Compat-Map For Real-time Texture Synthesis And Rendering

Simon Fenney, Linling Zhang
7 August 2015
Our Method

- Random access / concurrent synthesis.
- Low computational complexity.
- Small storage requirements.
Texture Stitching

Stitch together sections of the exemplar image.
Texture Stitching

*Stitch together sections of the exemplar image.*
Algorithm Overview

Three part algorithm.
Algorithm Overview

Three part algorithm.
Compact, Compatibility Map
AKA Compat-map

- Stores the best matching “next” block.
Compact, Compatibility Map
AKA Compat-map

- Stores the best matching “next” block.
Compact, Compatibility Map

AKA Compat-map

- Stores the best matching “next” block.
Compact, Compatibility Map

AKA Compat-map

- Stores the best matching “next” block.
Compact, Compatibility Map

AKA Compat-map
Compact, Compatibility Map

AKA Compat-map
Compact, Compatibility Map
AKA Compat-map
Algorithm Overview

Three part algorithm.

Compat-Map Generation → 2 → 3
Algorithm Overview

Three part algorithm.

Compat-Map Generation → Index-Table Generation → 3 → Result
**Index-Table**

- Represents a synthesised texture using tile IDs in place of pixels.

<table>
<thead>
<tr>
<th></th>
<th>41</th>
<th>47</th>
<th>238</th>
<th>156</th>
<th>175</th>
<th>139</th>
<th>159</th>
<th>158</th>
</tr>
</thead>
<tbody>
<tr>
<td>169</td>
<td>222</td>
<td>176</td>
<td>92</td>
<td>139</td>
<td>251</td>
<td>156</td>
<td>146</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>24</td>
<td>183</td>
<td>175</td>
<td>239</td>
<td>202</td>
<td>175</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>13</td>
<td>190</td>
<td>139</td>
<td>168</td>
<td>207</td>
<td>139</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>233</td>
<td>208</td>
<td>159</td>
<td>243</td>
<td>175</td>
<td>160</td>
<td>80</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>216</td>
<td>157</td>
<td>250</td>
<td>139</td>
<td>168</td>
<td>156</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>223</td>
<td>140</td>
<td>99</td>
<td>0</td>
<td>147</td>
<td>146</td>
<td>207</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>75</td>
<td>163</td>
<td>134</td>
<td>7</td>
<td>154</td>
<td>188</td>
<td>159</td>
<td></td>
</tr>
</tbody>
</table>
Index-Table

- Represents a synthesised texture using tile IDs in place of pixels.

<table>
<thead>
<tr>
<th></th>
<th>222</th>
<th>176</th>
<th>92</th>
<th>139</th>
<th>251</th>
<th>156</th>
<th>146</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>183</td>
<td>175</td>
<td>239</td>
<td>202</td>
<td>175</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>190</td>
<td>139</td>
<td>168</td>
<td>207</td>
<td>139</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>159</td>
<td>243</td>
<td>175</td>
<td>160</td>
<td>80</td>
<td>192</td>
<td></td>
</tr>
<tr>
<td>216</td>
<td>157</td>
<td>250</td>
<td>139</td>
<td>168</td>
<td>156</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>223</td>
<td>140</td>
<td>99</td>
<td>0</td>
<td>147</td>
<td>146</td>
<td>207</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>163</td>
<td>134</td>
<td>7</td>
<td>154</td>
<td>188</td>
<td>159</td>
<td></td>
</tr>
</tbody>
</table>
Algorithm Overview

Three part algorithm.

Compat-Map Generation → Index-Table Generation → 3
Algorithm Overview

Three part algorithm.

Compat-Map Generation → Index-Table Generation → Texture Synthesis
Texture Synthesis

Fully parallelisable

+ 

<table>
<thead>
<tr>
<th>222</th>
<th>176</th>
<th>92</th>
<th>139</th>
<th>251</th>
<th>156</th>
<th>146</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>183</td>
<td>175</td>
<td>239</td>
<td>202</td>
<td>175</td>
<td>153</td>
</tr>
<tr>
<td>13</td>
<td>190</td>
<td>139</td>
<td>168</td>
<td>207</td>
<td>139</td>
<td>98</td>
</tr>
<tr>
<td>208</td>
<td>159</td>
<td>243</td>
<td>175</td>
<td>160</td>
<td>80</td>
<td>192</td>
</tr>
<tr>
<td>216</td>
<td>157</td>
<td>250</td>
<td>139</td>
<td>168</td>
<td>156</td>
<td>200</td>
</tr>
<tr>
<td>223</td>
<td>140</td>
<td>99</td>
<td>0</td>
<td>147</td>
<td>146</td>
<td>207</td>
</tr>
<tr>
<td>75</td>
<td>163</td>
<td>134</td>
<td>7</td>
<td>154</td>
<td>188</td>
<td>159</td>
</tr>
</tbody>
</table>
Texture Synthesis

Fully parallelisable

\[(u, v) = (0.5, 0.8)\]
Algorithm Overview

Three part algorithm.

Compat-Map Generation

Index-Table Generation

Texture Synthesis

Parallel

Parallel
Minimum Cutting Path
Fast Method For Block Artifact Reduction
Minimum Cutting Path

Fast Method For Block Artifact Reduction
Minimum Cutting Path

Fast Method For Block Artifact Reduction

Diagram showing a before and after comparison of block artifacts in a video compression scenario.
Minimum Cutting Path
Fast Method For Block Artifact Reduction
Block Reflection

Create Exemplar Information
Block Reflection
Create Exemplar Information

ID0
H-flip
V-flip
VH-flip
Block Reflection
Create Exemplar Information

No Reflection

With Reflection
Block Reflection

Not Always Applicable

No Reflection

With Reflection
## Performance Tests

*Tested on a mobile tablet with an ARM A7@1.2Ghz*

<table>
<thead>
<tr>
<th>Synthesised Size</th>
<th>Analysis Time</th>
<th>Synthesis Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>280 x 280</td>
<td>0.82s</td>
<td>0.002s</td>
</tr>
<tr>
<td>460 x 460</td>
<td>0.82s</td>
<td>0.007s</td>
</tr>
<tr>
<td>910 x 910</td>
<td>0.82s</td>
<td>0.03s</td>
</tr>
</tbody>
</table>
## Performance Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>GPU</th>
<th>Analysis Time</th>
<th>Synthesis Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (256x256 Compat-Map)</td>
<td>NO</td>
<td>0.25s</td>
<td>≤0.001s</td>
</tr>
<tr>
<td>Wang-tiles (8 tiles)</td>
<td>NO</td>
<td>3s</td>
<td>≤0.001s</td>
</tr>
<tr>
<td>Lefebvre and Hoppe</td>
<td>YES</td>
<td>0s</td>
<td>0.02s</td>
</tr>
<tr>
<td>Liang and Liu</td>
<td>NO</td>
<td>0.678s</td>
<td>0.02s</td>
</tr>
</tbody>
</table>

Synthesis of 200 x 200 tiles from a 128 x 128 pixel exemplar.
Conclusion

Great Method For Mobile Platforms

- Random access / concurrent synthesis.
- Low computational complexity.
- Small storage requirements.
Thank you
Questions?