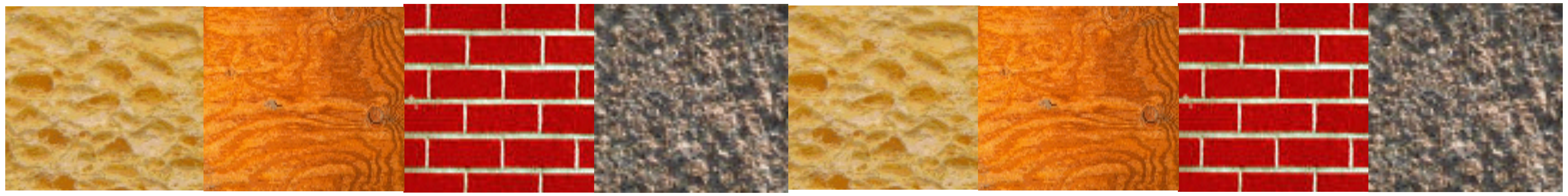
THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision



Lecture 11: Texture

(unless otherwise stated slides are taken or adopted from **Bob Woodham, Jim Little** and **Fred Tung**)

Menu for Today

Topics:

- Texture **Analysis, Synthesis**
- Filter Banks, Data-driven Methods
- **Quiz 3**
- Midterm practice questions

Readings:

- **Today's** Lecture: Forsyth & Ponce (2nd ed.) 3.1-3.3

Reminders:

- **Midterm** is next week **October 19th at 5pm**
- **Assignment 3**: Texture Synthesis is available (**due October 26th**)

Menu for Week Ahead

Reminders:

- — **No lecture** this Thursday **October 12th** (UBC Make Up Monday)
- — **No lecture** on Tuesday **October 17th** (preparation time for midterm)
- — **Midterm** is next Thursday **October 19th at 5pm**, 75 mins, closed book
- — Midterm example question are on Canvas (scroll to bottom of main page)
- — Office hours continue as usual, please make use of them!

Today’s “**fun**” Example: Texture Camouflage

https://en.wikipedia.org/wiki/File:Camouflage.jpg

Today’s “**fun**” Example: Texture Camouflage

Cuttlefish on gravel seabed

Seconds later…

http://www.marinet.org.uk/campaign- article/an-illustrated-guide-to-uk-marine-animals

Texture

What is **texture**?

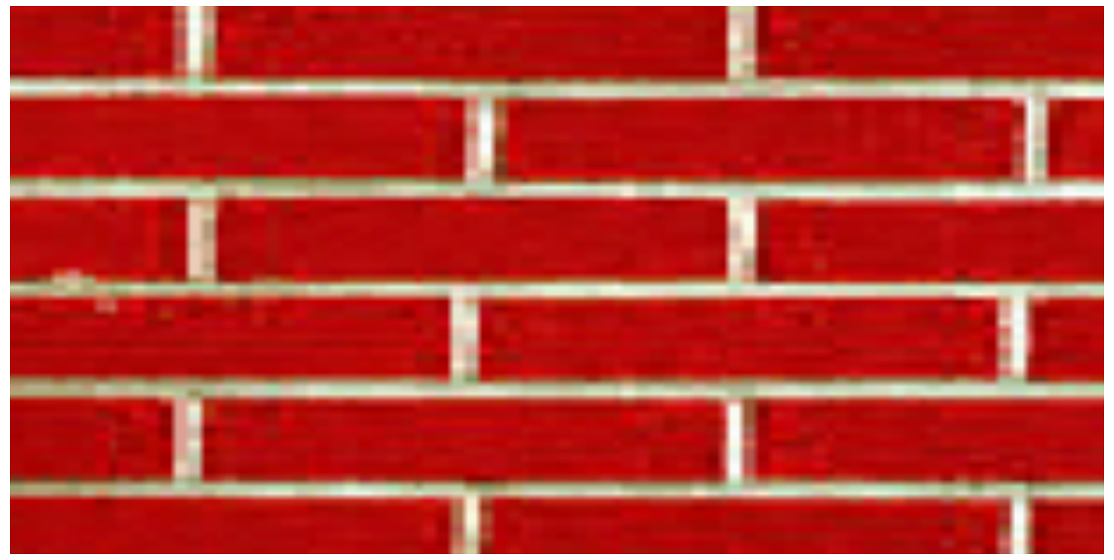
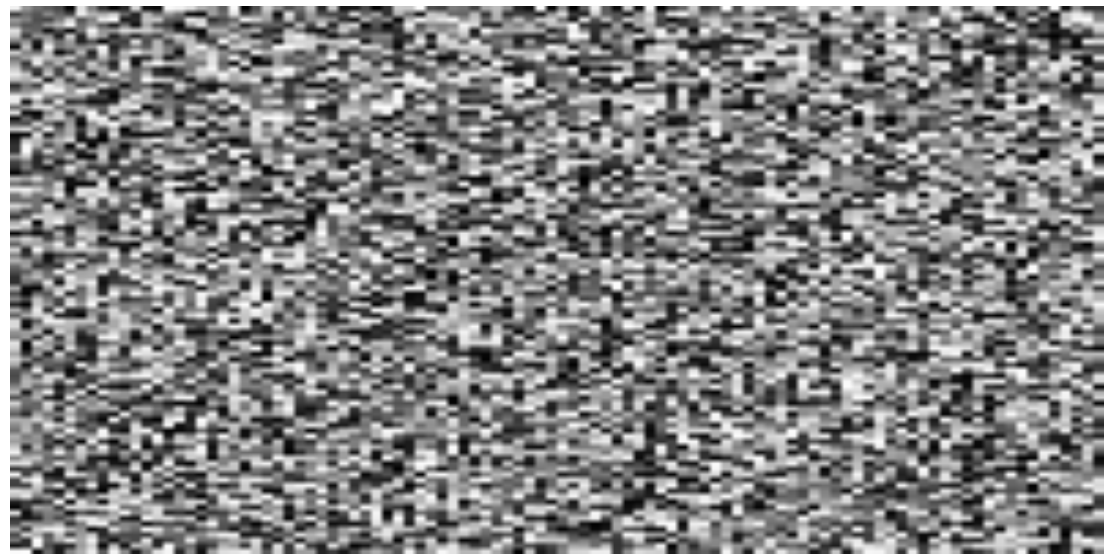




Figure Credit: Alexei Efros and Thomas Leung

Texture is widespread, easy to recognize, but hard to define

Views of large numbers of small objects are often considered textures
— e.g. grass, foliage, pebbles, hair

Patterned surface markings are considered textures
— e.g. patterns on wood

Definition of **Texture**

(Functional) **Definition**:

Texture is detail in an image that is at a scale too small to be resolved into its constituent elements and at a scale large enough to be apparent in the spatial distribution of image measurements

Sometimes, textures are thought of as patterns composed of repeated instances of one (or more) identifiable elements, called **textons**.

— e.g. bricks in a wall, spots on a cheetah

Uses of **Texture**

Texture can be a strong cue to **object identity** if the object has distinctive material properties

Texture can be a strong cue to an **object's shape** based on the deformation of the texture from point to point.
— Estimating surface orientation or shape from texture is known as "**shape from texture**"

1. How do we represent/recognise texture? → Texture **analysis**
2. How do we generate new examples of a texture? → Texture **synthesis**

Texture **Representation**

Observation: Textures are made up of generic sub-elements, repeated over a region with similar statistical properties

Idea: Find the sub-elements with filters, then represent each point in the image with a summary of the pattern of sub-elements in the local region

Question: What filters should we use?

Answer: Human vision suggests spots and oriented edge filters at a variety of different orientations and scales

Question: How do we "summarize"?

Answer: Compute the mean or maximum of each filter response over the region
— Other statistics can also be useful

