Art-Based Rendering of Fur, Grass, and Trees


SIGGRAPH 99

Michael Batchelder

Outline

- Introduction
  - Motivation
  - Dr. Seuss and others
- Background Work
- Approach
- Results
- Related Work
- Conclusion

Dr. Seuss – Self-Portrait of an Artist Worrying about His Next Book

Introduction

Motivation

- Would like to simulate real world objects of large complexity, such as grass or trees.
- Alternate (NPR) styles (ala Dr. Seuss).
- Should be fast enough for real-time animation.
- Would like some level of temporal coherence in animation.

Trees

Cover Art for The Lorax – Dr. Seuss

Fur/Hair

Hop on Pop – Dr. Seuss

Cover Art for The Lorax – Dr. Seuss

Curt Kirkwood
www.curtkirkwood.com

Dr. Seuss – The Lorax
Background Work

William Reeves “A Technique for Modeling a Class of Fuzzy Objects” - 1983
- Used particle systems used to represent explosions, fireworks, clouds, and water.
- Over time, particles can be added, removed, and moved to represent a dynamic 3D model.
- Motion-blurred to deal with temporal coherence.

Expanding Wall of Fire - Reeves

Alvy Ray Smith “Plants, Fractals, and Formal Languages” - 1984
- Used particle systems combined with recursively defined L-Systems that he called “graftals”.
- Flowering Plants are Graftals, Grasses are particle systems.

White Sands - Alvy Ray Smith

L-Systems introduced by Lindenmayer in 1968
- Parallel rewriting grammars.
- Bracketed L-Systems allow for branches to be attached at points within the sequence, like branches off a trunk (or off other branches).
  - Extension: Notion of left and right
  - Similar to Fractals.

Cartoon Tree - Kowalski calls it “direct precursor to the work in this paper”.

Cartoon Tree produced from Example Grammar
Alvy Ray Smith – Plants, Fractals, and Formal Languages

Example:
- Alphabet: \{ 0, 1, [ , ] \} note: (, )
- Productions:
  \{ 0 \rightarrow 1[0]1[0]0, 1 \rightarrow 11, \} \rightarrow [, ] \rightarrow \}
- Start: 0

Figure: Producer 0

Alvy Ray Smith - Plants, Fractals, and Formal Languages
Background Work

- Badler and Glassner generalize idea in “3D Object Modelling”: Graftals create surfaces via an implicit model that produces data when requested.

Outline

- Introduction
- Background Work
- Approach
  - Procedural textures (Graftals!)
  - Reference images & Difference Image Algorithm
- Results
- Related Work
- Conclusion

Approach

- Stroke-based textures are implemented within a general system for rendering polyhedral models using OpenGL
  - Models are divided into surface regions (patches) which are assigned procedural textures.
  - Textures can be smooth-shaded, wireframe, hatching, dithering, etc.
  - Color and ID references images maintained for use by procedures

Approach

- Color reference image.
  - Procedural textures are asked to render into their patches in some appropriate way, depending on how the texture will use the image.
  - For example, graftals use color image in special way to decide where to place tufts of fur, grass, or leaves.

Approach

- ID reference image.
  - Triangles (or edges) are rendered with unique color that identifies that triangle (or edge).
  - Pixels containing the ID of a triangle have their location stored in a list attached to the patch they are in.
  - Later, procedural textures have acces to these pixels in main rendering loop.
  - Example: the dithering texture just runs the Floyd-Steinberg algorithm on the patch’s pixels
Approach

- **Graftals**: specialized textures
  - Graftals must be placed with controlled screen-space density to match the required aesthetics of the texture.
  - Graftals also need to “stick” to surfaces to provide interframe coherence.
  - Difference Image Algorithm of Salisbury, et al used to place graftals.

---

Approach

- **Difference Image Algorithm**:  
  - In Salisbury, et al, DIA used to control density of hatching strokes to match gray tones of target image.
    - For each stroke, a blurred image of it is subtracted from the difference image.
    - Next stroke is placed by searching for the pixel “most in need” of darkening.
    - Density of strokes conveys gray tones of original image.

---

Approach

- **Graftals with DIA**:  
  - Texture draws its patch into the color ref. Image so darker tones correspond to areas requiring more graftals.
  - Example: reference image drawn darker at silhouettes and, optionally, explicitly darkened by user in certain regions. (feet)

---

Approach

- **Placing Graftals on the 3D Surfaces**:  
  - In first frame, use DIA to find graftal positions and convert 2D screen positions to 3D positions on model surfaces using ID reference image to find the edge / triangle the graftal belongs to.
  - In each successive frame, first try to place graftals from previous frame.
  - After previous graftals are accepted/rejected, execute DIA to place extra graftals as needed.

