Visual Processing in CogSketch
Some Preliminaries

• Visual versus Spatial relationships:
  – Visual relationships: Computed over glyphs.
  – Spatial relationships: Hold between what is denoted by the glyphs
  – Visual relationships + genre + pose → Spatial relationships

• Our visual computations are inspired by psychological evidence when available
  – Best guesses otherwise
  – We expect it to continue to evolve
Glyphs

• Glyphs have two parts: *Ink* and *Content*
• Content = the entity represented by the glyph
  – Instance of some collection in the KB
• Ink = visual representation of the content
  – Consists of all of the ink drawn between button presses
• Visual properties are computed on the ink
  – Only coarse visual properties computed automatically
    • Bounding box
    • Closed contour (ink needn’t be connected)
    • Major/minor axes
  – Small set of visual relationships between glyphs
  – Segmentation, other visual relationships computed on demand (e.g., perceptual sketchpad)
Some CogSketch spatial computations

- Grouping
- Voronoi diagrams
- Positional relations
- Qualitative Topology
- Shape decomposition
- Mental Rotation
Qualitative Spatial Reasoning

• **Claim:** Symbolic vocabularies of shape and space are central to human visual thinking (cf. Forbus 1980; Forbus, Ferguson & Usher 2001)
  - They are computed by our visual system
  - Their organization reflects task-specific conceptual distinctions and conventional symbol systems as well as visual distinctions
  - They provide the bridge between conceptual and visual representations
Metric Diagram/Place Vocabulary model

- Metric Diagram: Quantitative, visual representations and processing
- Place Vocabulary: Task-specific qualitative representations of shape and space, grounded in the metric diagram

FROB (Forbus, 1980)

SKETCHY
(Pisan, 1994)

GIS-based Trafficicability Reasoner
(Donlon & Forbus, 1999)
Spatial Reasoning in CogSketch

- Timestamped ink and interface events
- Multimodal integrator and parser
- Metric Diagram
  - Vector Processor
  - Ink Processor
  - New glyphs
  - Basic spatial properties and relationships of ink
- Working memory for sketch (includes logic-based TMS)
- Place vocabularies: Voronoi diagrams, visual groups, position-finding, path-finding
Qualitative Topology

Cohn et al’s RCC8 relational algebra

- Provides natural vocabulary for some visual concepts
  - Containment: NTPP, TPP
  - Touching: PO, EC
Using RCC8

• Compute relationships directly from ink
  – Transitivity algebra unnecessary
  – Need to be clever about noise

• Computed between every pair of glyphs on a layer
  – Incrementally updated when a glyph is moved or resized
  – Only computed across layers on demand

• Internal uses
  – Controlling computation of other relations
    • Positional relations not computed unless RCC8-DC
  – Direct inference of domain relations, depending on nature of contents (e.g., touching & containment)
Contained Glyph Groups

• When more than one glyph is NTTPi, TPPi of some other glyph
  – Single-level, groups can be found recursively
• (ContainedGlyphGroupFn
  (GlyphFn Object-9 User-Drawn-Sketch-Layer-1)
  (TheList (GlyphFn Object-15 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-16 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-19 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-20 User-Drawn-Sketch-Layer-1)))
Connected Glyph Groups

• Set of glyphs connected via EC or PO

  (ConnectedGlyphGroupFn
   (TheList (GlyphFn Object-10 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-11 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-12 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-21 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-22 User-Drawn-Sketch-Layer-1)
   (GlyphFn Object-9 User-Drawn-Sketch-Layer-1)))
Computing Glyph Groups

• Connection graph: Nodes = glyphs, Edges between all pairs that are EC or PO
  – Connected Glyph Groups = connected subsets of connection graph

• Containment graph: Nodes = glyphs, Edges between all pairs that are TPPi or NTPPi.
  – Contained glyph groups = All glyphs with more than one glyph inside of them, only counting directly inside glyphs

• Incrementally maintained as sketch updated
Glyph Groups Can Help Matching

Without glyph groups

With glyph groups
Voronoi Diagrams: A tutorial
Voronoi Diagrams: A tutorial

Red = cell boundary in Voronoi diagram
Green = arc in Delaunay triangulation
Voronoi diagrams and Delaunay triangulations are duals
Voronoi Diagrams: A tutorial

A is adjacent to B & C
Voronoi Diagrams: A tutorial

Edwards & Moulin (1998) argue that Voronoi diagrams are useful for capturing visual adjacency.
Voronoi Relationships

- Voronoi diagram = edges that are equidistant from a pair of points (called *sites*)
- Provides a notion of adjacency
- Generalizing to glyphs:
  - Use sample points along contour of glyphs to define standard Voronoi (site-level Voronoi)
  - Label edges with glyph membership
  - Define glyph-level relations in terms of site relations
    - E.g., two glyphs are *siteAdjacent* ⇔ ∃ samples on glyphs | edge-connected in site-level Delauney triangulation
- One Voronoi diagram computed per subsketch in CogSketch
Positional Relations