Texture **Representation:** Filter Bank

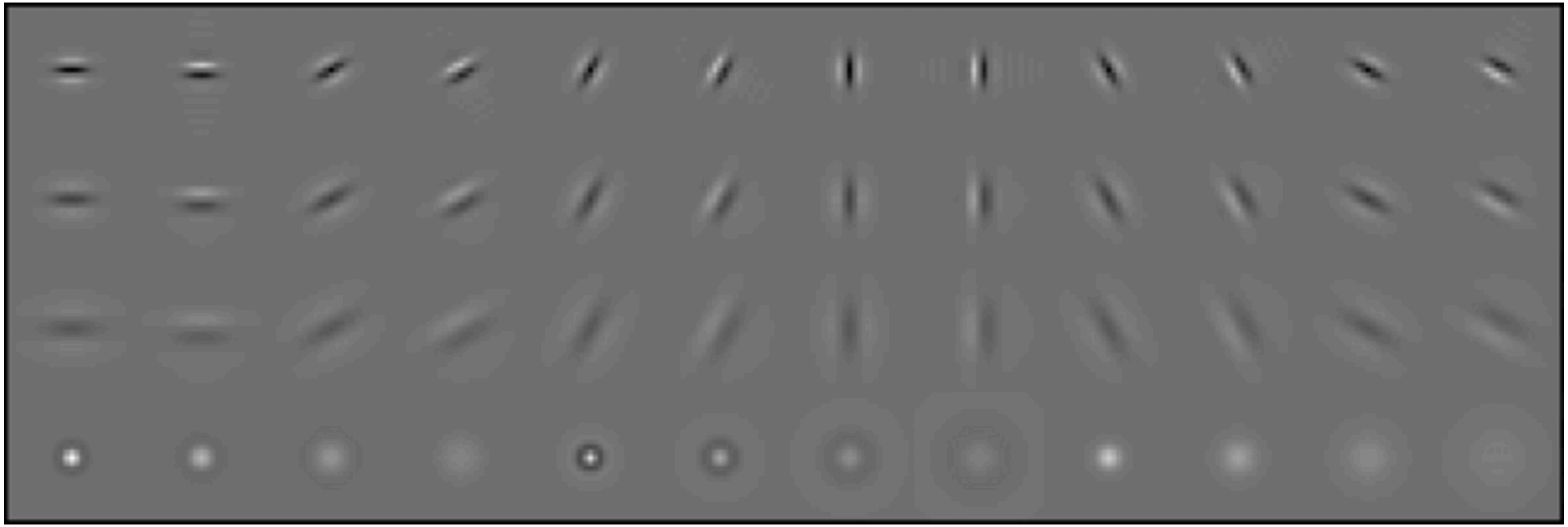
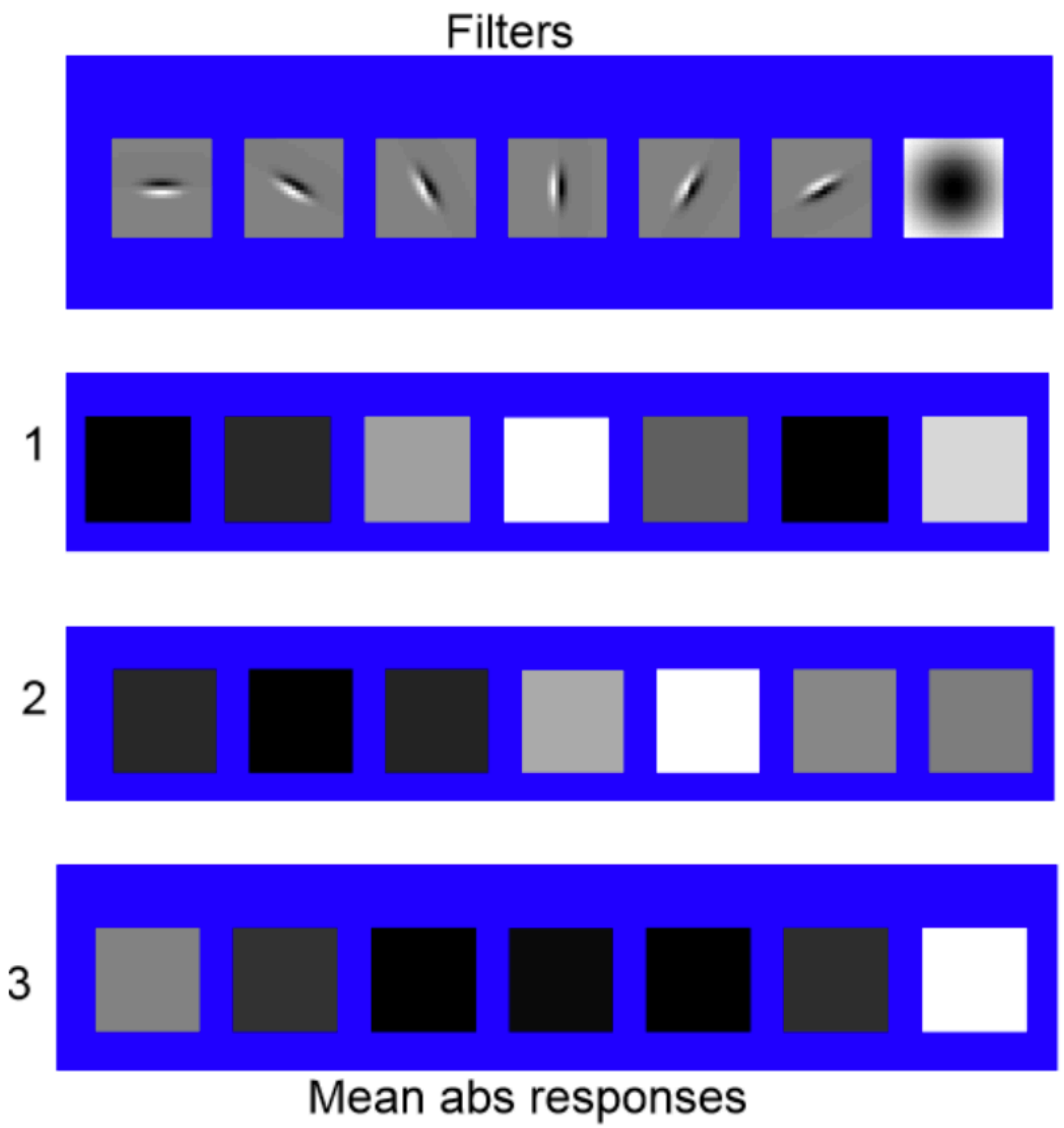
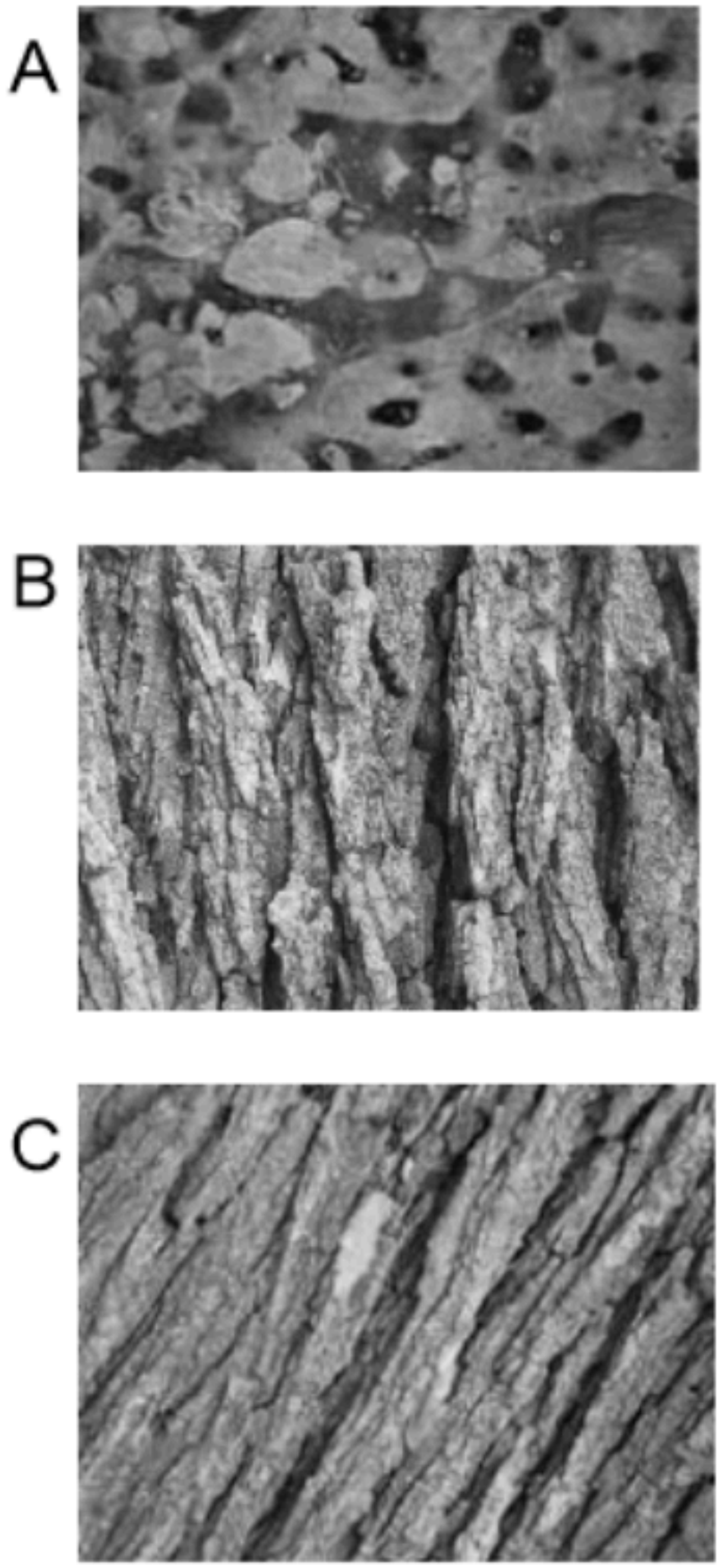


Figure Credit: Leung and Malik, 2001

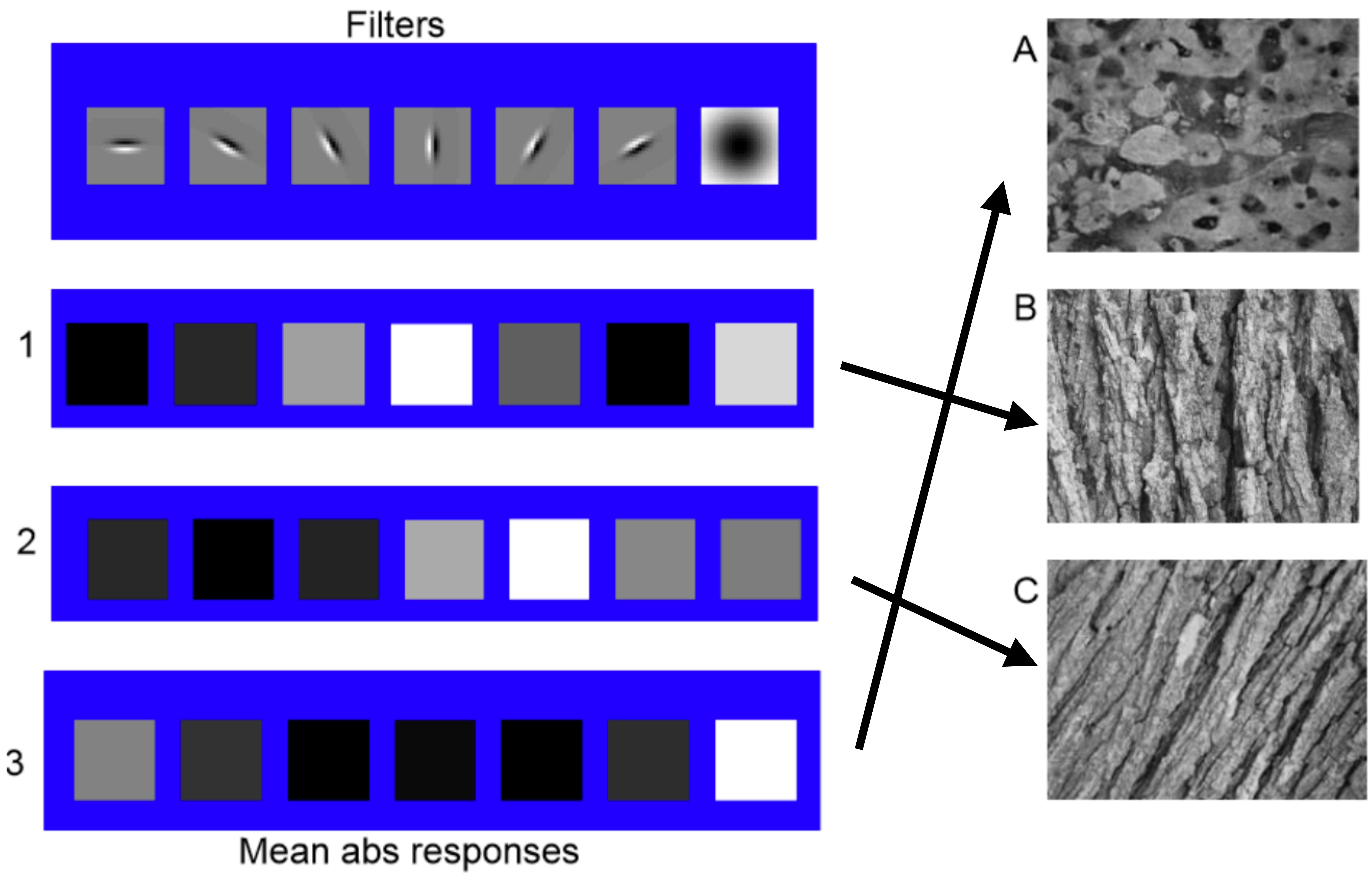
A Short **Exercise**: Match the texture to the response





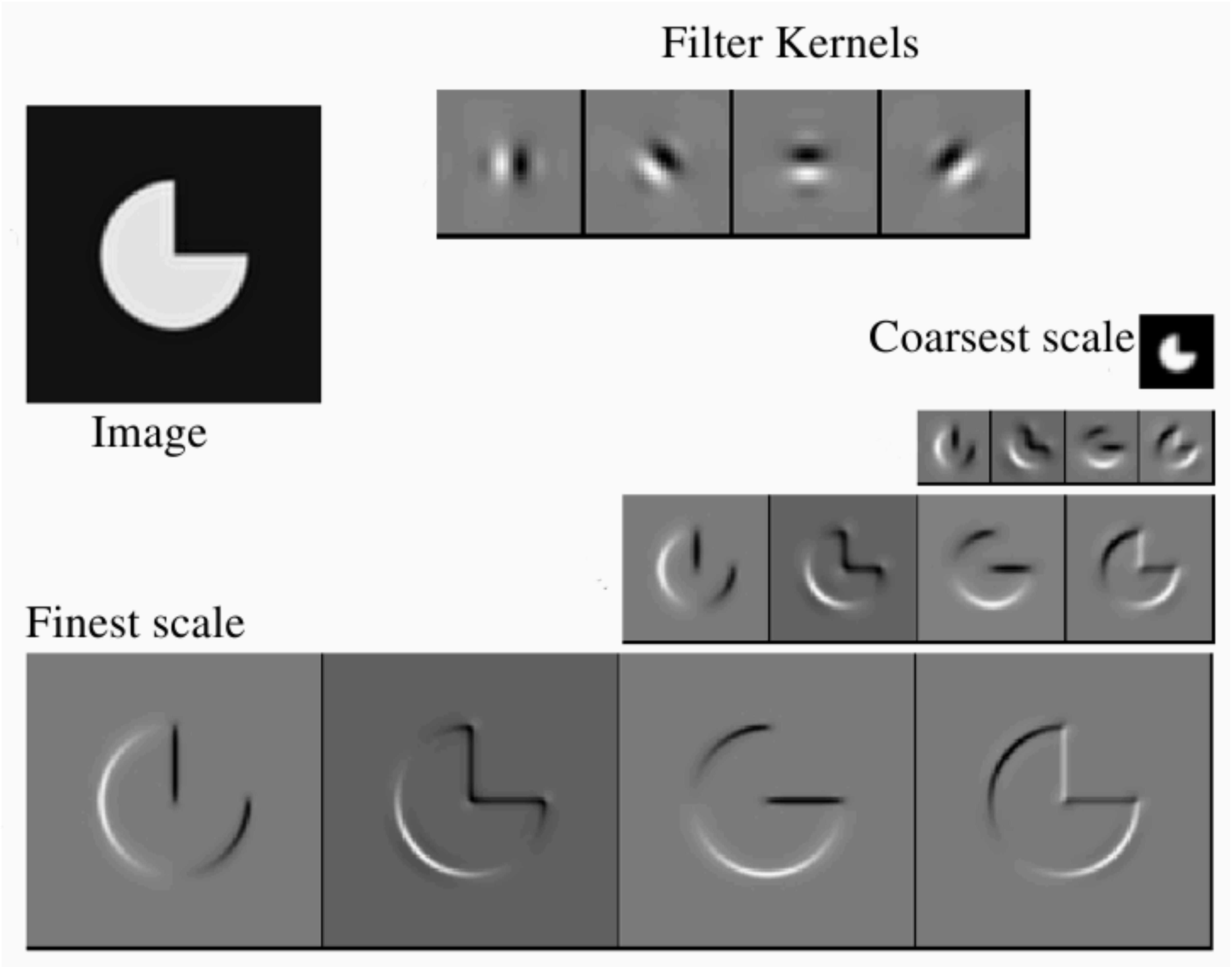
Slide Credit: James Hays

A Short **Exercise**: Match the texture to the response



Slide Credit: James Hays

Oriented Pyramids



Forsyth & Ponce (1st ed.) Figure 9.13

Idea: Apply an oriented filter at each layer

— represent image at a particular scale and orientation

— (We do not study details in this course)

