texture
examples of natural texture

[Malik et al. 2001]
stochastic vs. structured

[Wei and Levoy]
defining texture

- when/why do we perceive textures to be equivalent ("made of the same stuff")?

key idea: beyond simple pixel differences
applications of texture models

robust image segmentation

[Malik et al. 2001]
applications of texture models

object recognition

[Lazebnik et al. 2004]
applications of texture models

“3D shape from texture”
applications of texture models

texture synthesis

[Efros and Leung 1999] [Praun et al. 2000]
defining texture

✦ idea: texture is set of simple components along with their statistics
✦ zebra: “vertical stripes of varying width”
✦ sponge: “many craters of different sizes”
defining texture

✦ what are the fundamental components? (i.e., “texture alphabet”)
✦ experiments suggest dots and oriented edges that can be described by filter responses

[Malik & Perona 1990]
defining texture

[Malik & Perona 1990]
defining texture

• determine statistics
• average response over local region, etc.
image pyramids

Idea: Represent NxN image as a “pyramid” of 1x1, 2x2, 4x4, ..., \(2^k \times 2^k\) images (assuming \(N=2^k\))

- level \(k\) (= 1 pixel)
- level \(k-1\)
- level \(k-2\)
- level 0 (= original image)

[Szeliski]
Gaussian pyramid

\[ P_G(I)_{k+1} = S^\downarrow(B(P_G(I)_k)) \]
Laplacian pyramid

\[ P_L(I)_k = P_G(I)_k - S^\uparrow(P_G(I)_{k+1}) \]

Image from [Forsyth & Ponce]
Laplacian pyramid

\[ P_L(I)_k = P_G(I)_k - S^{\uparrow}(P_G(I)_{k+1}) \]

\[ P_G(I)_k = P_L(I)_k + S^{\uparrow}(P_G(I)_{k+1}) \]

working from coarsest to finest:
convert laplacian pyramid to gaussian pyramid

punch line: this is invertible

[Burt and Adelson 1981]
octaves in spatial domain

lowpass images

bandpass images

[Szeliski]
steerable pyramid

Laplacian pyramid layer

oriented pyramid levels

[Heeger and Bergen 1995]
steerable pyramid

hypothesis: “components” of texture captured in these orientation/scale subbands

[Heeger and Bergen 1995]
texture model

compile histogram of intensities output by each filter

[Heeger and Bergen 1995]
distance between two textures is defined as distance between these histograms:

\[ \text{Heeger and Bergen 1995} \]
texture synthesis

- start from white noise image
- adjust histograms to match response of target texture
- re-synthesize image from filter outputs (remember transform is invertible)
Given: two intensity histograms \( H_1 \) and \( H_2 \)

Goal: function that remaps intensities to make new histograms \( H_1' \) equal \( H_2 \)
histogram equalization

- compute CDF of histograms

- for each intensity, map through CDF 1 then lookup inverse in CDF 2
texture synthesis

original (exemplar)

synthesized texture

[Heeger and Bergen 1995]
textons

✦ elements ("textons") either identical or come from some statistical distribution
✦ can analyze in natural images

[Olhausen and Field]
textons

- output of bank of n filters; treat as point in n-dimensional space

- can cluster these responses using $k$-means [Malik et al. 2001]

- result: dictionary of most common textures

[Malik et al. 2001]
textons

image

clustered textons

texton to pixel mapping

[Malik et al. 2001]
using texture in segmentation

• compute histogram of how many times each of the $k$ clusters occurs in a neighborhood

• define similarity of histograms:

$$
\chi^2 = \frac{1}{2} \sum_k \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}
$$
texture segmentation
Markov Random Fields

- different way of thinking about textures
- premise: probability distribution of a pixel depends on values of neighbors
- probability the same throughout image
  - extension of Markov chains
probability of an image

\[ P(X = x) \]
joint probability of two pixels

\[ P(X = x) = P(P_1 = p_1, P_2 = p_2) \]
probability of an image

\[ P(X = x) \]

\[ P(P_1 = p_1, P_2 = p_2, \ldots, P_N = p_n) \]

50x50 images, 8-bit grayscale:

\[ \text{table size} = (255)^{(50 \times 50)} \]
probability of an image

**key** simplifying assumption:

joint distribution can be factored into product of many (much smaller) conditional distributions if we assume independence
Markov Random Field (MRF)

model of joint probability distribution function

rv dependence
\[ P(x_{ij} | x_{i-j}^+, x_{ij}^+, x_{i+j}^+, \cdots) \]
\[ P(X = x) = \prod_{ij}^N P(x_{ij} | \cdots) \]
probability of an image

50x50 images, 8-bit grayscale:

\[ P \left( P_1 = p_1, \cdots , P_N = p_N \right) \]

\[ P(X = x) = \prod_{ij}^N P(x_{ij} | \cdots ) \]

table size

\[ (255)^{(N)} \]

\[ (255)^{(8)} \]
probability of an image

\[ P(X = x) = \prod_{ij} P(x_{ij} | \cdots) \]

“markovian” texture models approximate single-pixel regions of \( P(X) \) on-the-fly
non-parametric sampling

- used for texture synthesis