Smart Graphics: Milestones: Interactive SG Systems

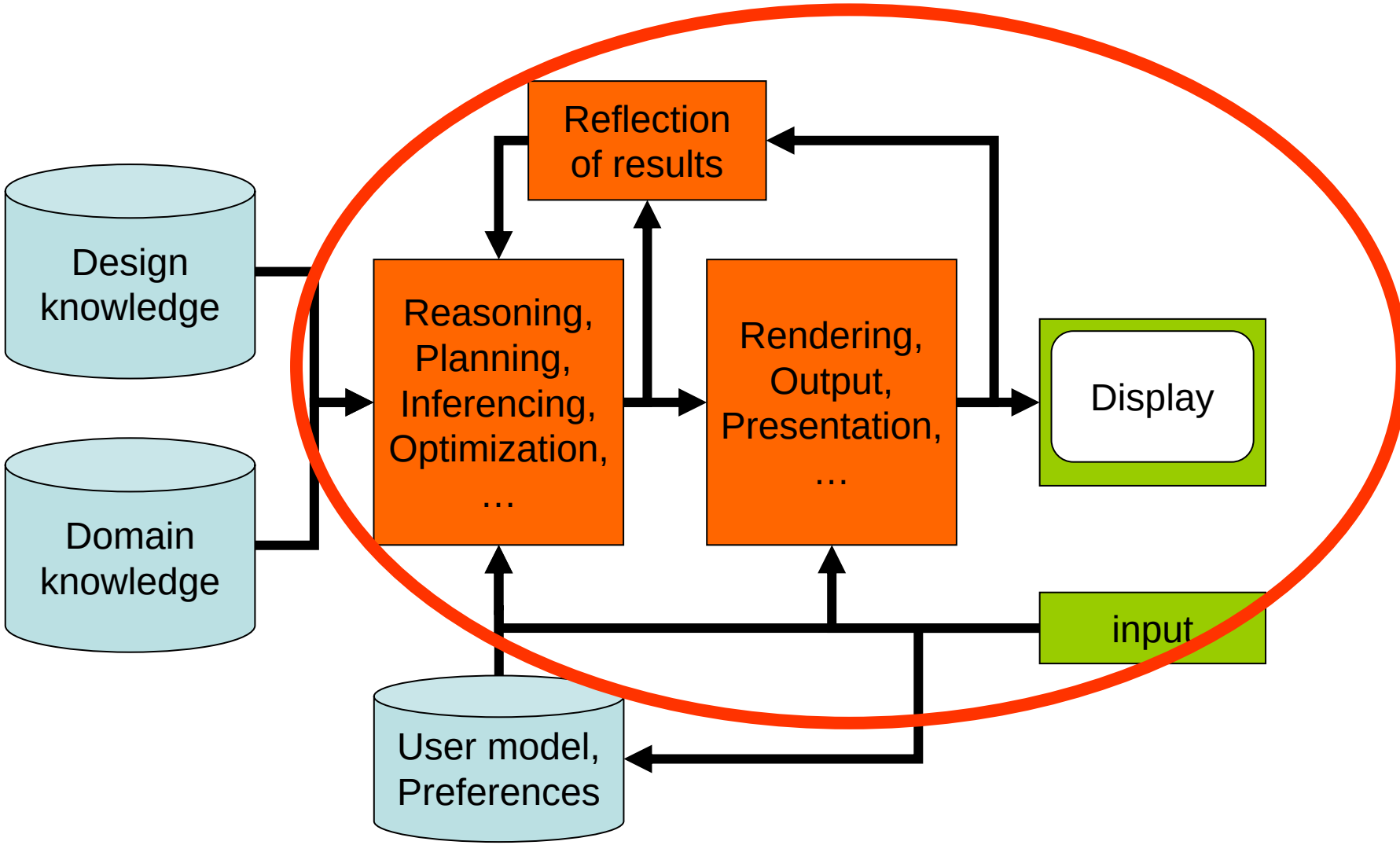
Vorlesung „Smart Graphics”

Andreas Butz, Otmar Hilliges

Mittwoch, 18. Januar 2006

Topics today

- So far only: intelligent generation of graphics
- Today: analysis, matching and processing of graphics and input according to cognitive criteria
- → different meaning of „Smart Graphics“
- Querying
  - Bitmap images
  - Vector drawings
  - 3D models
- Sketching
  - Simple polygonal shapes
  - Organic 3D models



..not so sure this model still works here…

Visual & tangible image query

Kresimir Matkovic et al.

Smart Graphics 2002 & 2004

Motivation

- Increasing amounts of image material exist
- Mechanisms for retrieval are insufficient
  - File name
  - Recorded meta data (Date, camera,…)
  - Manually entered meta data
  - (Image analysis, shapes, ??)
- Needed: visual input for retrieval
  - Find images with similar shapes
  - ..with similar colors
  - ..with similar color layout

Approach

- Go through the image database and compute a descriptor for each image
- Compute the same descriptor for the example image
- Compare with all the image descriptors
  - Standard indexing approach

- → How is the descriptor computed?

Computing image descriptors (1)

- Scale images to 128x128
- Reduce color depth
  - To reduce errors by slightly differing but similar colors
  - Done in CIE HSL (Hue, Saturation, Lightness) space
  - Only 8 basic hues: yellow, orange, red, magenta, violet, blue, cyan, cold green, and warm green
  - Only 5 steps for lightness and saturation

Computing image descriptors (2)

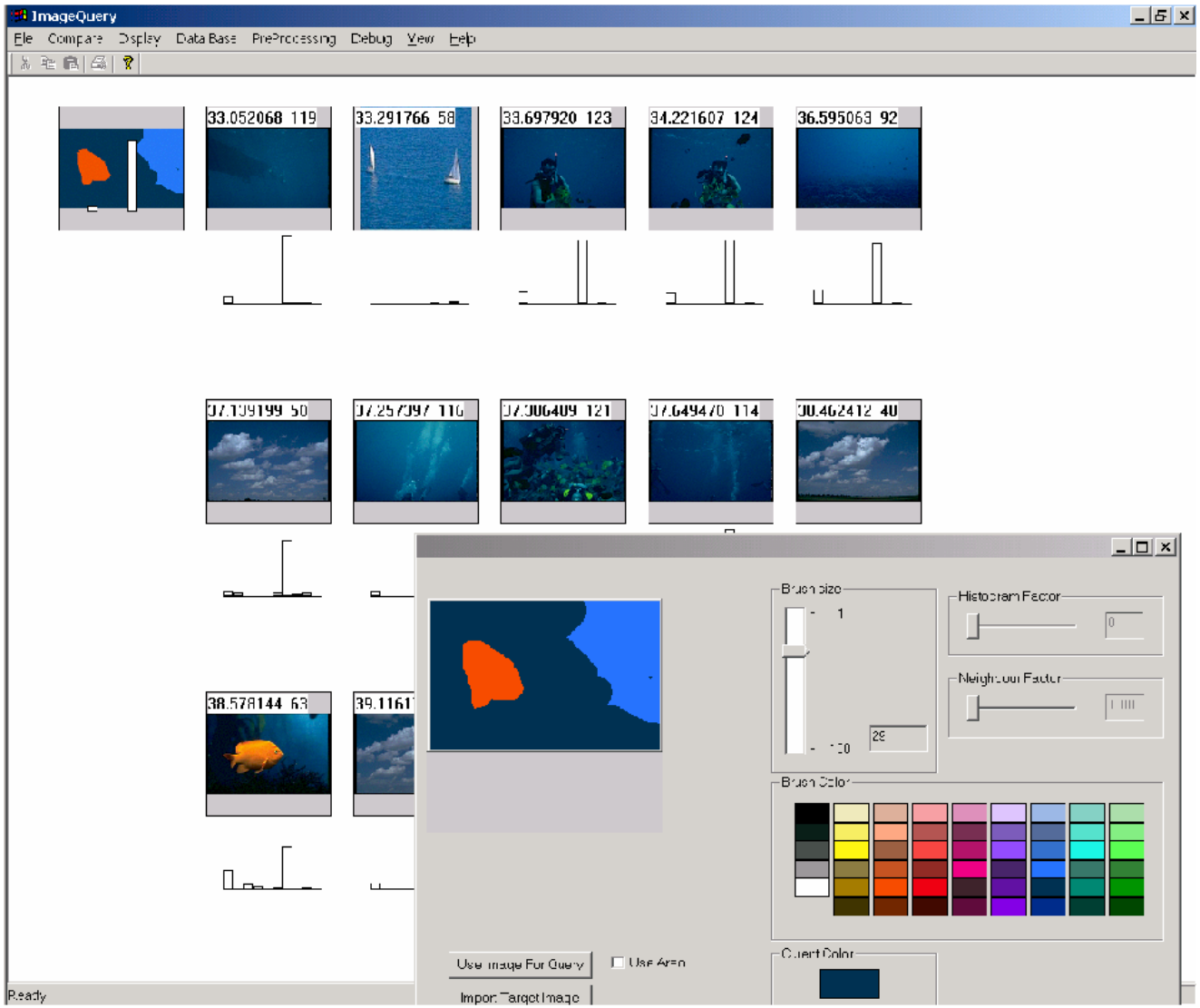
- Place 1.000 rectangles in the image
  - Various sizes and orientations
  - Various positions
  - Distribution semi-randomly
- Compute mean colors for each rectangle → store
  - Using summed area tables
  - Multi-resolution strategy
- Compute a reduced histogram → store

Image retrieval

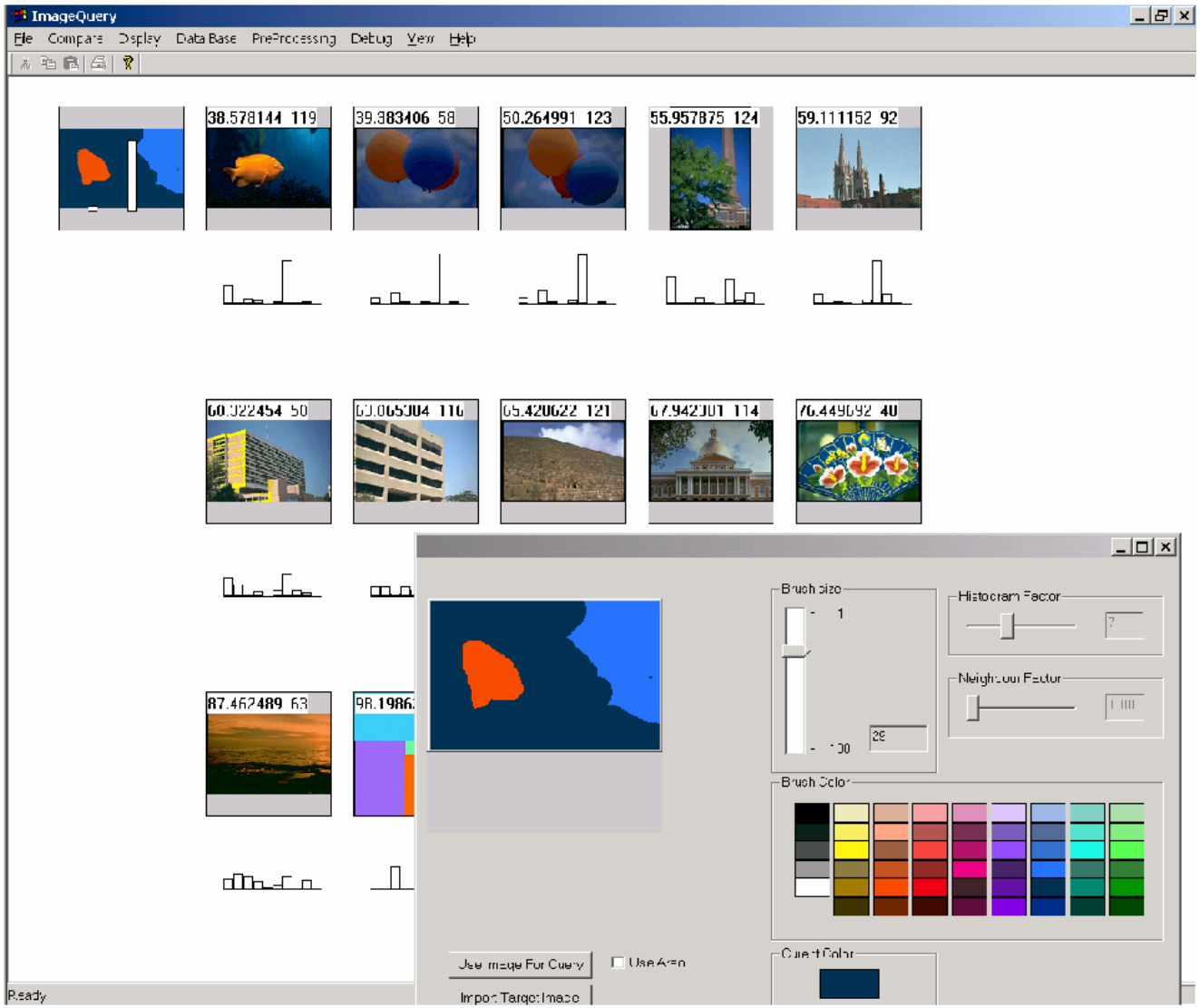
- Compute same descriptor for example image
- Compare in perceptually uniform color space

