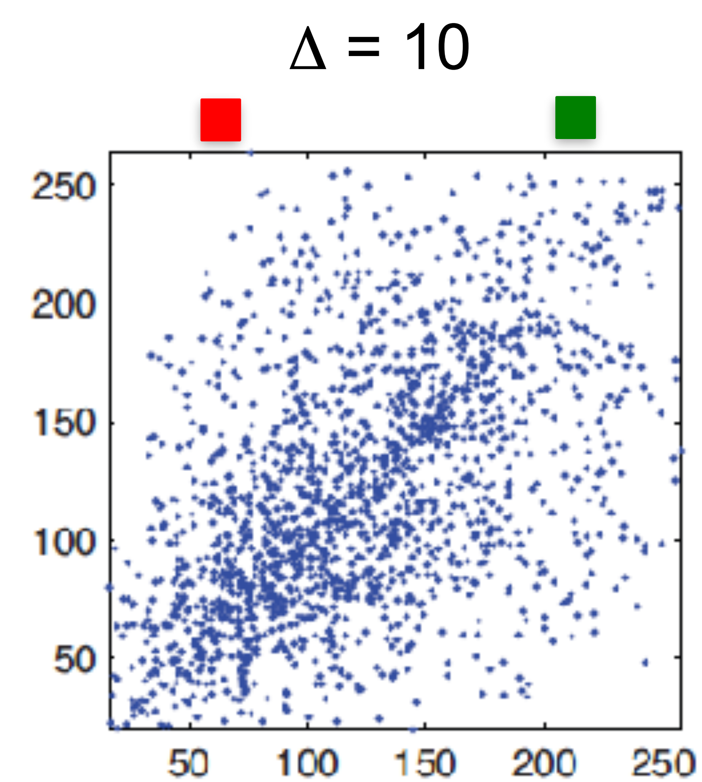
$$C(\Delta x, \Delta y) = \mathrm{E}[\mathbf{I}(x + \Delta x, y + \Delta y)\ \mathbf{I}(x, y)]$$

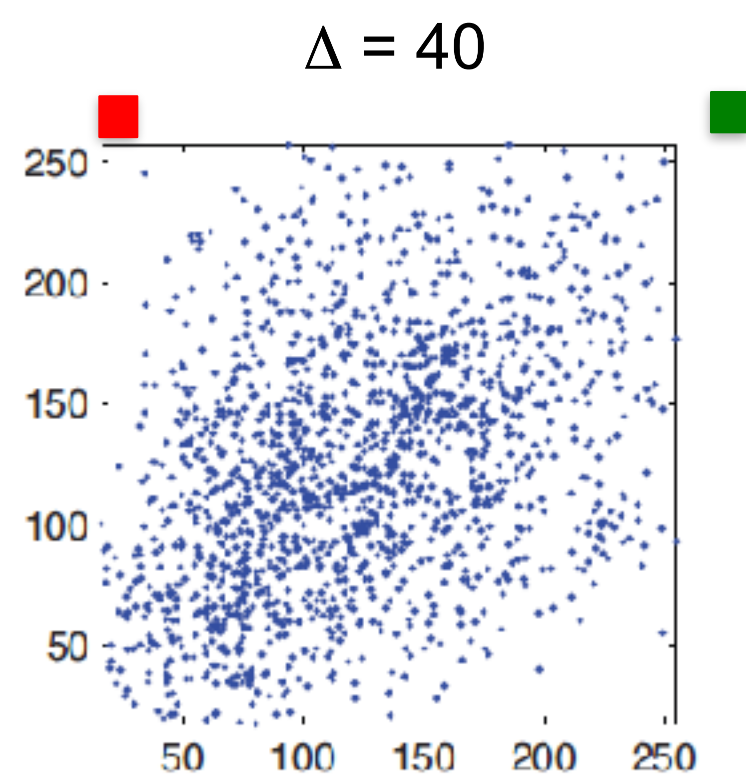
$$C(\Delta x, \Delta y) = \mathrm{E}\left[\mathbf{I}(x + \Delta x, y + \Delta y)\ \mathbf{I}(x, y)\right]$$











By the Wiener-Khinchin theorem, the Fourier transform of the auto-correlation function of the image is the power spectrum of the image, so the regularity in covariances leads to a regularity in natural image power spectra.

A remarkable property of natural images

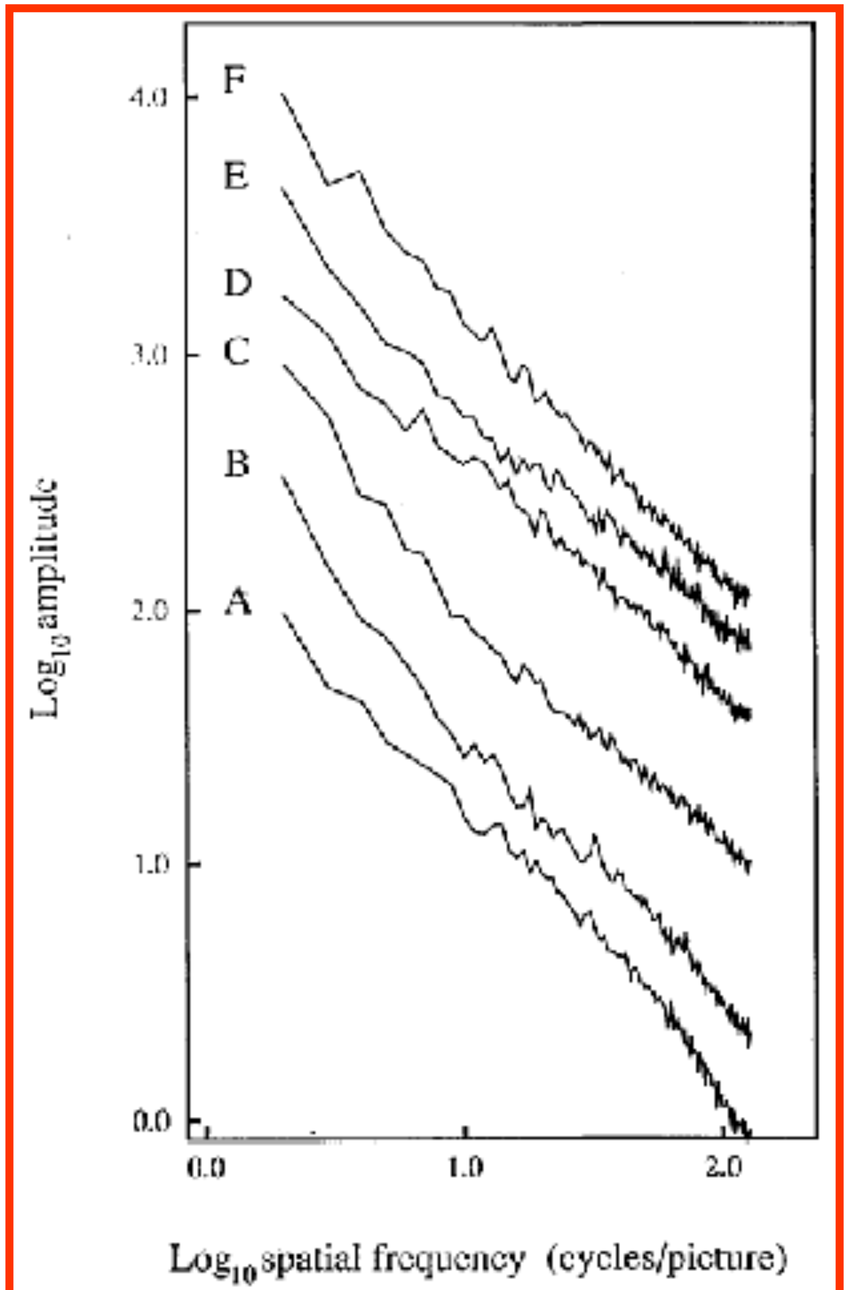
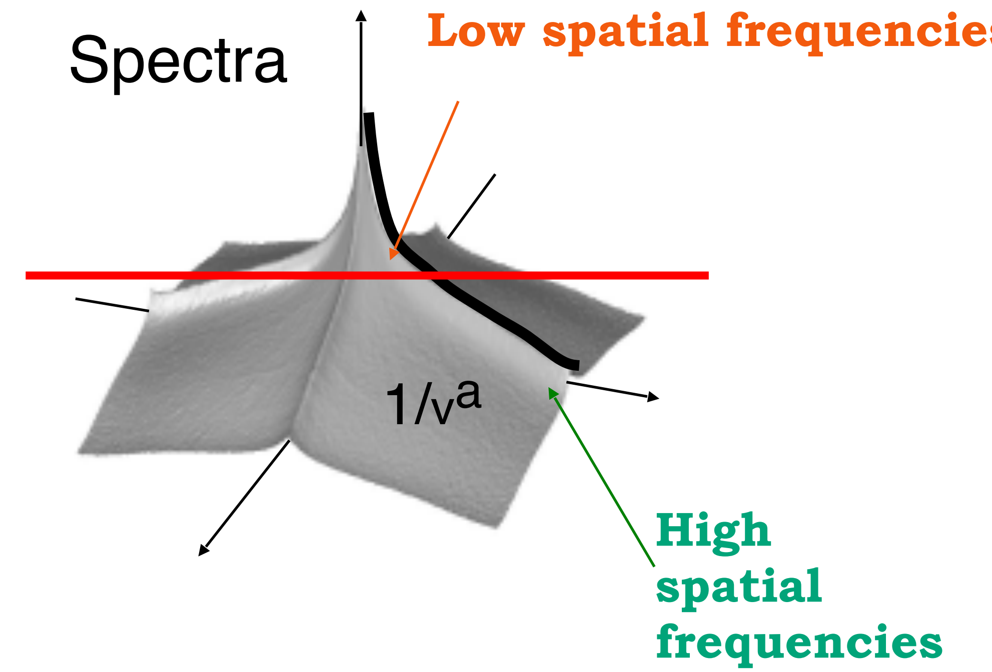


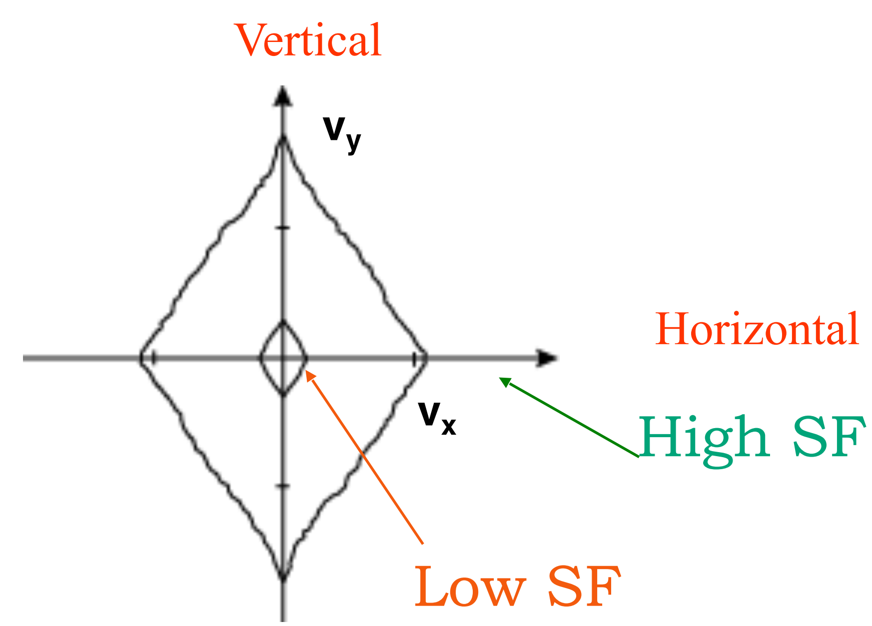
Fig. 8. Amplitude spectra for the six images A–F, averaged across all orientations. The spectra have been shifted up for clarity. On these log–log coordinates the spectra fall off by a factor of roughly 1/f (a slope of −1). Therefore the power spectra fall off as $1/f^2$.

Amplitude spectrum fall off as

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

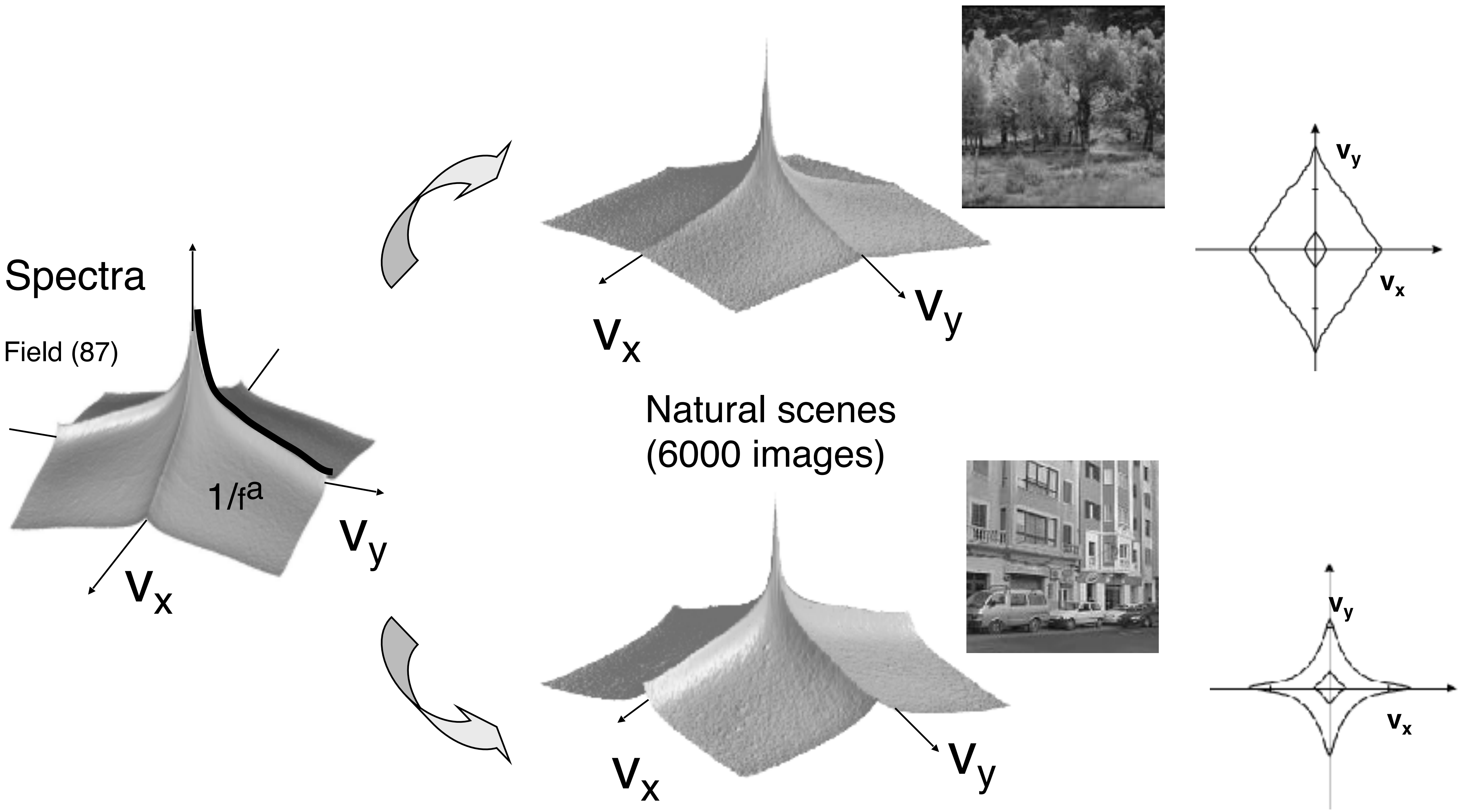
$\alpha \approx 1$





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)

A remarkable property of natural images



Man-made scenes
(6000 images)

Torralba and Oliva, *Statistics of Natural Image Categories*. Network: Computation in Neural Systems 14 (2003) 391-412.

Gaussian image model:
power spectrum is all that matters

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

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

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

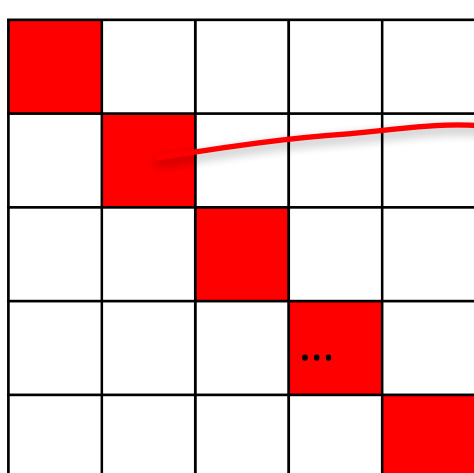
Stationarity assumption: Symmetrical circulant matrix

Diagonalization of circulant matrices: $C = EDE^T$

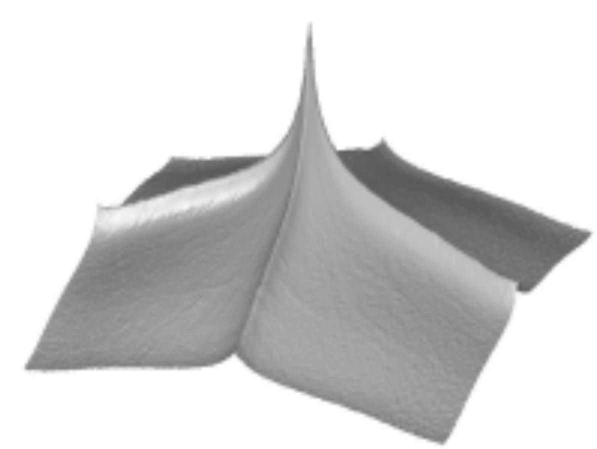
The eigenvectors are the Fourier basis

The eigenvalues are the squared magnitude of the Fourier coefficients

D=

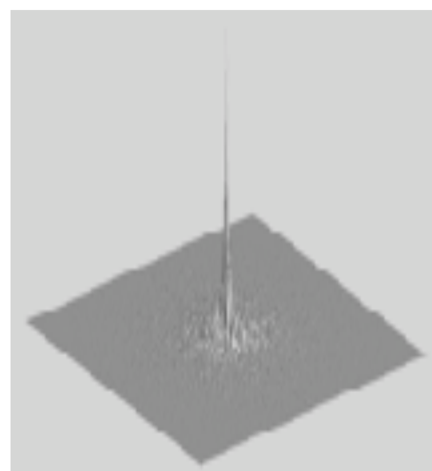


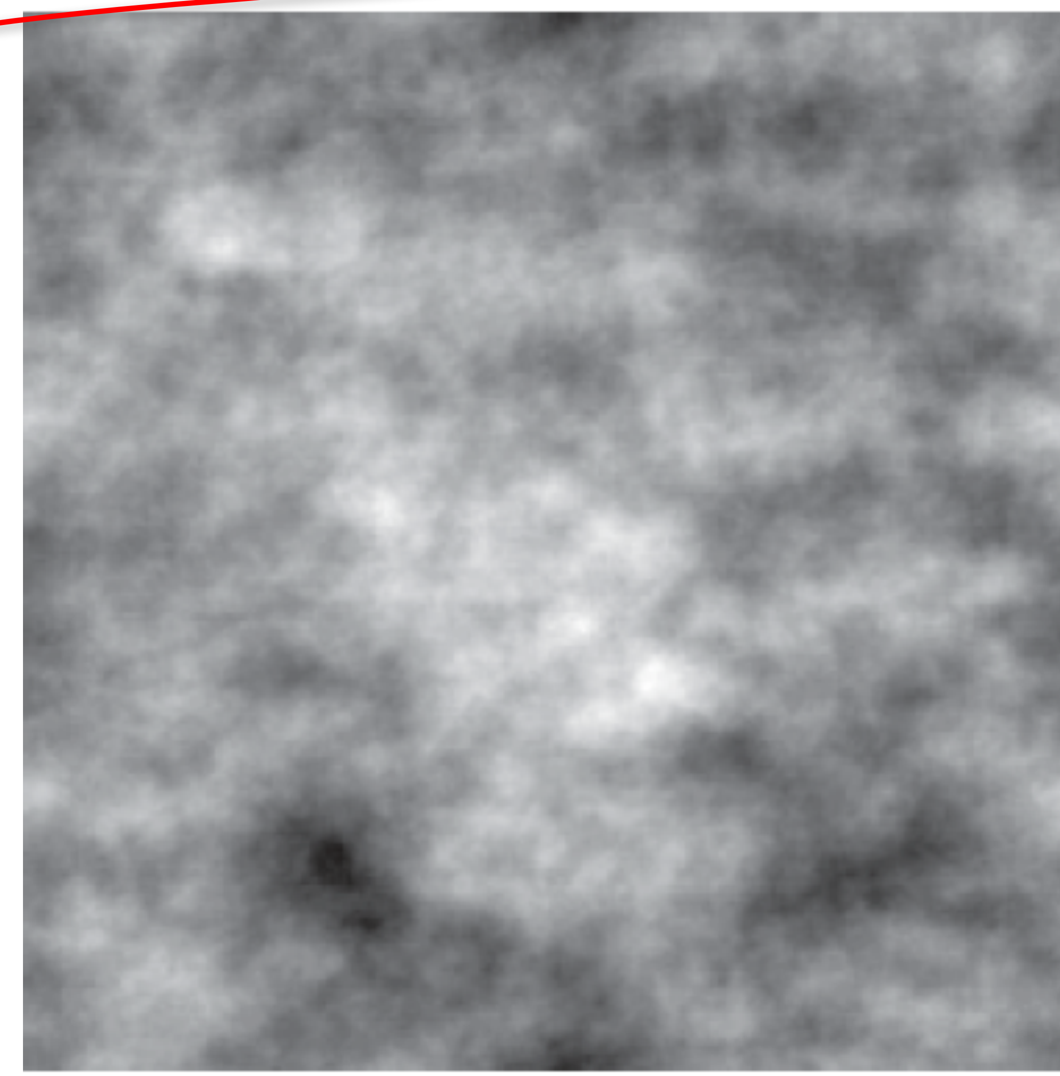
$$\left|\hat{\mathbf{I}}(v)\right|^2 \simeq \frac{1}{|v|^{\alpha}}$$



