Texture Synthesis and Hole-Filling

Computational Photography
Derek Hoiem, University of Illinois
Project 1

• Results page is up

• Aim to have it graded in the next few days (but there are lots to grade)
Next section: The digital canvas

Cutting and pasting objects, filling holes, and blending

Image warping and object morphing
Today’s Class

• Texture synthesis and hole-filling
Texture

- Texture depicts spatially repeating patterns
- Textures appear naturally and frequently

radishes
rocks
yogurt

Many slides from James Hays
Texture Synthesis

- Goal of Texture Synthesis: create new samples of a given texture
- Many applications: virtual environments, hole-filling, texturing surfaces
The Challenge

Need to model the whole spectrum: from repeated to stochastic texture
One idea: Build Probability Distributions

Basic idea
1. Compute statistics of input texture (e.g., histogram of edge filter responses)
2. Generate a new texture that keeps those same statistics

One idea: Build Probability Distributions

But it (usually) doesn’t work

• Probability distributions are hard to model well
Another idea: Sample from the image

• 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

Efros and Leung 1999 SIGGRAPH
Idea from Shannon (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.”
(example from fake single.net user Mark V Shaney)
Details

• How to match patches?
  – Gaussian-weighted SSD (more emphasis on nearby pixels)

• What order to fill in new pixels?
  – “Onion skin” order: pixels with most neighbors are synthesized first
  – To synthesize from scratch, start with a randomly selected small patch from the source texture

• How big should the patches be?
Size of Neighborhood Window
Varying Window Size

Increasing window size
Texture synthesis algorithm

• While image not filled
  1. Get unfilled pixels with filled neighbors, sorted by number of filled neighbors
  2. For each pixel, get top N matches based on visible neighbors
     - Patch Distance: Gaussian-weighted SSD
  3. Randomly select one of the matches and copy pixel from it
Synthesis Results

french canvas

rafia weave
More Results

white bread

brick wall
Homage to Shannon
Hole Filling
Extrapolation
In-painting natural scenes

Key idea: Filling order matters

In-painting Result

Image with Hole
Raster-Scan Order
Onion-Peel (Concentric Layers)
Gradient-Sensitive Order

Filling order

Fill a pixel that:

1. Is surrounded by other known pixels
2. Is a continuation of a strong gradient or edge

Comparison

Comparison

Concentric Layers

Gradient Sensitive
Summary

• The Efros & Leung texture synthesis algorithm
  – Very simple
  – Surprisingly good results
  – Synthesis is easier than analysis!
  – ...but very slow
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
Input texture

Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut
Minimal error boundary

overlapping blocks

vertical boundary

overlap error

2

min. error boundary
Solving for Minimum Cut Path

Cost of a cut through this pixel

1  3  4  1
2  1  2  3
4  2  1  4
Solving for Minimum Cut Path

1. prev = r1, cost = 4
2. prev = r1, cost = 2
3. prev = r2, cost = 4
4. cost = 6
5. cost = 4
6. cost = 3
7. cost = 7
8. cost = 5
9. cost = 6
10. cost = 7
Solving for Minimum Cut Path

Best Path

1 → 3 → 4

cost = 5

cost = 6

cost = 7
Solving for Minimum Cut Path

Region 1

Region 2

Mask Based on Best Path
Portilla & Simoncelli

Xu, Guo & Shum

Wei & Levoy

Quilting

input image
Political Texture Synthesis!

Bush campaign digitally altered TV ad

President Bush’s campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.

This section shows a sampling of the duplication of soldiers.

Original photograph
Texture Transfer

- Try to explain one object with bits and pieces of another object:
Texture Transfer

Constraint

Texture sample
Texture Transfer

Take the texture from one image and “paint” it onto another object

Same as texture synthesis, except an additional constraint:
1. Consistency of texture
2. Patches from texture should correspond to patches from constraint in some way. Typical example: blur luminance, use SSD for distance
Making sacred toast

[Image of a piece of toast]

+ [Image of a drawing of the Virgin Mary]

? [Image of a grilled cheese sandwich]

Project 2: texture synthesis and transfer

- https://courses.engr.illinois.edu/cs445/fa2015/projects/quiltin/g/ComputationalPhotography_ProjectQuilting.html
- Note: this is significantly more challenging than the first project
Texture Synthesis and Transfer Recap

For each overlapping patch in the output image

1. Compute the cost to each patch in the sample
   - Texture synthesis: this cost is the SSD (sum of square difference) of pixel values in the overlapping portion of the existing output and sample
   - Texture transfer: cost is \( \alpha \times SSD_{overlap} + (1 - \alpha) \times SSD_{transfer} \)  The latter term enforces that the source and target correspondence patches should match.

2. Select one sample patch that has a small cost

3. Find a cut through the left/top borders of the patch based on overlapping region with existing output
   - Use this cut to create a mask that specifies which pixels to copy from sample patch

4. Copy masked pixels from sample image to corresponding pixel locations in output image
PatchMatch

More efficient search:
1. Randomly initialize matches
2. See if neighbor's offsets are better
3. Randomly search a local window for better matches
4. Repeat 3, 4 across image several times

Reconstructing top-left image with patches from bottom-left image

Applications to hole-filling, retargeting; constraints can guide search

Barnes et al. Siggraph 2009
Related idea: Image Analogies

Image Analogies, Hertzmann et al. SG 2001
Image analogies

- Define a similarity between A and B
- For each patch in B:
  - Find a matching patch in A, whose corresponding A’ also fits in well with existing patches in B’
  - Copy the patch in A’ to B’
- Algorithm is done iteratively, coarse-to-fine
Blur Filter

Unfiltered source ($A$)  Filtered source ($A'$)

Unfiltered target ($B$)  Filtered target ($B'$)
Edge Filter

- Unfiltered source ($A$)
- Filtered source ($A'$)
- Unfiltered target ($B$)
- Filtered target ($B'$)
Artistic Filters

A

A'

B

B'
Colorization

Unfiltered source ($A$)  Filtered source ($A'$)

Unfiltered target ($B$)  Filtered target ($B'$)
Texture-by-numbers

A

A'

B

B'
Super-resolution

A

A′
Super-resolution (result!)

B

B'
Things to remember

• Texture synthesis and hole-filling can be thought of as a form of probabilistic hallucination

• Simple, similarity-based matching is a powerful tool
  – Synthesis
  – Hole-filling
  – Transfer
  – Artistic filtering
  – Super-resolution
  – Recognition, etc.

• Key is how to define similarity and efficiently find neighbors
Next class

- Cutting and seam finding