Example Filter Bank Texture Classification

Steps:

1. Compute oriented pyramid or other filter bank responses at each pixel

2. Square the output (makes values positive)

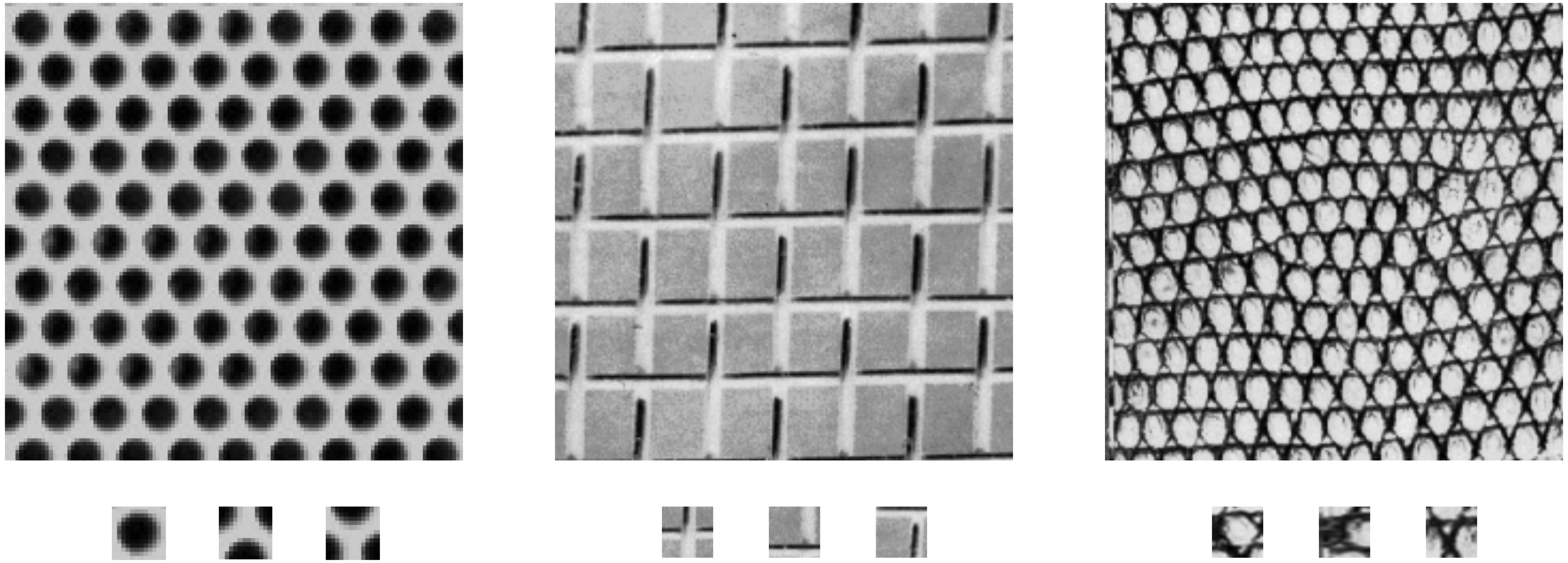
3. Average responses over a neighborhood or image

4. Take statistics of responses
— e.g., histogram of sum-square filter responses over the image

5. Use these statistics to classify different textures

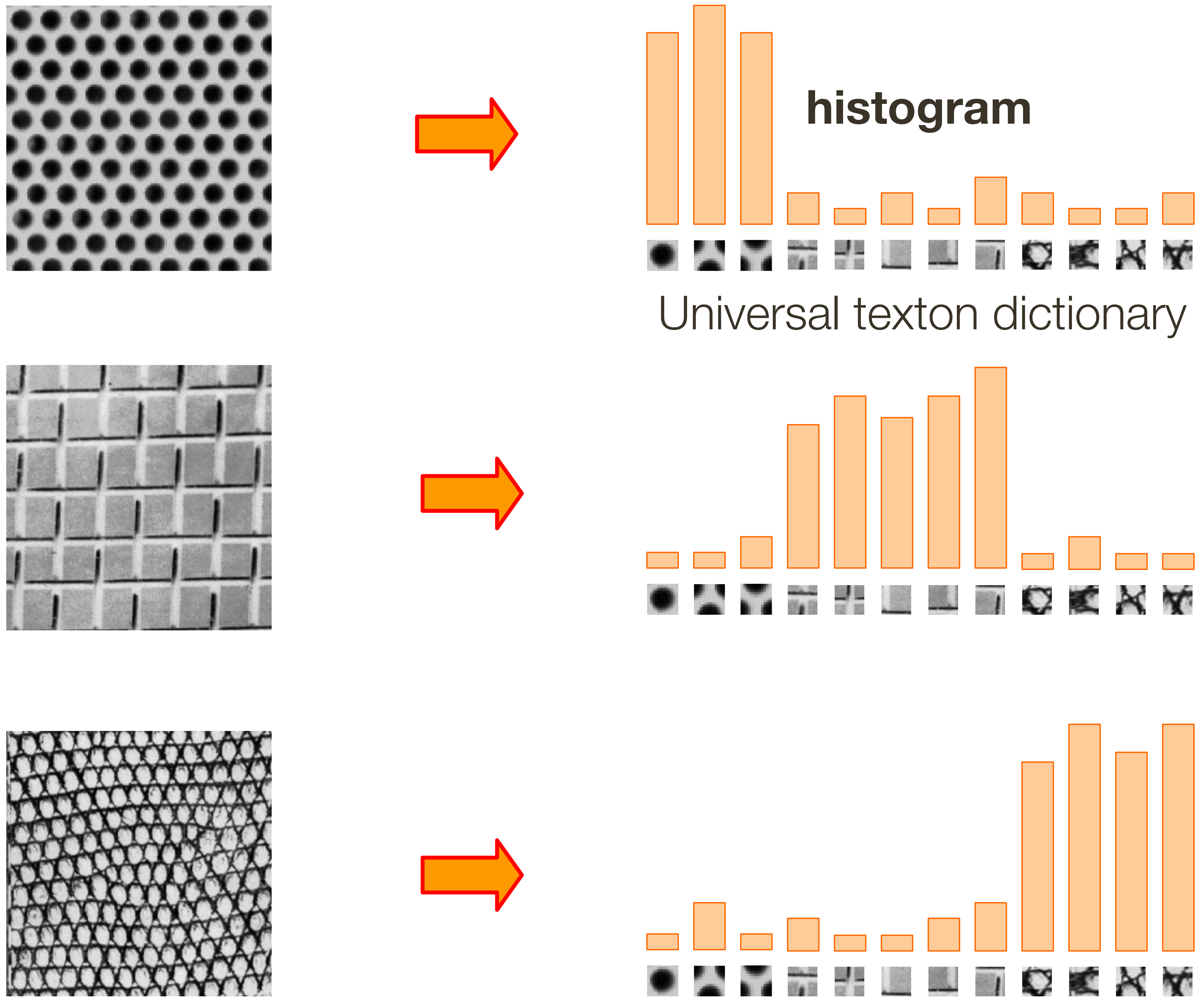
Example Texton representation

- Texture is characterized by the repetition of basic elements or **textons**
- Form a texton **dictionary**, e.g., by k-means or random sampling
- Compute **nearest neighbours** of each image patch to the texton dictionary
- Form distribution of **texton frequencies**, throw away spatial information



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

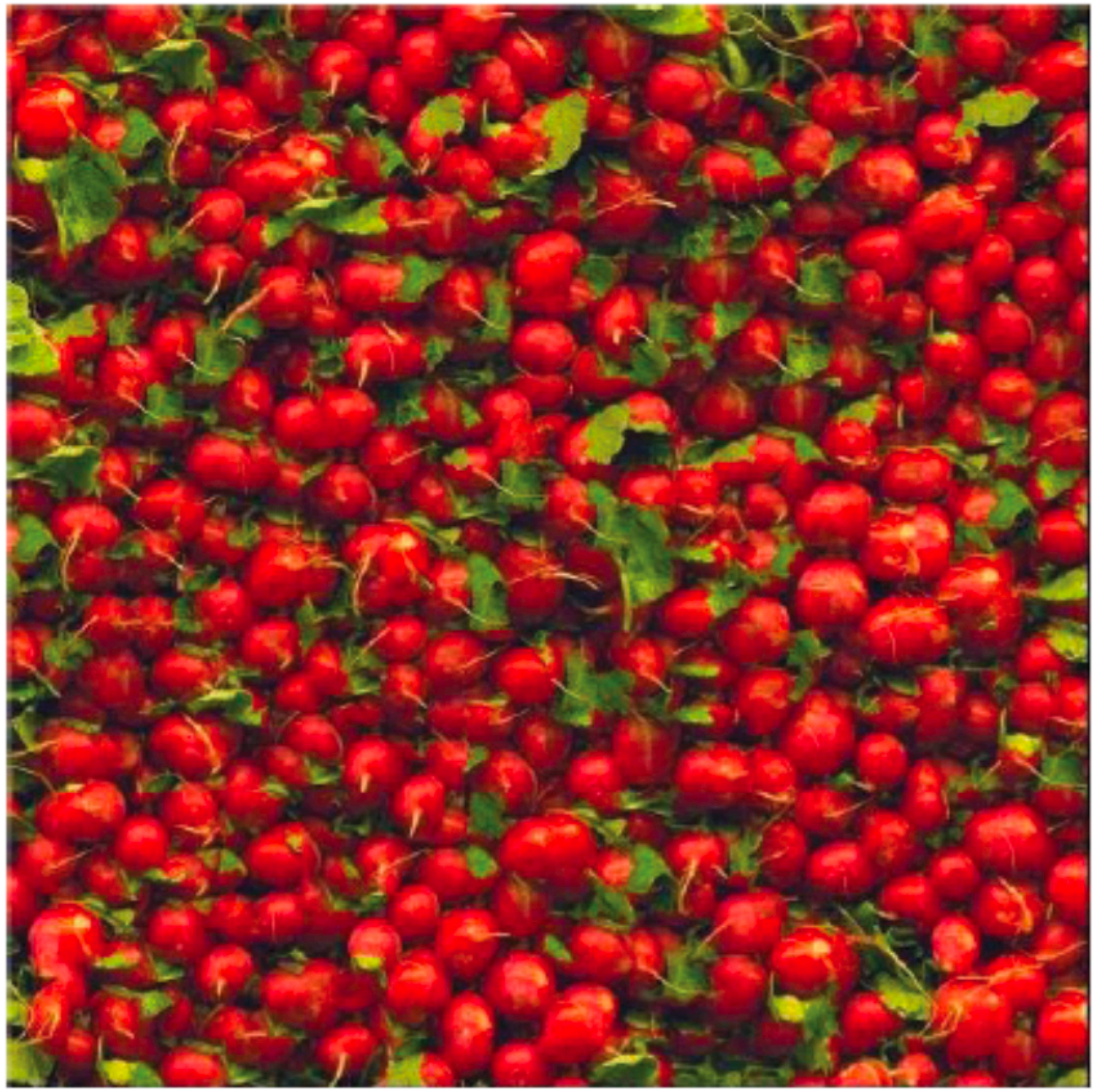
Example Texton representation



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Texture **Synthesis**

radishes



lots more radishes

Szeliski, Fig. 10.49

Texture **Synthesis**

Why might we want to synthesize texture?