Sampling new images

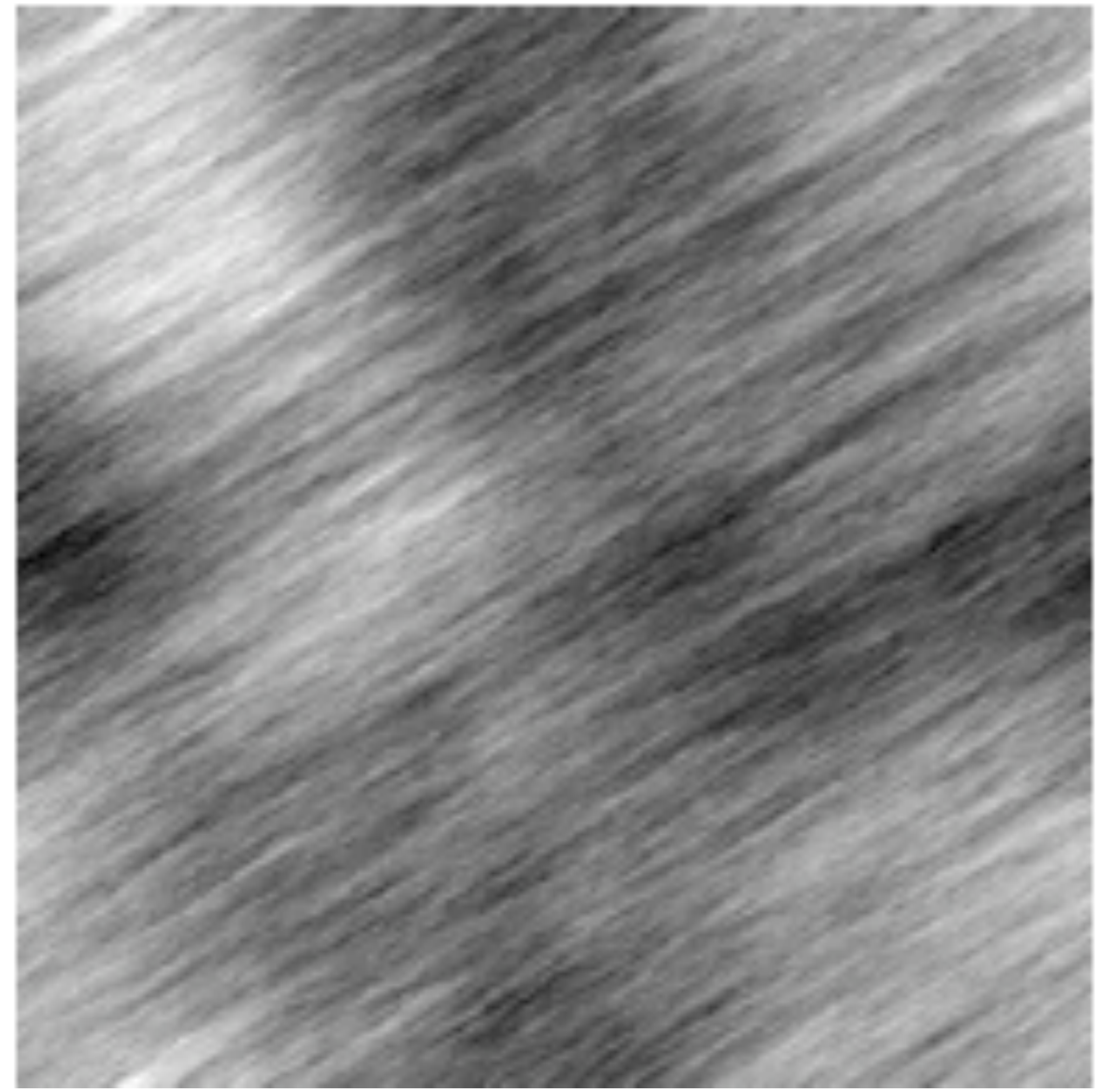
$$p(\mathbf{I}) = \exp\left(-\frac{1}{2}\mathbf{I}^T\mathbf{C}^{-1}\mathbf{I}\right)$$





Sample

Sampling new images



Randomizing the phase (if you fit the Gaussian image model to each of the images in the top row, then draw another random sample, you get the bottom row)

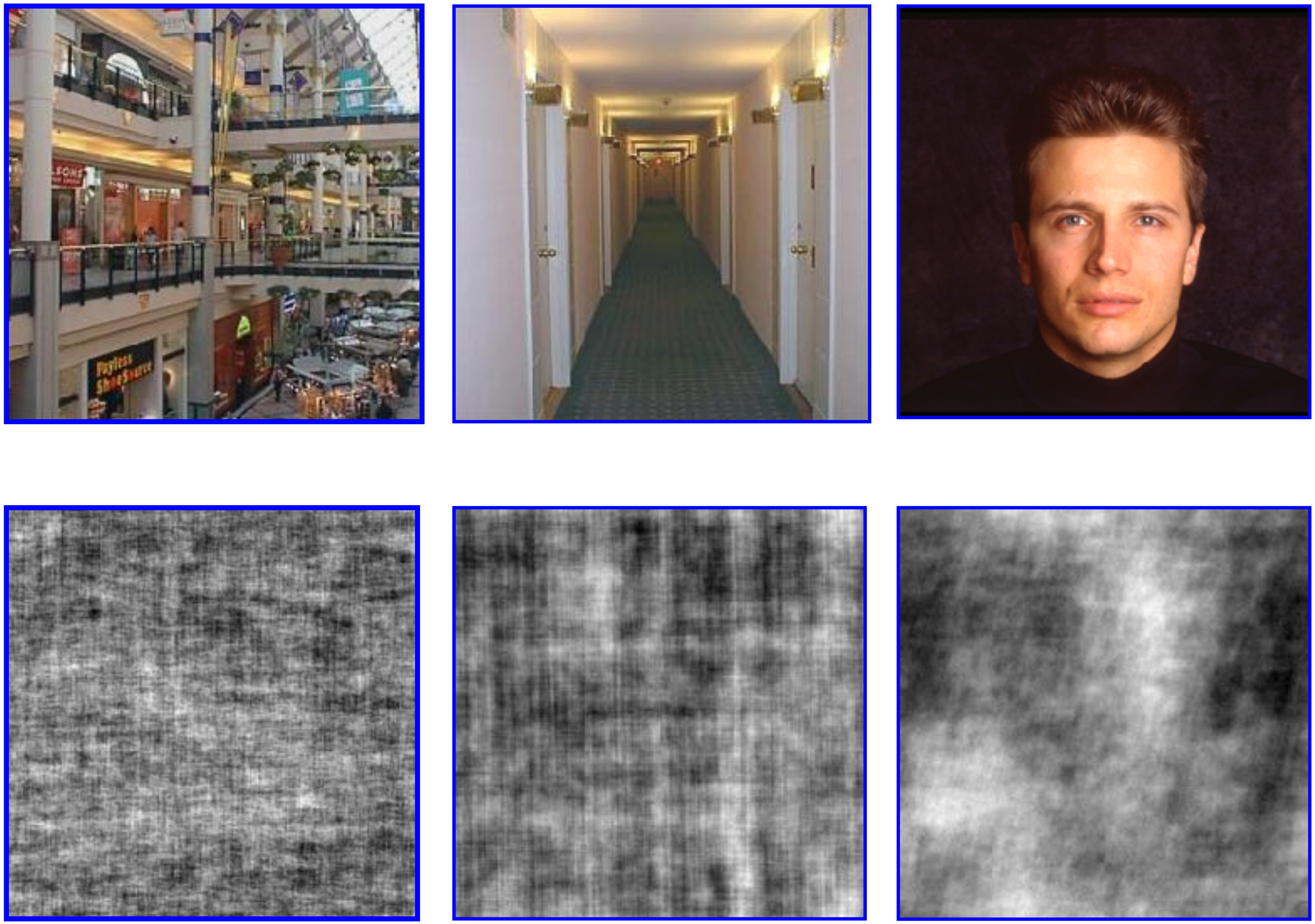
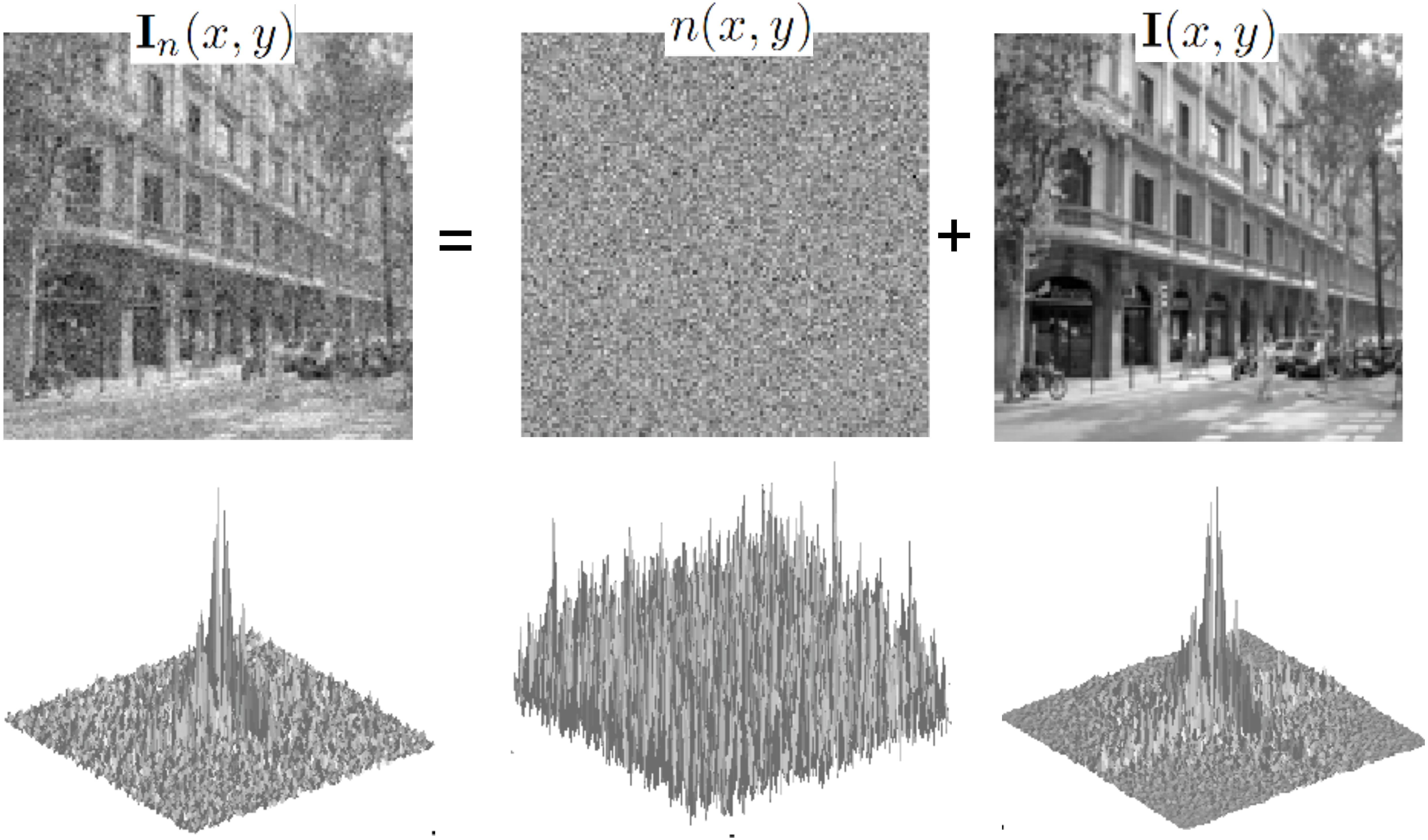


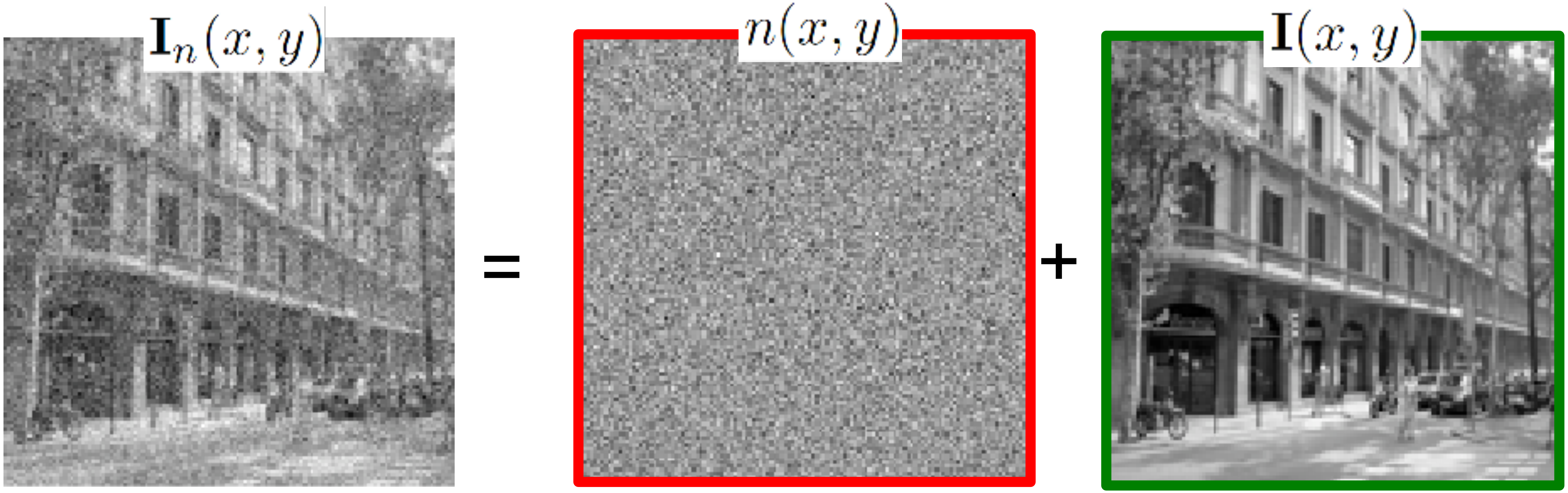
Image model application: Denoising

Desired decomposition of a noisy image:



Denoising

Desired decomposition of a noisy image:



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

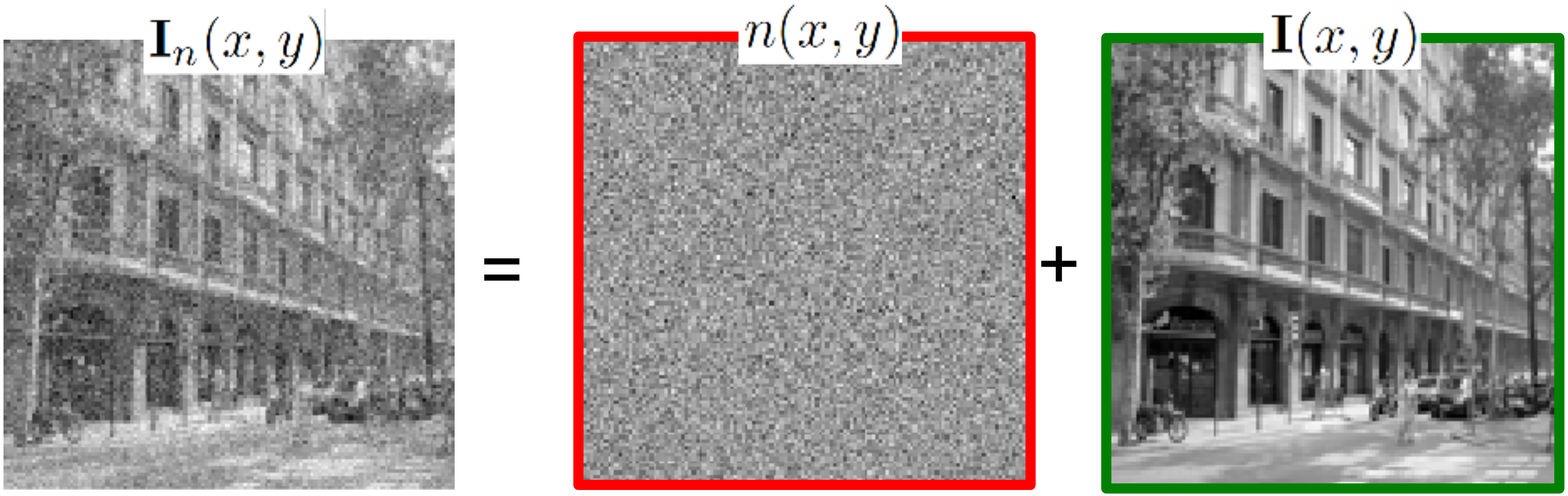
Find I(x,y) that maximizes the posterior (maximum a posteriori, MAP):

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

likelihood prior

Denoising

Desired 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 posteriori, MAP):

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

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

Denoising

$$\max_{\mathbf{I}} p(\mathbf{I}|\mathbf{I}_n) = \max_{\mathbf{I}} \quad p(\mathbf{I}_n|\mathbf{I}) \quad \text{x} \quad p(\mathbf{I})$$

likelihood prior

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

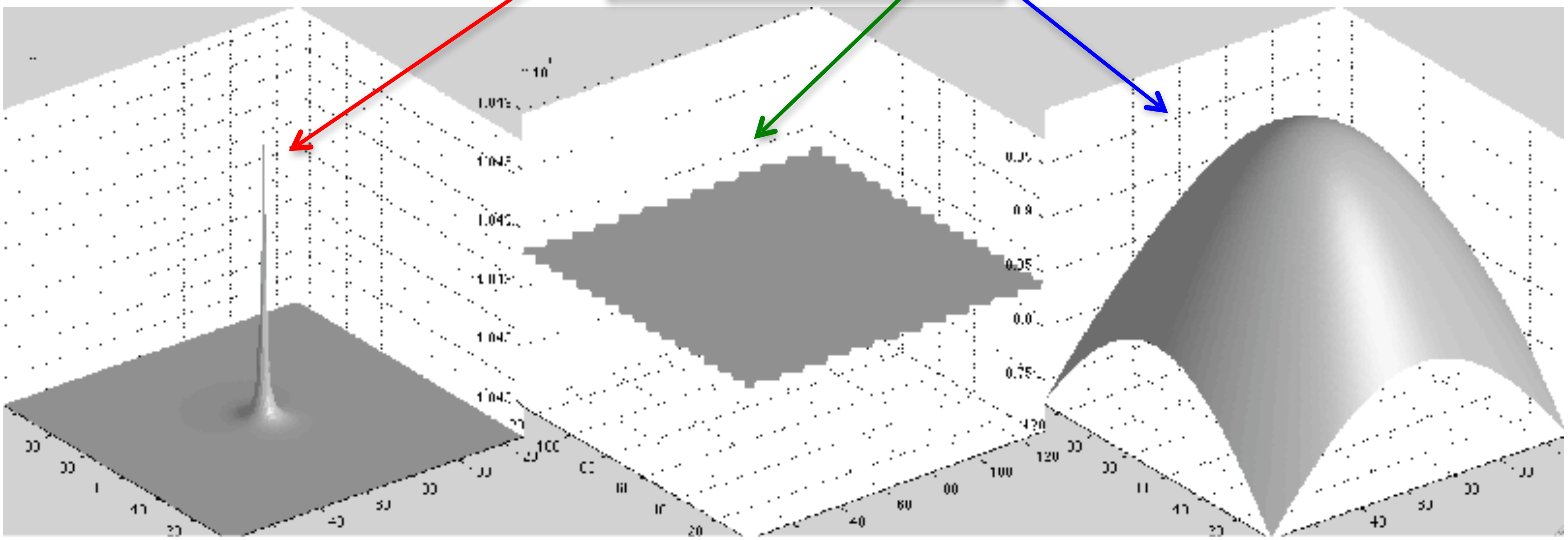
The solution is:

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

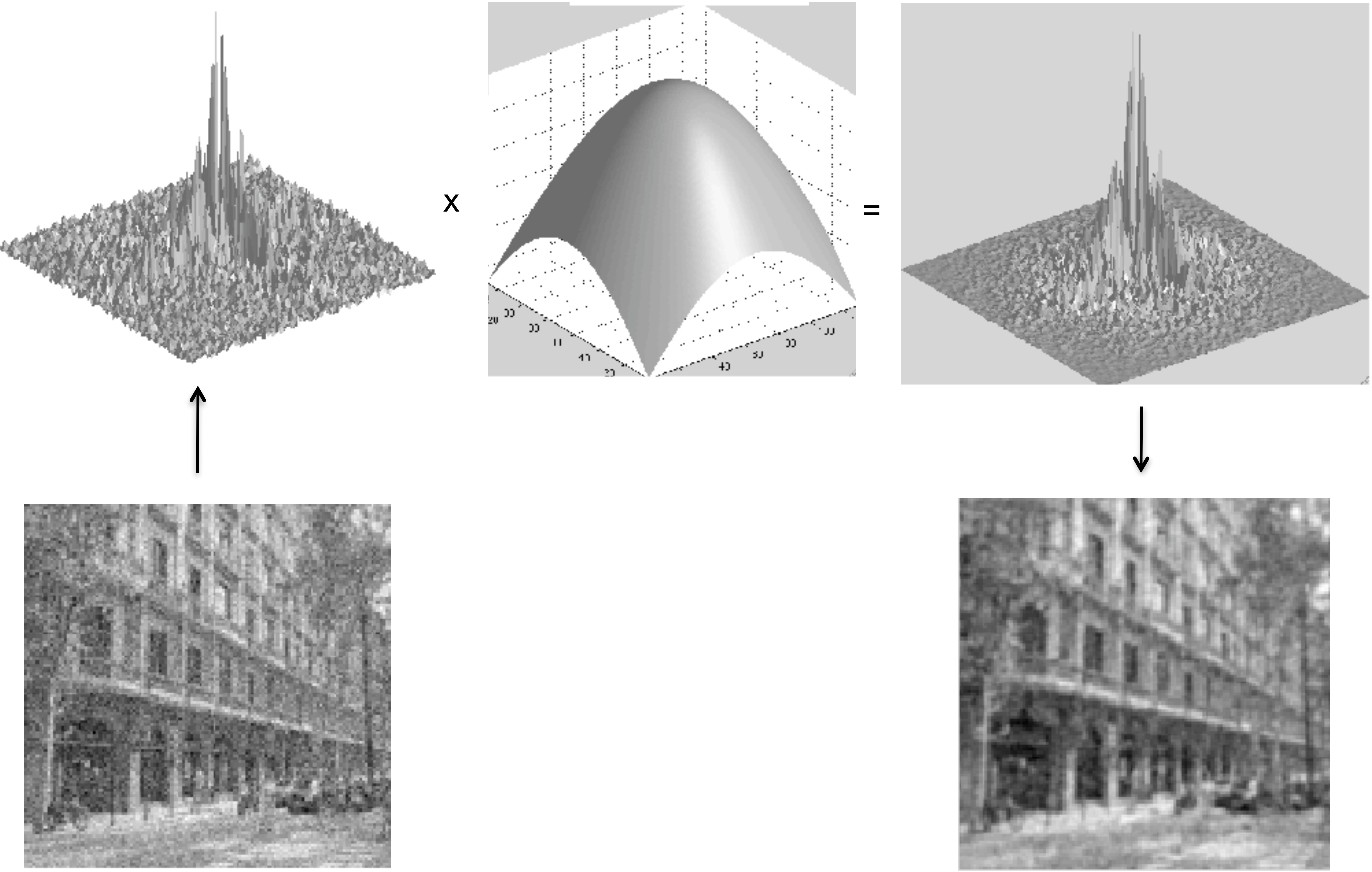
This can also be written in the Fourier domain, with C = EDE^T:

