Lecture 4: Image pyramids

- PS1 due at midnight
- PS2 out, due next Tues.
- No Thursday office hours this week
- If you're on the waitlist, submit your PS1 via email to the course staff.

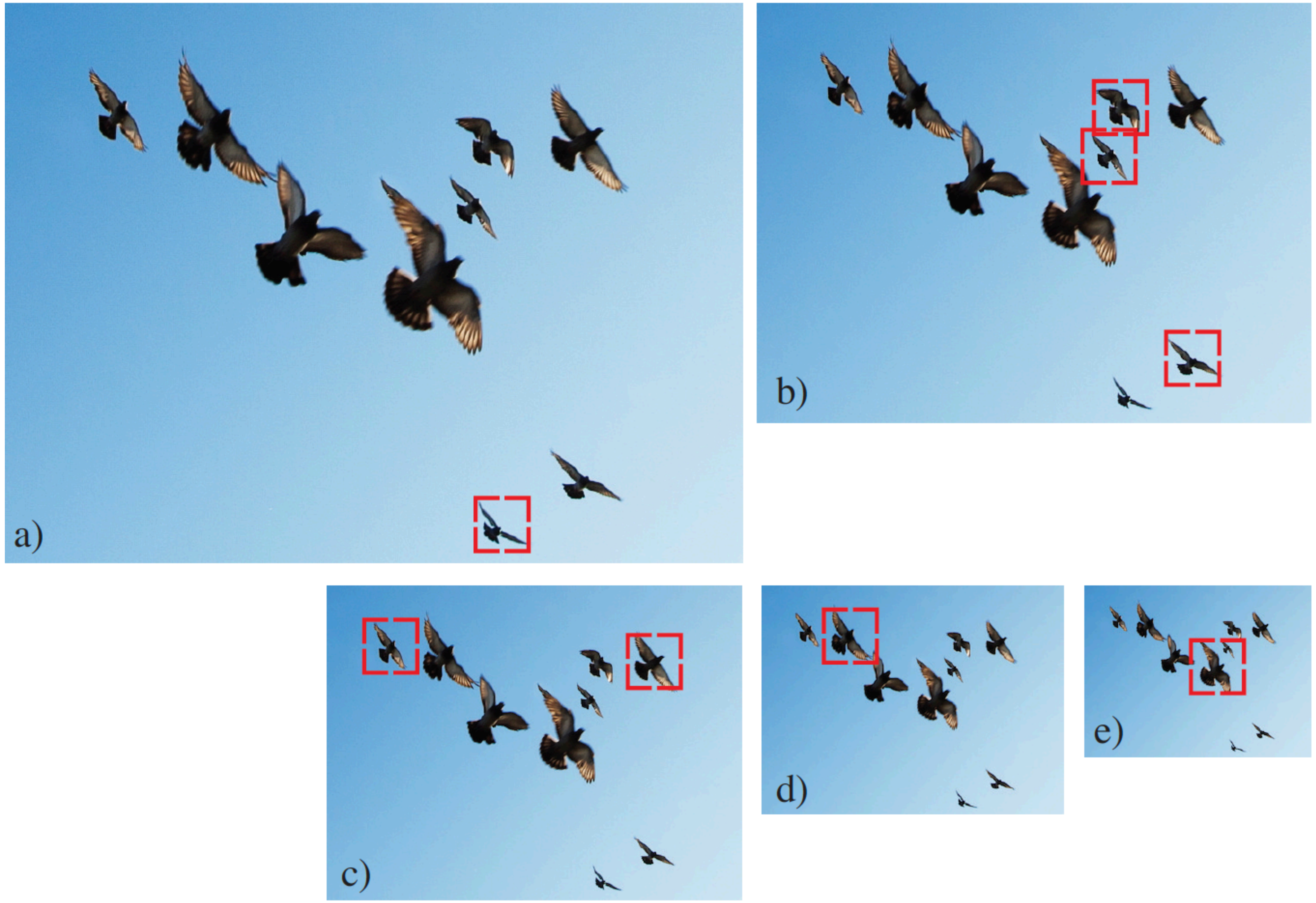
Today

- Image pyramids
- Texture

We want scale and translation invariance.

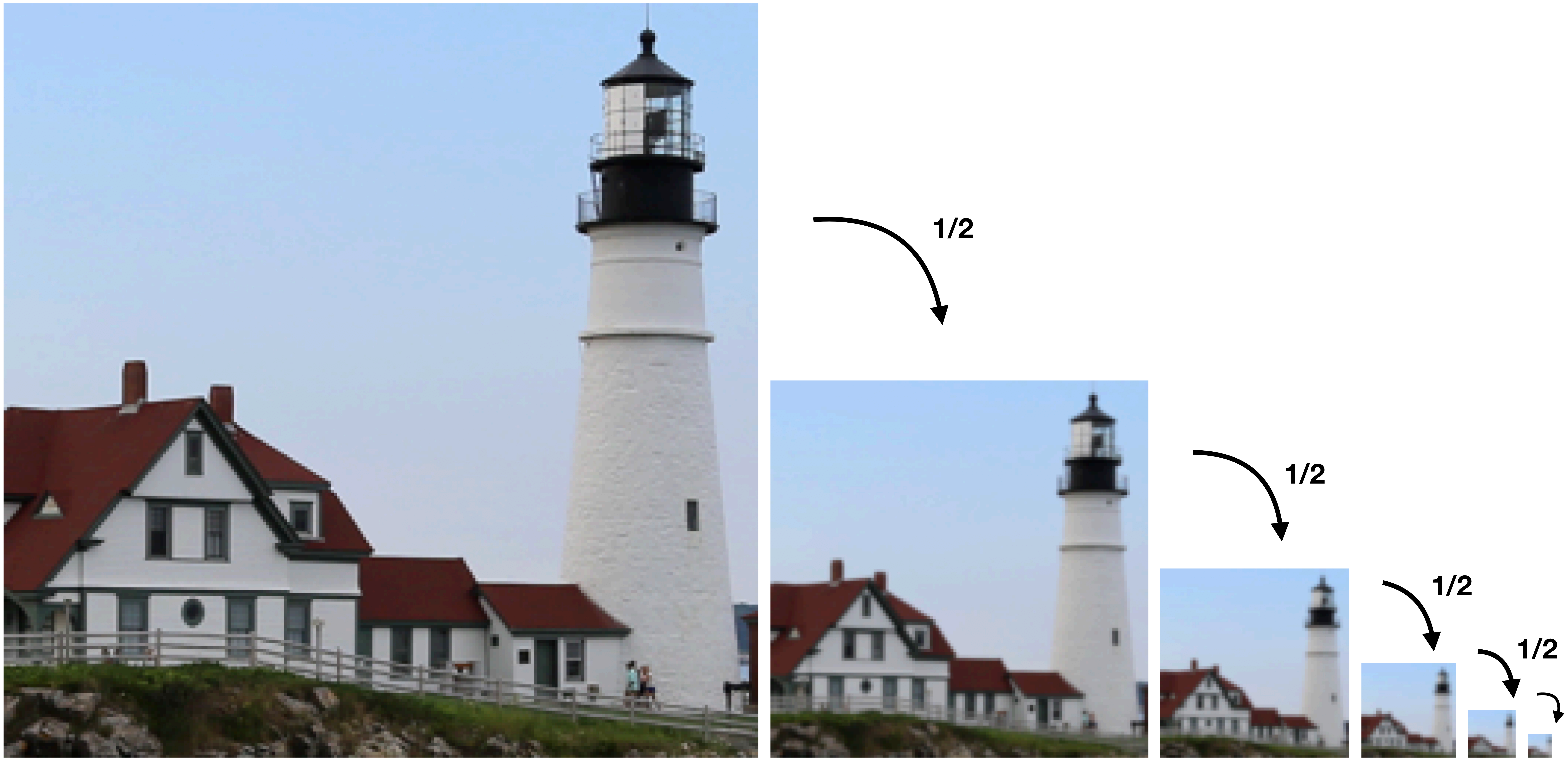
Source: Torralba, Freeman, Isola

Image pyramids



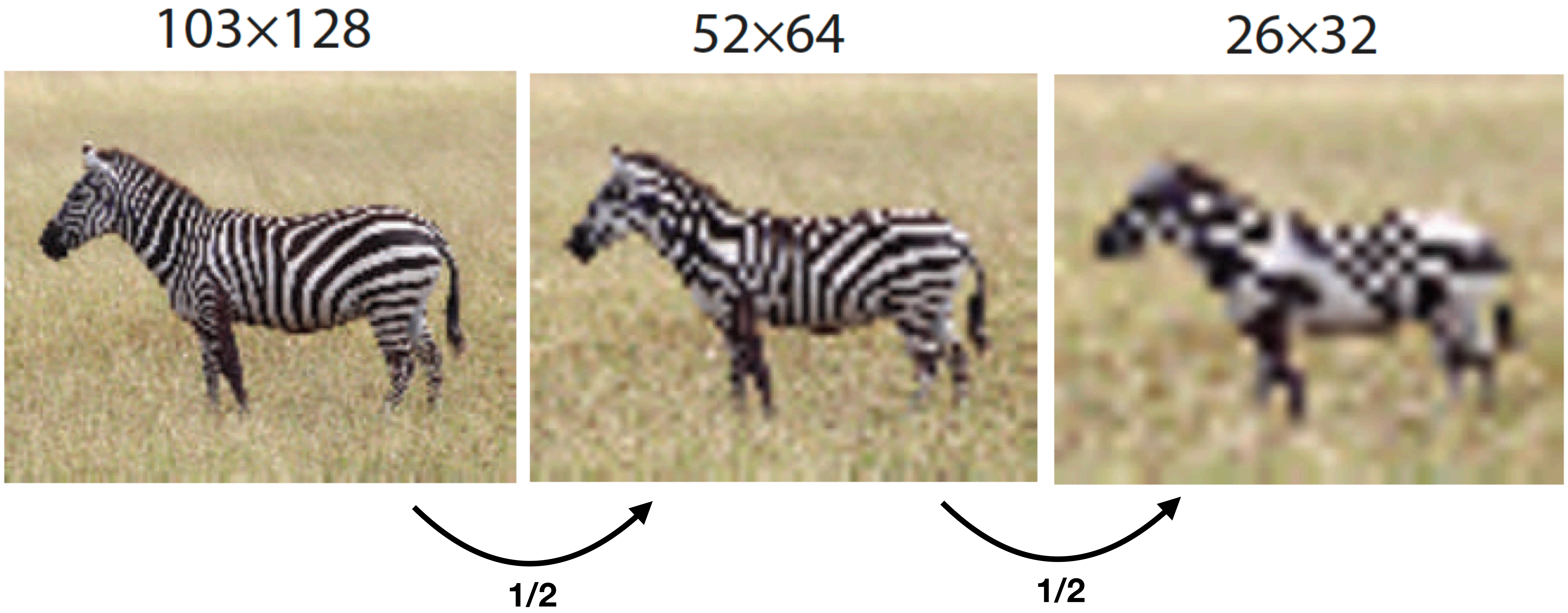
Source: Torralba, Freeman, Isola

Gaussian Pyramid



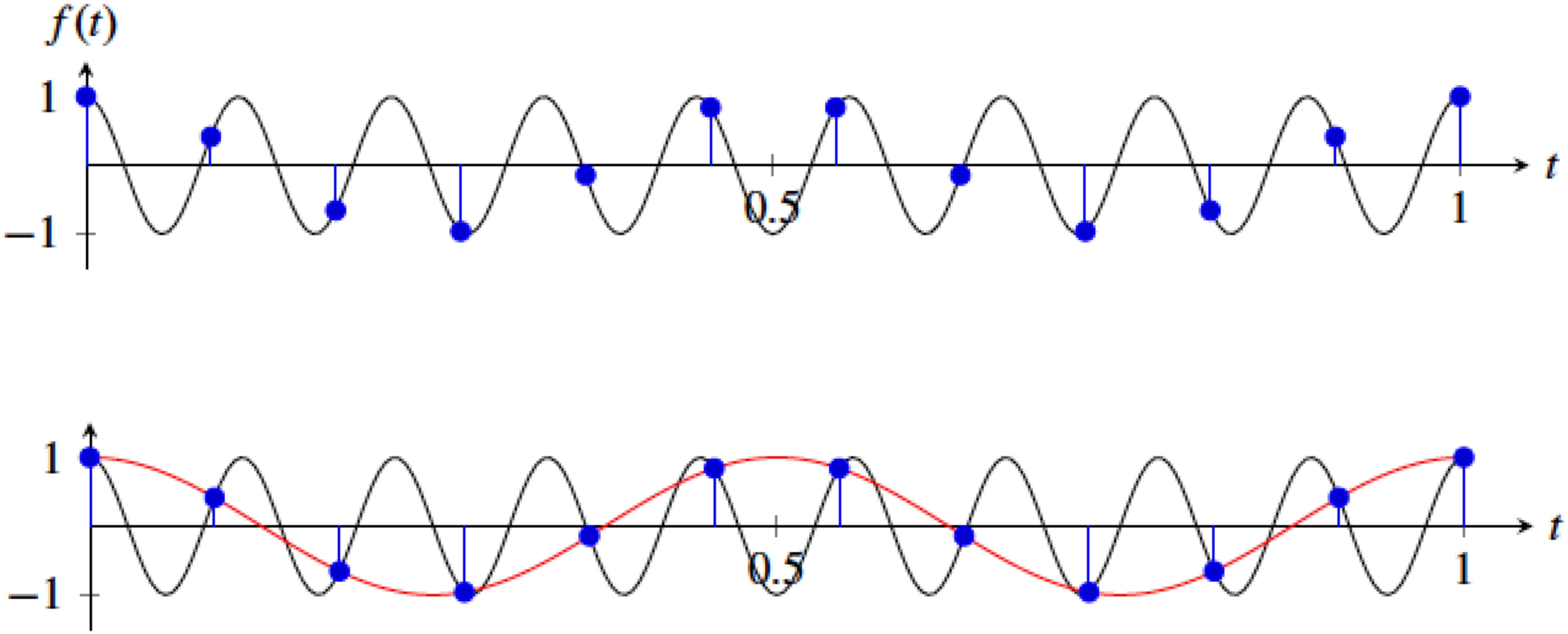
Source: Torralba, Freeman, Isola

Subsampling and aliasing



Source: Torralba, Freeman, Isola

Aliasing



Both waves fit the same samples. We “perceive” the red wave when the actual input was the blue wave.

Source: Torralba, Freeman, Isola

The Gaussian pyramid

For each level

1. Blur input image with a Gaussian filter

[1, 4, 6, 4, 1]

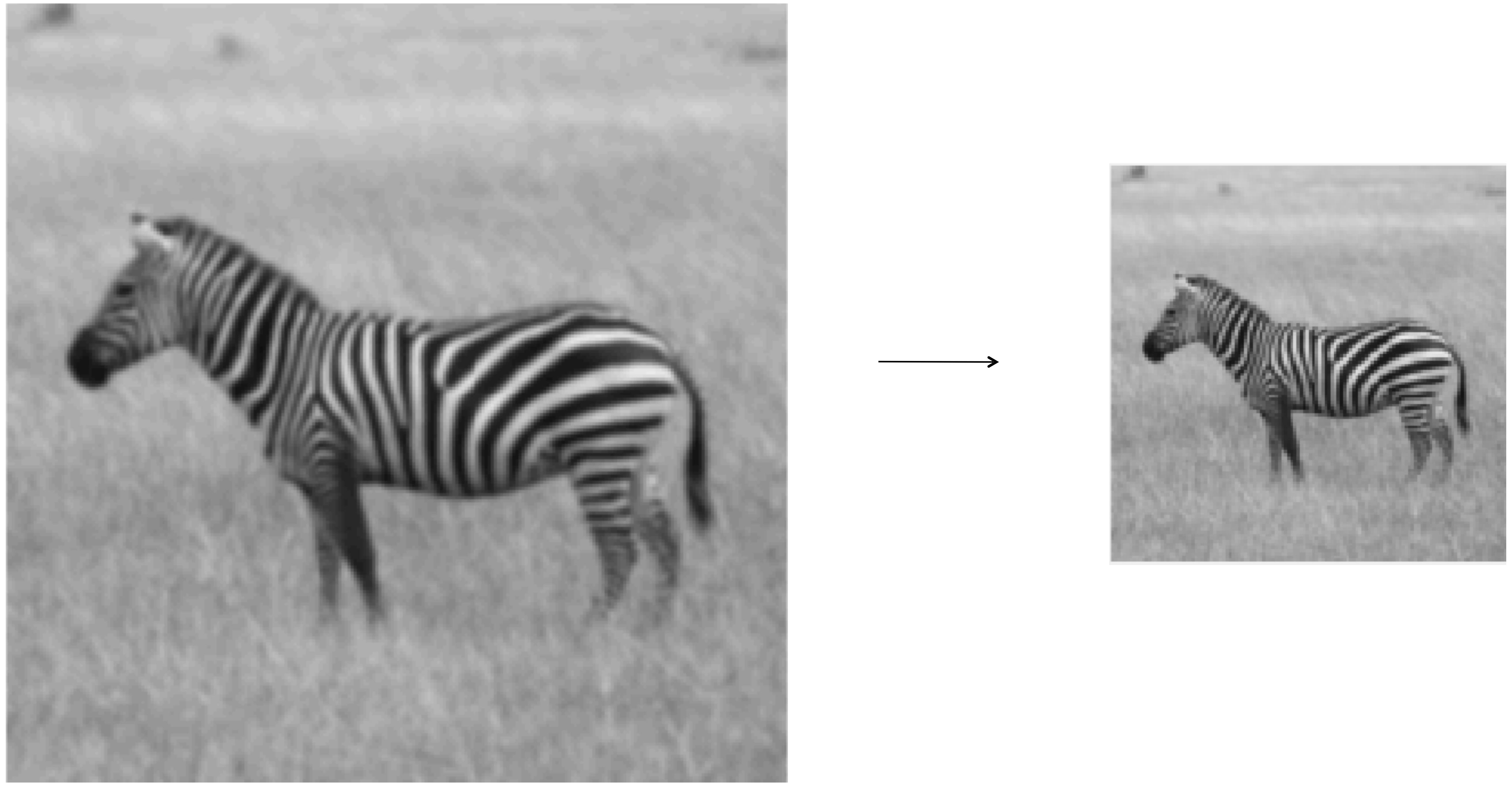
$$\bigcirc \, [1, 4, 6, 4, 1] \, \bigcirc \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} \longrightarrow$$

