# Non-Photorealistic Rendering

Interacting with NPR

Tobias Isenberg

# Overview: Interactivity in NPR

- rendering speeds to achieve interactive to real-time image refresh rates
  - challenge in most computer graphics
  - challenge, in particular, in NPR (despite GPUs)

  → sometimes need to render less (screen space)

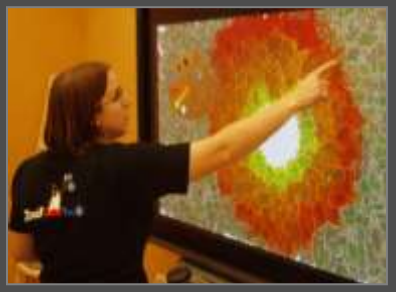
  → but need to figure out **what to render**

  → sometimes **many primitives** (strokes) to consider
- interaction models for affecting parameterization
  - not good: huge parameter lists, sliders, buttons, fields, ...
  - important: intuitiveness and usability
  - interactivity needs to be incorporated in the process

# Interacting with NPR: General Areas

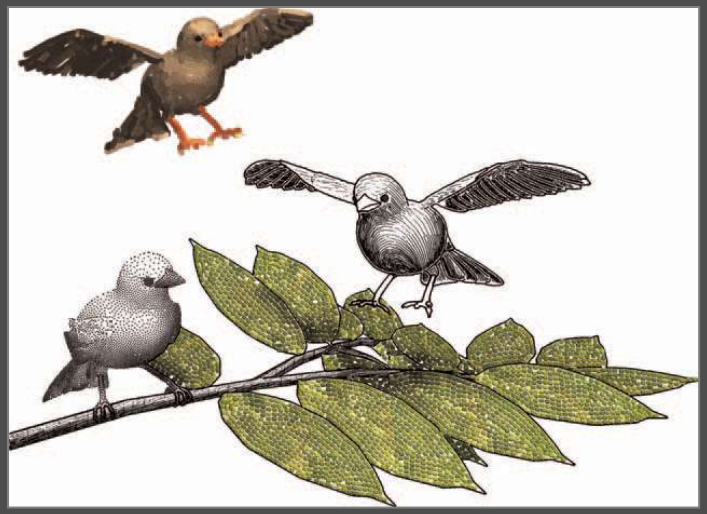
- interaction with stroke-based rendering
- interacting/drawing with strokes
- hardware-driven interaction
- triggering algorithmic response
- other related interaction approaches

→ not a clear classification



# Interacting with NPR

Interaction with Stroke-Based Rendering and Strokes Simulating the Artistic Environment

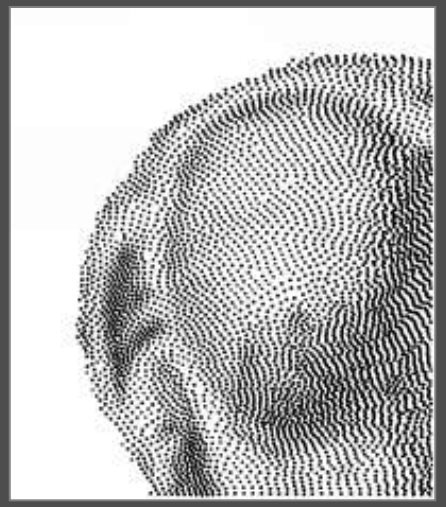


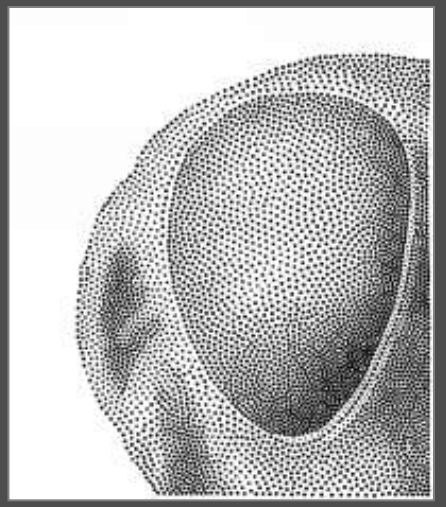
# Interacting with NPR

- first: three general areas together
  - interaction with stroke-based rendering
  - interacting with strokes
  - simulating the artistic environment/ hardware-driven interaction
- not always a clear distinction
- introduction of several approaches

# Interaction with Stroke-Based Rendering

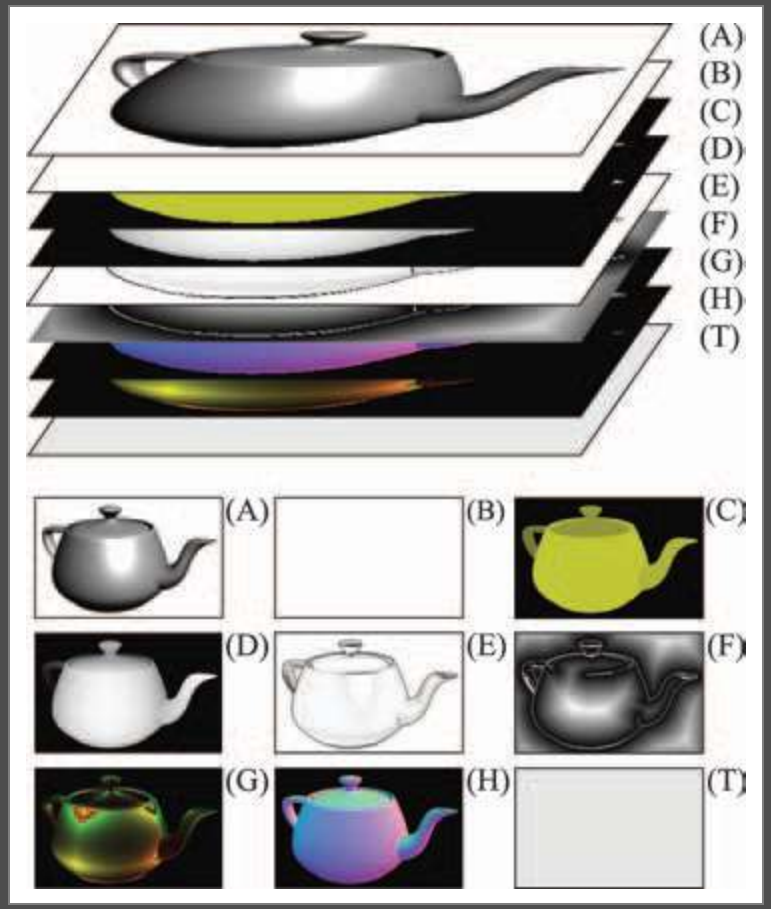
- interaction for user-controlled application of SBR with Lloyd's method
  - brushes for relaxation, jittering, stipple shape, new points
  - artists can control the relaxation process precisely
  - local, spatially explicit control of the SBR process



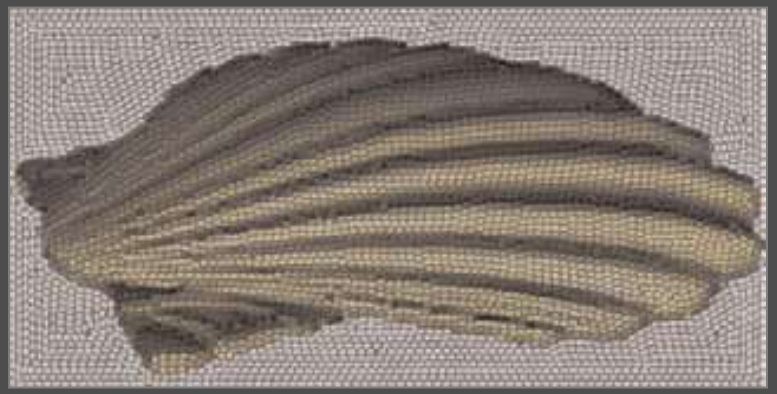


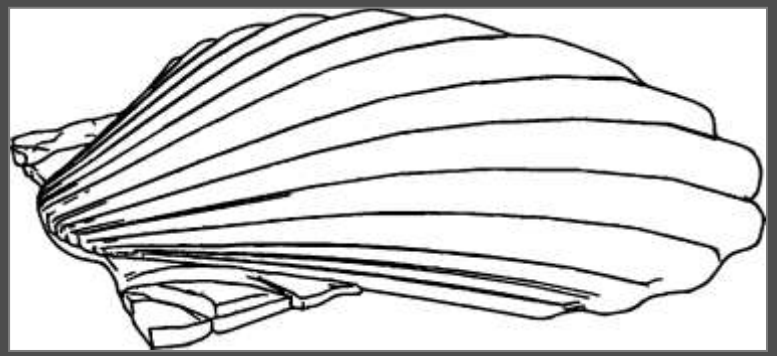
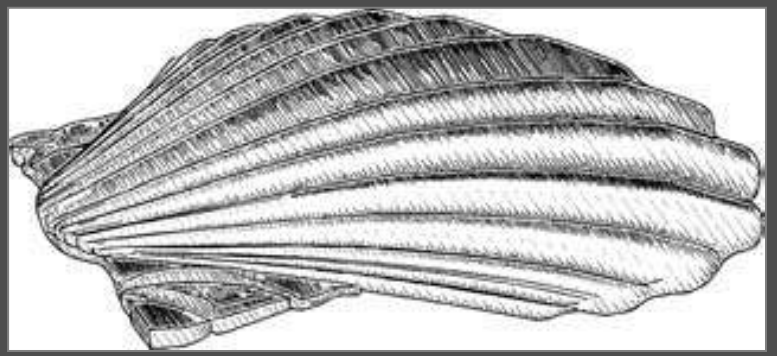
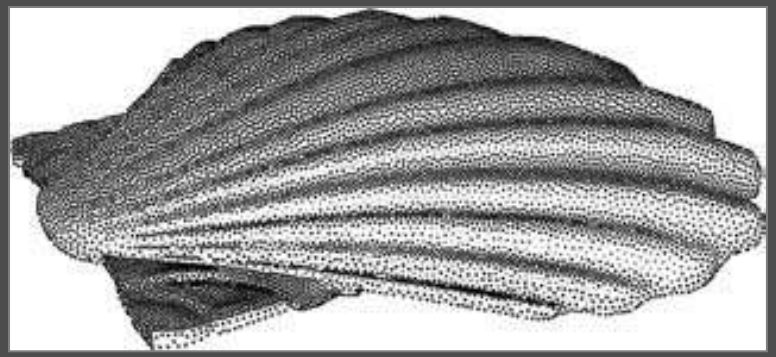
# Interaction with Stroke-Based Rendering

- RenderBots: user-controlled semi-automatic SBR
  - several layers of input data
  - RenderBot: local agent, can read input layers, sees its local neighborhood
  - iterative process, motion and rendering based on type, input data, local configuration
  - stipple bots: Lloyd's method
  - line bots: follow edges
  - mosaic bots: line up with edge-derived vector field
  - painting bots: render themselves as paint brushes