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

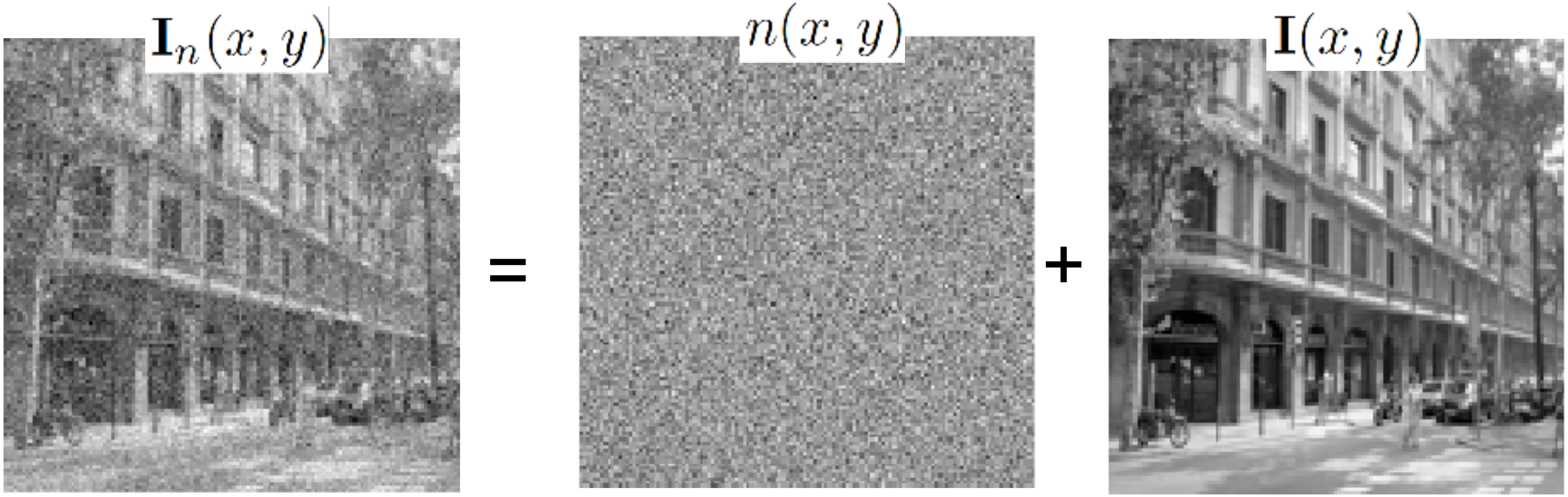
$$\widetilde{\mathbf{I}}(v) = \frac{A/|v|^{2\alpha}}{A/|v|^{2\alpha} + \sigma_n^2}\widetilde{\mathbf{I}}_n(v)$$



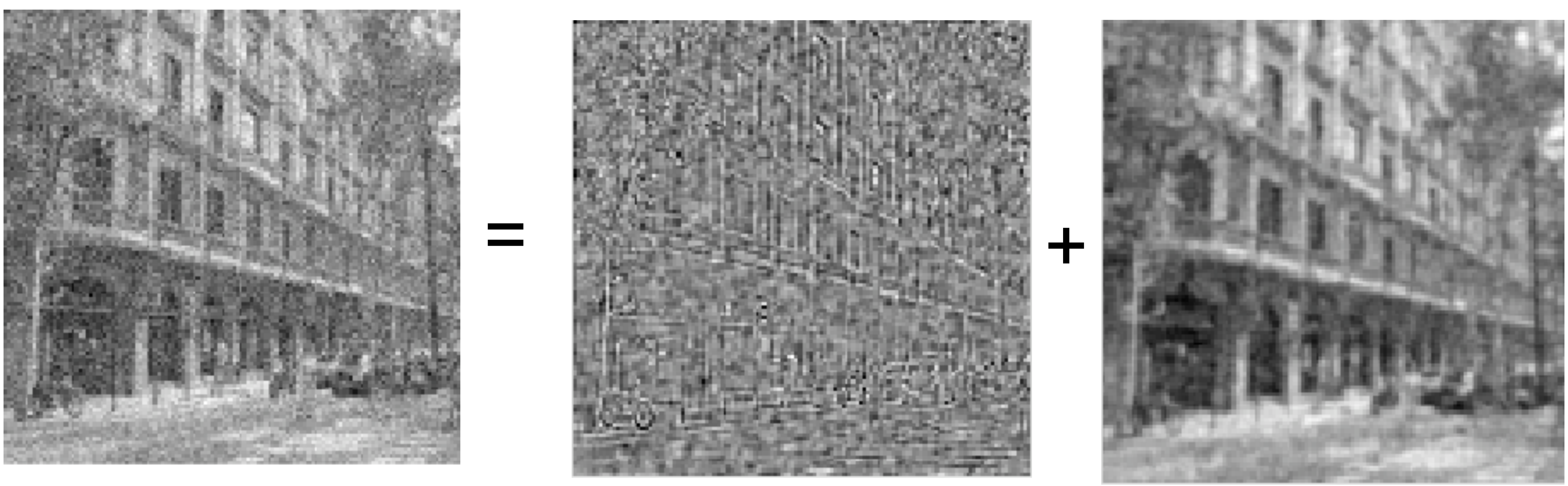
$$\frac{A/|v|^{2\alpha}}{A/|v|^{2\alpha} + \sigma_n^2}$$



The true decomposition:



The estimated decomposition:



And we got all this from just modeling the correlation between pairs of pixels!

Statistical Image Models

- Gaussian image model
  - image synthesis
  - Wiener filter denoising
- Kurtotic wavelets model
  - image synthesis
  - Bayesian denoising
- Non-parametric MRF model
  - image synthesis (Efros and Leung texture model)
  - Non-local means denoising

Extensions:

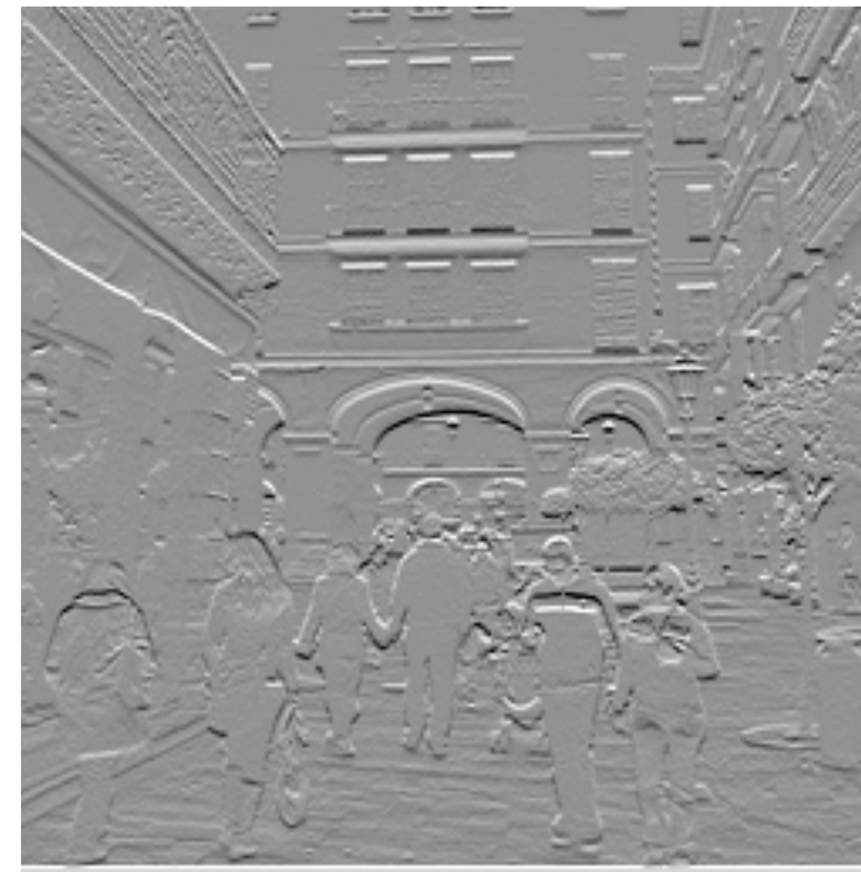
(1) Instead of using basis functions with global support (Fourier basis functions), we'll use localized filters.

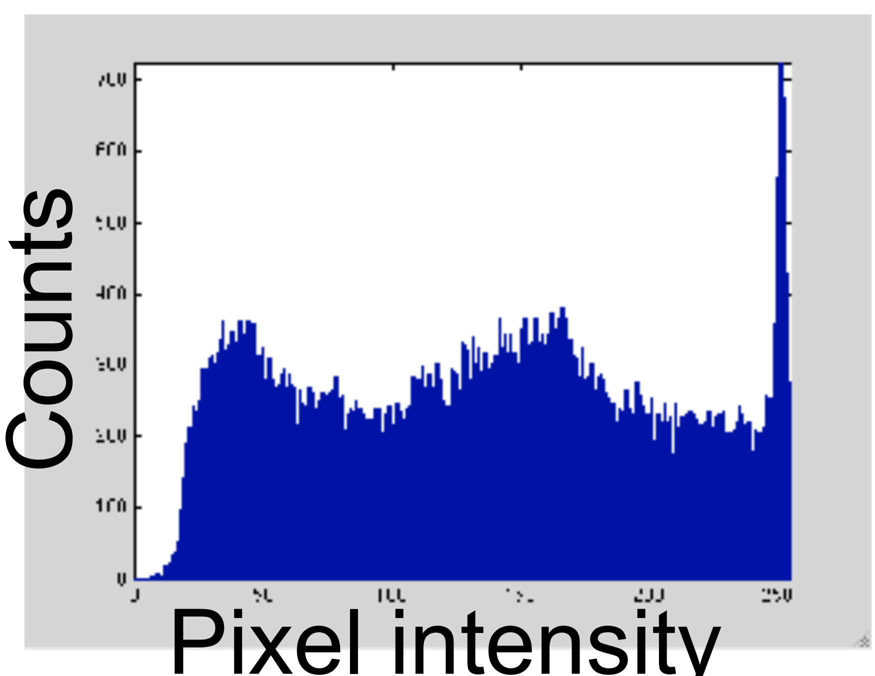
(2) Instead of looking at average power values of the basis functions, we'll examine the full marginal distribution (the histogram of their responses), generalizing to non-Gaussian distributions.

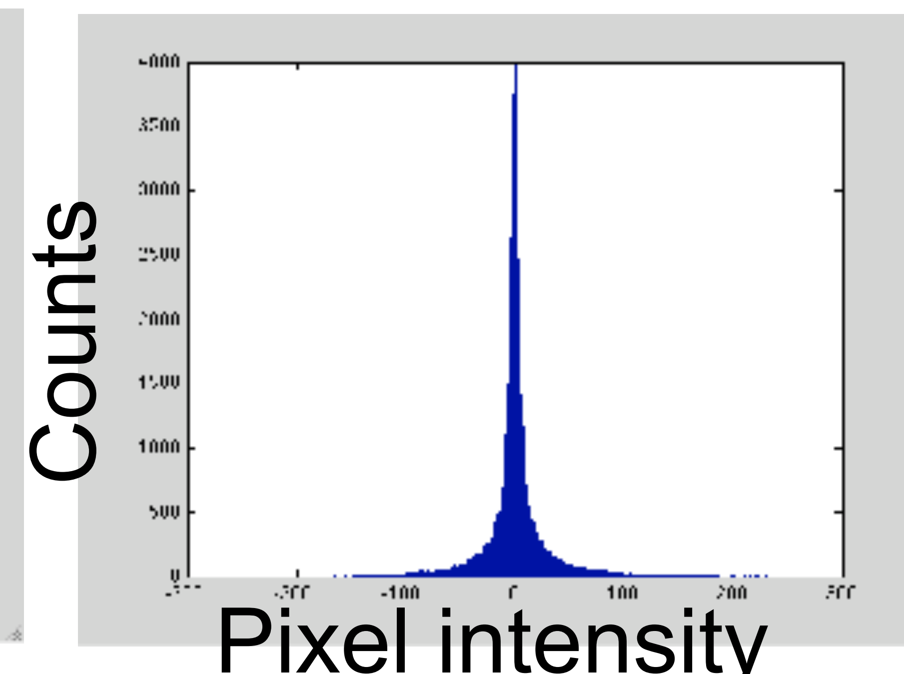
Observation: Sparse filter response

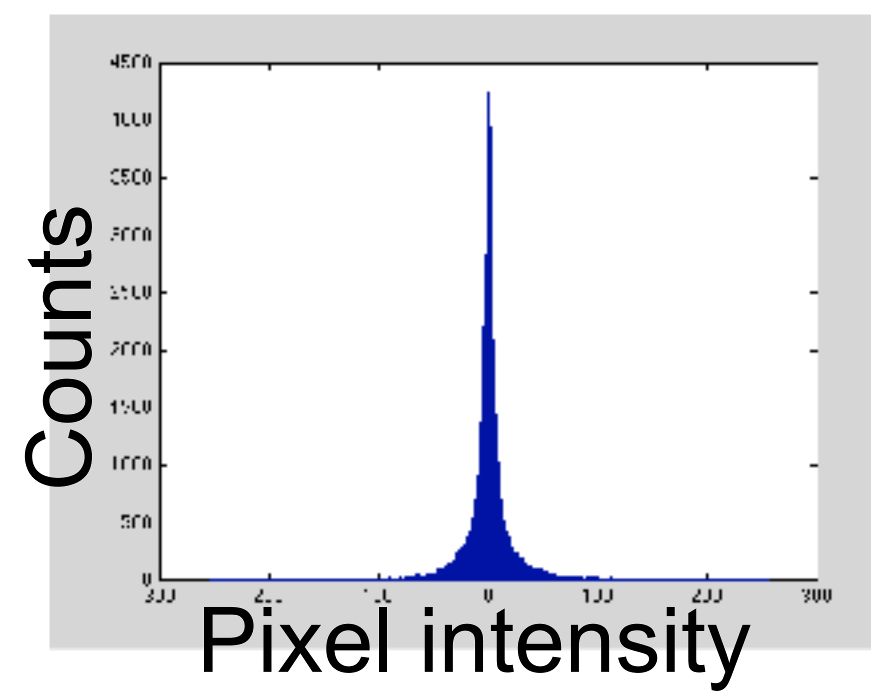
Before: regularity we exploited was the image power spectrum

Here: regularities we exploit are the histograms of bandpass filter outputs









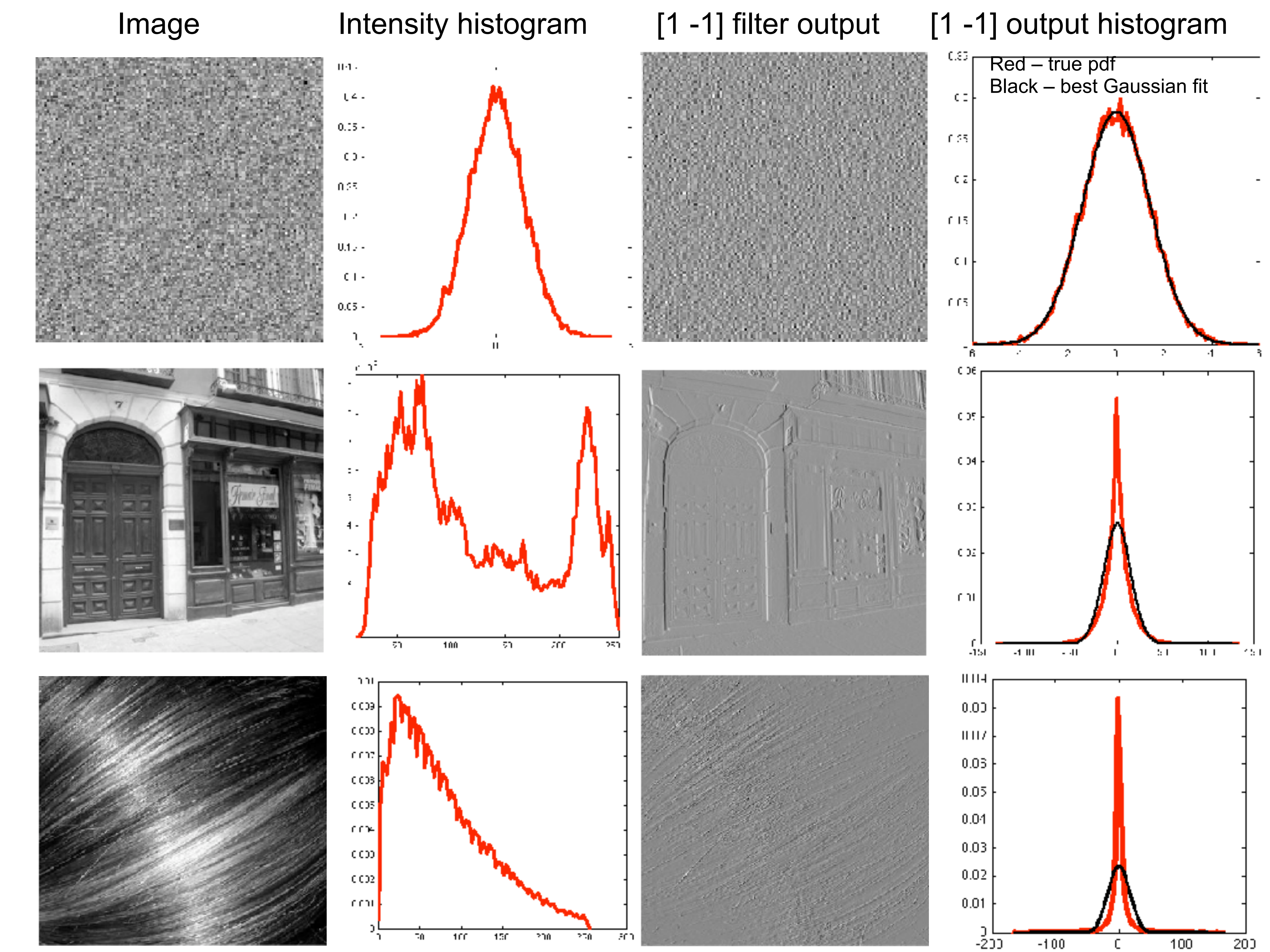
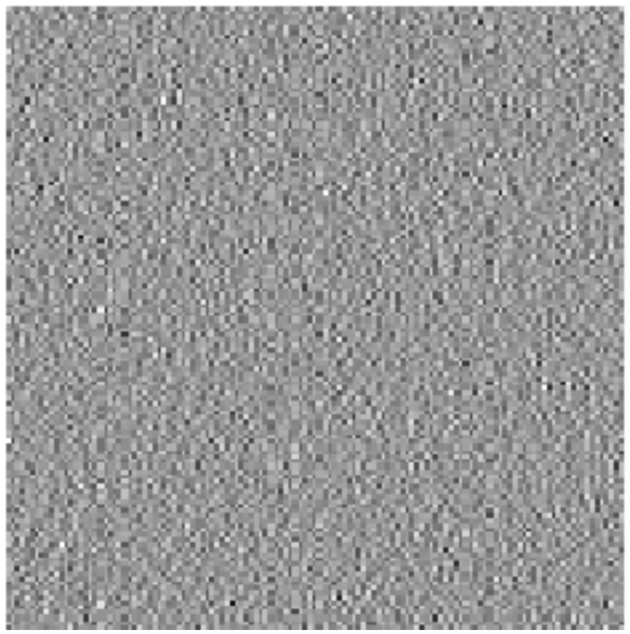
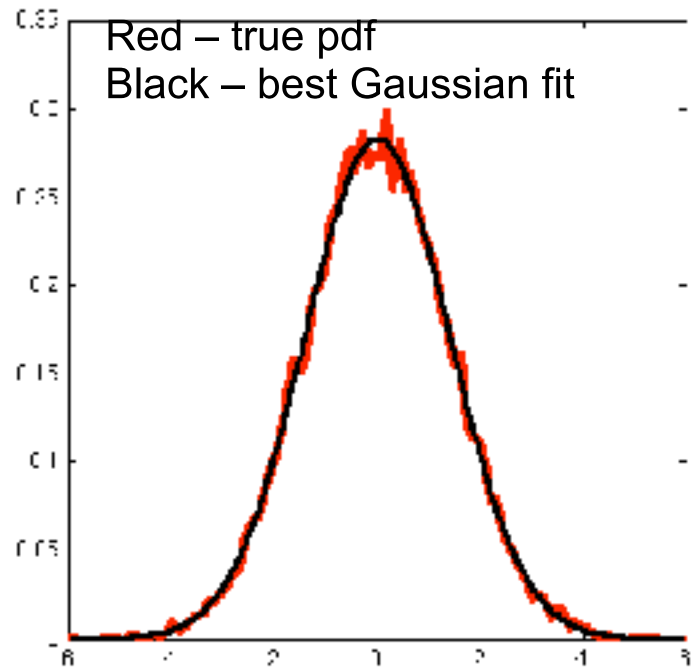


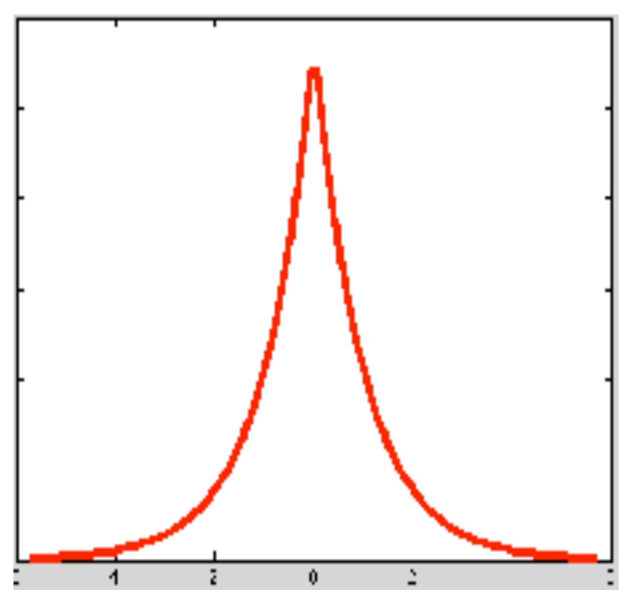
Image
Intensity histogram
[1 -1] filter output
[1 -1] output histogram
Red – true pdf
Black – best Gaussian fit
-200
-100
0
100
200

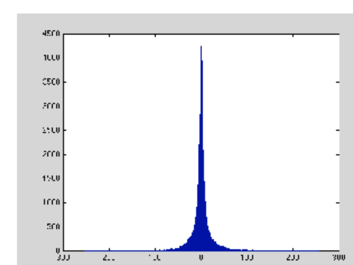
A model for the distribution of filter outputs





$$p(x) = \frac{\exp(-x^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}}$$





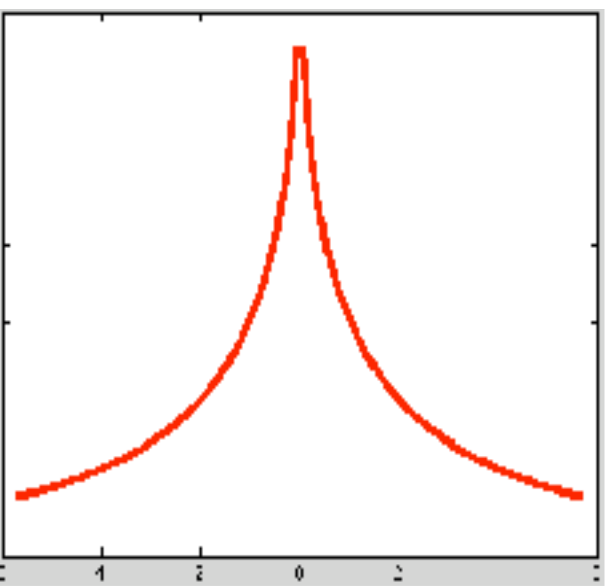
$$p(x) = \frac{\exp(-|x/s|^r)}{2s/r\Gamma(1/r)}$$

