



UNIVERSITY OF
SOUTH CAROLINA

CSCE 590 INTRODUCTION TO IMAGE PROCESSING

Texture

Texture

What are the main difference between different textured surfaces?

Change in

- pattern elements
- repetitions



Forsyth and Ponce, "Computer Vision – A Modern Approach 2e"

FIGURE 6.1: Although texture is difficult to define, it has some important and valuable properties. In this image, there are many repeated elements (some leaves form repeated "spots"; others, and branches, form "bars" at various scales; and so on). Our perception of the material is quite intimately related to the texture (what would the surface feel like if you ran your fingers over it? what is soggy? what is prickly? what is smooth?). Notice how much information you are getting about the type of plants, their shape, the shape of free space, and so on, from the textures. *Geoff Brightling © Dorling Kindersley, used with permission.*



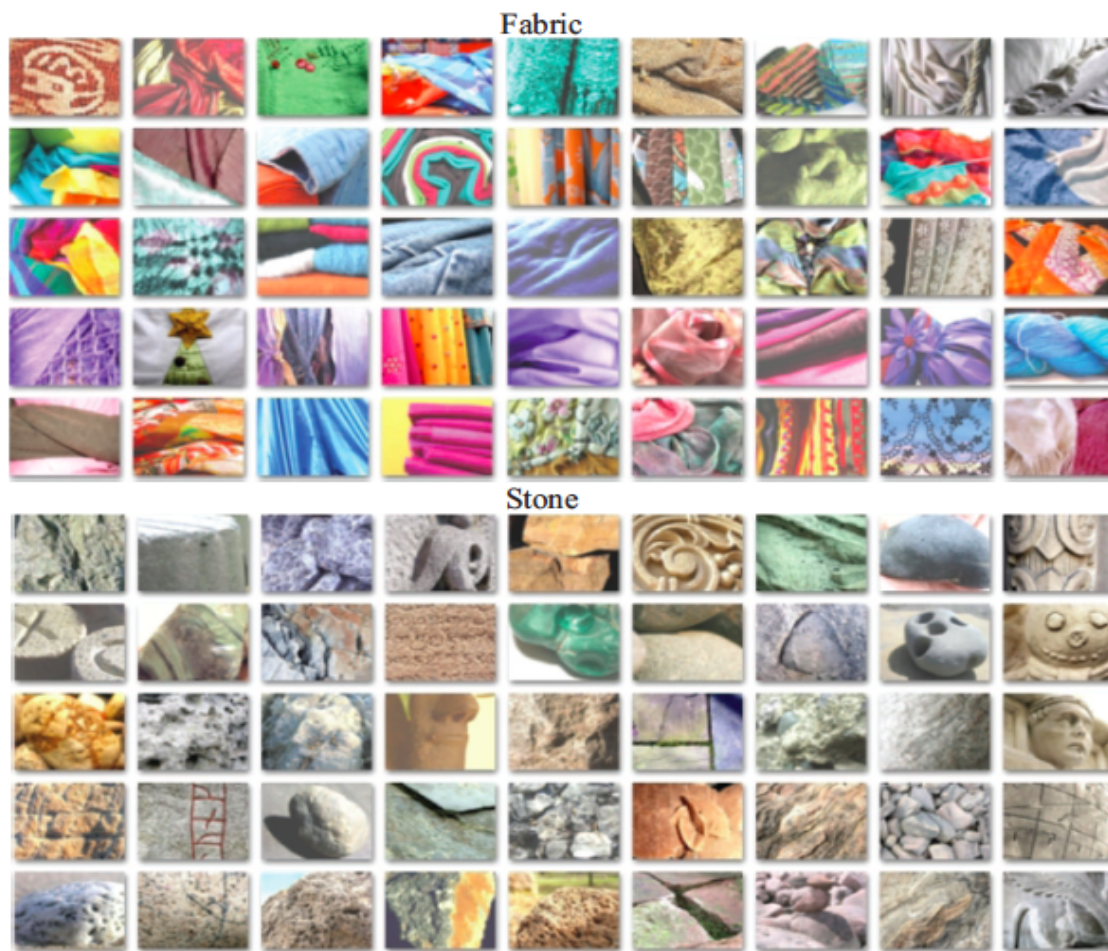


FIGURE 6.2: Typically, different materials display different image textures. These are example images from a collection of 1,000 material images, described in by Sharan *et al.* (2009); there are 100 images in each of the ten categories, including the two categories shown here (fabric and stone). Notice how (a) the textures vary widely, even within a material category; and (b) different materials seem to display quite different textures. This figure shows elements of a database collected by C. Liu, L. Sharan, E. Adelson, and R. Rosenholtz, and published at http://people.csail.mit.edu/lavanya/research_sharan.html. Figure by kind permission of the collectors.