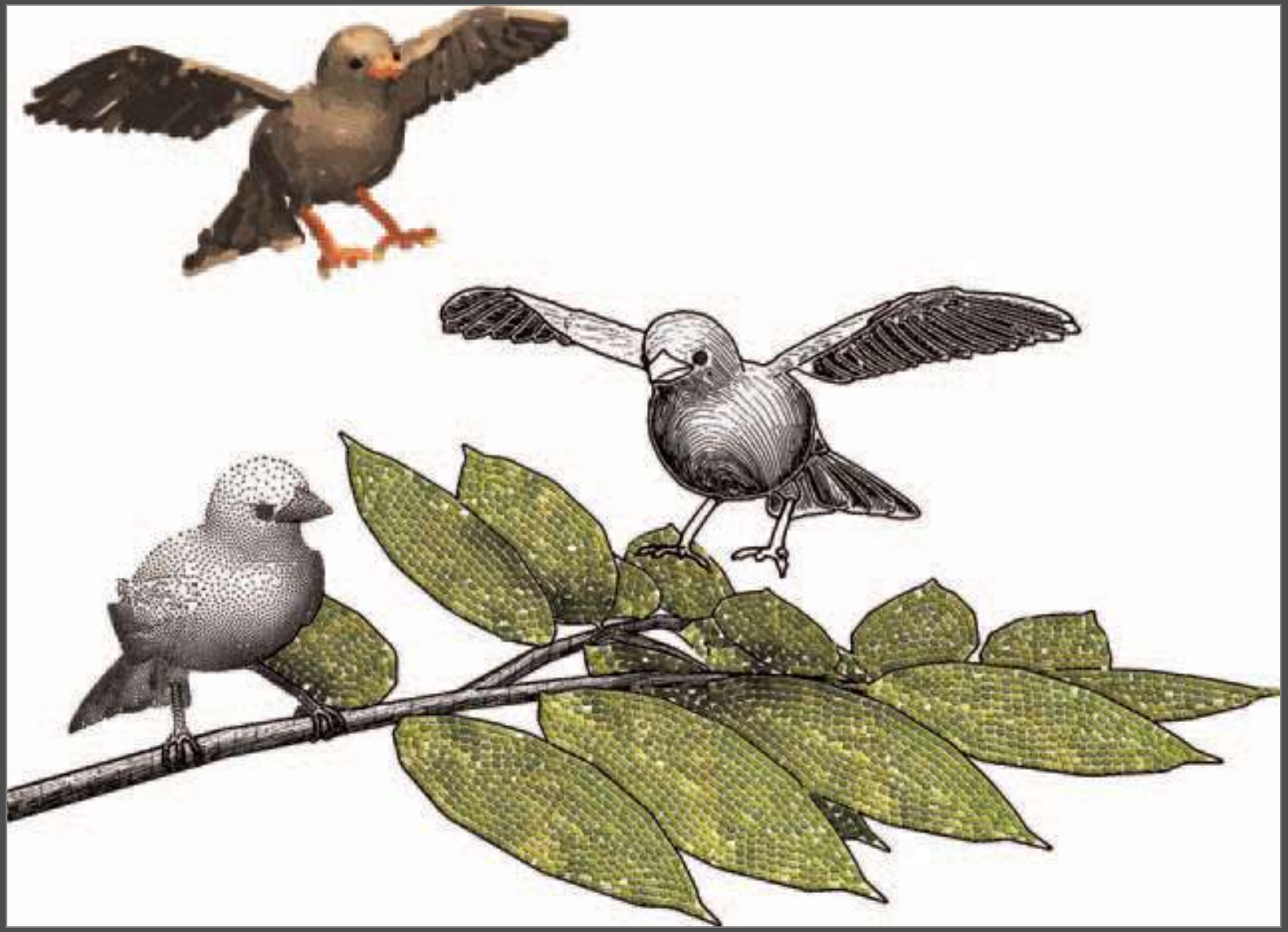


# Interaction with Stroke-Based Rendering



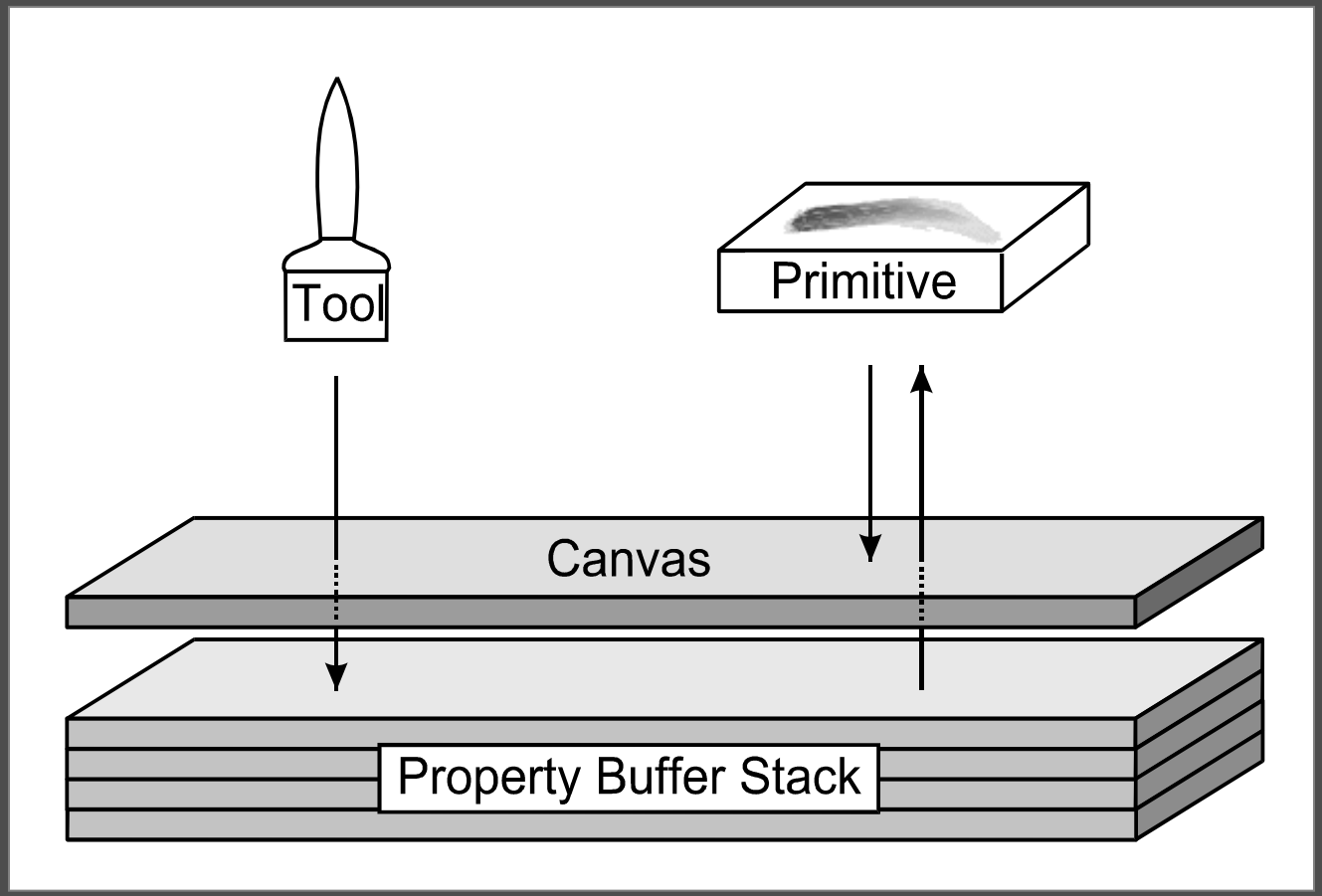




# Interaction with Stroke-Based Rendering

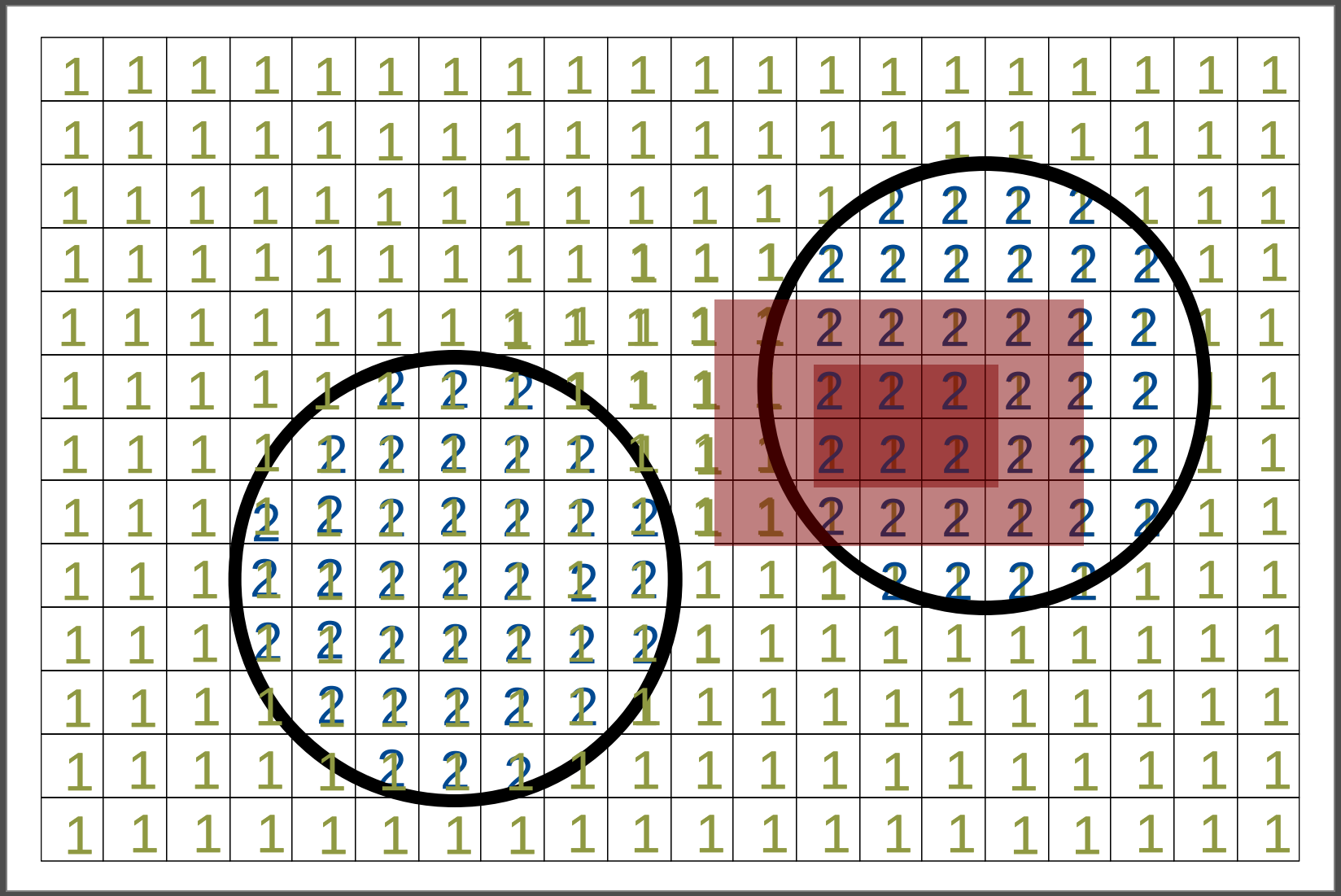


# Interaction with Stroke-Based Rendering

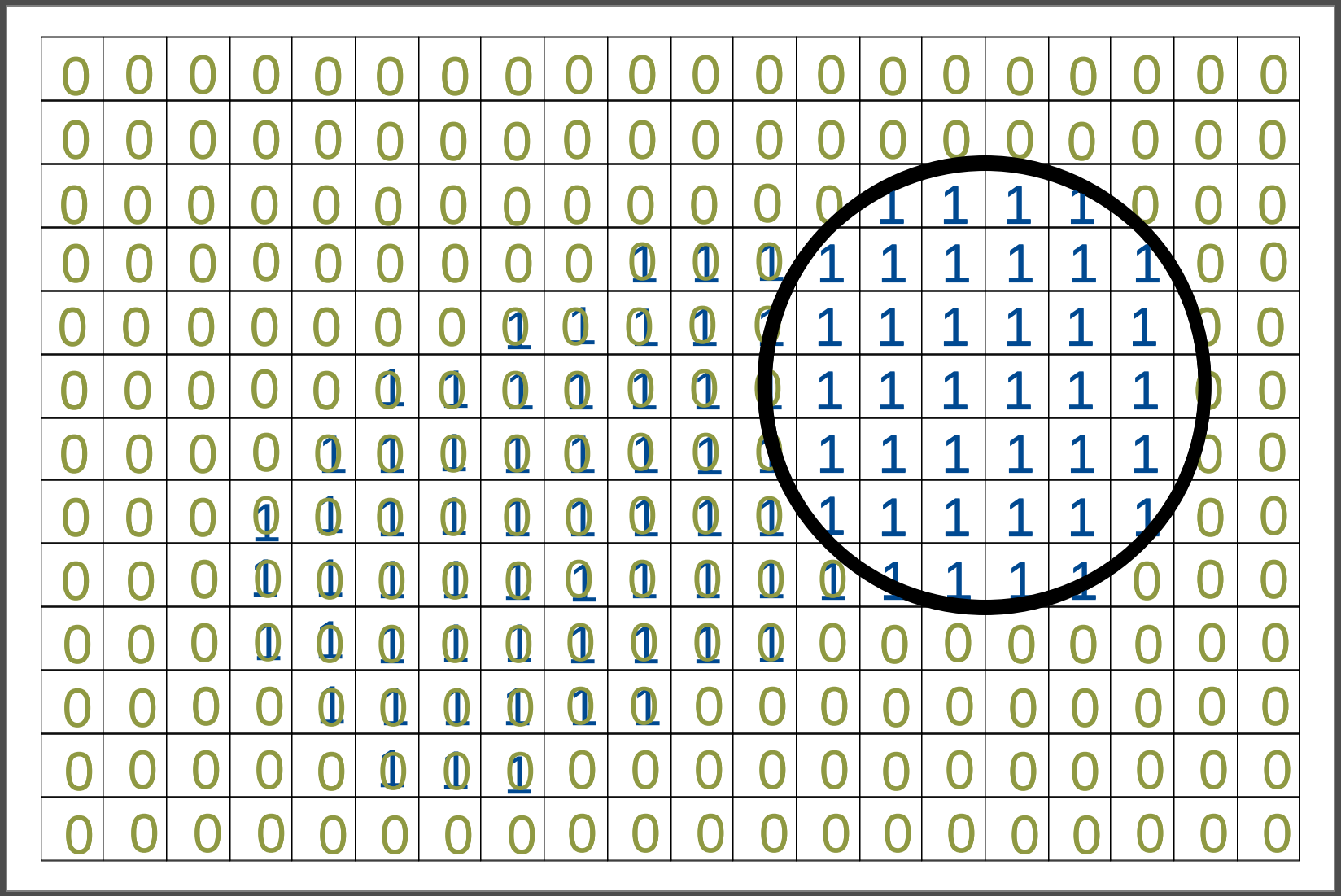
- interactive manipulation of data layers that control the strokes: use of interaction buffers (i-buffers)