$r \sim 0.8 \;\; (< 2)$

Generalized Gaussian

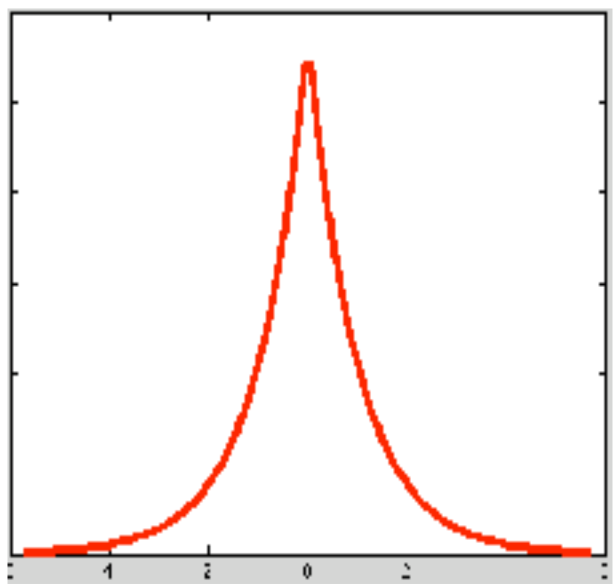
$$p(x) = \frac{\exp(-|x/s|^r)}{2s/r\Gamma(1/r)}$$

r = 0.5



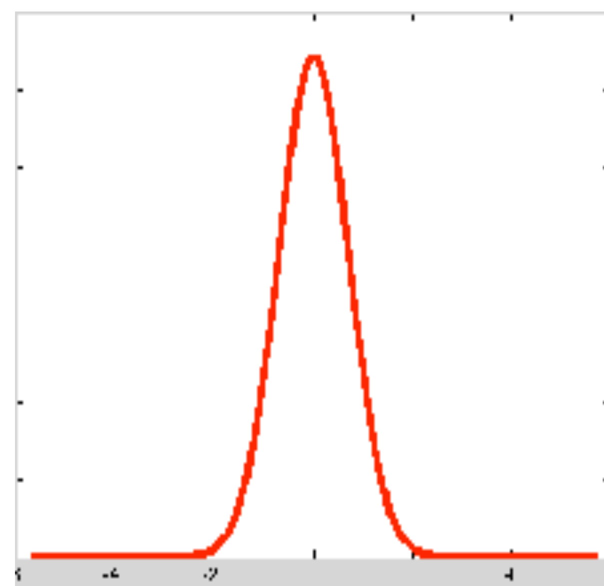
r = 1

Laplacian distribution

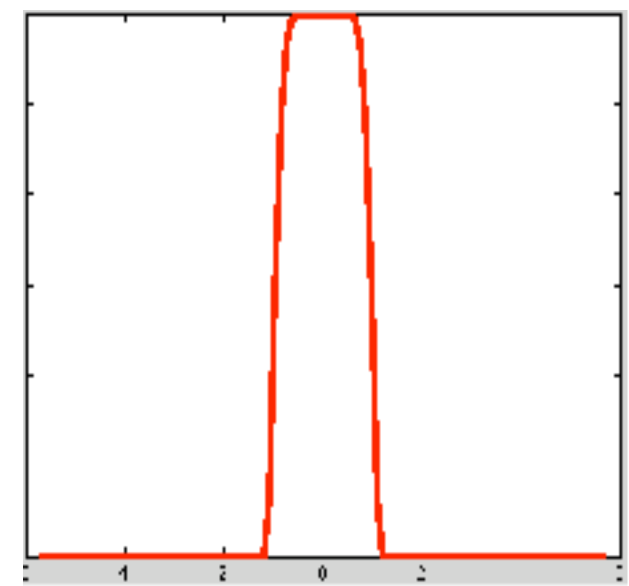


r = 2

Gaussian distribution



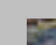
r = 10



Uniform distribution
r -> infinite

The wavelet marginal model

A small neighborhood

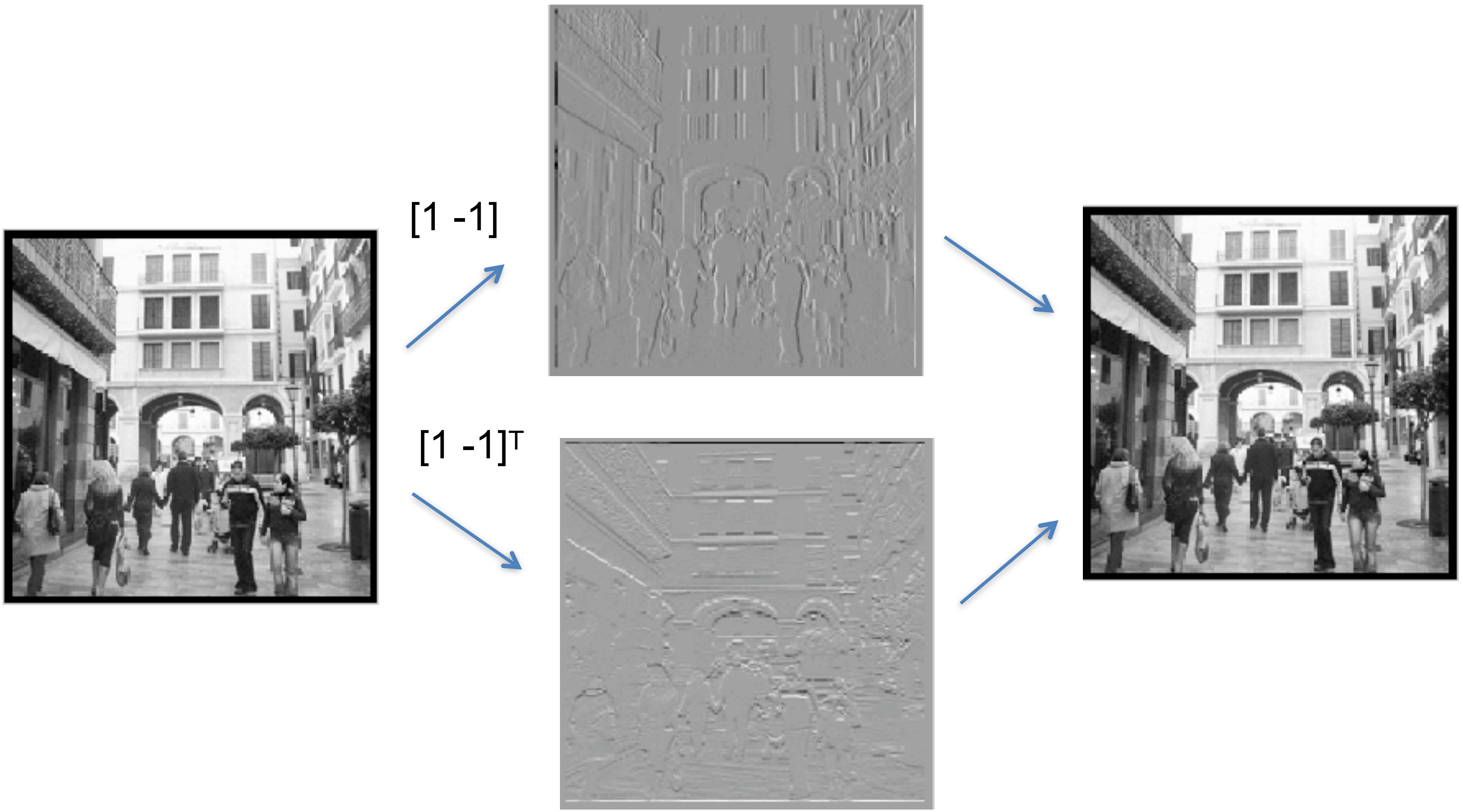


$$p(\mathbf{I}) = \prod_{k} \prod_{x,y} p(h_k(x,y))$$

All pixels and all outputs are independent

Filter outputs

The wavelet marginal model



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

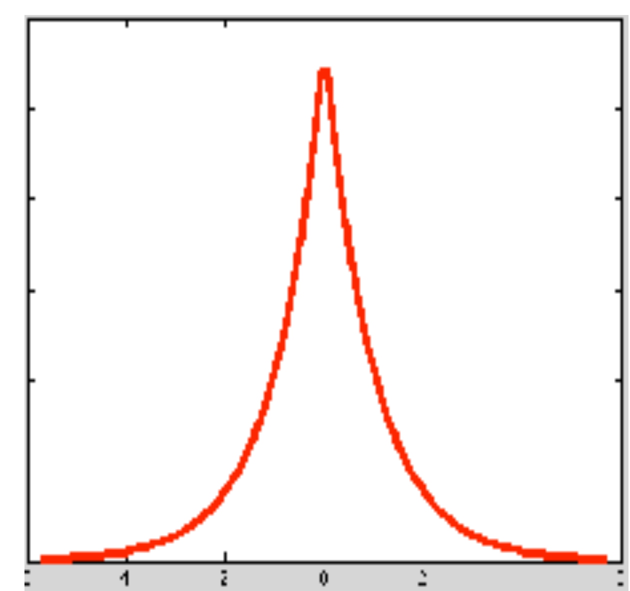
What is the most probable image under the wavelet marginal model?

[1 -1]

$[1\ -1]^T$

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

$$p(x) = \frac{\exp(-|x/s|^r)}{2s/r\Gamma(1/r)}$$



Sampling images

Gaussian model

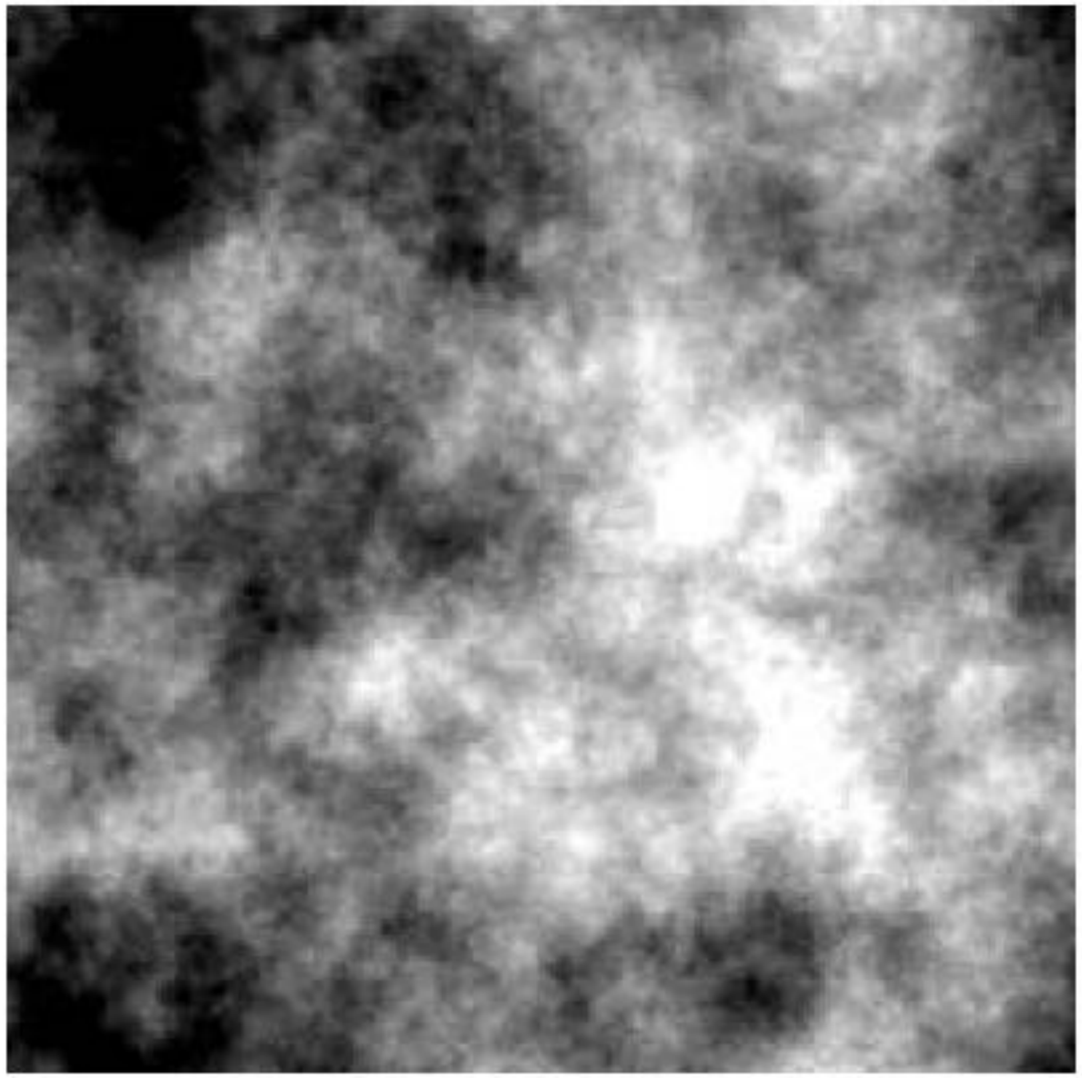


Fig. 3. Example image randomly drawn from the Gaussian spectral model, with $\gamma = 2.0$.

Wavelet marginal model

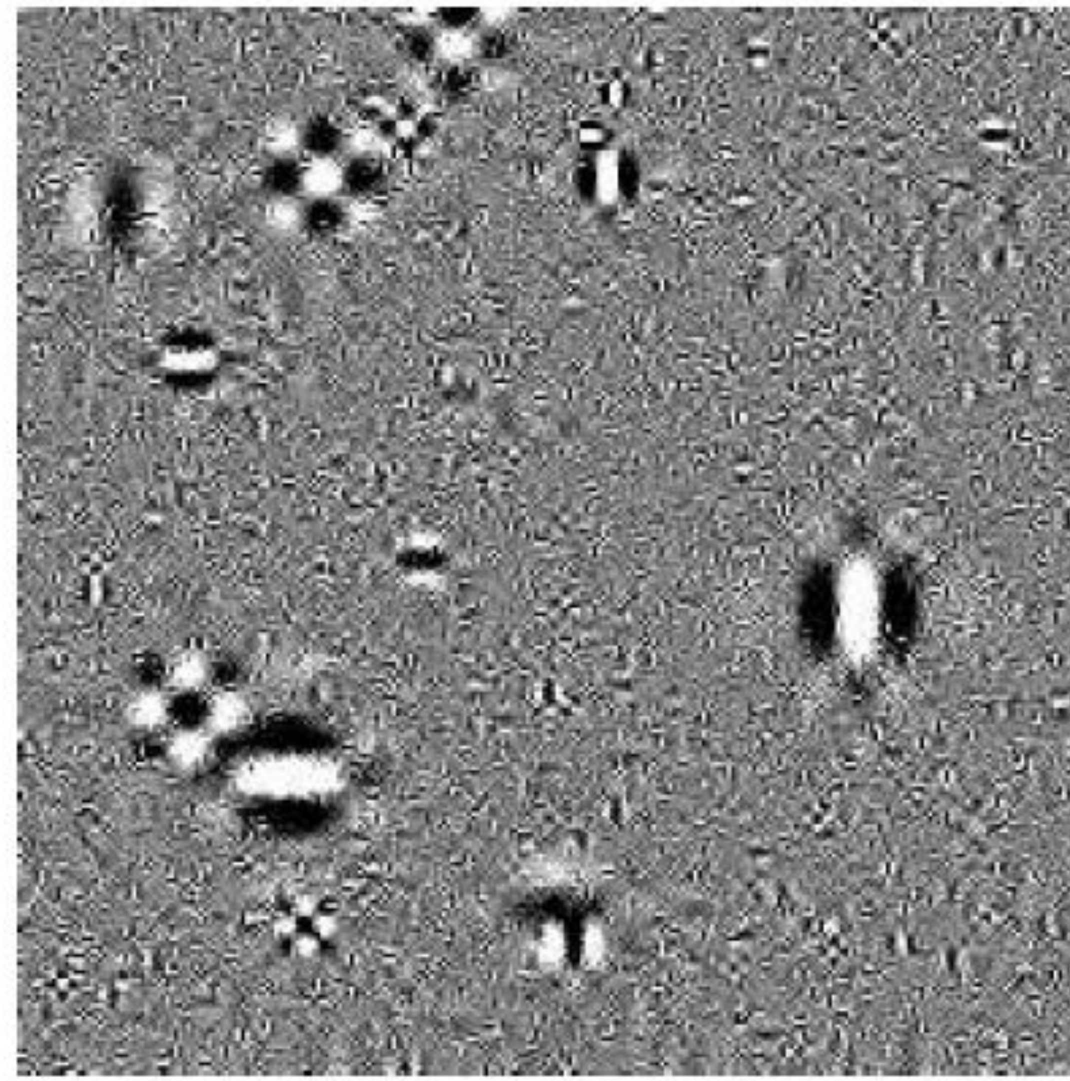


Fig. 6. A sample image drawn from the wavelet marginal model, with subband density parameters chosen to fit the image of Fig. 7.

Reminder from lecture 5:

Pyramid-Based Texture Analysis/Synthesis

David J. Heeger*
Stanford University

James R. Bergen†
SRI David Sarnoff Research Center

Abstract

This paper describes a method for synthesizing images that match the texture appearance of a given digitized sample. This synthesis is completely automatic and requires only the "target" texture as input. It allows generation of as much texture as desired so that any object can be covered. It can be used to produce solid textures for creating textured 3-d objects without the distortions inherent in texture mapping. It can also be used to synthesize texture mixtures, images that look a bit like each of several digitized samples. The approach is based on a model of human texture perception, and has potential to be a practically useful tool for graphics applications.

1 Introduction

Computer renderings of objects with surface texture are more interesting and realistic than those without texture. Texture mapping [15] is a technique for adding the appearance of surface detail by wrapping or projecting a digitized texture image onto a surface. Digitized textures can be obtained from a variety of sources, e.g., cropped from a photoCD image, but the resulting texture chip may not have the desired size or shape. To cover a large object you may need to repeat the texture; this can lead to unacceptable artifacts either in the form of visible seams, visible repetition, or both.

Texture mapping suffers from an additional fundamental problem: often there is no natural map from the (planar) texture image to the geometry/topology of the surface, so the texture may be distorted unnaturally when mapped. There are some partial solutions to this distortion problem [15] but there is no universal solution for mapping an image onto an arbitrarily shaped surface.

An alternative to texture mapping is to create (paint) textures by hand directly onto the 3-d surface model [14], but this process is both very labor intensive and requires considerable artistic skill.

Another alternative is to use computer-synthesized textures so that as much texture can be generated as needed. Furthermore, some of the synthesis techniques produce textures that tile seamlessly.

Using synthetic textures, the distortion problem has been solved in two different ways. First, some techniques work by synthesizing texture directly on the object surface (e.g., [31]). The second solution is to use solid textures [19, 23, 24]. A solid texture is a 3-d array of color values. A point on the surface of an object is colored by the value of the solid texture at the corresponding 3-d point. Solid texturing can be a very natural solution to the distortion problem: there is no distortion because there is no mapping. However, existing techniques for synthesizing solid textures can be quite cumbersome. One must learn how to tweak the parameters or procedures of the texture synthesizer to get a desired effect.

This paper presents a technique for synthesizing an image (or solid texture) that matches the appearance of a given texture sample. The key advantage of this technique is that it works entirely from the example texture, requiring no additional information or adjustment. The technique starts with a digitized image and analyzes it to compute a number of texture parameter values. Those parameter values are then used to synthesize a new image (of any size) that looks (in its color and texture properties) like the original. The analysis phase is inherently two dimensional since the input digitized images are 2-d. The synthesis phase, however, may be either two- or three-dimensional. For the 3-d case, the output is a solid texture such that planar slices through the solid look like the original scanned image. In either case, the (2-d or 3-d) texture is synthesized so that it tiles seamlessly.

2 Texture Models

Textures have often been classified into two categories, deterministic textures and stochastic textures. A deterministic texture is characterized by a set of primitives and a placement rule (e.g., a tile floor). A stochastic texture, on the other hand, does not have easily identifiable primitives (e.g., granite, bark, sand). Many real-world textures have some mixture of these two characteristics (e.g. woven fabric, woodgrain, plowed fields).

Much of the previous work on texture analysis and synthesis can be classified according to what type of texture model was used. Some of the successful texture models include reaction-diffusion [31, 34], frequency domain [17], fractal [9, 18], and statistical/random field [1, 6, 8, 10, 12, 13, 21, 26] models. Some (e.g., [10]) have used hybrid models that include a deterministic (or periodic) component and a stochastic component. In spite of all this work, scanned images and hand-drawn textures are still the principle source of texture maps in computer graphics.

This paper focuses on the synthesis of stochastic textures. Our approach is motivated by research on human texture perception. Current theories of texture discrimination are based on the fact that two textures are often difficult to discriminate when they produce a similar distribution of responses in a bank of (orientation and spatial-frequency selective) linear filters [2, 3, 7, 16, 20, 32]. The method described here, therefore, synthesizes textures by matching distributions (or histograms) of filter outputs. This approach depends on the principle (not entirely correct as we shall see) that all of the spatial information characterizing a texture image can be captured in the first order statistics of an appropriately chosen set of linear filter outputs. Nevertheless, this model (though incomplete) captures an interesting set of texture properties.

*Department of Psychology, Stanford University, Stanford, CA 94305. heeger@white.stanford.edu http://white.stanford.edu

†SRI David Sarnoff Research Center, Princeton, NJ 08544. jrb@sarnoff.com

Figure 5: (Top Row) Original digitized sample textures: red granite, berry bush, figured maple, yellow coral. (Bottom Rows) Synthetic solid textured teapots.