Forsyth and Ponce,
 “Computer Vision – A
 Modern Approach 2e”



Texture

- Patterns of structure from
 - changes in surface albedo (eg printed cloth)
 - changes in surface shape (eg bark)
 - many small surface patches (eg leaves on a bush)
- Texture tells us
 - what a surface is like
 - (sometimes) object identity
 - (sometimes) surface shape



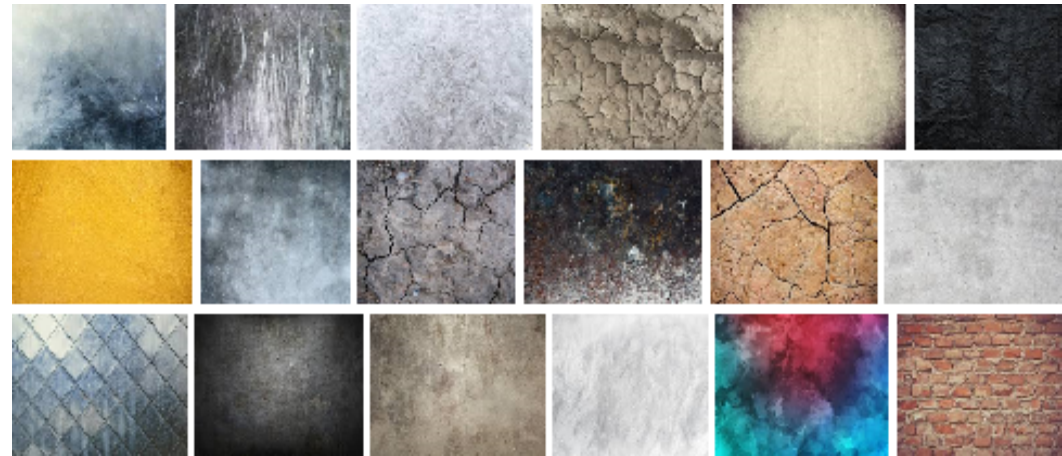
Texture

- Core problems:
 - Texture segmentation
 - Texture based recognition
 - Objects, materials, textures
 - Texture synthesis
 - Create synthetic images, fill in holes, image editing using computer graphics
 - Shape from texture
- Key issue: representing texture



Representing Textures

- Textures are made up of
 - stylized subelements
 - spatially repeated in meaningful ways
- Representation:
 - find the subelements
 - represent their statistics



Examples of texture images by a google search



Representing Textures - Subelements

- What are the subelements, and how do we find them?
 - Find subelements by applying filters, looking at the magnitude of the response
 - Extreme case
 - template matching by normalized cross correlation $1/MN \sum_{x,y} (f(x,y) - \bar{f})(T(x,y) - \bar{T}) / \sigma_f \sigma_T$



Representing Textures

- What filters?
 - experience suggests **spots** and **oriented bars** at a variety of different scales
 - Each filter corresponds to one pattern element
- What statistics?
 - The more the merrier
 - At least, mean and standard deviation
 - better, various conditional histograms.



