EECS 487
Non-photorealistic Rendering

Lee Markosian
April 9, 2007
Whether to use photorealism depends on the purpose of the image:

- Training/simulation
- Documentation
- Illustration
- Story-telling
- Expression
Whether to use photorealism depends on the purpose of the image:

- Training/simulation (yes)
- Documentation (yes)
- Illustration (not usually)
- Story-telling (sometimes)
- Expression (sometimes)
Qualities of hand-drawn images

• Many details left out
• Some details emphasized
• Stylization / abstraction
  – focus on essential rather than particular
  – simple 2D marks evoke complex things
• Recognizable individual style
Realistic modeling and rendering of plant ecosystems, SIGGRAPH 1998.
Elbow starting to be flexed by brachialis and biceps contracting
Vesalius (1514 - 1564)
Talk overview

• motivation
• technical illustration
• pen & ink rendering
• painterly rendering
• graftals
• stroke-based rendering
• tonal art maps
Technical illustration

• Saito and Takahashi, SIGGRAPH 1990
• “Comprehensible” rendering style
Method

- Render intermediate images
- Do image-processing
- Combine results
Fig. 9 Process of drawing illustrations.
Discussion

• Main idea: G-buffers
  – i.e., “reference images”

• Approach is general
  – basis for many subsequent algorithms
  – works regardless of shape representation
  – image space operations -- good for NPR
Discussion

• Main idea: G-buffers
  - i.e., “reference images”

• Approach is general
  - basis for many subsequent algorithms
  - works regardless of shape representation
  - image space operations -- good for NPR
    (e.g. uniform stroke spacing in image space)
Discussion

• Parameters need careful tuning
  – can be a problem in animations
    (e.g. when camera changes)
• Hard to convey strokes \textit{on} surfaces
• Many open questions
talk overview

- motivation
- technical illustration
- pen & ink rendering
- painterly rendering
- graftals
- stroke-based rendering
- tonal art maps
Pen and Ink

- Winkenbach and Salesin, SIGGRAPH 1994
- Purpose: render 3D models in pen & ink style
Method

• Annotate model with procedural “textures”
  – depend on tone and magnification
• Render tone “reference image”
• Use it to guide pen and ink textures
Smaller versions have fewer strokes (not just smaller strokes)
dependence on orientation
Comments

• Use of several different textures
• Note shadow (how is it achieved?)
• Believable as a sketch?
Comments

• Use of several different textures
• Note shadow (how is it achieved?)
• Believable as a sketch?
  – Yes... but a little too neat and uniform
  – (part of roof is missing!)
Indication

- User decides where detail is permitted
- It drops out in other places
user's marks selecting detail areas
Pen and Ink

- Salisbury, Anderson, Lischinski and Salesin, SIGGRAPH 1996
- Purpose: define a scale-independent representation for pen & ink images
Scaling with fixed number of strokes: bad
Scaling with variable number of strokes: good
Method

• store greyscale image annotated with discontinuities

• filter greyscale image to desired size, run stroke generation algorithm on it

• repeatedly try to generate a stroke at random location in image

• keep stroke if it does not exceed target darkness
Detect sharp features, then modify blurring algorithm to preserve them
Problems

- Only produces still images
  - Would not provide temporal coherence
- What’s the application?
talk overview

- motivation
- technical illustration
- pen & ink rendering
- painterly rendering
- graftals
- stroke-based rendering
- tonal art maps
Painterly rendering: Meier 1996
Painterly rendering

- Meier, SIGGRAPH 1996
- Problem: achieve “painterly” style with temporal coherence of strokes
- Method:
  - populate surfaces with stroke “particles”
  - get stroke attributes from reference images
video
Problem

• Particles have fixed distribution
  – Need prescribed camera path

• newer work addresses that:
  *A dynamic drawing algorithm for interactive painterly rendering:*
    http://artis.imag.fr/Publications/2006/VBTS06/
talk overview

• motivation
• technical illustration
• pen & ink rendering
• painterly rendering
• graftals
• stroke-based rendering
• tonal art maps
Graftal textures

Detail elements (graftals) generated as needed
Graftals

- simple bit of geometry (e.g. leaf or “tuft”)
- oriented in local frame
How to distribute graftals?

Needed:

– Controlled screen-space density
– Placement *on* surfaces
– Controlled placement (e.g. only near silhouettes)
– Persistence between frames
Method

• Use persistent collection of graftals
  – Each frame, some are created, some destroyed, others persist

• “Desire” image tells where graftals are needed
  – Dark tones = greater need for graftals

• Upon placement, each graftal locally removes darkness from the desire image
Problems

• Graftal textures defined in code
  – hard to edit
  – how to integrate with UI?