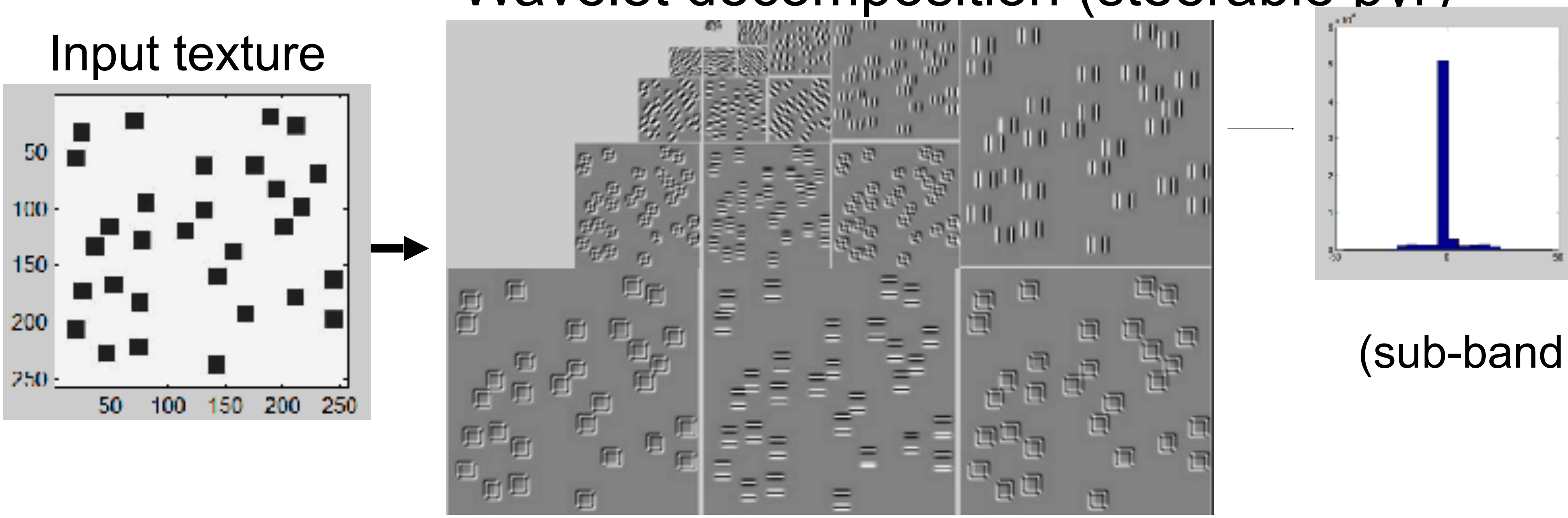
https://www.cns.nyu.edu/heegerlab/content/publications/Heeger-siggraph95.pdf

SIGGRAPH 1995

Texture analysis

Wavelet decomposition (steerable pyr)

Input texture



(sub-band histogram)

(Steerable pyr; Simoncelli & Freeman, '95)

Examples from the paper

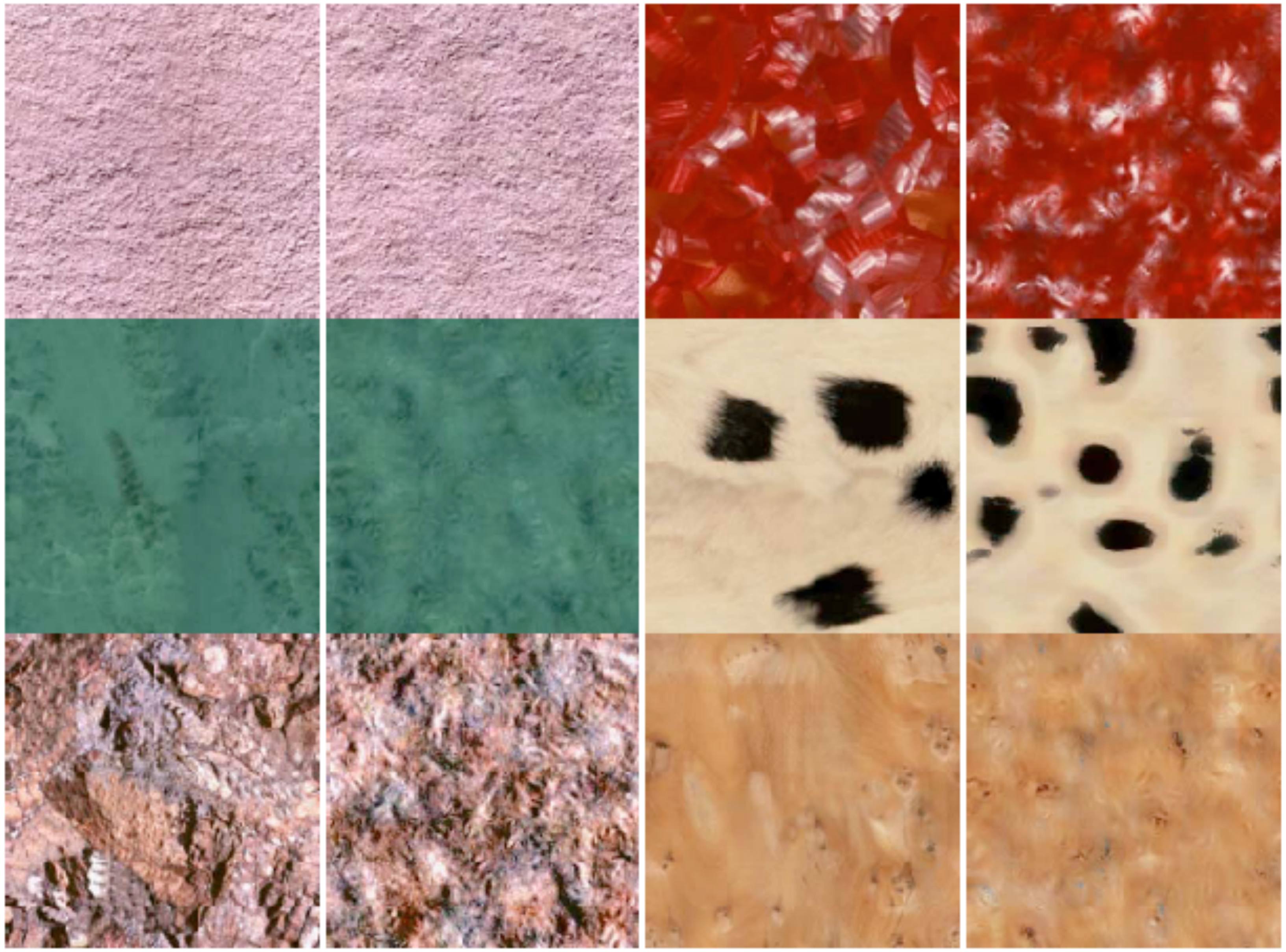
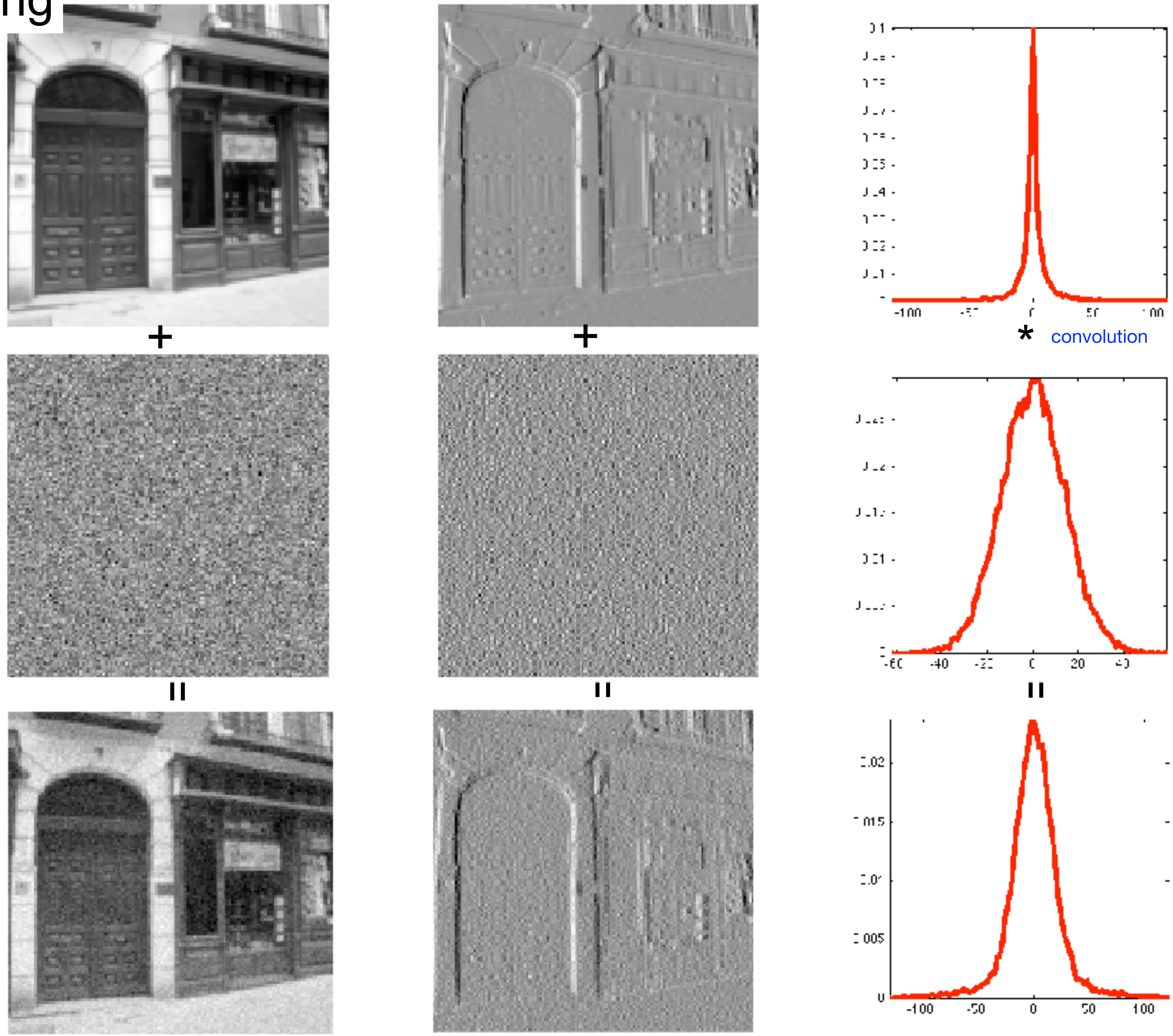


Figure 3: In each pair left image is original and right image is synthetic: stucco, iridescent ribbon, green marble, panda fur, slag stone, figured yew wood.

Heeger and Bergen, 1995

This texture synthesis work was the inspiration for the Gatys et al approach to force neural network co-occurence statistics to agree with those of a target style image in order to stylize an image.

Denoising



White Gaussian noise

Noisy image

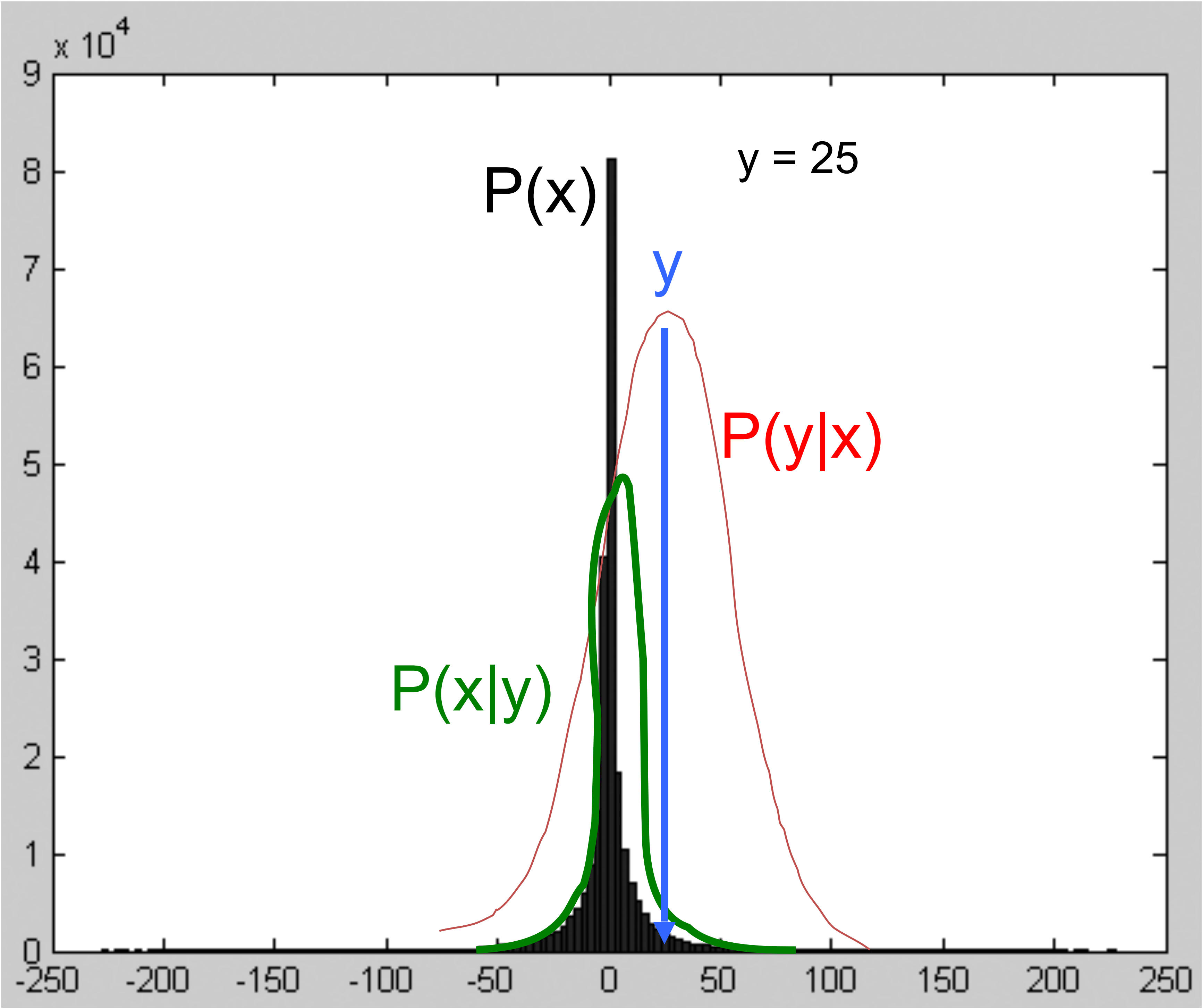
Denoising with the marginal wavelet model

Let y = noise-corrupted observation: y = x+n, with n ~ gaussian.

Let x = bandpassed image value before adding noise.

By Bayes theorem

P(x|y) ~ P(y|x) P(x)

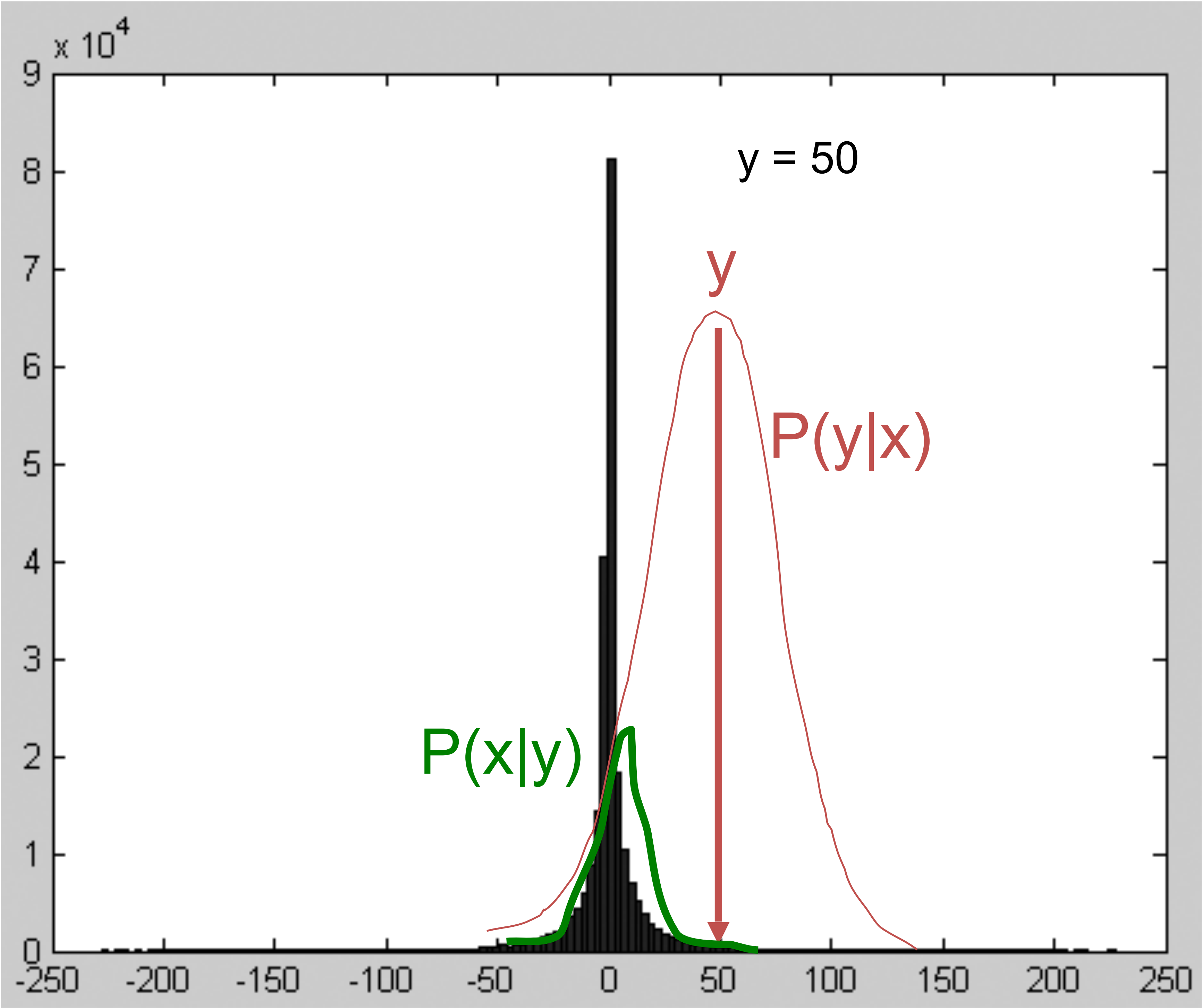


Denoising with the marginal wavelet model

Let x = bandpassed image value before adding noise.
Let y = noise-corrupted observation.

By Bayes theorem

P(x|y) ~ P(y|x) P(x)

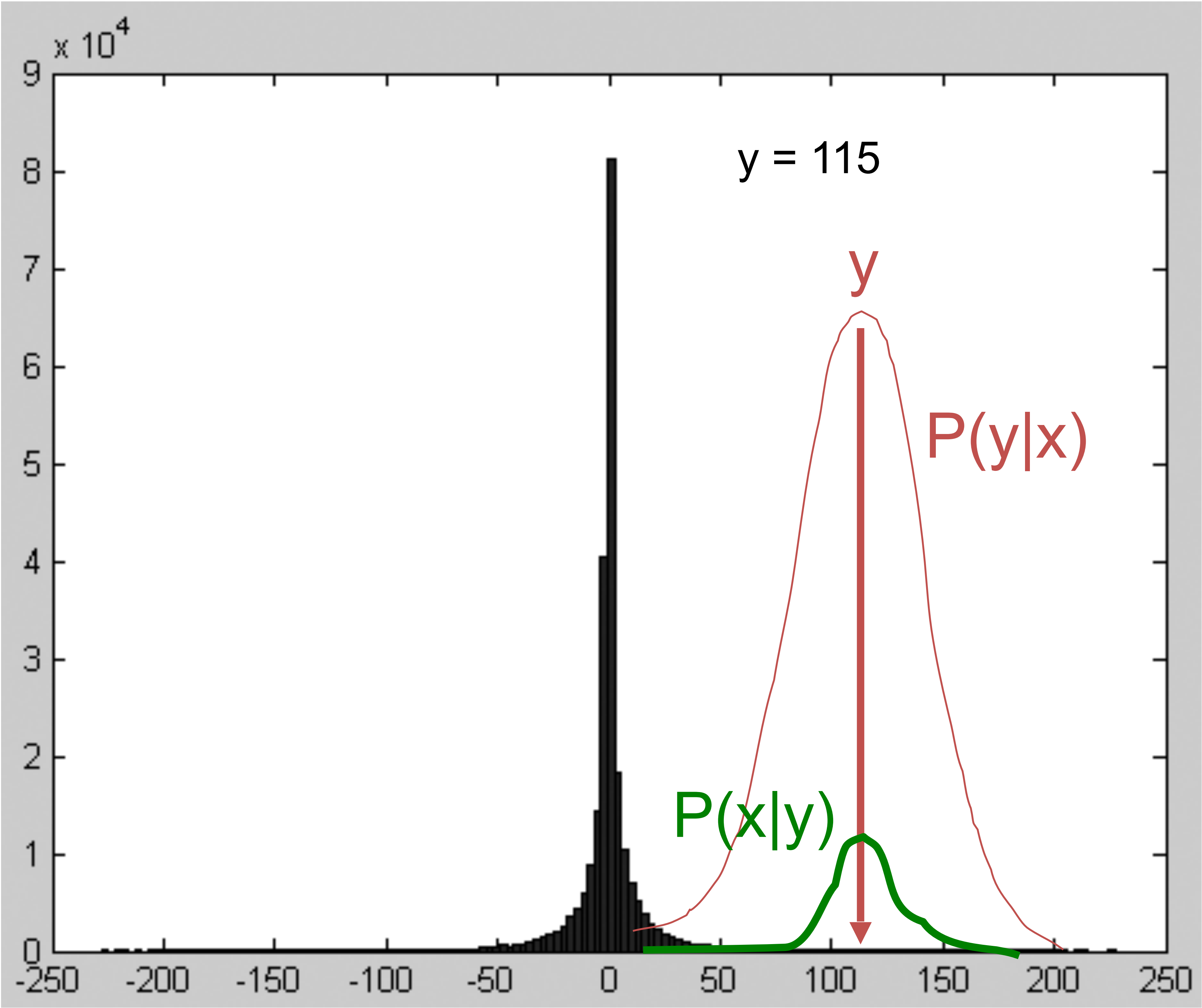


Denoising with the marginal wavelet model

Let x = bandpassed image value before adding noise.
Let y = noise-corrupted observation.