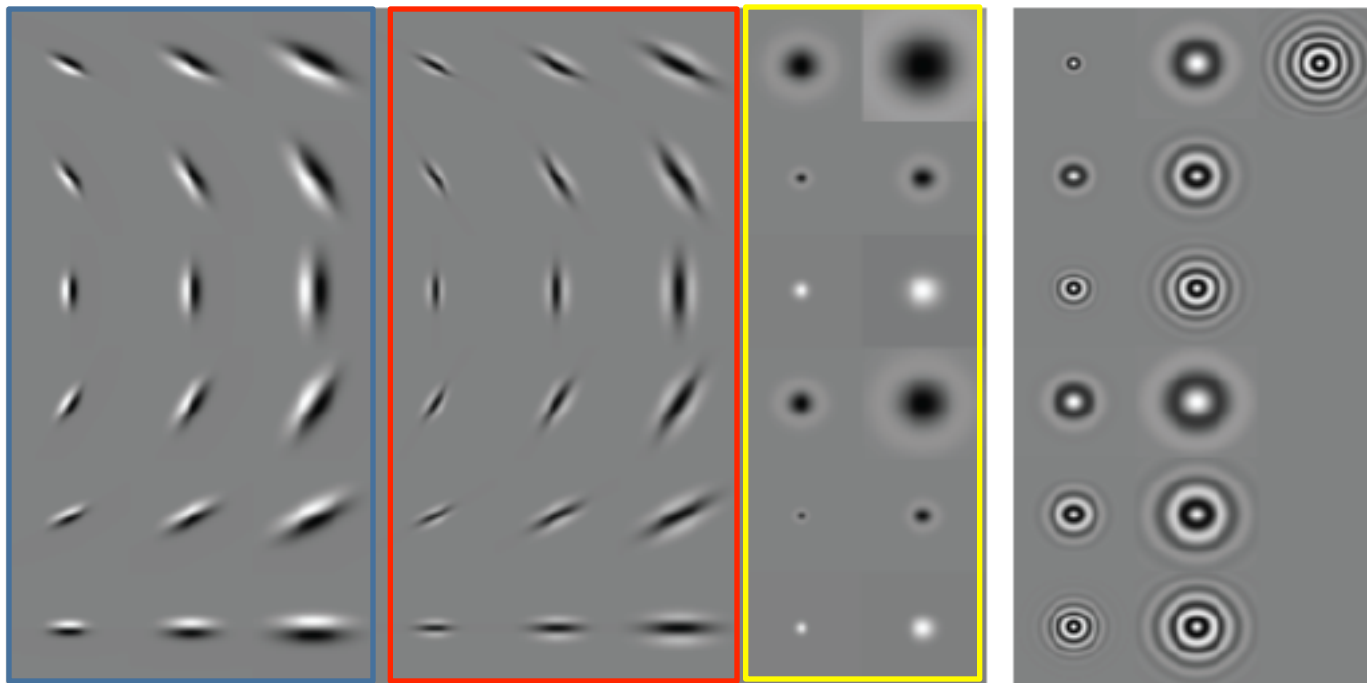
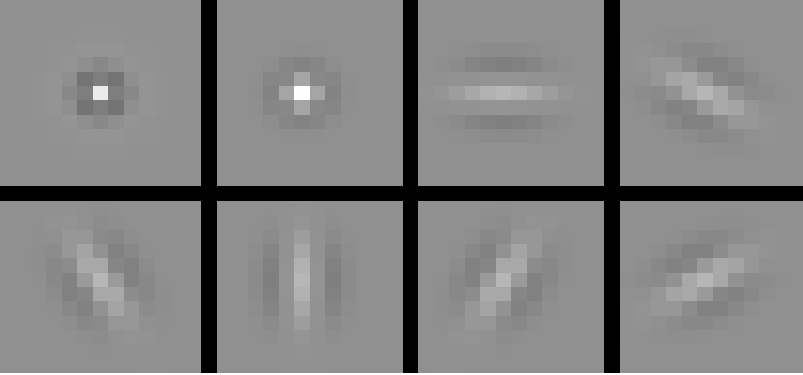
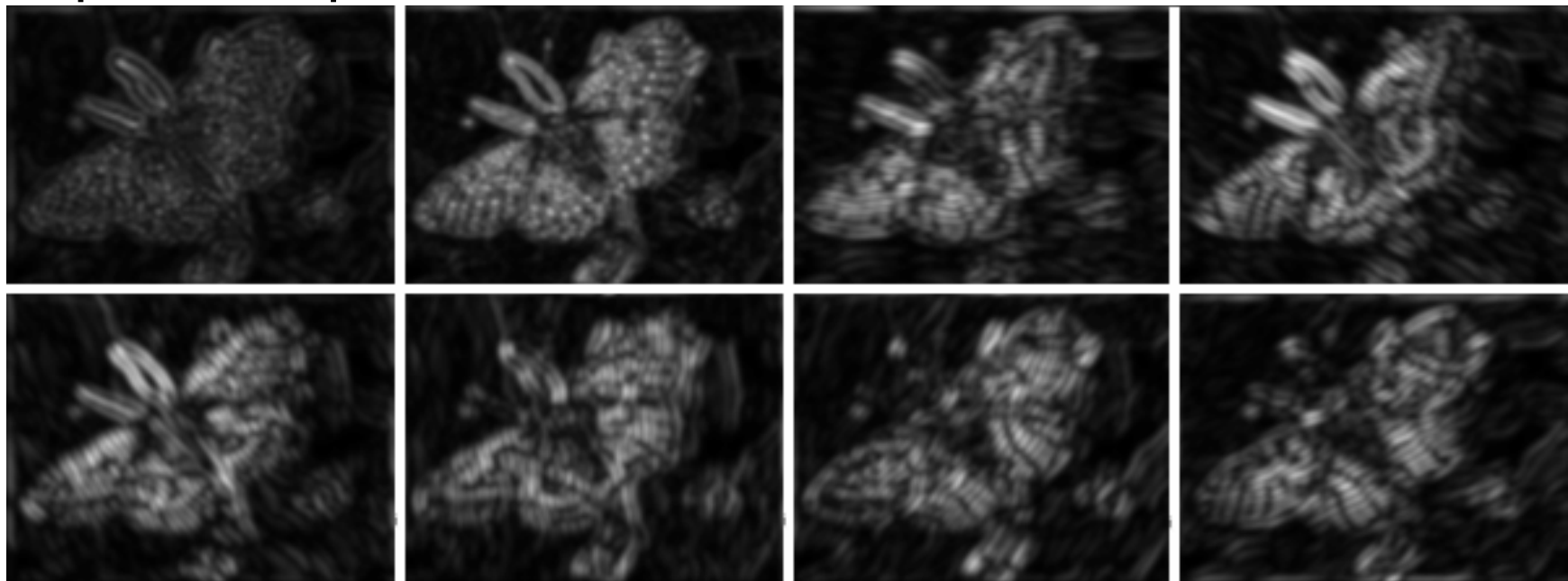


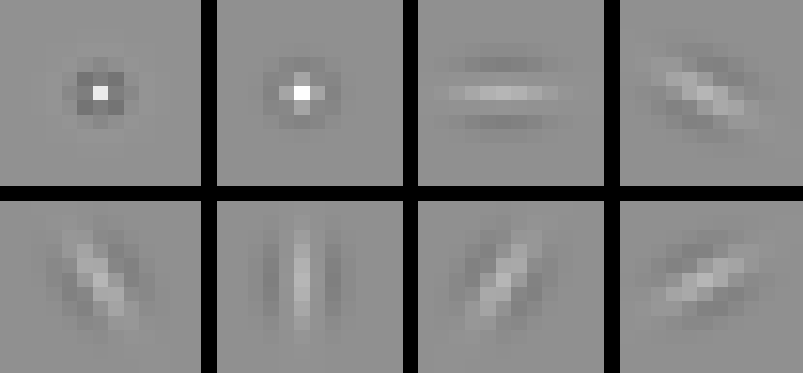
FIGURE 6.4: Left shows a set of 48 oriented filters used for expanding images into a series of responses for texture representation. Each filter is shown on its own scale, with zero represented by a mid-gray level, lighter values being positive, and darker values being negative. The left three columns represent edges at three scales and six orientations; the center three columns represent stripes; and the right two represent two classes of spots (with and without contrast at the boundary) at different scales. This is the set of filters used by Leung and Malik (2001). Right shows a set of orientation-independent filters, used by Schmid (2001), using the same representation (there are only 13 filters in this set, so there are five empty slots in the image). The orientation-independence property means that these filters look like complicated spots.