• Coherence
  – Graftals still appear/disappear suddenly
  – Better at low frame rates!
Art-based Rendering w/ Continuous Levels of Detail.
Markosian, Meier, Kowalski, Holden, Northrup, & Hughes.
NPAR 2000.
Basic graftals

- Collection of drawing primitives
  - triangle strips / fans
  - plus strokes
- Shared vertices
- Local coordinate frame
- Tuft: hierarchy of graftals
The local frame

- Base position (e.g. on surface)
- $y'$ (e.g. surface normal)
- $x'$ (e.g. cross product of $y'$ and view vector)
Placement and duplication

• Designer creates a few “example graftals”
• Copies can be generated on surfaces
  – explicitly, or
  – procedurally
• Random variation can be used
  – copies are not exact
  – looks less mechanical
Level of detail (LOD)

- Graftal computes current LOD
- Decides which primitives to draw
Computing LOD

• LOD can be derived from:
  – apparent size
  – orientation
  – elapsed time
$\sigma$: ratio of current size to "rest" size

$\sigma = 0.7$  $\sigma = 1$  $\sigma = 1.4$
Orientation

• Value used to selectively suppress LOD

• E.g.: \(1 - |v \cdot n|\)
Movie
Discussion

• Coherence: much better!
• Slower
• Introducing / removing elements
  – Fading & thinning work well
  – Growing looks creepy
• LOD mechanism too inflexible
• Need direct UI
Pen & Ink: trees

- Deussen and Strothotte, SIGGRAPH 2000
- Problem: temporally coherent pen and ink rendering of trees
- Method:
  - Draw leaf entities w/ controlled size/abstraction
  - Do image processing on depth buffer
talk overview

- motivation
- technical illustration
- pen & ink rendering
- painterly rendering
- graftals
- stroke-based rendering
- tonal art maps
Contributions

• Direct user-control for NPR
• Better silhouettes
• New media simulation
• Stroke synthesis by example
• Hatching with LODs
Overview of Components

- Base Coat
- Brush Style
- Paper Effect
- Decals
- Outlines
- Hatching
Brush Style

Per stroke:

- Color
- Width
- Paper effect

Rendered as triangle strips.
Strokes in OpenGL

Based on “Skeletal strokes”
Hsu et al., UIST ’93
Paper Effect

• Height field texture:
• Peaks catch pigment
• Valleys resist pigment

Implementation:
• Pixel shader
• Stroke alpha = pressure
Re-map alpha (stroke pressure) with a “paper texture” heightfield

- **Peak**: Graph shows a linear increase from 0 to 1.
- **Valley**: Graph is flat from 0 to 1.
- **Intermediate**: Graph shows a gradual increase from 0 to 1.
video
talk overview

• motivation
• technical illustration
• pen & ink rendering
• painterly rendering
• graftals
• stroke-based rendering
• tonal art maps
Real–Time Hatching

Emil Praun
Hugues Hoppe
Matthew Webb
Adam Finkelstein

Princeton University
Microsoft Research
Princeton University
Princeton University
Goal

• Stroke-based rendering of 3D models
• Strokes convey:
  – tone
  – material
  – shape
Challenges

- Interactive camera and lighting control
- Temporal (frame to frame) coherence
- Spatial continuity
- Artistic freedom
Approach

Example stroke

Preprocess

Set of textures

Mesh

Real-Time

Result
Tonal Art Maps

- Collection of stroke images
- Will blend → design with high coherence
- Stroke nesting property
Generating Tonal Art Maps

• Draw or import bitmap for one stroke
• Automatically fill TAM with strokes
  – When placing stroke in an image, add it to all finer & darker images
  – Fill table column by column, coarse to fine
  – Space strokes evenly
Even Spacing of Strokes

• Choose best stroke from large candidate pool
• Fitness = uniformity & progress towards tone
Even Spacing of Strokes

- Choose best stroke from large candidate pool
- Fitness = uniformity & progress towards tone
Texture Blending

6-way blend → final
Texture Blending

• Pack grayscale tones in R,G,B channels
  6 tones in 2 textures

• Use multitexture engine
  single-pass 6-way blend

• Vertex programs compute blend weights

!!VP1.0 #Vertex Program for Real-Time Hatching.
//output vertex homogeneous coordinates
DP4   R2.x, c[0], v[OPOS];
DP4   R2.y, c[1], v[OPOS];
DP4   R2.z, c[2], v[OPOS];
DP4   R2.w, c[3], v[OPOS];
MOV   o[HPOS], R2;
//stroke texture coordinates, transformed
DP3   o[TEX0].x, c[4], v[TEX0];
DP3   o[TEX0].y, c[5], v[TEX0];
DP3   o[TEX1].x, c[4], v[TEX0];
DP3   o[TEX1].y, c[5], v[TEX0];
// splotch mask coordinates
MOV   o[TEX2], v[TEX0];
//get the Gouraud shade
DP3   R1, c[8], v[NRML];
MAD   o[COL1], R2.y, c[A0.x + 11], R3;
MOV   R4, c[A0.x + 20];
Texturing Arbitrary Surfaces

• Lapped Textures
  [Praun et al. 2000]
Direction Field

- Based on surface principal curvatures
- Optimized to be smooth
  - [Hertzmann & Zorin 2000]
  - Symmetry: 180° instead of 90°
Summary

• Real-time hatching for NPR
• Strokes rendered as textures
• High coherence TAMs prevent blend artifacts
• 6-way blend very fast on modern graphics
Future Work

- More general TAMs
- View-dependent stroke direction
- Automatic indication
Next up

- Wednesday: more recent NPR
- Next Monday:
  - course evaluations
  - review for final
  - homework due
- Final exam:
  - April 20, 4-6 pm
  - CSE 1670