By Bayes theorem

P(x|y) ~ P(y|x) P(x)



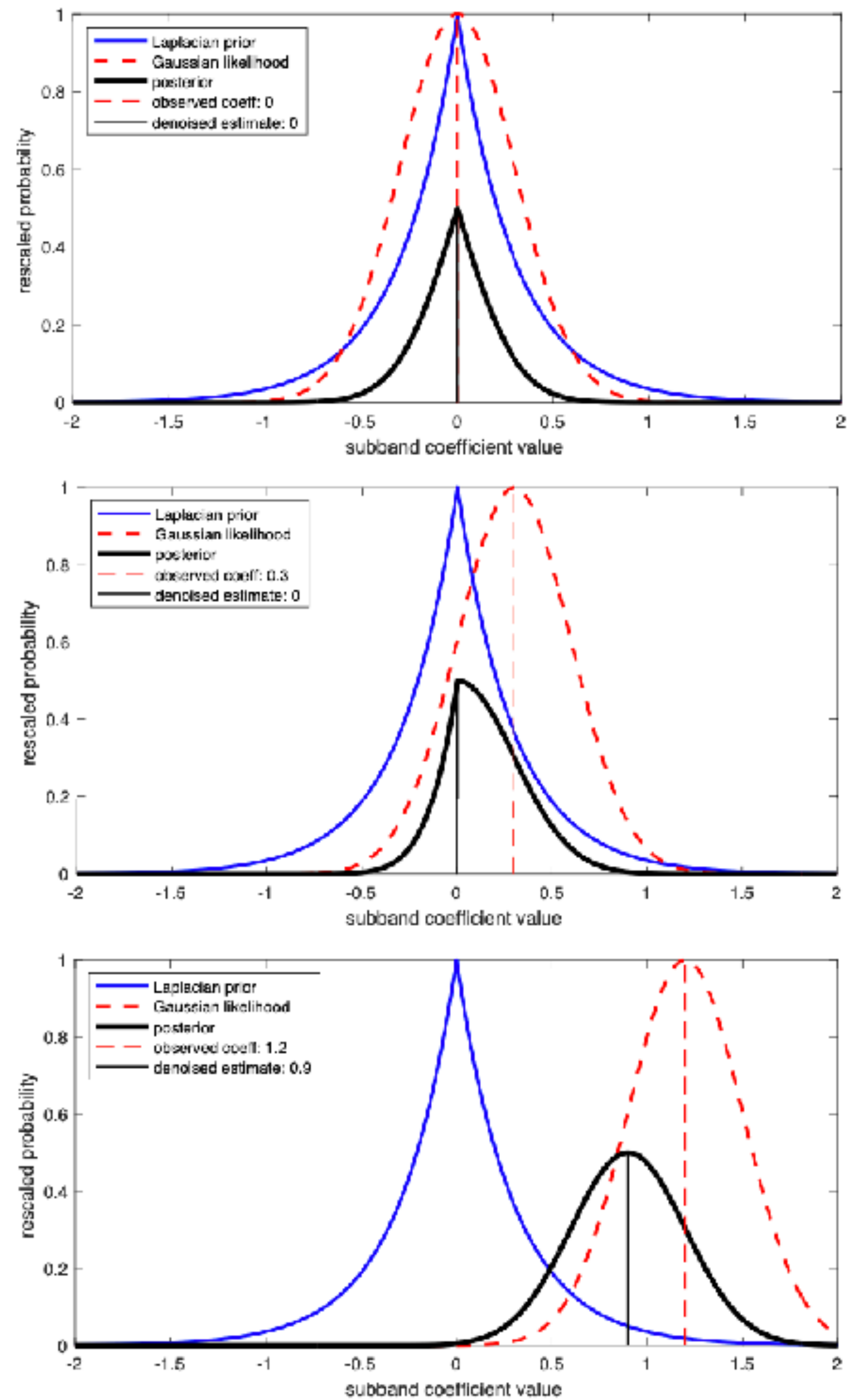


Figure 27.18: Showing the likelihood, prior, and posterior terms for the estimation of a subband coefficient from several noisy observations. The blue curve, showing the Laplacian prior probability for the subband values, is the same for all rows. Top: A zero subband coefficient is observed, resulting in the zero-mean Gaussian likelihood term (red), and the posterior (black) with a maximum at zero. Middle: An observation of 0.26 shifts the likelihood term, but the posterior still has a peak at zero. Bottom: an observed coefficient value of 1.22 yields a maximum posterior probability estimate of 0.9.

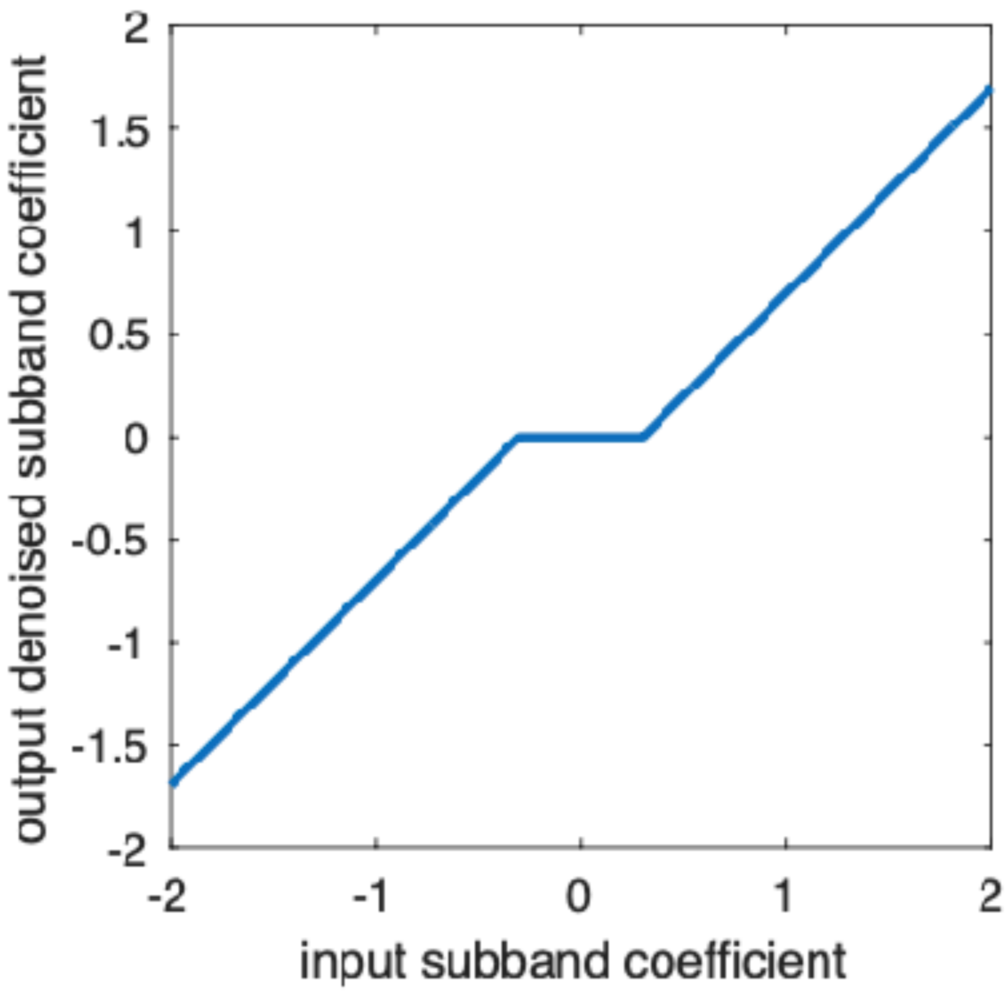
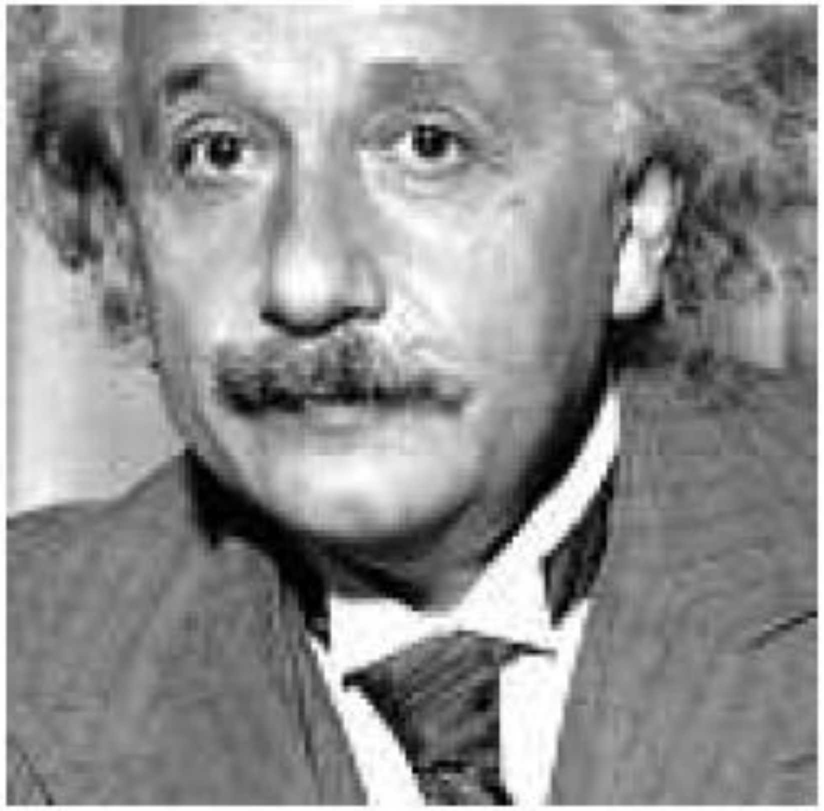
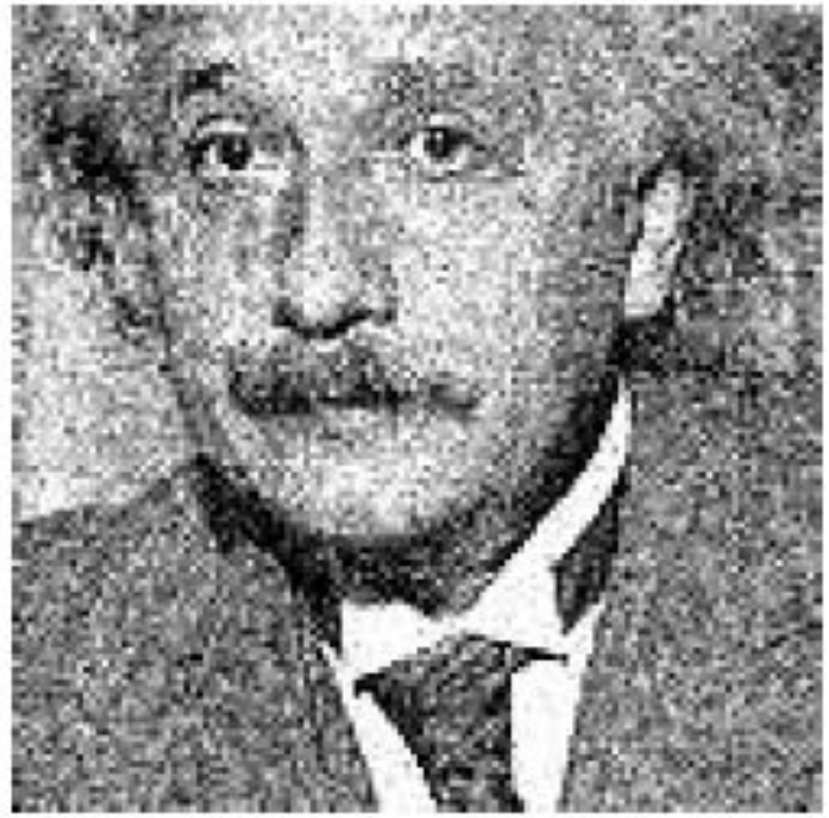


Figure 27.19: Input/output curve for maximum posterior denoising for the example of Fig. (27.18.

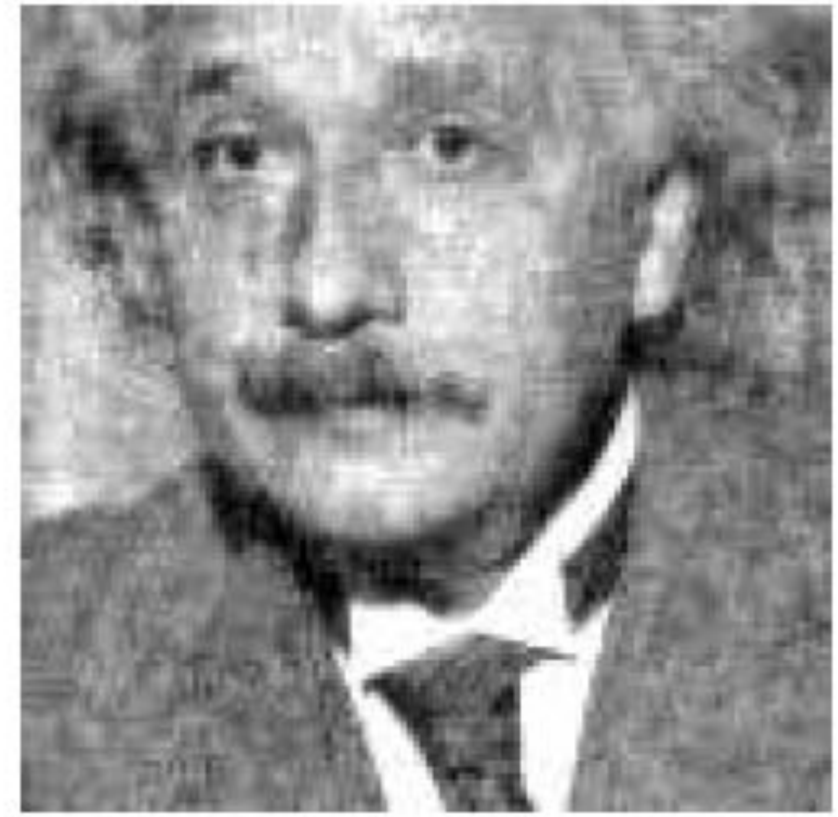
original

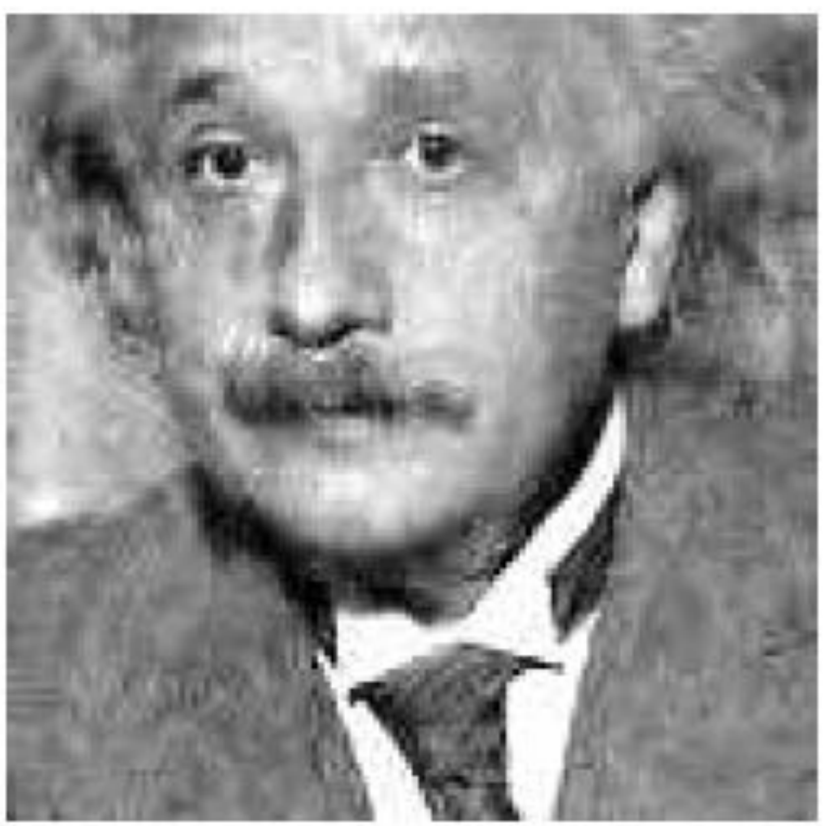


With Gaussian noise of std. dev. 21.4 added, giving PSNR=22.06 dB



(1) Denoised with Gaussian model, PSNR=27.87 dB





(2) Denoised with wavelet marginal model, PSNR=29.24 dB

http://www.cns.nyu.edu/pub/eero/simoncelli05a-preprint.pdf

Statistical Image Models

- Gaussian image model
  - image synthesis
  - Wiener filter denoising
- Kurtotic wavelets model
  - image synthesis
  - Bayesian denoising
- Non-parametric MRF model
  - image synthesis (Efros and Leung texture model)
  - Non-local means denoising

Instead of a sum of basis function samples taken independently, we'll generate the image bit by bit, modeling the image as a set of conditional sample draws.

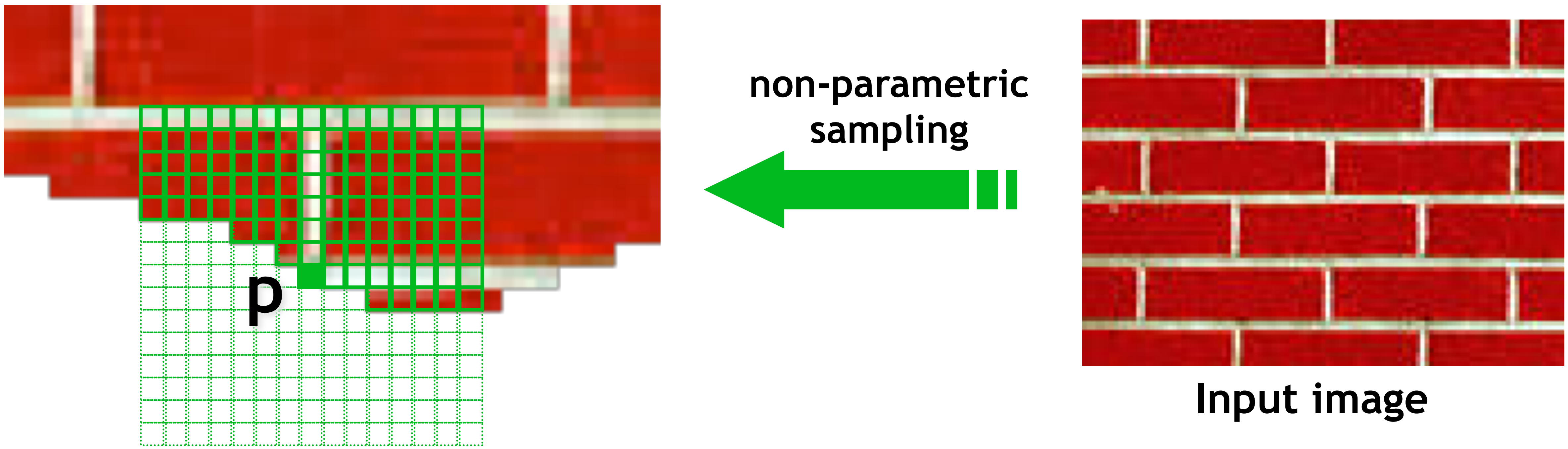
IEEE International Conference on Computer Vision, Corfu, Greece, September 1999

Texture Synthesis by Non-parametric Sampling

Alexei A. Efros and Thomas K. Leung
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776, U.S.A.
{efros,leungt}@cs.berkeley.edu

Image model: each image has a large set of "production rules"
If the local image values satisfy the conditions of one of the production rules, then you output a particular pixel value.

Efros & Leung Algorithm



Synthesizing a pixel

Input image

Assuming Markov property, compute P(**p**|N(**p**))

– Building explicit probability tables is 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

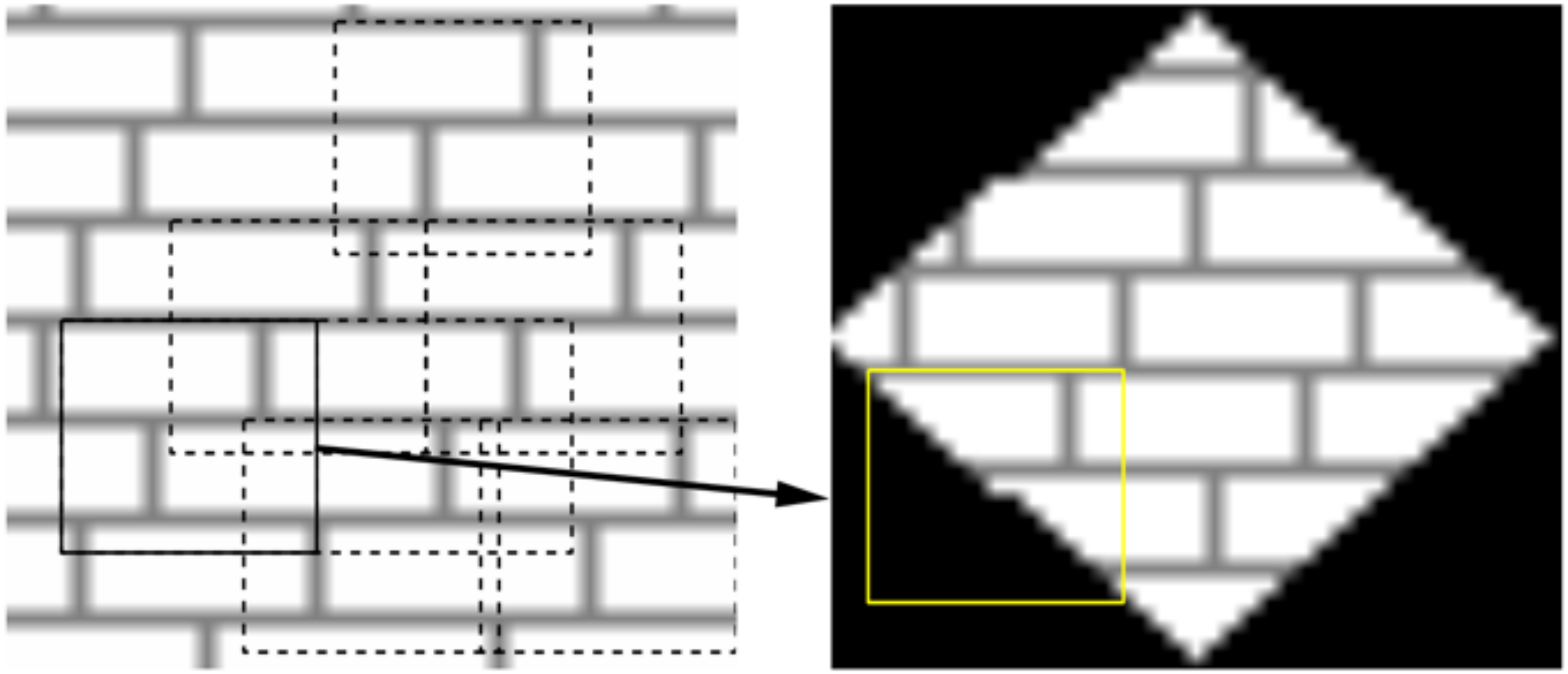
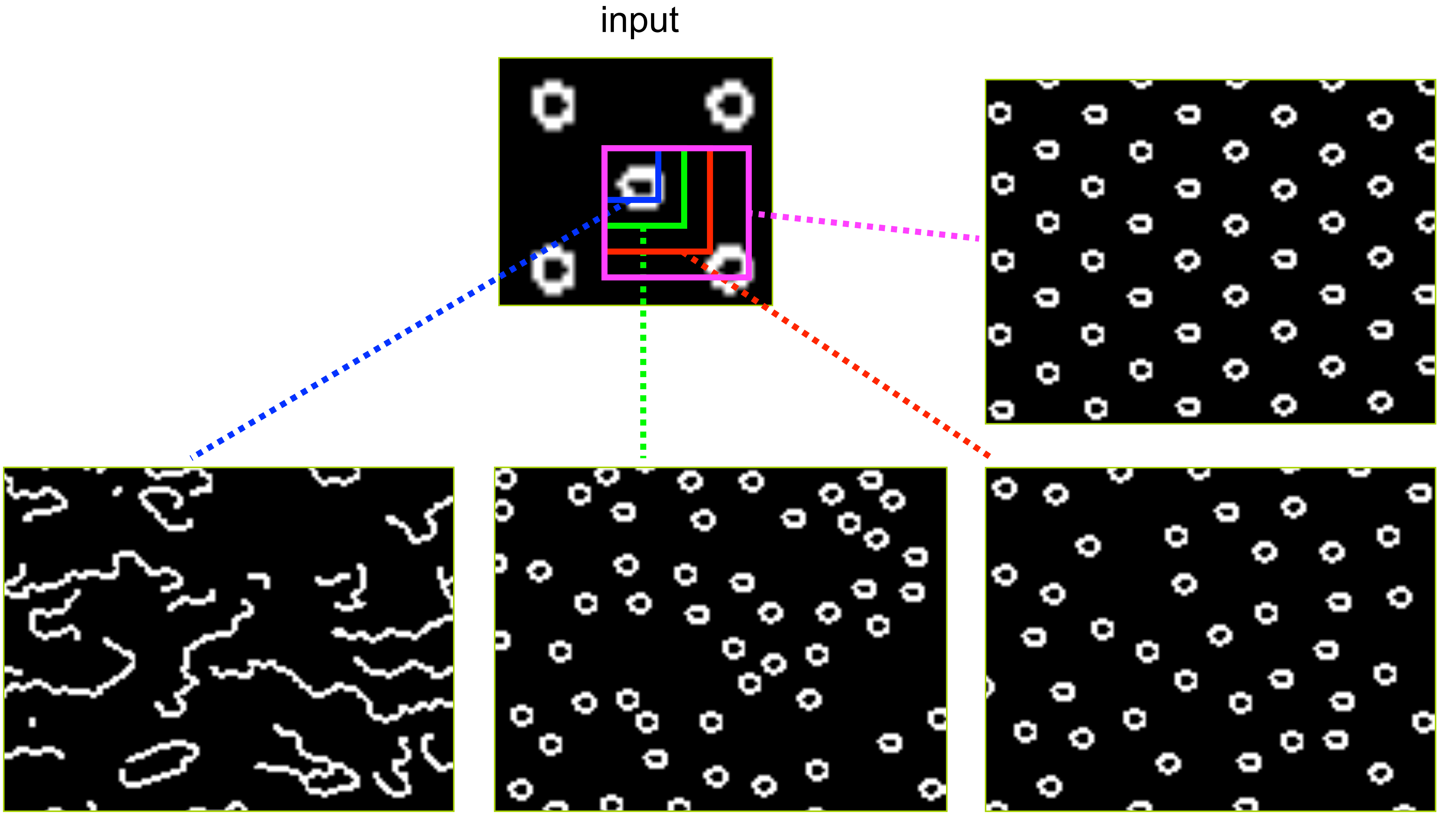
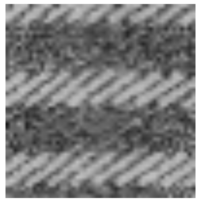