1. To fill holes in images (**inpainting**)

— Art directors might want to remove telephone wires. Restorers might want to remove scratches or marks.

— We need to find something to put in place of the pixels that were removed

— We synthesize regions of texture that fit in and look convincing

2. To produce large quantities of texture for computer graphics

— Good textures make object models look more realistic

Texture **Synthesis**

Cover of “The Economist,” June 19, 2010

Photo Credit (right): Reuters/Larry Downing

Assignment 3: Texture Synthesis

Task: Make donkey vanish

Assignment 3: Texture Synthesis

Task: Make donkey vanish

Method: Fill-in regions using texture from the white box

Assignment 3: Texture Synthesis

Task: Make donkey vanish

Method: Fill-in regions using texture from the white box

Texture Synthesis

Objective: Generate new examples of a texture. We take a “data-driven" approach

Idea: Use an image of the texture as the source of a probability model

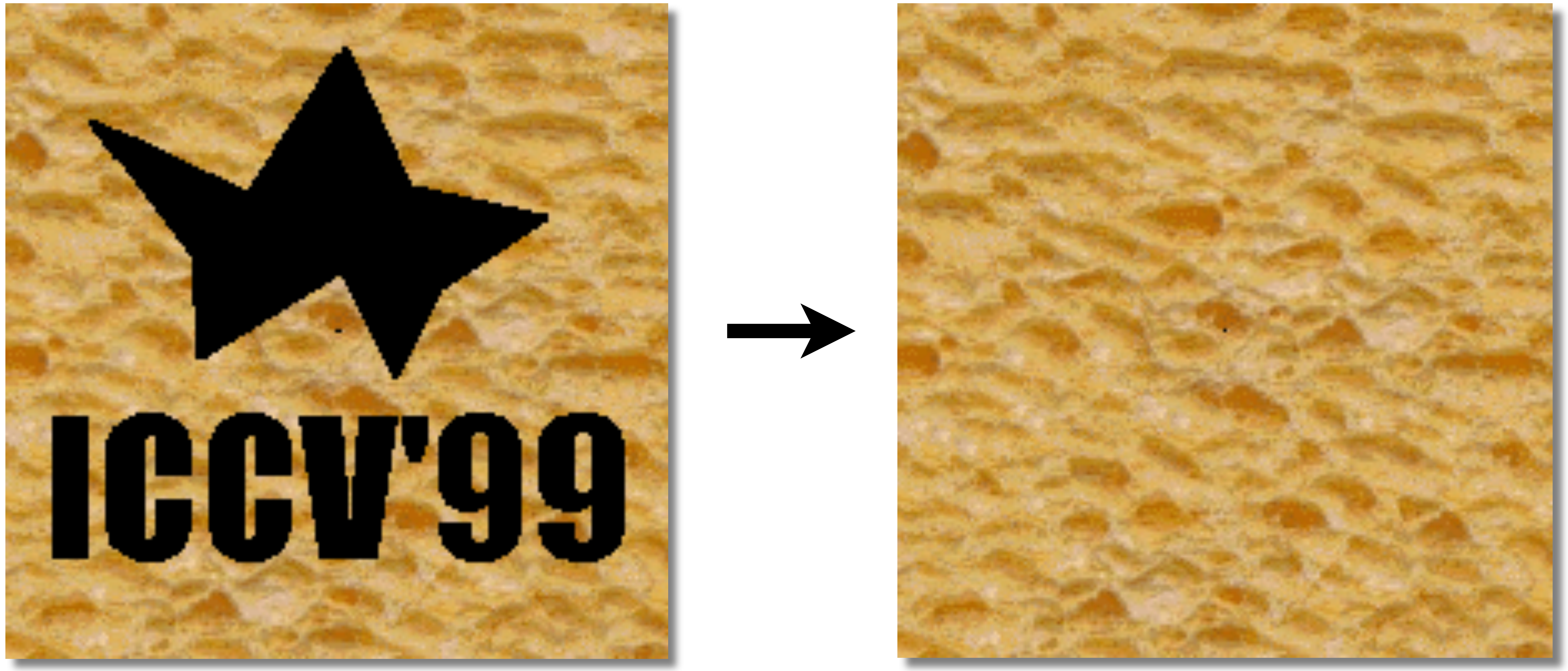
— Draw samples directly from the actual texture

— Can account for more types of structure

— Very simple to implement

— Success depends on choosing a correct “distance”

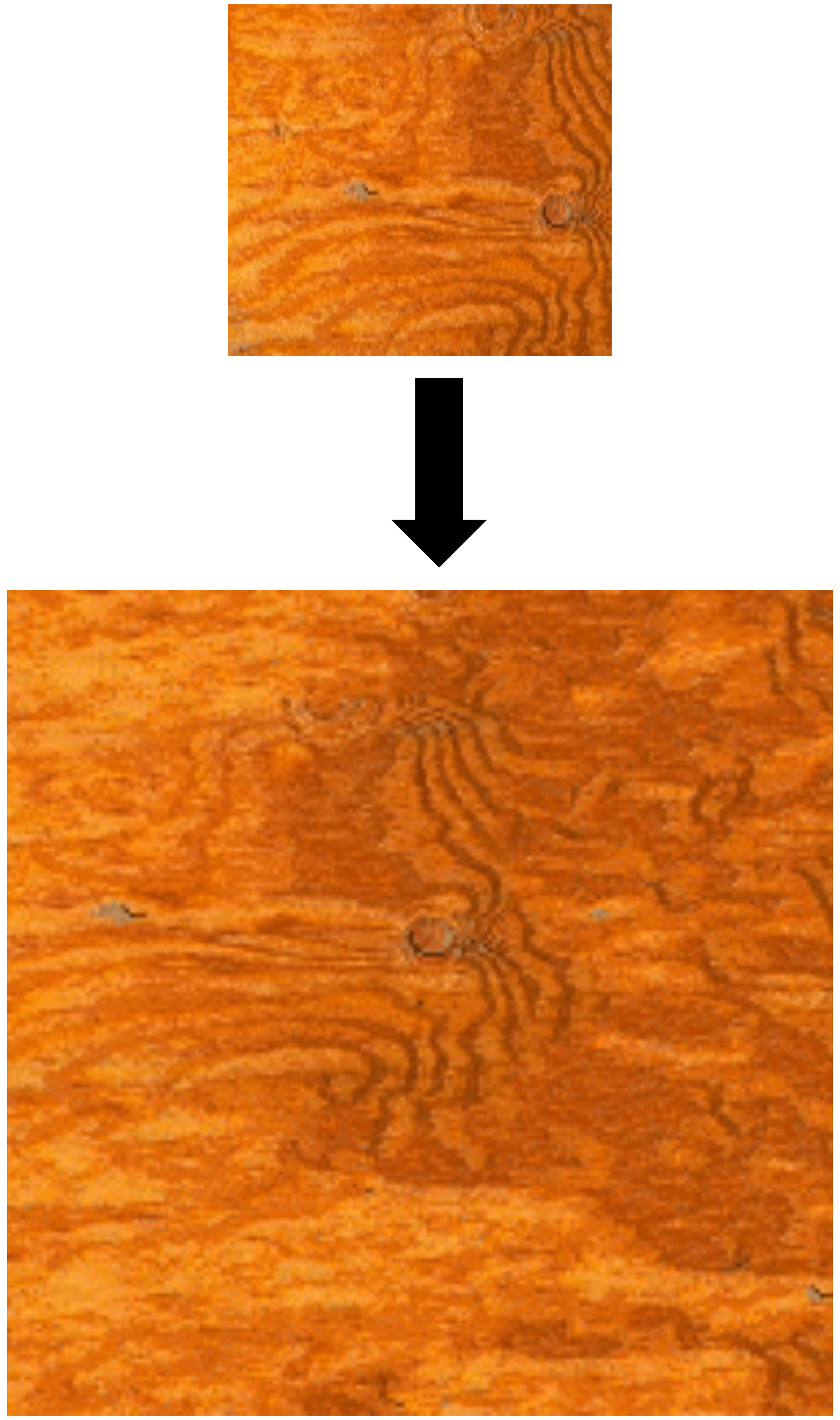
Texture Synthesis by Non-parametric Sampling



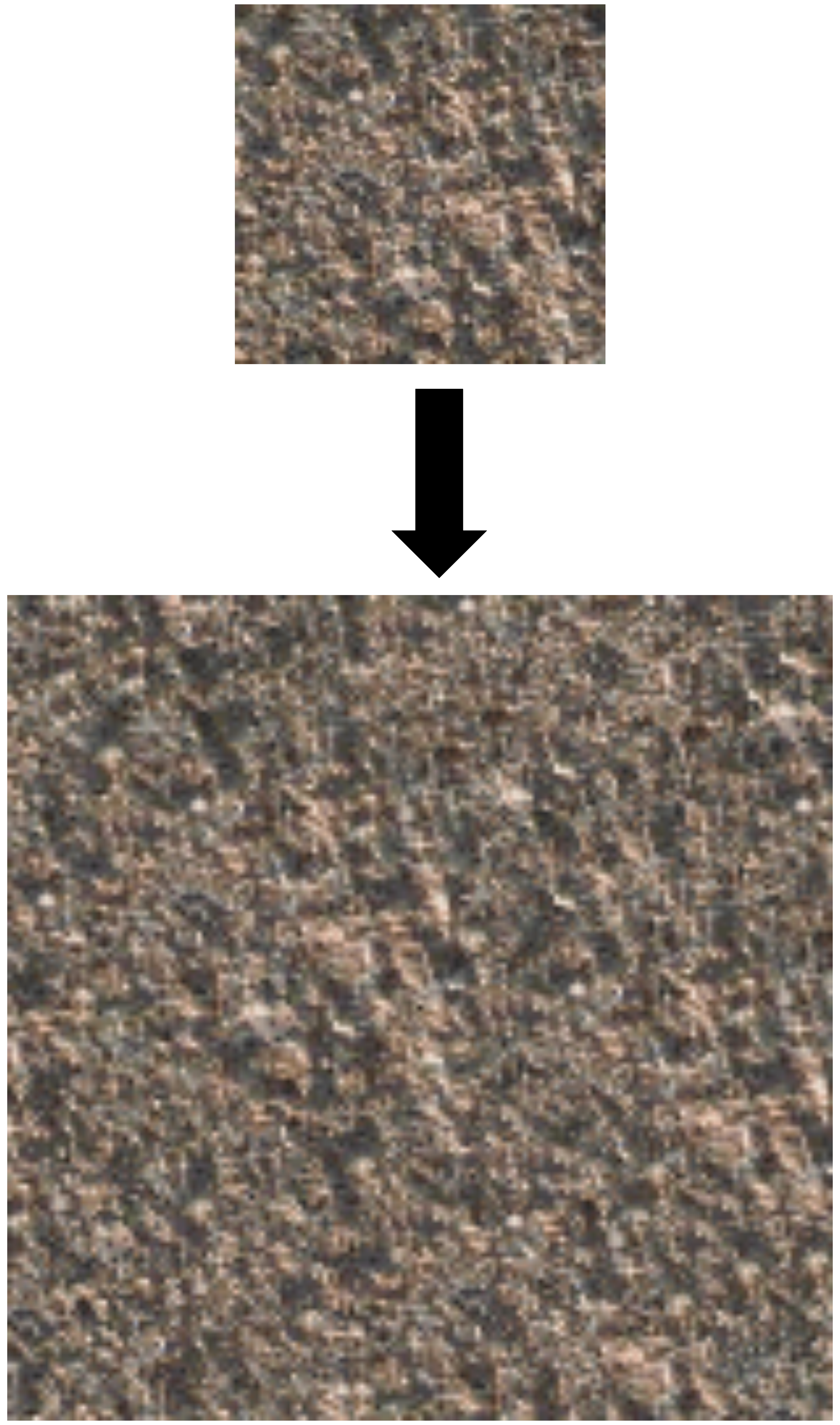
Alexei Efros and Thomas Leung
UC Berkeley