Source: Torralba, Freeman, Isola

The Gaussian pyramid

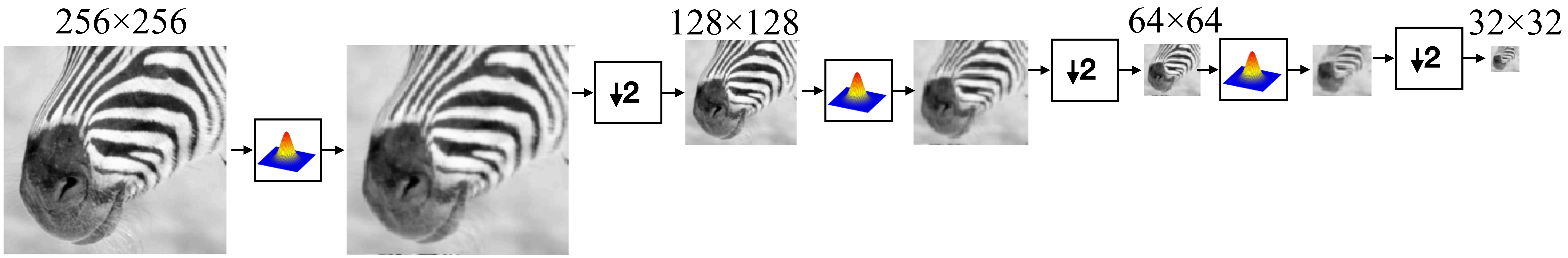
For each level

1. Blur input image with a Gaussian filter
2. Downsample image



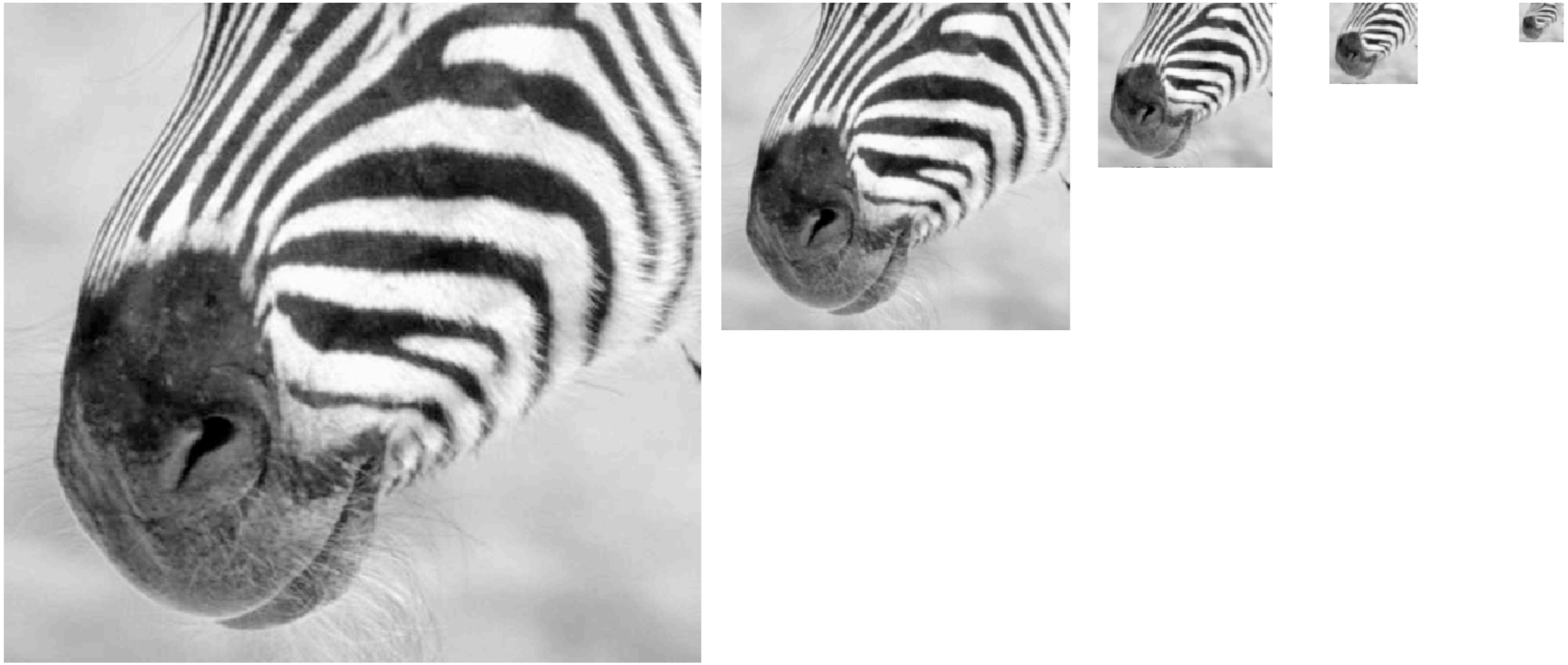
Source: Torralba, Freeman, Isola

The Gaussian pyramid



Source: Torralba, Freeman, Isola

The Gaussian pyramid



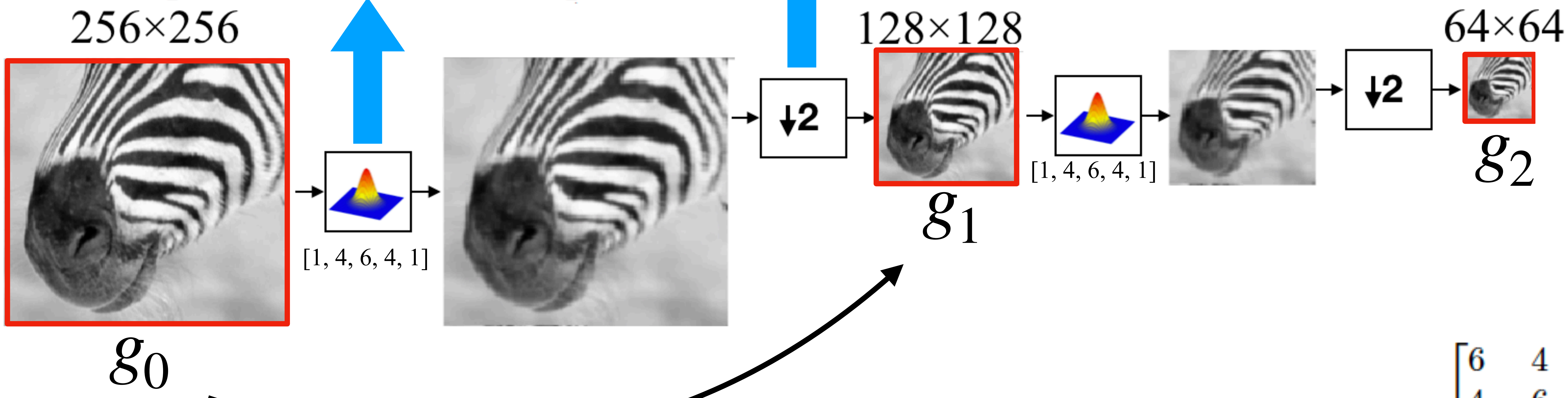
(original image)

Source: Torralba, Freeman, Isola

The Gaussian pyramid

$$\frac{1}{16}\begin{bmatrix}6&4&1&0&0&0&0&0\\4&6&4&1&0&0&0&0\\1&4&6&4&1&0&0&0\\0&1&4&6&4&1&0&0\\0&0&1&4&6&4&1&0\\0&0&0&1&4&6&4&1\\0&0&0&0&1&4&6&4\\0&0&0&0&0&1&4&6\end{bmatrix}$$

$$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0\end{bmatrix}$$



$$g_1 = G_0 g_0$$

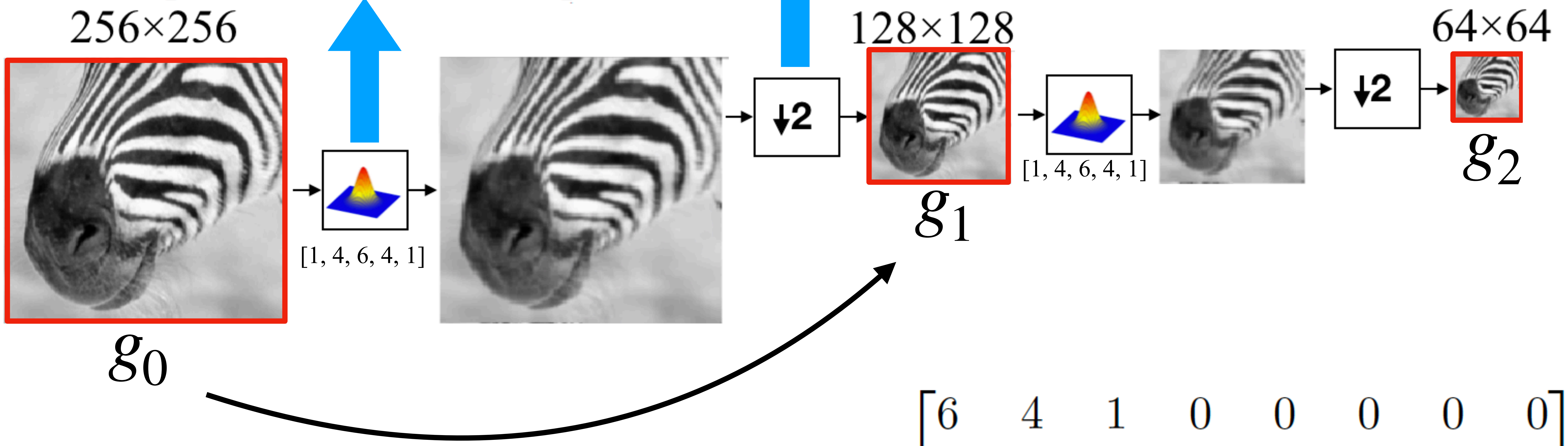
$$G_0 = \begin{bmatrix}1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0\end{bmatrix}\frac{1}{16}\begin{bmatrix}6&4&1&0&0&0&0&0\\4&6&4&1&0&0&0&0\\1&4&6&4&1&0&0&0\\0&1&4&6&4&1&0&0\\0&0&1&4&6&4&1&0\\0&0&0&1&4&6&4&1\\0&0&0&0&1&4&6&4\\0&0&0&0&0&1&4&6\end{bmatrix}$$

Source: Torralba, Freeman, Isola

The Gaussian pyramid

$$\frac{1}{16}\begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

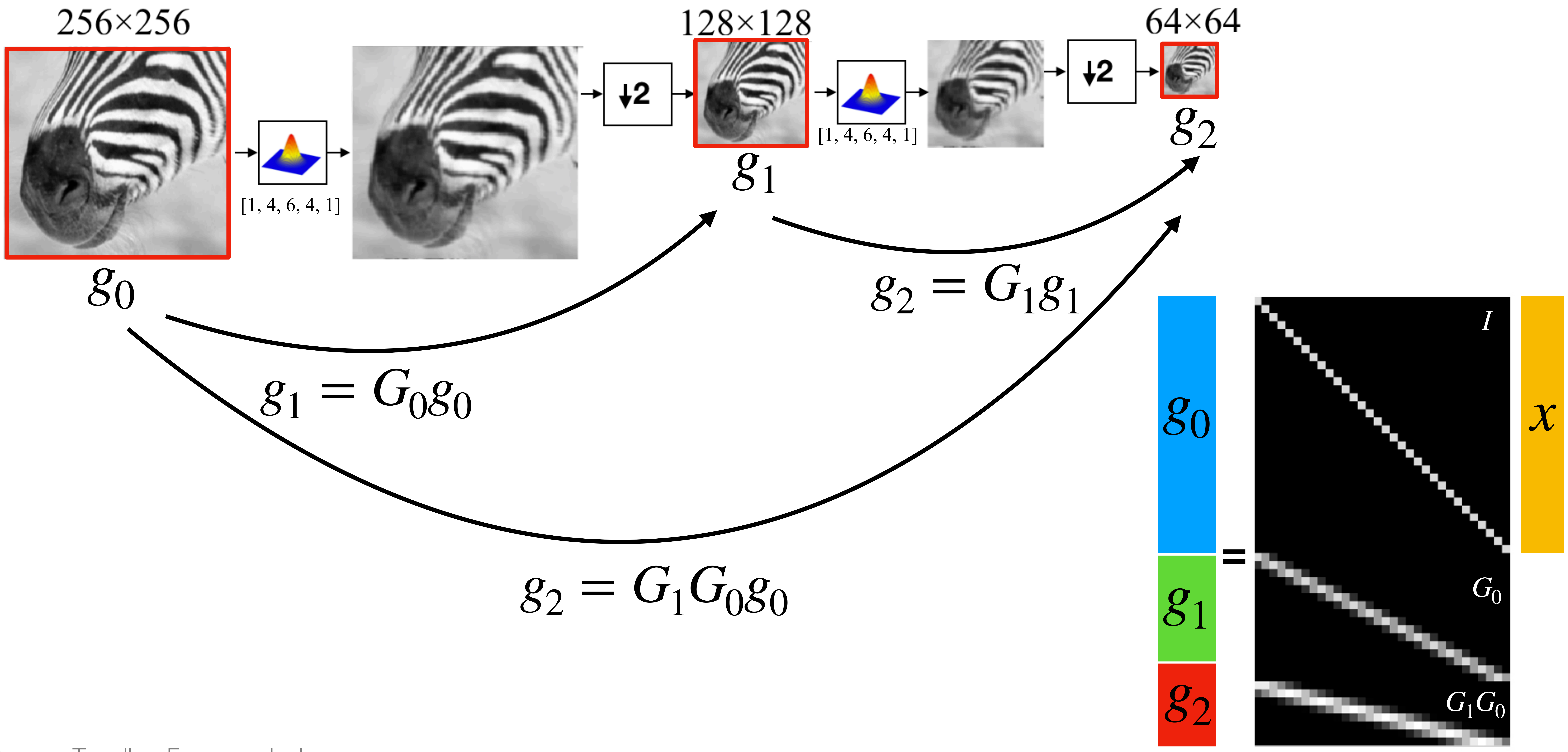


$$g_1 = G_0 g_0$$

$$G_0 = \frac{1}{16}\begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \end{bmatrix}$$

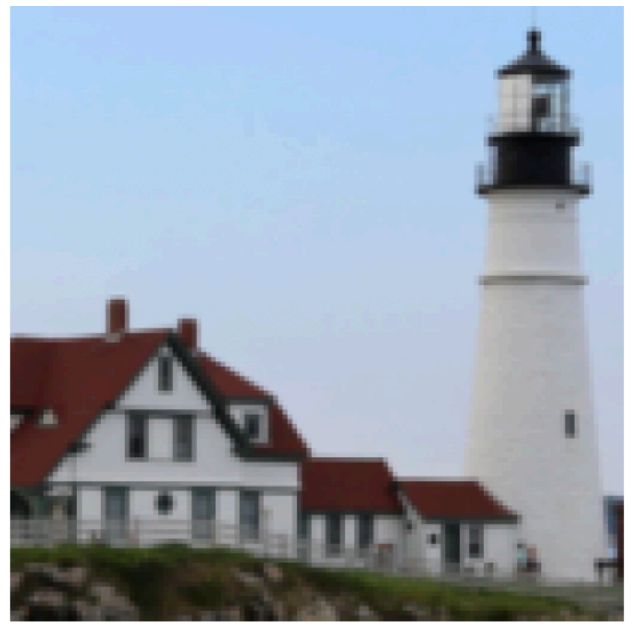
Source: Torralba, Freeman, Isola

The Gaussian pyramid



Source: Torralba, Freeman, Isola

The Gaussian pyramid



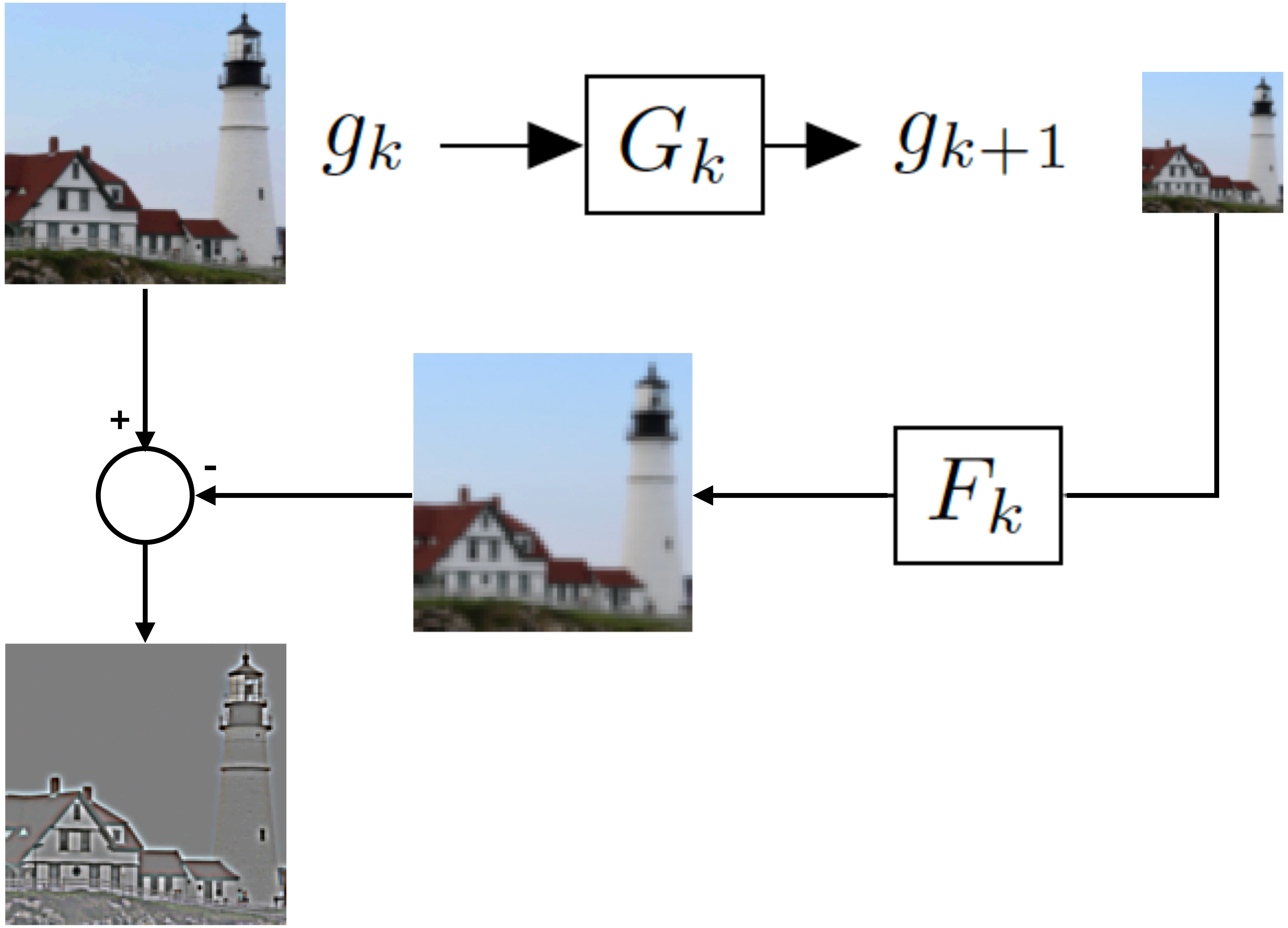
$$g_k \longrightarrow \boxed{G_k} \longrightarrow g_{k+1}$$

For each level

1. Blur input image with a Gaussian filter
2. Downsample image

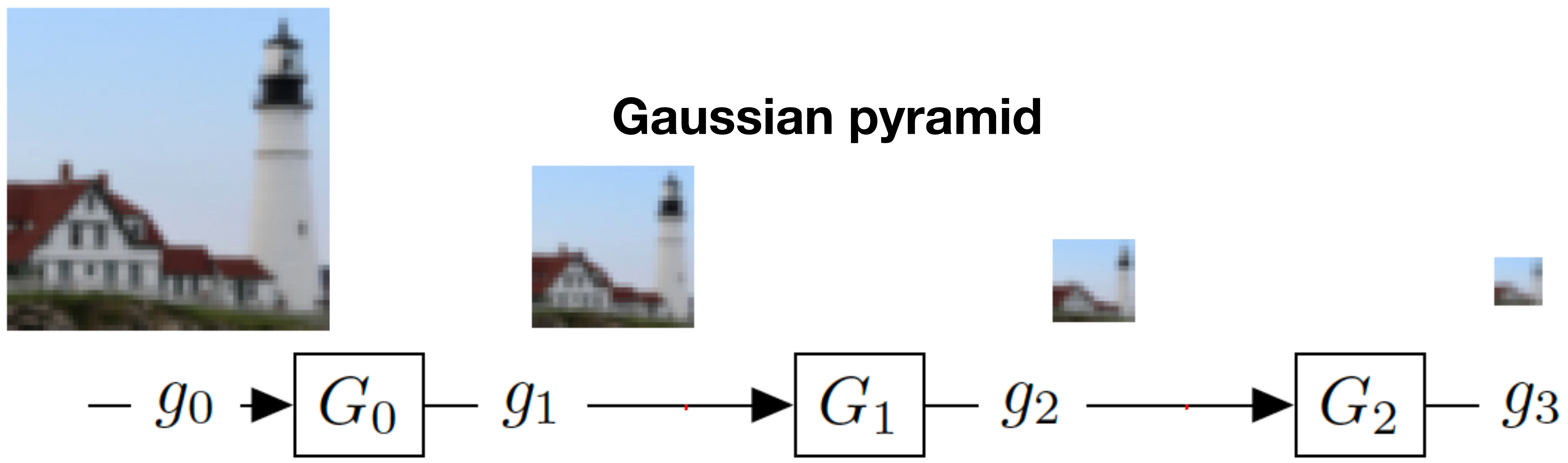
The Laplacian Pyramid

Compute the difference between upsampled Gaussian pyramid level k+1 and Gaussian pyramid level k. Recall that this approximates the blurred Laplacian.