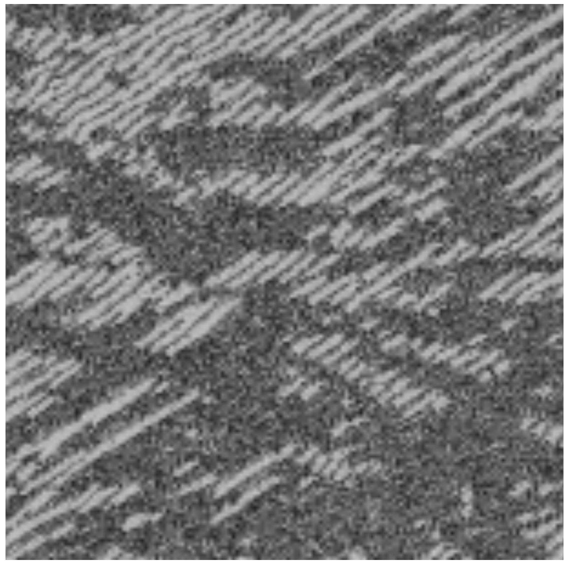
Figure 1. Algorithm Overview. Given a sample texture image (left), a new image is being synthesized one pixel at a time (right). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the left) that are similar to the pixel's neighborhood (box on the right) and then randomly chooses one neighborhood and takes its center to be the newly synthesized pixel.

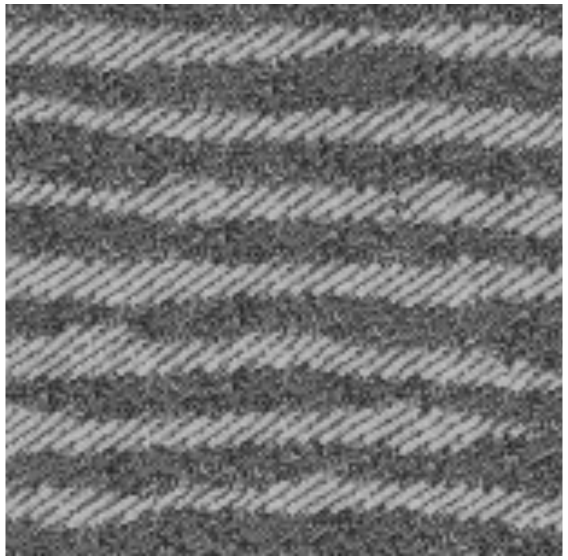
Effect of Neighborhood Window Size on Texture Generation

