



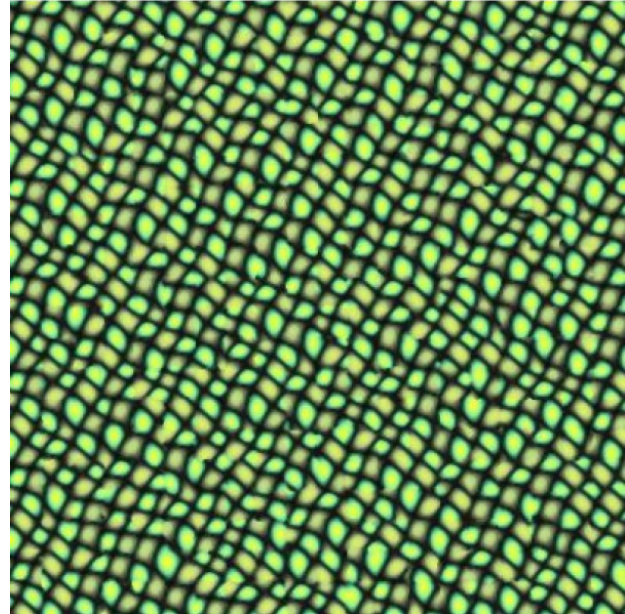
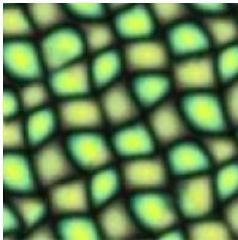
Imagination

Compat-Map For Real-time Texture Synthesis And Rendering

Simon Fenney, Linling Zhang
7 August 2015

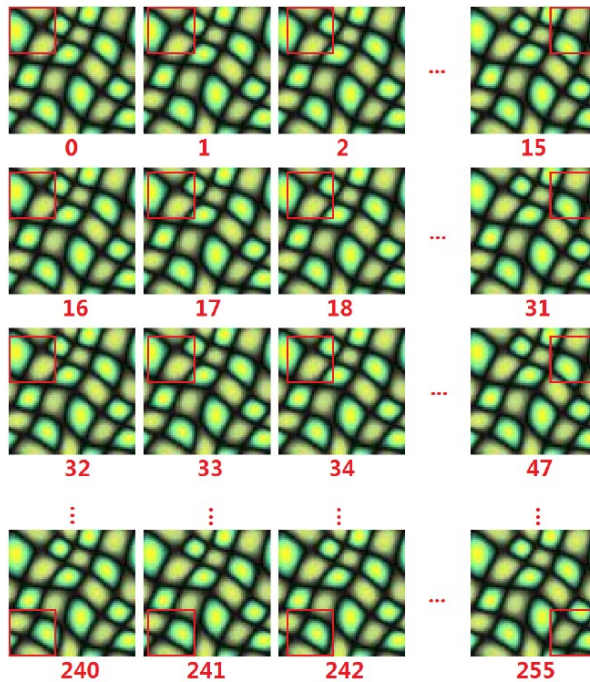
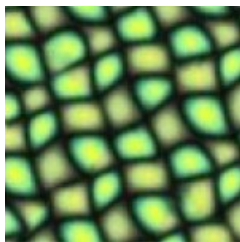
Our Method

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



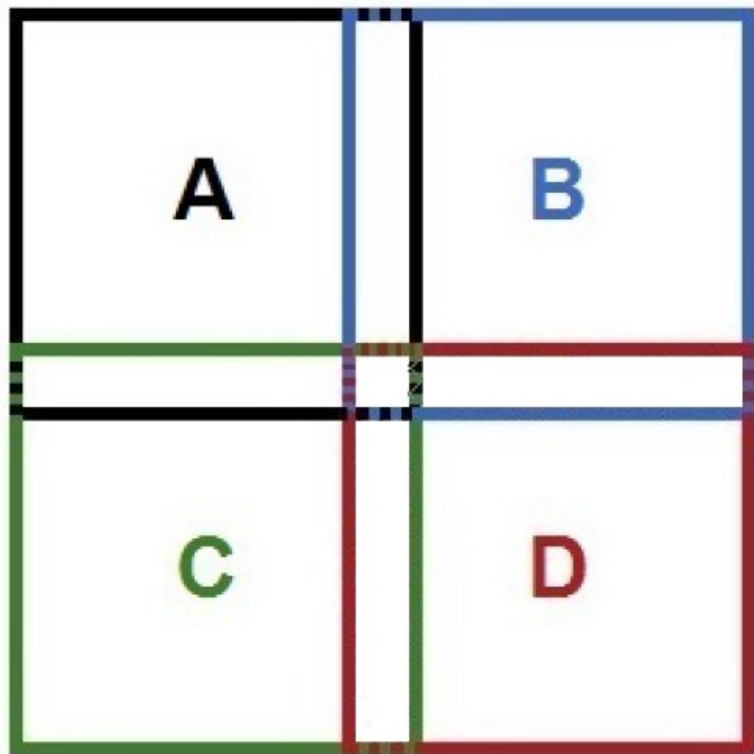
Texture Stitching

Stitch together sections of the exemplar image.