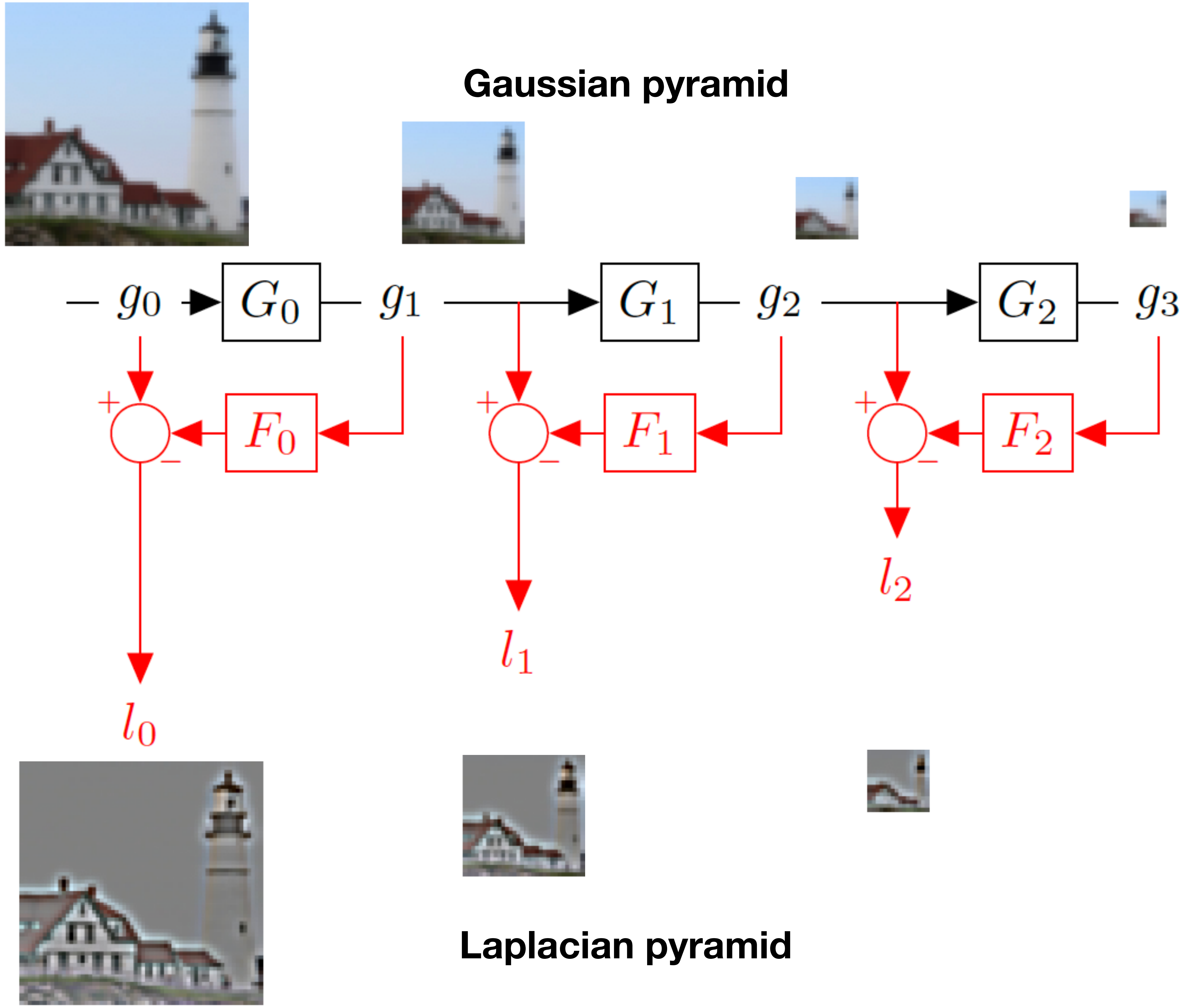
Source: Torralba, Freeman, Isola

The Laplacian Pyramid



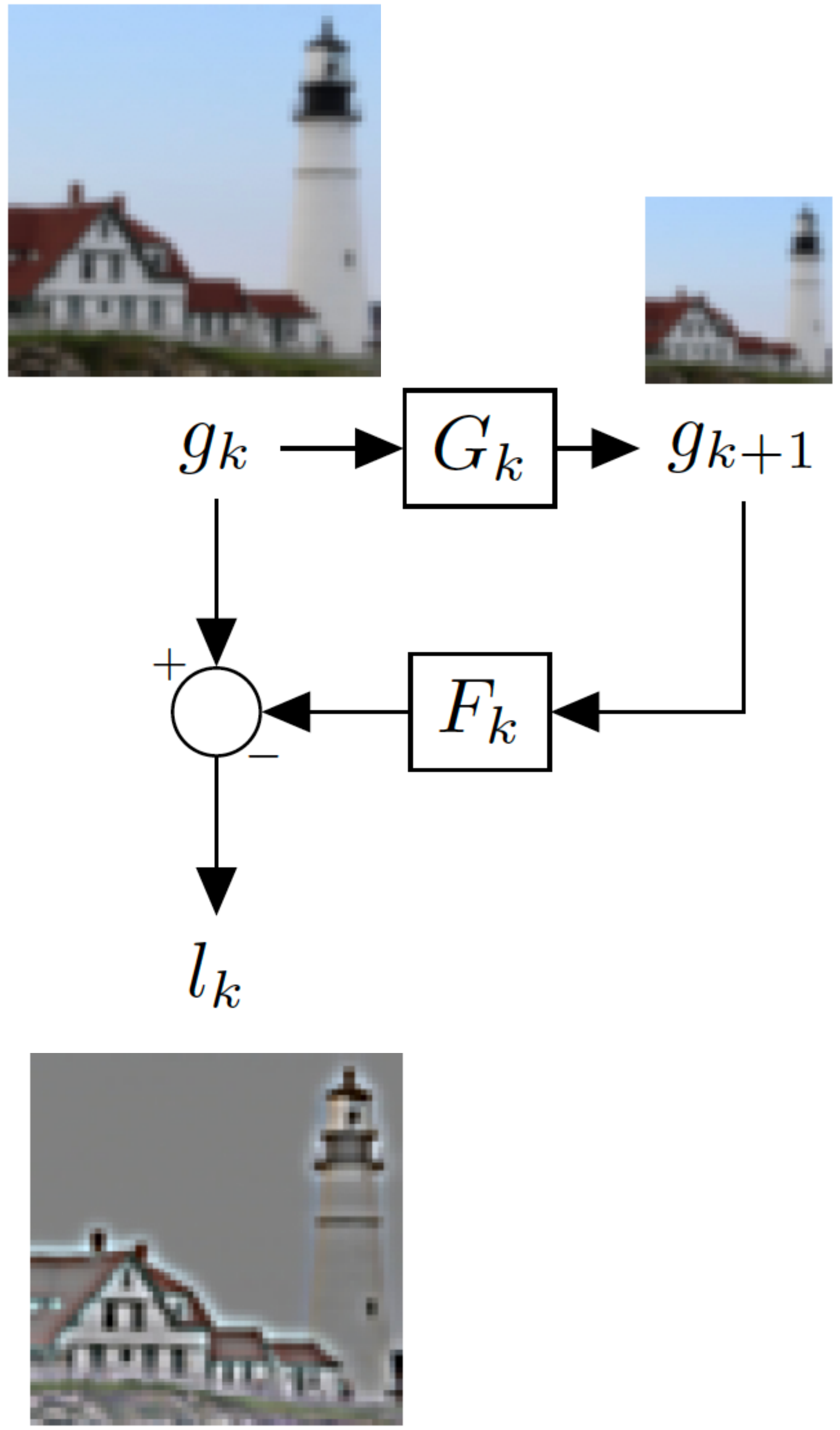
Source: Torralba, Freeman, Isola

The Laplacian Pyramid



Source: Torralba, Freeman, Isola

The Laplacian Pyramid



Blurring and downsampling:

$$G_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \frac{1}{16} \begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix}$$

(Downsampling by 2)

(blur)

Upsampling and blurring:

$$F_0 =$$

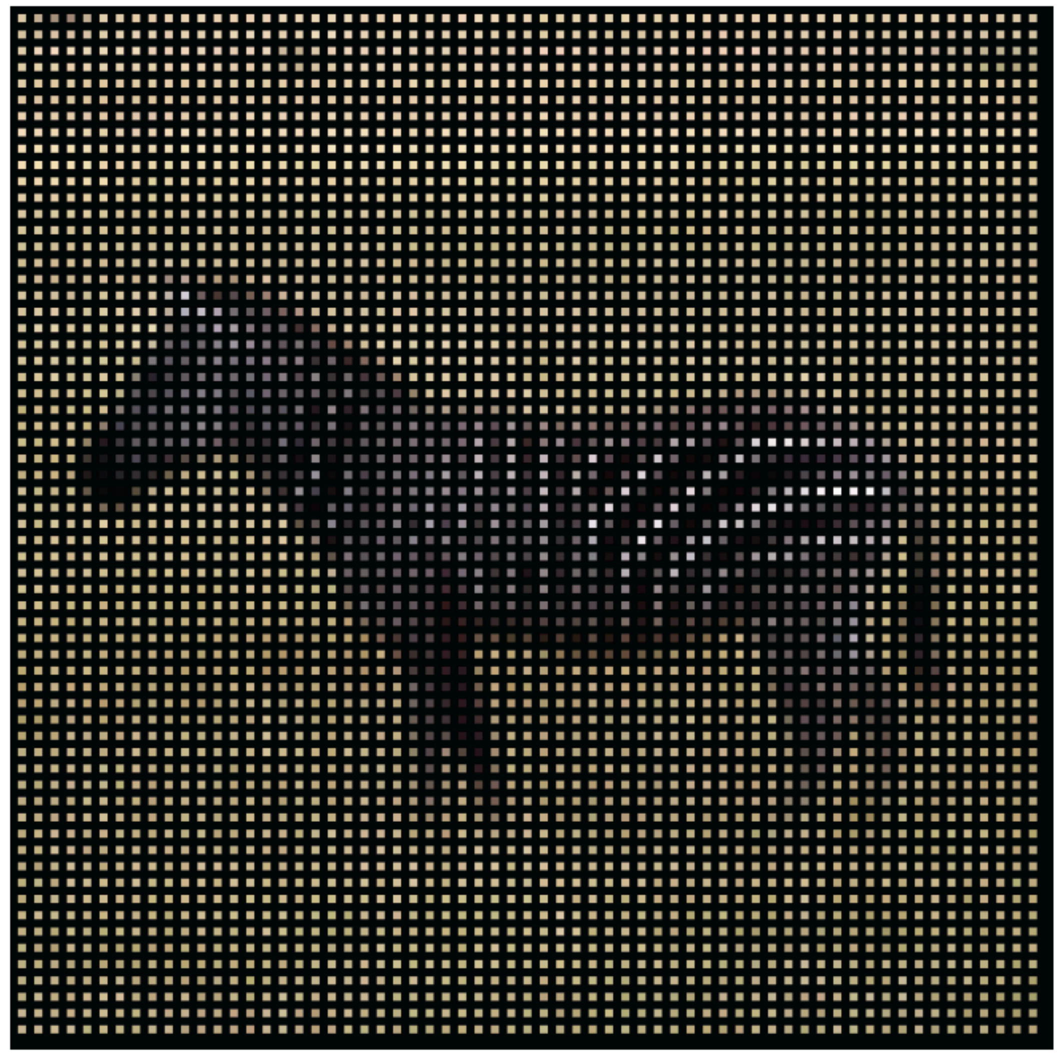
Source: Torralba, Freeman, Isola

Upsampling

64x64

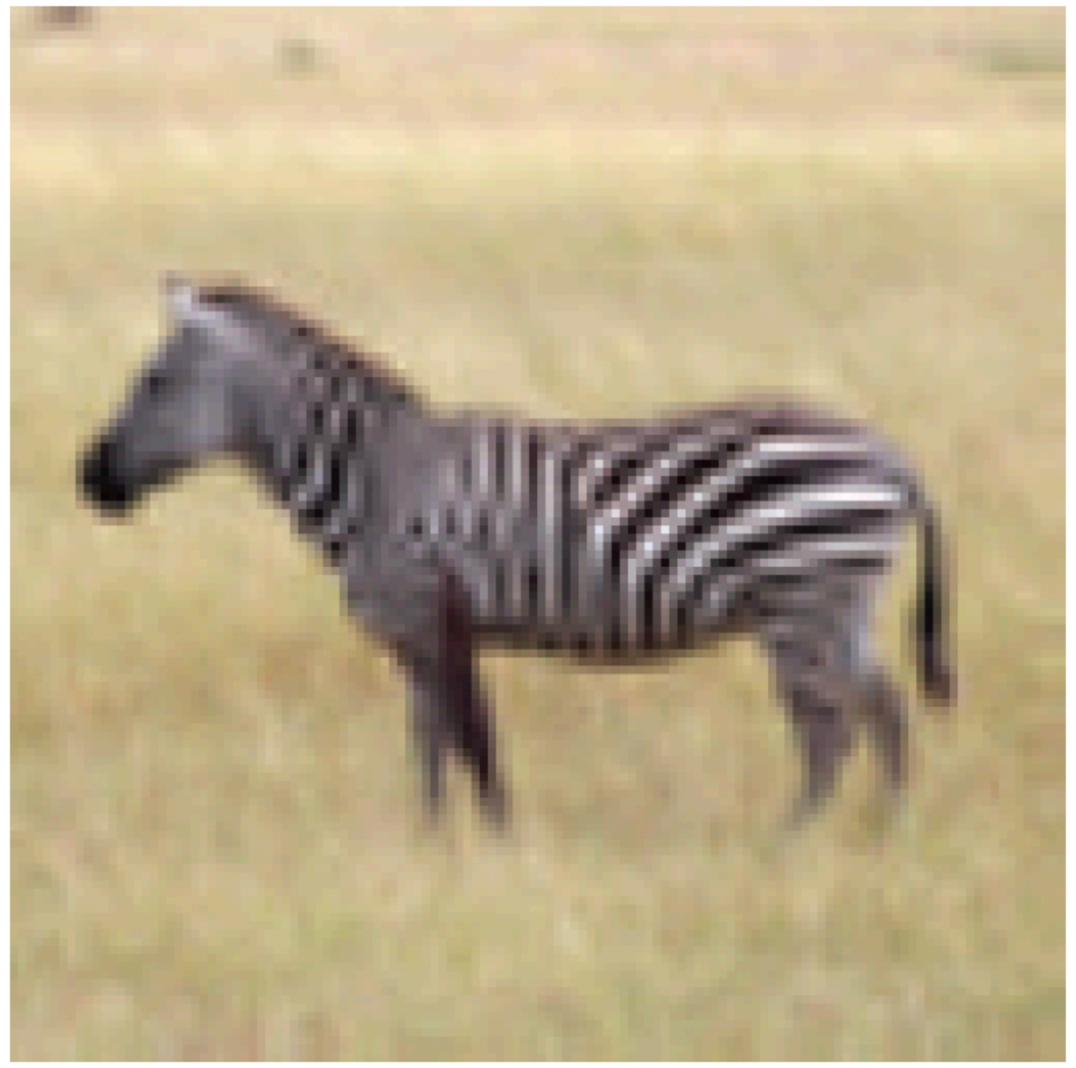
Insert zeros

128x128



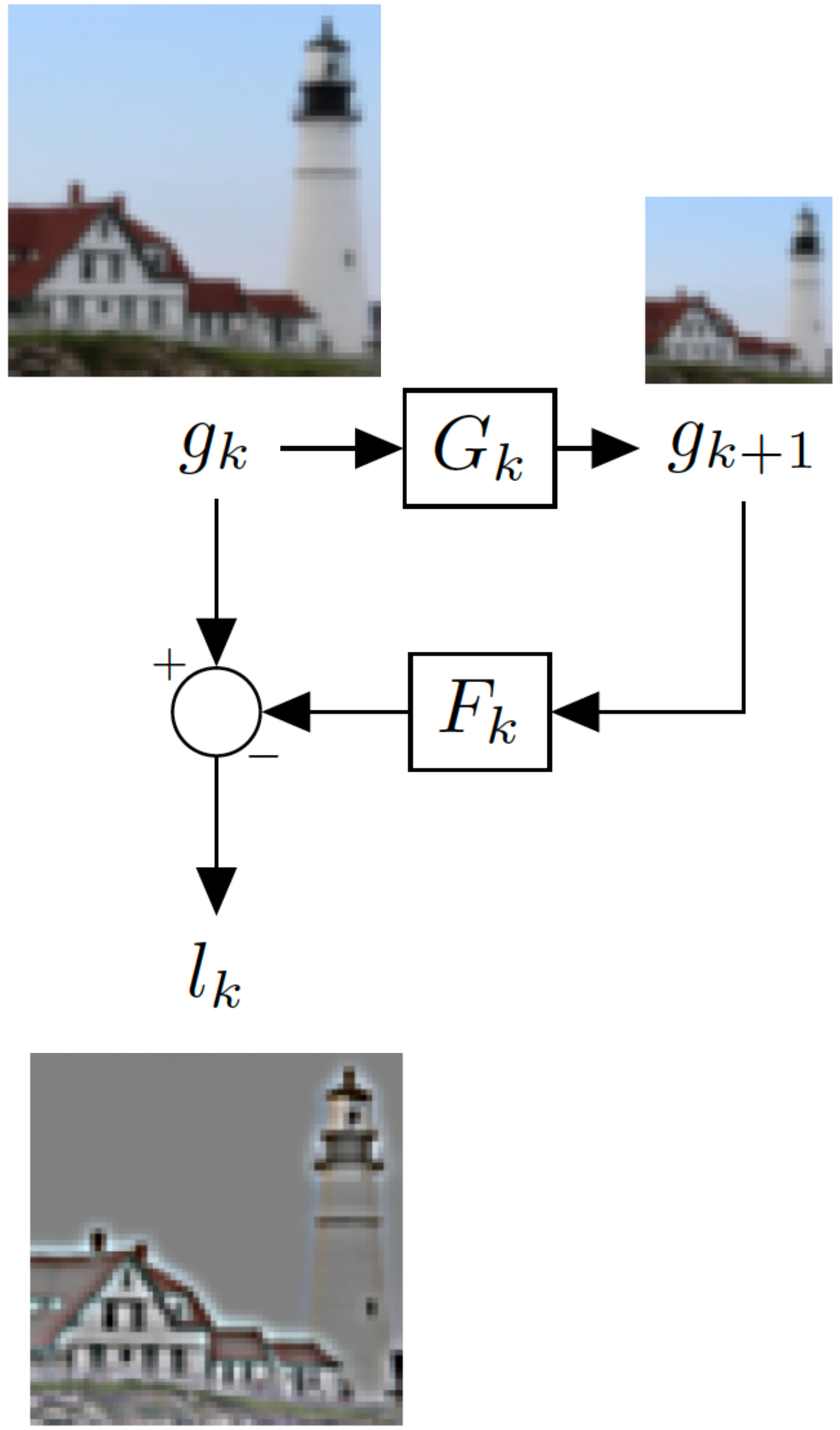
$$\bigcirc \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix} =$$

128x128



Source: Torralba, Freeman, Isola

The Laplacian Pyramid



Blurring and downsampling:

$$G_0 = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{(Downsampling by 2)}} \frac{1}{16} \underbrace{\begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix}}_{\text{(blur)}}$$