- Exclude results with incompatible histogram
- Rate, rank and show results

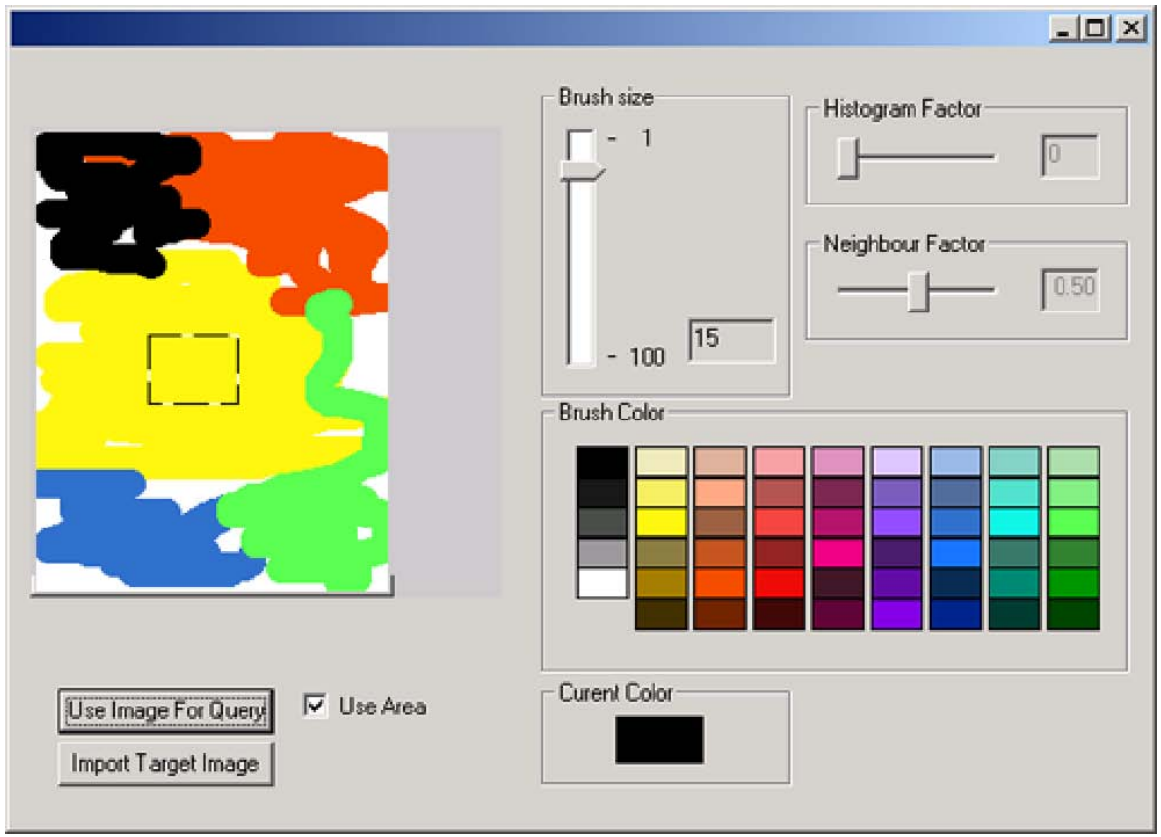
Before Histogram check



After Histogram check

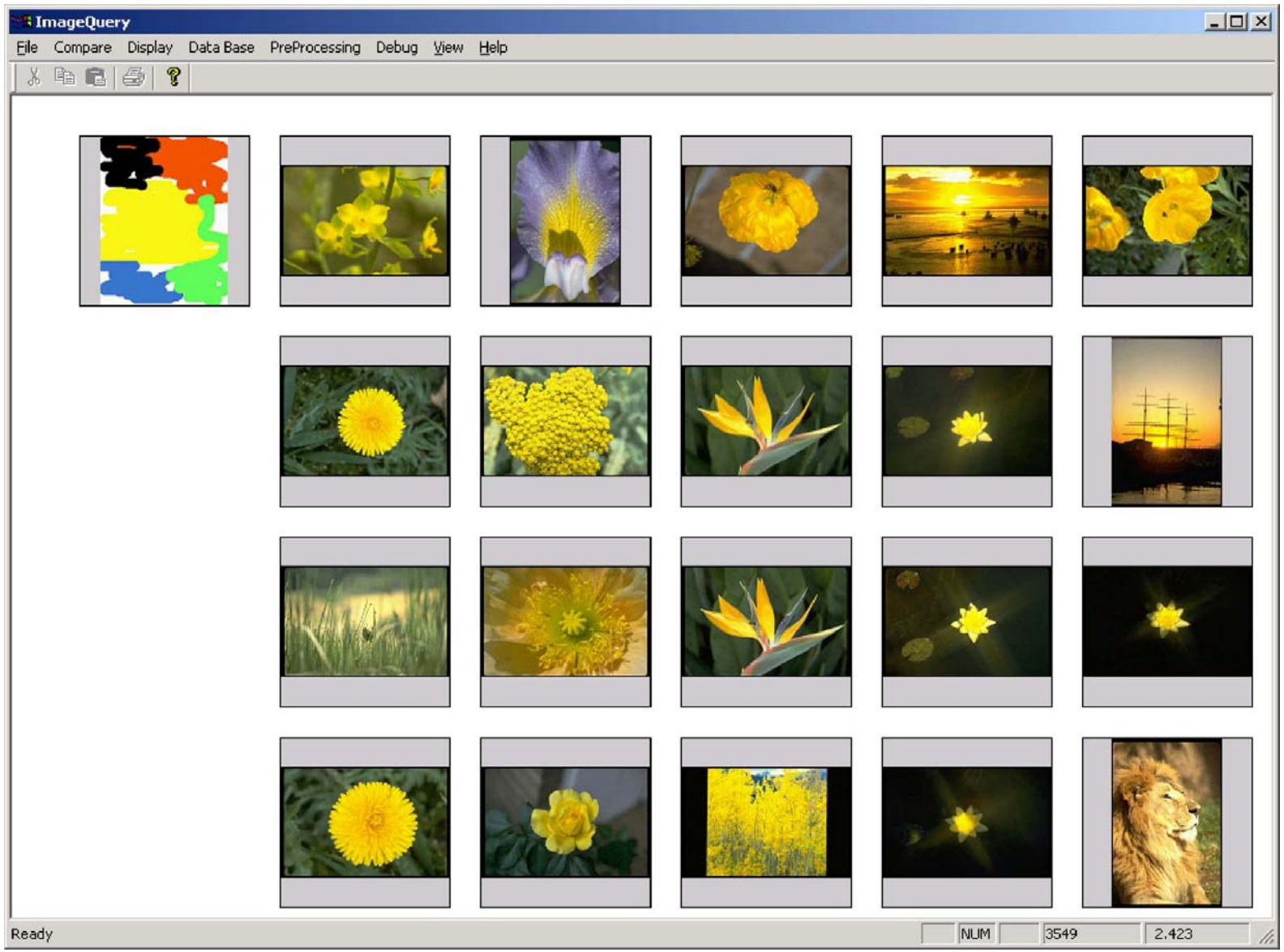


Sketching the example image

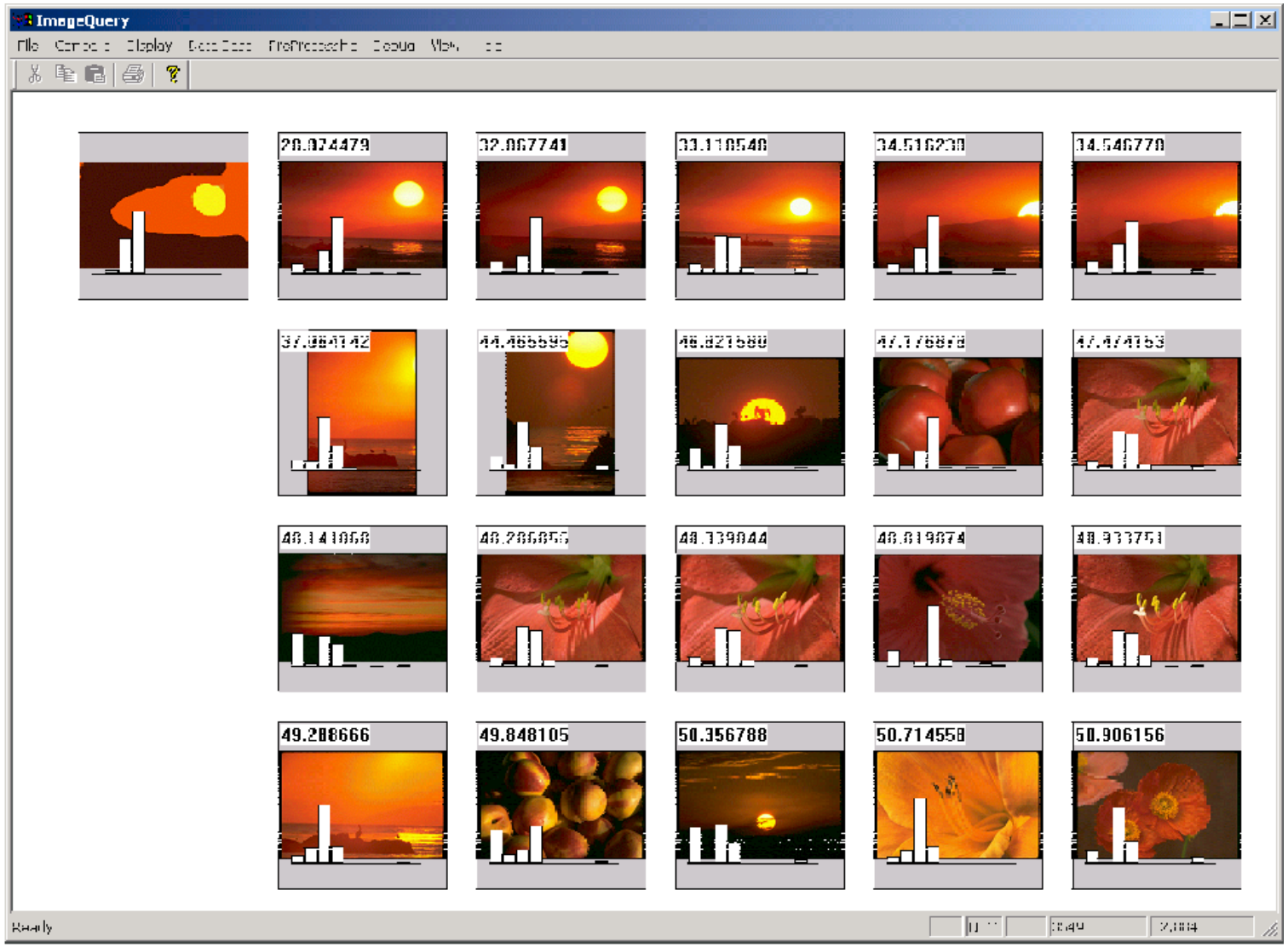


- Very reduced color palette
- Only abstract color distribution important

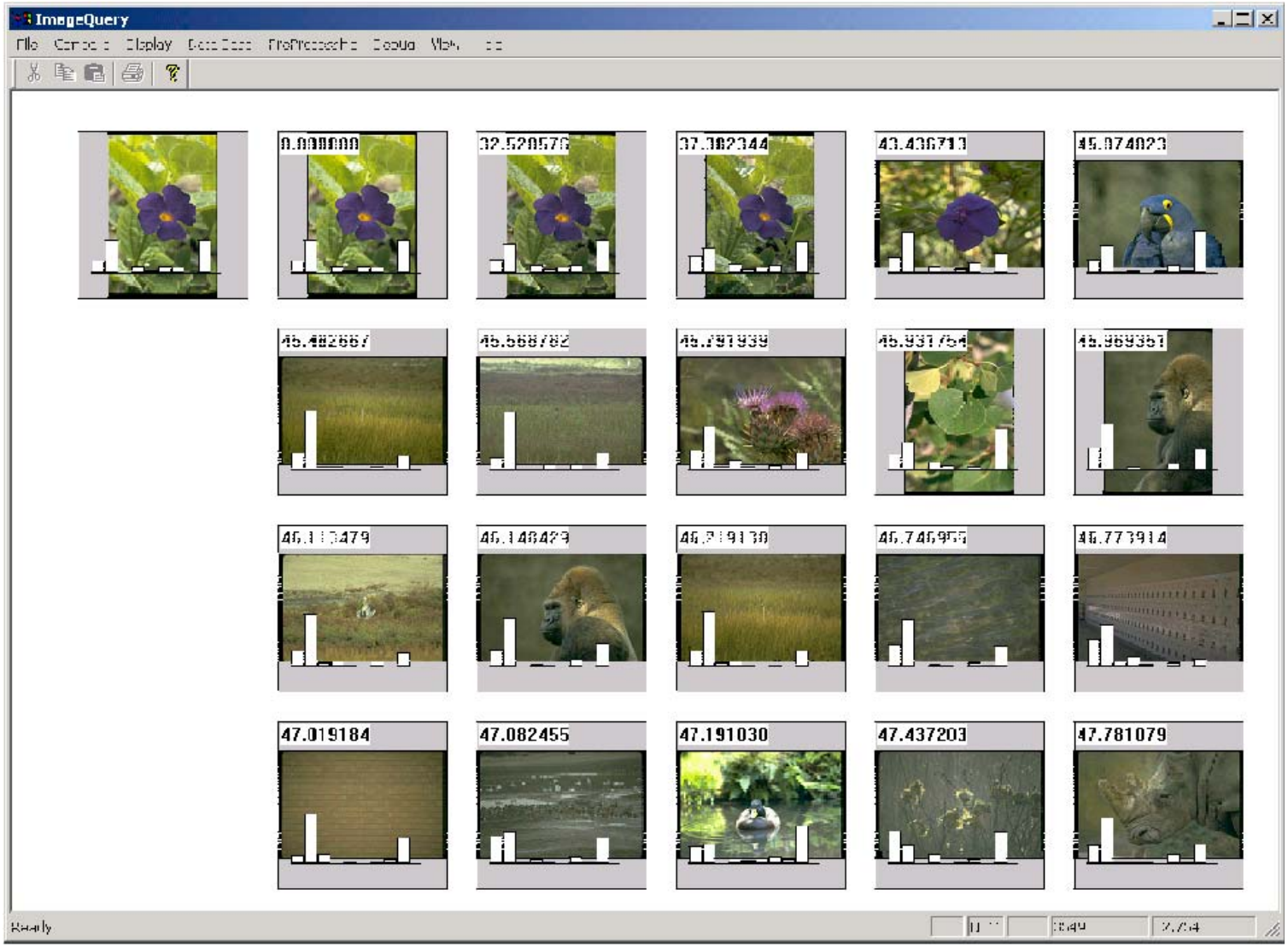
Results (1)


ImageQuery
File Compare Display Data Base PreProcessing Debug View Help
Ready
NUM
3549
2.423

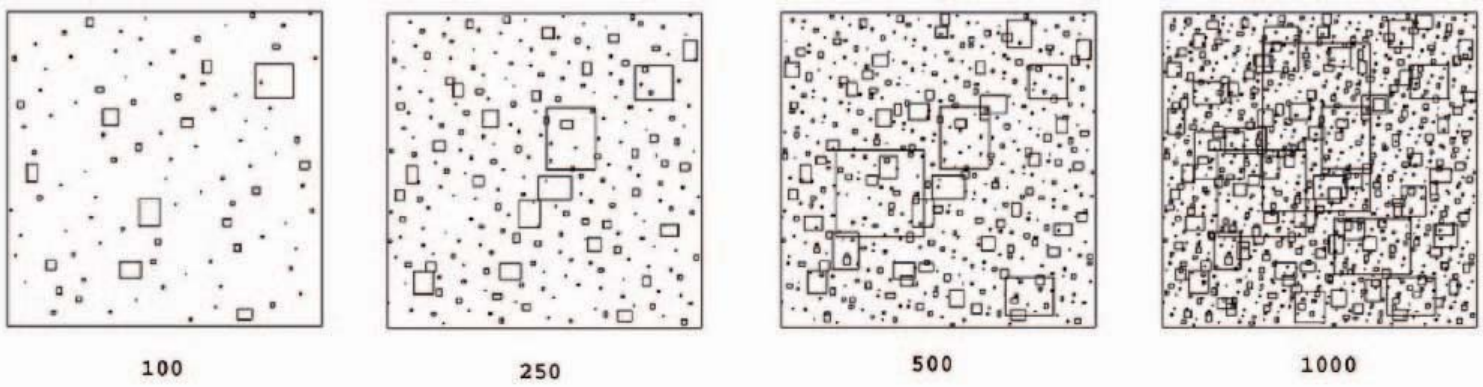
Results (2)


ImageQuery
28.074479
32.867741
33.118548
34.516230
34.546770
49.208666
49.848105
50.356788
50.714558
50.906156

Results (3)


ImageQuery
File
Display
0.000000
32.520576
37.382344
43.436713
45.974023
47.019184
47.082455
47.191030
47.437203
47.781079
Ready

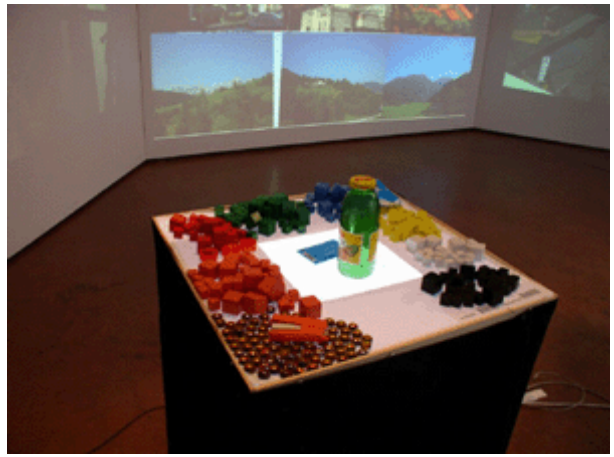
Improvement

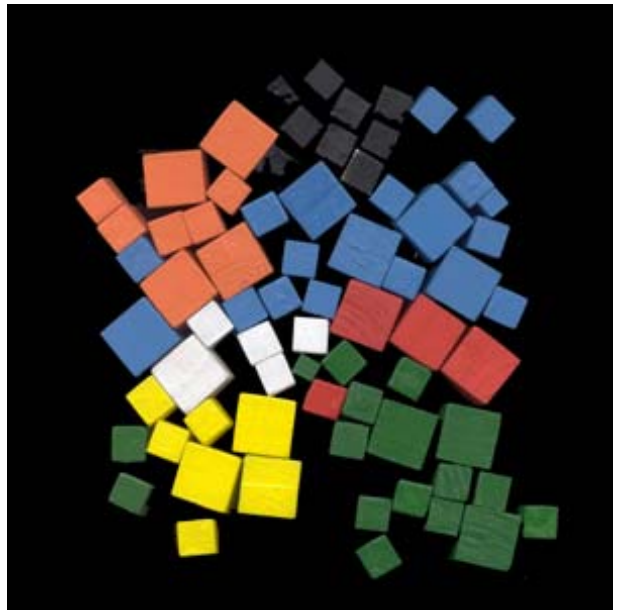


- Use only the rectangles in which the user has actually sketched

Tangible Image Query http://www.vrvis.at/scivis/TIQ/

- Instead of sketching: arrange colored blocks and things
- Camera under a semitrans-parent surface (real time)
- Can record colors without too much external influences