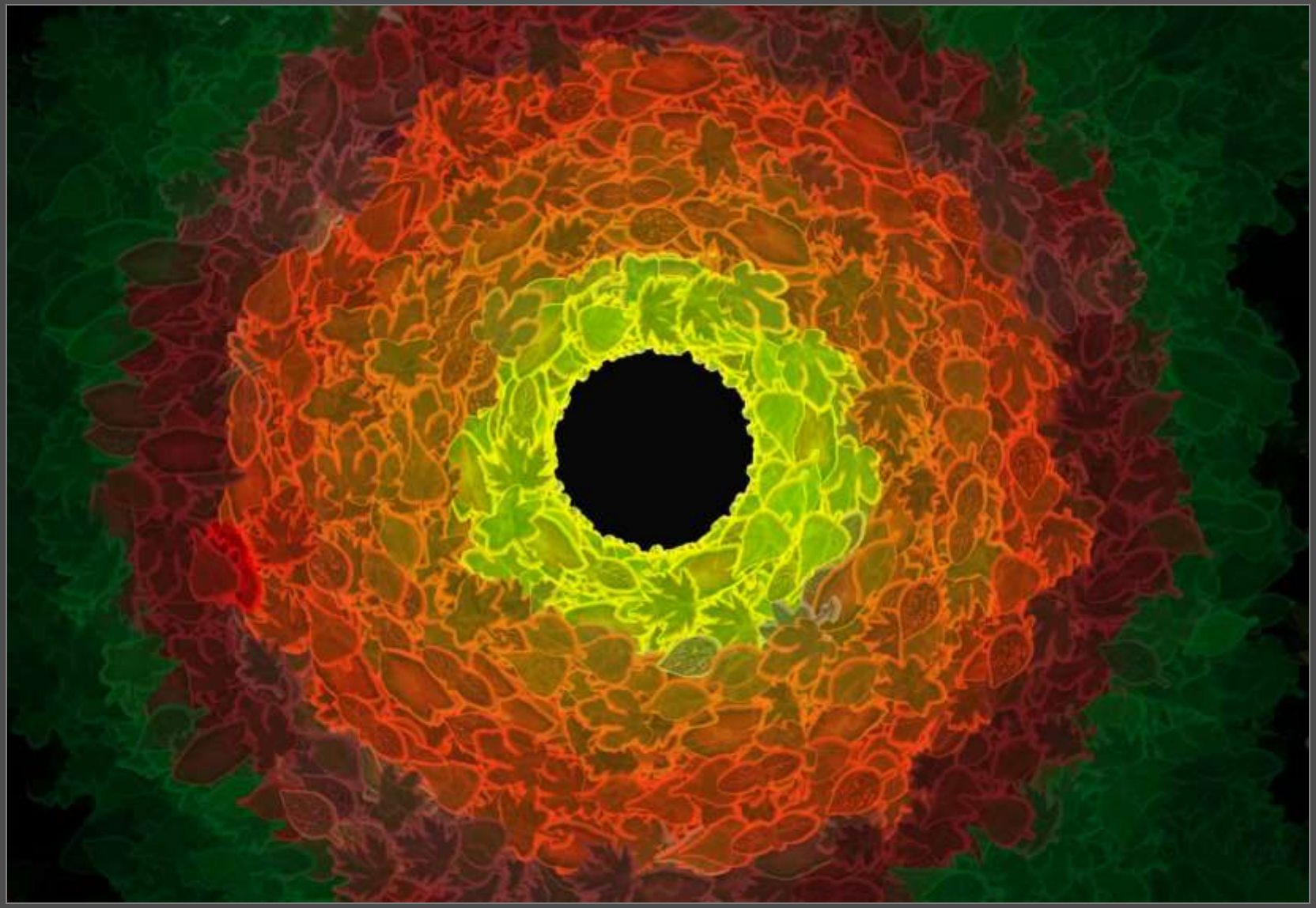
# Controlling Objects through i-Buffers



# Persistent vs. Instantaneous i-Buffers



# Stroke-Based Rendering with i-Buffers



with M. Schwarz & S. Carpendale

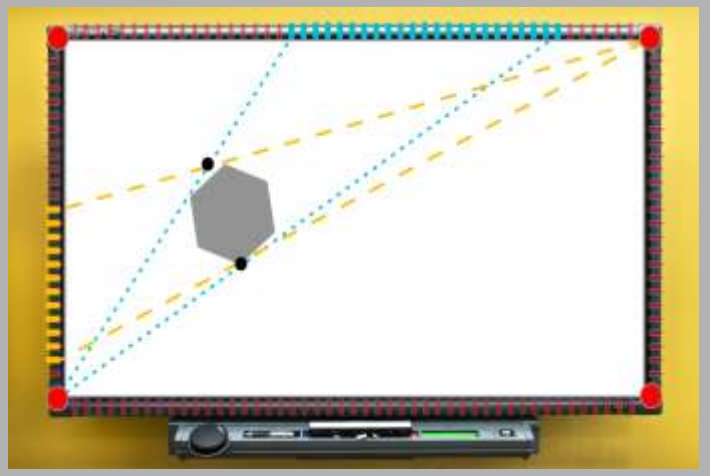
# Stroke-Based Rendering with i-Buffers

with M. Schwarz & S. Carpendale

# Stroke-Based Rendering with i-Buffers

with M. Schwarz & S. Carpendale

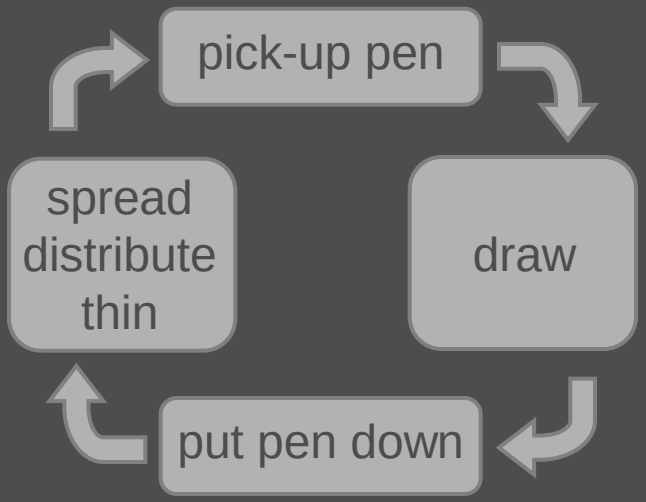
# Interaction with SBR via Hand Postures



# Interaction with SBR via Hand Postures

- direct-touch interaction for stroke placement

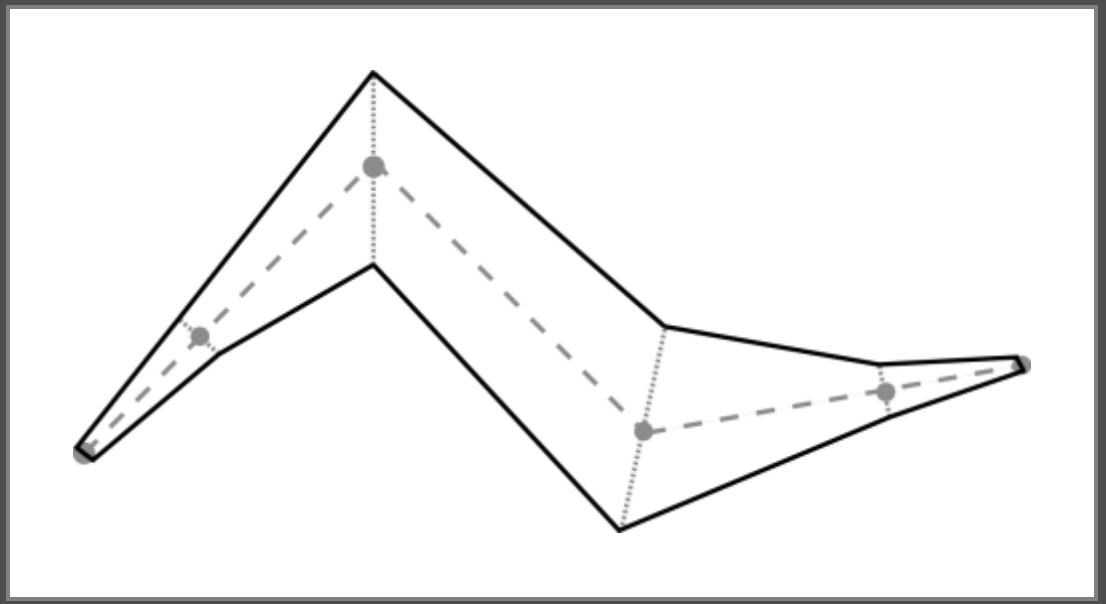
- direct-touch interaction for stroke creation



# Interaction with SBR via Hand Postures

[Haeberli 1989]

- initial line parameterization though postures
- line control through drawing speed



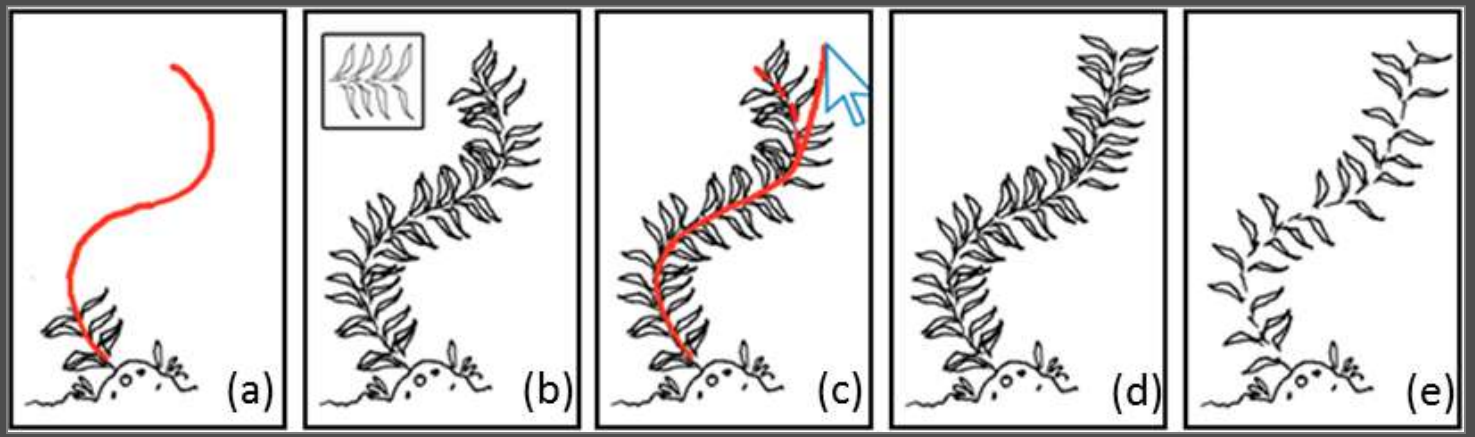
# Interaction with SBR via Hand Postures