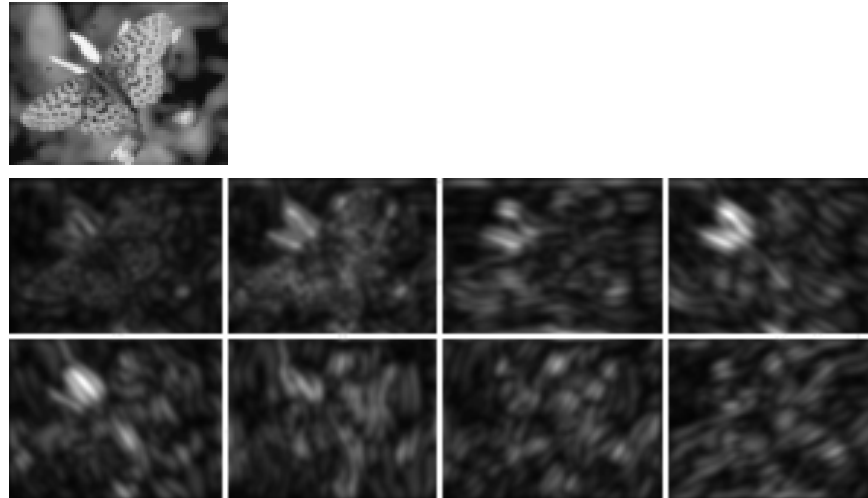


Squared response



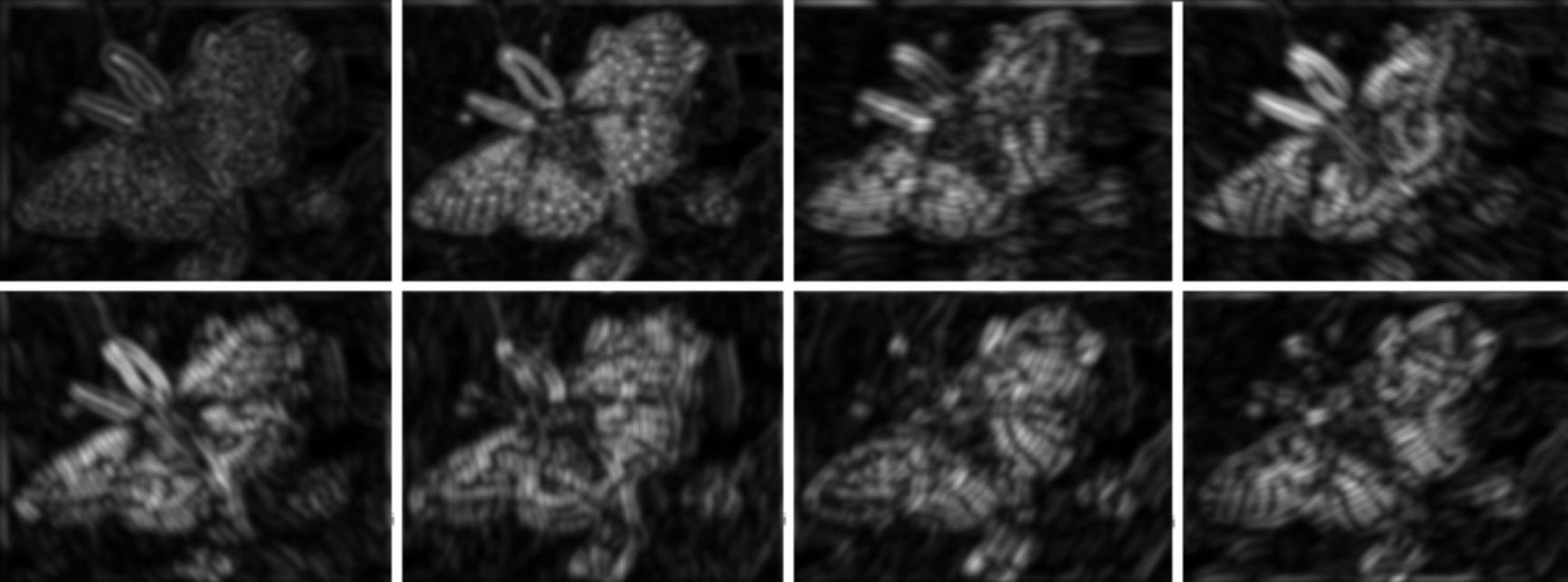


What happens if scale changes?