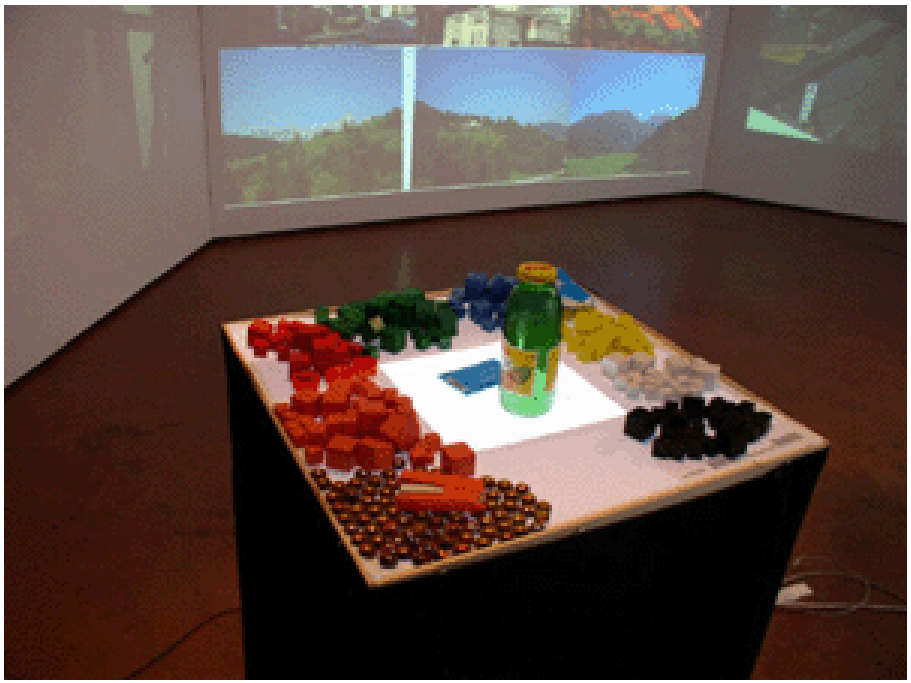


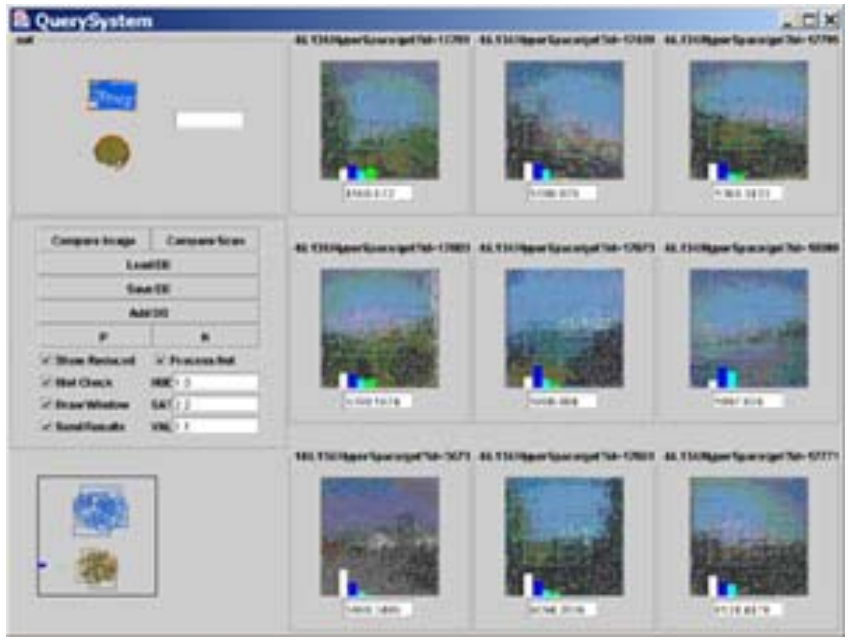


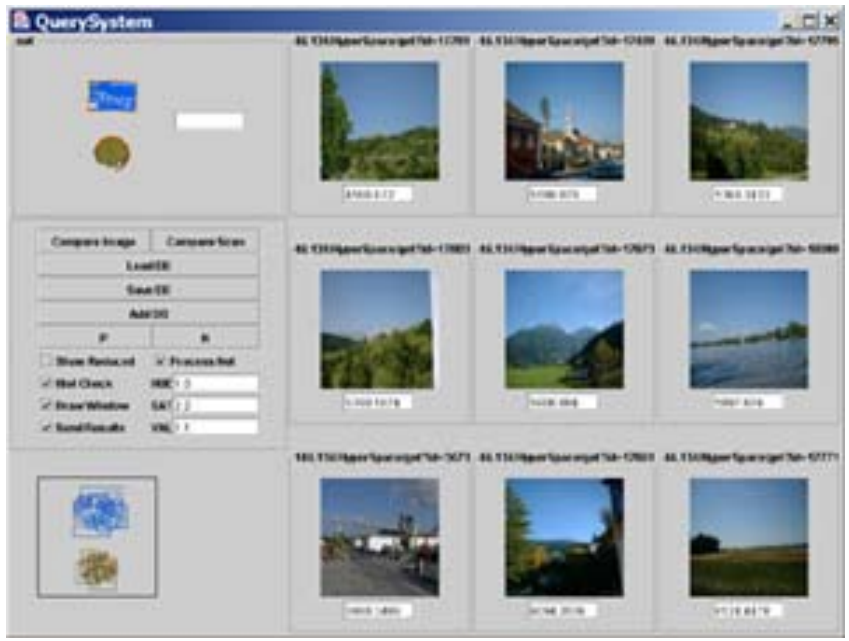
- Alternative: use a scanner above the table (not real time)
- Higher quality but less interactivity

Results

- Comparison with sketching input
- Tangible input faster
- Better acceptance





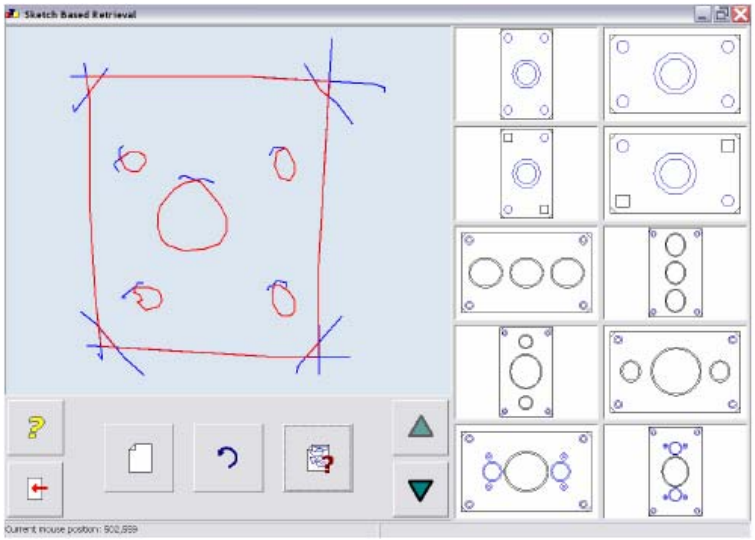


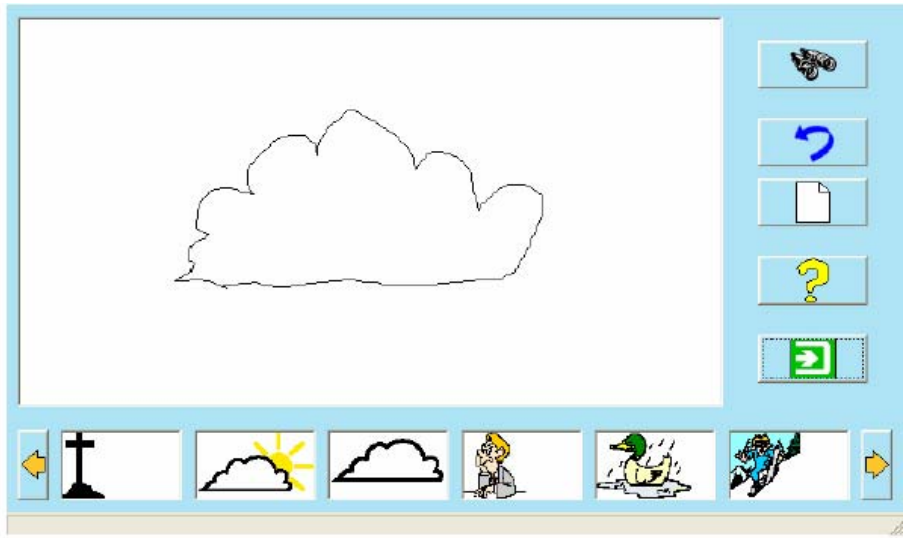
Retrieving vector graphics

Manuel J. Fonseca et al.

Smart Graphics 2004

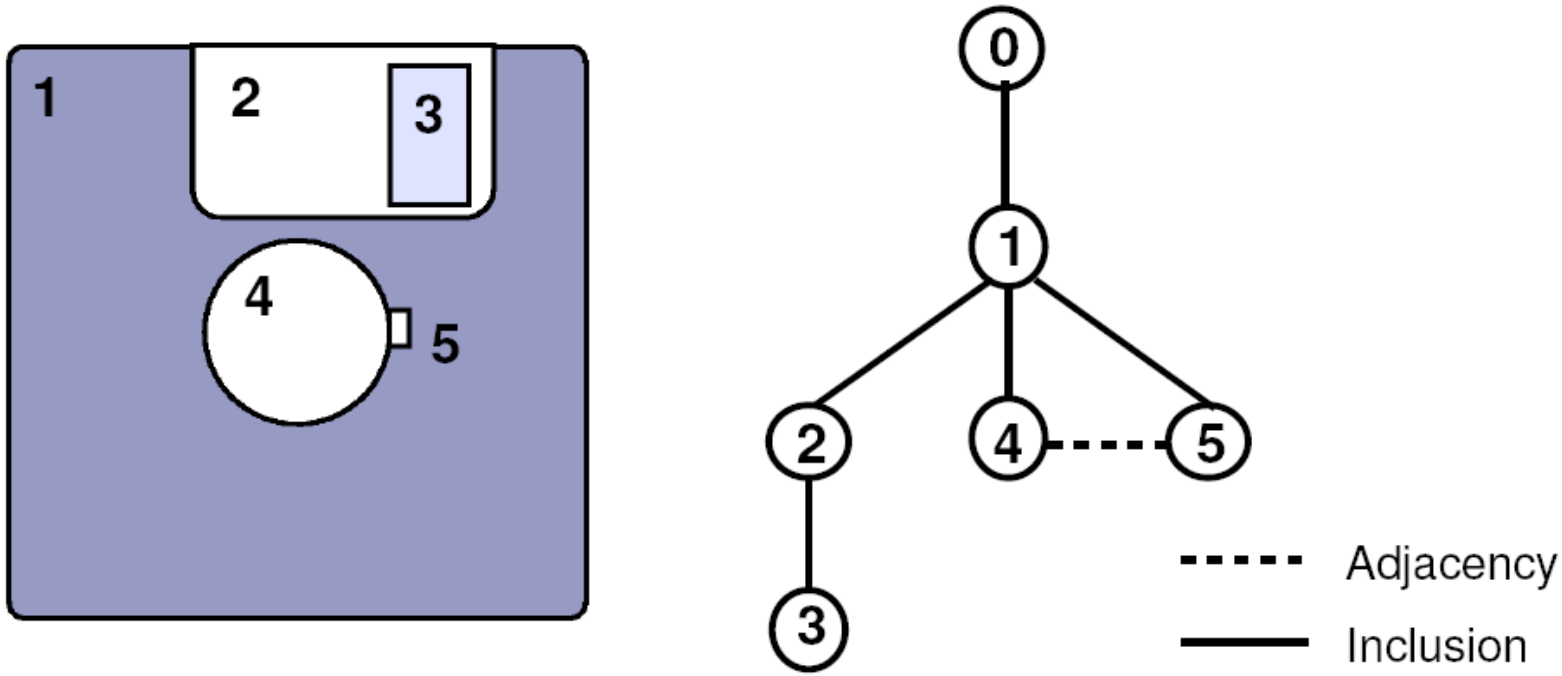
Motivation





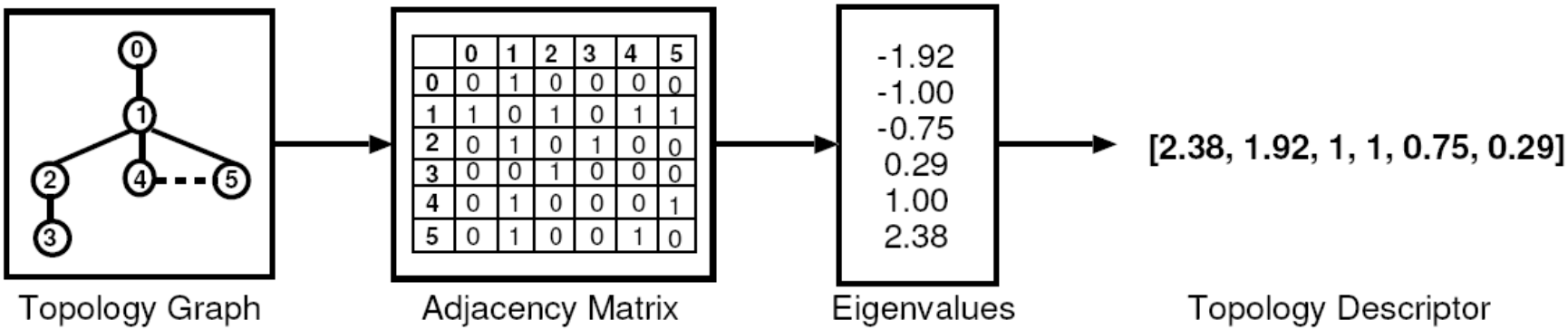
- Browse large collections of vector graphics
  - Technical drawings
  - Clip art
- Find structurally similar drawings

Geometry and topology analysis



- Decompose drawing into its base polygons
  - Discard small details
- Analyze them for adjacency and inclusion
- Store a topology graph of the drawing

Computing Descriptors from graphs

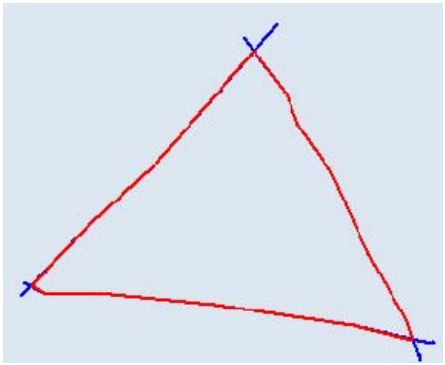
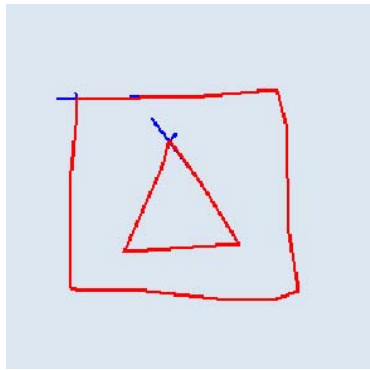
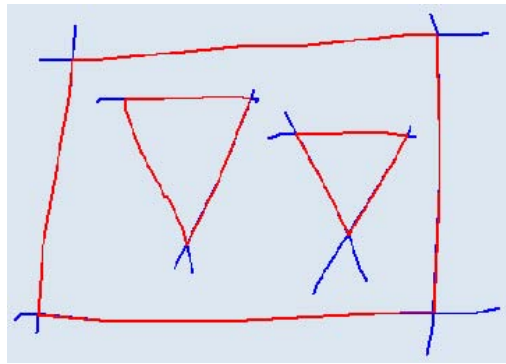
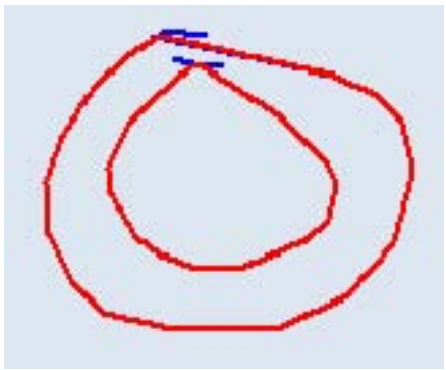
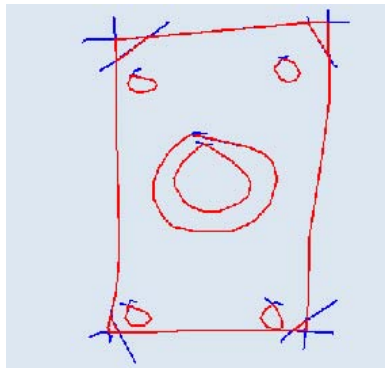
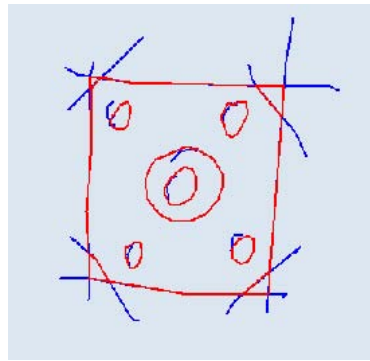


	0	1	2	3	4	5
0	0	1	0	0	0	0
1	1	0	1	0	1	1
2	0	1	0	1	0	0
3	0	0	1	0	0	0
4	0	1	0	0	0	1
5	0	1	0	0	1	0

