The Choices We Make

Frédo Durand
MIT CSAIL
Introduction

• My background
  – Math/CS
  – I have read much perception litterature
  – Amateur photographer
  – No artistic training, terrible draftsman

• Two example projects
  – Photography tonal management
  – Line drawing from 3D models
Introduction

• I don't build tools, I am an academic researcher, my deliverables are articles, not software
  – But it's not a complete excuse to write about useless tools

• Two types of "consumers/users"
  – Computer scientists who implement/extend my techniques
    * Do they understand choices I made, choices they can make?
  – En-users who use these tools
    * Are my choices relevant to them?
Tonal management

- Over and under-exposure is the largest cause of bad photographs
  - Here's a choice I make!

- Both for professional and consumers
High Dynamic Range

- Real-world contrast is high
- Display contrast is low
Our approach

- Non-linear two-scale decomposition
- Reduce contrast of large scale; preserve local detail
Live demo
Choices I made

• Decomposition: my contribution
• Compute in log space: I have good reasons
• How to separate intensity/color - incidental
• How to reduce large-scale layer – incidental
• Parameters I expose
• Default parameters – matter of taste
• Maybe other choices I don't realize I made
Anecdote about tone mapping evaluation

• Recent work has performed user experiments to evaluate competing tone mapping operators

• Interestingly, the former concludes my method is the worst, the latter that my method is the best!
  – They choose to test a different criterion: fidelity vs. preference

• More importantly, they focus on algorithm and ignore parameters
A programmable approach to Line Drawing
With Stephane Grabli, Emmanuel Turquin & François Sillion
Motivation: Style vs. Technique

• Non-Photorealistic Rendering
  – Imitate traditional media

• Each paper focuses on one particular style, which is usually hardcoded with a only few available parameters

• Stylistic choices mixed with technical ones
Goal: Decouple style from technique

• **First step: pure line drawing**
  System for rendering line drawing from 3D scenes
  – Including a flexible style description tool
Goal: Decouple style from technique

- First step: pure line drawing
  System for rendering line drawings of 3D scenes
  – Including a flexible style description tool
Goal: Decouple style from technique

- **First step: pure line drawing**
  - System for rendering line drawing from 3D scenes
    - Including a flexible style description tool
    - Ensuring model independence
Goal: Decouple style from technique
Style in line drawing

• Occlusion and nature $\rightarrow$ thickness
Style in line drawing

• Depth discontinuity $\rightarrow$ thickness
Hypothesis

• Drawing can be described as a process

• Stylistic decisions (line thickness, omission) are related to scene and image information

• These decision strategies can be embedded in procedures
Some relevant information

- Geometry (2D, 3D coordinates, normals, …)
- Differential geometry (2D, 3D curvatures, …)
- Line adjacency
- Line nature (silhouette, crease, contour, …)
- Occluding information (visibility, occluder, depth discontinuity, …)
- Material
- Object id
- Drawing density
Approach

3D « View Map » + information

Style

Drawing
class pyGuidingLineShader(StrokeShader):
    def shade(self, stroke):
        it = stroke.strokeVerticesBegin()
        itlast = stroke.strokeVerticesEnd()
        itlast.decrement()
        t = itlast.getObject().getPoint() - it.getObject().getPoint()
        itmiddle = StrokeVertexIterator(it)
        while(itmiddle.getObject().u()<0.5):
            itmiddle.increment()
        while(it.isEnd() == 0):
            it.getObject().SetPoint(itmiddle.getObject().getPoint() \
                                +t*(it.getObject().u()-itmiddle.getObject().u()))
            it.increment()

Operators.select(QuantitativeInvisibilityUP1D(0))
Operators.bidirectionalChain(ChainSilhouetteIterator())
Operators.recursiveSplit(
    Curvature2DF0D(),
    pyParameterUP0D(0.2,0.8),
    NotUP1D(LengthHigherUP1D(75)), 2)

shaders_list = [
    StrokeTextureShader("pencil.jpg", Stroke.DRY_MEDIUM, 1),
    ConstantColorShader(0,0,0,1),
    ConstantThicknessShader(2.0),
    pyGuidingLineShader(),
    pyBackboneStretcherShader(0.2)]

Operators.create(TrueUP1D(),shaders_list)
Style operators

- shade
- select
- chain
- split
Style operators

- shade
- select
- chain
- split
Shading

Plain strokes

Thickness

Geometry

Color

Information dependent
Shading

Depth discontinuity → thickness
Style operators

- shade
- select
- chain
- split
Splitting

Split at points of highest 2D curvature
Splitting

Split at points of highest 2D curvature
Results
Oriental style
Density-based emphasis
Technical illustration style
Recap

• **Procedural description for style in line drawing**
  – Line drawing from 3D models
  – Control topology, geometry & attributes of strokes

• **Choices**
  – Automatic picture generation from 3D model
  – Restrict to pure line drawing
  – Describe style using procedures
  – View map, types of lines
  – Information we provide
  – Types of operators
Choices We Make

- Model
- Algorithms
- Parameters
- User Interface

- Problems we choose
- Evaluation criteria

- In articles, we must explain the respective importance of choices
It's the question, stupid!

• The important is not the answer to a choice, it's the choice of the question

• And even more importantly, the implicit choices we make without asking the question
Other choices

• Do we target pro of casual users?
• How automatic should things be?
Bad choice consequences

• No uses our technique: we don't address any problem, or give the wrong solution
• People are frustrated by our technique
• We make something too easy, becomes uniform

• Ethical problems