with J. Grubert & S. Carpendale

# Interaction with SBR via Hand Postures

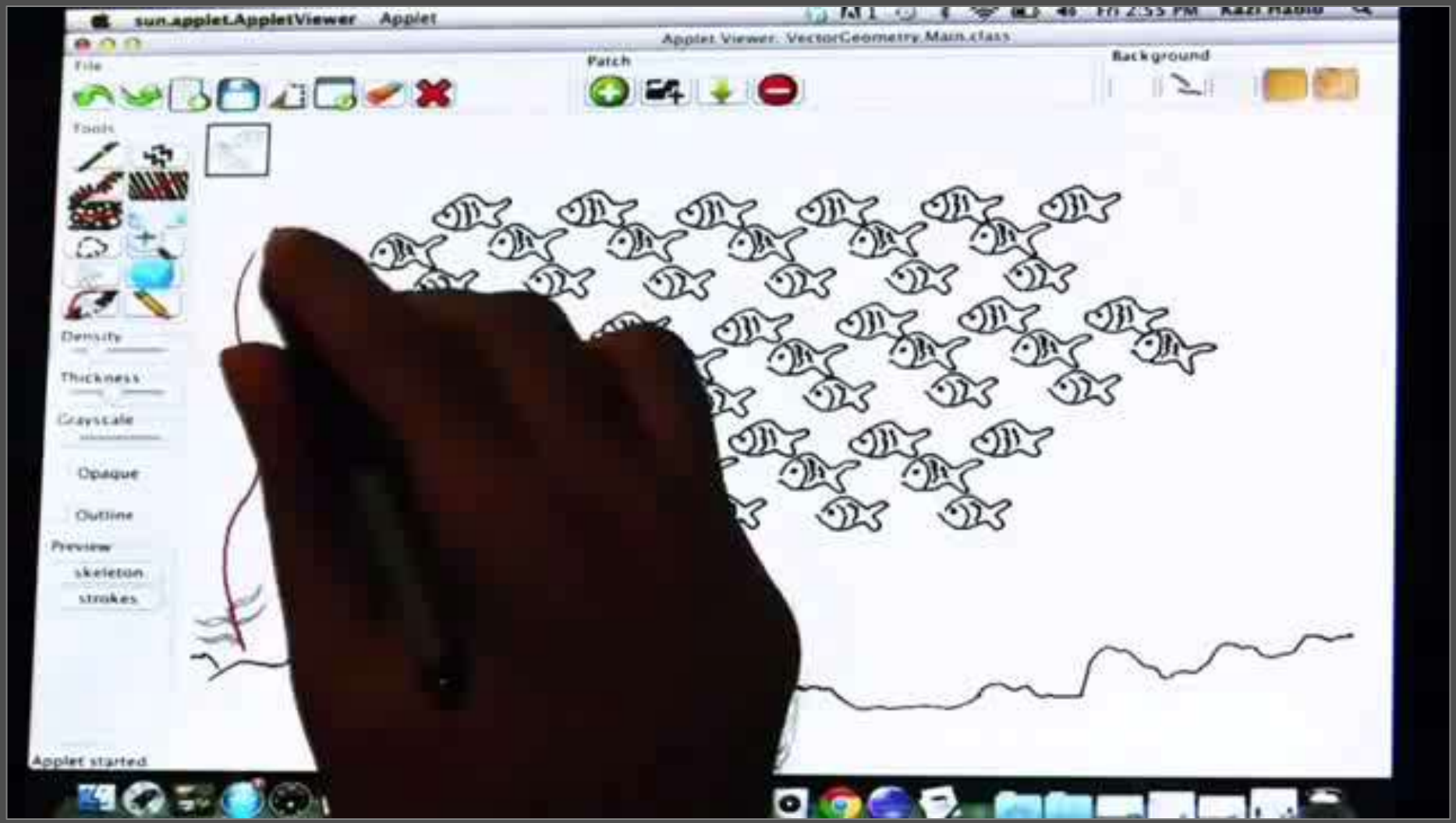
# Automation of Tedious Stroke Placement

- many NPR approaches – simulation of styles that repeat strokes many times, with small changes: stippling, hatching, more complex textures
- placing them is tedious and can be automated if placement depends on measurable properties (e.g., brightness, curvatures as in stippling/hatching)
- sometimes more control of interactive placement needed:




Kazi et al., 2012

# Automation of Tedious Stroke Placement




Kazi et al., 2012

# Interacting with Strokes

- so far: interactive control for largely automatic SBR
- other domain for user interfaces in NPR: interactive control for largely automatic sparse line rendering
- interaction goals:
  - stroke stylization
  - abstraction and emphasis
- interaction challenges:
  - intuitive “style” specification
  - spatially explicit control
  - local changes

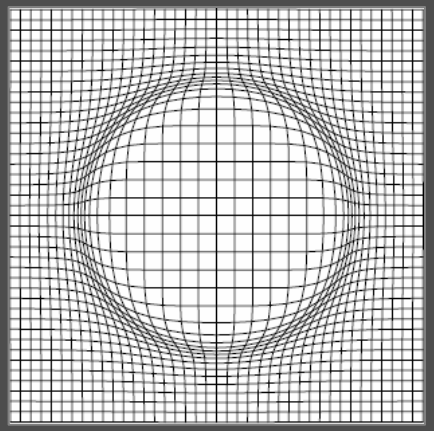
# Interacting with Strokes: WYSIWYG-NPR

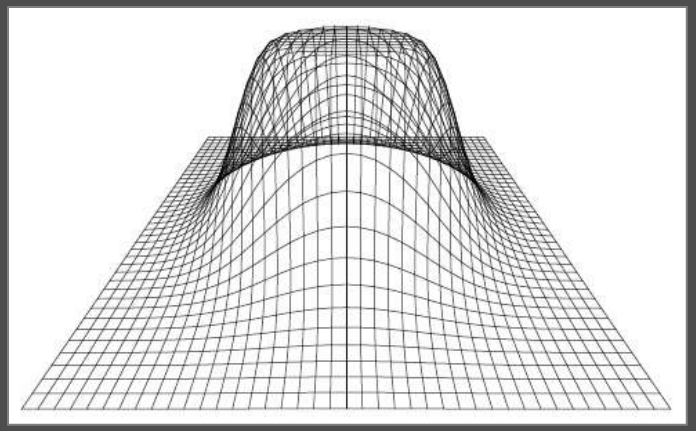
- example-based stylization of sparse line drawings rendered from 3D shapes
- interaction tasks
  - interact with the view on the model
  - draw directly onto a surface
  - specify an example style to be used for one line or to be propagated to all lines (rubber-stamping or example-based synthesis)
  - direct specification of stroke textures on the surface
  - specify local illumination (shadow, highlights)

# Interacting with Strokes: WYSIWYG-NPR

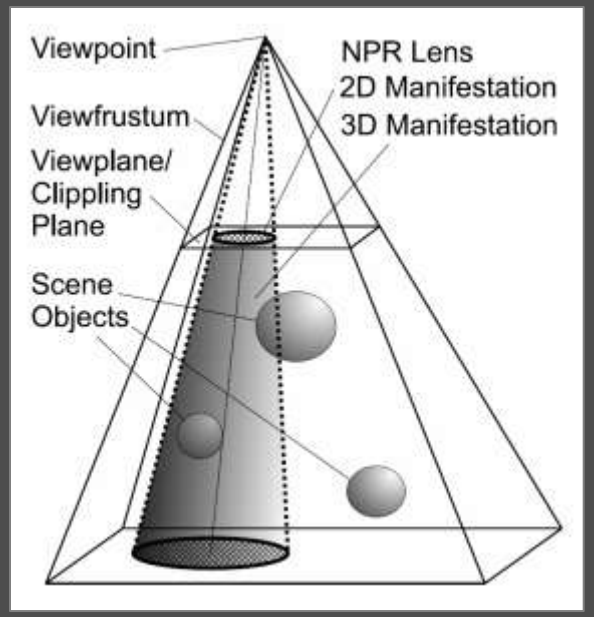
# Interacting with Strokes: NPR Lenses

- problem: global stylization of line properties
- idea: use distortion lenses for two purposes
  - provide the local, spatially explicit control
  - provide the gradual parameter change

