Lecture 7
Textures
What is a texture?
Which textures are we going to talk about in this lecture?

Stationary
Stochastic
When are two textures similar?

All these images are different instances of the same texture. We can differentiate between them, but they seem generated by the same process.
Texture Analysis

Compare textures and decide if they’re made of the same “stuff”.

True (infinite) texture

input image

ANALYSIS

generated image

“Same” or “different”
Given a finite sample of some texture, the goal is to synthesize other samples from that same texture. The sample needs to be "large enough"
Let’s get a feeling of the mechanisms for texture perception
What is special about texture perception?

• Pre-attentive texture discrimination
• Perception of sets and summary statistics
• Crowding
Research with texture pairs having identical second-order statistics has revealed that the pre-attentive texture discrimination system cannot globally process third- and higher-order statistics, and that discrimination is the result of a few local conspicuous features, called textons. It seems that only the first-order statistics of these textons have perceptual significance, and the relative phase between textons cannot be perceived without detailed scrutiny by focal attention.
Pre-attentive texture discrimination

Pre-attentive texture discrimination

Pre-attentive texture discrimination

This texture pair is pre-attentively indistinguishable. Why?

The uncrowded window of object recognition

Denis G Pelli & Katharine A Tillman
A summary-statistic representation in peripheral vision explains visual crowding

Benjamin Balas 1,
Lisa Nakano 2 and
Ruth Rosenholtz 3
SEEING SETS:
Representation by Statistical Properties

Dan Ariely
Massachusetts Institute of Technology
Representation of sets
Set

Is this element a member of the set?
Ebbinghaus illusion

The central circle is judged relative to the set properties of the circles surrounding it.

Attenuated by reducing the set grouping.
Representation
What a model should account for:

1. **Biological plausibility**: The stages of the model should be motivated by, and be consistent with, known physiological mechanisms of early vision.

2. **Generality**: The model should be general enough that it can be tested on any arbitrary gray-scale image.

3. **Quantitative match with psychophysical data**: The model should make a quantitative prediction about the salience of the boundary between any two textured regions. Rank ordering of the discriminability of different texture pairs should agree with that measured psychophysically.

From Malik & Perona, 1990
Julesz - Textons

Textons: fundamental texture elements.

Textons might be represented by features such as terminators, corners, and intersections within the patterns…
“We note here that simpler, lower-level mechanisms tuned for size may be sufficient to explain this discrimination.”
Early vision and texture perception

James R. Bergen* & Edward H. Adelson**

Ls 25% larger
contrast adjusted to keep mean constant

Ls 25% shorter
Preattentive texture discrimination with early vision mechanisms

Jitendra Malik and Pietro Perona

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley,
Berkeley, California 94720

Received July 7, 1989; accepted December 28, 1989

We present a model of human preattentive texture perception. This model consists of three stages: (1) convolution of the image with a bank of even-symmetric linear filters followed by half-wave rectification to give a set of responses modeling outputs of V1 simple cells, (2) inhibition, localized in space, within and among the neural-response profiles that results in the suppression of weak responses when there are strong responses at the same or nearby locations, and (3) texture-boundary detection by using wide odd-symmetric mechanisms. Our model can predict the salience of texture boundaries in any arbitrary gray-scale image. A computer implementation of this model has been tested on many of the classic stimuli from psychophysical literature. Quantitative predictions of the degree of discriminability of different texture pairs match well with experimental measurements of discriminability in human observers.
Threshold squared, blurred responses, then categorize texture based on those two bits.
Fig. 1. Simplified schematics of our model for texture perception. The image (bottom) is filtered using the kernels $F_1 \ldots F_m$ and is half-wave rectified to give the set of simple-cell responses $R_1 \ldots R_n$. The postinhibition responses $\text{PIR}_1 \ldots \text{PIR}_n$ are computed by thresholding the $R_i$ and taking the maximum of the result over small neighborhoods. The thresholds depend on the activity of all channels. The texture gradient is computed by taking the maximum of the responses of wide odd-symmetric filters acting on the postinhibition responses $\text{PIR}_i$. 
Two big families of models

1- Parametric models of filter outputs

2- Example-based non-parametric models
The trivial texture synthesis algorithm
Texture synthesis and representation

Set of equivalent textures: generated by exactly the same physical process
Texture synthesis and representation

Set of equivalent textures: generated by exactly the same physical process

Set of perceptually equivalent textures: “well, they just look the same to me”
If matching the averaged squared filter values is a good way to match a given texture, then maybe matching the entire marginal distribution (eg, the histogram) of a filter’s response would be even better.

