Texture synthesis and image analogies
Course announcements

• Please take Doodle for second make-up lecture, link on Piazza.

• Homework 3 is out.
  - Due October 12th.
  - Shorter, but longer bonus component.
Overview of today’s lecture

• Reminder: non-local means.
• Texture synthesis.
• Texture by non-parametric sampling.
• Image quilting.
• Inpainting.
• Texture transfer.
• Image analogies.
• Deep learning teaser.
Slide credits

Most of these slides were adapted from:

• Kris Kitani (15-463, Fall 2016).

Some slides were inspired or taken from:

• Fredo Durand (MIT).
• James Hays (Georgia Tech).
Reminder: non-local means
Redundancy in natural images
Non-local means

No need to stop at neighborhood. Instead search everywhere in the image.

\[
\hat{x}(i) = \frac{1}{C_i} \sum_j y(j) e^{-\frac{SSD(y(N_i) - y(N_j))}{2\sigma^2}}
\]
Last couple of classes: adding things to the image
This class: removing things from the image
This class: removing things from the image
Texture synthesis
Texture

- Depicts spatially repeating patterns
- Appears naturally and frequently
Texture

- Large variety of textures
Texture synthesis

Goal: create new samples of a given texture.
Applications:
• hole filling
• virtual environments
• view expansion
• texturing surfaces
• ....
How would you do texture synthesis for this sample?
How would you do texture synthesis for this sample?

Input

tiling

random
Approach 1: probabilistic modeling

Basic idea:
• Compute statistics of input texture (e.g., histogram of edge filter responses).
• Generate a new texture that keeps these same statistics.

Heeger and Bergen, “Pyramid-based texture analysis/synthesis,” SIGGRAPH 1995
Approach 1: probabilistic modeling

Probability distributions are hard to model well.

Any other ideas?
Texture by non-parametric sampling
Approach 2: sample from the image

Run template matching, get N best matches, and sample one at random.

What are sampling from?

Efros and Leung, “Texture synthesis by non-parametric sampling,” ICCV 1999
Approach 2: sample from the image

Run template matching, get N best matches, and sample one at random.

- Similar nearby images define a non-parametric PDF $P(p|N(p))$
- By selecting a random sample, we are sampling from this PDF
Implementation details

How do you define patch similarity?
How do you define patch similarity?

• Gaussian-weighted SSD (emphasis on nearby pixels).

In what order should you synthesize?
Implementation details

How do you define patch similarity?
- Gaussian-weighted SSD (emphasis on nearby pixels).

In what order should you synthesize?
- Onion-peel ordering – pixels with most neighbors are synthesized first.

How do you synthesize from scratch?
Implementation details

How do you define patch similarity?
- Gaussian-weighted SSD (emphasis on nearby pixels).

In what order should you synthesize?
- Onion-peel ordering – pixels with most neighbors are synthesized first.

How do you synthesize from scratch?
- Pick a small patch at random from source.
Ideas from information theory

- Generate English-sounding sentences by modeling the probability of each word given the previous words (n-grams)

- Large “n” will give more structured sentences

“I spent an interesting evening recently with a grain of salt.”

Claude Elwood Shannon
(1916–2001)
Size of neighborhood window matters a lot
Size of neighborhood window matters a lot
Texture synthesis algorithm

While image not filled

1. Get unfilled pixels with filled neighbors
2. Sort by number of filled neighbor
3. For each pixel
   a) Get top N matches of visible neighbor (Patch Distance: Gaussian-weighted SSD)
   b) Randomly select one of the matches
   c) Copy pixel value
Examples

French canvas

raﬁa weave
Examples

white bread

brick wall
Homage to Shannon
Hole filling
Image extrapolation
Texture synthesis using non-parametric sampling:

- Very simple
- Surprisingly good results
- Synthesis is easier than analysis!
- But very slow

Why is it so slow and how could we make it faster?
Image quilting
Observation: neighboring pixels are highly correlated.