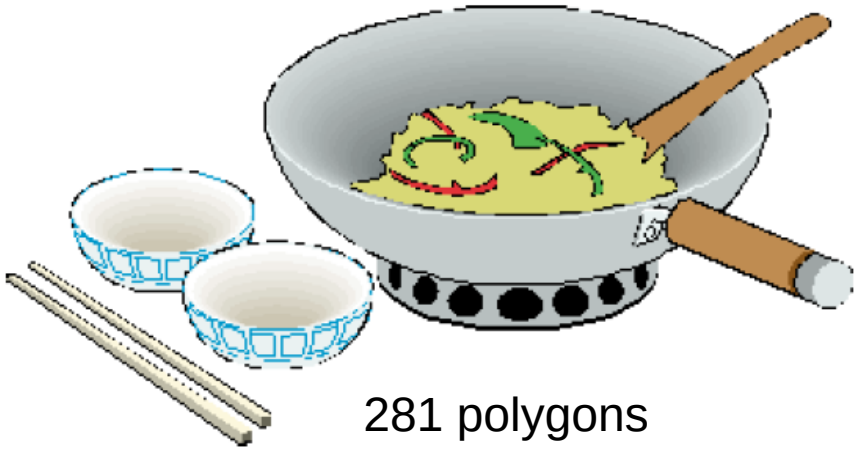
- Problem: graph matching is NP-complete
- Solution: compute a vector from the Eigenvalues of the adjacency matrix, describing the graph
  - High-dimensional for complex drawings
  - Low-dimensional for simple
  - Similar for structurally similar graphs
  - Stable with small changes in graph topology
- Do this at different levels of detail and for subgraphs

Sketched input

- Identify shapes from hand-sketched input
- Transform sketch into vector graphics
- Apply same analysis as for the DB

Analyzing clip art

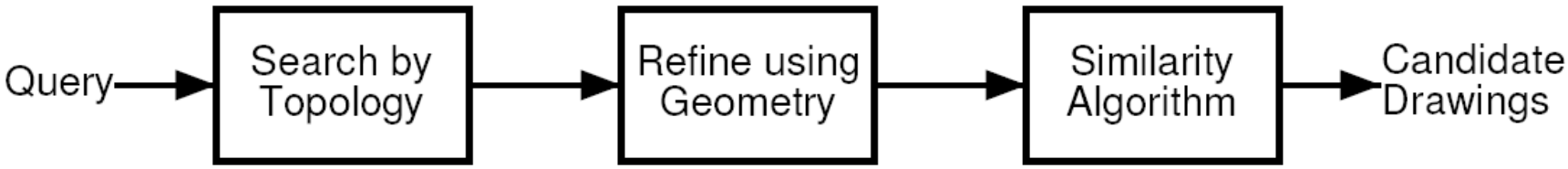


281 polygons

13 polygons

- WMF format uses 2 polygons for area and outline → discard one
- Clip art uses polygons with color gradients
  - Problem: shapes are found by color
  - → Simplify to uniform colors

Matching



- Compute the same descriptors from the query sketch

1) find topologically similar drawings

2) compare geometries used in the drawings

3) compute a similarity measure

- Return a ranked list of matches
- Tested for 100.000 drawings

Retrieving 3D models based on their shape

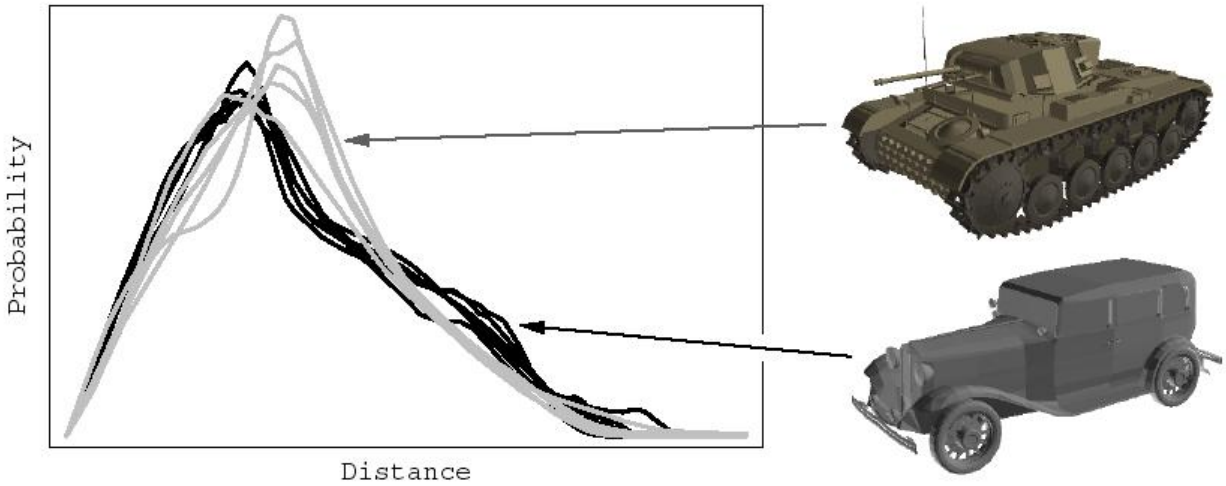
Thomas A. Funkhouser

http://shape.cs.princeton.edu/

Motivation

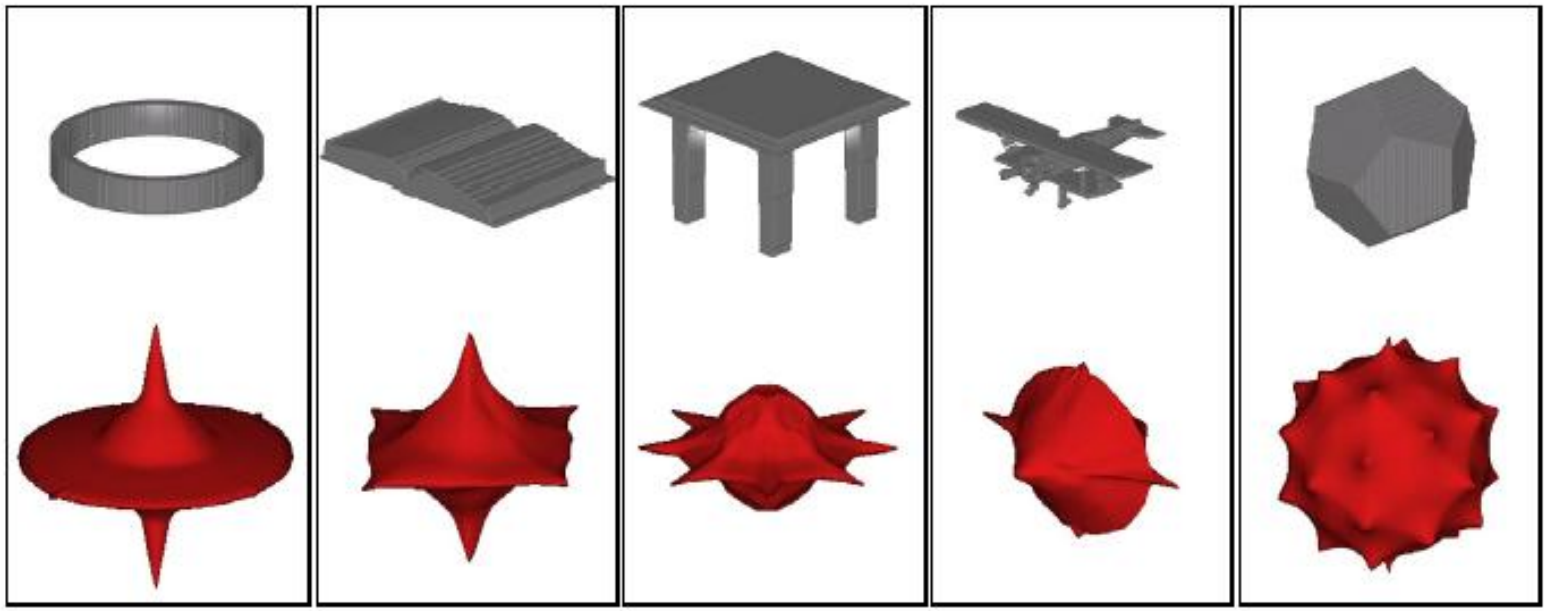
- shape-based retrieval and analysis of 3D models
  - To query model databases
  - To classify 3D models
- develop effective shape representations
- develop effective query interfaces

Shape Distributions



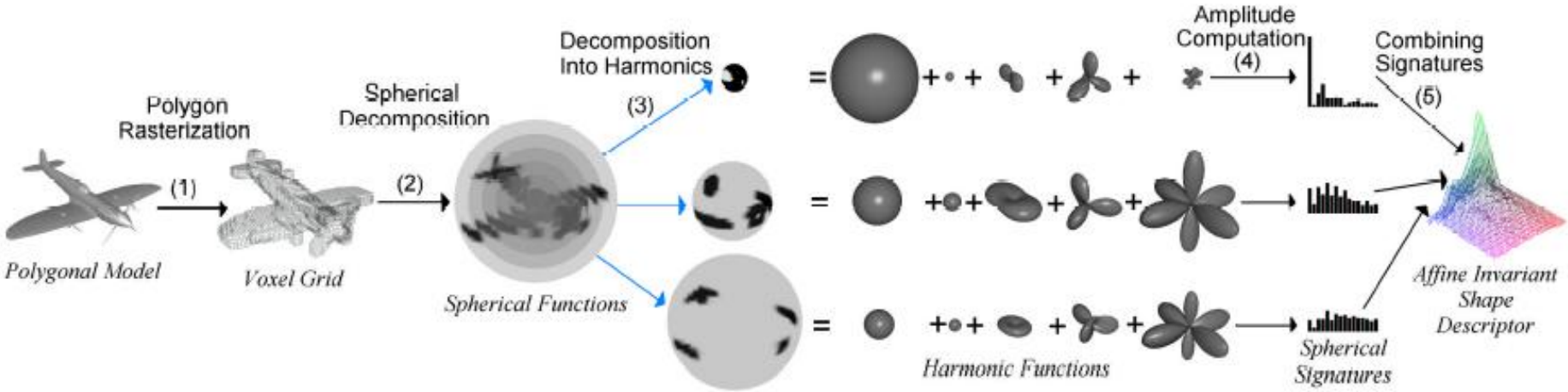
- E.g., probability distribution of Euclidean distances between pairs of randomly selected points on its surface
- samples can be computed quickly and easily
- resulting distributions are invariant to similarity transformations, noise, tessellation, cracks, etc.
- normalization step for scale invariance

Reflective Symmetry Descriptors



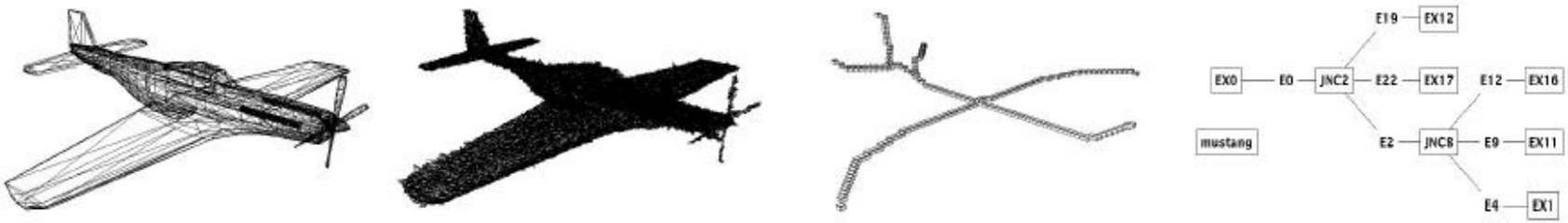
- Put planes through the object‘s center of mass
- Determine reflective symmetry w/r to these planes
- Scale a unit sphere with the degree of symmetry along the normal vector of the plane
- → descriptor is invariant with scale, easily comparable
- → describes global symmetry properties of a 3D shape

Spherical Harmonics



- Compute voxel grid from model
- Sample it on spheres around the center of mass
- Decompose results into harmonic base frequencies (like Fourier transform)
  - Invariant under rotation!
- Combine signatures to a complete shape descriptor
  - Invariant under affine transformations

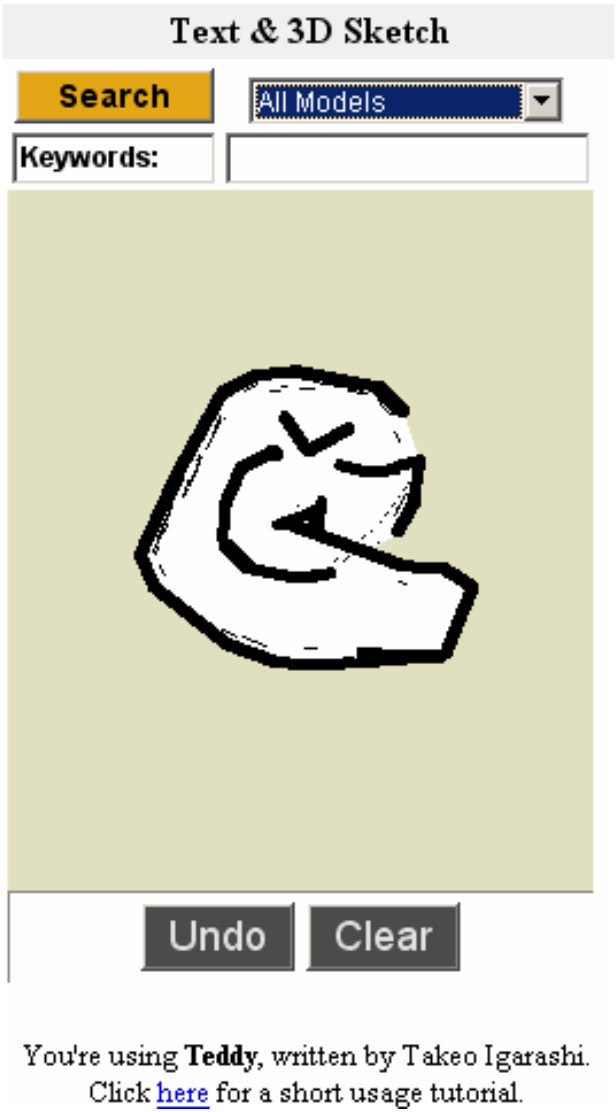
Skeletal Graphs

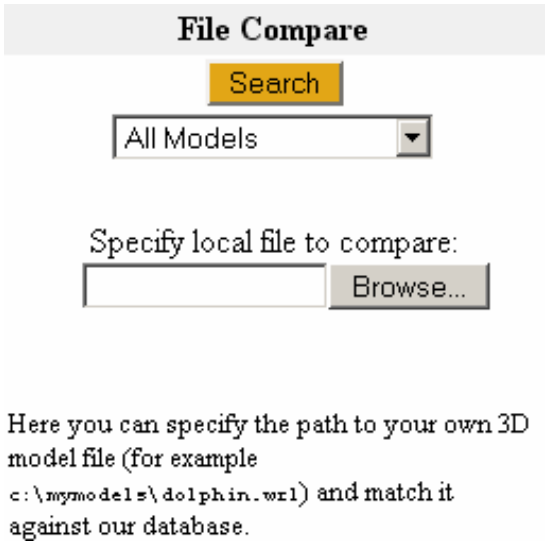


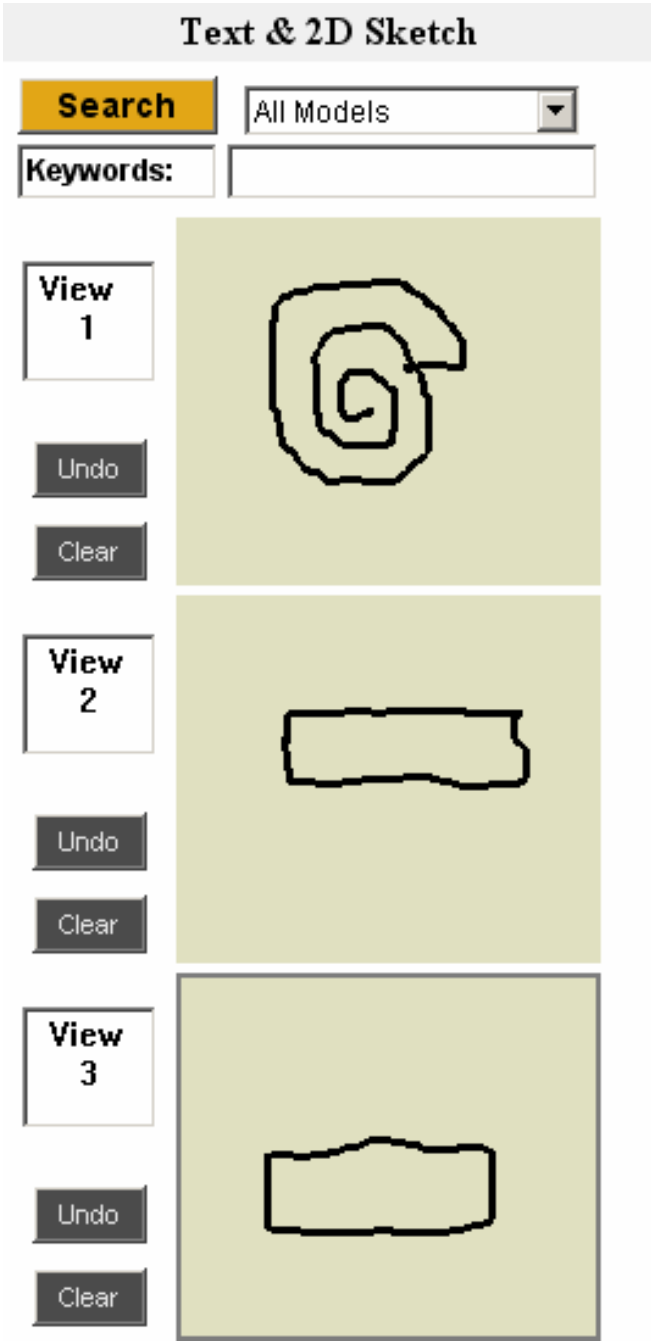
- Transform object into voxels
- Skeletonize it, i.e. compute center lines
- Derive a structure graph from this skeleton
  - Invariant with scale
  - Similar for similar shapes