• Provide qualitative position, orientation information with respect to global frame of reference
  – For glyphs, leftOf, rightOf, above, below
  – For contents, depends on genre and viewpoint
    • Physical/side: Same as glyphs
    • Geospatial/TopDown: northOf, southOf, eastOf, westOf
    • Abstract or Discrete: No implications for contents
  – Two versions
    • Take relative sizes into account
    • Use centroid
Local Relational Neighborhood Hypothesis

• When to compute positional relations? Between every pair of glyphs on a layer, like RCC8?
  – Bad idea! Loses locality
• Idea: Network of positional relations should provide “framing effect” in visual structure.
• Necessary condition: Glyphs must be siteAdjacent on their subsketch’s Voronoi diagram
  – Can also be computed on demand
• Hypothesis: This use of local neighborhood structure corresponds to default encoding method in human sketch perception
Voronoi adjacency guides positional relation finding

Positional relations only created between site-adjacent glyphs
Positional Relations help frame visual structure

Corresponds to what people choose in fast response-time task

Corresponds to what people choose when given more time
Spatial relations suggest conceptual relations

- Qualitative spatial relationship \textit{rcc8-TPP} in \textit{PhysicalView} indicates \textit{inRegion}
- \textit{inRegion} specializations suggest possible conceptual interpretations
  - Nucleus is part of Cell.
  - Nucleus is found in Cell.

- Qualitative spatial relationship \textit{rcc8-EC} suggestions include
  - Virus is connected to Cell.
  - Virus touches Cell.
  - Virus is adjacent to Cell.
  - Virus covers Cell like hair.
  - ...

World knowledge or linguistic input is often needed to disambiguate conceptual relations.
Perceptual Sketchpad
Perceptual Sketchpad Motivation

• Facility for experimenting with expressive representation of shapes
  – Decomposing glyphs
  – Within-glyph relationships also important
    • e.g., symmetry
  – Modeling mental rotation

• Still experimental, hence separate subsystem
  – Not all CogSketch users need it
  – As it stabilizes, it will become part of the default CogSketch visual processing
Understanding Form

• Focus is on understanding the form of glyphs
  – *Don’t* recognize a glyph
  – *Do* recognize that two glyphs are the same shape
  – Identify transformations between two glyphs’ shapes
    • Scaling
    • Rotation
    • Reflection
Two Levels of Representational Focus

1) Shape Representation
   - Default CogSketch representation level
   - Glyphs are the entities
   - Represent attributes of, relations between glyphs
Two Levels of Representational Focus

2) Edge Representation
   - Glyph is automatically segmented into edges
   - Edges are the entities
   - Represent attributes of, relations between edges within a glyph
Shape Relations

1) Compare two glyph’s edge representations to find corresponding edges
2) Compare orientations of corresponding edges to identify rotations or reflections
Shape Relations

1) Compare two glyph’s edge representations to find corresponding edges

2) Compare orientations of corresponding edges to identify rotations or reflections
1) Compare two glyph’s edge representations to find corresponding edges

2) Compare orientations of corresponding edges to identify rotations or reflections

Reflection: X Axis
Sampling of Spatial Vocabulary

- **Relations**
  - Relative position
  - Topology (rcc8)
  - Frame-of-reference
  - Shape

  **Transformations**
  - Same-shape
  - Rotation
  - Reflection
  - Relative Size

- **Attributes**
  - Fill color
  - Edge color
  - Shape Type
  - Symmetry

- **Shapes**
Sampling of Spatial Vocabulary

Edges

• Relations
  – Corners
    • Concave/Convex
  – Relative orientation
    • Parallel/Perpendicular
  – Relative length

• Attributes
  – Straight/Curved
  – Horizontal/Vertical
Using the Perceptual Sketchpad

• CogSketch comes with a Perceptual Sketchpad demo
  – Choose “New Perceptual Sketchpad” from the File Menu

OR

– Open one of the examples from the sketches directory
  • PSketchpad_Example1
  • PSketchpad_Example2
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad

PSketchpad_Example2
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad
Using the Perceptual Sketchpad

- If there is one glyph in each entry
  - Edge representations will be used

- If there are multiple glyphs
  - Shape representations will be used

- Elements will be color-coded to indicate correspondences
  - Right-click and choose “Unmark all glyphs” to remove colors
Using the Perceptual Sketchpad

• If there is one glyph in each entry
  – Edge representations will be used

• If there are multiple glyphs
  – Shape representations will be used

• Elements will be color-coded to indicate correspondences
  – Right-click and choose “Unmark all glyphs” to remove colors