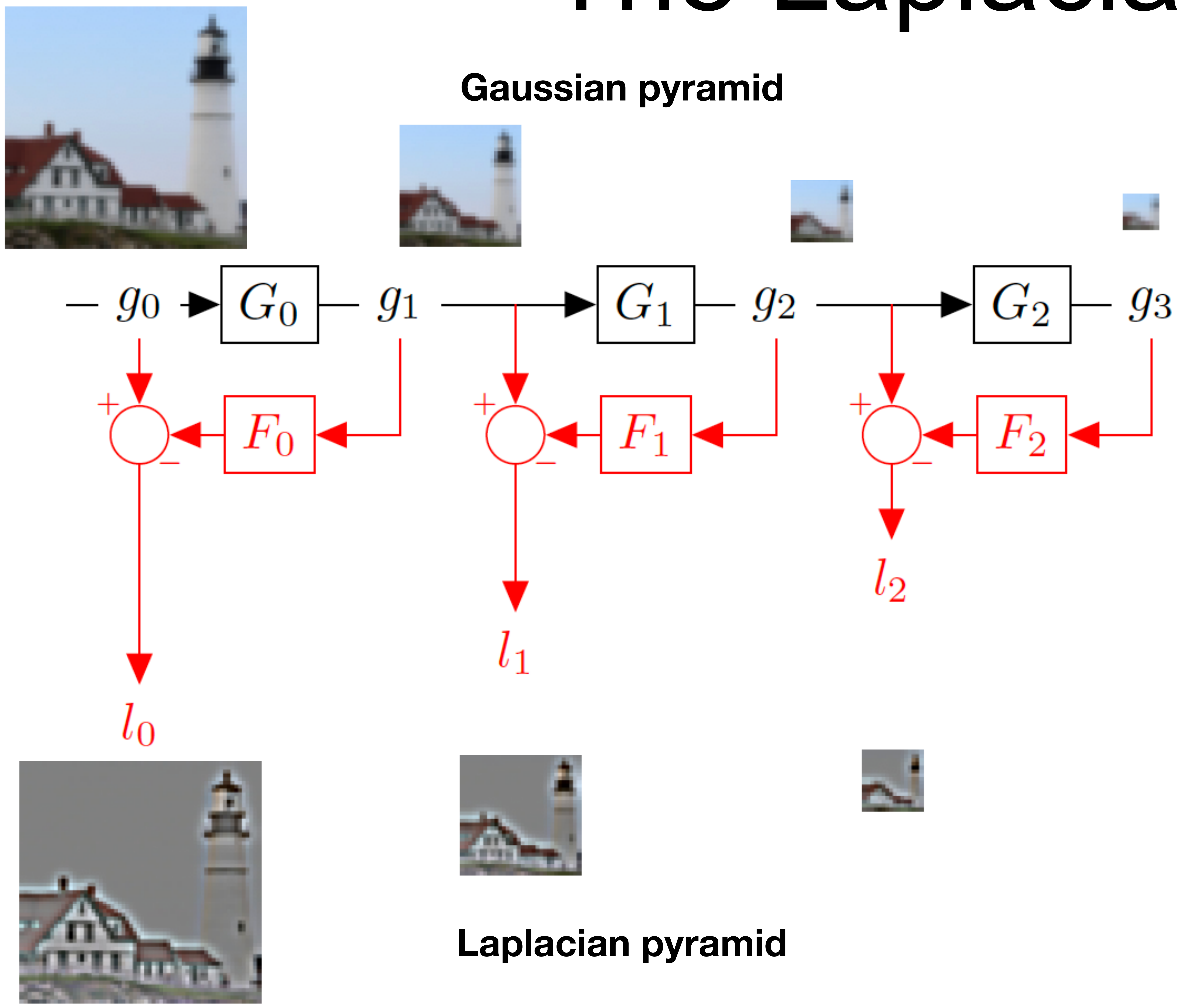
Upsampling and blurring:

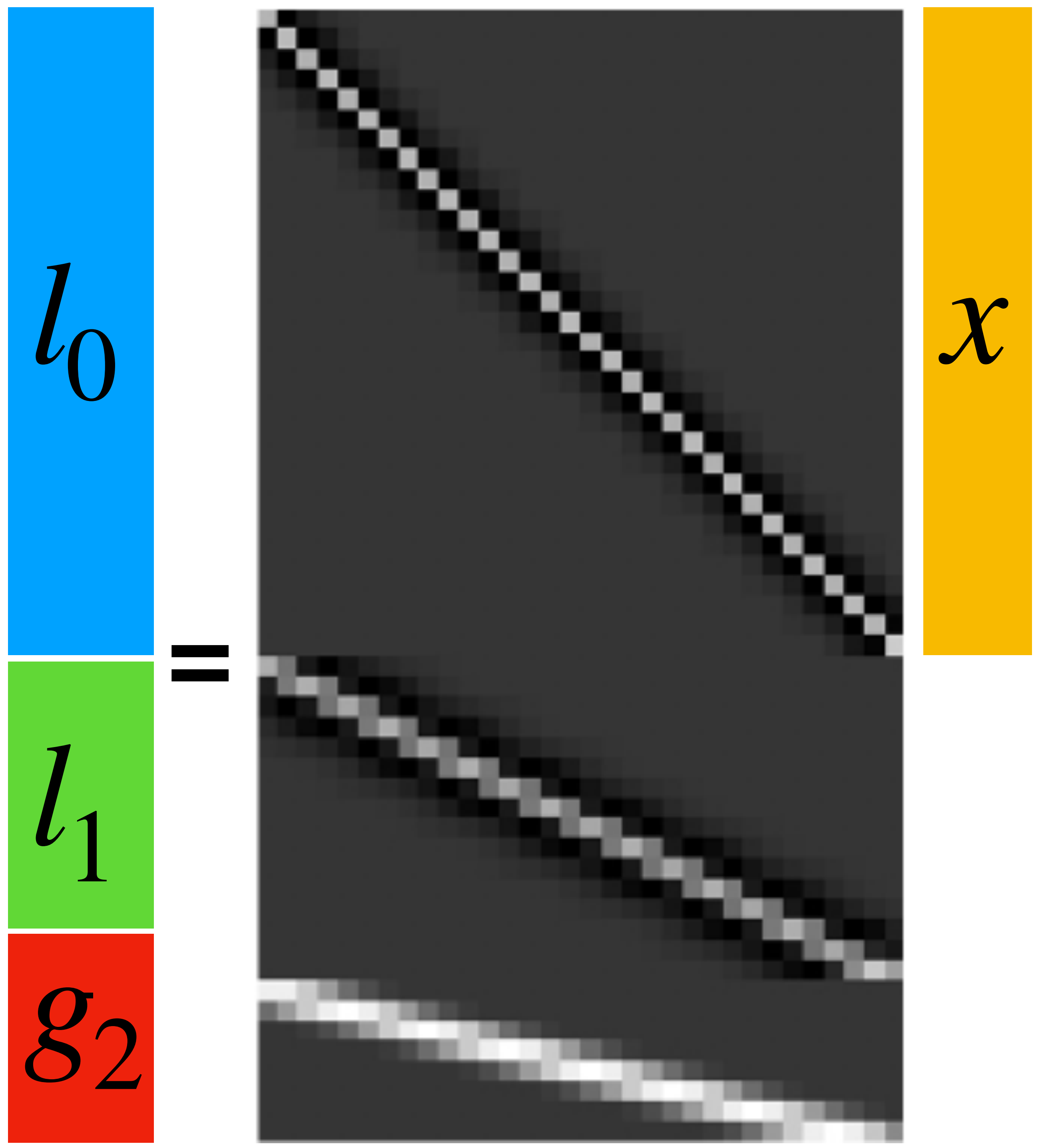
$$F_0 = \frac{1}{8} \underbrace{\begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 4 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 6 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 6 & 4 & 1 & 0 \\ 0 & 0 & 0 & 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 4 & 6 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 4 & 6 \end{bmatrix}}_{\text{(blur)}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{(Upsampling by 2)}}$$

$$\boxed{l_0 = (I_0 - F_0 G_0) g_0}$$

Source: Torralba, Freeman, Isola

The Laplacian Pyramid





Source: Torralba, Freeman, Isola

The Laplacian Pyramid

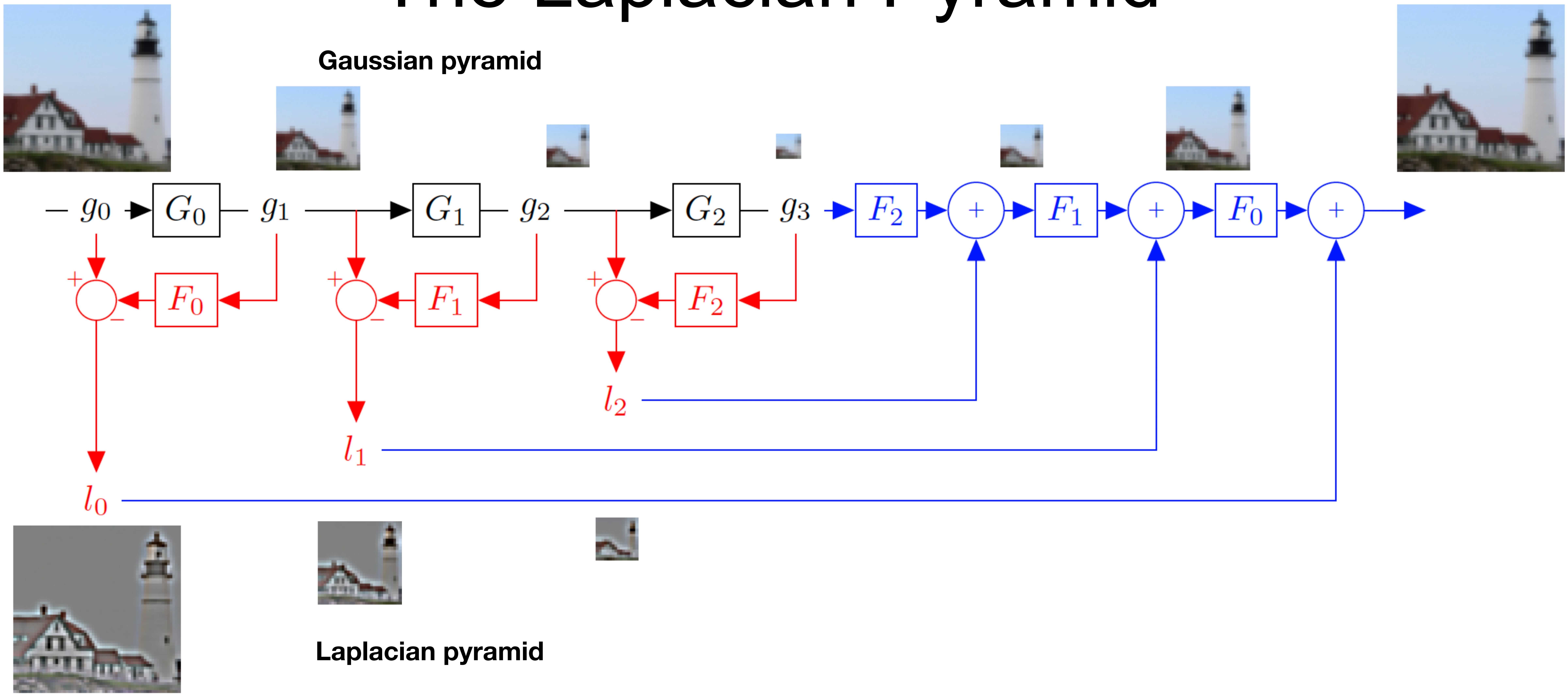
Gaussian residual

Laplacian pyramid

Can we invert the Laplacian Pyramid?

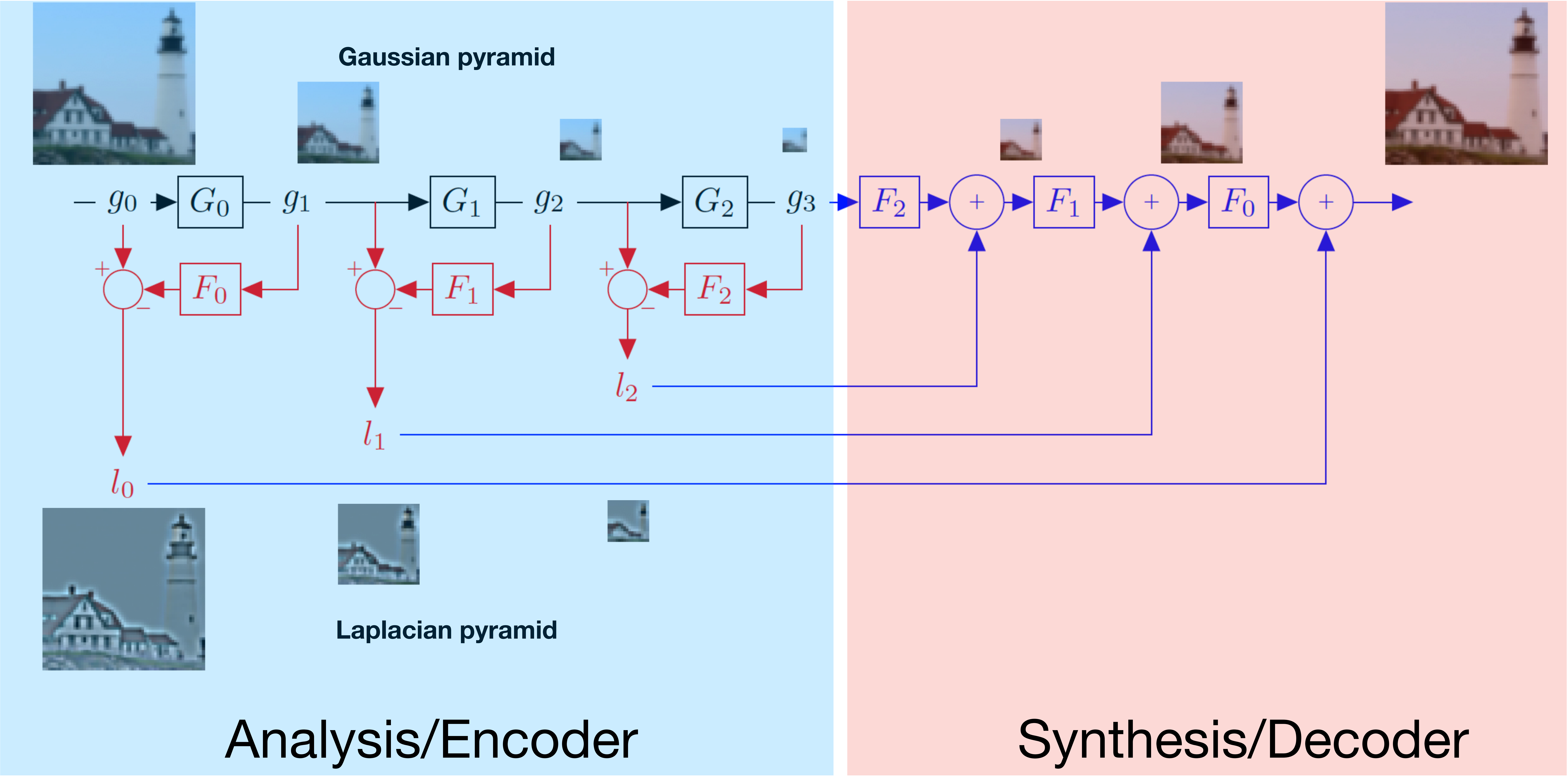
Source: Torralba, Freeman, Isola

The Laplacian Pyramid



Source: Torralba, Freeman, Isola

The Laplacian Pyramid



Source: Torralba, Freeman, Isola

Laplacian pyramid applications

- Texture synthesis
- Image compression
- Noise removal
- Computing image features (e.g., SIFT)

Source: Torralba, Freeman, Isola

Image Blending

Source: Torralba, Freeman, Isola

Image Blending

Source: Torralba, Freeman, Isola

Image Blending

I^A $\quad$ I^B $\quad$ m $\quad$ I

$$I = m * I^A + (1 - m) * I^B$$

Source: Torralba, Freeman, Isola

Image Blending with the Laplacian Pyramid

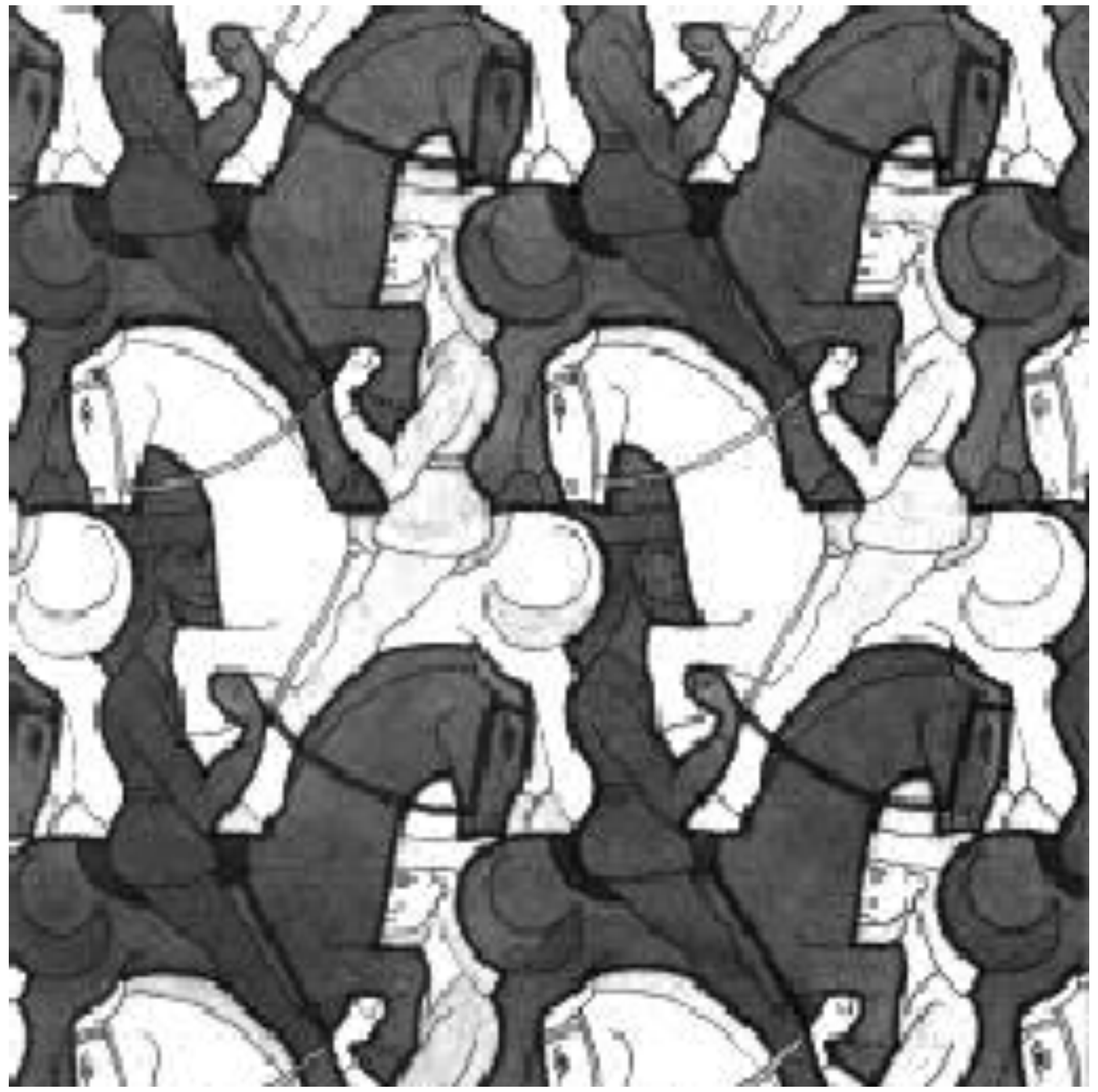
$$l_k = l_k^A * m_k + l_i^B * (1 - m_k)$$

Source: Torralba, Freeman, Isola

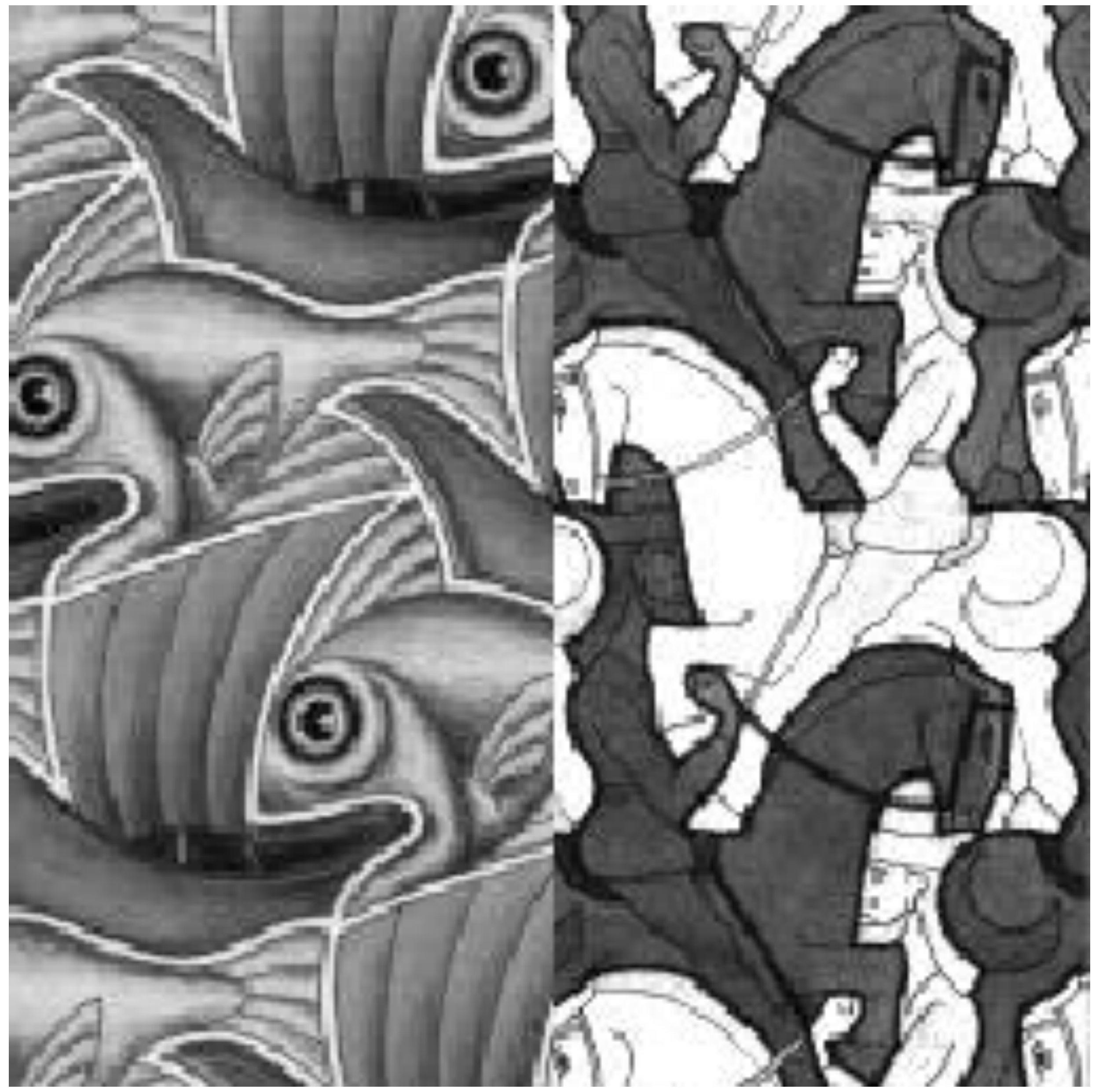
Image Blending with the Laplacian Pyramid

Source: Torralba, Freeman, Isola

+



=



Simple blend

With Laplacian pyr.