---

Approach

- **Why would the previous frame’s graftals be rejected?**  
  - It may not be visible (occluded, or off screen)
  - Insignificant desire
    - Scene has been zoomed out so too many graftals making for too dense a patch
    - Scene has been rotated/transformed such that graftal is no longer near the silhouette

---

Approach

- **How to blur difference image?**  
  - Represent graftals as dots centered at its pixel
  - Dots are gaussian-blurred
  - Dot size is proportional to graftal’s “volume” - the amount of screen-space area it takes up
  - Dots represent “lack of desire” - that is, the greater the number of graftals you have, the less desire there is for more
  - What happens when the scene is zoomed in/out?
**Approach**

- **Scaling**
  - Convert object length $L$ to screen space measurement $s$.  
  - Choose scale factor $r$ to multiply $L$ by.  
  - Desired screen space is $d$ with volume $v_0$.  
  - Kowalski chose weighted $r = w \left( \frac{d}{s} \right) + (1 - w)$ where $w = 0.25$ to moderate degree of scaling.  
  - So volume in each frame is $v = v_0 \left( \frac{r s}{d} \right)^2$  

- **Gaussian Blur**
  - The value in the desire image at the graftals (visible) screen position $x$ is $d_x$ and let $v > 0$ be the volume.  
  - Find a 2D Gaussian function $g$ such that:  
    - This is given by:  
    
    $$g(x) = d_x e^{-\frac{x^2}{4}}$$

- **Graftal Detail**
  - A graftal can only subtract as much desire as is available at a pixel.  
  - If all goes well, it subtracts $v$ desire and the graftal is drawn.  
  - If the less than $v$ is subtracted, graftal may draw with less detail.  
  - If less than some threshold (0.5 in paper), graftal is drawn at all.  
  - To reduce "popping" graftals may initially be drawn with low detail, and quickly ramp up to full detail over a few frames and vice-versa when being removed.

- **Drawing Graftals**
  - Example of Fur graftals:

![Figure 5. A fur graftal is based on a plane polyline and table of widths, used to construct a GL triangle strip. The graftal can consist of itself as their steps. It consists of filled polygons with streaks along both borders (blue and red) so that a hair on the spine (c).](image)

  Fur graftal geometry – Kowalski et al.
Approach

- Drawing Graftals
  - Graftals drawn in the plane of the surface normal and most nearly orthogonal to the eye vector.
  - De Facto graftal placement: bending down.
  - Can be oriented to follow directions “painted” on texture patches, as on furry creature’s feet:

![Furry creature feet](image)

Outline

- Introduction
- Background Work
- Approach
- Results
  - Trees, Furry Creature, Dr. Seuss Scene
  - Videos
- Related Work
- Conclusion

Results

- Furry Creature – Kowalski, et al.
- Tree – Kowalski, et al.
- “Dr. Seuss” Rendered Scene – Kowalski, et al.

Related & Future Work

- Improve “popping”
  - DIA has no interframe consistency.
    - Could use alpha blending and fading but it’s limited in it’s usefulness.
  - Maintain graftals for back-facing surfaces so silhouette graftals do not pop in and out.
  - Maintain static graftals with level-hierarchy.
- Explore new styles
  - Example: bi-layered fur, one dark one light to suggest complex lighting effects.

Related & Future Work

- Three-level detail hierarchy of Truffula tree – Kowalski, et al.
Related & Future Work

*More recent paper “Art-based Rendering with Continuous Levels of Detail” (NPAR 2000) Addressed issues in original paper. (with Meier)*
- Abstract procedural textures to a texture file.
- Static graftals, with complex level of detail model
- Look/behaviour of graftals as they disappear and re-appear animated to create smooth transitions.

Conclusion

- Good and interesting results for single-frame.
- Still some issues with animations, even in newer paper.
- Average computers today can handle large FPS without much problems.
- Limited to programmer-artists due to need for writing code to produce procedural textures.
- Lots of work to "describe" a scene with all its different patches.
- Very similar to Stipple paper I presented before!

The End

“Art Based Rendering of Fur, Grass, and Trees”
SIGGRAPH 99

Adobe Systems: Lubomir Bourdev
Pixar: Ronen Barzel