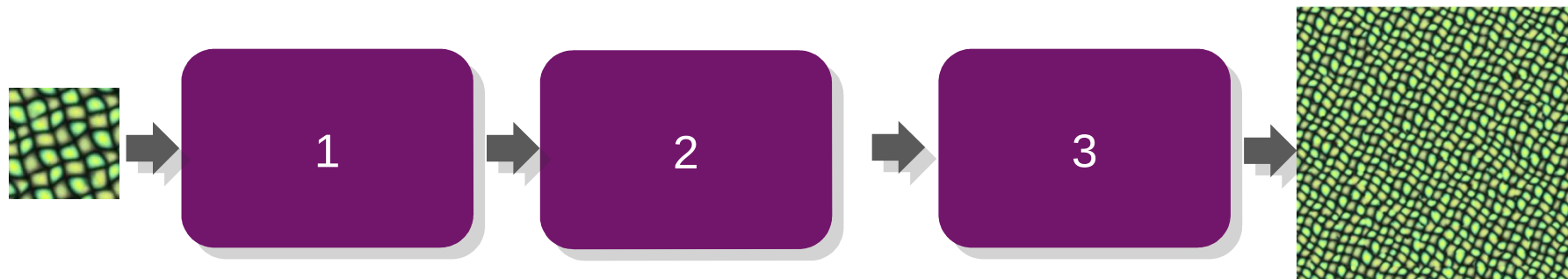
Texture Stitching

Stitch together sections of the exemplar image.



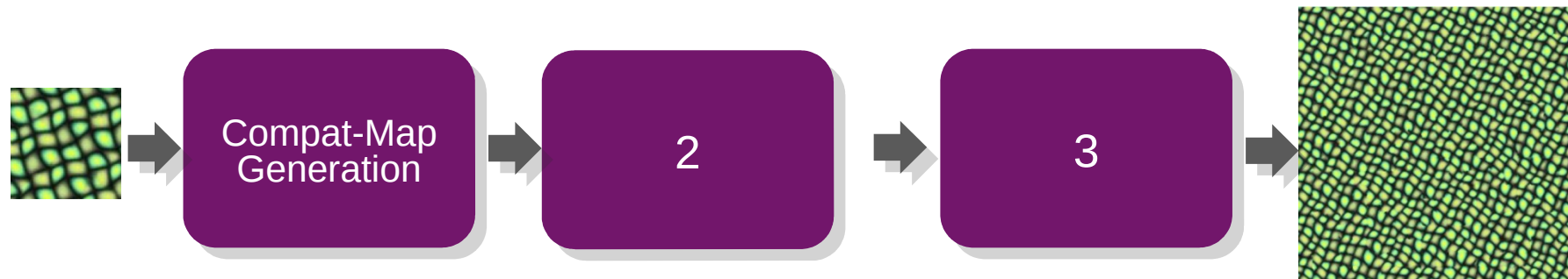
Algorithm Overview

Three part algorithm.



Algorithm Overview

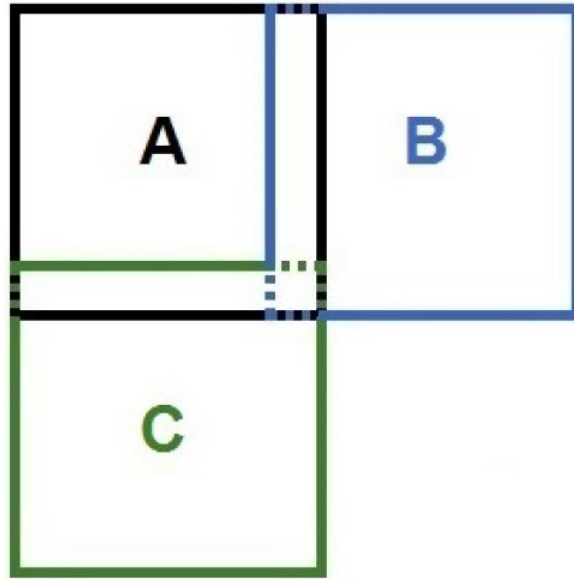
Three part algorithm.



Compact, Compatibility Map

AKA Compat-map

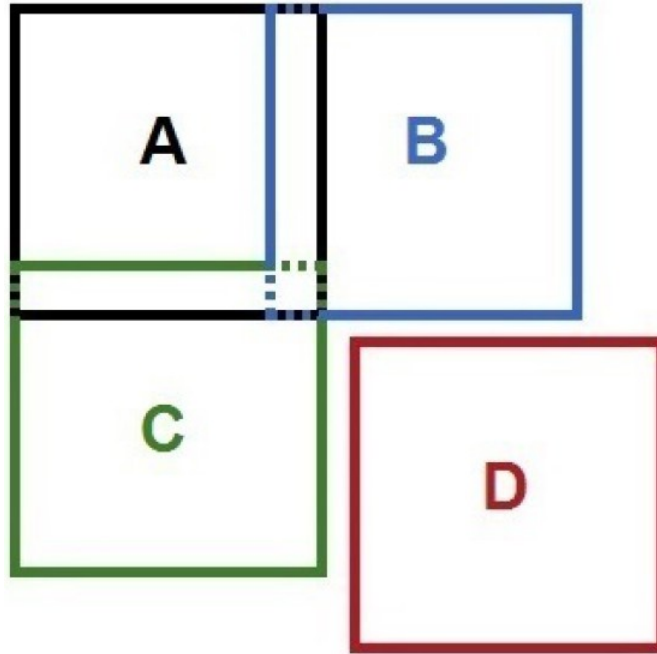
- Stores the best matching “next” block.



Compact, Compatibility Map

AKA Compat-map

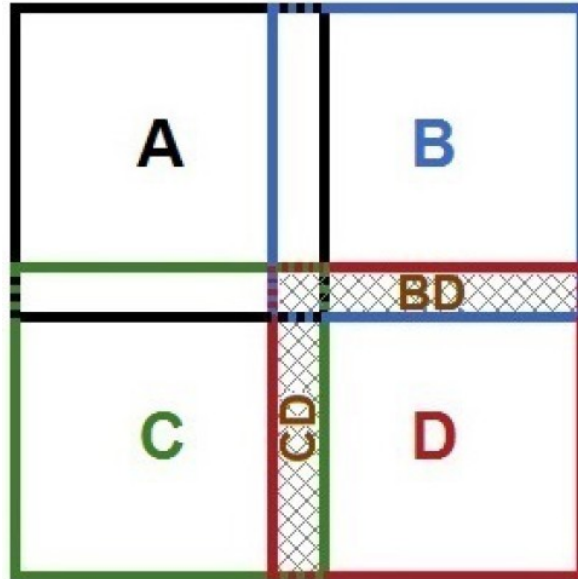
- Stores the best matching “next” block.



Compact, Compatibility Map

AKA Compat-map

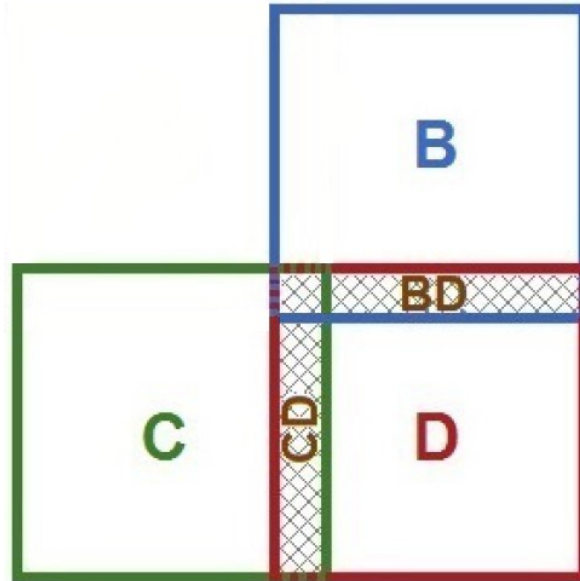
- Stores the best matching “next” block.



Compact, Compatibility Map

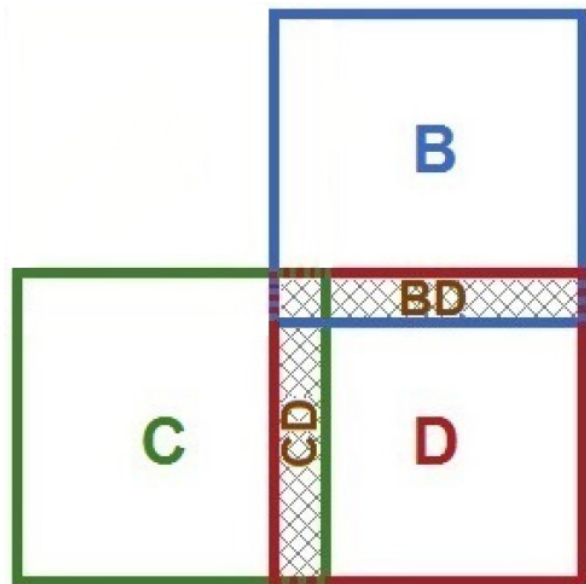
AKA Compat-map

- Stores the best matching “next” block.



Compact, Compatibility Map

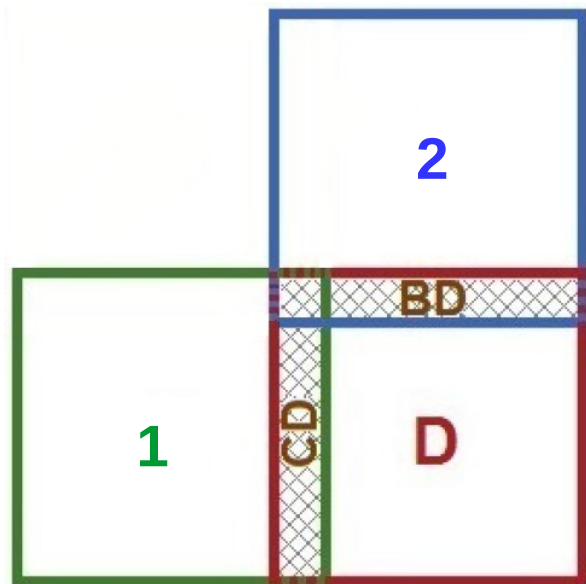
AKA Compat-map



y ^x	0	1	2	...	255
0				...	
1				...	
2				...	
⋮				⋮	
255					

Compact, Compatibility Map

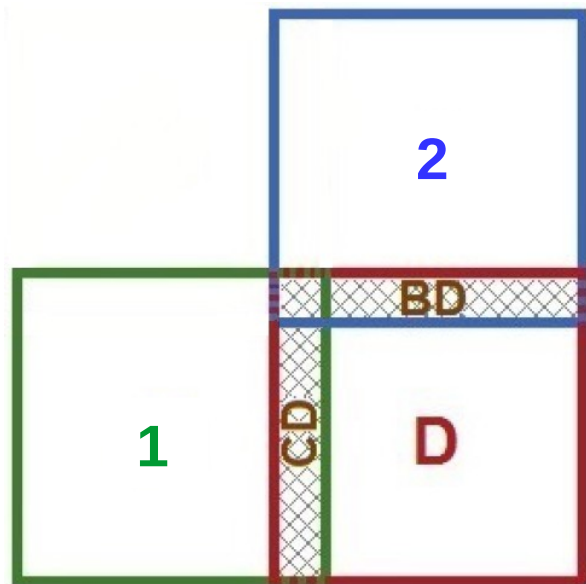
AKA Compat-map



y ^x	0	1	2	...	255					
0				...						
1				...						
2				...						
⋮										
⋮										
255										

Compact, Compatibility Map

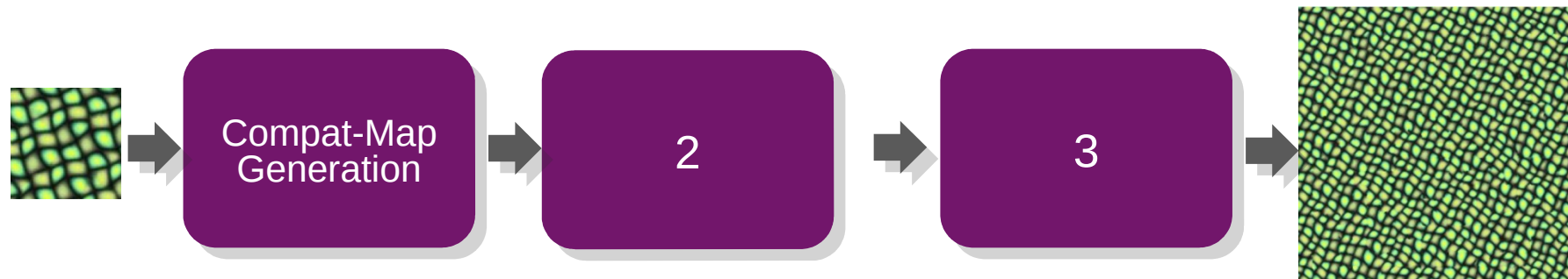
AKA Compat-map



y ^x	0	1	2	...	255
0				...	
1			X	...	
2				...	
⋮				⋮	
255					

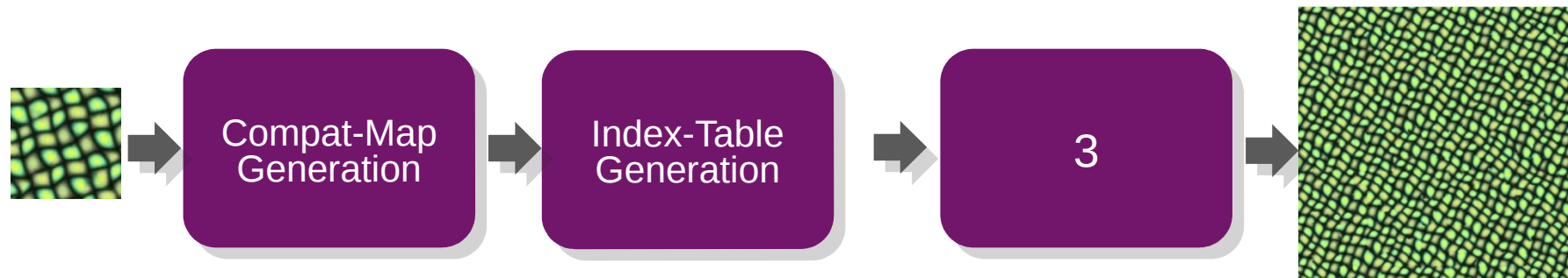
Algorithm Overview

Three part algorithm.



Algorithm Overview

Three part algorithm.



Index-Table

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

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

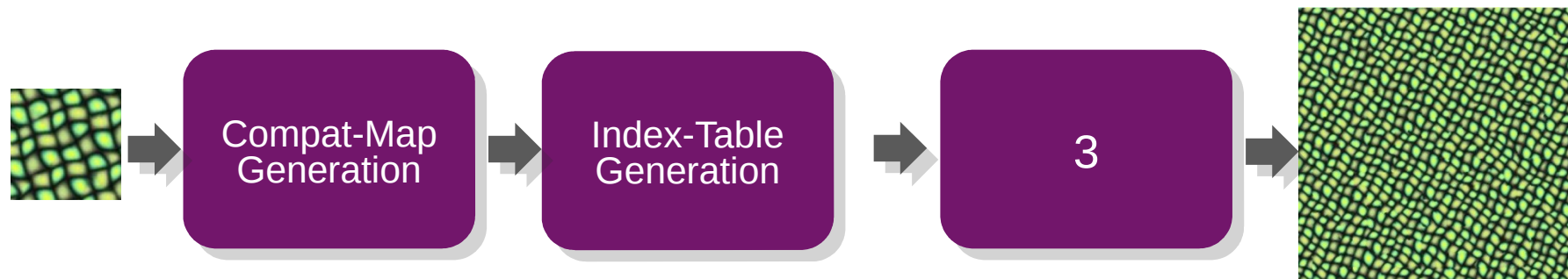
Index-Table

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

222	176	92	139	251	156	146
24	183	175	239	202	175	153
13	190	139	168	207	139	98
208	159	243	175	160	80	192
216	157	250	139	168	156	200
223	140	99	0	147	146	207
75	163	134	7	154	188	159

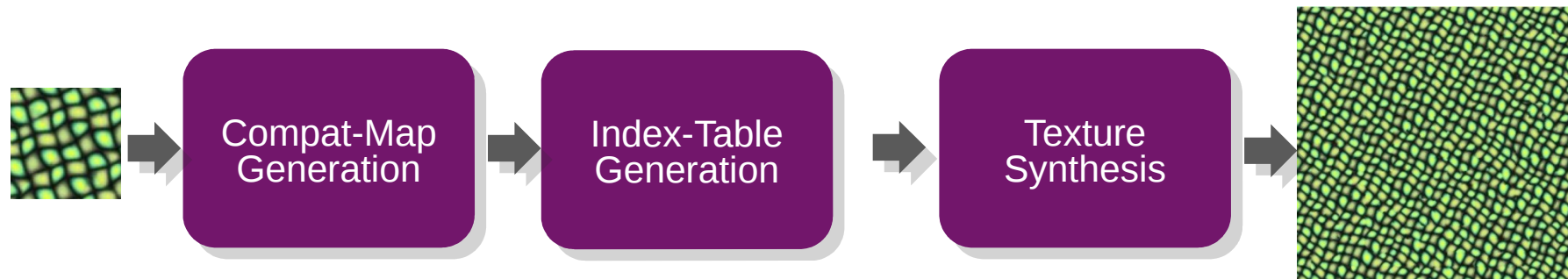
Algorithm Overview

Three part algorithm.



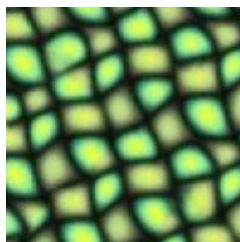
Algorithm Overview

Three part algorithm.



Texture Synthesis

Fully parallelisable



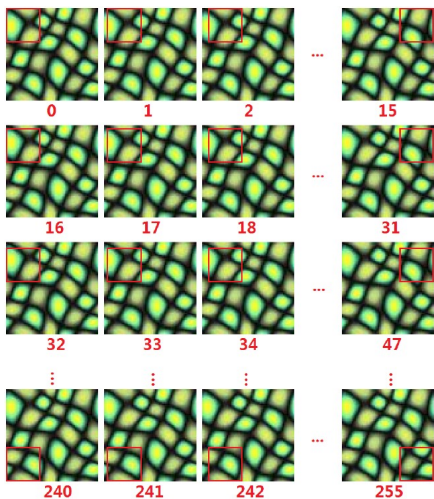
+

222	176	92	139	251	156	146
24	183	175	239	202	175	153
13	190	139	168	207	139	98
208	159	243	175	160	80	192
216	157	250	139	168	156	200
223	140	99	0	147	146	207
75	163	134	7	154	188	159

Texture Synthesis

Fully parallelisable

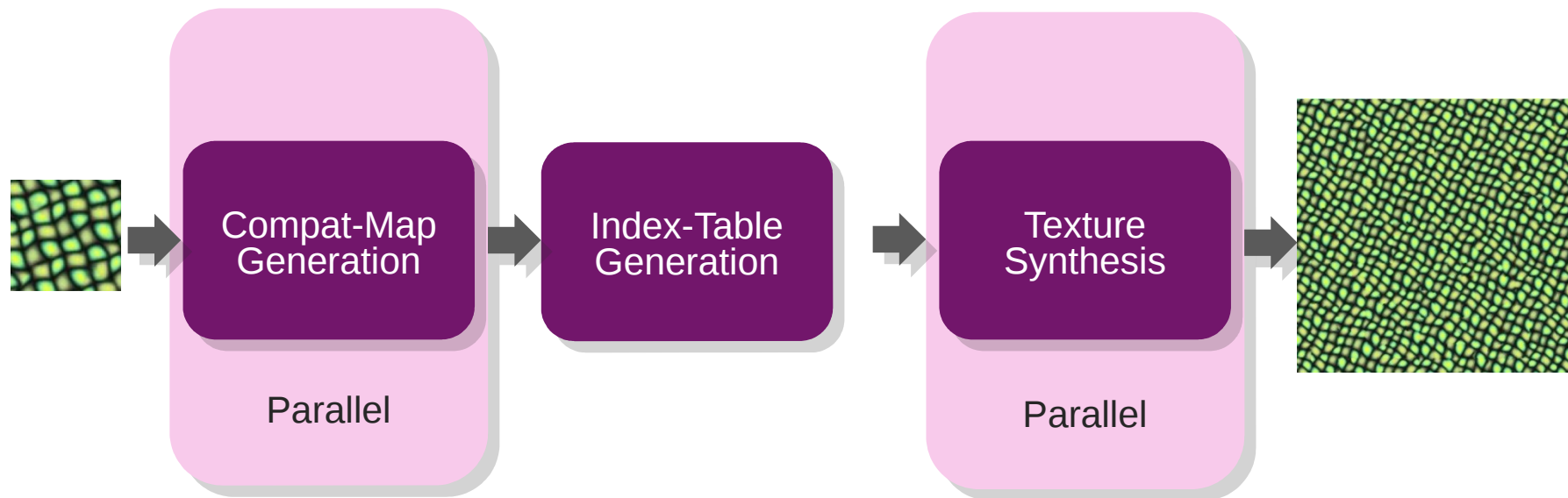
$(u, v) = (0.5, 0.8)$



222	176	92	139	251	156	146
24	183	175	239	202	175	153
13	190	139	168	207	139	98
208	159	243	175	160	80	192
216	157	250	139	168	156	200
223	140	99	0	147	146	207
75	163	134	7	154	188	159

Algorithm Overview

Three part algorithm.



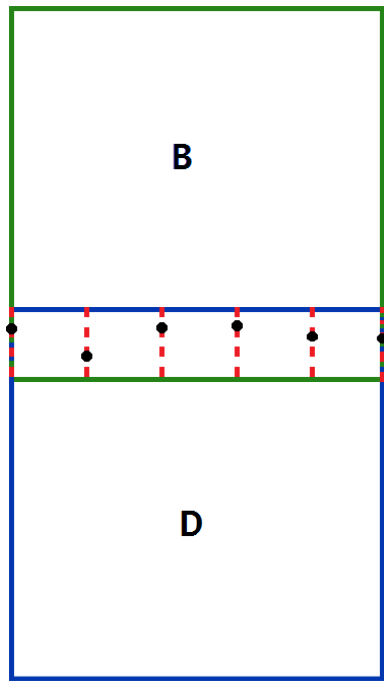
Minimum Cutting Path

Fast Method For Block Artifact Reduction



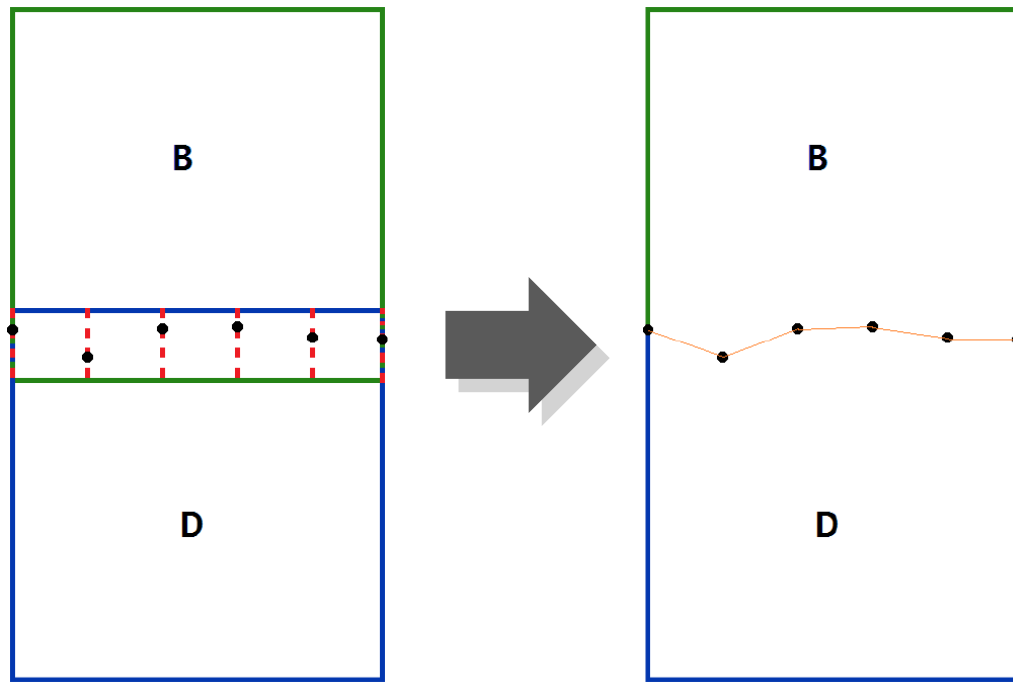
Minimum Cutting Path

Fast Method For Block Artifact Reduction



Minimum Cutting Path

Fast Method For Block Artifact Reduction



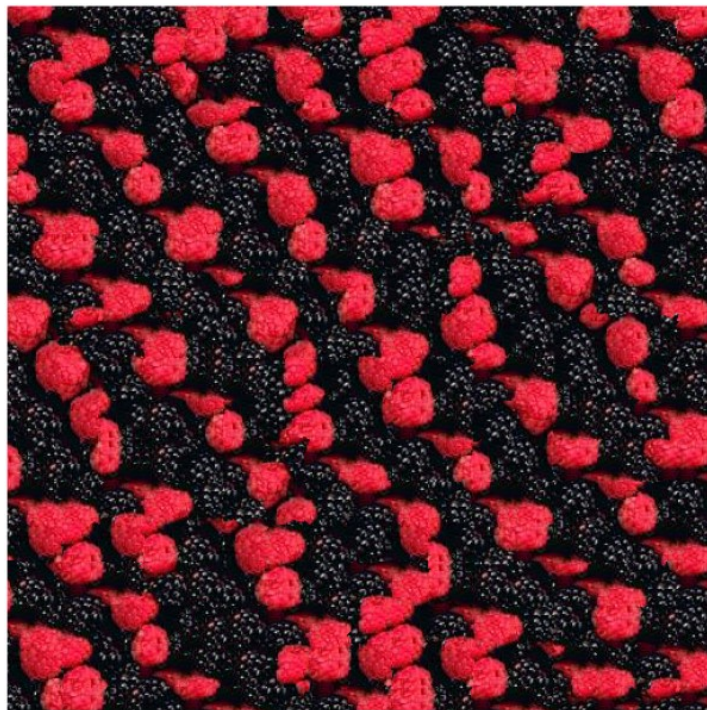
Minimum Cutting Path

Fast Method For Block Artifact Reduction



Block Reflection

Create Exemplar Information



Block Reflection

Create Exemplar Information



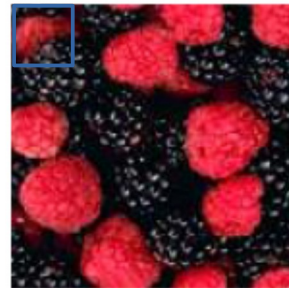
ID0



H-flip



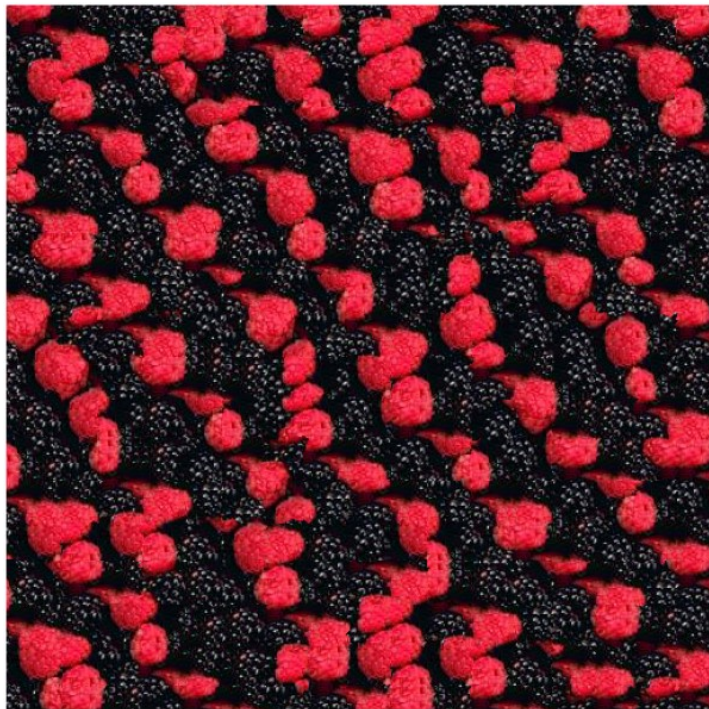
V-flip



VH-flip

Block Reflection

Create Exemplar Information



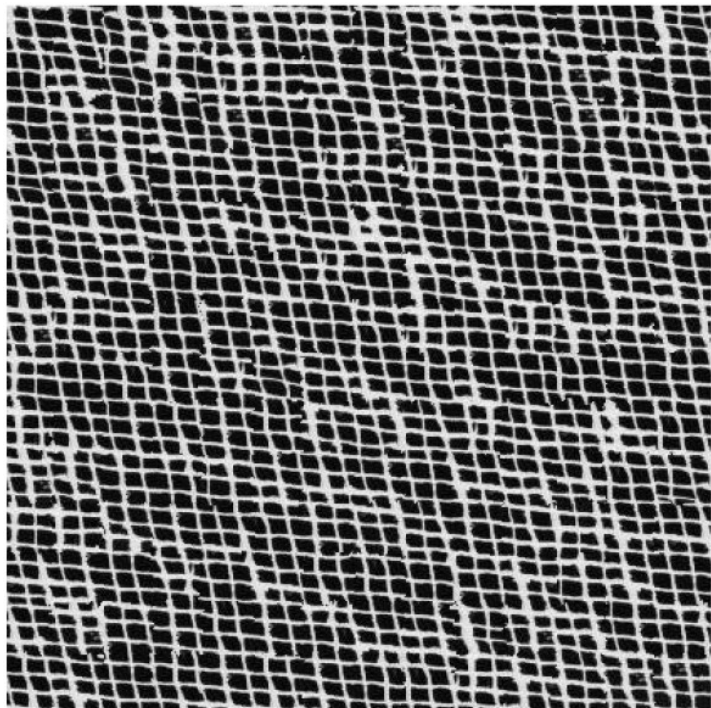
No Reflection



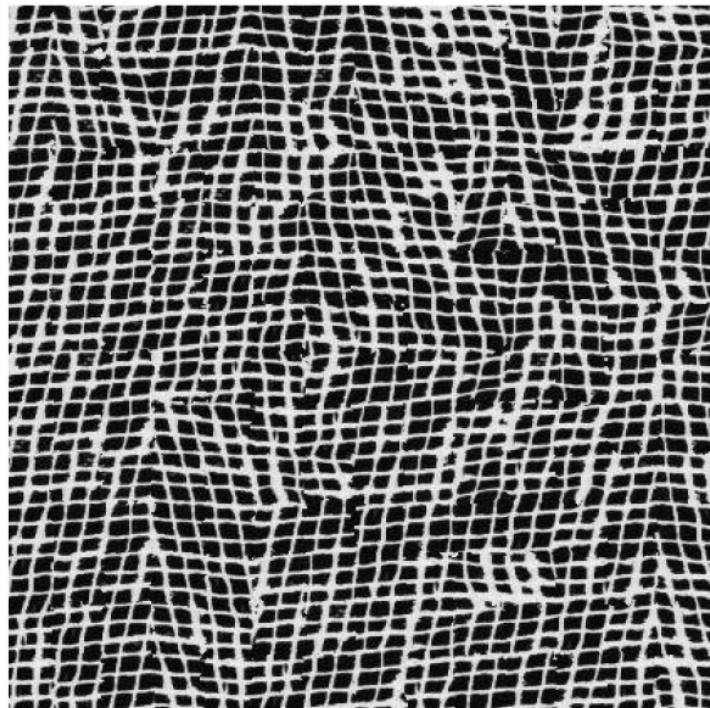
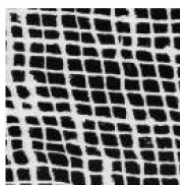
With Reflection

Block Reflection

Not Always Applicable



No Reflection



With Reflection

Performance Tests

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

Synthesised Size	Analysis Time	Synthesis Time
280 x 280	0.82s	0.002s
460 x 460	0.82s	0.007s
910 x 910	0.82s	0.03s

Performance Comparison

Method	GPU	Analysis Time	Synthesis Time
Ours (256x256 Compat-Map)	NO	0.25s	$\leq 0.001s$
Wang-tiles (8 tiles)	NO	3s	$\leq 0.001s$
Lefebvre and Hoppe	YES	0s	0.02s
Liang and Liu	NO	0.678s	0.02s

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?