Slide Credit: http://graphics.cs.cmu.edu/people/efros/research/NPS/efros-iccv99.ppt

Efros and Leung

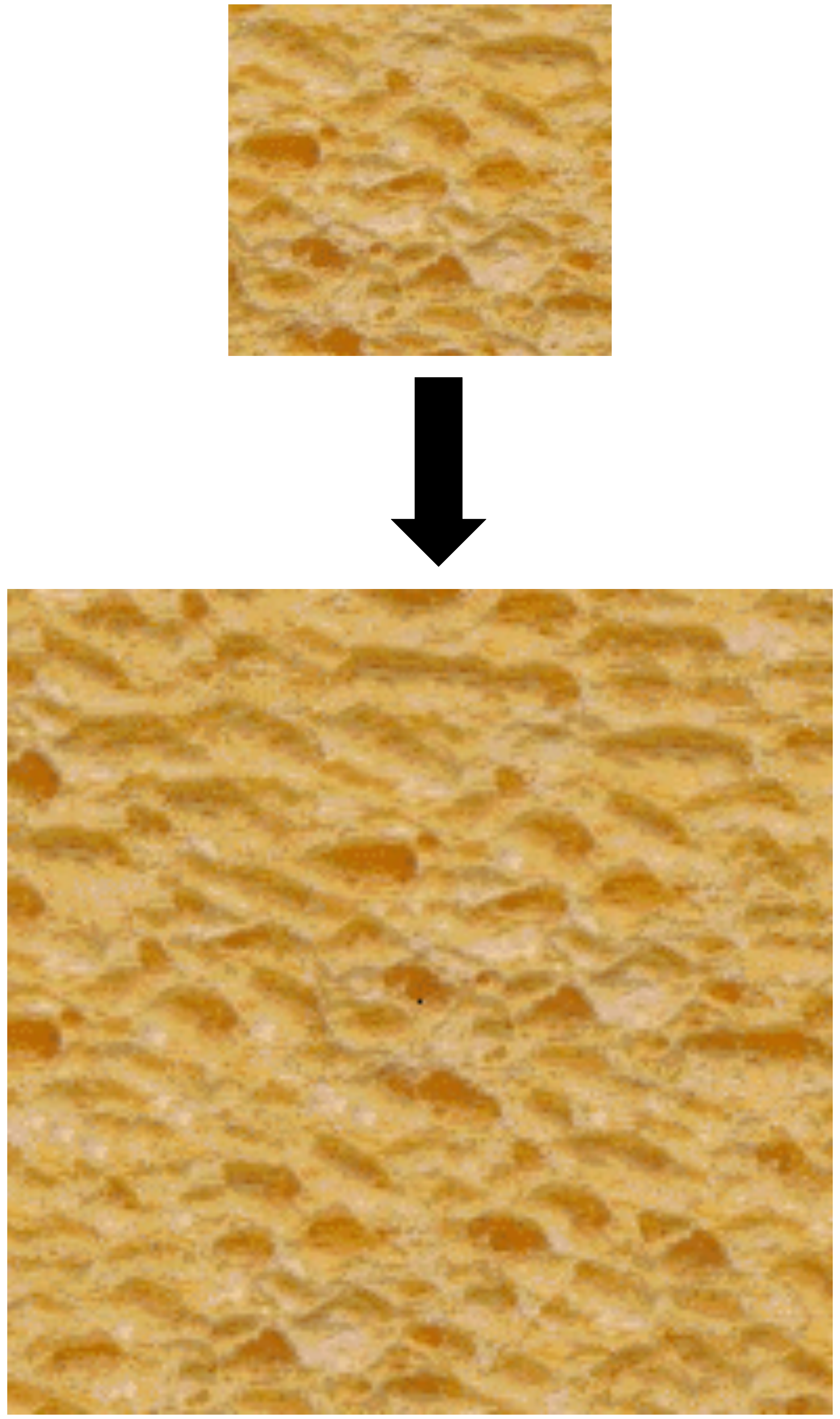


wood

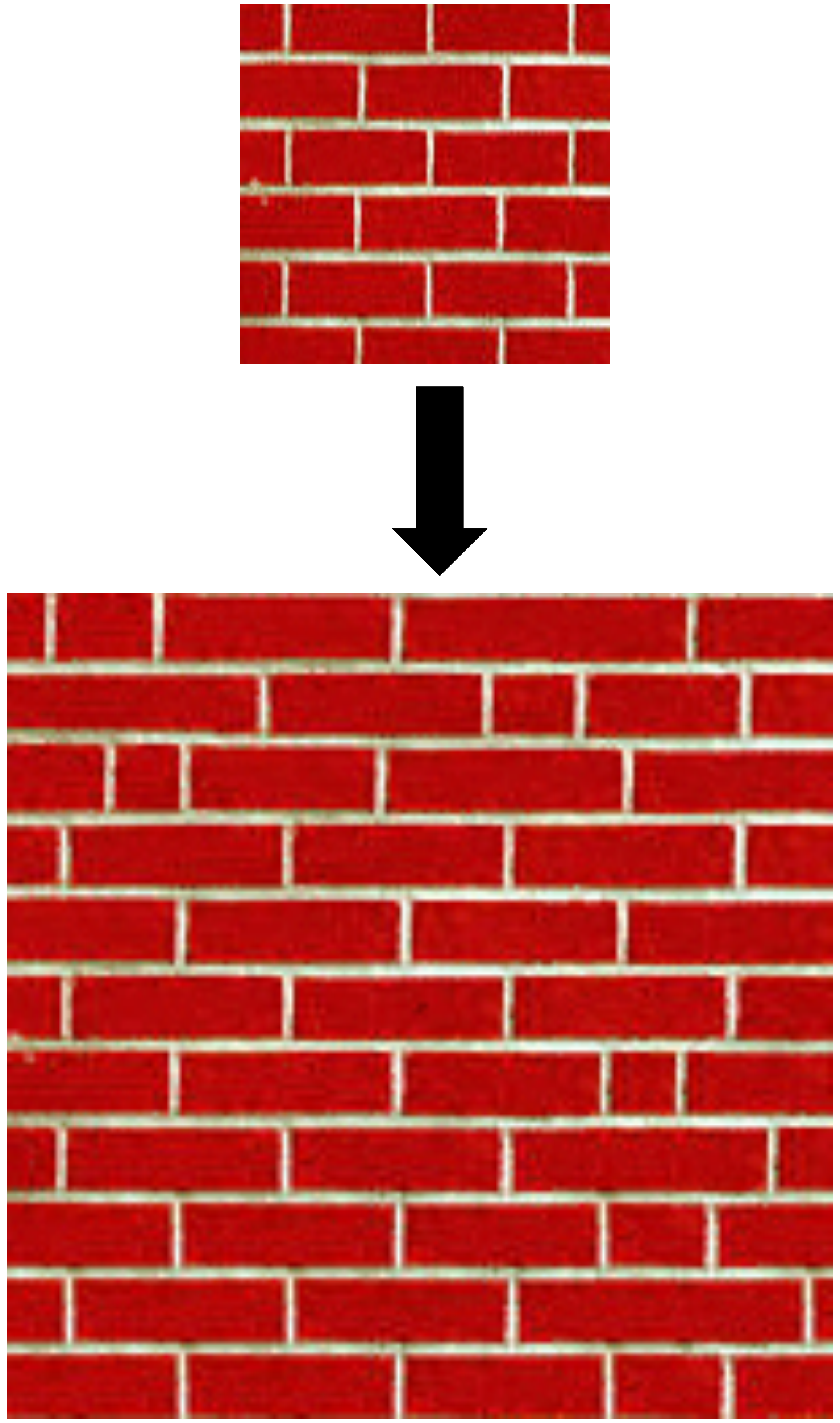


granite

Efros and Leung

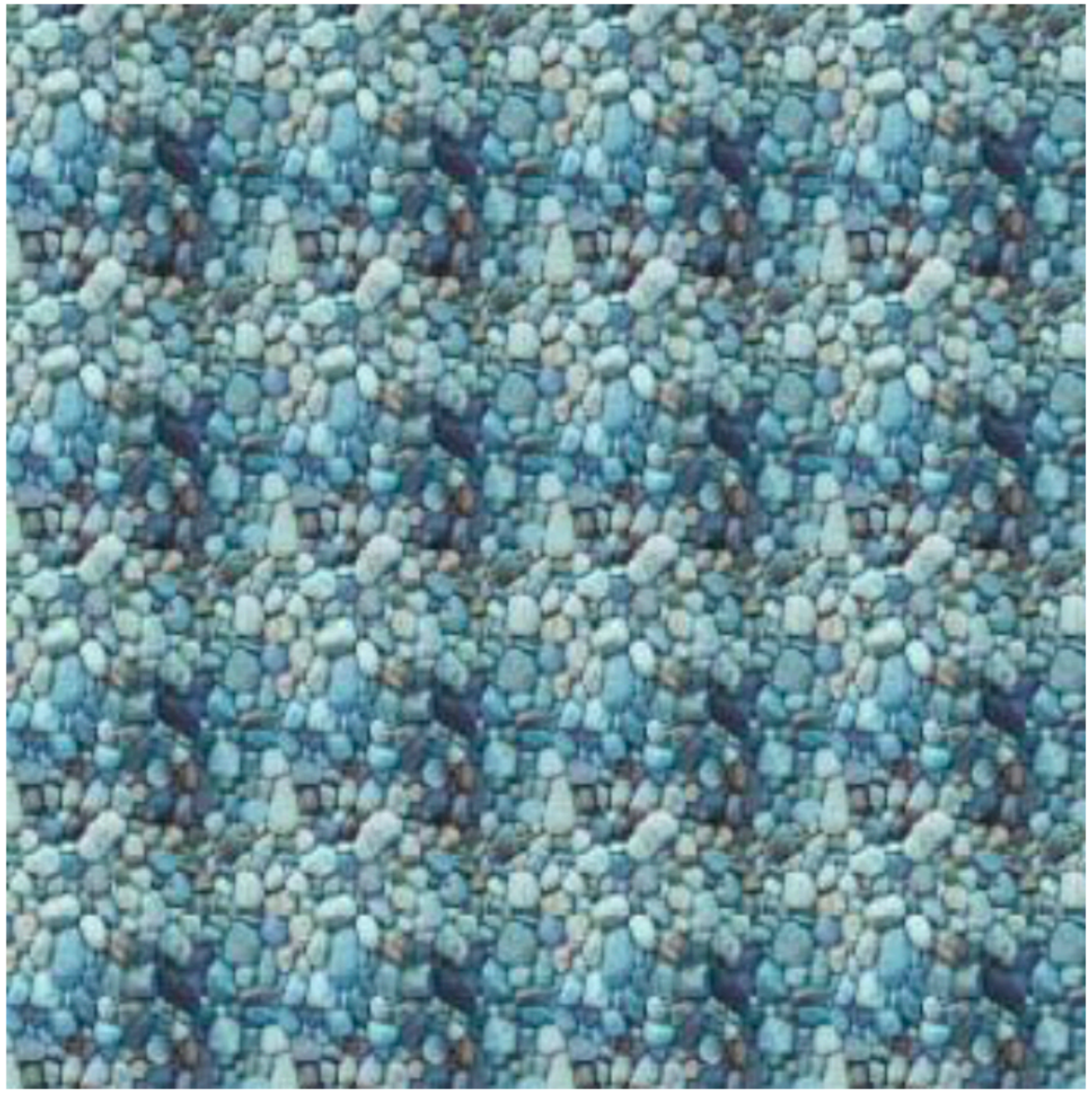


white bread

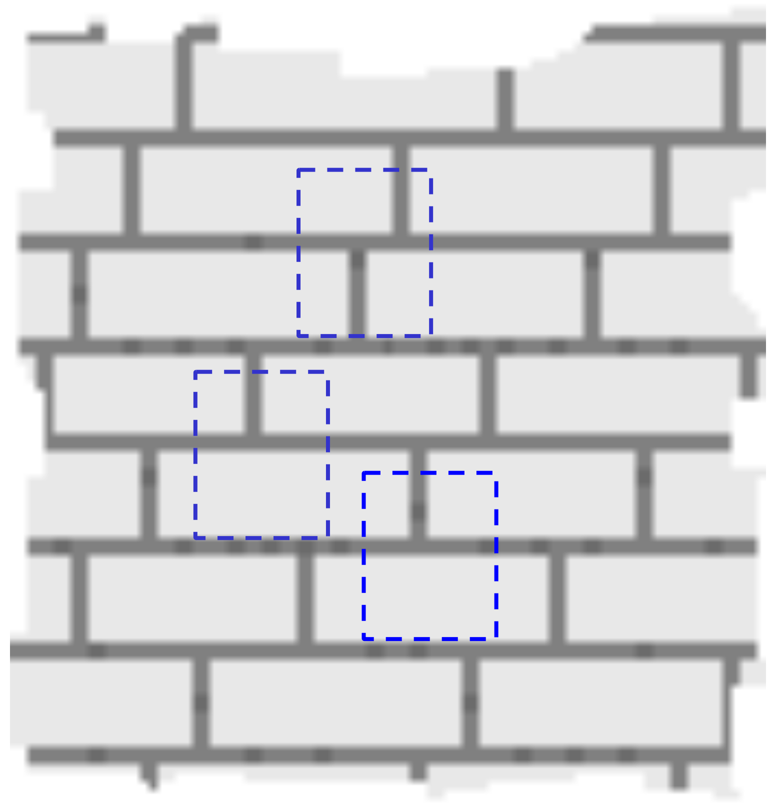


brick wall

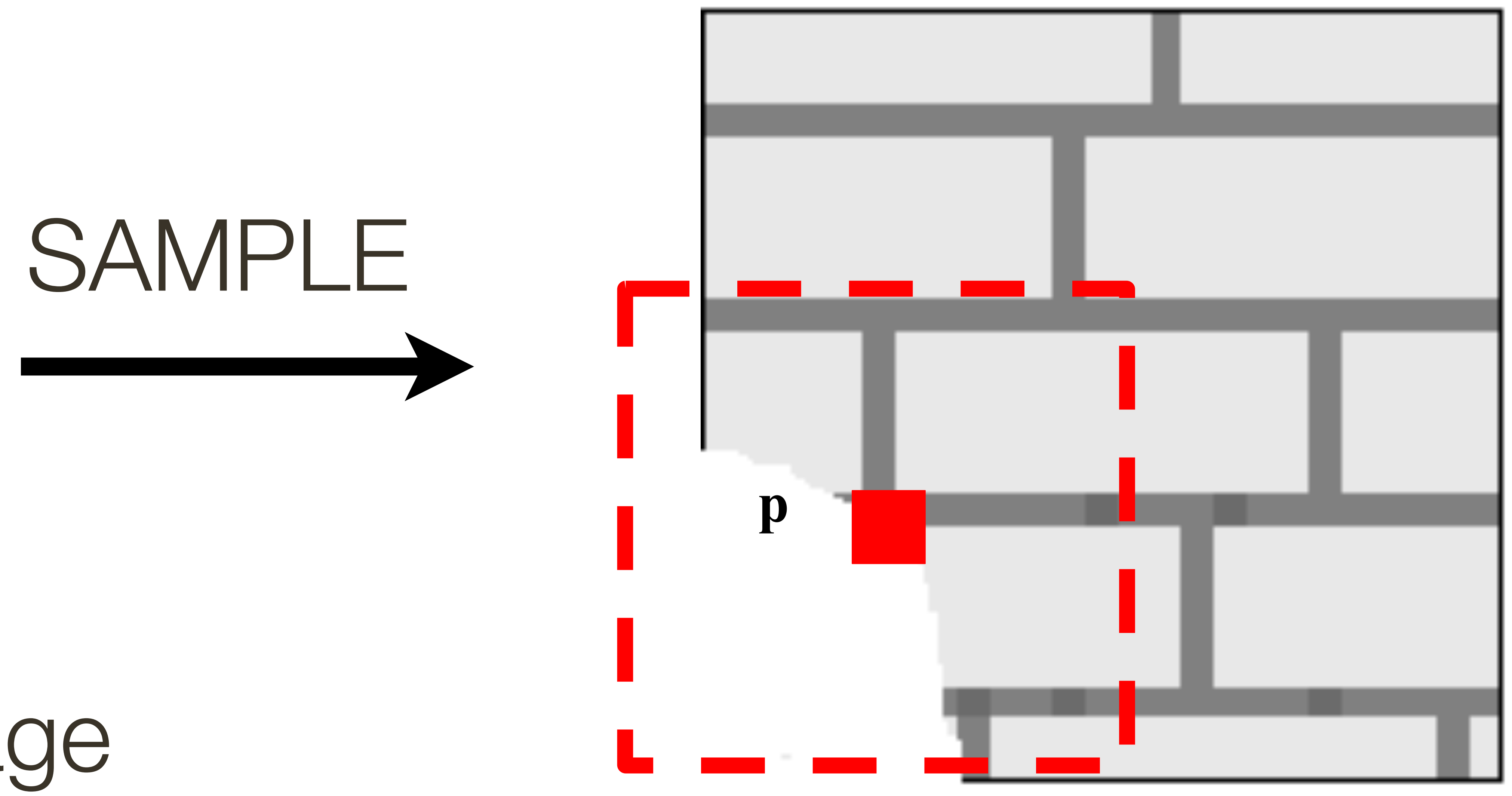
Like **Copying**, But not Just Repetition



Efros and Leung: Synthesizing One Pixel



Infinite sample image

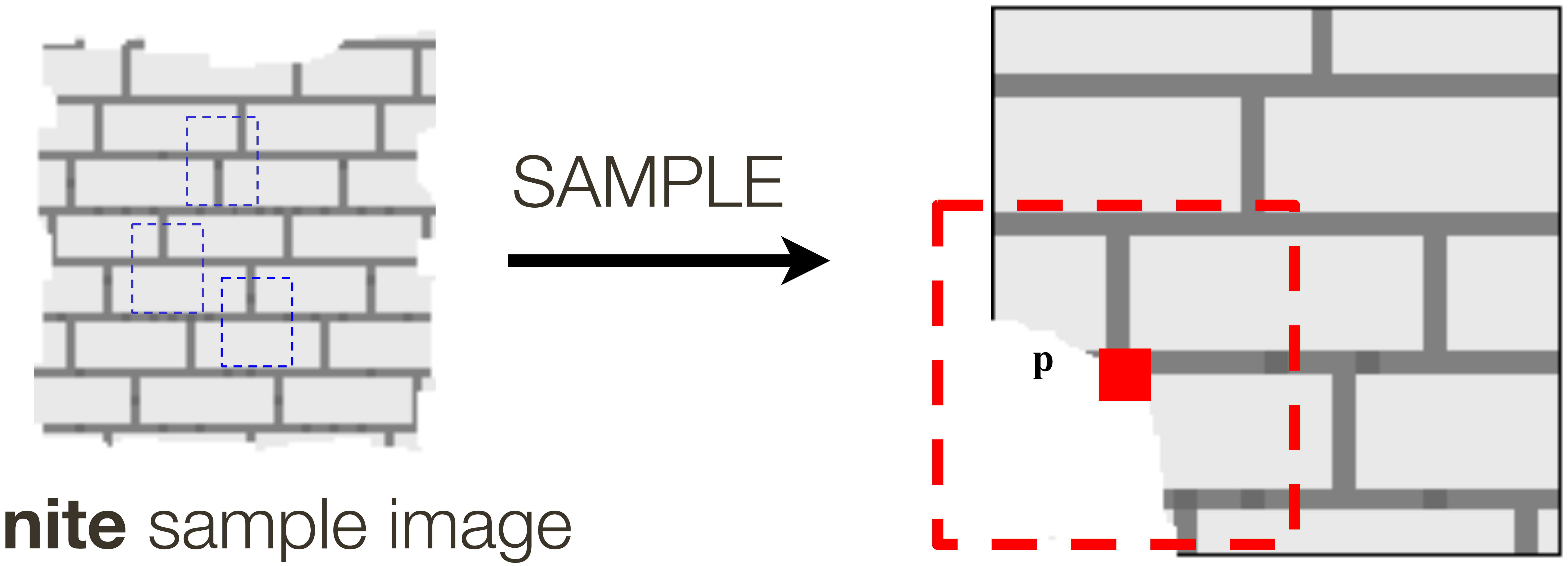


— What is **conditional** probability distribution of *p*, given the neighbourhood window?