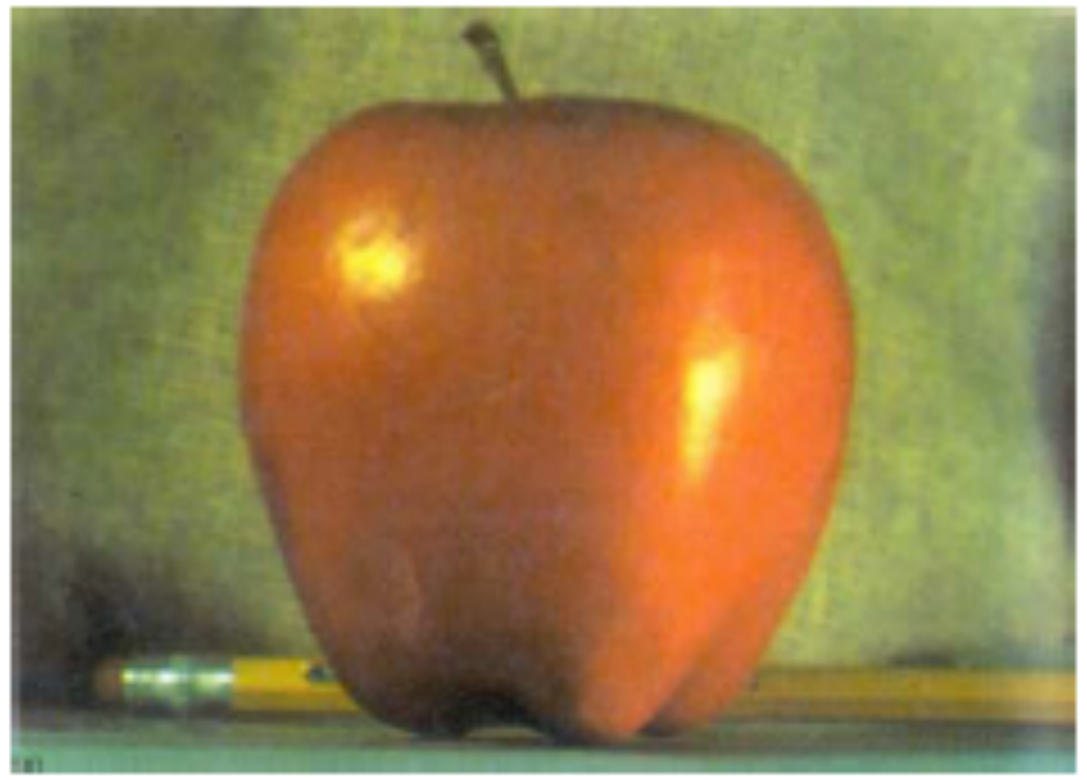
Source: A. Efros

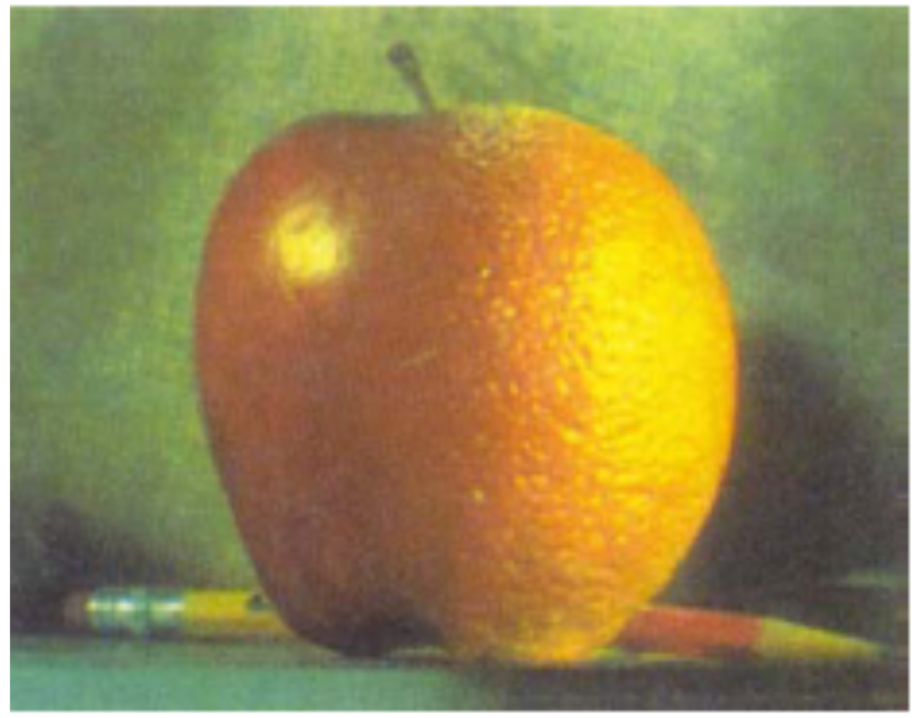
Photo credit: Chris Cameron

Source: A. Efros

Image Blending (PS2 problem)

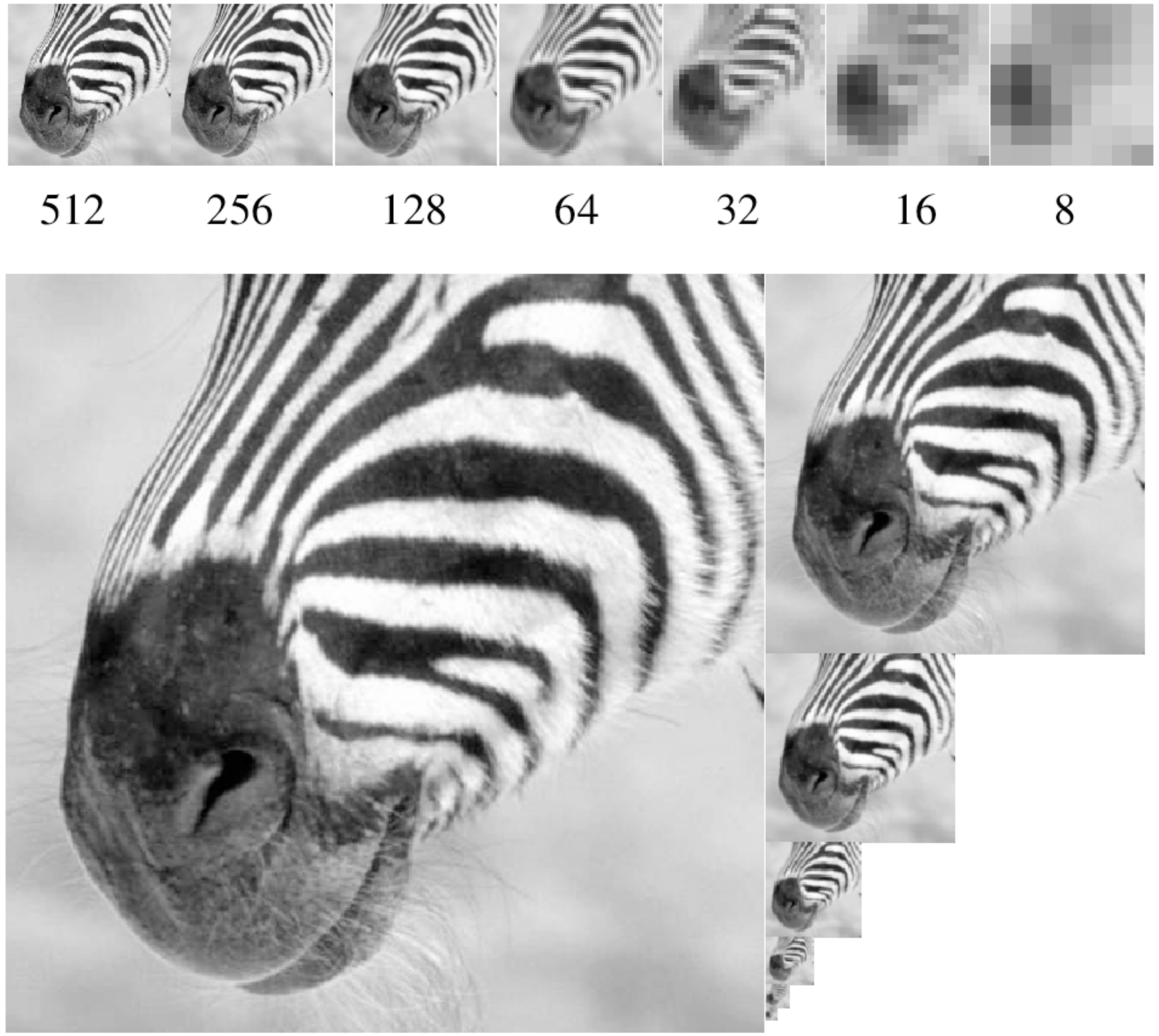
- Build Laplacian pyramid for both images: L_A, L_B
- Build Gaussian pyramid for mask: G
- Build a combined Laplacian pyramid:
- Collapse L to obtain the blended image



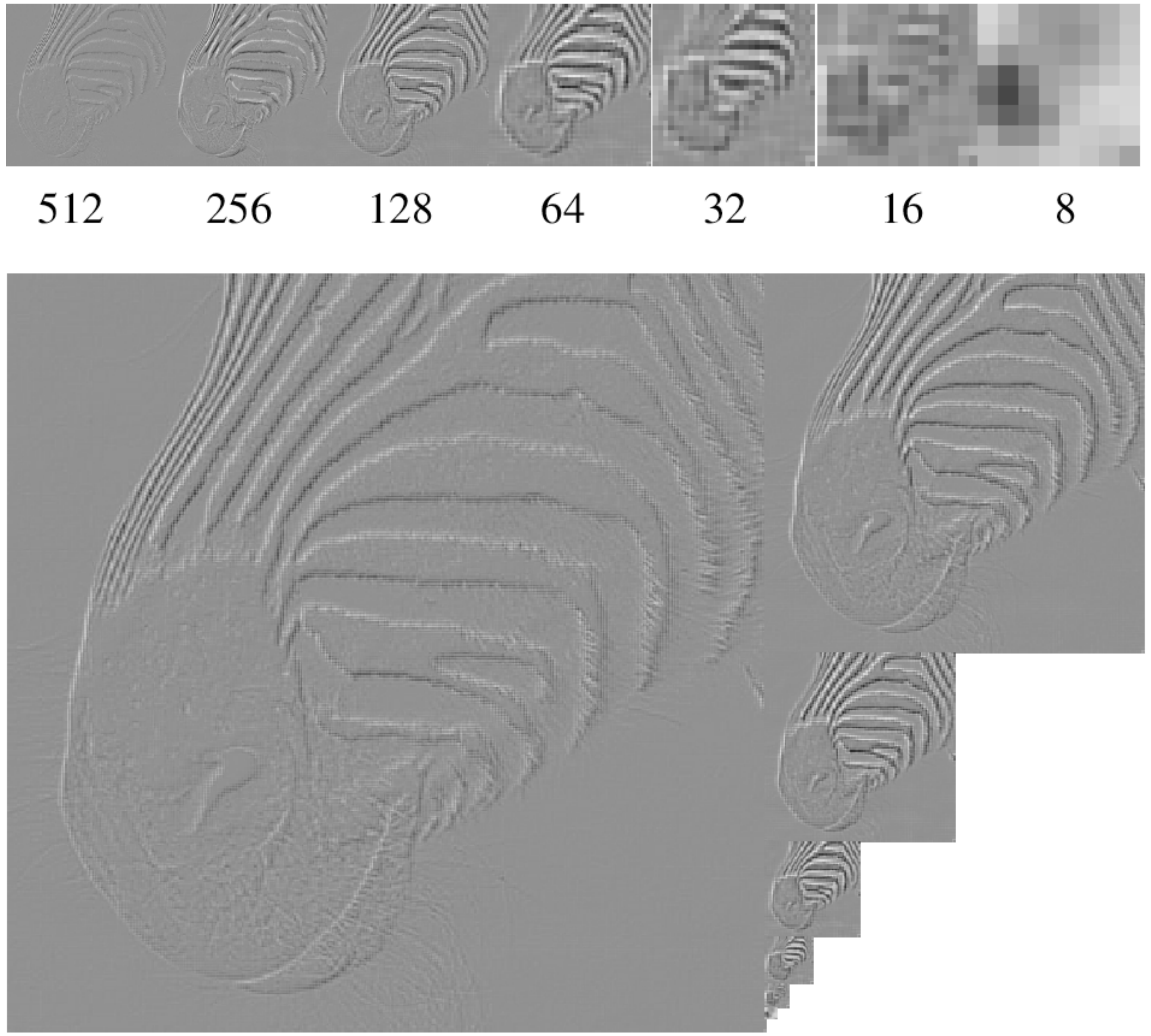


Source: Torralba, Freeman, Isola

Image pyramids



Gaussian Pyr.



Laplacian Pyr.

And many more: steerable filters, wavelets, … convolutional networks!

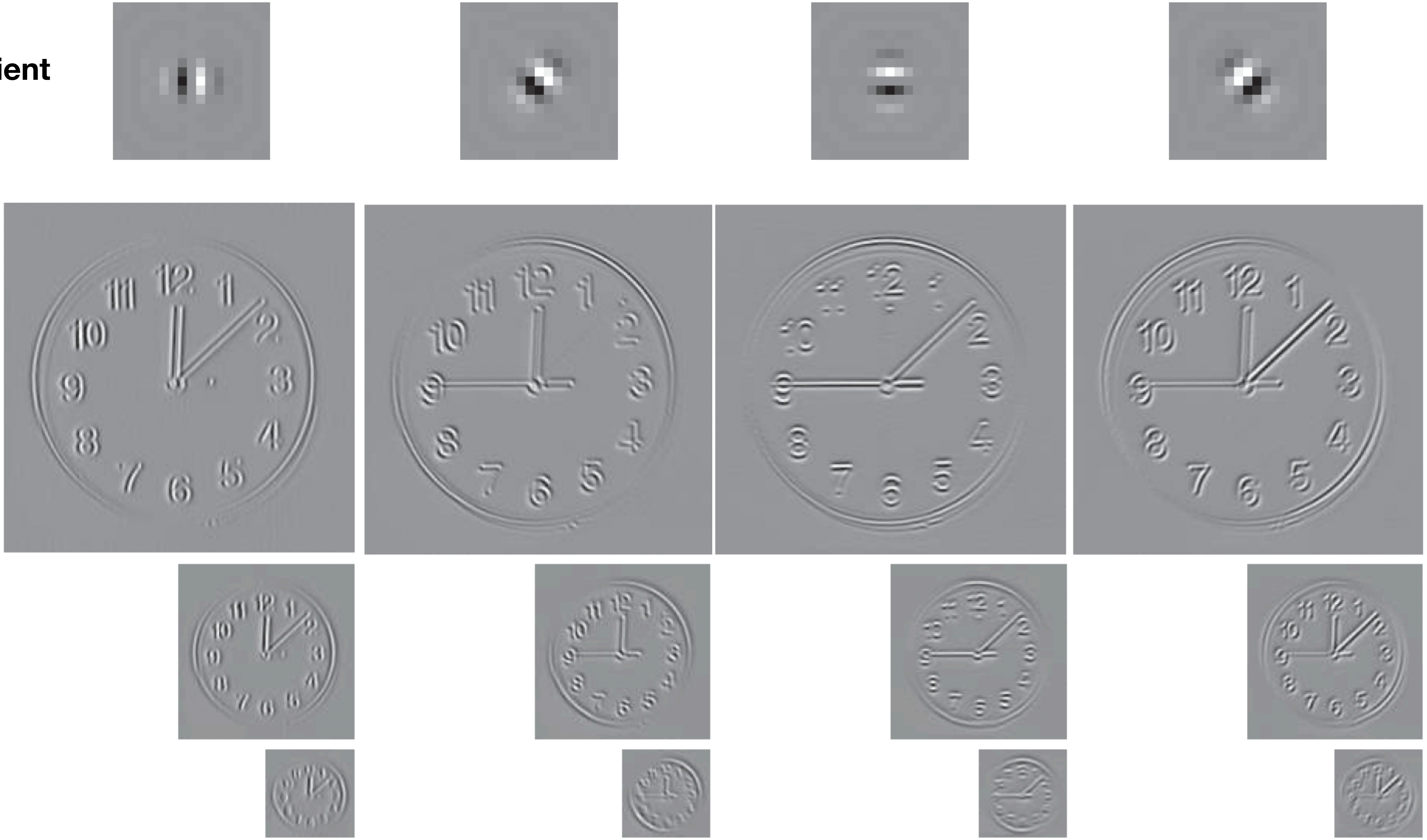
Source: Torralba, Freeman, Isola

Orientations

Source: Torralba, Freeman, Isola

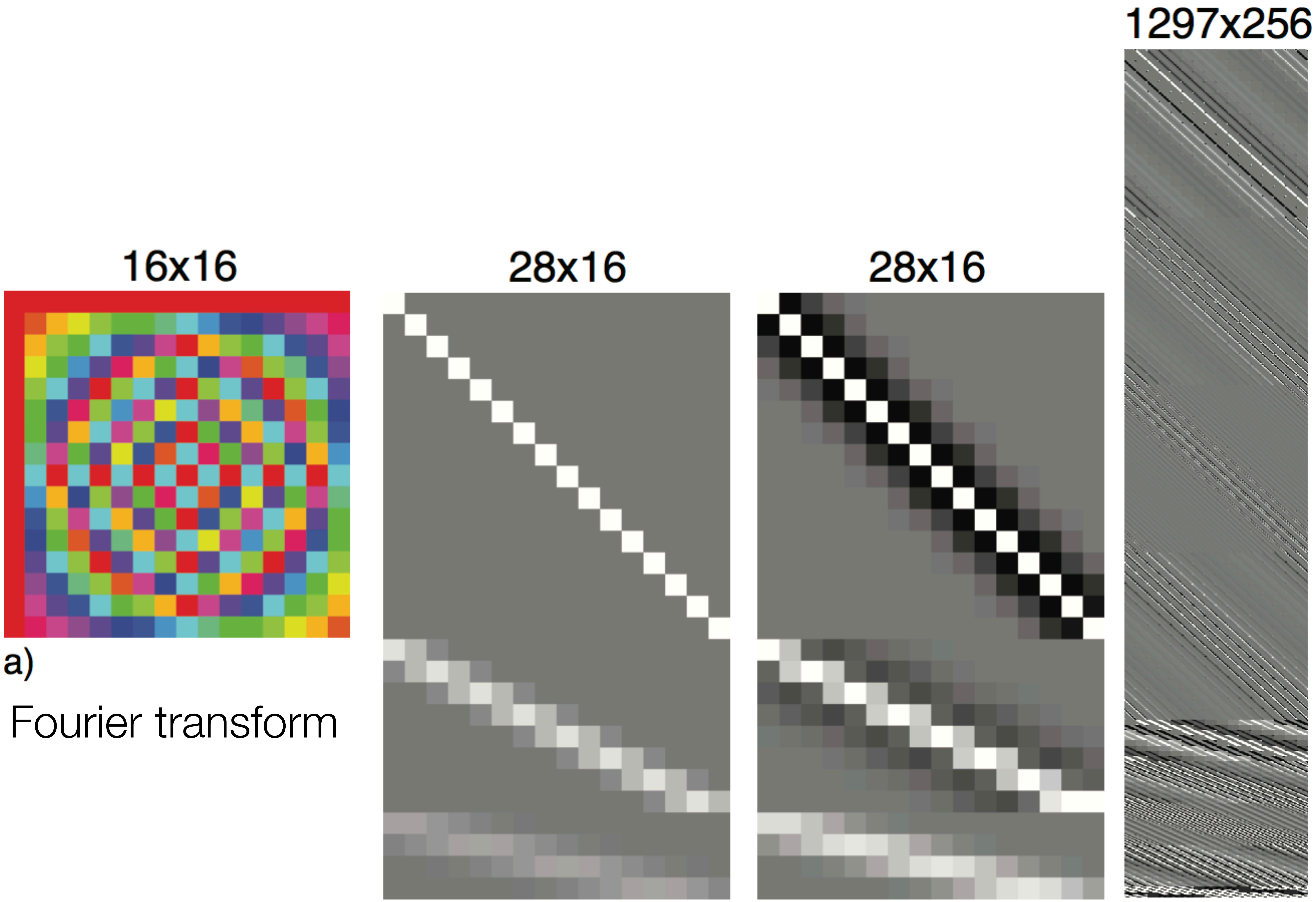
Steerable Pyramid

Oriented gradient



Source: Torralba, Freeman, Isola

Linear Image Transforms



Fourier transform

Gaussian pyr. Laplacian pyr. Steerable pyr.