- → see vector graphics retrieval 10 min. ago ;-)

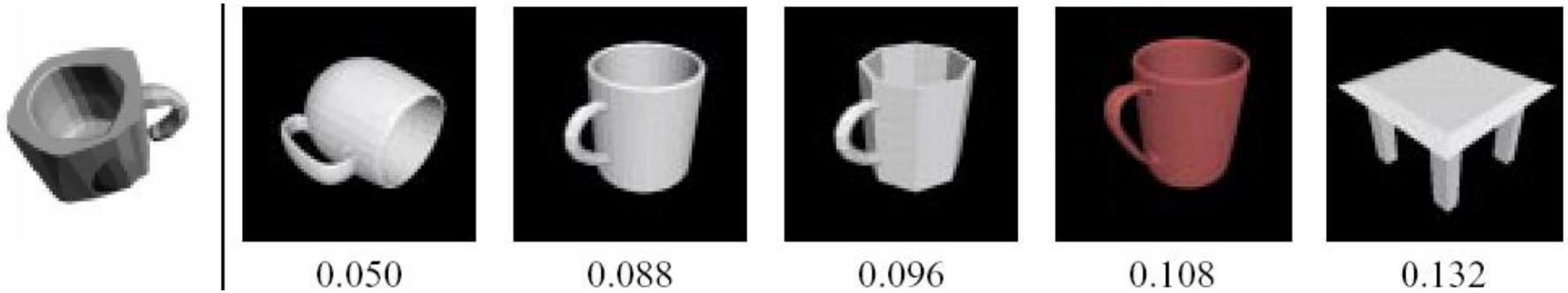
Query interfaces

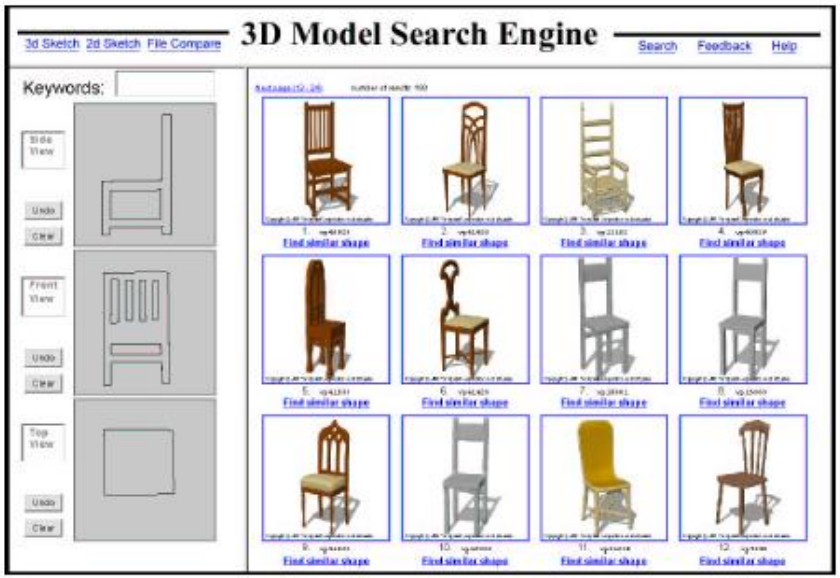


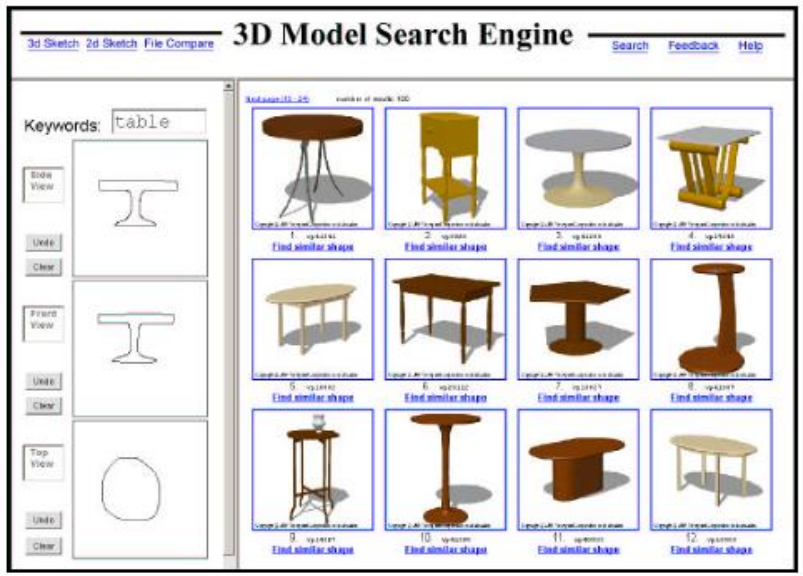




Results (Movie)





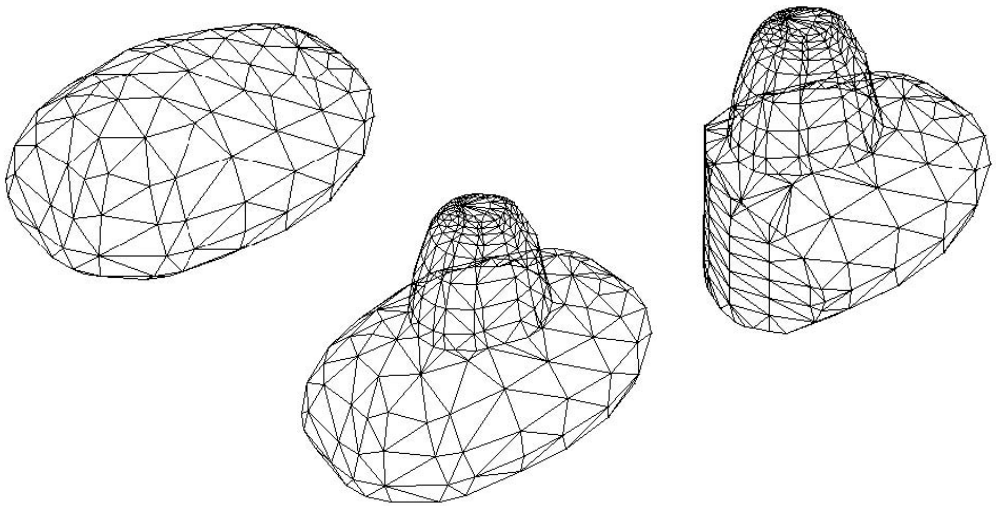


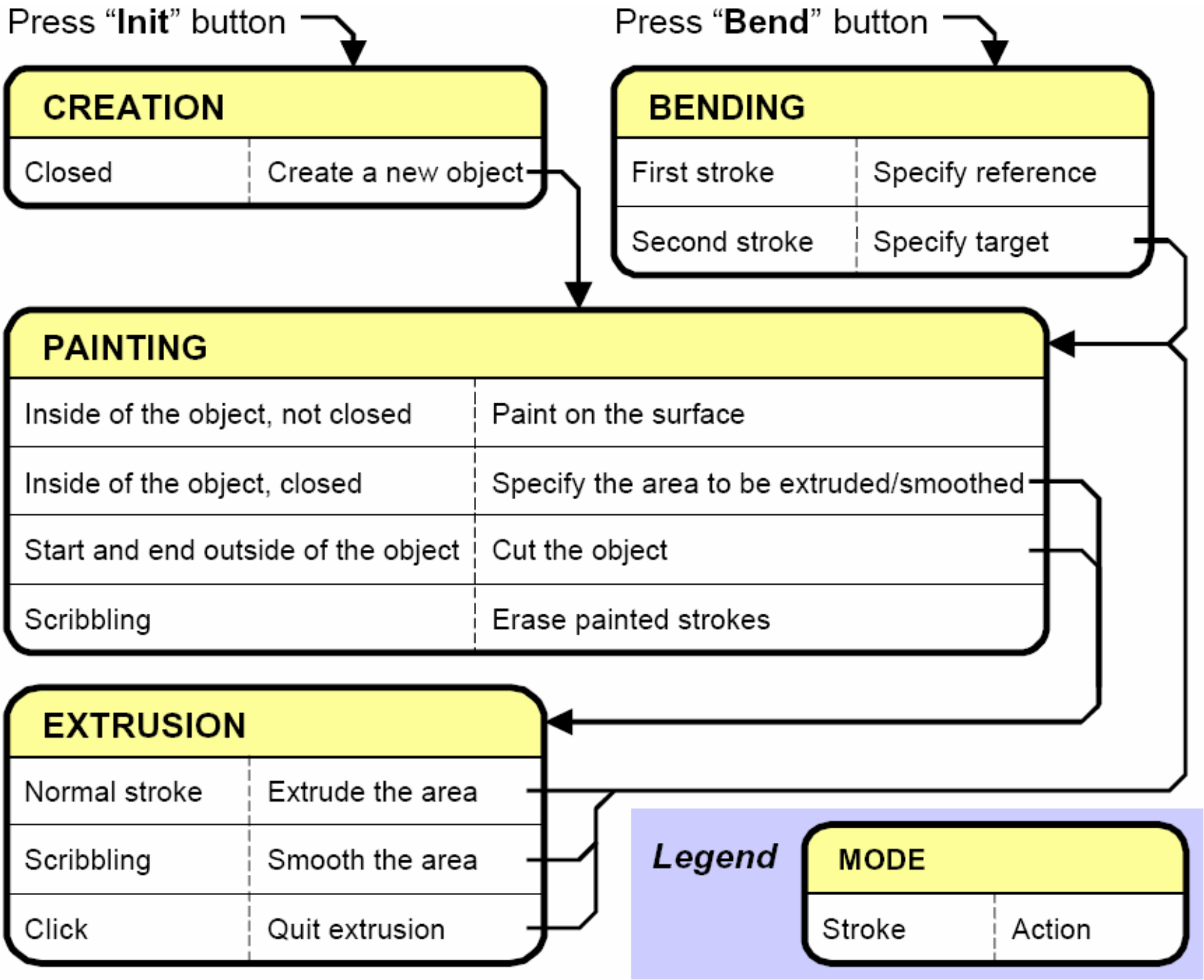
3D sketching 1

Some of Takeo Igarashi‘s work

Teddy [Igarashi, Siggraph 99]

- Idea: sketch shapes of freeform objects on a 2D surface
- Inflate the shape in both directions with the amount dep. on its width
- Allow extrusions and boolean operations
- Allow deformations of objects



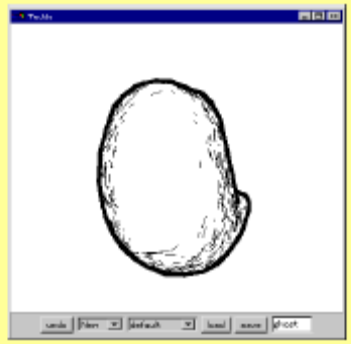

Press "Init" button
Press "Bend" button
CREATION
Closed
Create a new object
BENDING
First stroke
Specify reference
Second stroke
Specify target
PAINTING
Inside of the object, not closed
Paint on the surface
Inside of the object, closed
Specify the area to be extruded/smoothed
Start and end outside of the object
Cut the object
Scribbling
Erase painted strokes
EXTRUSION
Normal stroke
Extrude the area
Scribbling
Smooth the area
Click
Quit extrusion
Legend
MODE
Stroke
Action

Basic shapes and painting

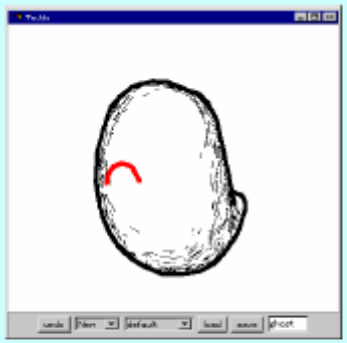
a) initial state

b) input stroke

c) result of creation

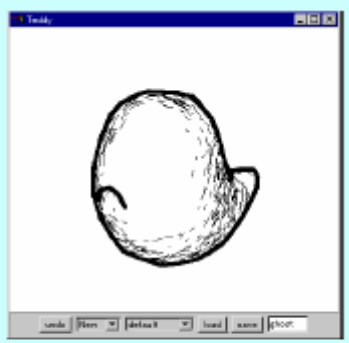


d) rotated view



e) painting stroke

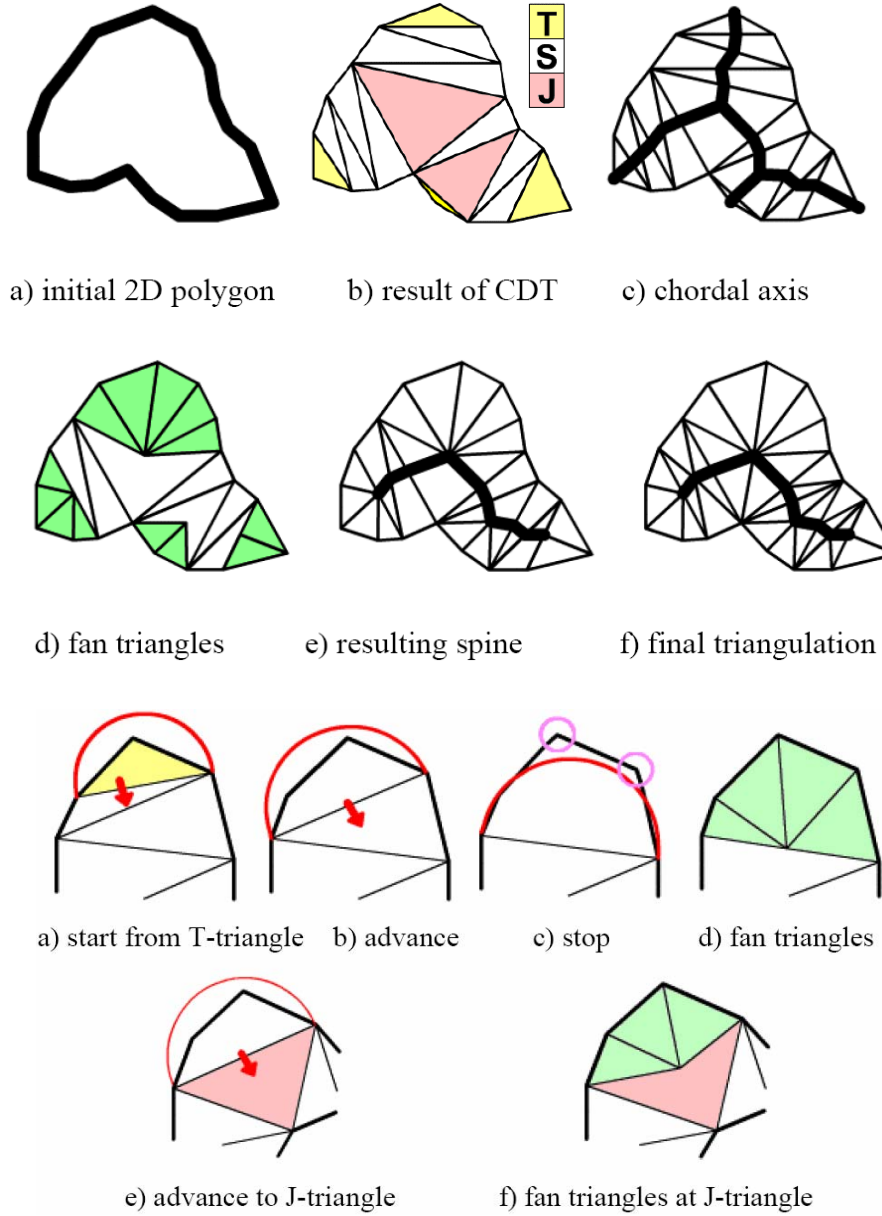
f) result of painting



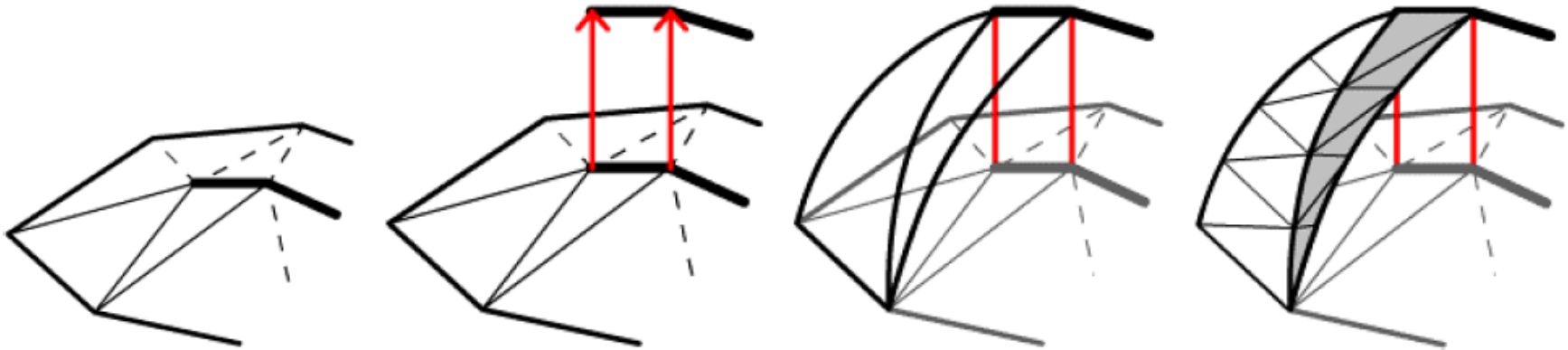
g) rotated view

Finding the spine

- Constrained Delaunay Triangulation
- 3 classes of triangles
  - Terminal (2 outer edges)
  - Sleeve (1 outer edge)
  - Junction (no o. edge)
- Expand T triangles →
- Create „fan triangles“ →
- Spine = line through midpoints of interior edges of S triangles

