Ullman's Visual Routines and Tekkotsu Sketches

15-494 Cognitive Robotics
David S. Touretzky & Ethan Tira-Thompson
Carnegie Mellon
Spring 2012
Parsing the Visual World

● How does intermediate level vision work?
  – How do we parse a scene?

● Is the x inside or outside the closed curve?
Ullman: Visual Routines

- Fixed set of composable operators.
- Wired into our brains.
- Operate on “base representations”, produce “incremental representations”.
- Can also operate on incremental representations.
# Base Representations

- Derived automatically; no decisions to make.
- Derivation is fully parallel.
  - Multiple parallel streams in the visual hierarchy.
- Describe local image properties such as color, orientation, texture, depth, motion.
- Marr's "primal sketch" and "2 ½-D Sketch"

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Viewer centred</th>
<th>Object centred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceived intensities</td>
<td>Primal Sketch</td>
<td>3-D Model Representation</td>
</tr>
<tr>
<td></td>
<td>Zero crossings, blobs, edges, bars, ends, virtual lines, groups, curves boundaries.</td>
<td>3-D models hierarchically organised in terms of surface and volumetric primitives</td>
</tr>
<tr>
<td></td>
<td>2 1/2-D Sketch</td>
<td>Local surface orientation and discontinuities in depth and in surface orientation</td>
</tr>
</tbody>
</table>
Primal Sketch

(a) input image
(b) sketch graph — configuration
(c) pixels covered by primitives
(d) remaining texture pixels
(e) texture pixels clustered
(f) reconstructed image
Incremental Representations

- Constructed by visual routines.
- Describe relationships between objects in the scene.
- Construction may be inherently sequential:
  - tracing and scanning take time
  - the output of one visual routine may be input to another
  - pipelining may speed things up
- Can't compute everything; too many combinations.
- The choice of which operations to apply will depend on the task being performed.
- What are these operations? Ullman gives 5 examples.
(1) Shift of Processing Focus

- Attentional operation
- Determines where in the image the next operation will be applied, e.g.:
  - A particular point
  - A particular contour
- There is extensive psychological and neurophysiological data on selective attention.
(2) Indexing

- "Odd man out" phenomenon
  - Easy to find the one element that differs from all the rest
  - But only if it differs in a basic property

- Indexable properties include:
  - Color, texture
  - Shape, size, orientation
  - Motion

- Indexing may provide the target for a shift of processing focus.
  - Example task: report the orientation of the red bar in a field of mostly green bars.
Triesman's Visual Search Expt.

Find the green letter:
Triesman's Visual Search Expt.

Find the O:
Triesman's Visual Search Expt.

Find the green O:
(3) Bounded Activation (Coloring)

- Mark a starting point and spread activation outward.
- Spread is blocked by “boundaries”.
- Can use this to determine inside/outside relations.
- What is the subfigure containing the dot?
Bounded Activation in Tekkotsu

• Using a Sketch<bool> as a boundary:
  - visops::seedfill(index_t point, Sketch<bool> &boundary)
  - visops::fillInterior(Sketch<bool> &boundary)
  - visops::fillExterior(Sketch<bool> &boundary)

• Using a line shape as a boundary:
  - leftHalfPlane(Shape<LineData> &line)
    also rightHalfPlane, topHalfPlane, bottomHalfPlane

• Using a polygon shape as a boundary:
  - isInside(Point p)
(4) Boundary Tracing

- Trace along the contour until some condition is met.
- Example: detect open vs. closed curves.
  - Open curves have termination.
- Does any curve contain two x's?

- Contours may not be trivial to recognize: could be broken, or implicit.
(5) Marking

- Place a marker at a location.
- Useful for remembering locations or structures that have already been examined. Are any two x's on a common curve?

![Diagram with marked points]

- Can also be used to designate a point of interest for later processing.
Points in Tekkotsu

- `fmat::Column<3>` or `fmat::Column<4>`
  - Used internally for arithmetic calculations

- Point
  - Contains an `fmat::Column<3>`
  - Also contains a `ReferenceFrameType_t`
  - Used by shapes for point arithmetic

- `EndPoint`
  - Includes `valid` and `active` booleans

- `Shape<PointData>`
Marking in Tekkotsu

• Marking a point:
  - Can use a Sketch<bool> with a single pixel set.
  - Can use a Shape<PointData>

• Marking an object:
  - Can use a Sketch<bool> to show rendering of the object.
  - Can add a shape to a SHAPEVEC
(6) Ray Tracing

- Not included in Ullman's list.
- But mentioned in an earlier section of the paper.
- Start at a point and trace outward in a straight line until you reach something of interest.
- Which way should the line go?
  - Trace in a particular direction, e.g., “upward”?
  - Trace toward an object of interest?
- Used by Agre & Chapman in Pengi.
Agre & Chapman's Pengi

An AI program that plays the Pengo video game:

See videos of the original Pengo arcade game on YouTube.
Finding *the-block-that-the-block-I-just-kicked-will-collide-with* using ray tracing and dropping a marker.
Visual Routines in Pengi

Finding *the-block-to-kick-at-the-bee* when lurking behind a wall.
Visual Routines in Game AI

• Forbus et al.: visual routines could be used for qualitative spatial reasoning, such as path finding in AI strategy games.

• Example: Voronoi diagram of open space on a map can be used for route finding.

\[ VDdiag(a) = edge(read(labelcc(a), link(a))) \]
Application to Tekkotsu?

- Can create sketch spaces for local or world maps.
- `setTmat(scale,tx,ty)` controls the mapping of shape space coordinates to sketch space pixels.
- `getRendering()` converts shapes to sketches.
- Marking and coloring can be implemented using sketches.
- Might use this to implement Pengi-like logic for robotics applications.
- But we need more primitives...
Do Tekkotsu's Representations Fit Ullman's Theory?

- What are the base representations?
  - color segmented image: sketchFromSeg()
  - intensity image: sketchFromRawY()
  - depth image: sketchFromDepth()
  - extracted regions

- What are the incremental representations?
  - Sketches
  - Shapes

- What's missing?
  - Attentional focus; boundary completion; lots more.
What Do Human Limitations Tell Us About Cognition?

• Subjects can't do parallel visual search based on the intersection of two properties (Triesman).

• This tells us something about the architecture of the visual system, and the capacity limitations of the Visual Routines Processor.
  – Base can't do intersection.
  – VRP can't process whole image at once.
  – There must be a limited channel between base and VRP.

• But in Tekkotsu, we can easily compute intersections of properties.
  – Is that a problem?
Science vs. Engineering

• Science: figure out how nature works.
  – Limitations of a model are good if they suggest that the model's structure reflects reality.
  – Limitations should lead to nontrivial predictions about comparable effects in humans or animals.

• Engineering: figure out how to make useful stuff.
  – Limitations aren't desirable.
  – Making a system “more like the brain” doesn't in itself make it better.

• What is Tekkotsu trying to do?
  – Find good ways to program robots, drawing inspiration from ideas in cognitive science.