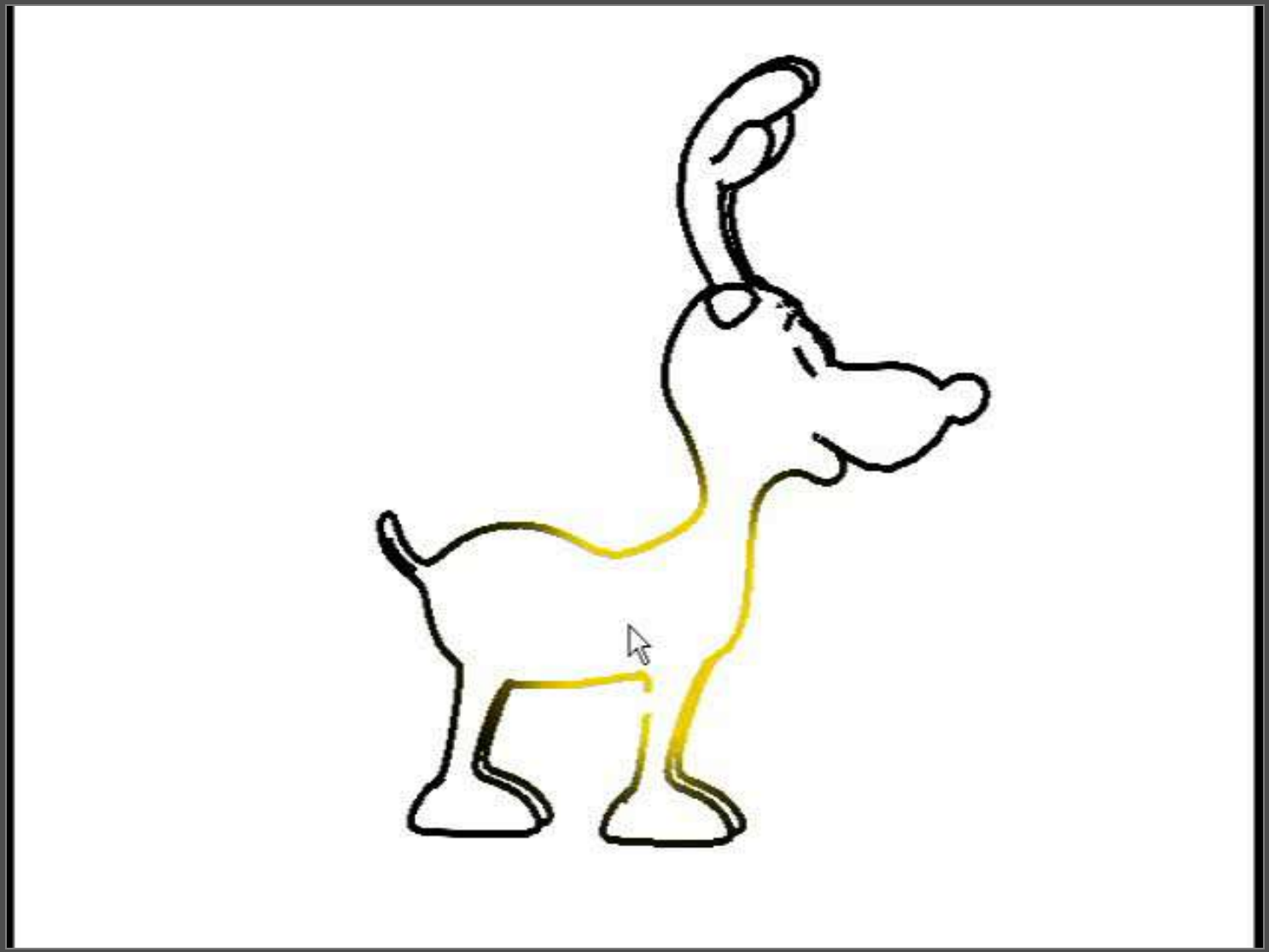




- lens
  - 2D object, lives on the screen
  - has effect in projected 3D space

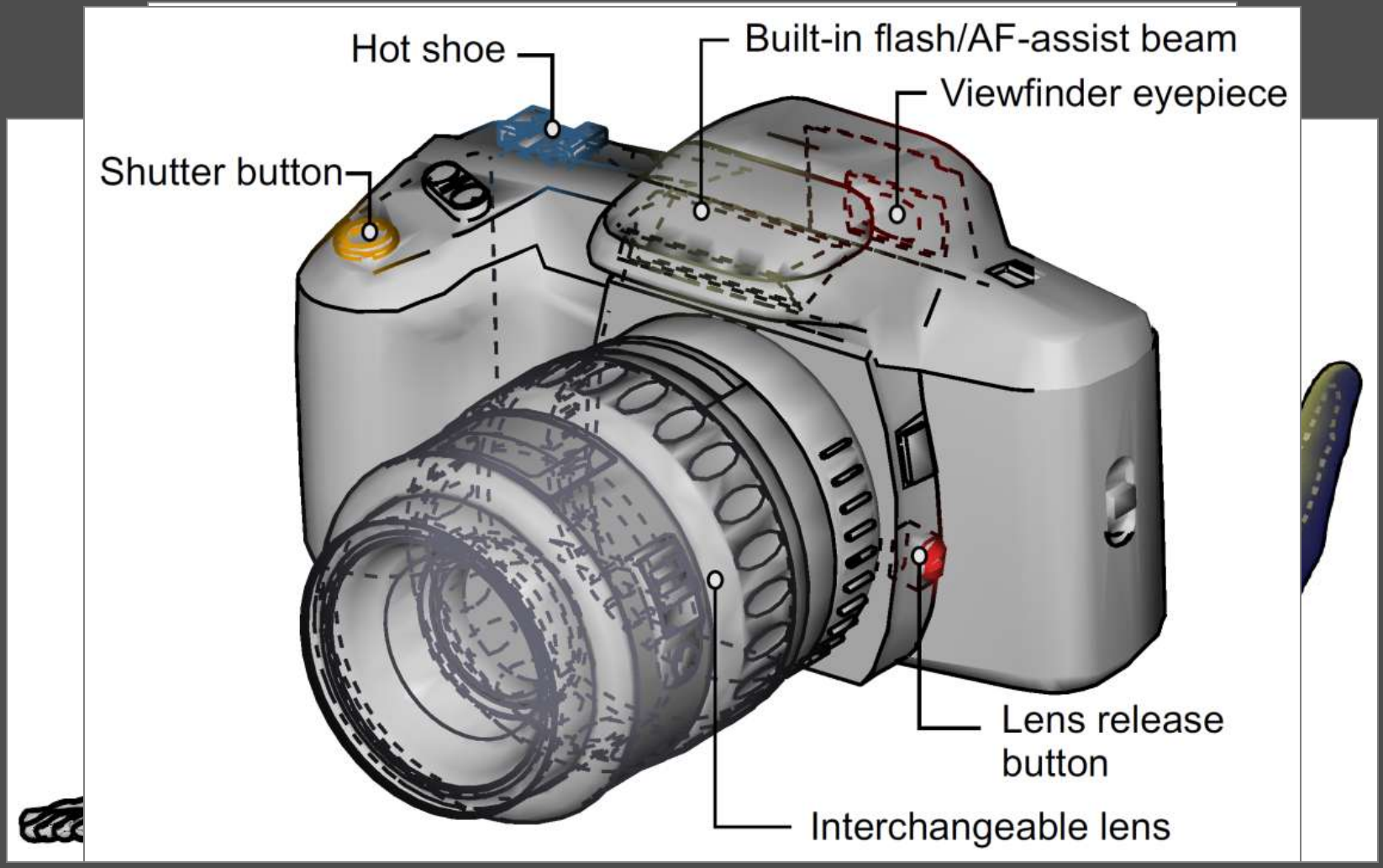


# Interacting with Strokes: NPR Lenses



with P. Neumann & S. Carpendale

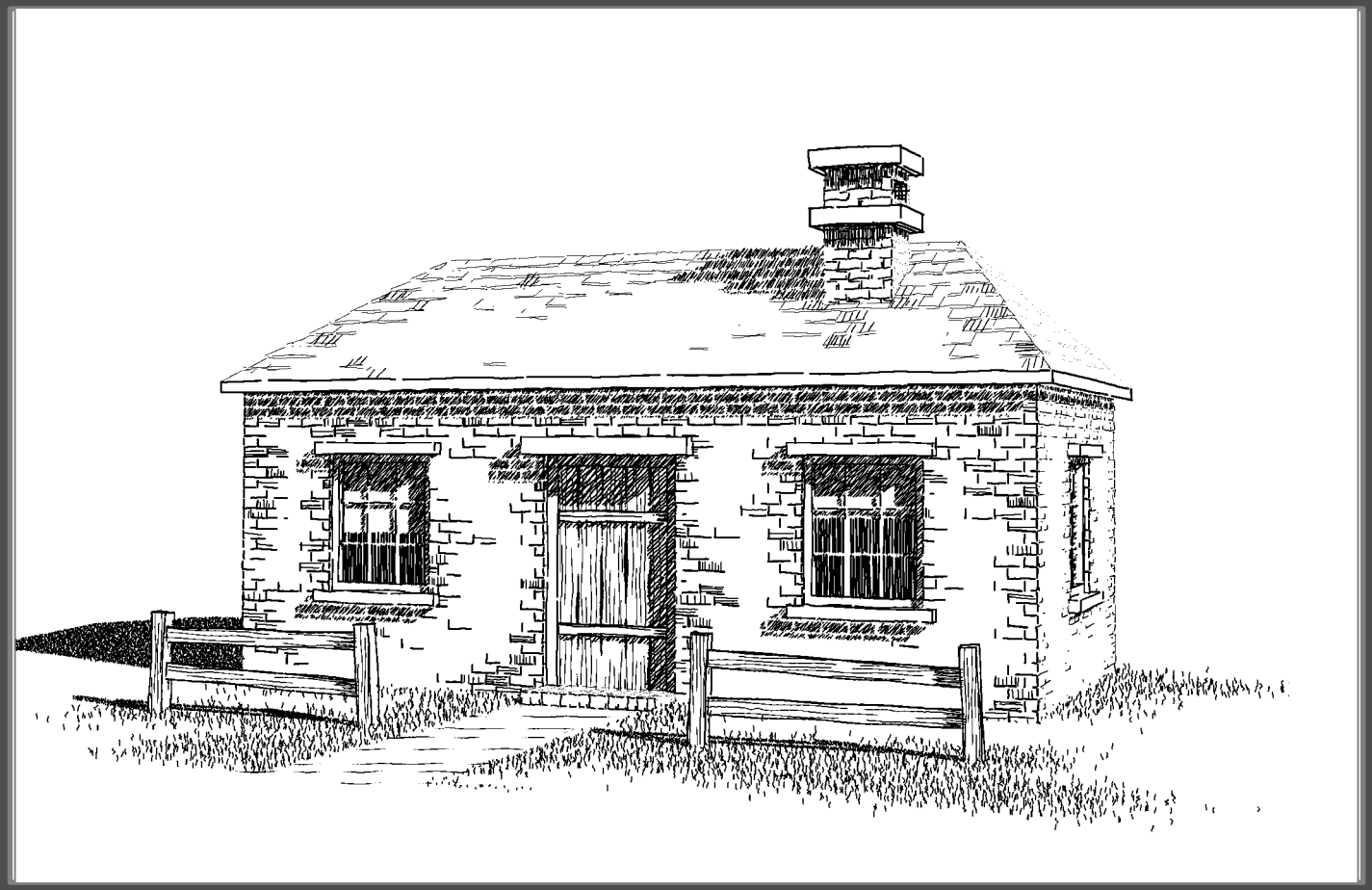
# Interacting with Strokes: NPR Lenses




with P. Neumann & S. Carpendale

# Interacting with Strokes: Abstraction

- interaction for controlling abstraction and emphasis



# Interacting with NPR

Hardware-Driven Interaction

# Interactively Drawing Strokes

- "traditional" platforms
  - mouse (not very intuitive for drawing)
  - tablets (used by many artists)
- novel platforms
  - large displays
  - CAVE VR environments
  - table-based VR environments
  - haptic feedback devices
  - dedicated devices that simulate the traditional experience

# 3D Freeform Sketching in VR

- drawing with hand directly in 3D environment
- interactive assembly of 3D surface model

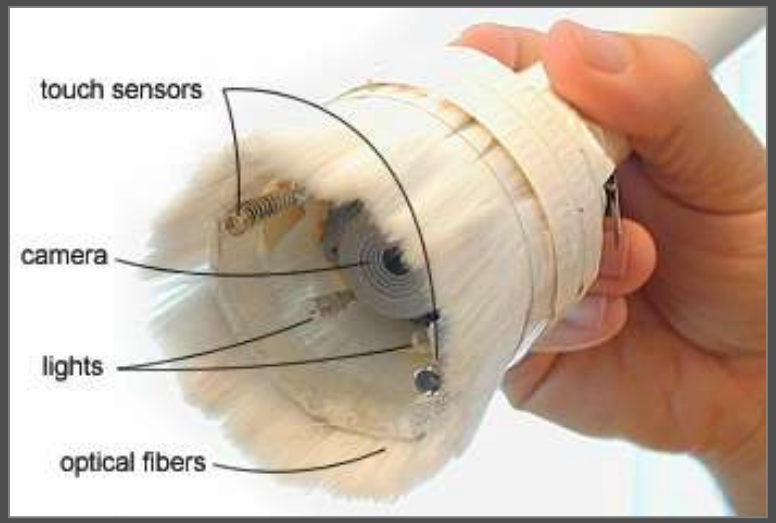
Schkolne et al., 2001

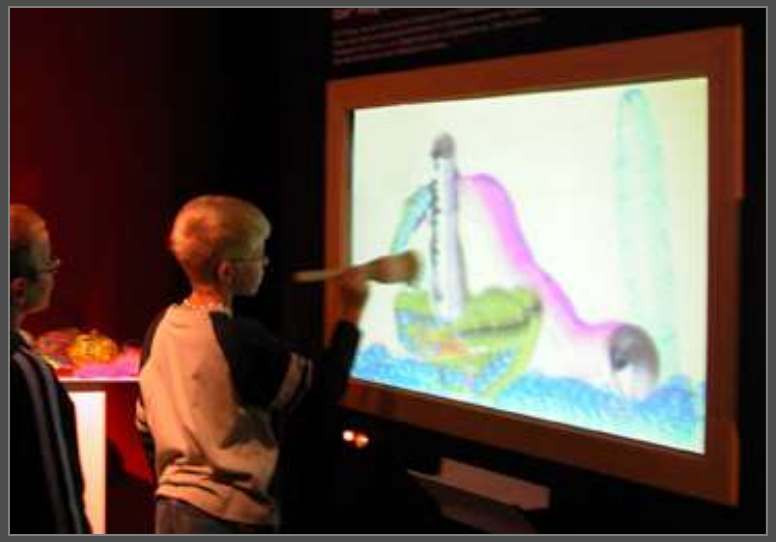
# Cave Painting & Drawing on Air

Keefe et al., 2001 and 2007

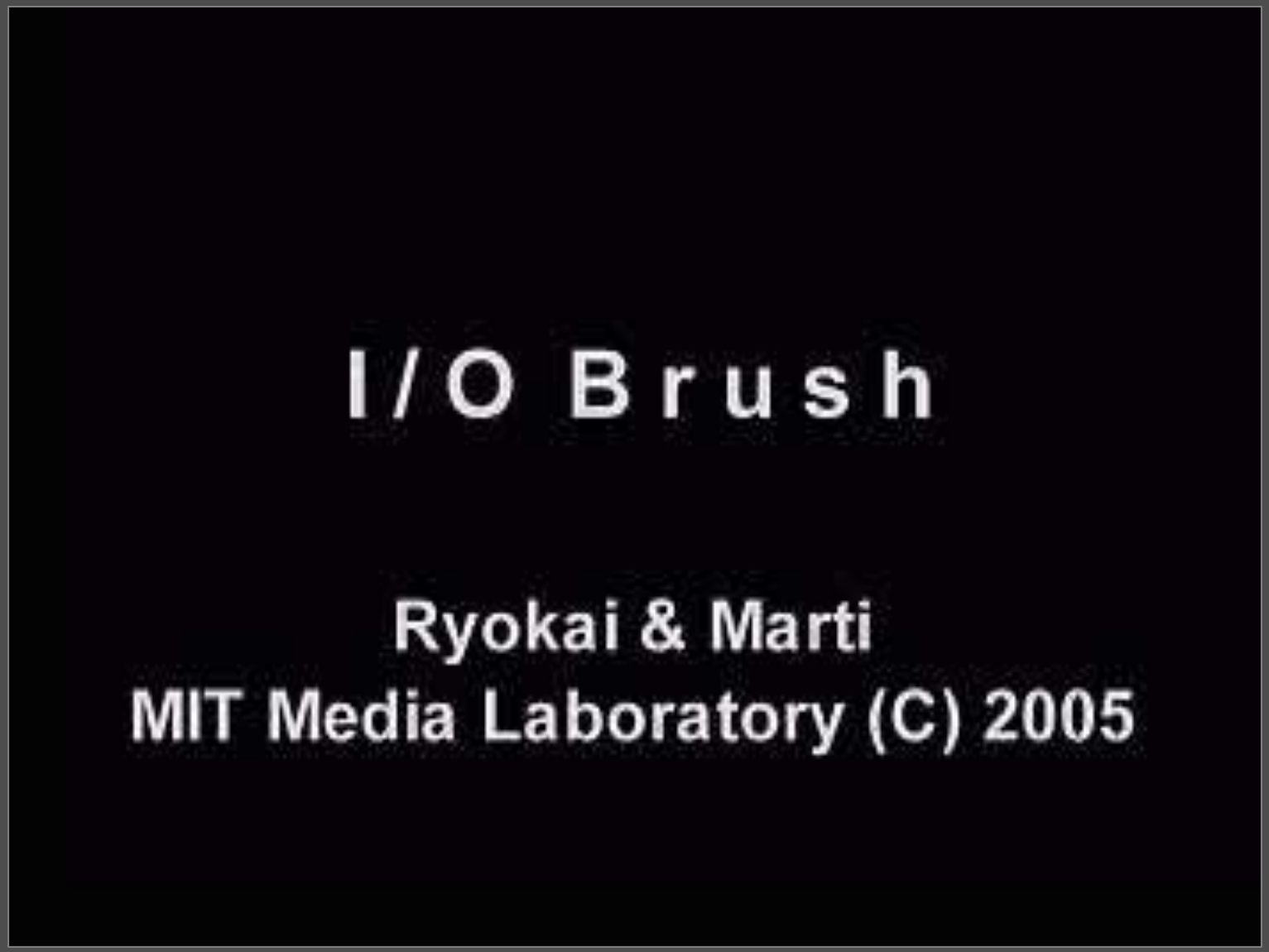
# I/O Brush: Textures from the Real World