Idea: Instead of single pixels, synthesize entire blocks
- Exactly analogous procedure as before, except we now sample $P(B \mid N(B))$
- Much faster since we synthesize all pixels in a block at once

Dealing with boundaries

random placement of blocks

neighboring blocks constrained by overlap
Dealing with boundaries

input texture

block

random placement of blocks

neighboring blocks constrained by overlap

minimal error boundary cut

How can we achieve this?
Dealing with boundaries

How can we compute this boundary efficiently?
Examples
Examples
Examples
Examples
Examples
Examples
Failure case (Chernobyl tomatoes)
Examples

input image

Portilla & Simoncelli

Xu, Guo & Shum

Wei & Levoy

Quilting
Examples

Portilla & Simoncelli
Xu, Guo & Shum

Wei & Levoy
Quilting
It even made the news
Inpainting
Inpainting natural scenes

Key idea: Filling order matters

Toy inpainting example:

- image with hole
- raster-scan order
- onion-peel

Any ideas on how to do better filling?
Key idea: Filling order matters

Toy inpainting example:

- image with hole
- raster-scan order
- onion-peel
- gradient-sensitive order

**Gradient-sensitive order**: Fill a pixel that
- is surrounded by other known pixels; **and**
- is a continuation of a strong gradient or edge.
Examples

original

with hole

onion-peel fill

gradient-sensitive
Examples

onion-peel

gradient-sensitive
Texture transfer
Texture transfer

Try to explain one object with bits and pieces of another object

How would you do this?

Texture transfer

Same as texture synthesis, except search for texture blocks by comparing with target image patches ("constraints").
Some less creepy examples
Some less creepy (?) examples
Some less creepy examples
Image analogies
Image analogies

Why stop at textures?

given pair of image analogies

input image

synthesized image

Hertzmann et al., “Image analogies, ” SIGGRAPH 2001
Image analogies
How would you do this?
How would you do this?

Implementation:

Define a similarity between A and B

For each patch in B:
1. Find a matching patch in A, whose corresponding A’ also fits in well with existing patches in B’
2. Copy the patch in A’ to B’

Algorithm is run iteratively (coarse-to-fine)
Blurring by analogies

unfiltered source (A)

filtered source (A’)

unfiltered target (B)

filtered target (B’)

Edges by analogies

unfiltered source (A)

filtered source (A’)

unfiltered target (B)

filtered target (B’)

[Images of various scenes and filters]
Artistic filters

unfiltered source (A)

filtered source (A')

unfiltered target (B)

filtered target (B')
Colorization

unfiltered source (A)

filtered source (A')

unfiltered target (B)

filtered target (B')
“Texture by numbers”
“Texture by numbers”
Super-resolution

unfiltered source (A)

filtered source (A’)

unfiltered target (B)

filtered target (B’)

Super-resolution

unfiltered target (B)

filtered target (B’)

Deep learning teaser
A return to parametric models

\[ E_L = \sum (G^L - \hat{G}^L)^2 \]

\[ G^L = \sum_k \hat{F}^k \hat{f}^k \]

\[ \hat{F}^L \]

\[ \hat{F}^{L-1} \]

\[ \frac{\partial E_L}{\partial \hat{F}^L} \]

\[ \frac{\partial E_L}{\partial \hat{F}^{L-1}} \]

\[ \frac{\partial L}{\partial \hat{x}} \]

Gradient descent

\[ \hat{x} := \hat{x} - \alpha \frac{\partial L}{\partial \hat{x}} \]

Step 1: forward pass
Input image

Step 2: define loss wrt forward pass responses

Step 3: update white noise image according to gradient descent
Texture synthesis examples
Texture synthesis examples

Synthesised

Source

Synthesised

Source
Texture synthesis examples
Texture synthesis examples
Texture synthesis examples

Synthesised

Source

Synthesised

Source
Texture synthesis examples
Parameter number matters
Style transfer examples
References

Basic reading:
- Szeliski textbook, Section 10.5.
the titles of the above four papers should be self-explanatory.

Additional reading:
texture synthesis using deep learning.
implementing image analogies using deep learning.
implementing photo-realistic style transfer using deep learning.