Pattern Based Procedural Textures

Sylvain Lefebvre          Fabrice Neyret

iMAGIS - GRAVIR / IMAG - INRIA

http://www-imagis.imag.fr/Membres/Sylvain.Lefebvre/pattern
Overview

• Motivations
• Previous Work
• Contributions
• Our Framework
  – Case study
  – Results
• Conclusion
Motivations

• Texturing large areas
  – Landscapes in simulators
  – Video games

• Requirements
  ⇒ Low memory cost and high resolution
  ⇒ Avoiding periodicity
  ⇒ User control
Previous work

• Large explicit texture
• Extensions
  – Clipmaps [Tanner et al. 98]
  – Texture Compression
  – Empty space compression [Krauss Ertl 02]
• Drawbacks
  – Memory cost
  – Lossy Compression
Previous work

• Procedural textures
  – [Perlin 85] [Worley 96] [Ebert 94]

• Drawbacks
  – Calculation cost
  – Control by the artist not trivial
  – Not all materials
Previous work

• Pattern based texturing
  – Tiling
  – Aperiodic tiling [Stam 97]
  – Triangular patterns [Neyret Cani 99]
  – Virtual atlases [Soler 02]
  – Sparse convolutions [Lewis 89, Erbert 94]

• Drawbacks
  – Mesh dependency
  – Local variations not easy
Previous work

• Pattern based texturing
  – Tiling
  – Aperiodic tiling [Stam 97]
  – Triangular patterns [Neyret Cani 99]
  – Virtual atlases [Soler 02]
  – Sparse convolutions [Lewis 89, Erbert 94]

• Drawbacks
  – Mesh dependency
  – Local variations
Previous work

• Pattern based texturing
  – Tiling
  – Aperiodic tiling [Stam 97]
  – Triangular patterns [Neyret Cani 99]
  – Virtual atlases [Soler 02]
  – Sparse convolutions [Lewis 89, Erbert 94]

• Drawbacks
  – Mesh dependency
  – Local variations
Contributions

• Framework for creating large textures by combining patterns
  – No constraint on mesh (texture space)
  – Low memory cost

• Runs on today graphics hardware

• As generic as possible
Method overview

Texture coordinates → pattern

Texture space

(u,v)
Method overview

Texture coordinates $\rightarrow$ pattern $\rightarrow$ color

Texture space
Method overview

Texture coordinates → pattern → color

Texture space

(u,v)
Method overview

Texture coordinates \(\rightarrow\) color
Corresponds to hardware *fragment program*

Programmable Graphics Hardware
Method overview

Texture coordinates $\rightarrow$ color
Corresponds to hardware fragment program

Relies on indirection textures (like [Krauss Ertl 02])

(u,v)
Our Framework

• Set of basic blocks

• 1 block = 1 functionality

• Textures by combining blocks
Overview

• Motivations
• Previous Work
• Contributions
• Our Framework
  – Case study
  – Results
• Conclusion
Case study

- Aperiodic tiling
  - \( N \times N \) virtual cells
  - \( T \times T \) patterns
  - 3 blocks
Aperiodic tiling

• Virtual Tile Map
  – cell index $g$
  – relative coordinates $u_{tile}$
Aperiodic tiling

- Virtual Indirection Map
  - random index \( p \) from \( g \) (aperiodic)
  - uses permutation table \( \sigma \)

\[
p = \sigma \left( \frac{g}{T^2} + \sigma \left( \frac{g}{T} + \sigma (g) \right) \right)
\]
Aperiodic tiling

• Reference Texture
Aperiodic tiling

Virtual Tile Map

Virtual Indirection Map

Reference Texture
Blocks

– Pattern choice and positioning
– Transitions between neighboring patterns
– Animation
Blocks

- Pattern choice and positioning

- Transitions between neighboring patterns

- Animation
Blocks

– Pattern choice and positioning

– Transitions between neighboring patterns

– Animation
Overview

• Motivations
• Previous Work
• Contributions
• Our Framework
  – Case study
  – Results
• Conclusion
Probability distribution control

- Areas map

Interpolation of probabilities
Probability distribution control

- Areas map

Interpolation of probabilities
Transitions

WarCraft3© Blizzard Entertainment

only one quad
Transitions

only one quad
Random positioning
Animations
Animations

Using explicit positionning

Memory cost = pattern + 16x16 positionning map
Performances

- Measures on GeForce FX prototype (half speed)
- 32 bits precision
- texture covers full screen

<table>
<thead>
<tr>
<th>Code length</th>
<th>Aperiodic tiling</th>
<th>Areas map</th>
<th>Dithered Areas Map</th>
<th>Dithered Areas Map &amp; transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>56 instr.</td>
<td>65 instr.</td>
<td>117 instr.</td>
<td>512 instr.</td>
</tr>
<tr>
<td>Tex. lookups</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>39</td>
</tr>
<tr>
<td>320x200</td>
<td>113 fps</td>
<td>73 fps</td>
<td>36 fps</td>
<td>5 fps</td>
</tr>
<tr>
<td>640x480</td>
<td>24.5 fps</td>
<td>15.5 fps</td>
<td>8.5 fps</td>
<td>1.1 fps</td>
</tr>
</tbody>
</table>
Filtering

- MIP – mapping
- Close view point
  - Linear interpolation
- Far view point
  - Indirection $\rightarrow$ average color
- Issue in general with indirection textures
Conclusion

• Framework
• Runs on hardware
• Allows very large aperiodic textures
• Low memory cost

⇒ Filtering issue
⇒ Exploiting the framework
Thank you!

Questions?
Pattern Based Procedural Textures

Sylvain Lefebvre          Fabrice Neyret

iMAGIS - GRAVIR / IMAG - INRIA

http://www-imagis.imag.fr/Membres/Sylvain.Lefebvre/pattern