Texture synthesis

- **Goal:** 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

Markov chains

• A sequence of random variables $X_1, X_2, \ldots, X_n$
• $X_t$ is the state of the model at time $t$

\[
\begin{align*}
X_1 & \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow X_5
\end{align*}
\]

• Markov assumption: each state is dependent only on the previous one
  - dependency given by a conditional probability:
  \[ p(x_t|x_{t-1}) \]
• The above is actually a first-order Markov chain
• An $N$th-order Markov chain:
  \[ p(x_t|x_{t-1}, \ldots, x_{t-N}) \]

Source: S. Seitz

Markov Chain Example: Text

“A dog is a man’s best friend. It’s a dog eat dog world out there.”

\[
\begin{array}{cccccc}
\text{a} & \text{dog} & \text{is} & \text{man’s} & \text{best} & \text{friend} \\
\text{1/3} & 1/3 & 1/3 & 1/3 & 1/3 & 1/3 \\
\text{p(x_t|x_{t-1})} & & & & & \\
\end{array}
\]

Source: S. Seitz

Text synthesis

Create plausible looking poetry, love letters, term papers, etc.
Most basic algorithm
1. Build probability histogram
   - find all blocks of $N$ consecutive words/letters in training documents
   - compute probability of occurrence $p(x_t|x_{t-1}, \ldots, x_{t-(n-1)})$
2. Given words $X_1, X_2, \ldots, X_{k-1}$
   - compute $X_k$ by sampling from $p(x_t|x_{t-1}, \ldots, x_{t-(n-1)})$

WE NEED TO EAT CAKE

Source: S. Seitz

Text synthesis

• “As I’ve commented before, really relating to someone involves standing next to impossible.”
• “One morning I shot an elephant in my arms and kissed him.”
• “I spent an interesting evening recently with a grain of salt”

**Synthesizing Computer Vision text**

- What do we get if we extract the probabilities from a chapter on Linear Filters, and then synthesize new statements?


**Synthesized text**

- This means we cannot obtain a separate copy of the best studied regions in the sum.
- All this activity will result in the primate visual system.
- The response is also Gaussian, and hence isn’t bandlimited.
- Instead, we need to know only its response to any data vector, we need to apply a low pass filter that strongly reduces the content of the Fourier transform of a very large standard deviation.
- It is clear how this integral exist (it is sufficient for all pixels within a \(2k+1 \times 2k+1 \times 2k+1 \times 2k+1\) — required for the images separately.)

**Markov Random Field**

A Markov random field (MRF)
- generalization of Markov chains to two or more dimensions.

First-order MRF:
- probability that pixel \(X\) takes a certain value given the values of neighbors \(A, B, C,\) and \(D\): 

\[
P(X|A, B, C, D)
\]

![Markov Random Field Diagram](source: S. Seitz)

**Texture synthesis**

Can apply 2D version of text synthesis

Texture corpus (sample)

Output

![Texture Synthesis](Efros & Leung, ICCV 99)
Texture synthesis: intuition

- Before, we inserted the next word based on existing nearby words…
- Now we want to insert pixel intensities based on existing nearby pixel values.

Distribution of a value of a pixel is conditioned on its neighbors alone.

Synthesizing one pixel

- What is $P(x|\text{neighborhood of pixels around } x)$?
- Find all the windows in the image that match the neighborhood
- To synthesize $x$
  - pick one matching window at random
  - assign $x$ to be the center pixel of that window
- An exact neighbourhood match might not be present, so find the best matches using SSD error and randomly choose between them, preferring better matches with higher probability.

Neighborhood window

Varying window size
Growing Texture

- Starting from the initial image, "grow" the texture one pixel at a time

Synthesis results

- french canvas
- rafia weave

Synthesis results

- white bread
- brick wall
Failure Cases

- Growing garbage
- Verbatim copying

Texture synthesis

- The Efros & Leung algorithm
  - Simple
  - Surprisingly good results
  - Synthesis is easier than analysis!
  - … but can be very slow
    - \([n \times m]\) image synthesis from \([p \times q]\) image requires \(n \times m \times p \times q\) patch comparisons

Image Quilting [Efros & Freeman 2001]

- 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

min. error boundary

Slide from Alyosha Efros
Failures
(Chernobyl Harvest)

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 =

+ =

(Manual) texture synthesis in the media

Style transfer using CNNs

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

Style transfer with texture attributes

Tsung-Yu Lin, Subhransu Maji, CVPR 16

http://vis-www.cs.umass.edu/texture/