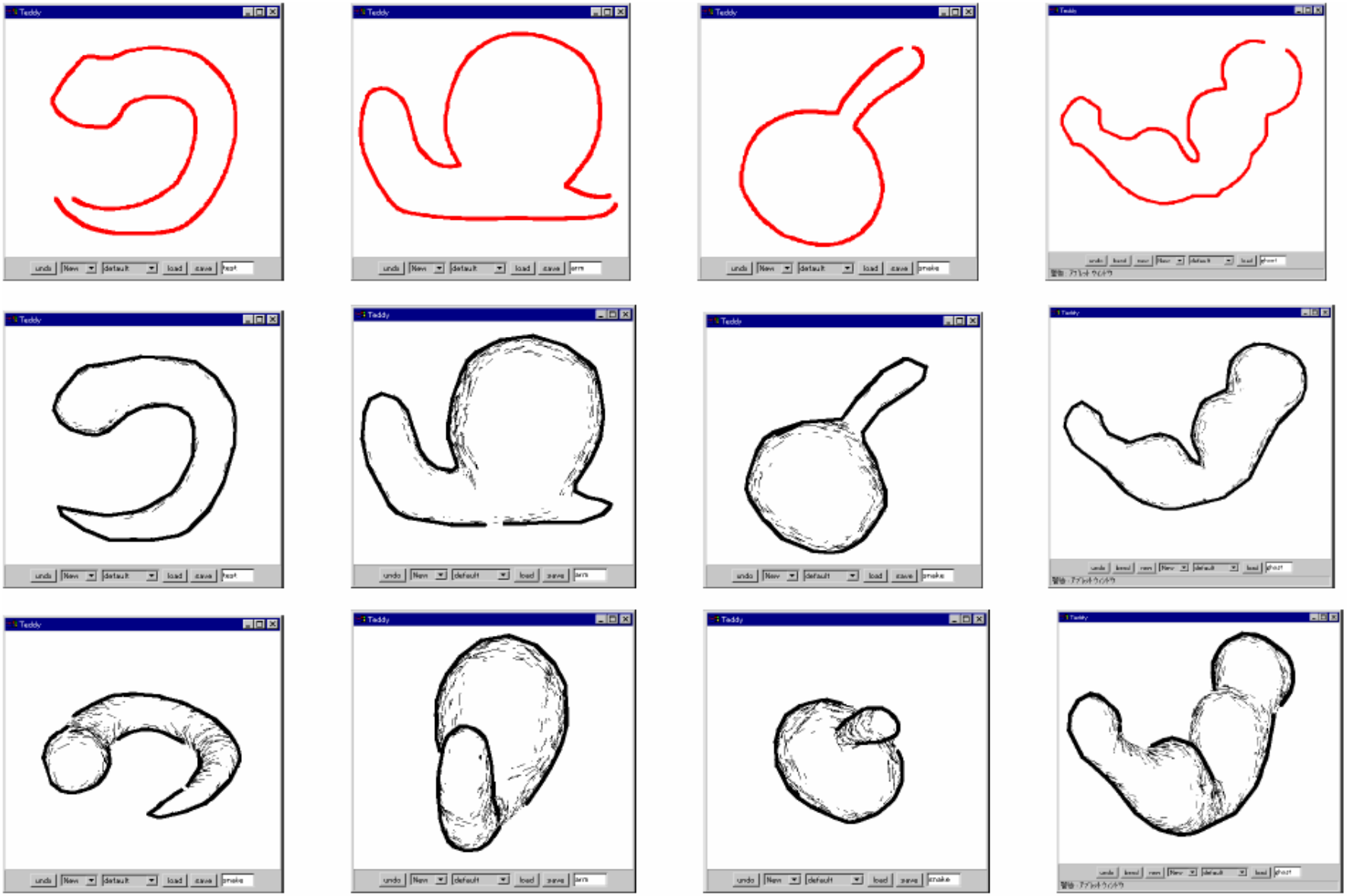


Inflating the spine



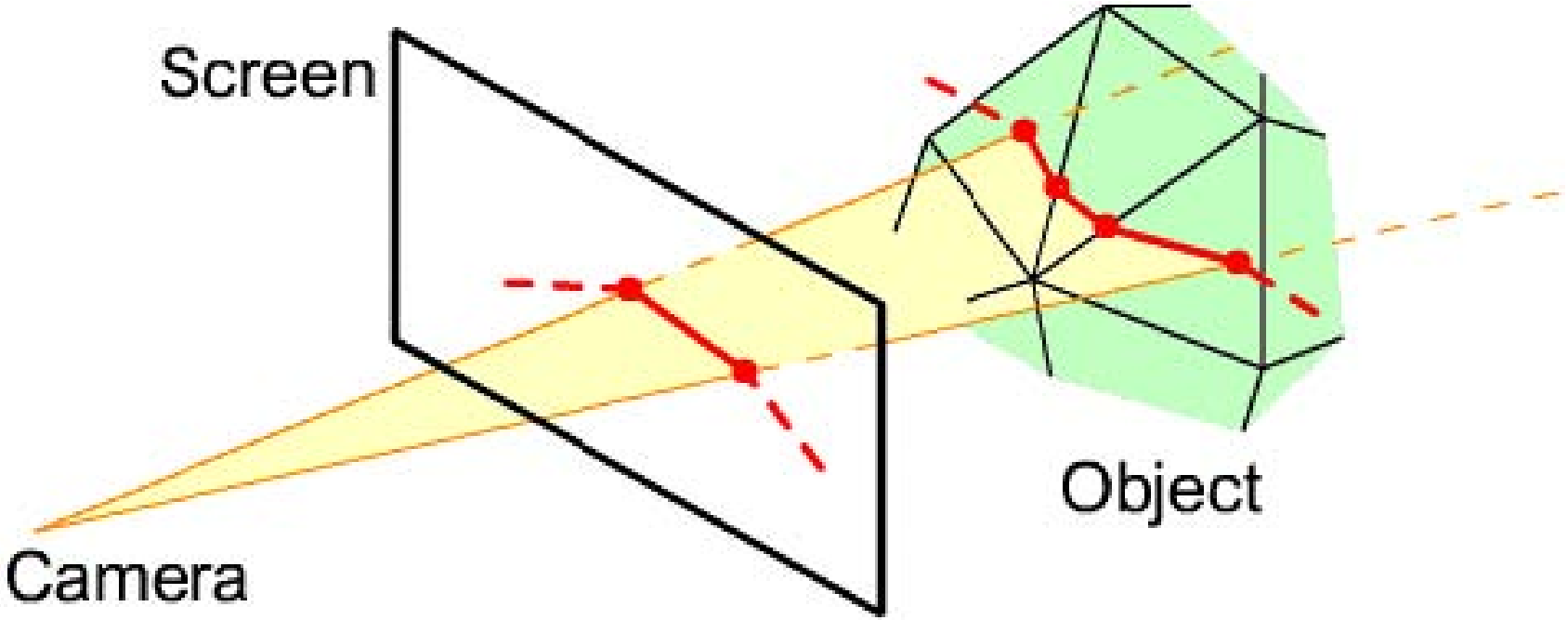
a) before b) elevate spines c) elevate edges d) sew elevated edges

- Split polygons at the spine
- Elevate the spine, depending on shape width
- Elevate the edges such that intersections form ovals
- Sew the elevated edges together again.

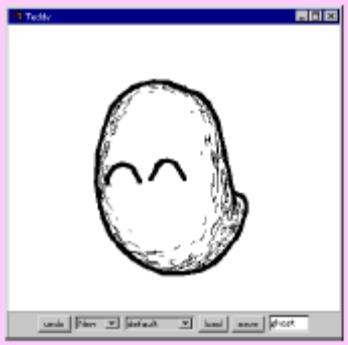


a) snake b) snail c) cherry d) muscular arm

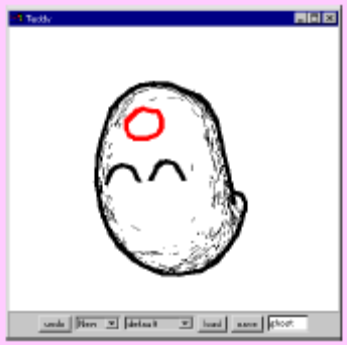
Painting on the surface



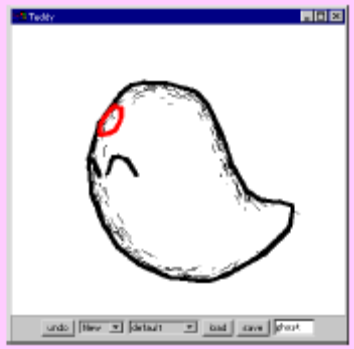
- Project the pen stroke onto the object surface



h) before extrusion



i) closed stroke



j) rotated view

k) extruding stroke

l) result of extrusion

m)