Jim Bergen proposed this...
Pyramid-Based Texture Analysis/Synthesis

David J. Heeger
Stanford University

James R. Bergen
SRI David Sarnoff Research Center

SIGGRAPH 1994
The main idea: it works by ‘kind of’ projecting a random image into the set of equivalent textures.
Overview of the algorithm

Match-texture(noise,texture)

```
Match-Histogram (noise,texture)
```

analysis-pyr = Make-Pyramid (texture)

Loop for several iterations do

```
synthesis-pyr = Make-Pyramid (noise)
```

Loop for a-band in subbands of analysis-pyr

```
for s-band in subbands of synthesis-pyr
```

```
do
```

```
Match-Histogram (s-band,a-band)
```

```
n = Collapse-Pyramid (synthesis-pyr)
```

```
noise = Collapse-Pyramid (synthesis-pyr)
```

Match-Histogram (noise,texture)

Two main tools:

1- steerable pyramid

2- matching histograms
1-The steerable pyramid
1-The steerable pyramid

But why do I want to represent images like this?
1-The steerable pyramid

Argument used by H & B: Statistical measures in the subband representation seem to provide a “distance” between textures that correlates with human perception better than pixel-based representations.
1-The steerable pyramid

In general seems a good idea to have a representation that:

- Preserves all image information (we can go back to the image)
- Provides more independent channels of information than pixel values (we can mess with each band independently)

But all this is just indirectly related to the texture synthesis task. But let assume is good enough...
1-The steerable pyramid
Overview of the algorithm

Match-texture(noise, texture)

- Match-Histogram (noise, texture)

analysis-pyr = Make-Pyramid (texture)

Loop for several iterations do

- synthesis-pyr = Make-Pyramid (noise)

Loop for a-band in subbands of analysis-pyr

for s-band in subbands of synthesis-pyr do

- Match-Histogram (s-band, a-band)

noise = Collapse-Pyramid (synthesis-pyr)

Match-Histogram (noise, texture)

Two main tools:

1- steerable pyramid

2- matching histograms
2-Matching histograms

9% of pixels have an intensity value within the range $[0.37, 0.41]$

75% of pixels have an intensity value smaller than 0.5

5% of pixels have an intensity value within the range $[0.37, 0.41]$
2-Matching histograms

Z(x,y)

We look for a transformation of the image Y

Y’ = f(Y)

Such that

Hist(Y) = Hist(f(Z))

Problem: there are infinitely many functions that can do this transformation.

A natural choice is to use f being:
- pointwise non linearity
- stationary
- monotonic (most of the time invertible)
2-Matching histograms

The function \( f \) is just a look up table: it says, change all the pixels of value \( Y \) into a value \( f(Y) \).

\[
Y' = f(Y)
\]

\( Y(x,y) \)

\( Y = 0.8 \)

Original intensity

\( Y' = 0.5 \)

New intensity
2-Matching histograms

\[ Y' = f(Y) \]
Another example: Matching histograms

10% of pixels are black and 90% are white

5% of pixels have an intensity value within the range [0.37, 0.41]
Another example: Matching histograms

The function $f$ is just a look up table: it says, change all the pixels of value $Y$ into a value $f(Y)$.

$$Y' = f(Y)$$

Original intensity

New intensity

$Y(x,y)$

$Y = 0.8$

$Y' = 1$
Another example: Matching histograms

$Y' = f(Y)$

In this example, $f$ is a step function.
Matching histograms of a subband
Matching histograms of a subband

\[ Y' = f(Y) \]
Texture analysis

The texture is represented as a collection of marginal histograms.

Wavelet decomposition (steerable pyr)

(Steerable pyr; Freeman & Adelson, 91)
Texture synthesis

Heeger and Bergen, 1995

Input texture

(histogram)

(histogram)
Why does it work? (sort of)
Why does it work? (sort of)

The black and white blocks appear by thresholding (f) a blobby image.
Why does it work? (sort of)

The black and white blocks appear by thresholding (f) a blobby image
Why does it work? (sort of)

After 6 iterations

Histograms match ok

red = target histogram, blue = current iteration
Color textures

Three textures

R

G

B

Original texture
Color textures

Original texture

R

G

B
Color textures

This does not work

Original texture
Color textures

Problem: we create new colors not present in the original image.

