Texture Synthesis

Michael Kazhdan

(601.457/657)

An Image Synthesizer. Perlin, 1985

Texture Synthesis by Non-Parametric Sampling. Efros and Leung, 1999

Image Quilting for Texture Synthesis and Transfer. Efros and Freeman, 2001

Wang Tiles for Image and Texture Generation. Cohen et al., 2003
Overview

• Texture synthesis
  ○ Procedural texture synthesis
  ○ Image-based texture synthesis
  ○ Tiling
What is a texture?

Courtesy Paul Bourke
What is a texture?

Courtesy Paul Bourke
What is a texture?

Courtesy Paul Bourke
What is not a texture?

“Pirates of the Caribbean”
What is not a texture?
What is not a texture?

“Lilo and Stitch”
What is a texture?

Texture is an image that exhibits:

• Stationarity -- different regions “look similar”

Courtesy Paul Bourke
What is a texture?

Texture is an image that exhibits:

• Stationarity -- different regions “look similar”

• Locality -- individual pixels related only to small set of neighbors
What is a texture?

Texture is an image that exhibits:

- Stationarity -- different regions “look similar”
- Locality -- individual pixels related only to small set of neighbors

Note:
Any image can be texture-mapped.
We are focusing on images that are qualitatively *textures*. 
Texture Synthesis Problem

How do we go from this...

Or from this...

…to this?

…to this?

Ex nihilo

Ex materia
How can we get textures?

- Photographs
- Artist
- Procedural generation
- Texture “extrapolation”
Photographs

Easy and Fast!

- What if we can’t find the exact texture we want?
- What if our photo is not big enough?
Photographs

Easy and Fast!

• What if we can’t find the exact texture we want?
• What if our photo is not big enough?
  ◦ Stretching changes scale, image quality

Courtesy NVIDIA
Photographs

Easy and Fast!

• What if we can’t find the exact texture we want?
• What if our photo is not big enough?
  ○ Stretching changes scale, image quality
  ○ Tiling looks repetitive
How can we get textures?

- Photographs
- Artist
- Procedural generation
- Texture “extrapolation”
Artist

• There are “texture painters”…
  ✗ Time consuming
  ✗ Difficult
How can we get textures?

- Photographs
- Artist
- Procedural generation
- Texture “extrapolation”
Procedural Textures

• Generated by math instead of an artist

• Good for certain natural phenomena:
  ○ Wood grain
  ○ Marble
  ○ Fire
  ○ Etc.
Perlin-noise Textures

Key Idea:

- Many natural objects have many levels of detail.
- We can create natural looking textures by adding up “noisy” functions at a range of different scales.

Perlin-noise is good function for this!
Perlin-noise Textures

We need:

• Noise function (random number generator)
• Interpolation function

Noise

Interpolation

Courtesy Hugo Elias
Perlin-noise Textures

We need:

- Noise function (random number generator)
- Interpolation function

Frequency := Distance between Samples
Amplitude := Magnitude of the random number
Perlin-noise Textures

So, if we produce noise at different frequencies and amplitudes and add these up…

Courtesy Hugo Elias
Perlin-noise Textures

...we get this!

Sum of Noise Functions = ( Perlin Noise )

Courtesy Hugo Elias
Perlin-noise Textures

…we get this!

How much data is used to represent the texture?

If we sample at $n$ positions we need $2n$ values:
  • $n$ at the finest level
  • $n/2$ at the next level,
  • etc.

In $d$-dimensions, we would need $O(n^d)$ values.
Perlin-noise Textures

…we get this!

Sum of Noise Functions = ( Perlin Noise )

How much data is used to store the texture?

If our random number generator can always generate the same “random number” at index $i$, (e.g. srand) then we only need to store the amplitudes.

Courtesy Hugo Elias
Perlin-noise Textures

...we get this!

Assuming linear interpolation, we need two values at each level. Since there are \( \log(n) \) levels, this implies:

\[ 2 \log(n) \] random numbers

How much computation is required to get the value at a point?

Courtesy Hugo Elias
Perlin-noise Textures

Same idea with 2D images

Courtesy Hugo Elias
Perlin-noise Textures

And even 3D textures

Hugo Elias
Procedural Textures

Pros

• Good for certain natural phenomena
• Constant memory overhead
• Can be computed efficiently $O(2^d \cdot \log(n))$

Cons

• May be hard to find the right function (flowers?)
Perlin-noise Textures

How do we go from this…

Or from this…

…to this?