- so far: focus on how to interact with strokes
- easy acquiring of textures for brush strokes from the world around us
  - intuitive capturing
  - intuitive use of textures
- used application similar to typical painting environment
- aimed specifically as an interface for children



# I/O Brush: Textures from the Real World



# Simulating a Realistic Environment

- traditional painting:
  - intuitive interface
  - parameters easy to control
- digital painting:
  - more parameters (good and bad)
  - digital advantages (e.g., undo, save, …)
- goal: combine these into one interface
  - similar in interaction as the real canvas
  - using touch-sensitive surfaces
  - using interaction tools similar to real painting

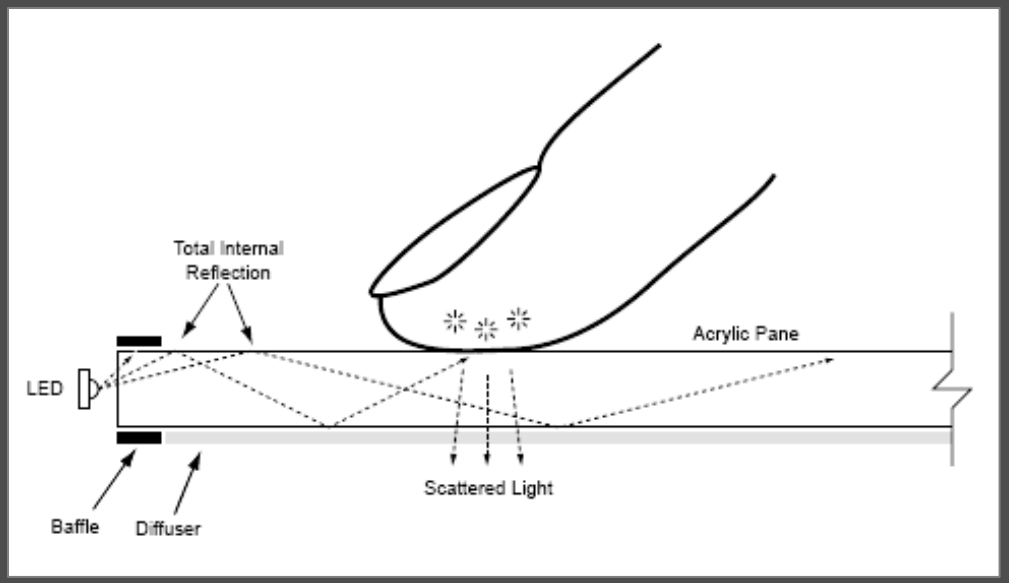
# A Realistic Environment: IntuPaint

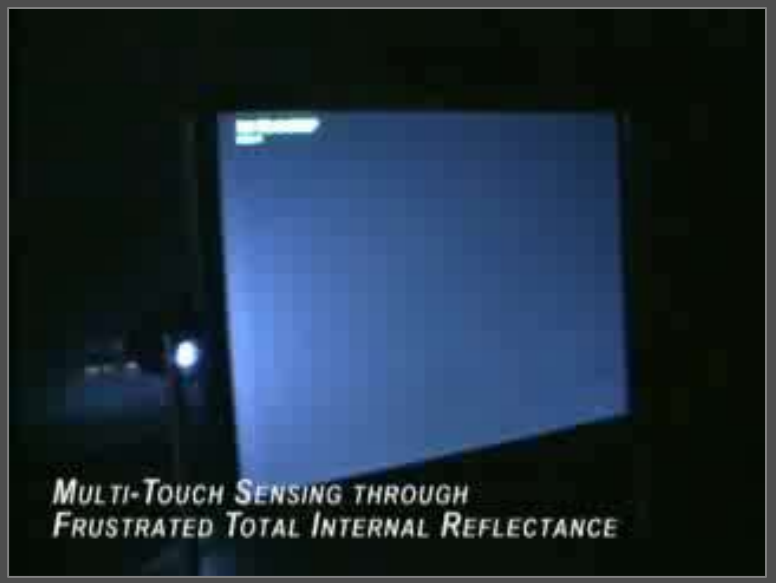
- brushes: fiber-optic bristles IR-based tracking of brush position & touching shape

- slightly titled vertical display
- back-end: simulation of paint
- simulates oil painting with realistic feedback

# Side Note: FTIR to Detect Touch

- Frustrated Total Internal Reflection (Han, 2005)

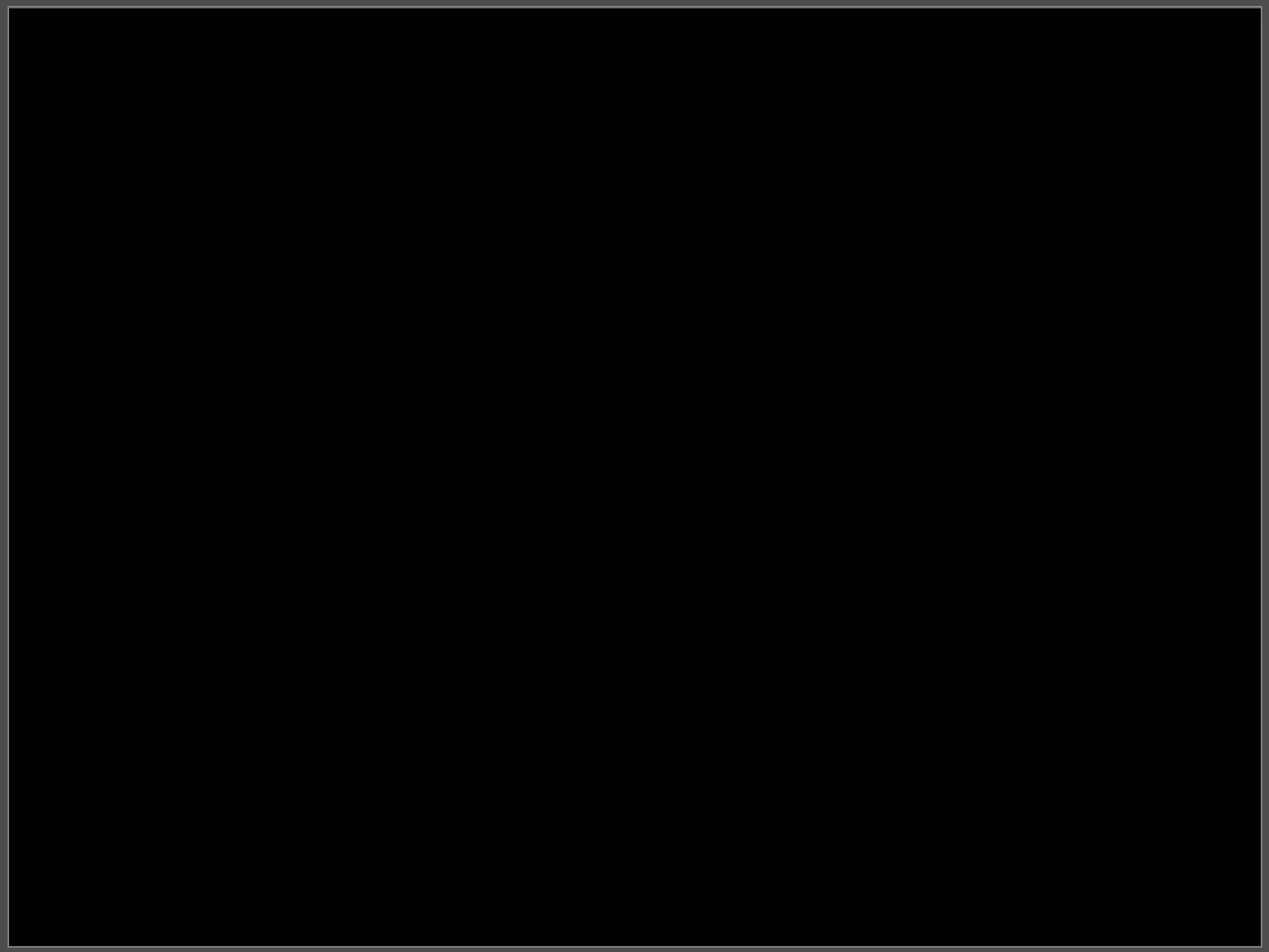



Han, 2005

- domains: tables & (large) walls (e.g., CNN etc.)
- can detect many simultaneous touches
- easy to build

# A Realistic Environment: IntuPaint



# Special Artistic Style: Sand Canvas

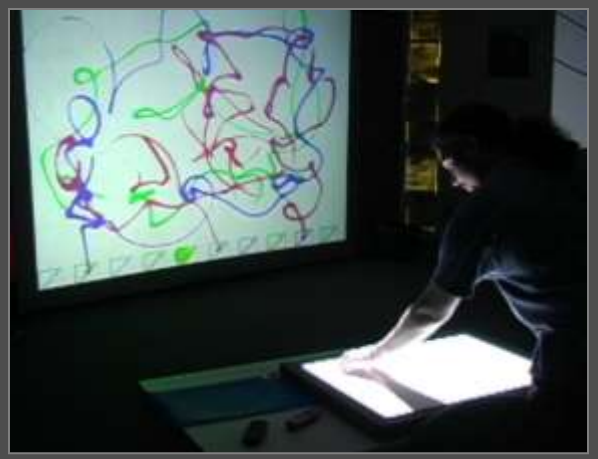
- direct-touch platforms useful for other interactive expressive media: sand canvas/sand painting
  - traditional expressive & highly interactive art
  - simulation of sand being applied to and interacted with on an illuminated and projected surface
  - NPR: combination of the simulation of the sand medium and the interaction with it, using the specific affordances of direct-touch surface interfaces