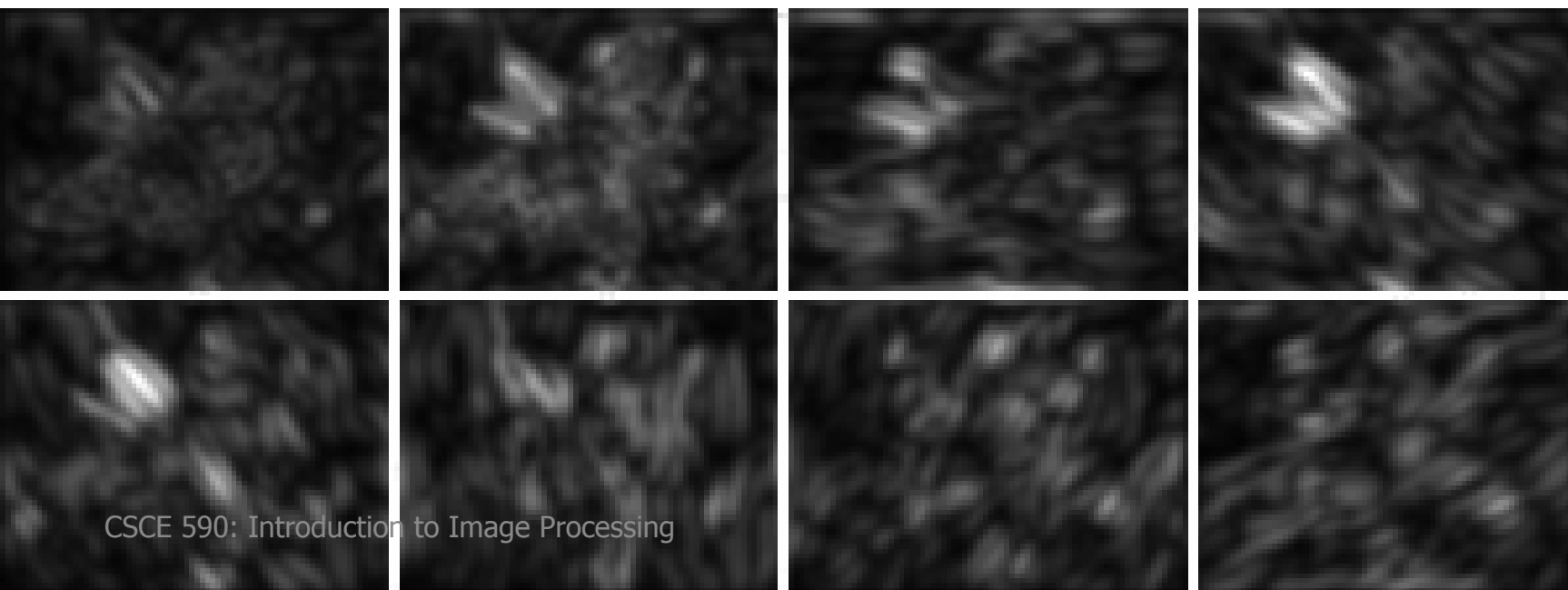
squared response





Fine

Slides courtesy of Prof. Yan Tong



Coarse

Final Texture Representation

- Form an oriented pyramid (or equivalent set of responses to filters at different scales and orientations).
- Square the output
- Take statistics of responses
 - mean of each filter output
 - Standard deviation of each filter output



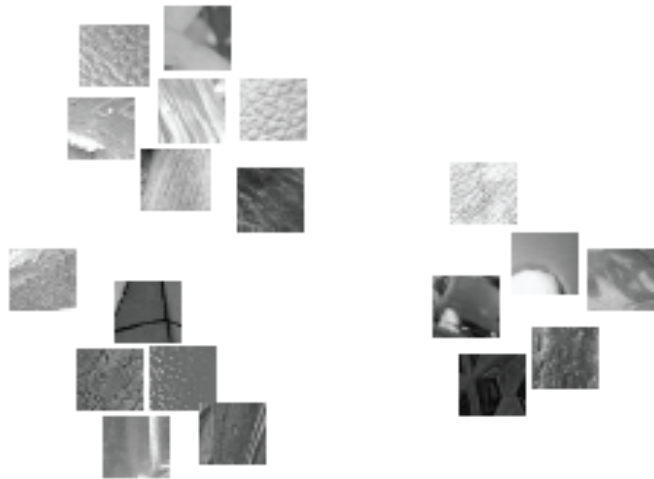
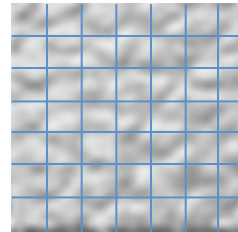
Example based Texture Representations

- How does one choose the filters?
- Solution
 - build a dictionary of subelements from pictures
 - describe the image using this dictionary

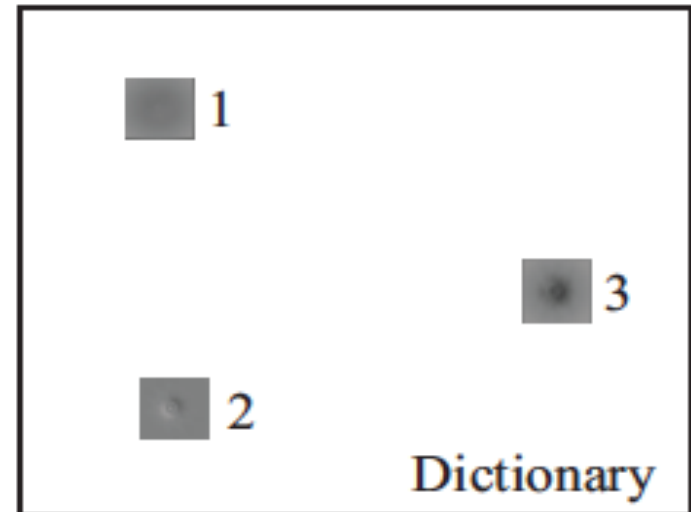


Building a Dictionary

Cut region into patches



Cluster
→



Learning a dictionary



Clustering the Examples

- K-means
 - represent patches with
 - intensity vector
 - vector of filter responses over patch

Choose k data points to act as cluster centers

Until the cluster centers change very little

Allocate each data point to cluster whose center is nearest.

Now ensure that every cluster has at least

one data point; one way to do this is by

supplying empty clusters with a point chosen at random from points far from their cluster center.

Replace the cluster centers with the mean of the elements in their clusters.

end

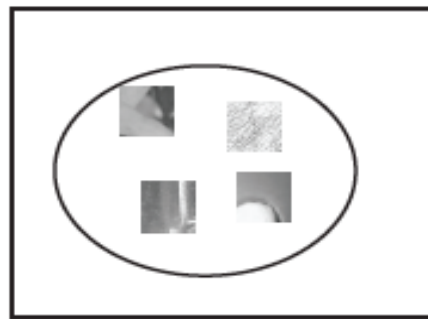
Algorithm 6.3: Clustering by K-Means.