Why? Color channels are not independent.
PCA and decorrelation

In the original image, R and G are correlated, but, after synthesis,...
PCA and decorrelation

The texture synthesis algorithm assumes that the channels are independent.
What we want to do is some rotation

See that in this rotated space, if I specify one coordinate the other remains unconstrained.
PCA and decorrelation

PCA finds the principal directions of variation of the data. It gives a decomposition of the covariance matrix as:

\[ C = D D' \]

By transforming the original data (RGB) using \( D \) we get:

\[ \begin{align*}
\begin{bmatrix}
U1 \\
U2 \\
U3 
\end{bmatrix} &= 
\begin{bmatrix}
0.6347 & 0.6072 & 0.4779 \\
0.6306 & -0.0496 & -0.7745 \\
0.4466 & -0.7930 & 0.4144 
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B 
\end{bmatrix} 
\end{align*} \]

The new components (U1,U2,U3) are decorrelated.
Color textures

These three textures look similar (high dependency)

These three textures look less similar (lower dependency)
Color textures

Inverse Rotation Matrix

R

G

B

Original texture
Color textures

These three textures look similar (high dependency)

These three textures look less similar (lower dependency)
Color channels

Without PCA

With PCA
Color channels

Original texture

Synthetic texture

Synthetic texture
Color channels
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
Examples from the paper

Figure 4: In each pair left image is original and right image is synthetic: red gravel, figured sepele wood, broccoli, bark paper, denim, pink wall, ivy, grass, sand, surf.
Examples not from the paper

But, does it really work even when it seems to work?
But, does it really work???
How to measure how well the representation constraints the set of equivalent textures?

All the textures in this set have the same parameters.
How to identify the set of equivalent textures?

This does not reveal how poor the representation actually is.
We need a space that is more perceptual.

In a perceptual space all these noise images are very close. But in pixel space, they are very far away.

How big is this set in a pixels space?
We need a space that is more perceptual

In a perceptual space all these noise images are very close. But in pixel space, they are very far away.

How big is this set in a pixels space?

How big is this set in a perceptual space?
How to identify the set of equivalent textures?

These trajectories are more perceptually salient.

This set is huge.
How to identify the set of equivalent textures?
How to identify the set of equivalent textures?

These trajectories are more perceptually salient
Portilla and Simoncelli

- Parametric representation, based on Gaussian scale mixture prior model for images.
- About 1000 numbers to describe a texture.
- Ok results; maybe as good as DeBonet.
Portilla and Simoncelli
Portilla & Simoncelli

Heeger & Bergen

Portilla & Simoncelli
Now they look good, but maybe they look too good…

How to identify the set of equivalent textures?

Portilla & Simoncelli
A summary-statistic representation in peripheral vision explains visual crowding

Benjamin Balas ¹,
Lisa Nakano ² and
Ruth Rosenholtz ³
• Principled approach. Based on an assumption of heavy-tailed distributions for an over-complete set of filters.
• Synthesis quality not great, but ok.
Zhu, Wu, & Mumford

- Cheetah
- Synthetic
De Bonet (and Viola)  
SIGGRAPH 1997

Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images

Jeremy S. De Bonet - Learning & Vision Group  
Artificial Intelligence Laboratory  
Massachusetts Institute of Technology

EMAIL: jsd@ai.mit.edu  
HOMEPAGE: http://www.ai.mit.edu/~jsd
DeBonet

Learn: use filter conditional statistics across scale.

Figure 8: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the “parent” structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

Figure 9: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.
Two big families of models

1- Parametric models of filter outputs

2- Example-based non-parametric models
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
Efros & Leung Algorithm

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

- Building explicit probability tables 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
Neighborhood Window

input
Varying Window Size

Increasing window size
Synthesis Results

french canvas

rafia weave
More Results

white bread

brick wall
Homage to Shannon
Hole Filling
Extrapolation
**Image Quilting [Efros & Freeman]**

**Observation:** neighbor pixels are highly correlated

**Idea:** unit of synthesis = block

- Exactly the same but now we want $P(B|N(B))$
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!
Input texture

Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut
Minimal error boundary

overlapping blocks

vertical boundary

overlap error

min. error boundary
Texture Transfer

• Take the texture from one object and “paint” it onto another object
  – This requires separating texture and shape
  – That’s HARD, but we can cheat
    – Assume we can capture shape by boundary and rough shading

• Then, just add another constraint when sampling: similarity to underlying image at that spot
parmesan + rice =

parmesan + rice =

parmesan + rice =

parmesan + rice =
Project ideas
Non stationary texture synthesis
Project ideas: 3D textures
Project ideas: 3D textures

Can you create a 3D volume that you can navigate? Assume that all slices should have the same statistics. Need knowledge about alpha map?

This is not a solid texture. This is a 3D scene texture.