# Special Artistic Style: Sand Canvas

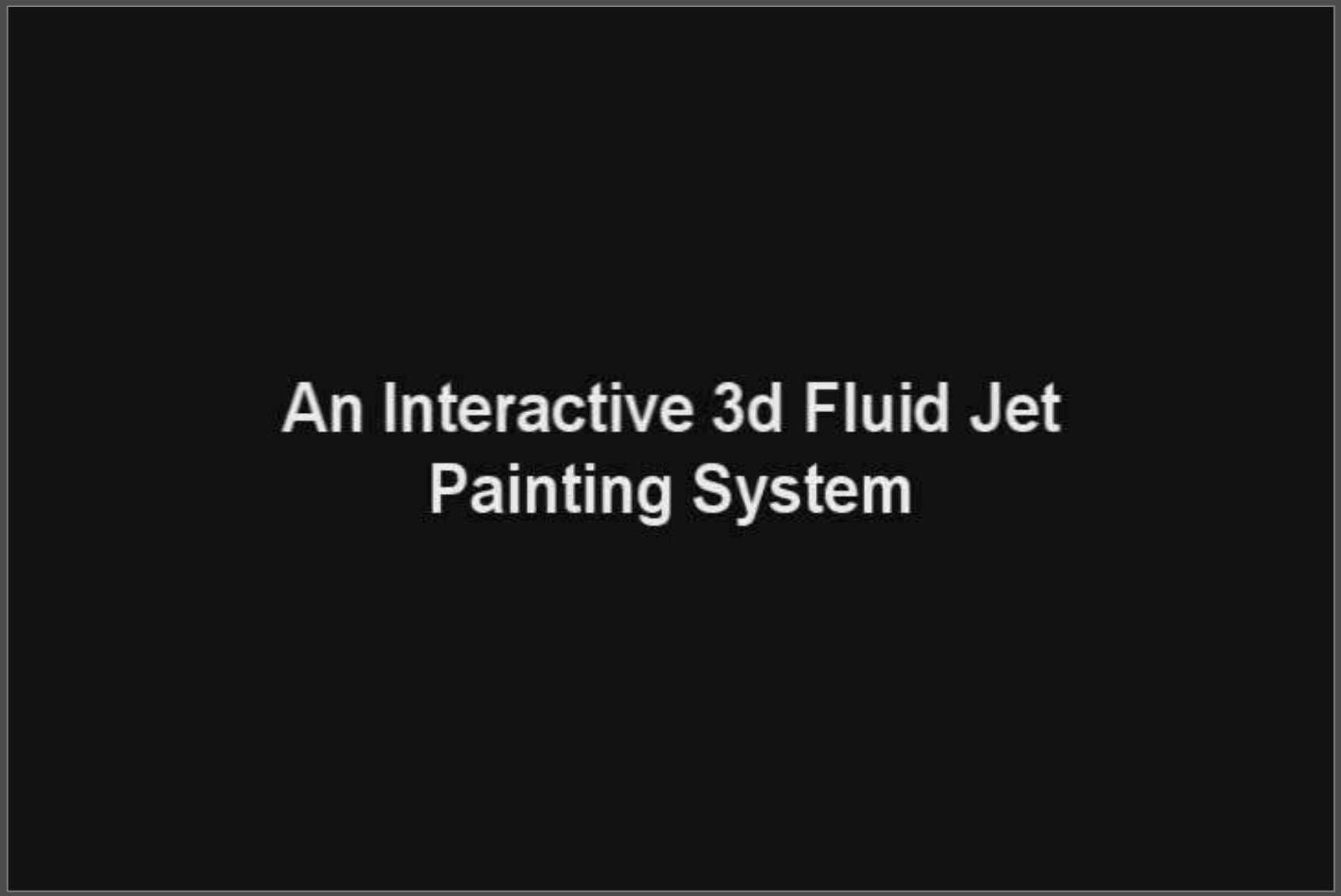
Kazi et al., 2011; http://sandcanvas.com/

# Special Artistic Style: Jackson Pollock

- inspired by Jackson Pollock's abstract art
- combination of fluid simulation and interaction
  - simulation of a fluid flowing/dripping down, including splashing effects
  - touch-sensitive interaction hardware to simulate a more realistic environment



# Special Artistic Style: Jackson Pollock

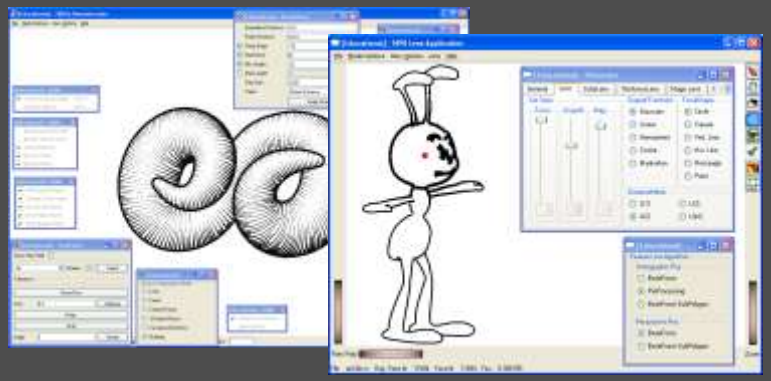


# Interacting with NPR

Triggering Algorithmic Response

# Triggering Algorithmic Response

- interaction with NPR techniques and algorithms
  - direct control of parameters through traditional interface elements (check boxes, sliders, values, buttons, ...)
  - use of alternative interaction platforms, often inspired by traditional artistic media (touch-displays, brushes, cloth, dripping paint, ...)



- other techniques: use automatic NPR method and indirect, higher-level control over the outcome

# Painterly Rendering for Video

- indirect control of a painterly rendered video stream
- interaction: control of the source video stream
- automatic painterly rendering
  - look at changes in the video
  - paint over the parts that change
  - optical flow to associate successive frames, moving existing strokes appropriately
  - results in “live painting”
  - off-line processing also possible

# Painterly Rendering for Video

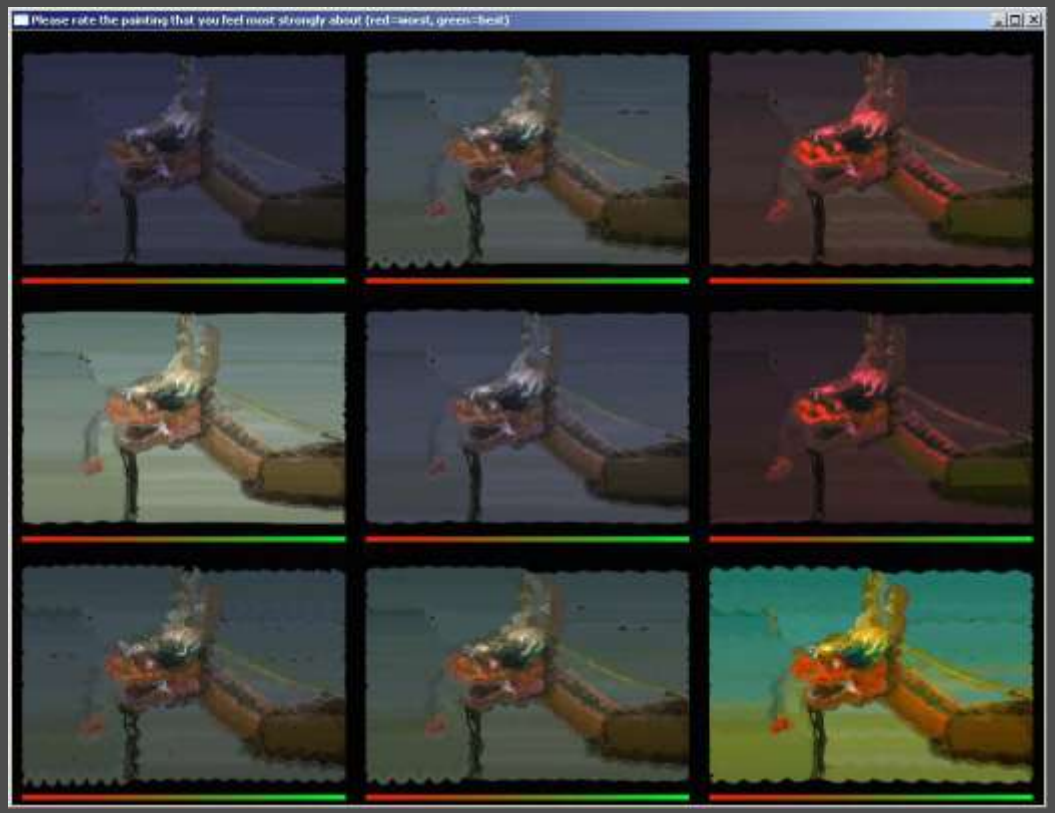
# Triggering Algorithmic Response

- high- vs. low-level control
  - programmers tend to expose numerous parameters in interfaces
  - higher-level control often more interesting for artists
- many artistic pieces could be seen as NPR algorithms
  - e.g., "Wooden Mirror" by Daniel Rozin like animated interactive mosaic rendering, similar to painterly rendering for video
  - some more examples

"Wooden Mirror" by Daniel Rozin

# Human-Guided Genetic Search

- idea: use human opinion to guide genetic algorithm
  - decision what is better or worse sometimes hard to encode in an energy function
  - let a person decide: rate a selection
  - iterative refinement until the desired goal is reached
  - result: goal-centered evolutionary search



# Empathic Painting

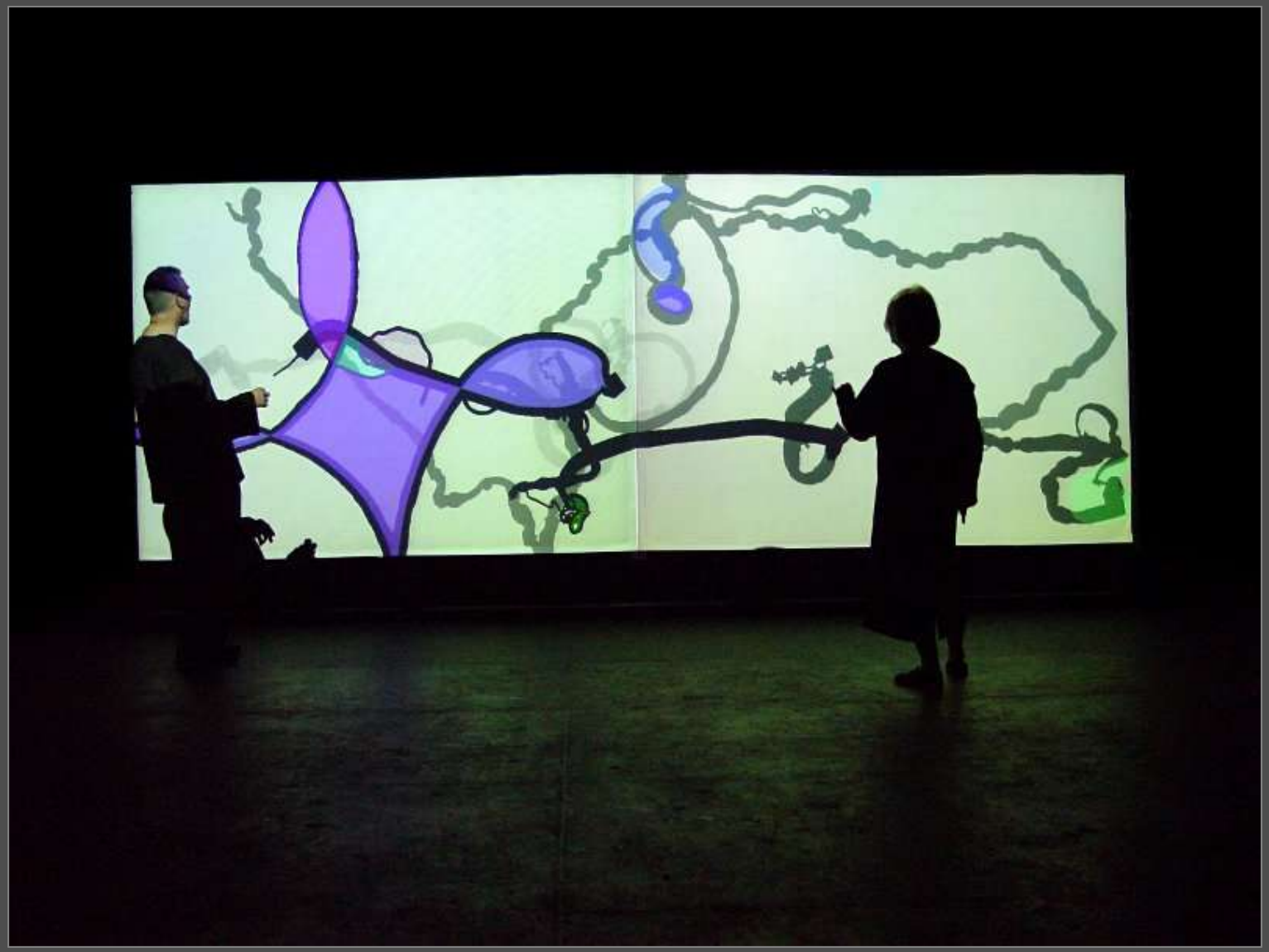
- control of painterly rendering through the viewer's emotional state
  - tracking of facial features
  - mapping to positions in space of prototypical emotional states
  - mapping of mood to parameters of rendering algorithm (color scheme, stroke jaggedness,  etc.)