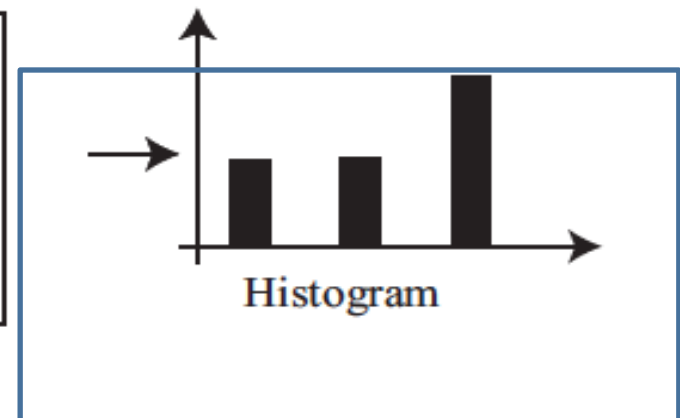
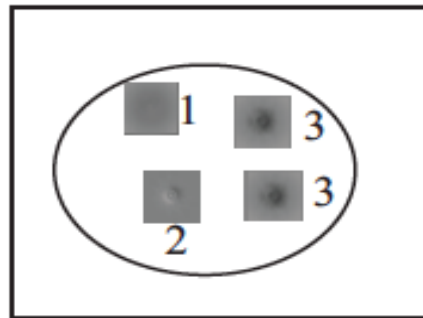
Representing a Region

- Cut region into patches
- Vector Quantization
 - Represent a high-dimensional data item with a single number
 - Find and use the index of the nearest cluster center in dictionary
 - Visual words – a vector of quantized image patches
- Build histogram of resulting numbers

Representing a region



Replace
with
closest
cluster
center



Representing a Region

Build a dictionary:

Collect many training example textures

Construct the vectors x for relevant pixels; these could be a reshaping of a patch around the pixel, a vector of filter outputs computed at the pixel, or the representation of Section 6.1.

Obtain k cluster centers c for these examples

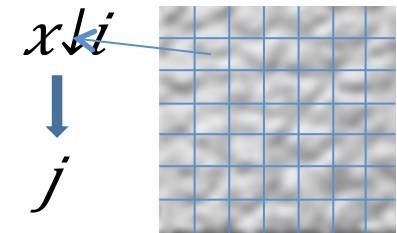
Represent an image domain:

For each relevant pixel i in the image

Compute the vector representation x_i of that pixel

Obtain j , the index of the cluster center c_j closest to that pixel

Insert j into a histogram for that domain



Algorithm 6.2: Texture Representation Using Vector Quantization.