Source: Torralba, Freeman, Isola

Source: Torralba, Freeman, Isola

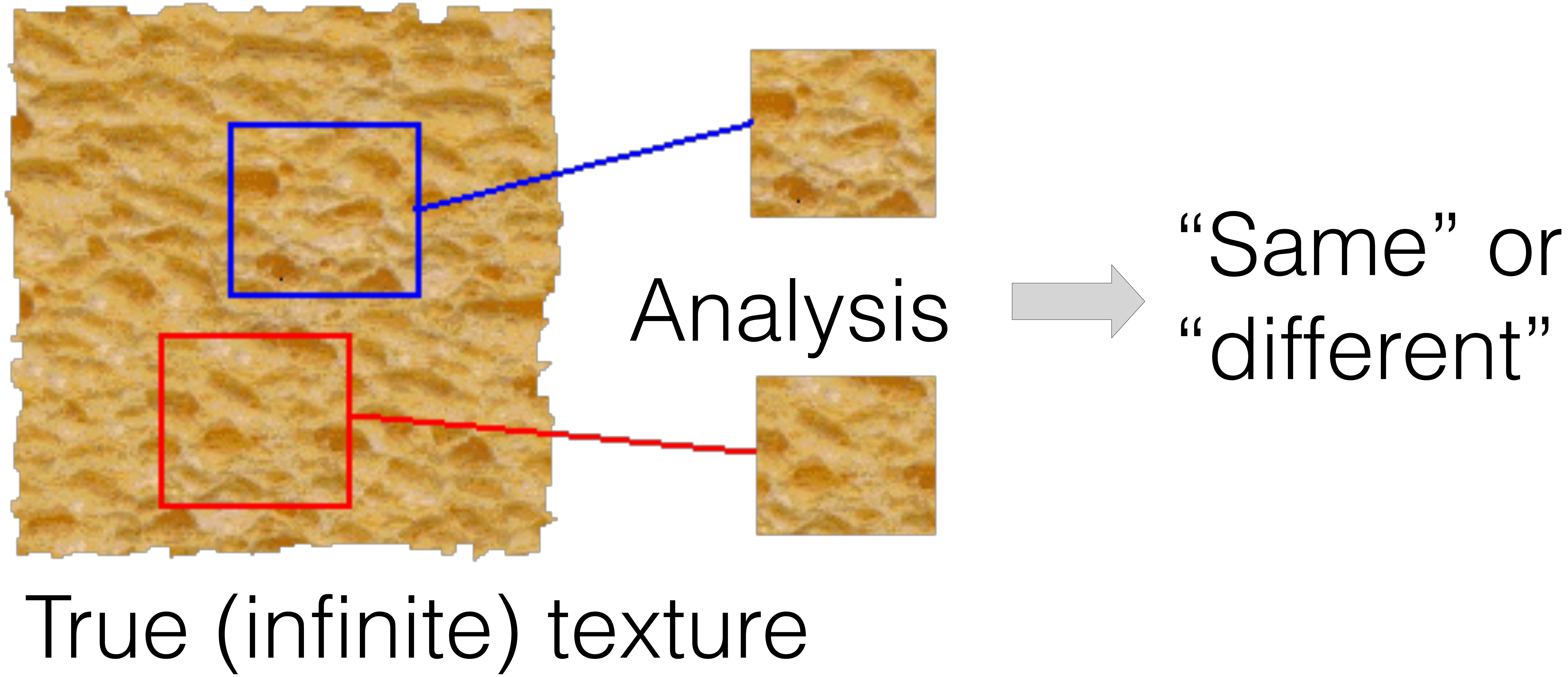
Texture

Stationary

Stochastic

Source: Torralba, Freeman, Isola

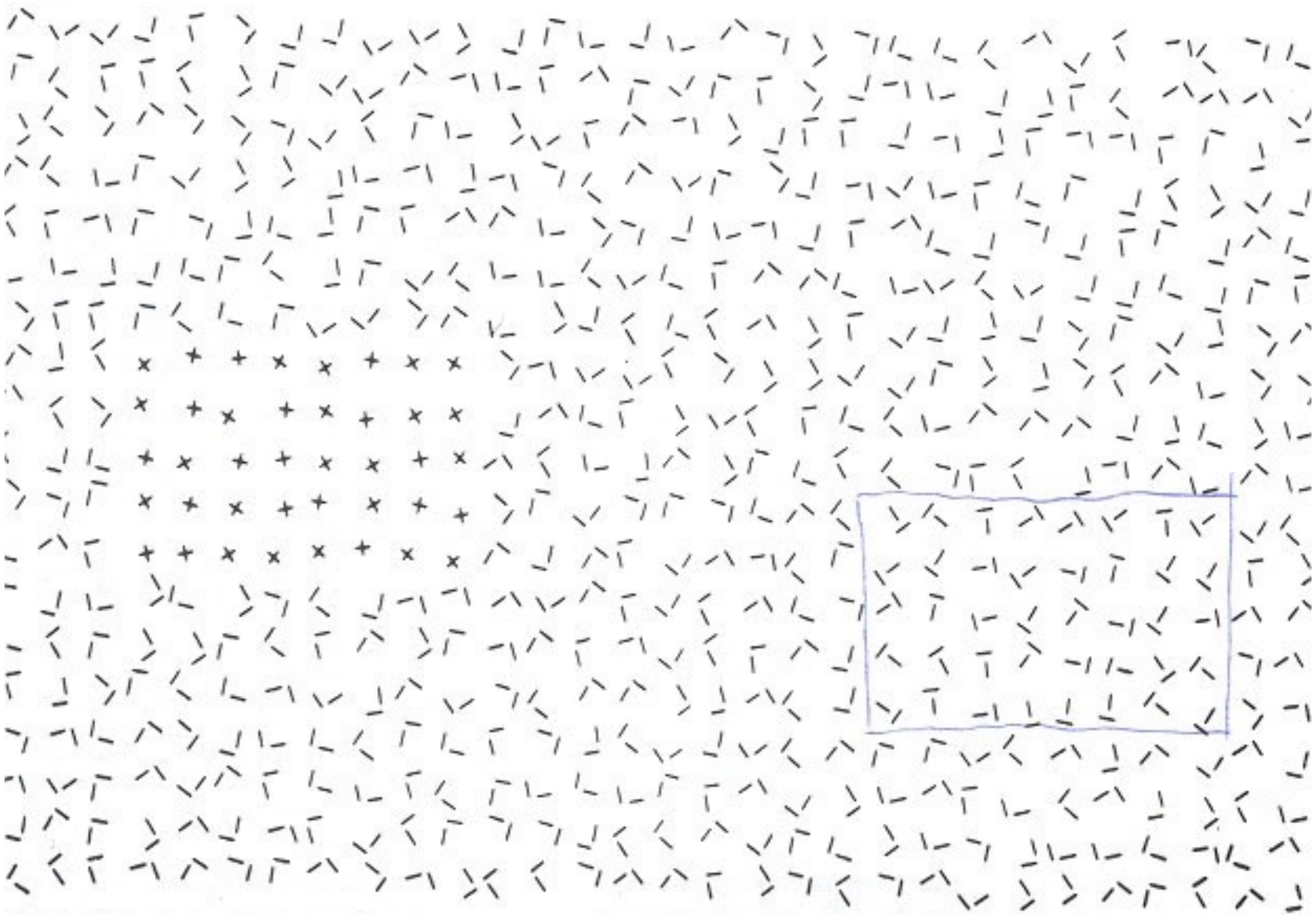
Texture analysis

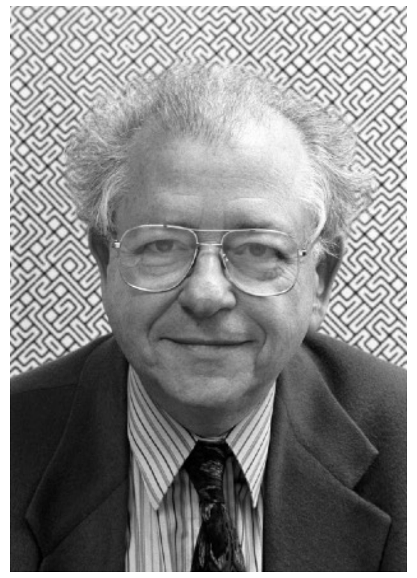


What we'd like: are they made of the same "stuff". Are these textures similar?

Source: A. Efros

How do humans analyze texture?





Human vision is sensitive to the difference of some types of elements and appears to be “numb” on other types of differences.

Béla Julesz

Source: A. Efros

Pre-attentive texture discrimination

Bela Julesz, "Textons, the Elements of Texture Perception, and their Interactions". Nature 290: 91-97. March, 1981.

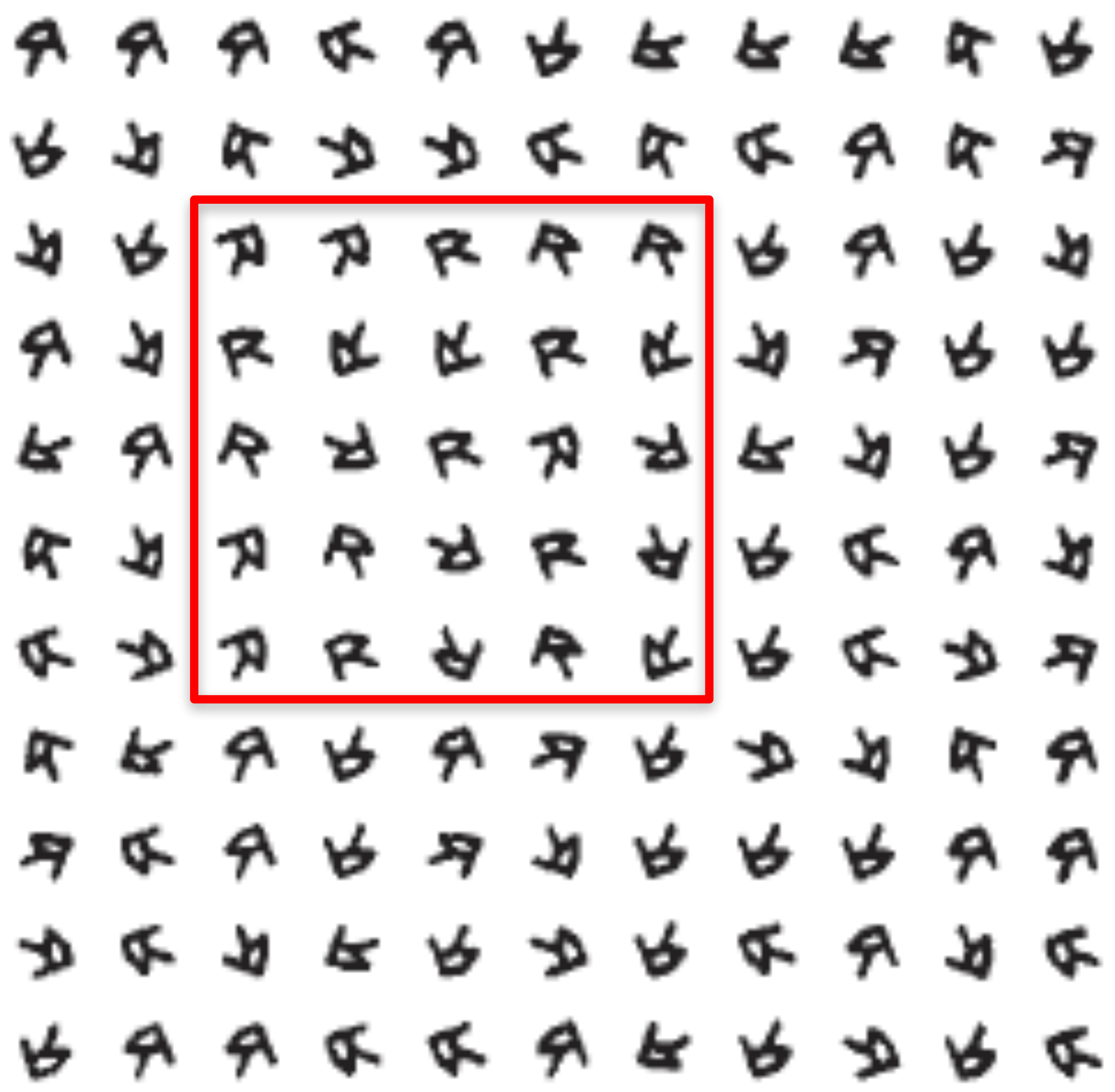
Source: Torralba, Freeman, Isola

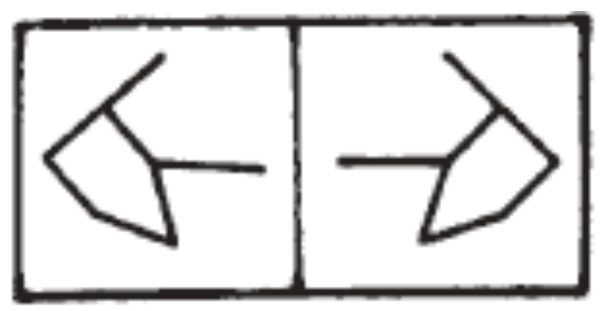
Pre-attentive texture discrimination

Bela Julesz, "Textons, the Elements of Texture Perception, and their Interactions". Nature 290: 91-97. March, 1981.

Source: Torralba, Freeman, Isola

Pre-attentive texture discrimination



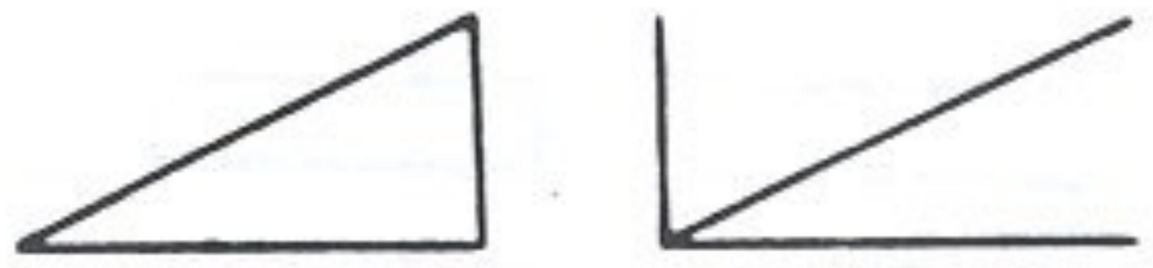


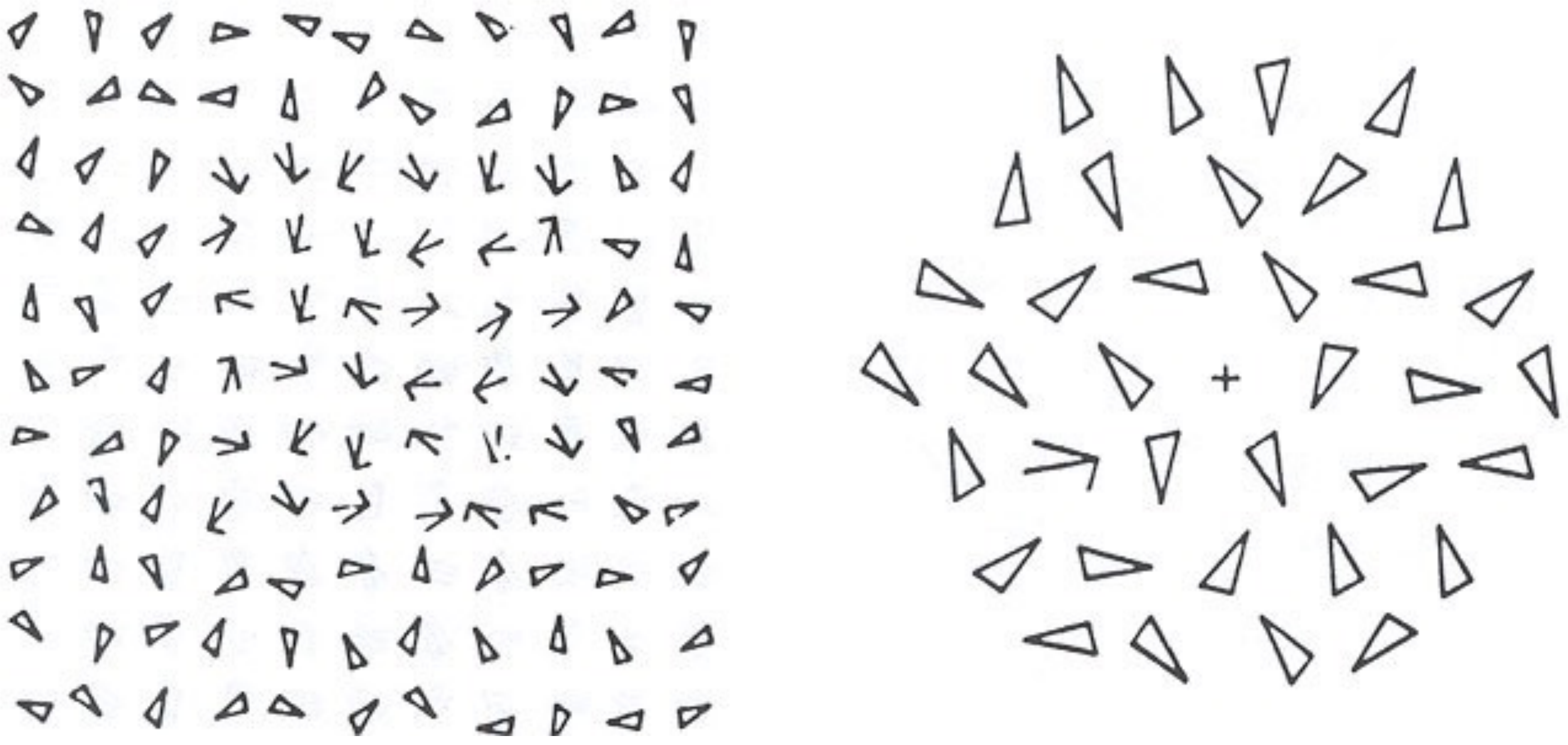
This texture pair is pre-attentively indistinguishable. Why?