# Empathic Painting

# Voice & Full Body Interaction

Messa di Voce 2003

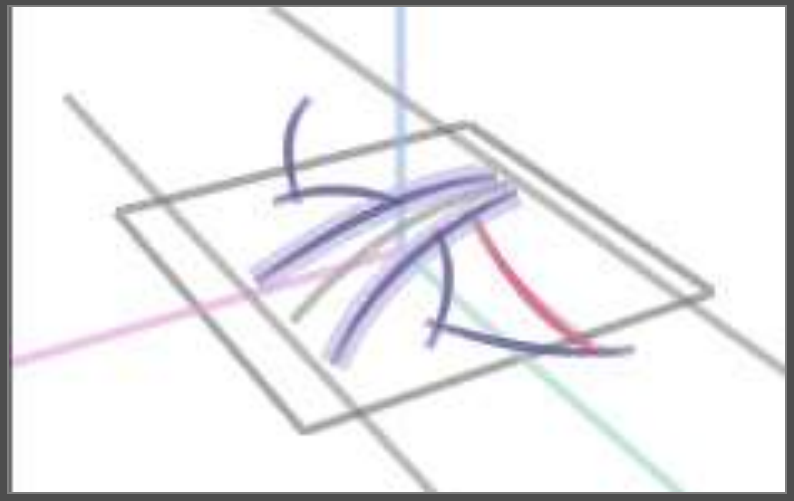
# Voice & Full Body Interaction



Messa di Voce 2003

# Interacting with NPR

Other Related Approaches

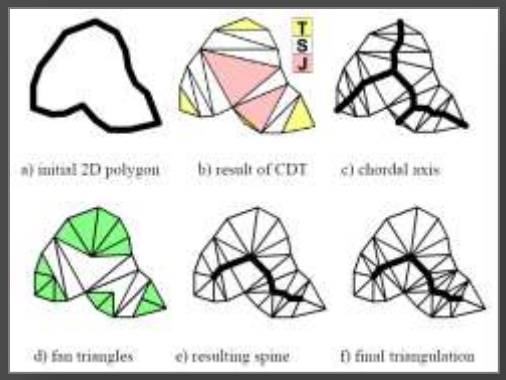


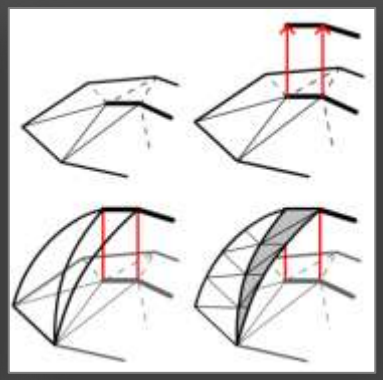
# Approaches Related to Interactive NPR

- sketch-based modeling (SBM)
  - traditional sketching often for initial designs
  - interactive, computer-supported sketching
- NPR to support sketch-based modeling
  - drawing both inspiration for NPR and SBM
  - sketchiness to emphasize the status of initial design
  - inspiration from iterative drawing practices
    visual scaffolding: geometric massing, eraser marks
- stroke-based modeling
  - the stroke as modeling primitive
  - drawing to design in 3D space

# Sketch-Based Modeling: Teddy

- sketching for quick and easy design of freeform models
- technique:
  - draw silhouette & extrude
  - other modeling operations possible: cutting, surface detail, bending, distortion
  - extrusion by inflating shapes



Igarashi et al., 1999

# Sketch-Based Modeling: Teddy

Igarashi et al., 1999

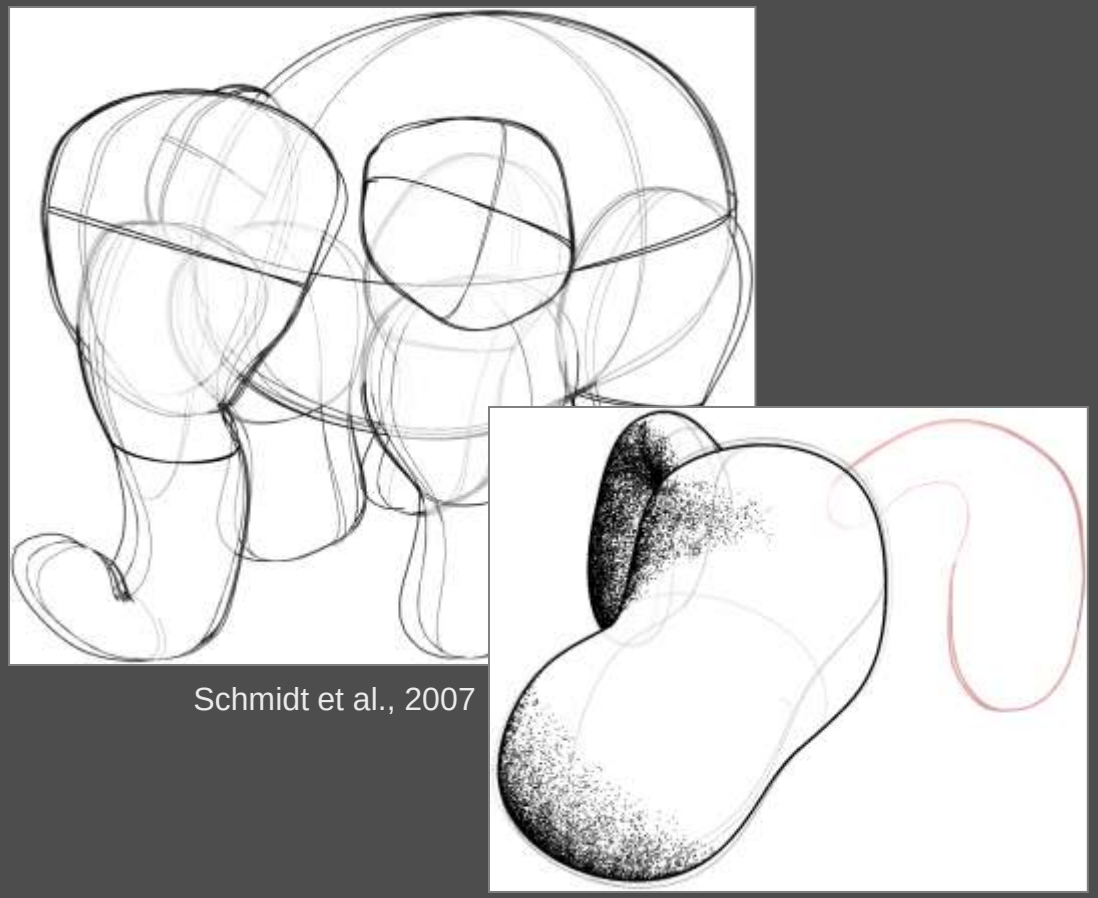
# Sketch-Based Modeling

- problems & issues of sketch-based modeling
  - typically under-constrained problem:
    2D drawing converted into 3D shape
    → use of reasonable defaults
    → use of interactive techniques to change these
    → use of 3D "sketching" input
  - lack of precision due to inherent imprecision
    → use of constrained interaction techniques
  - sketchy character of results (sometimes)
    → use as an advantage
    → use for prototyping applications

# Sketch-Based Modeling

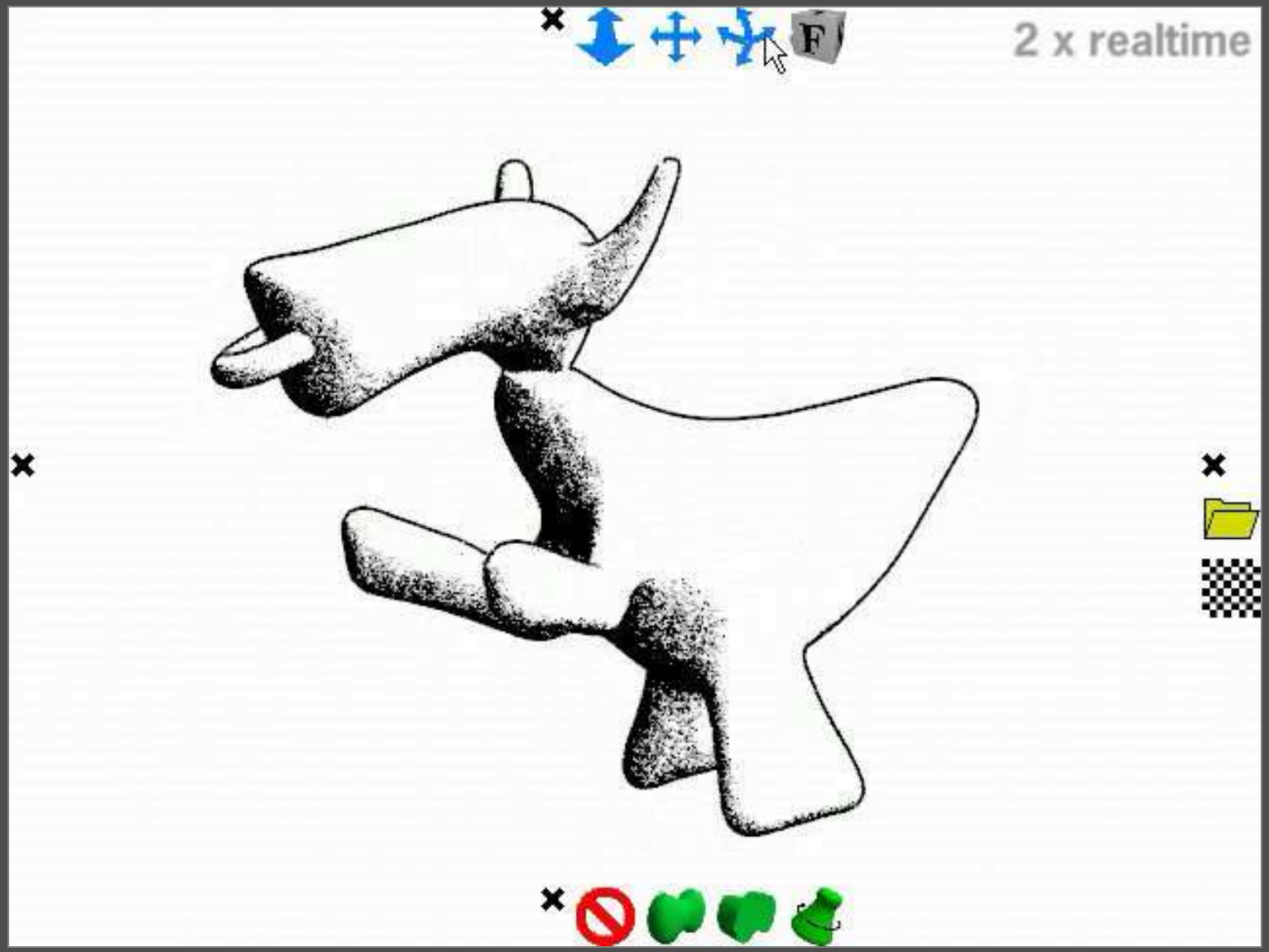
- interactive 3D modeling procedure
- use of elements from real sketches
  - sketching interaction
  - representation (line drawings)
  - visual scaffolding: geometric massing, eraser marks

  → visual history of the modeling process



Schmidt et al., 2007

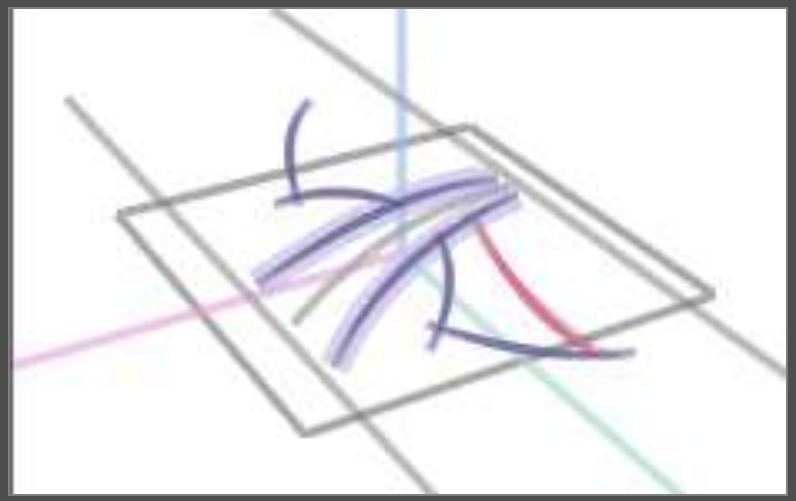
# Sketch-Based Modeling: Example




Schmidt et al., 2007

# Stroke-Based Modeling

- traditional stroke-based modeling
  - sketching as 2D drawing, converted into 3D shape
  - need to use reasonable defaults, interactive changes
- idea: stroke as the drawing primitive
  - individual lines/strokes often in one plane
  - system to interpret input to adapt to drawing and interaction gestures



# Sketch-Based Modeling

Bae et al., 2008

# Interacting with NPR

Summary

# Interacting with NPR

- interaction is not evil or a last resort!
- interaction is necessary for control over the result
  - fully automatic is nice,
    but not satisfying for artists
  - interfaces need to be intuitive,
    need to support creativity
  - should not be used as an excuse

- many forms of interaction possible
  - depend on type of NPR technique
  - depend on type of hardware/platform
  - depend on application and goal