— Directly search the input image for all such neighbourhoods to produce a **histogram** for *p*

— To **synthesize** *p*, pick one match at random

Efros and Leung: Synthesizing One Pixel



Infinite sample image

— Since the sample image is finite, an exact neighbourhood match might not be present

— Find the **best match** using SSD error, weighted by Gaussian to emphasize local structure, and take all samples within some distance from that match

Efros and Leung: Synthesizing Many Pixels

For multiple pixels, "grow" the texture in layers

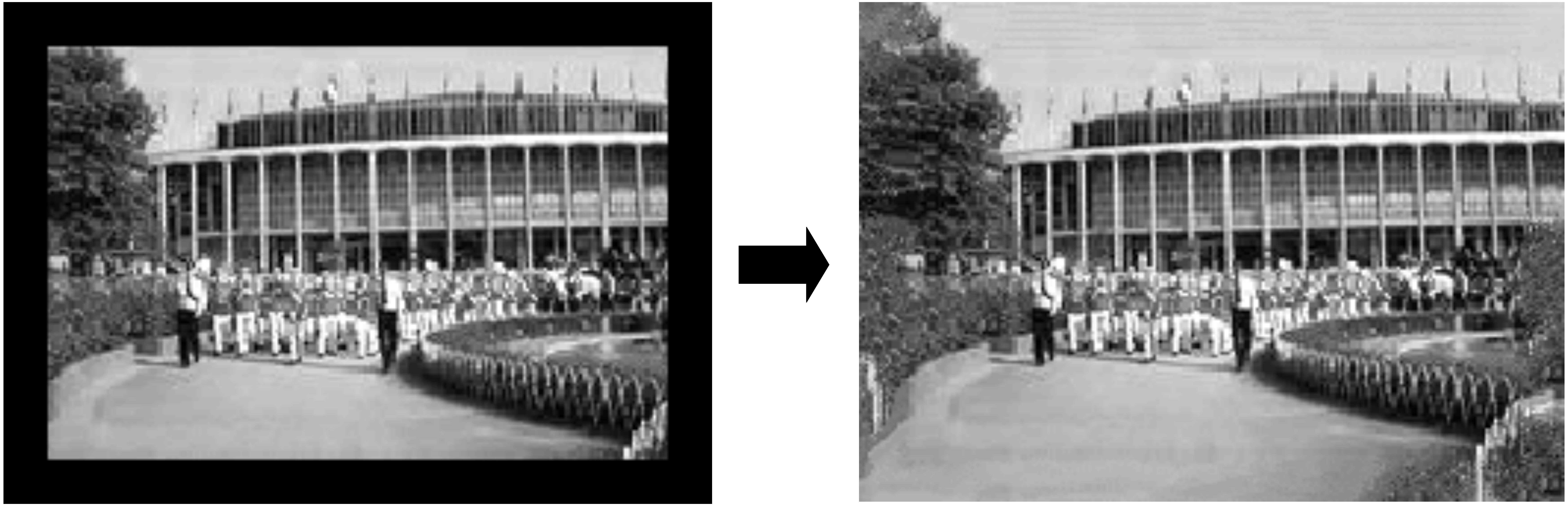
— In the case of hole-filling, start from the edges of the hole