Bela Julesz, "Textons, the Elements of Texture Perception, and their Interactions". Nature 290: 91-97. March, 1981.

Source: A. Efros

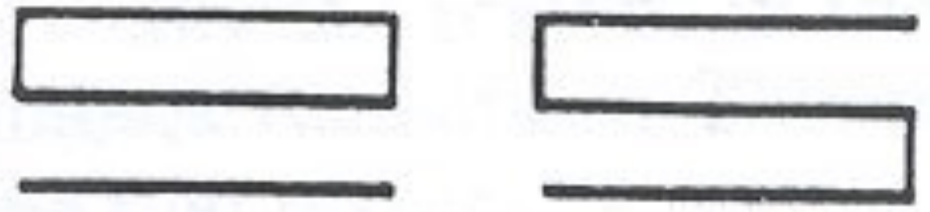
Search Experiment I

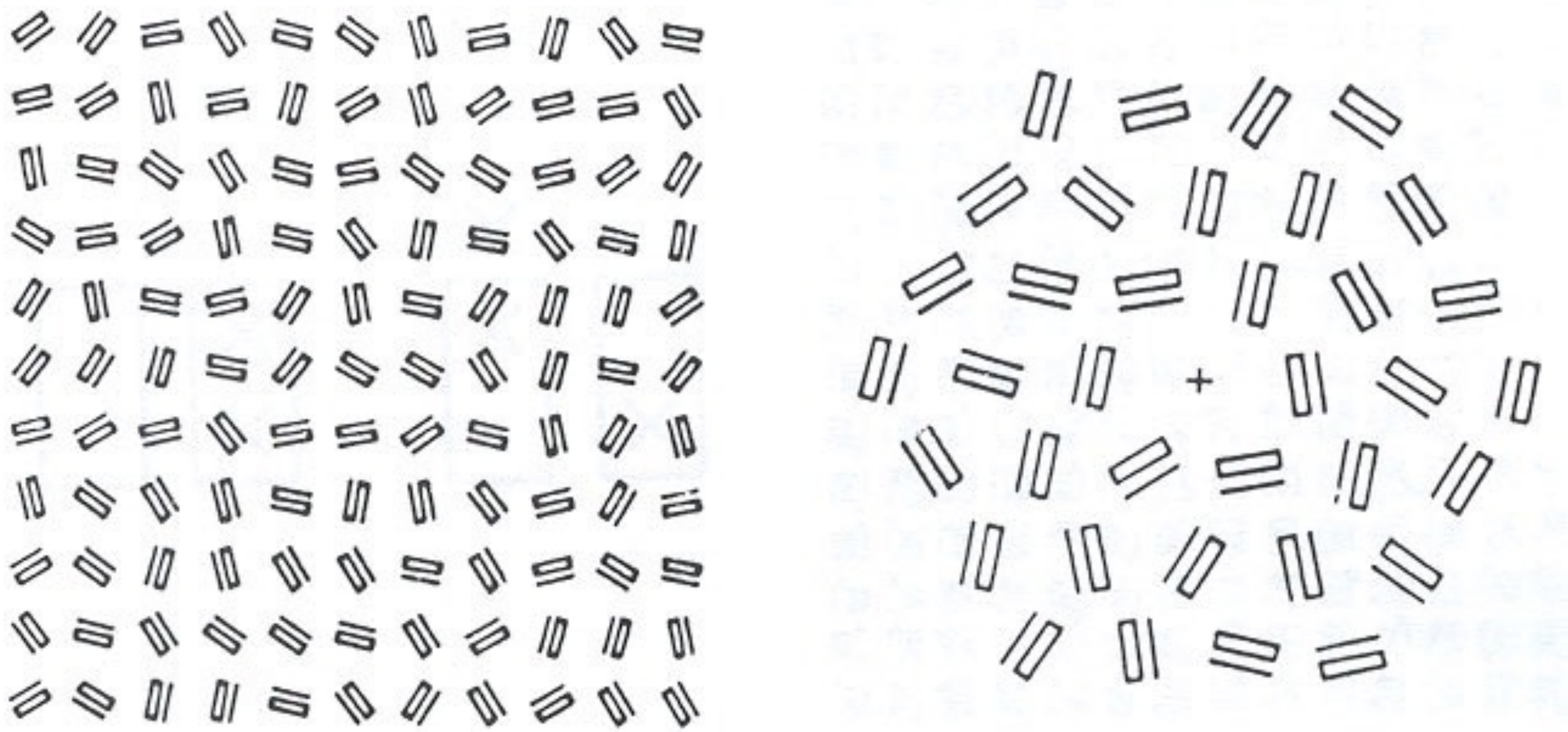

(a)



The subject is told to detect a target element in a number of background elements.
In this example, the detection time is independent of the number of background elements.

Search Experiment II


(a)



Here detection time is proportional to the number of background elements, and thus suggests that the subject is doing element-by-element scrutiny.

Heuristic

Julesz then conjectured the following:

Human vision operates in two distinct modes:

1. Preattentive vision

   parallel, instantaneous *(~100--200ms),* without scrutiny,
   independent of the number of patterns, covering a large visual field.

2. Attentive vision

   serial search by focal attention in 50ms steps limited to small aperture.

Source: A. Efros

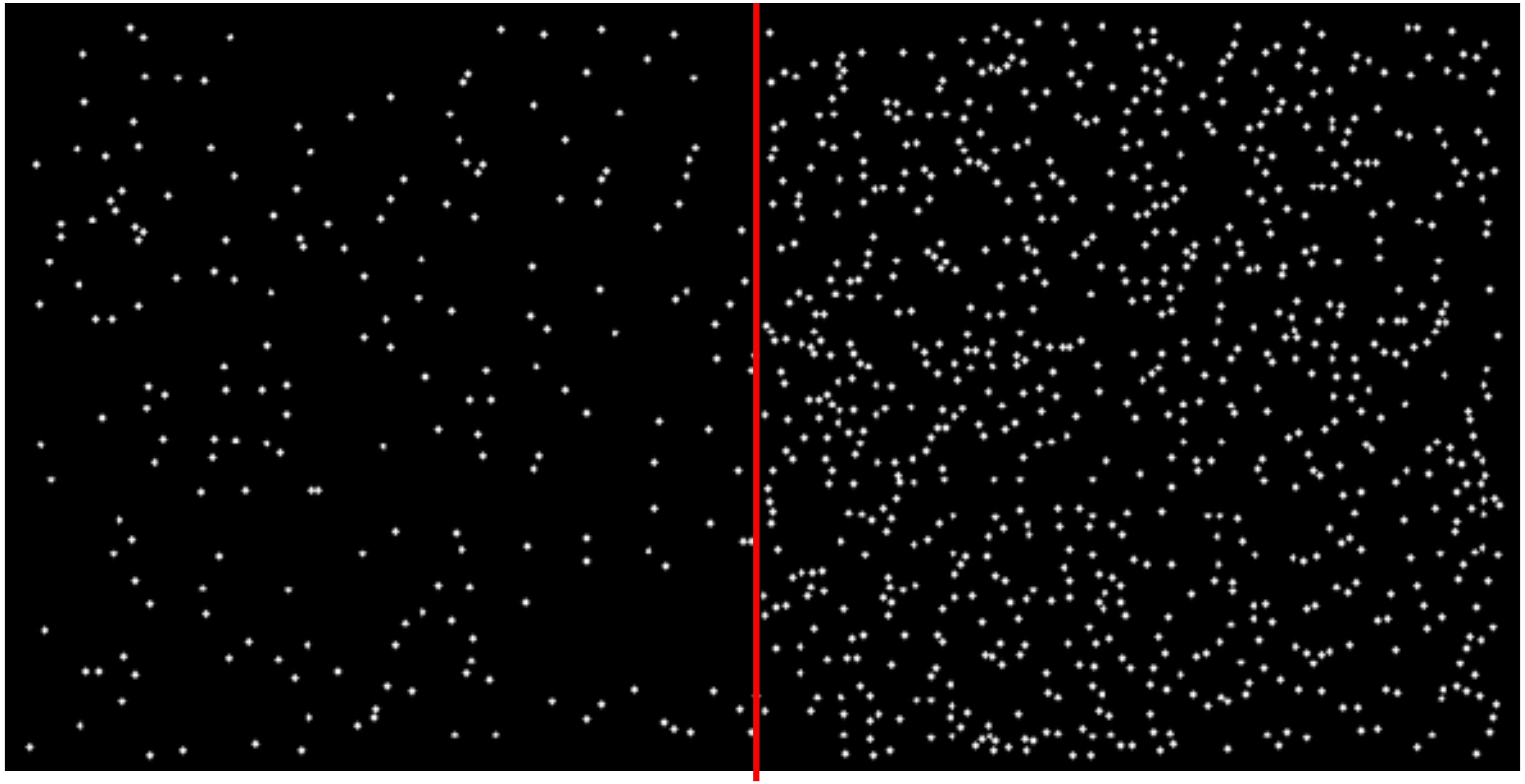
Julesz Conjecture

Textures cannot be spontaneously discriminated if they have the same first-order and second-order statistics and differ only in their third-order or higher-order statistics.

(later proved wrong)

Source: A. Efros

1^{st} order statistics differ

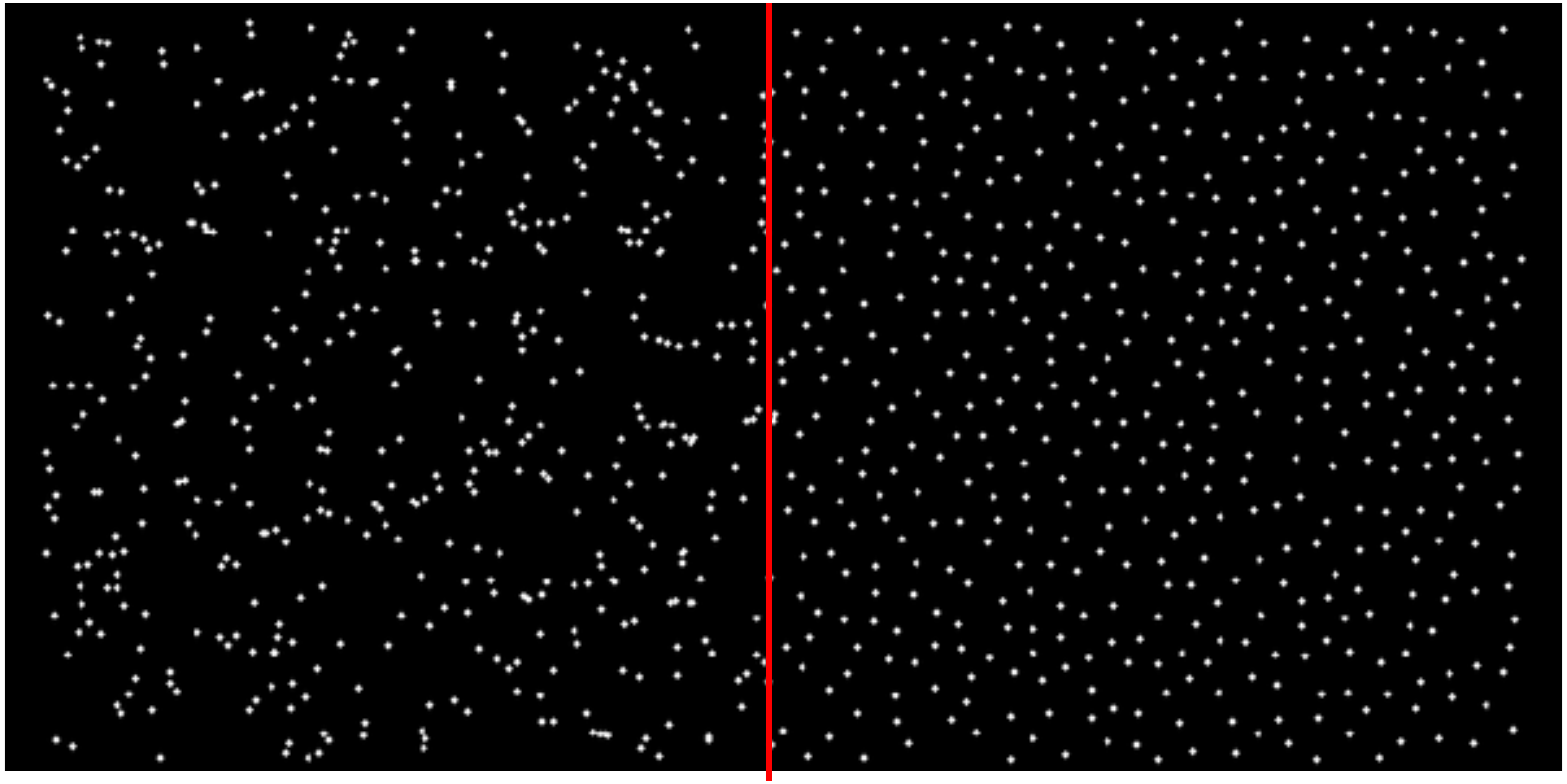


5% white

20% white

Source: A. Efros

2nd order statistics differ

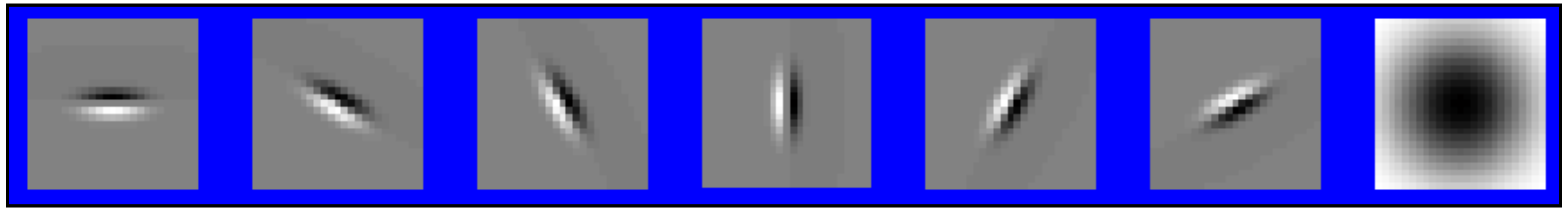


10% white

Source: A. Efros

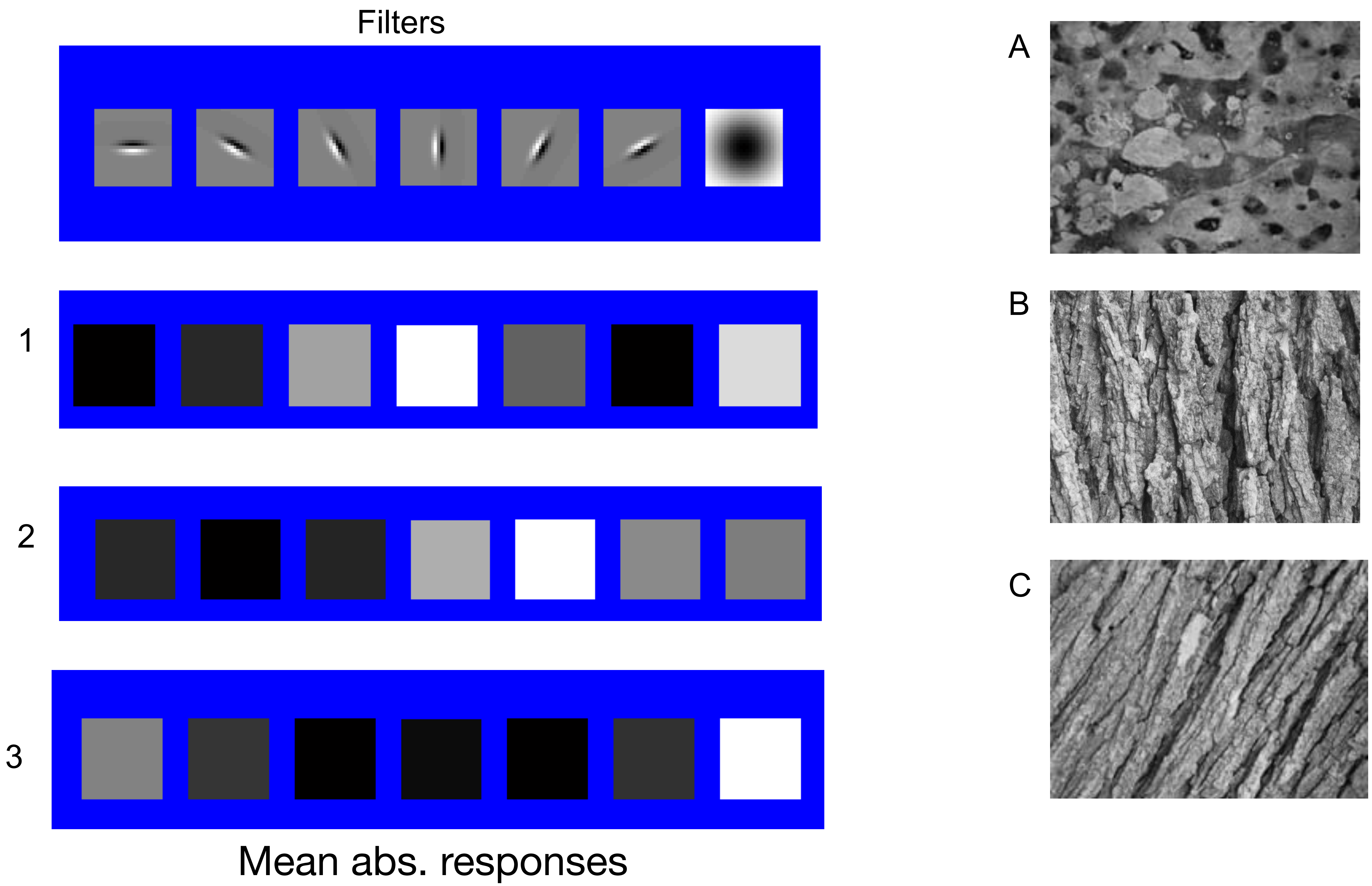
How can we represent texture in natural images?

- Idea 1: Record simple statistics (e.g., mean, std.) of absolute filter responses



Source: A. Efros

Can you match the texture to the response?



Source: A. Efros

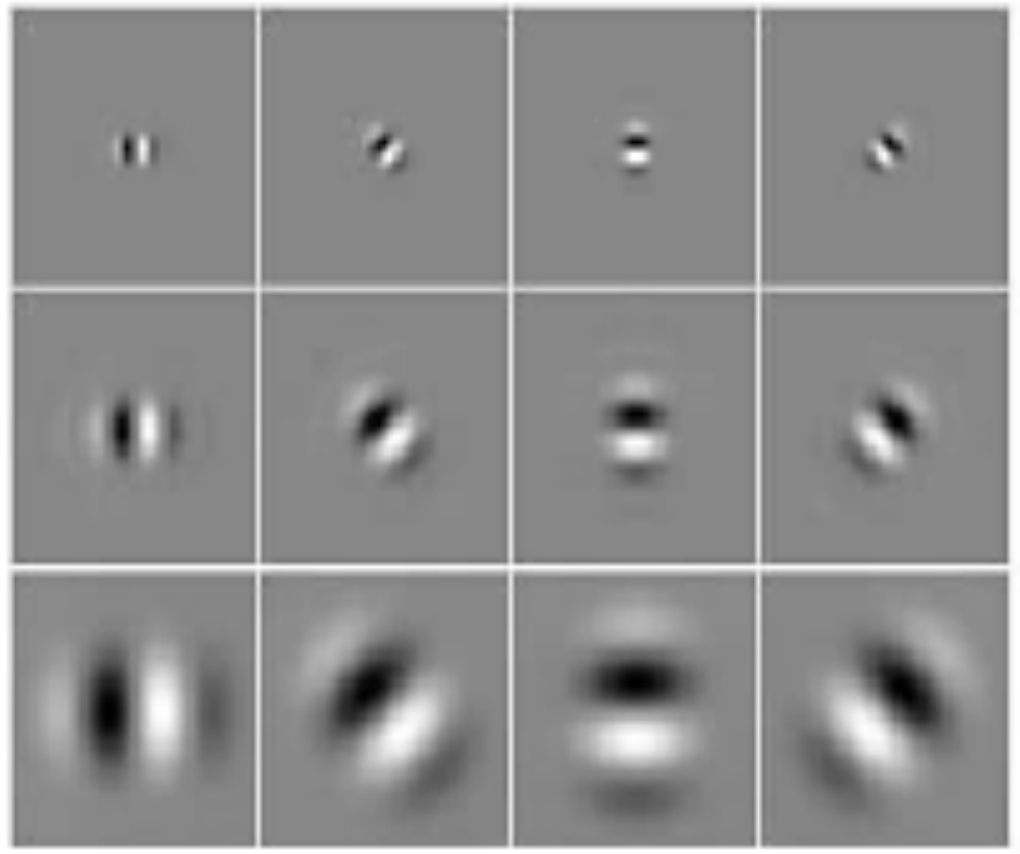
How can we represent texture in natural images?

