rpTextures:
SYSTEMATIC LAYERING FOR LARGE TEXTURE GENERATION

Austin E. MacKay and Jonathan D. Denning Taylor University, Indiana, USA

ABSTRACT

We used systematic layering of variously sized layers to quickly create large, seemingly non-repeating textures. This leads to significantly more control for artists to create large visual scenes without the need for large teams to create massive textures. Our method maintains the visual appeal of seamless and non-repeating design, as well as uses little memory and fast rendering.

RELATED WORK

Alex Walker wrote an article [2011] discussing how the Cicada Principle relates to web design. Walker describes how layers with prime widths and heights generate large textures. Our work extends and formalizes his work for game textures.

Alternative to creating large textures, artists can use noise filtering [Stam 1997], aperiodic tiling methods [Parzer 2013], and texture compression [Wang et al. 2008]. These methods are dificult to generalize across a variety of categories of texture structures.

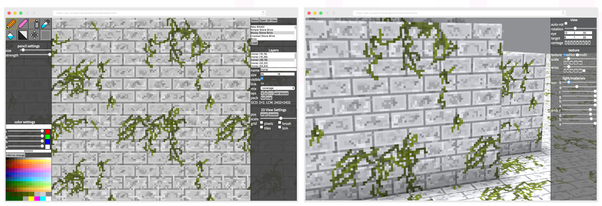


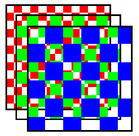
Figure1: (left) Web-based rpTexture editor (right) 3D viewer

IMPLEMENTATION

We created a prototype web application using WebGL to create low-res textures. The prototype includes a simple editor with basic tools, a color palette, a layer editor, and a 3D viewer to preview the texture in a scene. A GLSL shader produces the final color by covering or adjusting HSV each layer sampled at the texture coordinate mod the layer's size. The layers pack into a single glTexture.

DESIGN PRINCIPLES

The key insights to our work are:



1) create layers of different sizes that have periods corresponding to the features in the texture, and 2) add non-structural noise to decrease repetition.

Textures with features that are noisy and unstructured, such as grass, dirt, and sand, are created using layers with sizes that are relatively prime and therefore have a small GCD. For example, we could choose layer widths of 5, 7, and 11, which would generate a texture with a period of 5·7·11=385.

Textures with features that are highly regular and structured, such as brick, are created using layers with a GCD as a multiple of the feature width. For example, if the brick feature is 8 pixels wide, we could choose layer widths of 8·5=40, 8·7=56, and 8·11=88, which generates a texture with a period of 3080. Additional layers can be used to add texture features that have a different structure, such as moss on brick.

Textures that are structured but not regular, such as stone walls, water, and painted pavement, use layers with sizes that are relatively prime and imply structure by covering details using neutral colors and adding features.

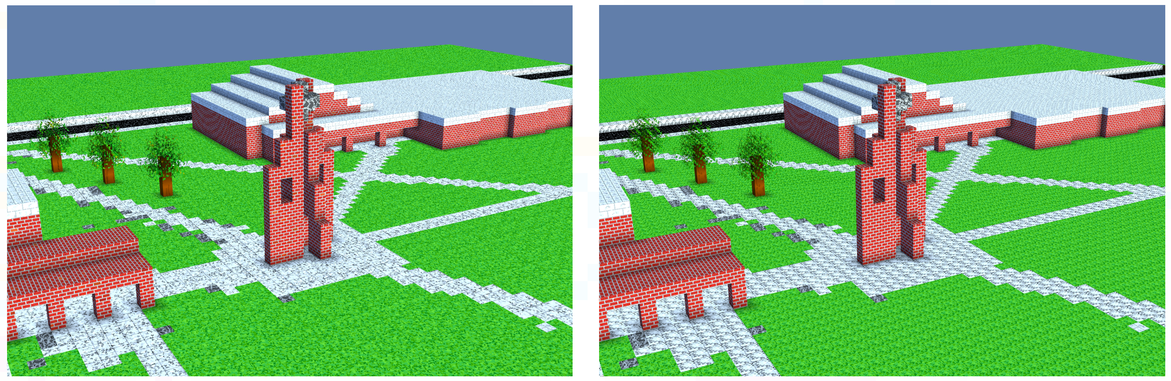


Figure 2: Large 3D scenes using rpTextures (left) and 16x16 tiled textures (right). Although the textures in both images are scaled the same, rpTextures avoids the repetition seen using tiled textures.