For an interactive demo, see

https://una-dinosauria.github.io/efros-and-leung-js/

(written by Julieta Martinez, a previous CPSC 425 TA)

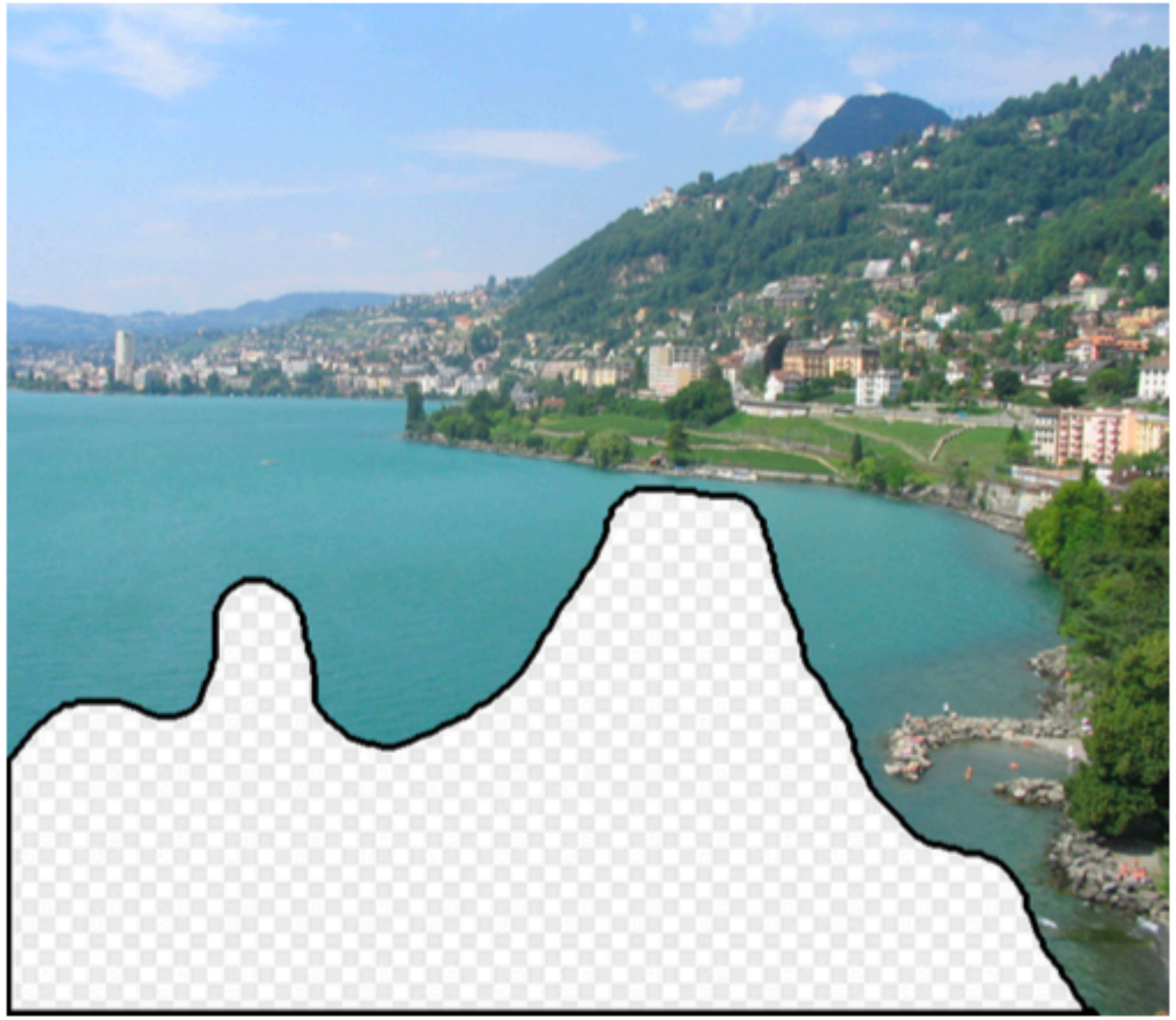
Efros and Leung: Image Extrapolation



Slide Credit: http://graphics.cs.cmu.edu/people/efros/research/NPS/efros-iccv99.ppt

“**Big** Data” Meets Inpainting

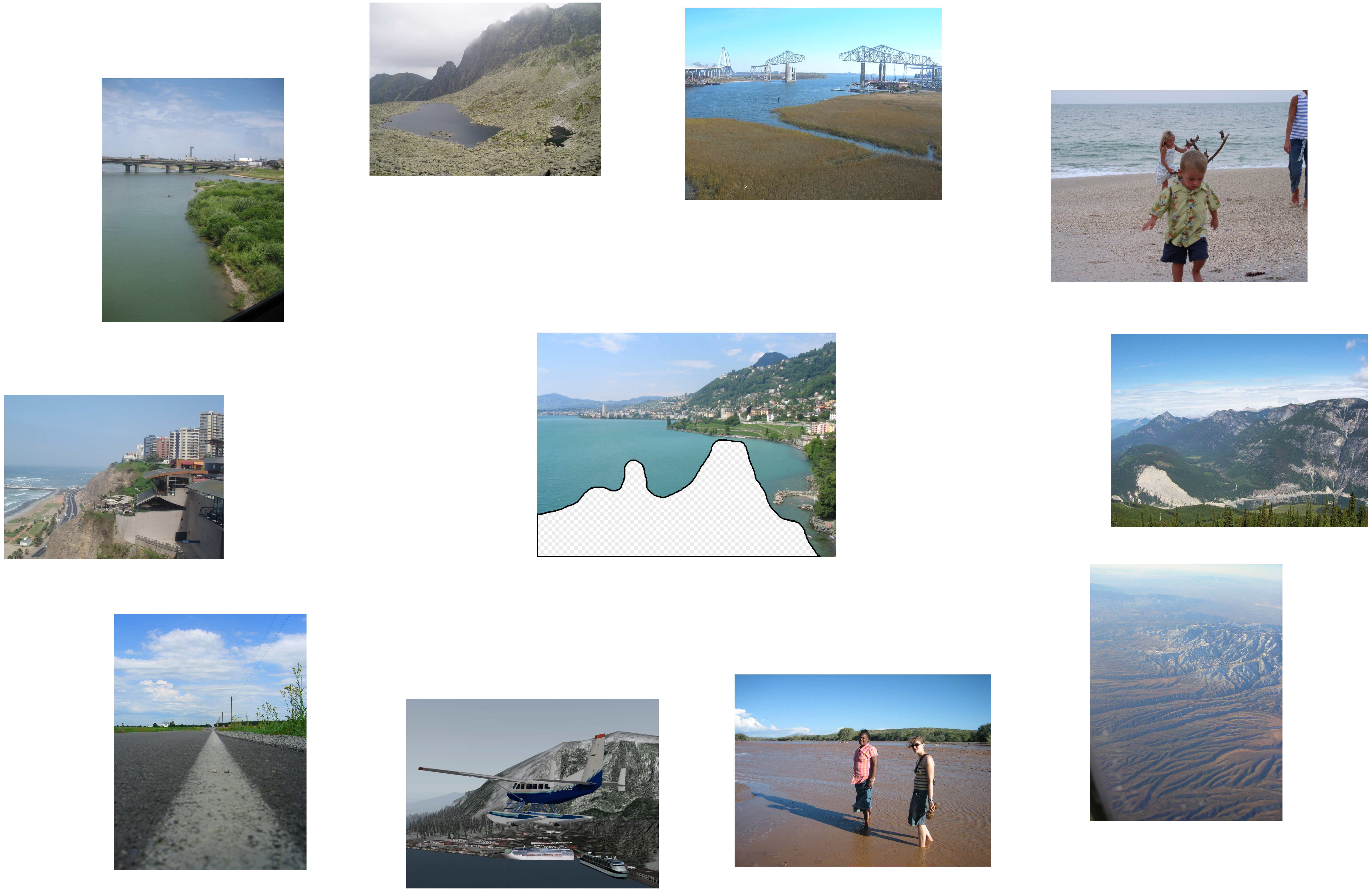
Original Image



Input

Figure Credit: Hays and Efros 2007

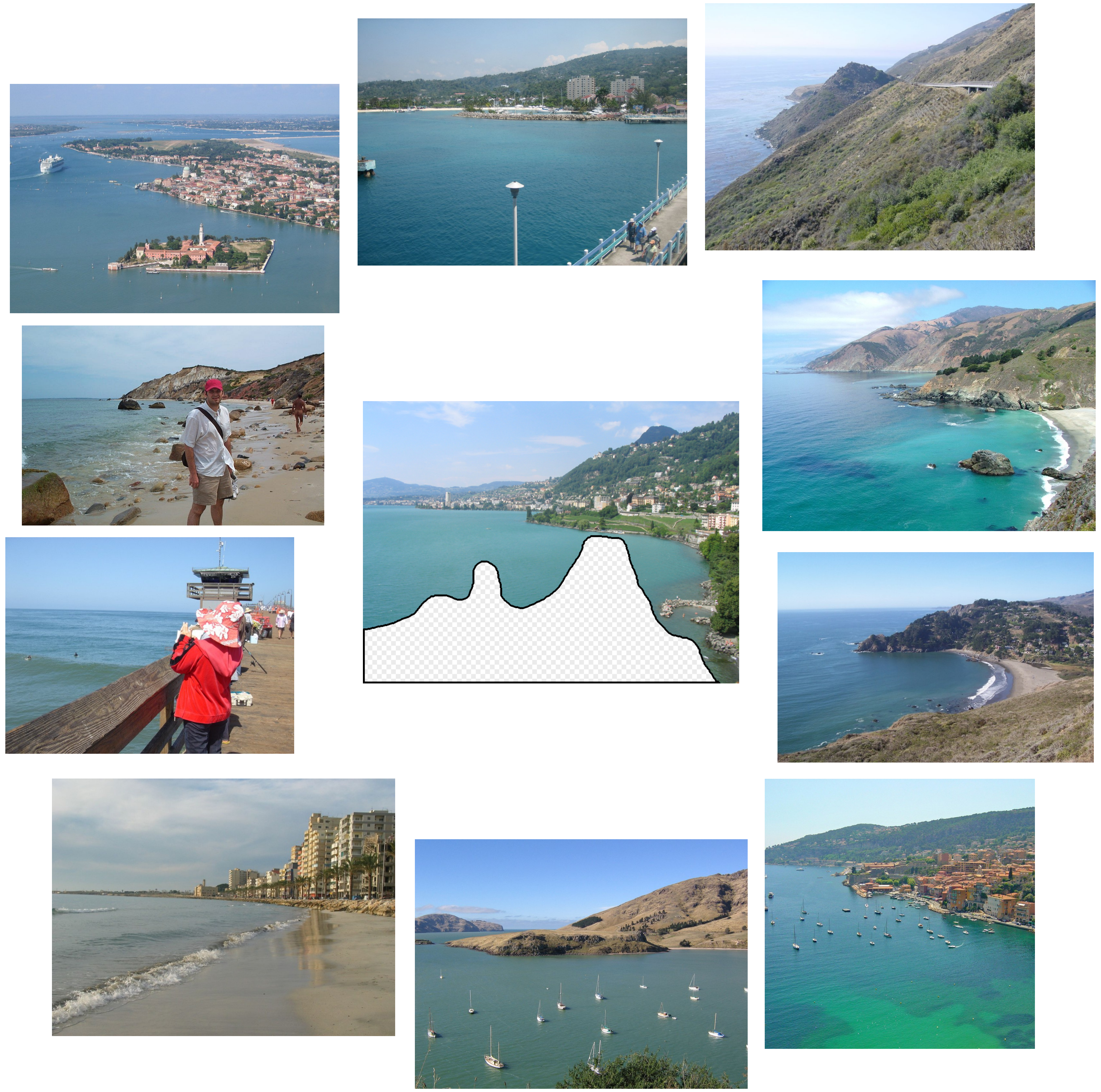
Effectiveness of “Big Data”