- Generalize this to “orientation histogram”
- Idea 2: Marginal histograms of filter responses
    - One histogram per filter

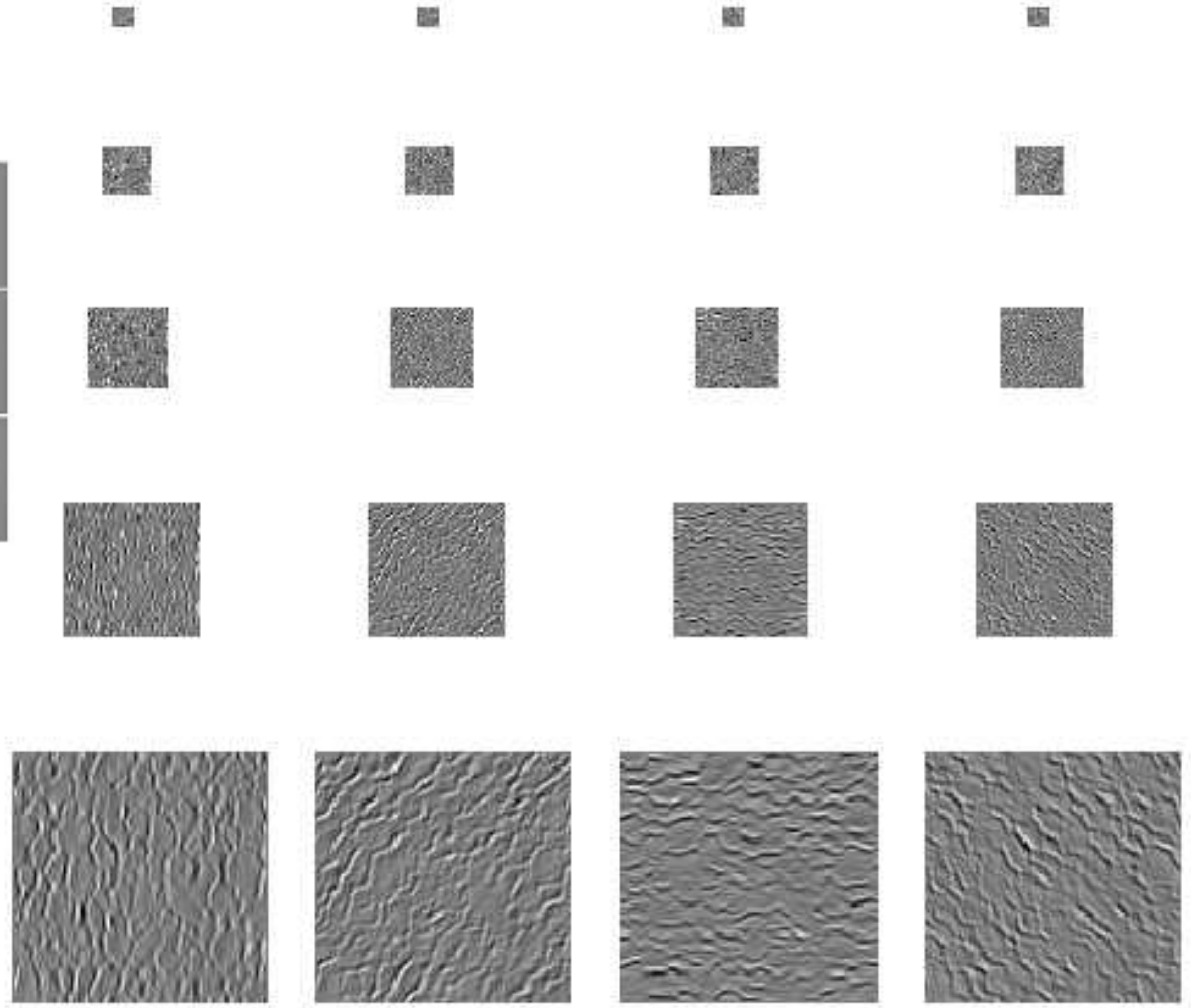
Source: A. Efros

Steerable filter decomposition

Filter bank

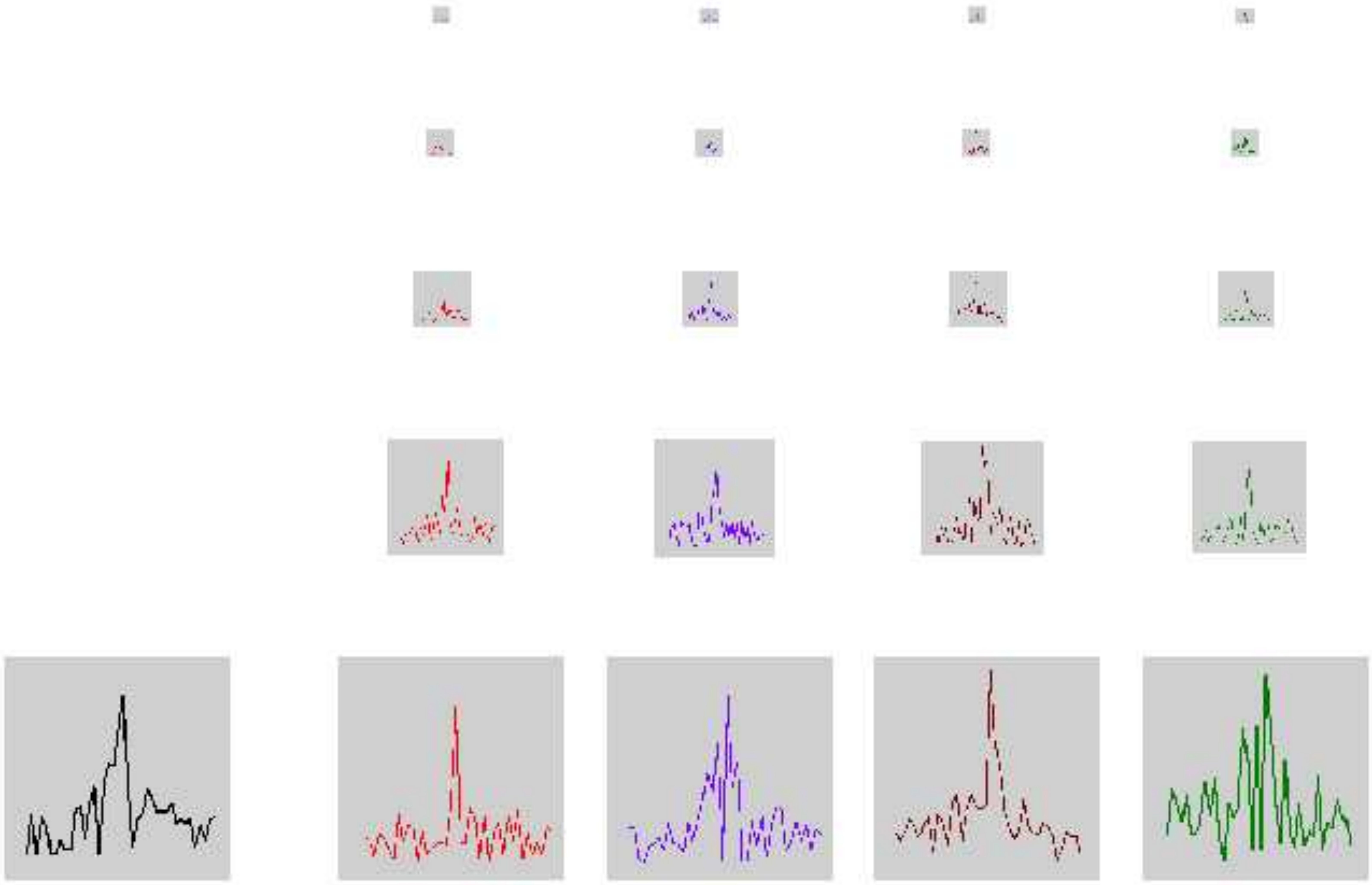


Input image



Source: A. Efros

Filter response histograms



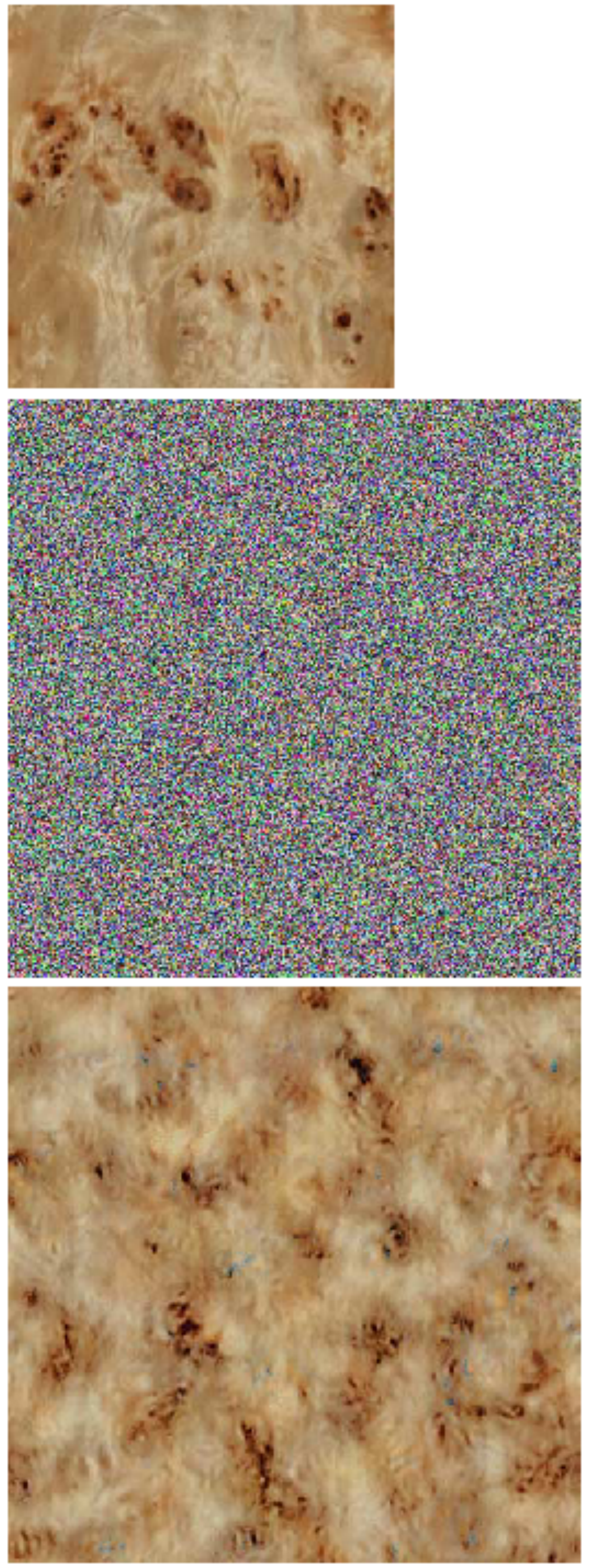
Source: A. Efros

Texture synthesis

Start with a noise image as output.

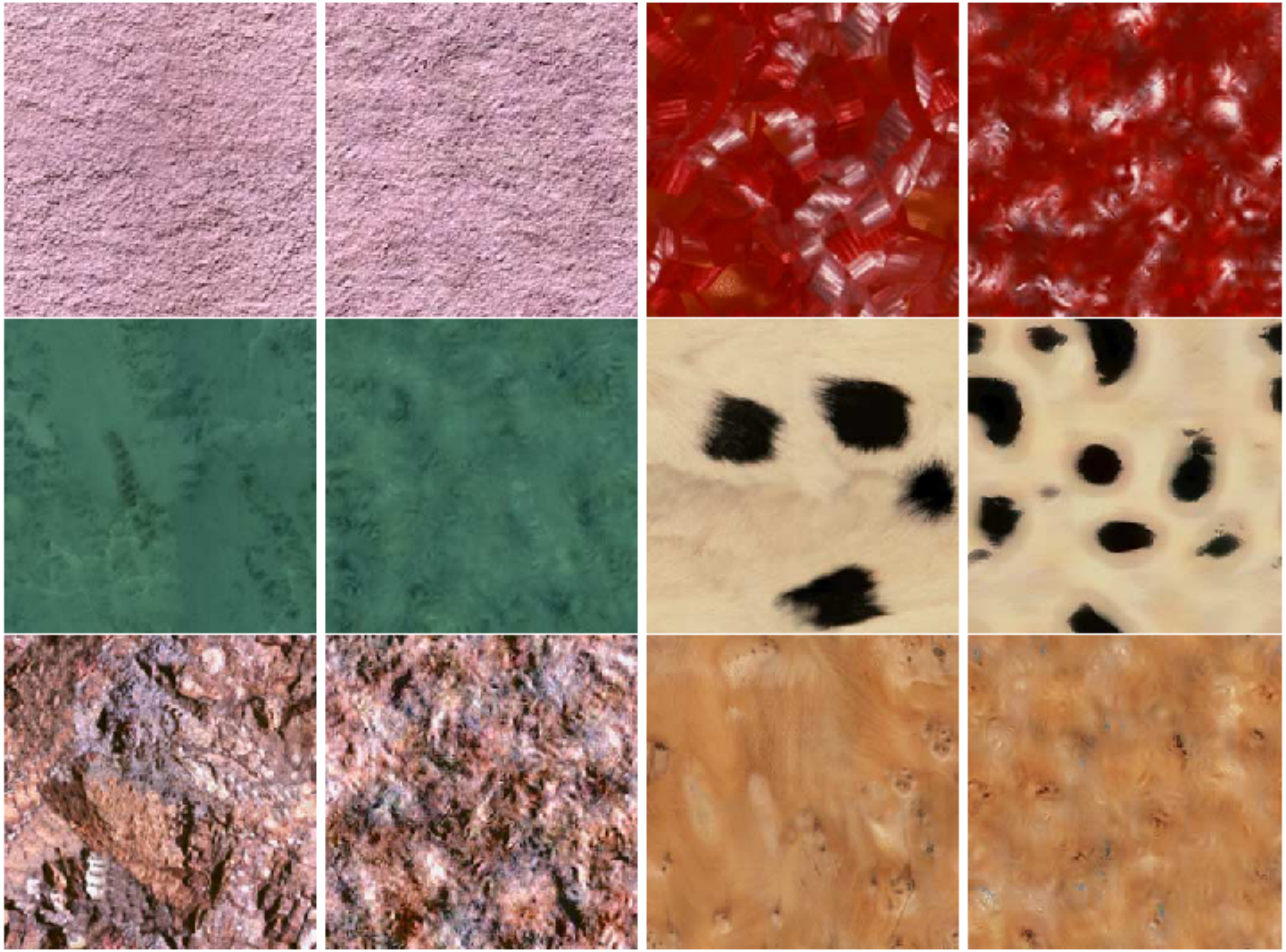
Main loop:

- Match pixel histogram of output image to input
- Decompose input/output images using a Steerable Pyramid
- Match subband histograms of input and output pyramids
- Reconstruct input and output images (collapse the pyramids)



Heeger, Bergen, Pyramid-based texture analysis/synthesis, SIGGRAPH 1995

Source: A. Efros



Source: A. Efros

Source: A. Efros

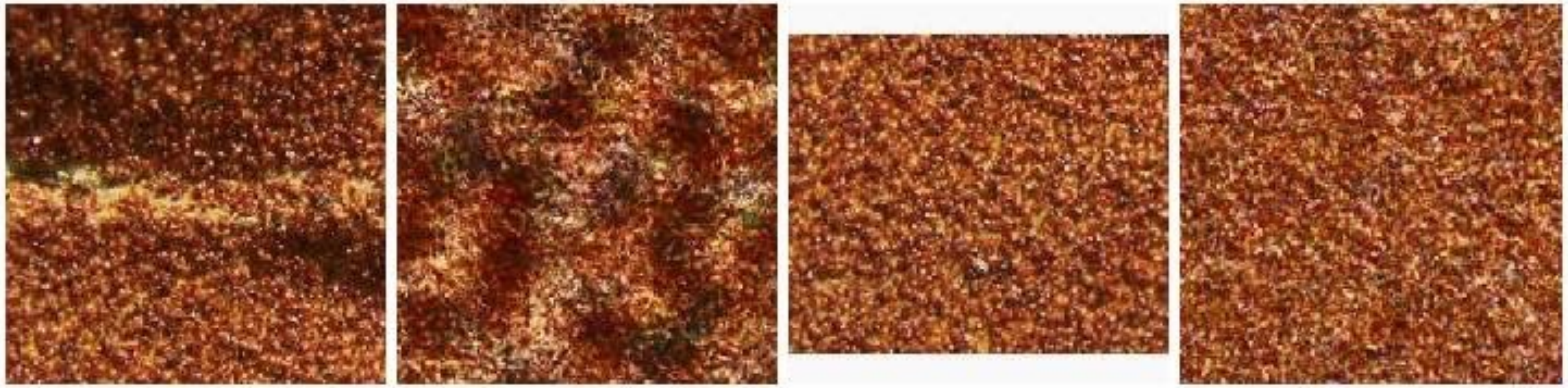


Figure 7: (Left pair) Inhomogeneous input texture produces blotchy synthetic texture. (Right pair) Homogenous input.

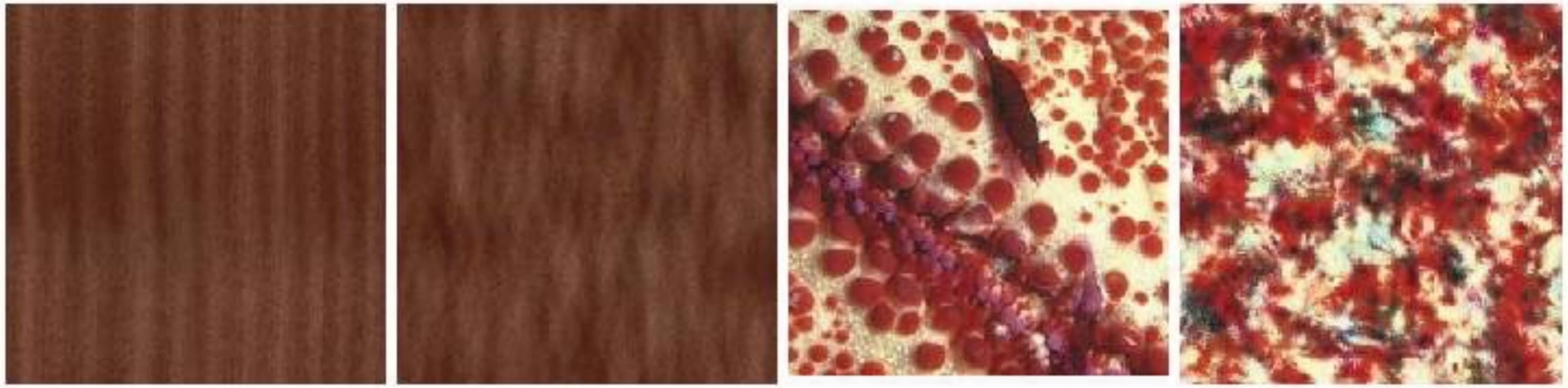


Figure 8: Examples of failures: wood grain and red coral.

Figure 9: More failures: hay and marble.

Source: A. Efros