DESIGN STRATAGIES

We focused on creating low-resolution textures rather than realistic or highly stylized textures, so our strategies may not apply to all types of textures. However, we found the following basic guidelines helped with creating textures.

- As with most creative work, have a final goal in mind prior to starting.
- For irregular textures, paint the bottom layer in broad strokes with very few detail to set the tone. Regular textures can have basic structured details in bottom layer.
- Use large layer sizes and spread out the usage. Then use multiple layers of spread usage to cover the entire texture with the desired design.
- Add one or two additional HSV-adjusting layers to add "noise" to the texture, effectively reducing the repetitiveness without changing the overall structure..

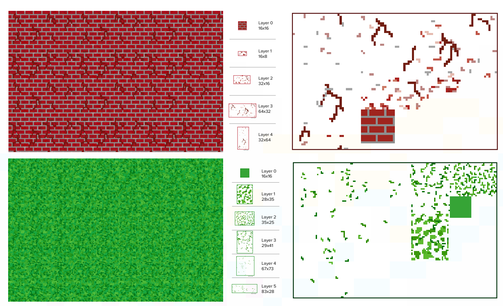


Figure 3: cracked brick and grass rpTextures (left), layers generating textures (middle), packed glTextures (right).

FUTURE WORK

We plan to run studies to determine general rules for creating textures that avoid repetition, and we plan to compare the performance of our method against methods common in industry. Currently, we focus on using textures for diffuse reflection of surface. We believe this work can easily extend to control other surface properties, such as specularity, emission, normal perturbation, and displacement. We plan to experiment with other ways of combining layer samples. We also believe that this method can extend to other domains, such as audio.

DATA

Least Common Denominator

Texture:	Width (px)	Height (px)
Water	131,710,070,791	131,710,070,791
Grass	90,310,640	8,380,400
Gravel	474,221,170,640	23,574,065,200
Stone	63,840	45,600
Wood Planks	533,907,780	12,600
Parking Lot	1,745,471,677,800	4,263,629,180,200
Brick Wall	24,960	6,336
Tree Bark	246,290,273,544	3,988,450,467,656

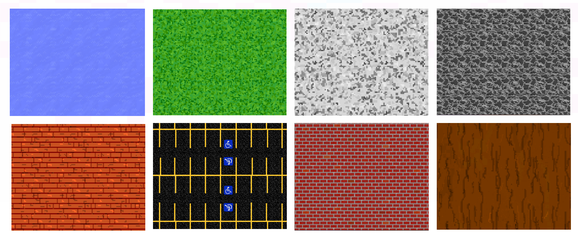


Figure 4: Various textures created with the prototype editor 1.) water 2.) grass 3.) gravel 4.) stone 5.) wood planks 6.) parking lot 7.) brick wall 8.) tree bark

REFERENCES

PARZER, S. 2013. Irrational Image Generator. Diploma thesis (Vienna Univ. Technol., Vienna).

STAM, J. 1997. Aperiodic Texture Mapping. European Research Consortium for Informa cs and Mathematics.

WALKER, A. 2011. The Cicada Principle and Why It Ma ers to Web Designers. www.sitepoint.com/the-cicada-principle-and- why-it-ma ers-to-web-designers/.

WANG, H., WEXLER, Y., OFEK, E., HOPPE, H. 2008. Factoring repeated content within and among images. ACM TOG 27, 3.