n) before cutting

o) cutting stroke

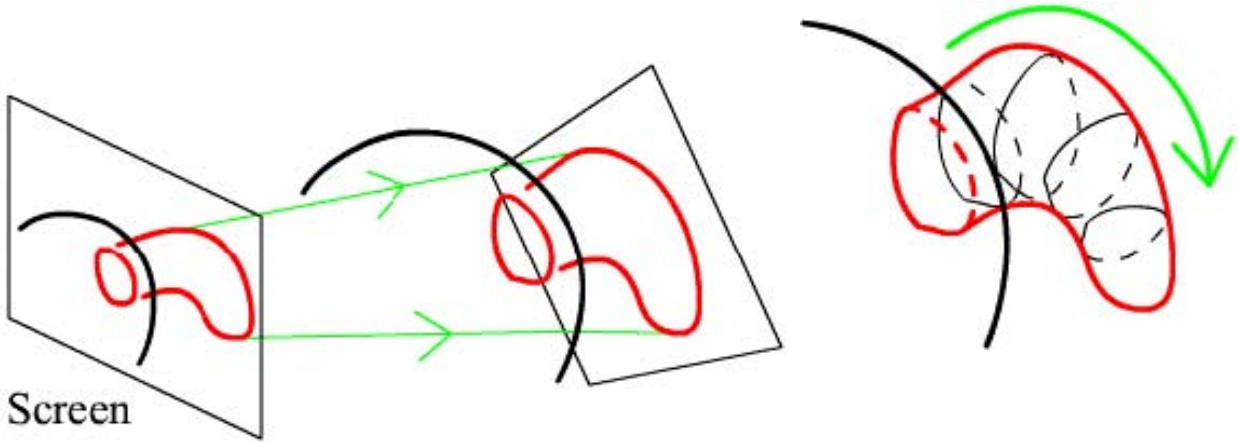
p) result of cutting

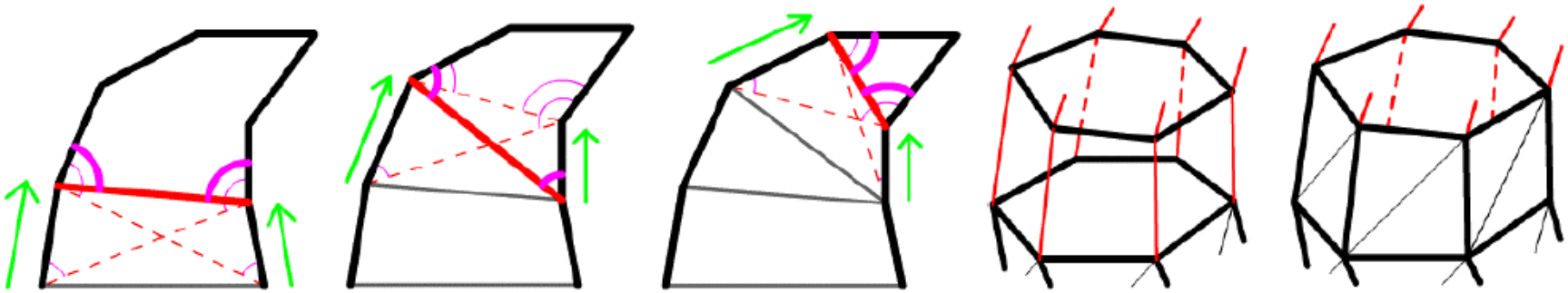
q) result of click

r) extrusion after cutting s) result of extrusion t) rotated view

Extrusion & cutting

Extrusion

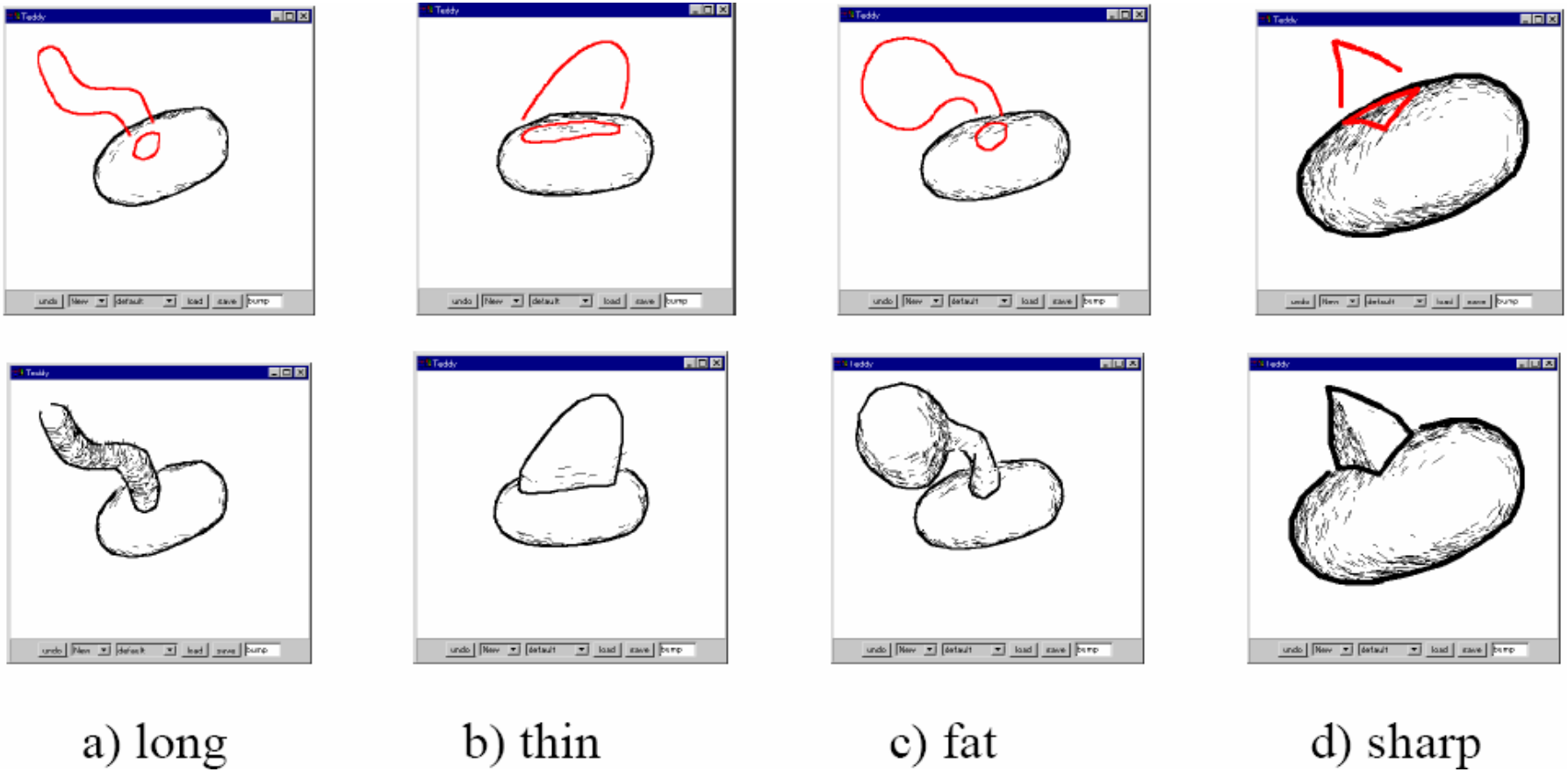




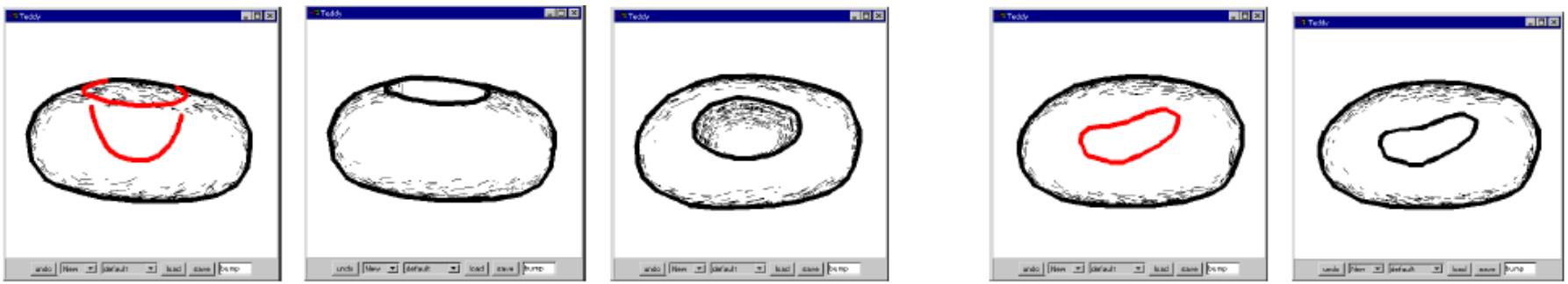
a) pointer advancing

b) sewing adjacent rings

- Project extrusion stroke to a plane in space,
  - perpendicular to the base ring
  - projection onto screen matches original stroke
- Sweep base ring along extrusion by alternatingly moving 2 pointers
- Connect the new rings, cut out the original base ring, sew together

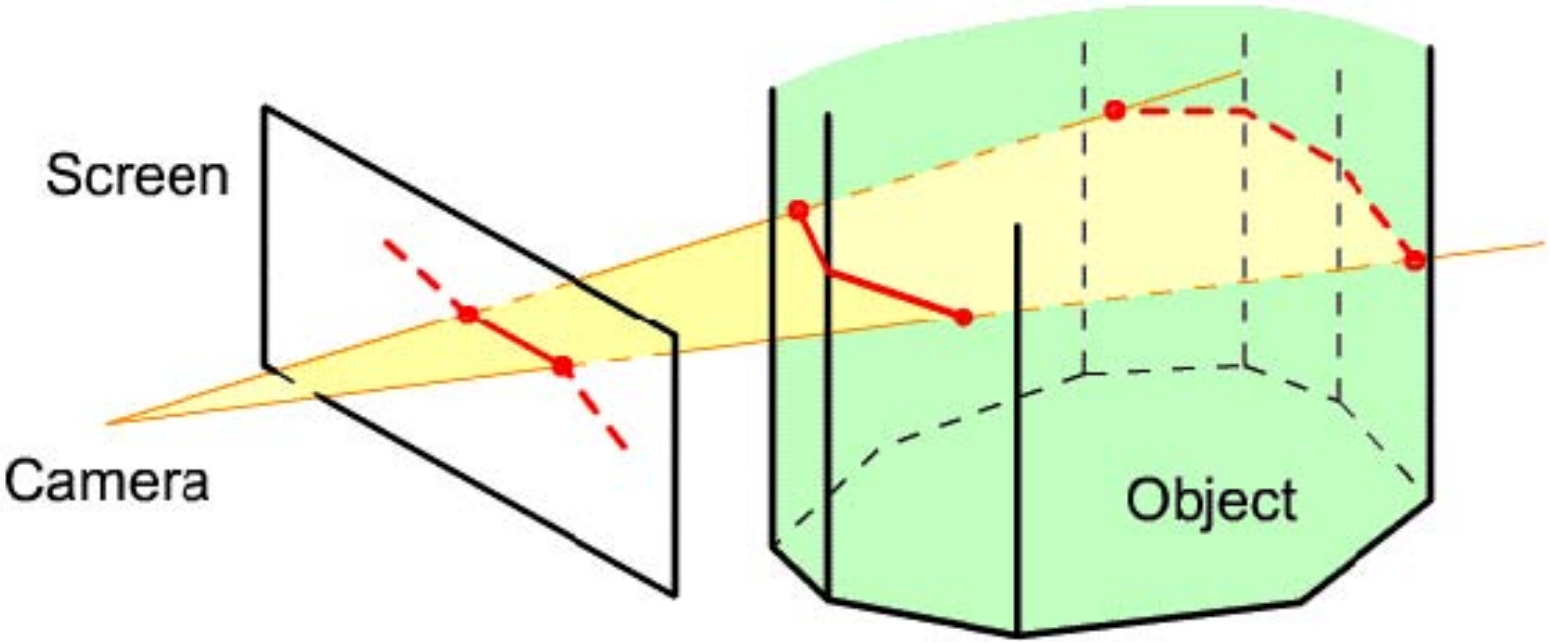


a) long b) thin c) fat d) sharp



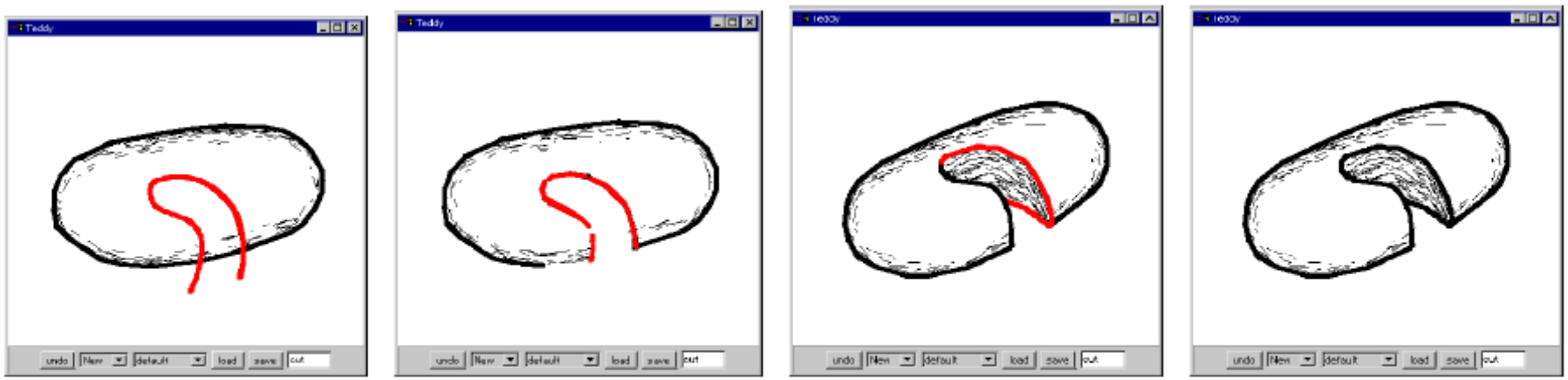
a) digging stroke b) result c) rotated d) closed stroke e) after click

Cutting

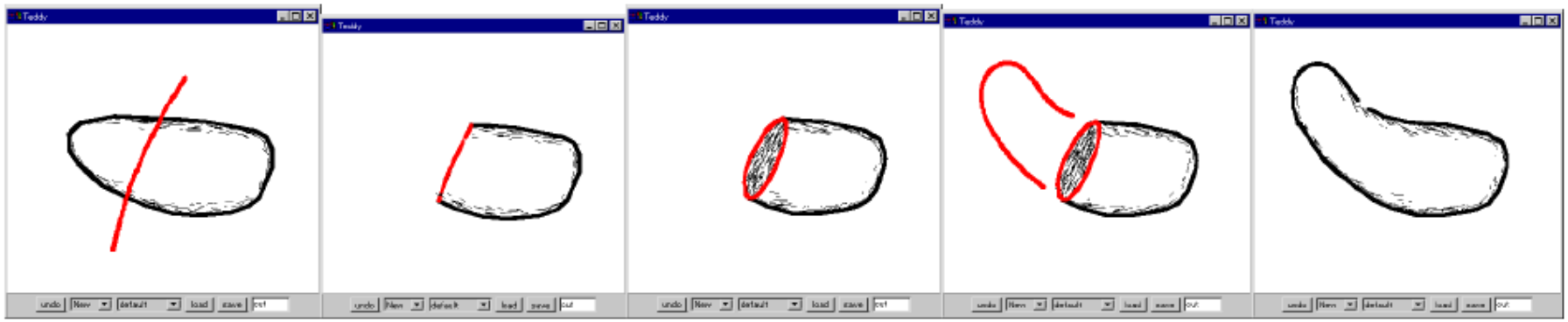


- Similar to painting
- Projection onto object‘s front and back sides
- Connect front and back points with a planar polygon

Examples

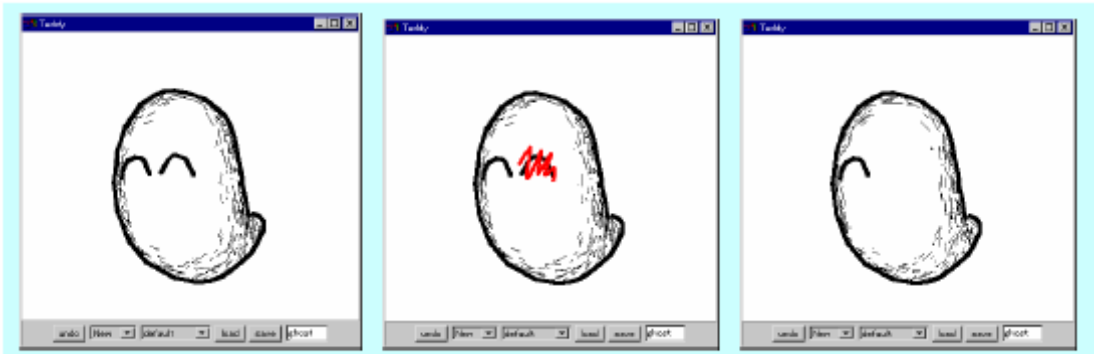


a) biting stroke b) result c) rotated view d) after click

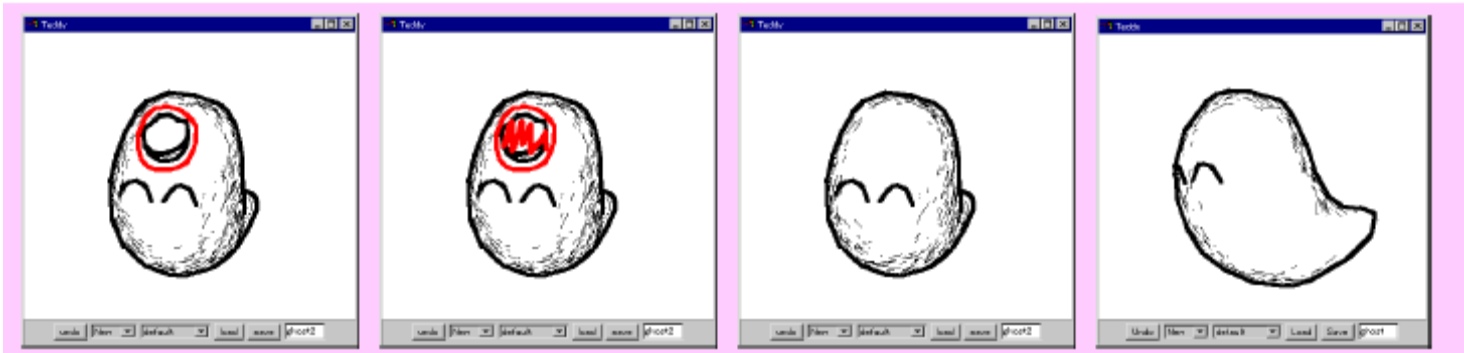


a) cutting stroke b) result c) rotated d) extruding stroke e) result

Erasing



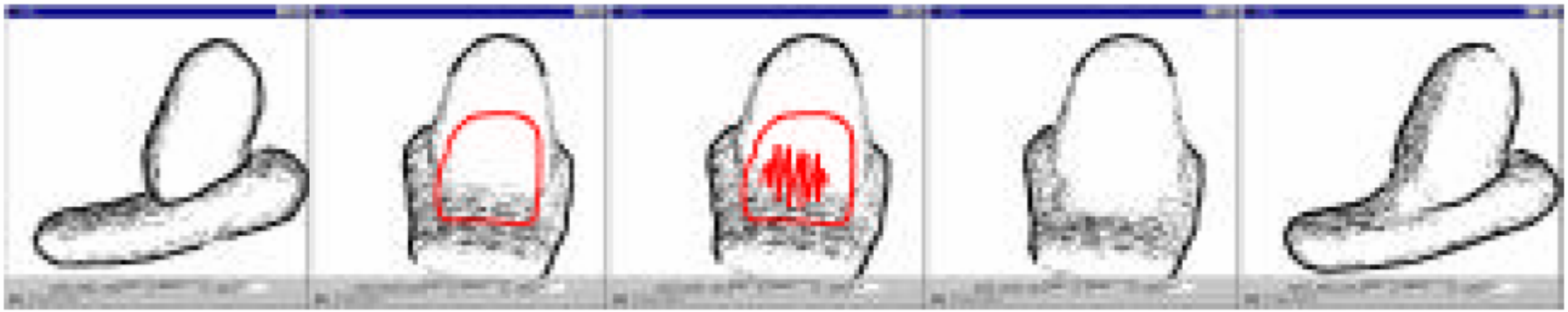
u) before erasing v) scribbling w) result of erasing