…to this?

Ex nihilo

Ex materia
How can we get textures?

- Photographs
- Artist
- Procedural generation
- Texture “extrapolation”
  - Creating bigger textures
  - Hole filling
Texture Synthesis

How do we go from this…

…to this?

Or from this…

…to this?
Markov Models: 1D Example

• Assume we have:
  ◦ A fixed alphabet (a through z)
  ◦ An input text such as agggcagcggggcg

• A 0\textsuperscript{th}-order Markov Model:
  ◦ Assign probabilities to the characters based on the frequency of their occurrence in the input text:
    \[ P(a) = \frac{2}{13} \quad P(c) = \frac{3}{13} \quad P(g) = \frac{8}{13} \]
  ◦ Assuming occurrence of a character is independent of previous characters, we can generate new string by “flipping coins”.
Markov Models: 1D Example

But each character *is not* independent of previous characters!

- A $k^{th}$-order Markov Model:
  - Assigns probabilities to a character’s occurrence that uses the context of the previous $k$ characters.
Markov Models: 1D Example

• Assume we have input text with:
  ◦ 100 occurrences of \textit{th}
  ◦ 50 of which followed by \textit{e} (the, then, etc.)
  ◦ 25 of which followed by \textit{i} (this, thin, etc.)
  ◦ 20 of which followed by \textit{a} (that, thank, etc.)
  ◦ 5 of which followed by \textit{o} (though, thorn, etc.)

• 2\textsuperscript{nd}-order Markov model predicts that:
  \[ P(e|\textit{th}) = \frac{1}{2} \quad P(i|\textit{th}) = \frac{1}{4} \quad P(a|\textit{th}) = \frac{1}{5} \quad P(o|\textit{th}) = \frac{1}{20} \]

• Given this probabilistic model and a seed, we can generate new text!
Markov Models: 1D Example

Snippet of original text: “As You like it” by Shakespeare:

DUKE SENIOR:

Now, my co-mates and brothers in exile,
Hath not old custom made this life more sweet
Than that of painted pomp? Are not these woods
More free from peril than the envious court?
Here feel we but the penalty of Adam,
The seasons' difference, as the icy fang
And churlish chiding of the winter's wind,
Which, when it bites and blows upon my body,
Even till I shrink with cold, I smile and say
'This is no flattery: these are counsellors
That feelingly persuade me what I am.’

....
Markov Models: 1D Example

Snippet of generated text with 6th-order Markov Model:

DUKE SENIOR:

Now, my co-mates and thus bolden'd, man, how
now, monsieur Jaques,
Unclaim'd of his absence, as the holly!
Though in the slightest for the fashion of his
absence, as the only wear.
Markov-based Synthesis

How do we go from this…

Or from this…

…to this?

…to this?

Efros & Leung Approach: by using a Markov model!
2D Markov Texture Synthesis

Use this as original “text”

and this as seed

to get this result!

Figure out values of new pixels based on surrounding known pixels
2D Markov Texture Synthesis

Use this as original “text” and this as seed to get this result!

Figure out values of new pixels based on surrounding known pixels.
2D Markov Texture Synthesis

Use this as original “text”

and this as seed

to get this result!

Figure out values of new pixels based on surrounding known pixels
2D Markov Texture Synthesis

Use this as original “text” and this as seed to get this result!

Figure out values of new pixels based on surrounding known pixels.
2D Markov Texture Synthesis

Problems:

• For a given neighborhood, might be only 1 exact match
  ◦ Resulting texture too obviously similar to the first

• For a given neighborhood, there may be no exact matches

Solution:

• Choose among best $N$ matches based on Euclidean distance
2D Markov Texture Synthesis

Examples:
2D Markov Texture Synthesis

**Pros:**
- Conceptually simple/sound
- Often produces good results
- Never chooses a pixel/color NOT found in source

**Cons:**
- Need to choose correct window size
- Very slow! (increasing window size makes this worse)
- Doesn’t always work (can get stuck in a rut)
2D Markov Texture Synthesis