10 nearest neighbors from a collection of 20,000 images

Figure Credit: Hays and Efros 2007

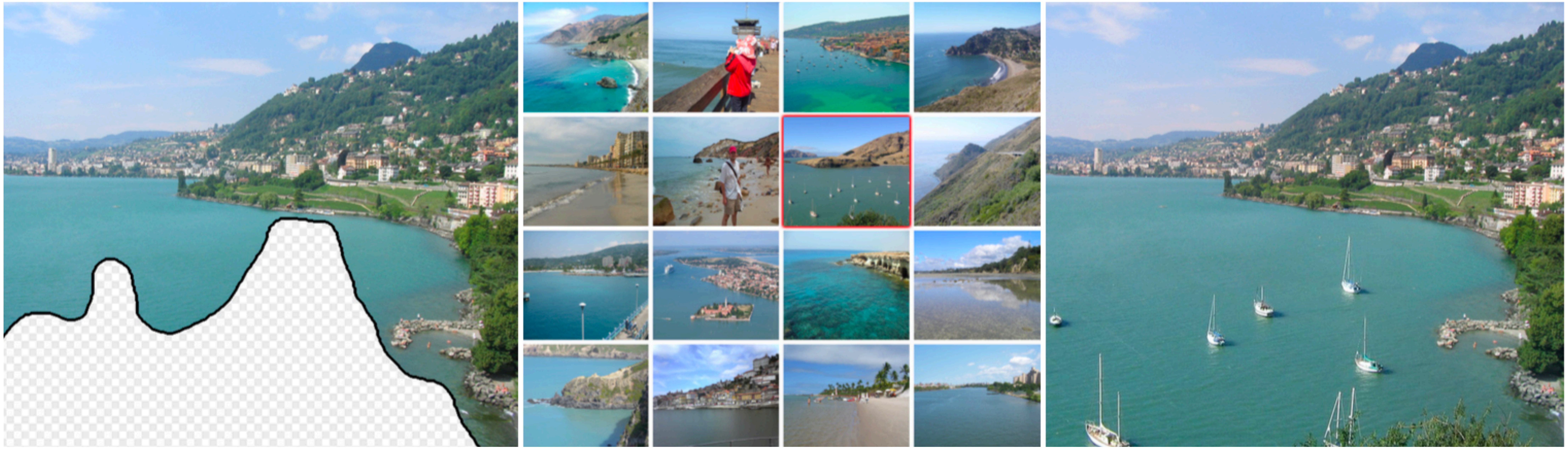
Effectiveness of "Big Data"



10 nearest neighbors from a collection of 2 million images

Figure Credit: Hays and Efros 2007

“**Big** Data” Meets Inpainting



Input Scene Matches Output

Figure Credit: Hays and Efros 2007

“**Big** Data” Meets Inpainting

Figure Credit: Hays and Efros 2007

“**Big** Data” Meets Inpainting

Algorithm sketch (Hays and Efros 2007):

1. Create a short list of a few hundred “best matching" images based on global image statistics

2. Find patches in the short list that match the context surrounding the image region we want to fill

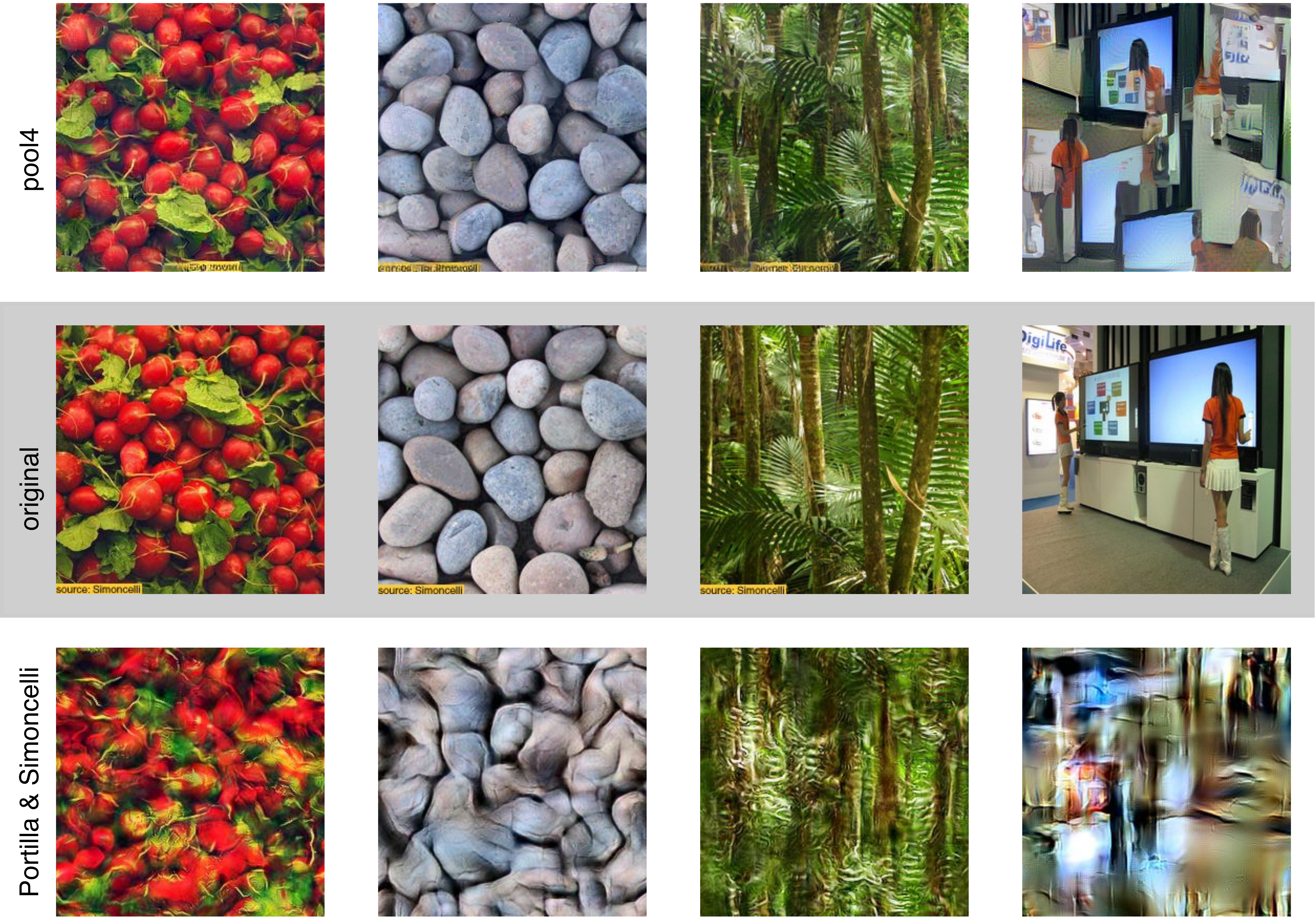
3. Blend the match into the original image

Purely **data-driven**, requires no manual labeling of images

“**Big** Data” Meets Inpainting

Figure Credit: Hays and Efros 2007

Texture Synthesis using **CNNs**



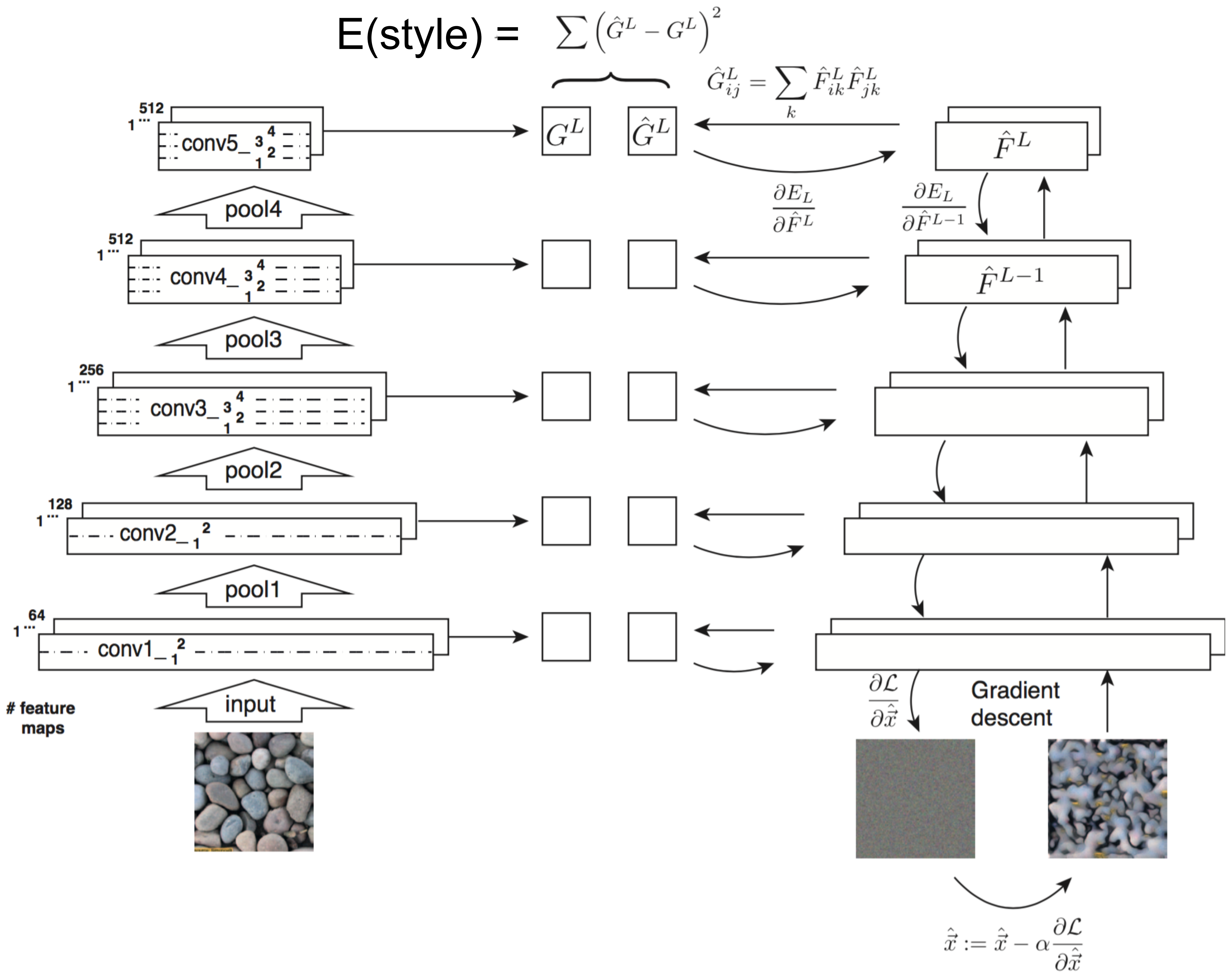
[Gatys et al, CNNs]

[Portilla, Simoncelli, Filter Banks]

Texture Synthesis using **CNNs**

- Start from noise, match statistics of neural net activations at multiple levels
- Uses correlation of activations (Gram Matrix) and a CNN pre-trained for visual recognition

[Gatys et al 2015]



Summary

Texture representations:

— Filter banks, oriented pyramid, patches/texton distribution

— Non-parametric sampling, e.g., nearest neighbours

— Statistics of filter banks or CNN activations

Texture **synthesis:** generate new examples of a texture

— Efros and Leung: Draw samples directly from the texture to generate one pixel at a time. A “data-driven" approach.

Menu for Today

Topics:

- Texture **Analysis, Synthesis**
- Filter Banks, Data-driven Methods
- **Quiz 3**
- Midterm practice questions

Readings:

- **Today's** Lecture: Forsyth & Ponce (2nd ed.) 3.1-3.3

Reminders:

- **Midterm** is next week **October 19th at 5pm**
- **Assignment 3**: Texture Synthesis is available (**due October 26th**)

Next: Please get your **iClickers** —

Quiz 3: **6** questions