x) closed stroke y) scribbling z) result of smoothing z´) rotated view

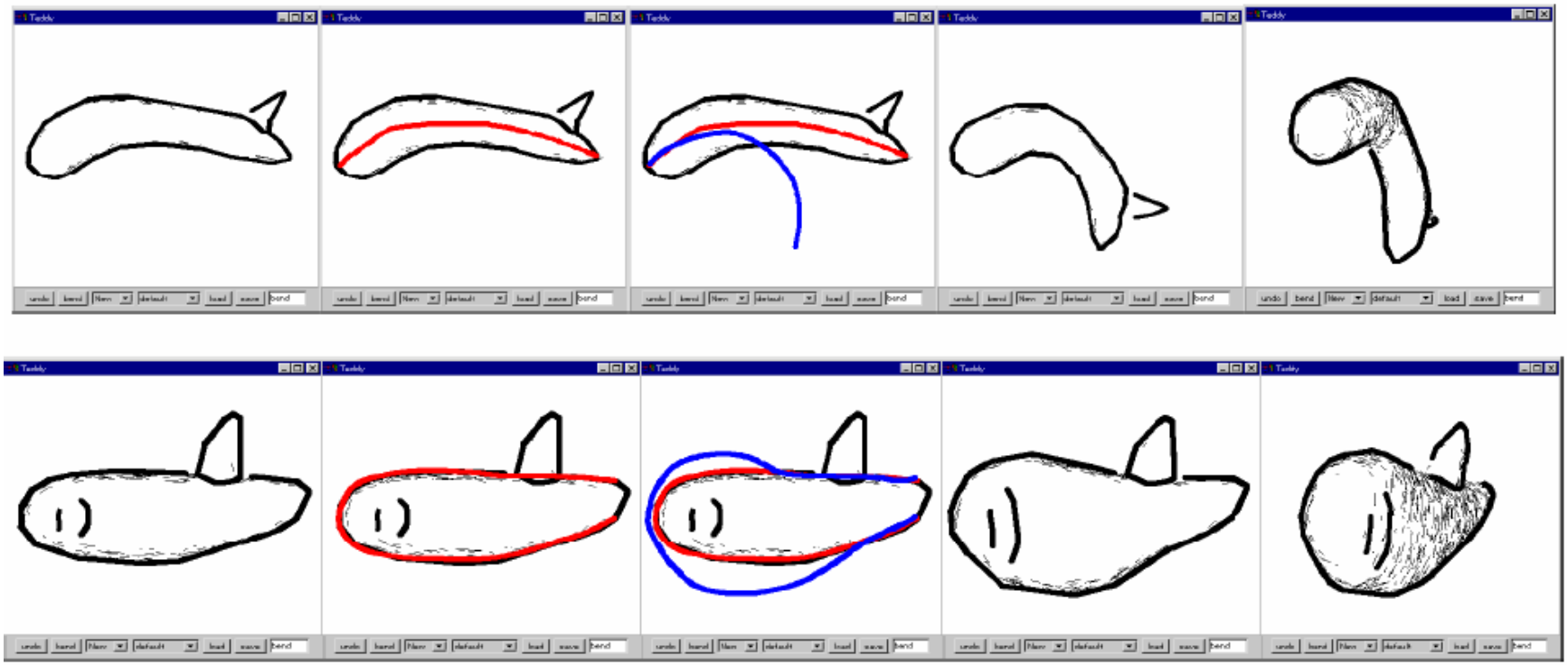
Examples

a) cleaning a cavity



b) smoothing a sharp edge

Transformation



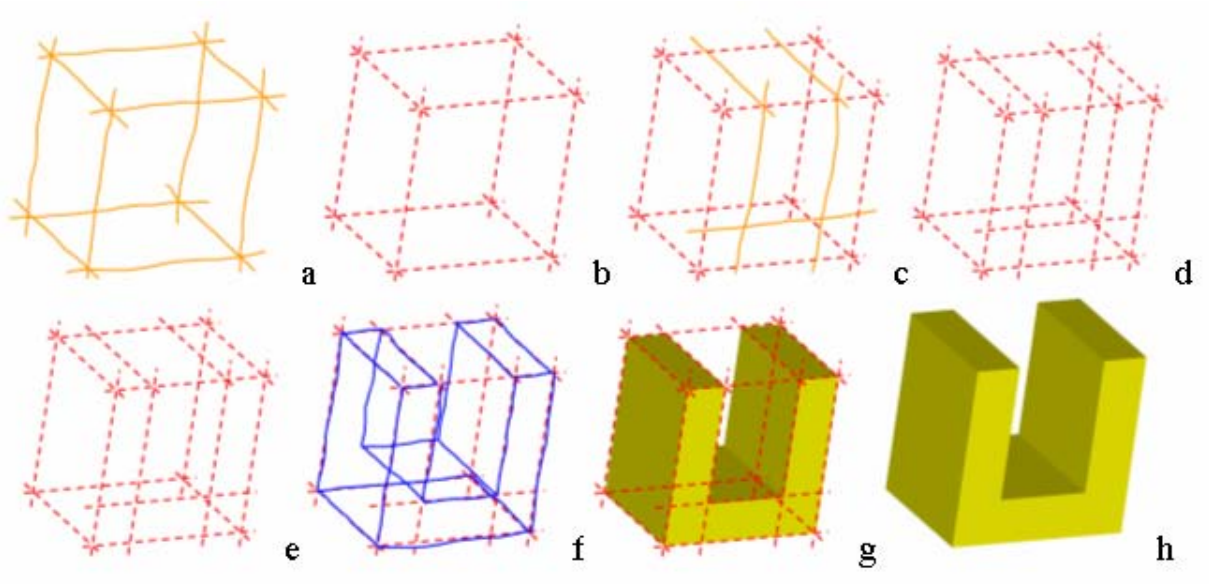
a) original b) reference stroke c) target stroke d) result e) rotated

3D sketching 2

Ferran Naya et al.

Smart Graphics 2003

3D Reconstruction based modeling

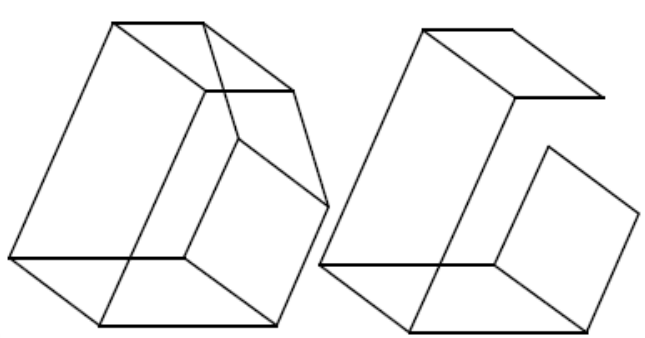


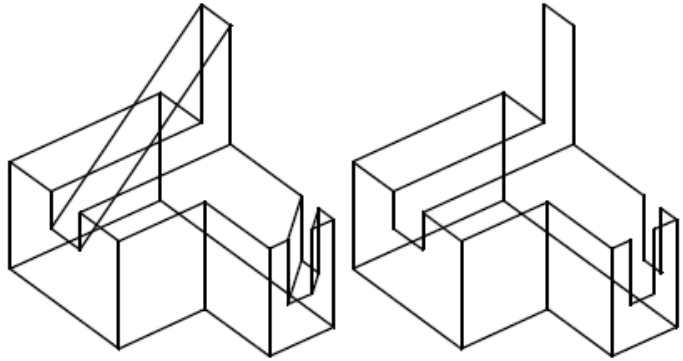
- Draw line strokes on the screen with pen (light pressure)
- Use them as leader lines as a reference frame
  - constant camera position
- Draw strong lines along leader lines to form a 3D model
- When a valid 3D model is recognized → shade

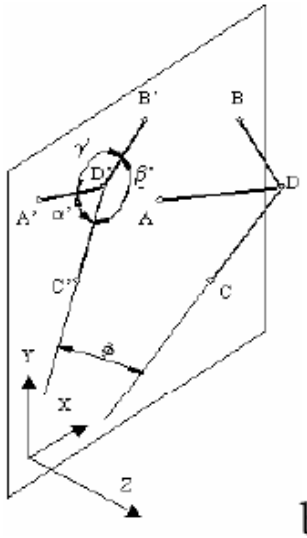
Line stroke beautification

- Automatic line slope adjustment
  - checking whether the new line is parallel to any of the principal axes of the sketch within a slope tolerance.
- Vertex point snap
  - look for vertices close to the line endpoints, within a given proximity tolerance.
  - If several such vertices, select the one closest to that line endpoint.
- Vertex on line snap and automatic line breaking
  - for endpoints of the new line which do not lie close to a model vertex, analyze whether the points are close to an existing edge, within a given tolerance.

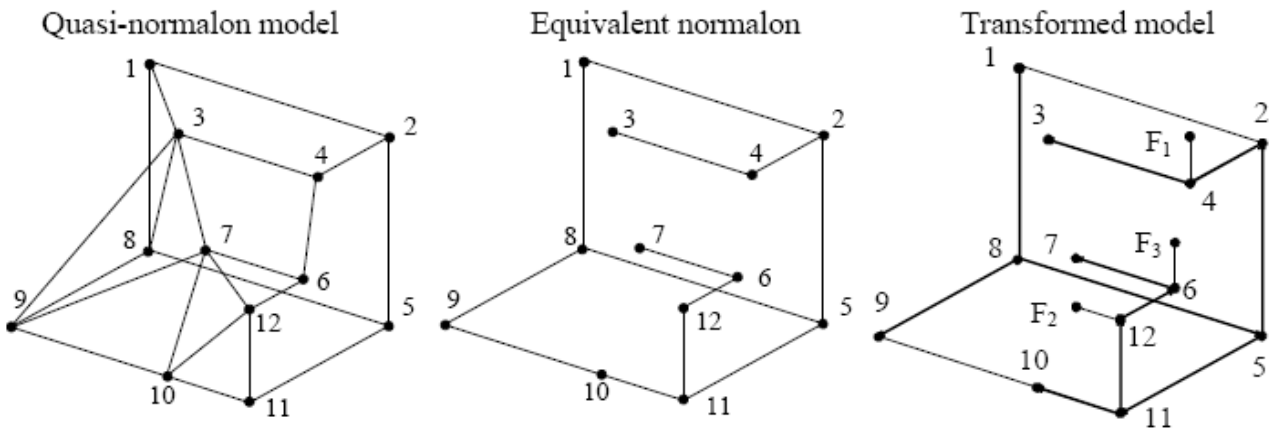
Inflation







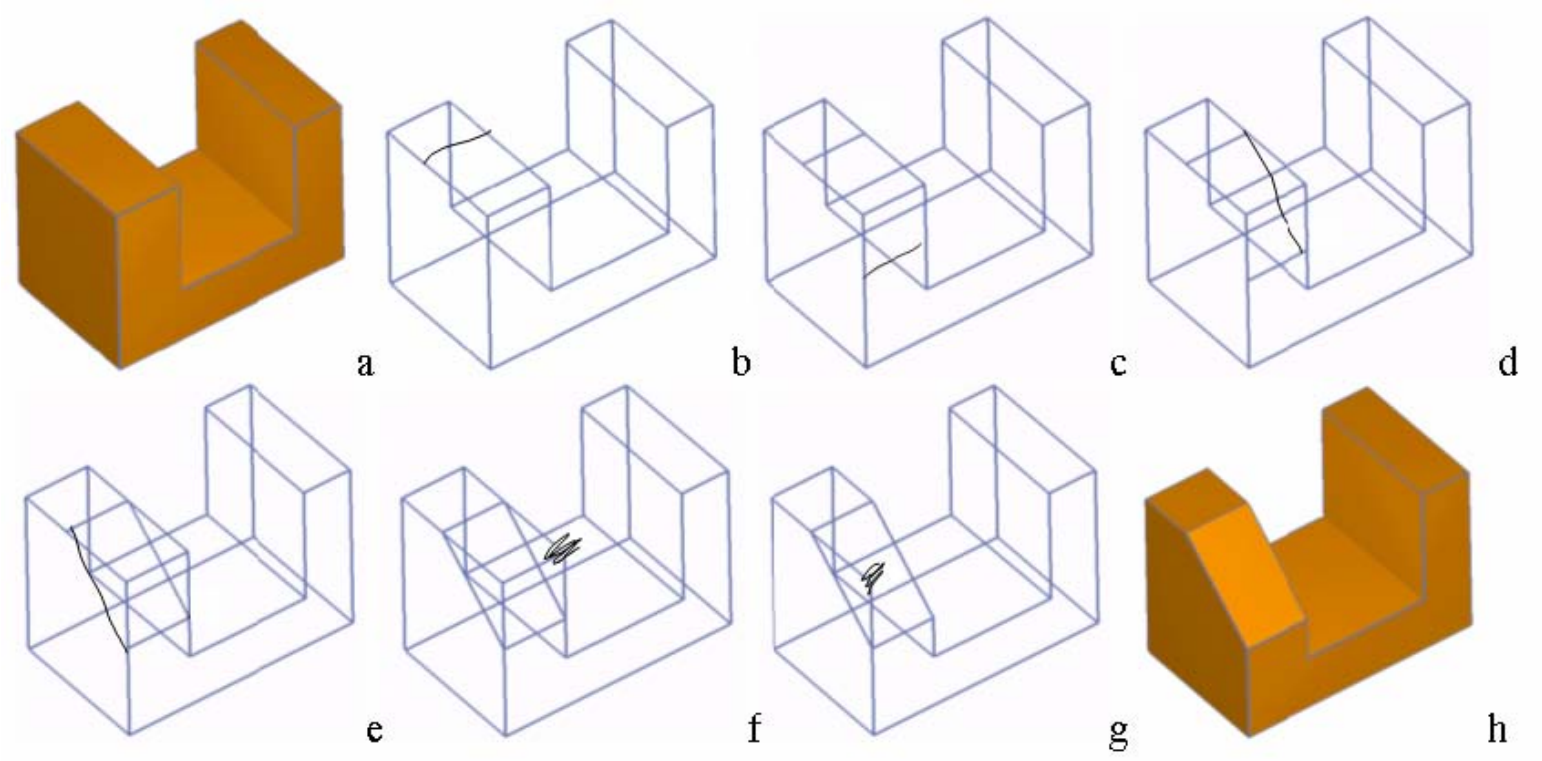
a



b

- Assume only right angles in objects (normalon)
- Analytically determine the orientation for each corner based on its edges to adjacent corners
- Find a plausible 3D reconstruction

Refining geometry by additional strokes



- Draw additional lines on the geometry
- Scratch gesture over edge
  - Construct smallest quadrilateral containing the gesture
  - Remove all intersecting lines

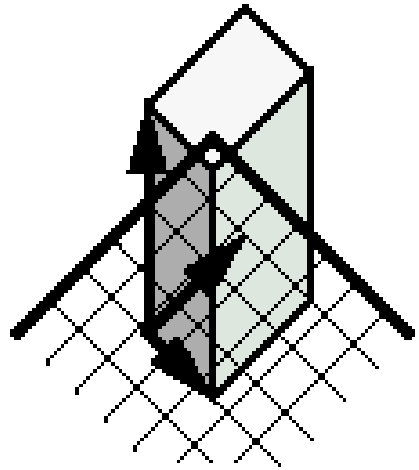
3D sketching 3

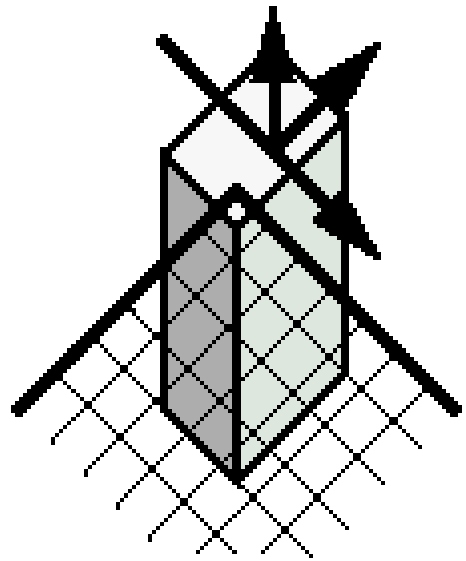
The classic: SKETCH

[Bob Zeleznik, Siggraph 96]

Sketch: drawing a cube

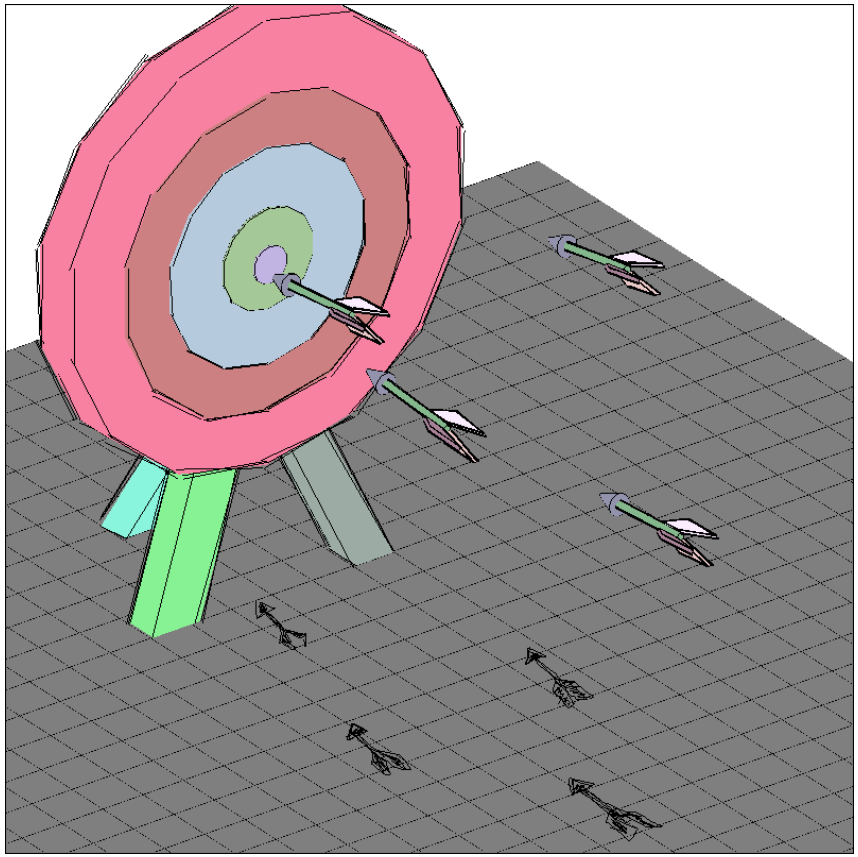
- Three line segment strokes,
  - axis-aligned with coordinate axes of the world
  - all strokes meet (more or less) at a single point
- The cuboid constructed from this gesture has its dimensions determined by the strokes





- Similar ways for other primitives

Sketch: some results



THE END