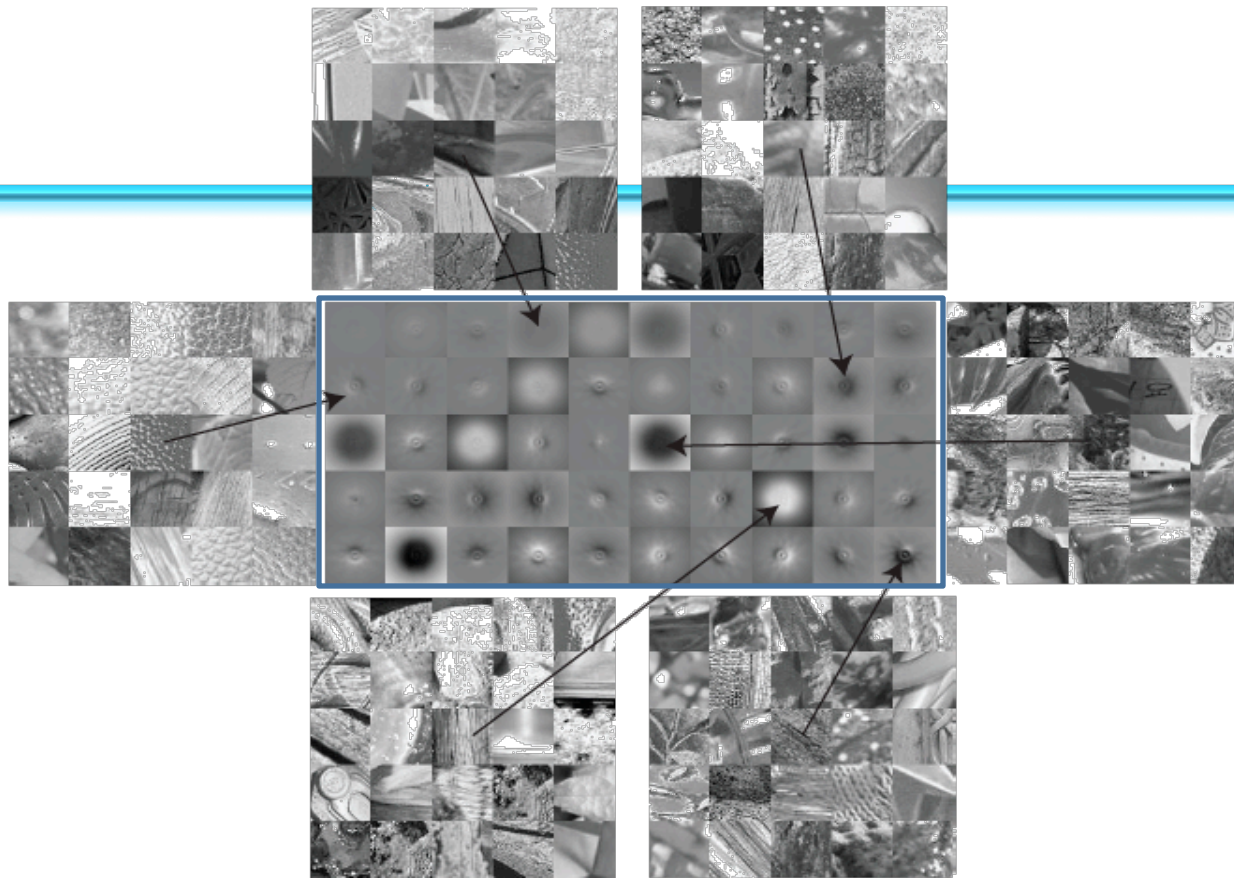


FIGURE 6.9: Pattern elements can be identified by vector quantizing vectors of filter outputs, using k-means. Here we show the top 50 pattern elements (or textons), obtained from all 1,000 images of the collection of material images described in Figure 6.2. These were filtered with the complete set of oriented filters from Figure 6.4. Each subimage here illustrates a cluster center. For each cluster center, we show the linear combination of filter kernels that would result in the set of filter responses represented by the cluster center. For some cluster centers, we show the 25 image patches in the training set whose filter representation is closest to the cluster center. *This figure shows elements of a database collected by C. Liu, L. Sharan, E. Adelson, and R. Rosenholtz, and published at http://people.csail.mit.edu/lavanya/research_sharan.html. Figure by kind permission of the collectors.*



Texture Synthesis

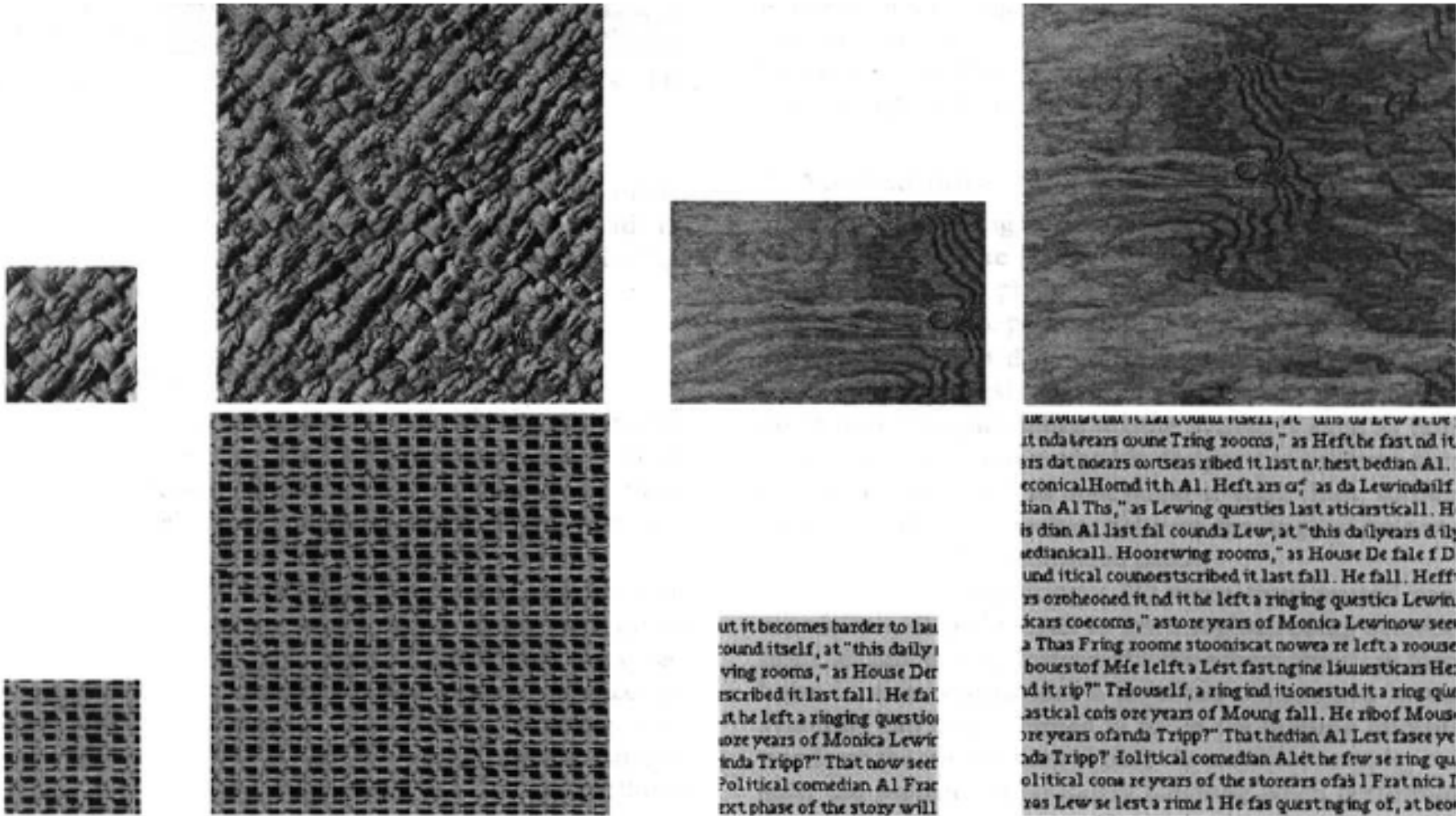
- Problem:
 - Take a small example image of pure texture
 - Use this to produce a large domain of “similar” texture
- Why:
 - Computer graphics demands lots of realistic texture
 - Fill in holes in images created by removing objects