Increasing window size

Courtesy Alexei Efros
2D Markov Texture Synthesis

Pros:
- Conceptually simple/sound
- Often produces good results
- Never chooses a pixel/color NOT found in source

Cons:
- Need to choose correct window size
- Very slow! (increasing window size makes this worse)
- Doesn’t always work (can get stuck in a rut)
2D Markov Texture Synthesis

Pros:
- Conceptually simple/sound
- Often produces good results
- Never chooses a pixel/color NOT found in source

Cons:
- Need to choose correct window size
- Very slow! (increasing window size makes this worse)
- Doesn’t always work (can get stuck in a rut)
2D Markov Texture Synthesis

Growing garbage

Verbatim copying

Courtesy Alexei Efros
Markov-Based Synthesis

How do we go from this...
...to this?

Or from this...
...to this?

Ex nihilo

Ex materia
Wang Tiles

What if our photo is not big enough?

• **Tiling**:
  - discontinuities
  - repetitive
Wang Tiles

What if our photo is not big enough?

- **Tiling:**
  - discontinuities
  - repetitive

- **Extrapolate** (e.g. Efros & Leung)
  - memory overhead
Wang Tiles

Key Idea:
Given a set of colors, and given a sufficiently large set of square tiles whose edges are marked with one of these colors:
Wang Tiles

Key Idea:

The plane can be tiled with edge-matching squares:

Base Tiles

Tiled Image
How Wang Tile Works

Application:

If we associate a texture to each of these tiles:

Input tiles
How Wang Tile Works

Application:

Choosing a tiling of the plane we get a new texture:

Input tiles

Slide courtesy of:
http://www.graphicshardware.org
Wang Tiles

Application:

Choosing a tiling of the plane we get a new texture:
Wang Tiles

Application:

Choosing a tiling of the plane we get a new texture:

The texture will be seamless if the tiles match on common color edges.
Wang Tiles

Tile Complexity:

For the texture not to appear repetitive, we have to be able to (randomly) choose which tiles to use.

How many tiles do we need, assuming $k$ different colors on the edges?
Wang Tiles

Tile Complexity:

In general, we have two restrictions when we introduce a new tile – the colors of the West and North edges.
Wang Tiles

Tile Complexity:

In general, we have two restrictions when we introduce a new tile – the colors of the West and North edges.

For $k$ colors, this means that we need to have $\sim k^2$ tiles to be able to find one that will fit.

In order to be able to make a random choice, this implies that we need to have $\sim 2k^2$ tiles.
Wang Tiles

Tile Generation:
In order to generate these tiles, we have to ensure that if two tiles have a linking edge with the same color label, they match across the boundary.

Otherwise, discontinuity seams will become visible:
Wang Tiles

Tile Generation:

- Associate a source diamond to each colored edge
Wang Tiles

Tile Generation:

- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
Wang Tiles

Tile Generation:

- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
Wang Tiles

**Tile Generation:**
- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
Wang Tiles

Tile Generation:
- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
Wang Tiles

Tile Generation:

- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
Wang Tiles

Tile Generation:

- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
- Quilt the overlap region by solving a graph-cut problem for the minimum discontinuity path
Wang Tiles

Tile Generation:
- Associate a source diamond to each colored edge
- Given a tile, paste the diamonds onto the edges
- Quilt the overlap region by solving a graph-cut problem for the minimum discontinuity path

Since the two-sides of an edge come from the same diamond, they are guaranteed to meet seamlessly!
Wang Tiles

More Complex Effects:

If the tiles are generated carefully, one can control effects like distribution:

- Given a pair of tiles, with different distributions:
Wang Tiles

More Complex Effects:

If the tiles are generated carefully, one can control effects like distribution:

- Given a pair of tiles, with different distributions:
- The two tiles can be used to generate a set of 16 tiles with different distributions:
Wang Tiles

More Complex Effects:

If the tiles are generated carefully, one can control effects like distribution:

- Given a pair of tiles, with different distributions:
- The two tiles can be used to generate a set of 16 tiles with different distributions:

A marker in the corner implies that the second image should be used to generate that corner. Otherwise, the first image should be used.
Wang Tiles

More Complex Effects:

Given a tiling where the adjacent corners either all have a marker, or all don’t:
Wang Tiles

More Complex Effects:

A tiled image can be obtained that exhibits the desired distribution: