Procedural Texturing

By Madis Janno
What do I mean by Procedural Texturing?
Two possible things:

Procedurally generating textures

Procedurally applying textures
Todays focus is:

Applying existing textures in a way that creates new textures
Why?
Why?

Tiling
Why?

Procedural geometry
Why?

Making custom textures for everything is difficult
Topics are:

- Texture blending basics
- Texture splatting
- Contrast correction
- Height blending
- Texture bombing (+ workshop)
- Triplanar texturing (pretty much just showing off my project)
Will post shadertoy links in chat + top of slides

Will pause at the end of each chapter to let you look at code, ask any questions
Texture blending basics
Simple blending

www.shadertoy.com/view/WdSczc

Texture1*w + Texture2*(1-w)
Simple blending

www.shadertoy.com/view/WdSczc

Texture1*w + Texture2*(1-w)

Seems off somehow?
Simple blending

www.shadertoy.com/view/WdSczc

Texture1*w + Texture2*(1-w)

Darker in the middle
Simple blending

Colours in images and on screen are not linear (sRGB)

$(0.5, 0, 0.5)$ is darker than $(1, 0, 0)$
Simple blending

www.shadertoy.com/view/WdSczc

Colours in images and on screen are not linear (sRGB)

(0.5, 0, 0.5) is darker than (1, 0, 0)

How to fix?
Simple blending

www.shadertoy.com/view/WdSczc

Colours in images and on screen are not linear

(0.5, 0, 0.5) is darker than (1, 0, 0)

How to fix?

Convert to linear: \( \text{lin}_\text{rgb} = \text{rgb}^{2.2} \)

Display on screen: \( \text{rgb} = \text{lin}_\text{rgb}^{(1.0/2.2)} \)
Effect less pronounced when textures have similar colours
Look at middle, without gamma correction tile texture nearly invisible
Lesson

Gamma correct before anything (images sRGB by default)

Always convert back at the end

Darker regions when blending or blurring means a lack of gamma correction

Easy to forget, even image editing software screws up
Texture splatting
Texture splatting

Texture weights can be read from textures.

A single texture can contain weights for up to 5 textures.

Data textures should not be gamma corrected on read.

www.shadertoy.com/view/3s2yzc
Texture splatting

Can construct textures from this

Can add splat texture to linear blend to make more natural looking blends

www.shadertoy.com/view/3s2yzc
Lesson

Any sort of data can come from textures
Contrast correction
Lagrangian Texture Advection: Preserving both Spectrum and Velocity Field
Contrast correction

www.shadertoy.com/view/td2cRV

Blended textures lose contrast

Values pushed towards mean
Contrast correction

www.shadertoy.com/view/td2cRV

Blended textures lose contrast

Values pushed towards mean

Multiplying by values <1 lowers contrast, adding two reduced contrast images together does not restore all
Contrast correction  www.shadertoy.com/view/td2cRV

One example of correcting this: “On Histogram-preserving Blending for Randomized Texture Tiling” from Disney

Convert textures into gaussian distributions and store previous histograms, blend gaussians and restore variance, restore histograms
Contrast correction

One example of correcting this: “On Histogram-preserving Blending for Randomized Texture Tiling” from Disney

Convert textures into gaussian distributions and store previous histograms, blend gaussians and restore variance, restore histograms

Way too complicated
Contrast correction

www.shadertoy.com/view/td2cRV

Simpler method from:
Lagrangian Texture Advection: Preserving both Spectrum and Velocity Field

In simpler terms: final color = \frac{(blended color-mean)}{\sqrt{\sum w^2))} + mean
Contrast correction

Simpler method from: Lagrangian Texture Advection: Preserving both Spectrum and Velocity Field

In simpler terms: final color = (blended color-mean)/sqrt(sum(w^2)) + mean

Mean value of a texture can be grabbed from the highest mipmap, or precomputed
Contrast correction

www.shadertoy.com/view/td2cRV

Simpler method from:
Lagrangian Texture Advection: Preserving both Spectrum and Velocity Field

\[ R'(x) = \frac{\sum w_i(x)(R(u_i(x)) - \hat{R})}{\sqrt{\sum w_i^2(x)}} + \hat{R} \]

Mathematically: corrects new blended texture to have the same variance as original
Contrast correction

www.shadertoy.com/view/td2cRV

Problems:

Assumes colour distributions have a normal distribution
Contrast correction  

www.shadertoy.com/view/td2cRV

Problems:

Assumes colour distributions have a normal distribution

Assumes blended colors are independant, overcorrects if blended textures correlate
Contrast correction  

www.shadertoy.com/view/td2cRV

Problems:

Assumes colour distributions have a normal distribution

Assumes blended colors are independant, overcorrects if blended textures correlate

Can generate values not present in original textures, causes clipping when values go negative or too high
Contrast correction

www.shadertoy.com/view/td2cRV

Sometimes good: www.shadertoy.com/view/tsVGRd
Contrast correction  

Sometimes good: www.shadertoy.com/view/tsVGRd

Problem: Example hasn’t gamma corrected
www.shadertoy.com/view/ts2cRV

Example with 3 textures
Contrast correction

Tends to overcompensate in practise

Used in my CGP work, toned down by using fifth root instead of square root
Lesson

There are ways of boosting contrast if the texture creation or blending process removes too much.

Don’t go overboard.
Height blending
Height blending

In reality things don’t blend smoothly

If blending between smaller and bigger rocks, bigger rocks just “phase” out
Height blending

Solution?

www.shadertoy.com/view/wdSczc
Height blending

www.shadertoy.com/view/wdSczc

Account for the heights of the textures.
Height blending

Account for the heights of the textures.

Requires heightmaps
Height blending

www.shadertoy.com/view/wdSczc

Account for the heights of the textures.

Requires heightmaps

Greyscale can work in a pinch
Height blending

Principle:
Multiply heights by weights
Compare heights -> texture weights

www.shadertoy.com/view/wdSczc
Height blending

Way 1:

Heights*Weights -> Compare ratios

www.shadertoy.com/view/wdSczc
Height blending

Way 2:

Heights*Weights

www.shadertoy.com/view/wdSczc
Height blending

Way 2:
Heights*Weights
Floor = (highest height - blend factor)

www.shadertoy.com/view/wdSczc
Height blending

Way 2:

Heights*Weights

Floor = (highest height - blend factor)

Heights -= Floor

www.shadertoy.com/view/wdSczc
Height blending

Way 2:
Heights*Weights
Floor = (highest height - blend factor)
Heights -= Floor
Compare Height ratios

www.shadertoy.com/view/wdSczc
Height blending

Way 2:

Allows for sharper borders

Can tweak by altering blend factor

www.shadertoy.com/view/wdSczc
Lesson

You can use extra data to alter blending

Thinking in real world terms can help
Texture bombing
Texture bombing

www.shadertoy.com/view/tsVGRd

A way of removing/reducing tiling

A way of adding elements to random locations on texture

developer.download.nvidia.com/books/HTML/gpugems/gpugems_ch20.html
Texture bombing

www.shadertoy.com/view/tsVGRd

Principle:

- Divide area into cells (can be 3D)
- During rendering get data from corners of current cell
- Blend or draw stuff based on data
Texture bombing

www.shadertoy.com/view/tsVGRd

Data can include:

- Rotations
- UV coordinates of some shape in atlas
- UV coordinates of location on tiling texture
- Colors
- Etc
Workshop

www.shadertoy.com/view/3d2cRc

Implement

height blending and/or contrast correction
Lesson

You can combine everything we have talked about

www.shadertoy.com/view/3d2cRc
Triplanar texturing

Applicable for terrain, buildings

madisjanno.github.io/Hexi/
Triplanar texturing

Applicable for terrain, buildings

Basic principle is to combine 3 textures to texture all sides of some shape
Triplanar texturing

3 textures, 1 for each plane
XY, YZ, XZ
Coordinates on that plane
determine texture UV’s
We use surface normal as blend weights

madisjanno.github.io/Hexi/
Triplanar texturing

End result smoothly combines all 3 textures

There are some artifacts when surface normals don’t point at planes
Lesson

Everything you learned also applies to 3d

madisjanno.github.io/Hexi/
Stuff you can use all this for:

- Automatically adding details to roads and streets
- Dynamically “damaging” enemies
- Easily texturing procedurally generated building
- And more!
Any questions?
Thanks for listening!