Texture synthesis

- Simple case -- Fill holes
 - Synthesize a single pixel in a large image
 - Approach:
 - Match the window around that pixel to other windows in the image
 - Choose a value from the matching windows
 - most likely, uniformly and at random
- Expand to large images
 - Start: take a piece of the example image
 - Fill in pixels on the boundary
 - Each time you fill in a pixel, you can use that to match

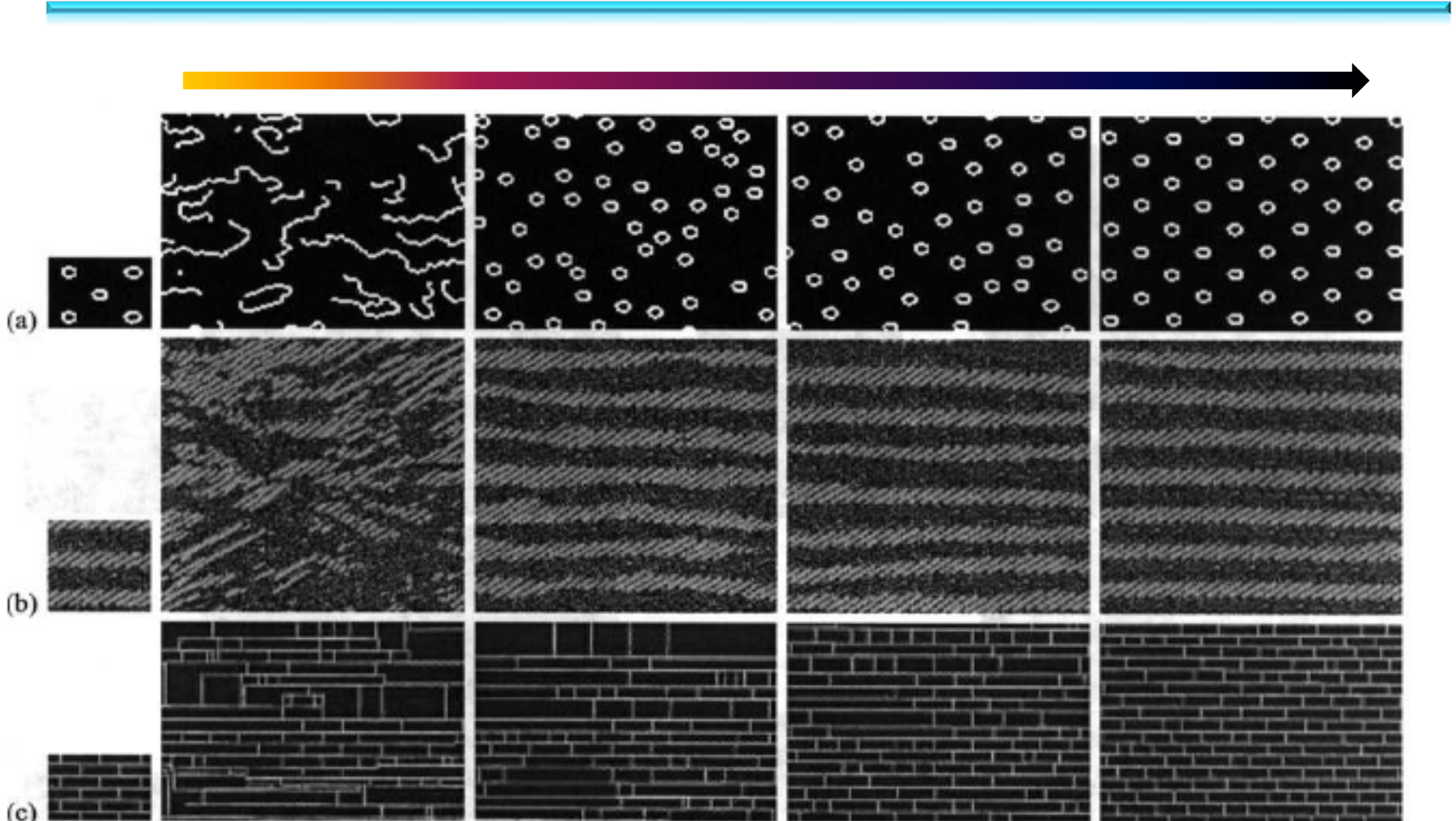




Small blocks are examples, large are synthesized. Notice how (for example) synthesized text looks like actual text.



Neighborhood size







Initial Image



Object masked out



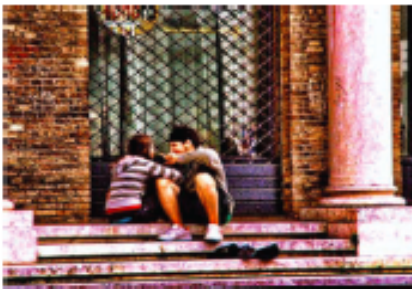
Initial Image



Object masked out



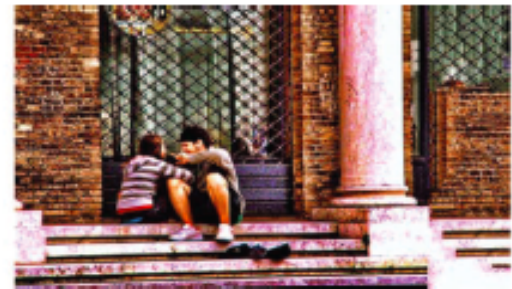
Object composited back



Initial Image



Hole



Extended by hole filling

State-of-the-art in image fill-in combines texture synthesis, coherence, and smoothing by Bugeau et al., IEEE

Questions?