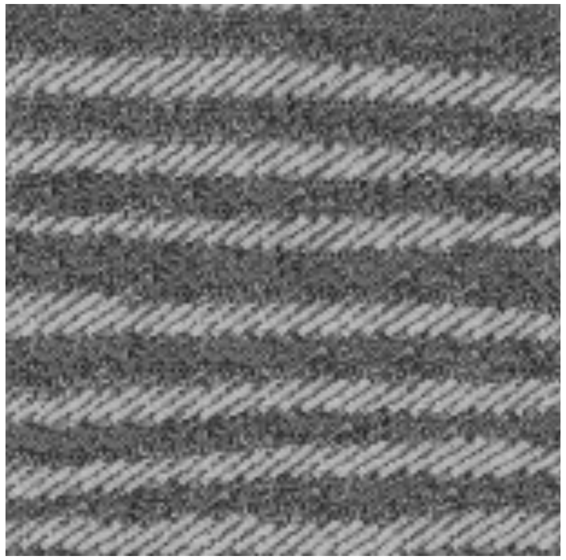


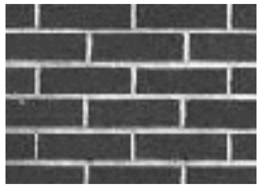
Varying Window Size

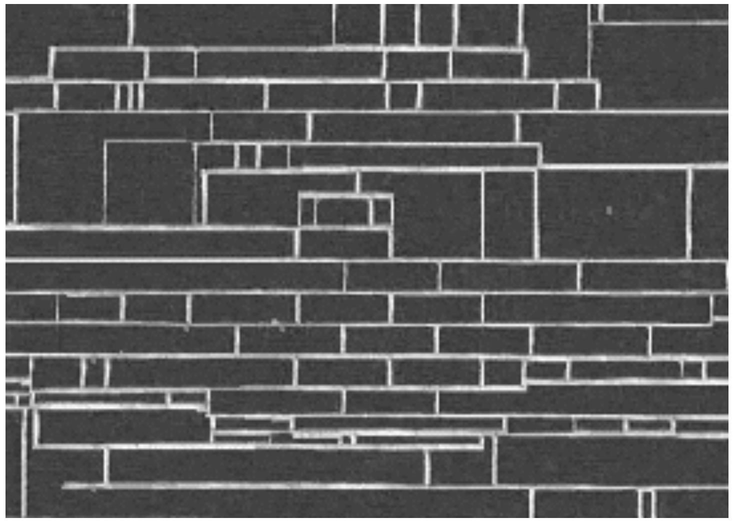


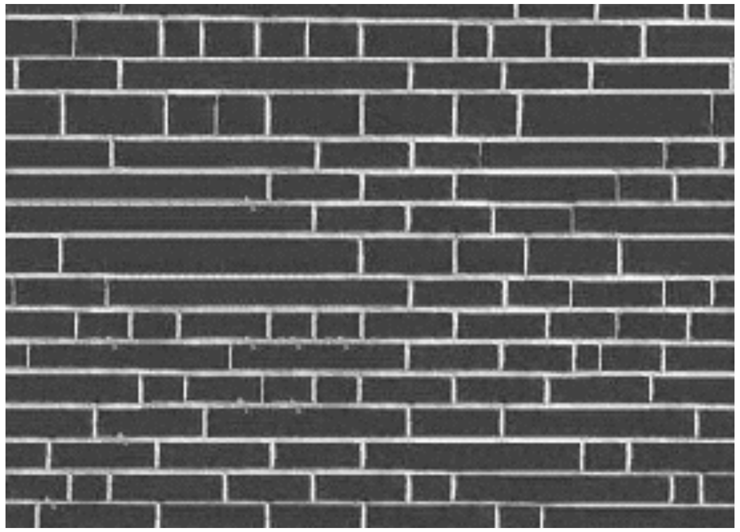


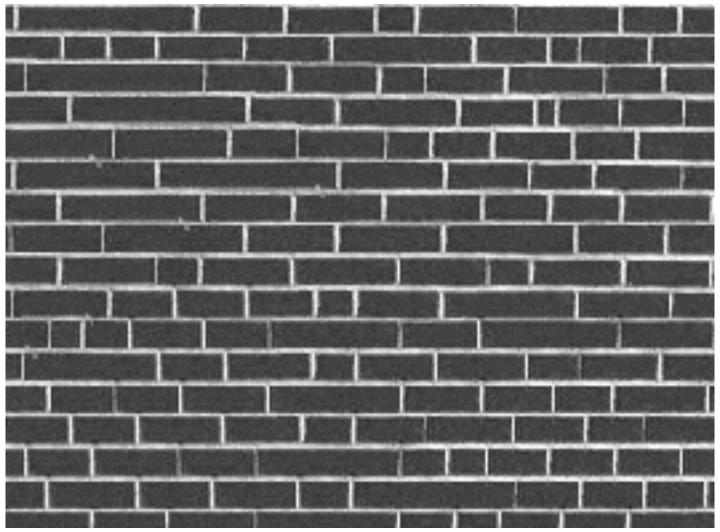








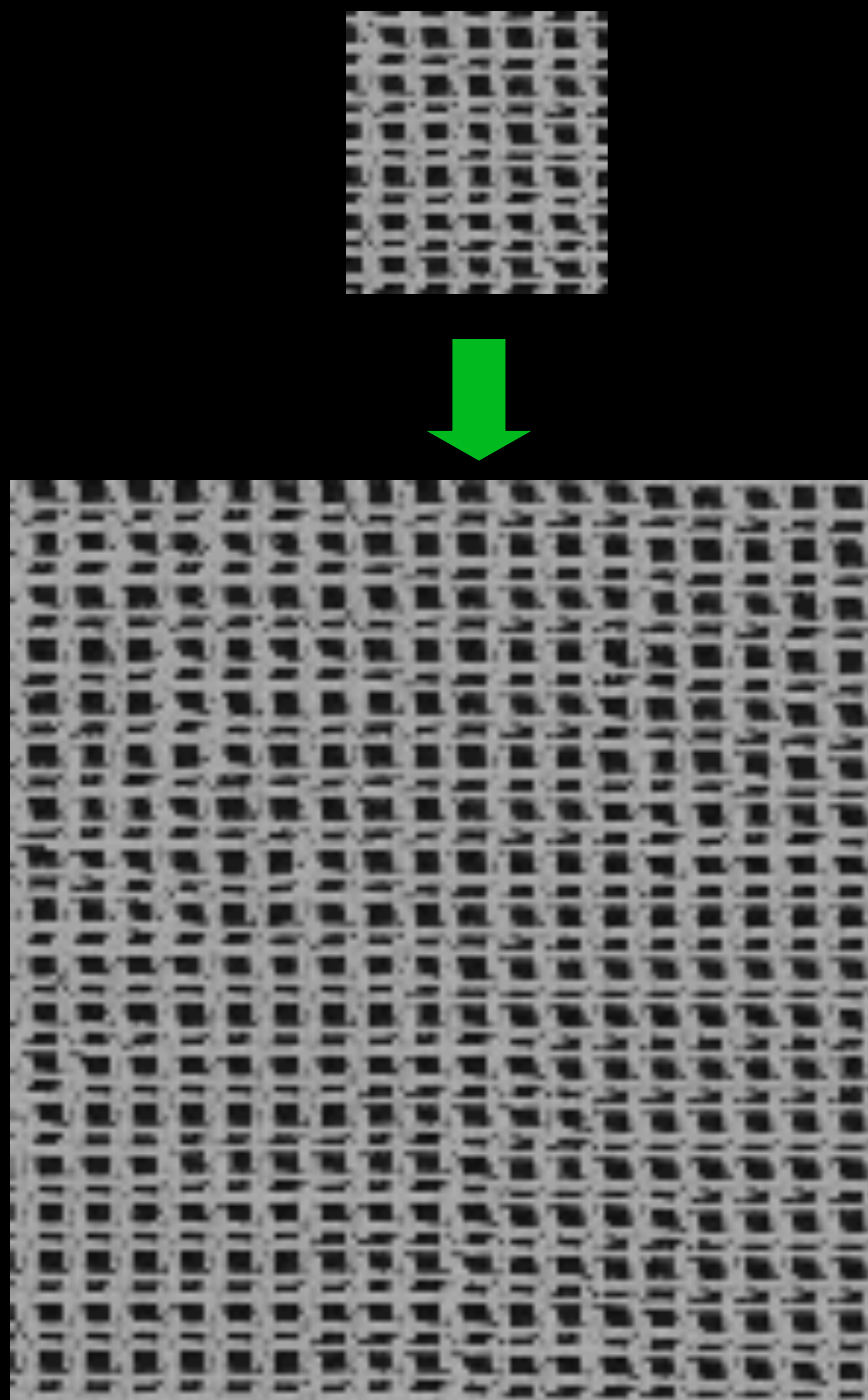




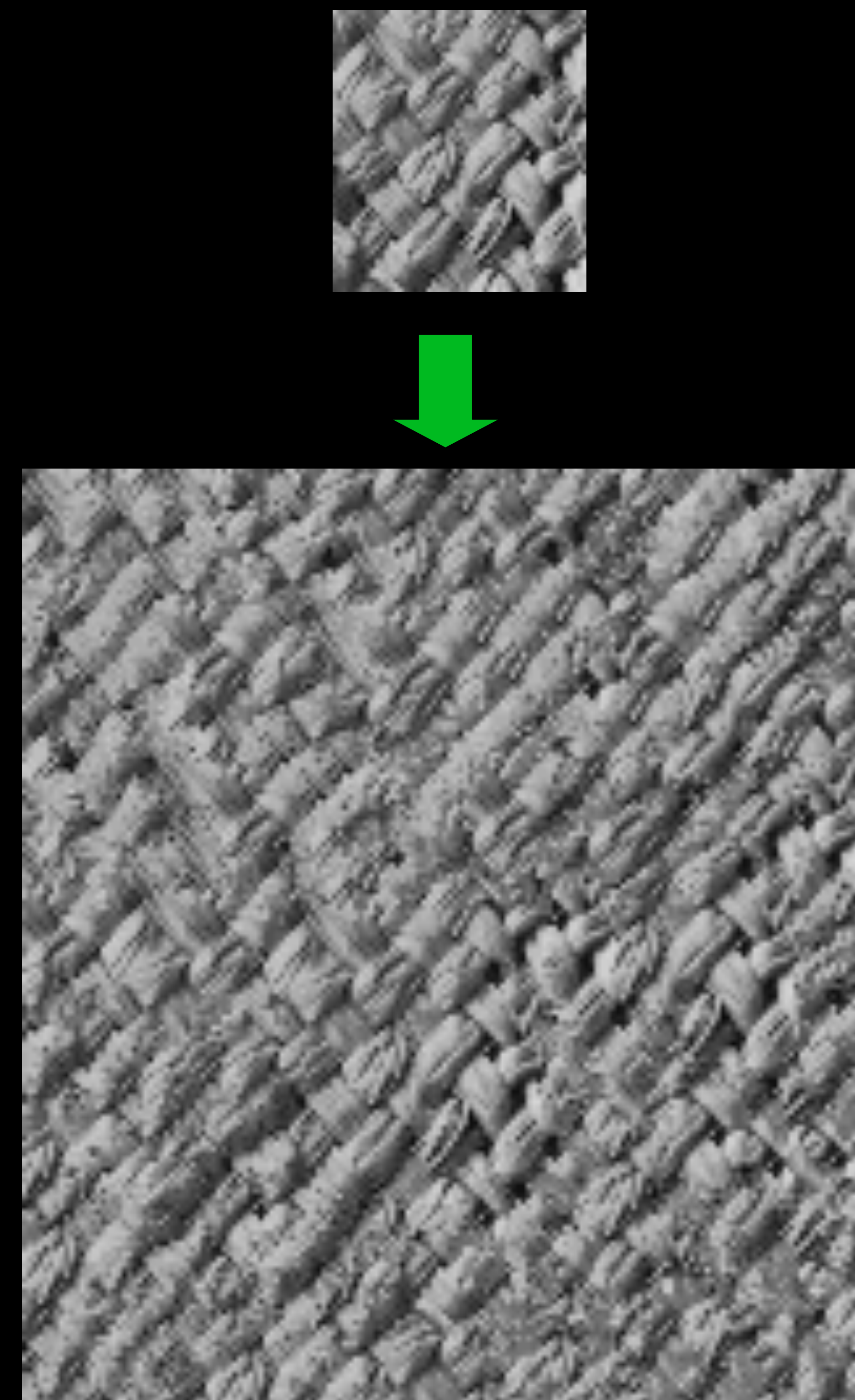
Increasing window size

Synthesis Results

french canvas

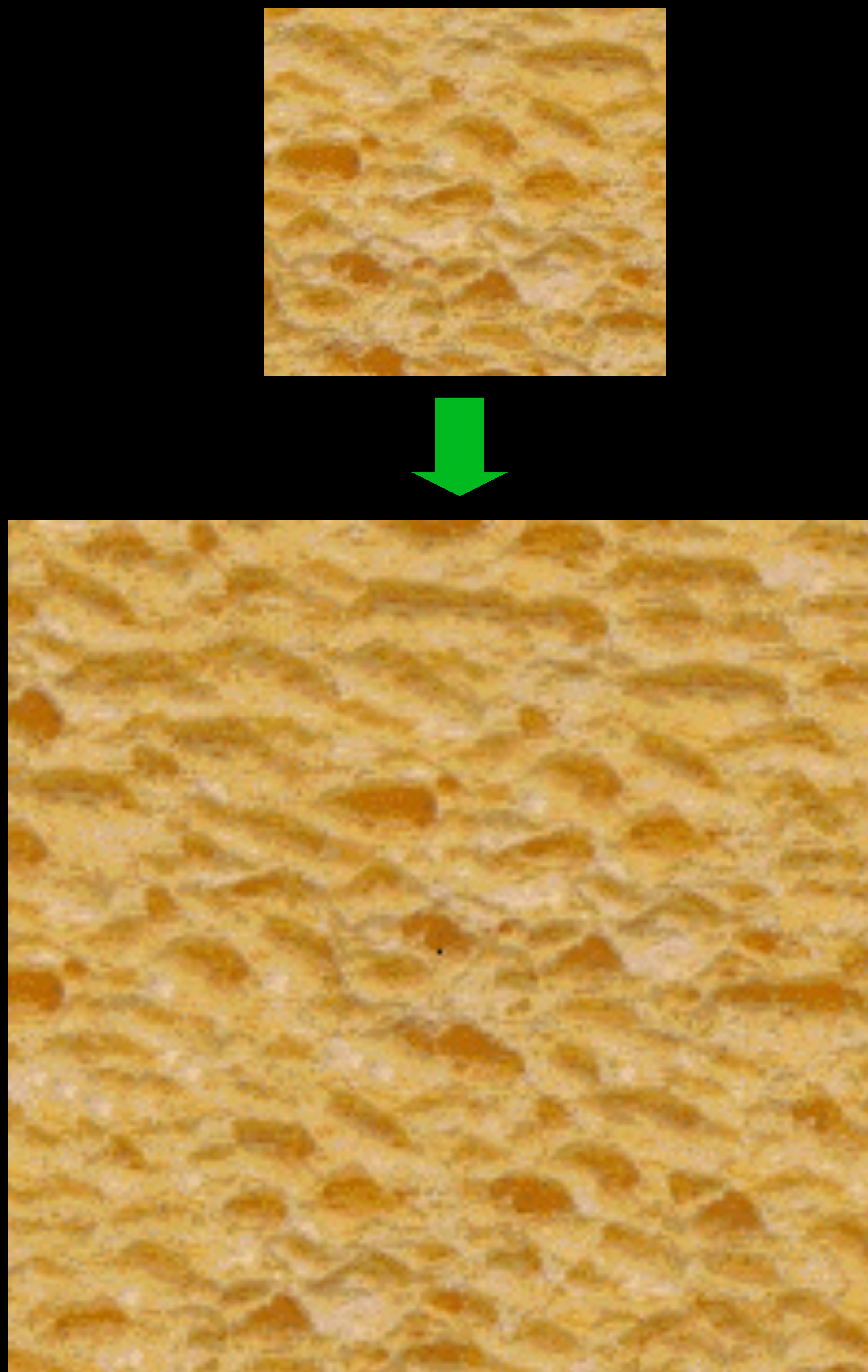


rafia weave

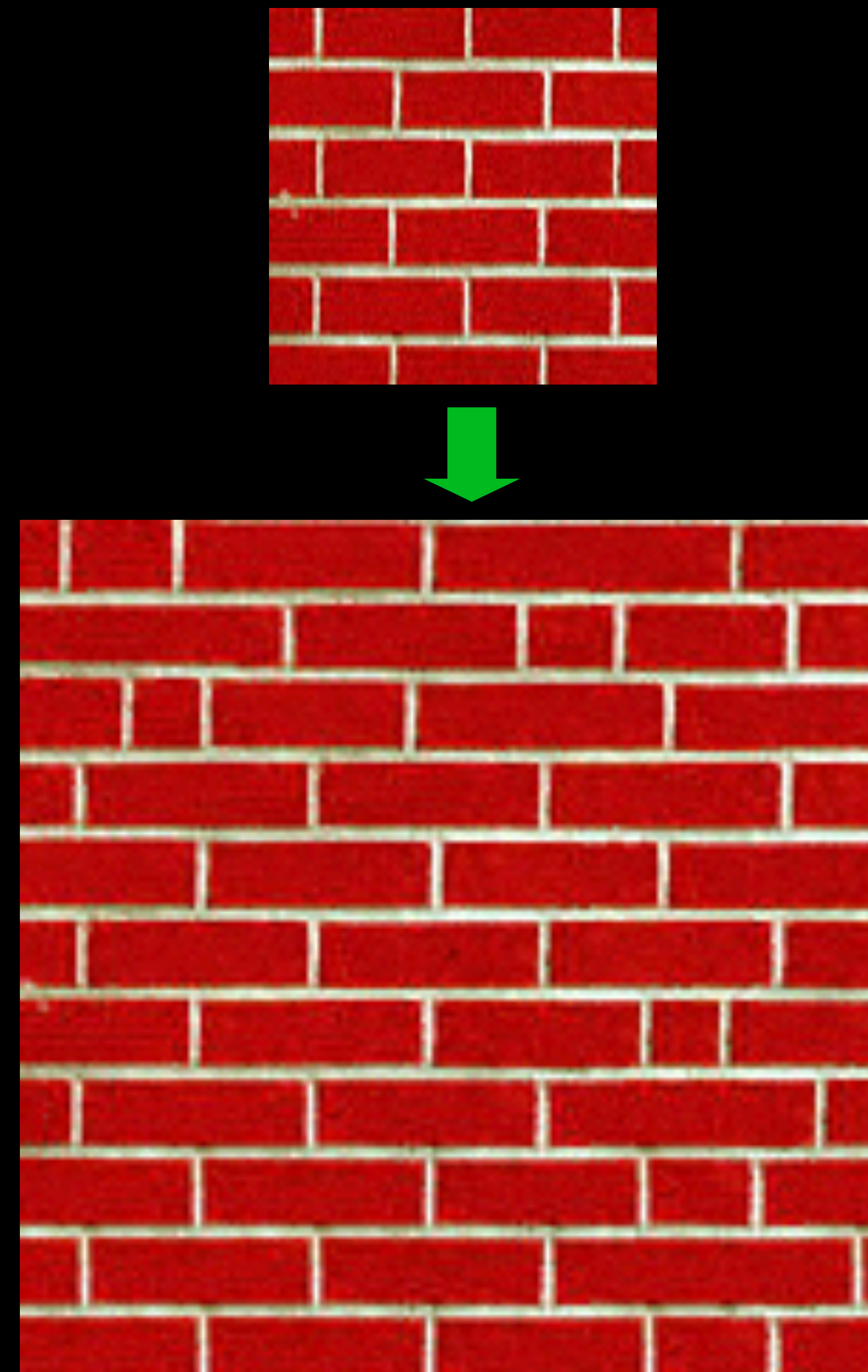


More Results

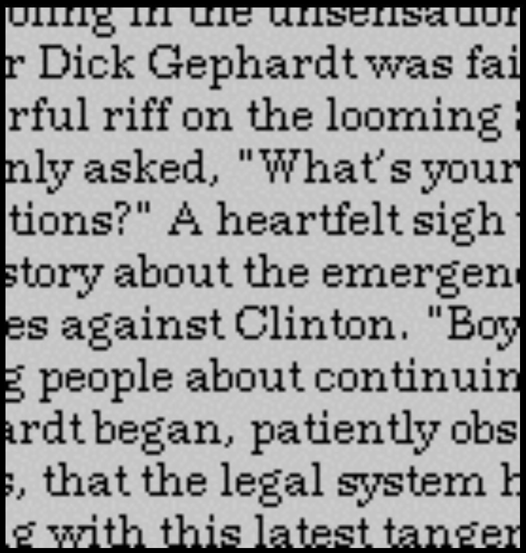
white bread

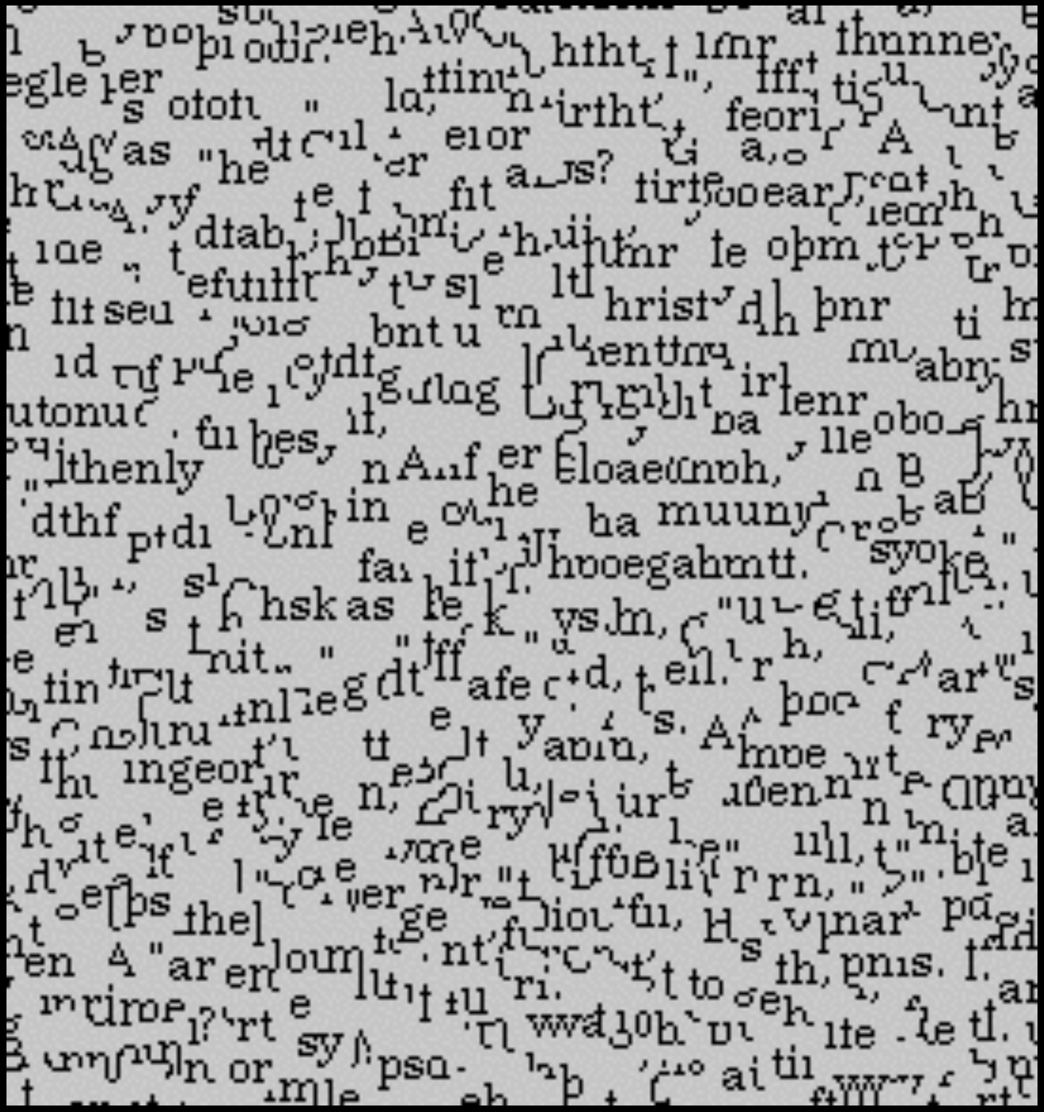


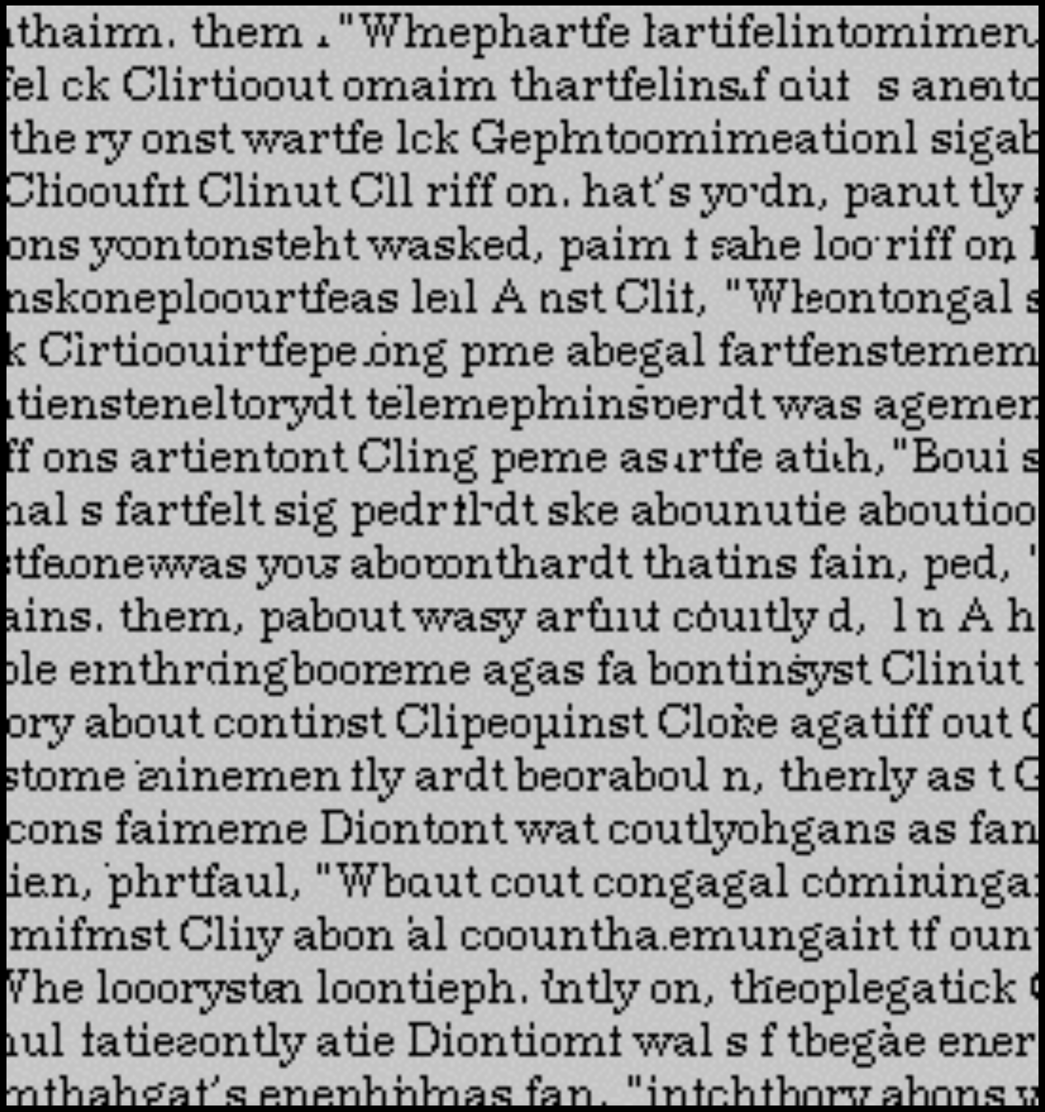
brick wall



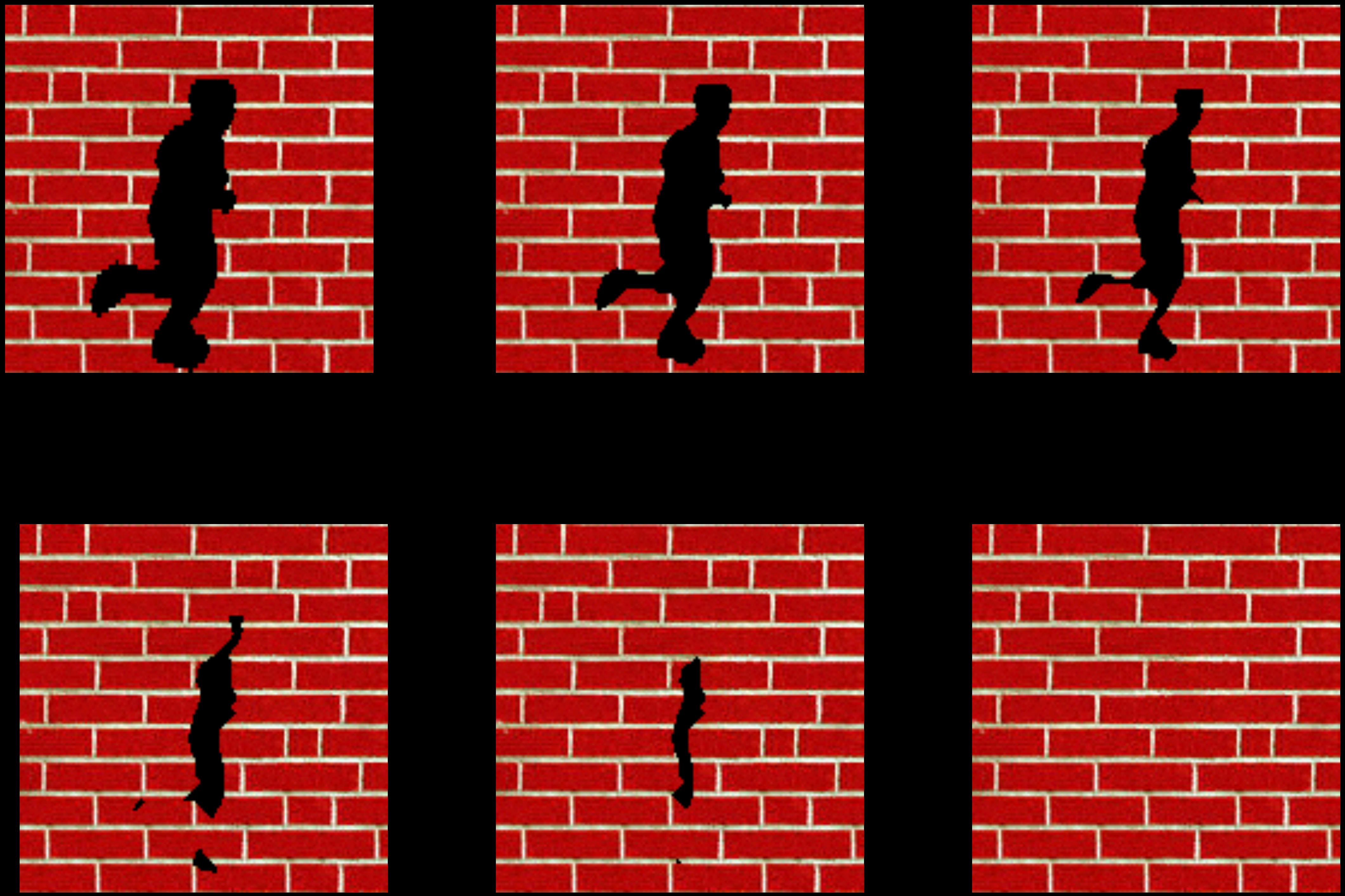
Homage to Shannon



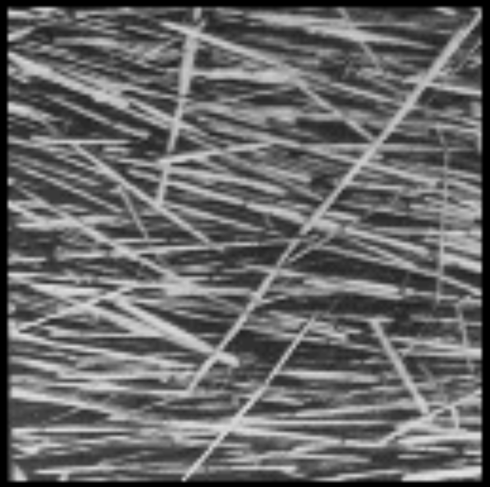


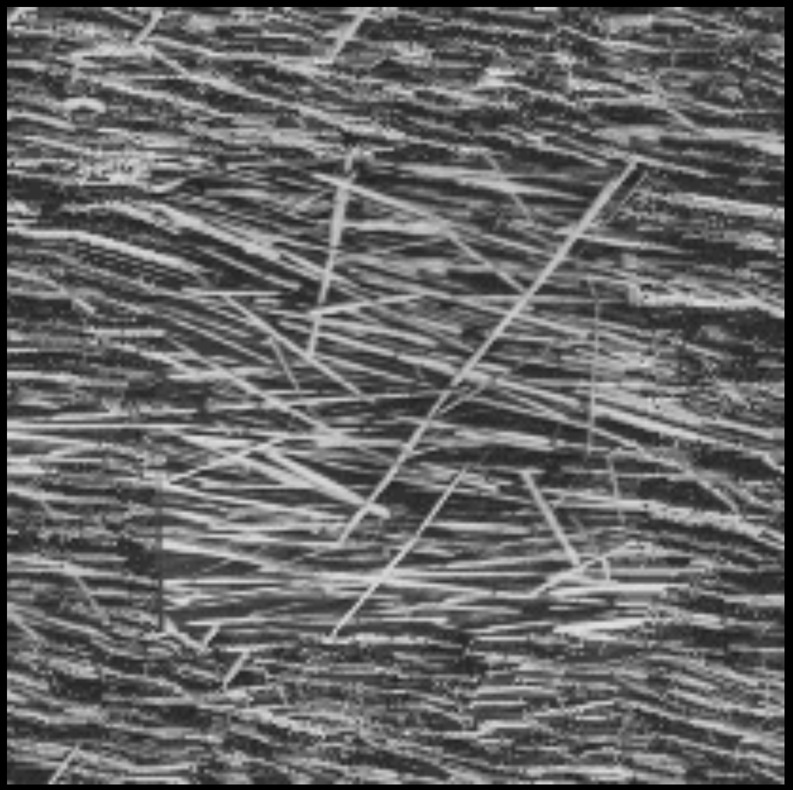


Hole Filling



Extrapolation





What denoising algorithm would result from this non-parametric model for image generation?

A denoising algorithm which implicitly assumes this non-parametric model for image generation

A non-local algorithm for image denoising

Antoni Buades, Bartomeu Coll
Dpt. Matemàtiques i Informàtica, UIB
Ctra. Valldemossa Km. 7.5,
07122 Palma de Mallorca, Spain
vdmiabc4@uib.es, tomeu.coll@uib.es

Jean-Michel Morel
CMLA, ENS Cachan
61, Av du Président Wilson
94235 Cachan, France
morel@cmla.ens-cachan.fr

Non-local means

tificial shocks which can be justified by the computation of its method noise, see [3].

3. NL-means algorithm

Given a discrete noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL[v](i)$, for a pixel i, is computed as a weighted average of all the pixels in the image,

$$NL[v](i) = \sum_{j \in I} w(i,j) v(j),$$

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||v(\mathcal{N}_i) - v(\mathcal{N}_j)||^2_{2,a}}{h^2}},$$

where $Z(i)$ is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{||v(\mathcal{N}_i) - v(\mathcal{N}_j)||^2_{2,a}}{h^2}}$$

and the parameter h acts as a degree of filtering. It controls the decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

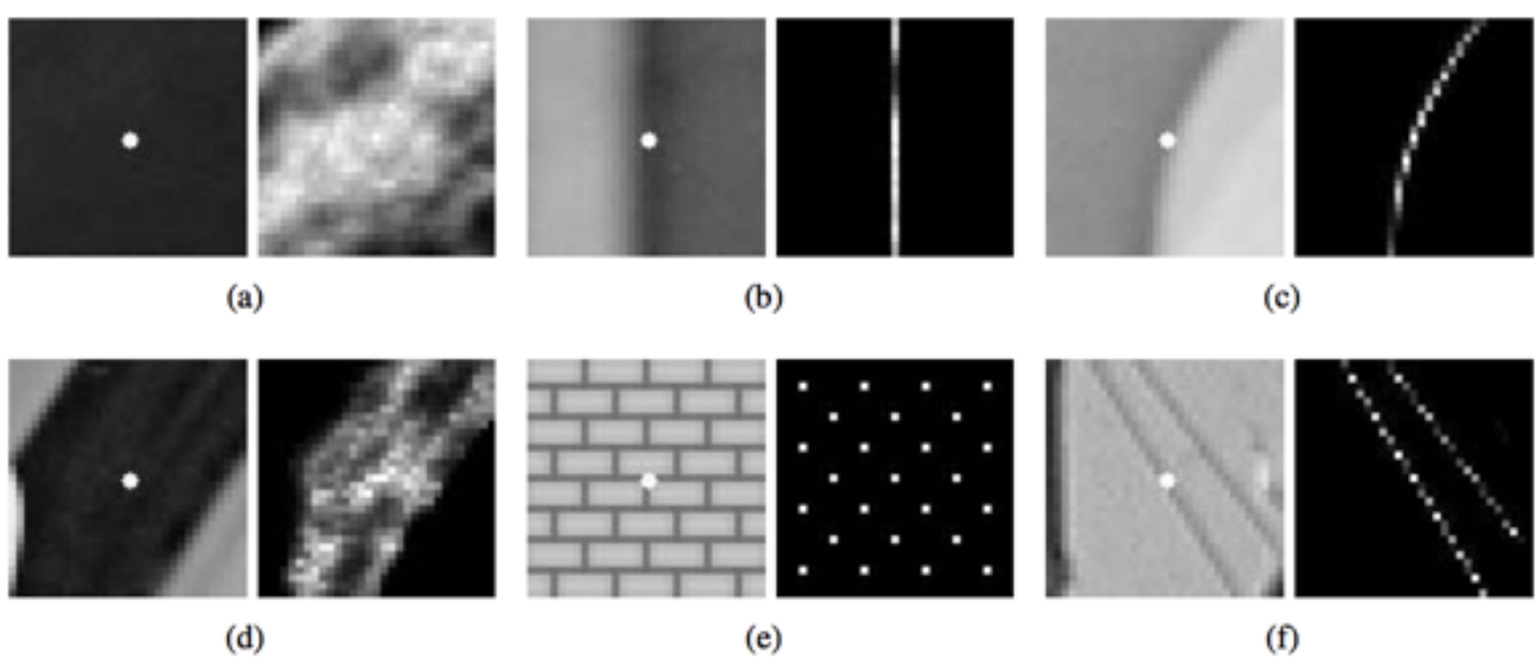


Figure 2. Display of the NL-means weight distribution used to estimate the central pixel of every image. The weights go from 1(white) to zero(black).

Non-local means

Noisy original

Wiener filter output

Total variation output

Non-local means output

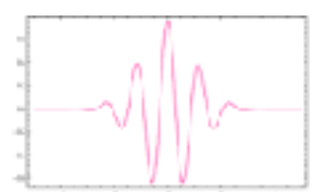
igure 5. Denoising experience on a natural image. From left to right and from top to botto
tandard deviation 20), Gauss filtering, anisotropic filtering, Total variation, Neighborhood f
eans algorithm. The removed details must be compared with the method noise experience, F

Statistical Image Models

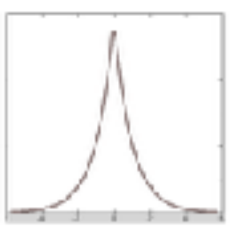
- Gaussian image model
  - image synthesis
  - Wiener filter denoising

Random sinusoid amplitudes of a specified variance

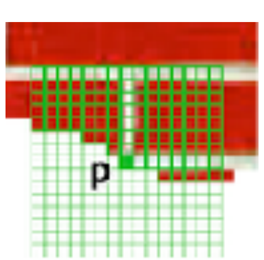
- Kurtotic wavelets model
  - image synthesis
  - Bayesian denoising



Random wavelet amplitudes of a specified distribution



- Non-parametric MRF model
  - image synthesis (Efros and Leung texture model)
  - Non-local means denoising



production rules