Hierarchical, multi-feature visual grouping processes for ACT-R

Patrick Rice
Department of Psychological Sciences
Rice University
Houston, TX
pjr@rice.edu
VEGA - Visicon Entry Grouping Algorithm

- Presented at 2018 ACT-R workshop by John Lindstedt, now at SUNY Oswego
- Motivated by observation of behavioral error associated with certain ballot designs
- Two core concepts:
  - Collision methods
    - Parameterized by a radius value
  - Agglomeration

[Diagram showing point and box collisions with steps 1 to 5 for the algorithm]
How does it work?
  - Code that handles grouping is implemented in python rather than lisp
  - Relies on a set of command monitors which track changes to the visicon and the (proc-display) call
    - :force-visual-commands is necessary

<table>
<thead>
<tr>
<th>ACT-R</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Device code modifies visicon</td>
<td>VEGA maintains a record of the visicon</td>
</tr>
<tr>
<td>2. Visicon modification results in (proc-display) call</td>
<td>VEGA groups visicon entries before (proc-display) call executes</td>
</tr>
<tr>
<td>Name</td>
<td>Att</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
<tr>
<td>VITALITY</td>
<td>16</td>
</tr>
</tbody>
</table>
For Partisan Office and Referendum

Notice to Electors:
This ballot may be invalid unless initialed by 2 election inspectors. In cast as an absentee ballot, the ballot must bear the initials of the municipal clerk or deputy clerk.

To vote for the candidate of your choice, blacken the oval to the left of the candidate's name. To vote for a person whose name does not appear on the ballot, write the person's name on the line provided and blacken the oval to the left of the line.

Straight Party
If you desire to vote for a straight party ticket for all state, congressional, legislative, and county offices, blacken the oval to the left of the party of your choice. A straight party vote cannot be cast for independent candidates of your choice.

Important:
Blacken the oval to the left of the name of the candidate. When voting for governor and lieutenant governor, you may vote only for the candidate on their joint ticket.
VEGA - 2023 edition

- VEGA 2023 does everything that the 2018 version did
  - Plus bug fixes!
- How has it been extended?
  - Hierarchical grouping
  - Iterative Radii

<table>
<thead>
<tr>
<th>Radius value</th>
<th># Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>


VEGA - 2023 edition

- What does it do?
  - Everything that the 2018 version did
  - Hierarchical grouping
    - Iterative Radii
      - “Top-down”/“bottom-up”
      - Naturally identify number of levels of hierarchy
      - Difficult to tell what belongs to what
      - Manual setting of radius values
      - Relatively computationally intensive

<table>
<thead>
<tr>
<th>Radius value</th>
<th># Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>
VEGA - 2023 edition

- What does it do?
  - Everything that the 2018 version did
  - Hierarchical grouping
    - Iterative Radii
    - Add Group Features
VEGA - 2023 edition

- What does it do?
  - Everything that the 2018 version did
  - Hierarchical grouping
    - Iterative Radii
    - Add Group Features
      - “Bottom-up” only
      - Naturally identify number of levels of hierarchy
      - More straightforward to interpret groups
      - Radius driven by object/group positioning
      - Less computationally intense
What does it do?
- Everything that the 2018 version did
- Hierarchical grouping
  - Iterative Radii
  - Add Group Features
- New collision method
VEGA - 2023 edition

• What does it do?
  ○ Everything that the 2018 version did
  ○ Hierarchical grouping
    ■ Iterative Radii
    ■ Add Group Features
  ○ New collision method
    ■ Consideration of multiple attributes of visual objects
      ● Proximity
      ● Alignment
        ○ Centroids on vertical/horizontal axes
        ○ Edges
      ● Difference in area
      ● “Identity”
So far, have performed a set of three experiments to test and validate the implemented grouping theory.

- **Experiment 1**
  - Participants given a set of visual scenes, and asked to indicate perceived groups by hand. Instructions prompt participants to signify hierarchy in group perception.

- **Experiment 2:**
  - Participants are show a series of visual scenes. For each scene, they are first asked to consider their perception of any potential visual groups, and then select from a set of four different groupings of that scene.

- **Experiment 3:**
  - A refinement of experiment 2, where the stimuli are varied more consistently, and the response set is limited to three options.
Experiments

Frequency of choice selection for each row-spread stimulus, across participants

selected_choice

freq

cols rows zrcrc
Looking forward

- Refinement of hierarchical grouping processes and multi-feature collision
  - Interaction between the two? (expt 1)
- Temporal inheritance of grouping information
- Self-contained module as an addon to ACT-R
- Not just visual grouping, but perceptual grouping
  - Grouping in other senses
    - Audicon
  - Cross-modal grouping
    - Co-occurring stimuli
      - Visuo-auditory temporal delay
  - Integration of ACT-R modules
Thanks!

- Mike Byrne
- Xianni Wang, Fabrizio Chavez, Teddy Taffese, Charlie Weeks
- Linda Mao, Ariana Zhu, Emily Wu
- CHIL Lab!
VEGA - Visual Entity Grouping Algorithm

- Introduced at 2018 ACT-R workshop
  - John Lindstedt, now at SUNY Oswego
- Motivated by observation of behavioral error associated with certain ballot designs
VEGA - Visual Entity Grouping Algorithm

- How did it work?
  - Two core concepts:
    - Collision methods
VEGA - Visual Entity Grouping Algorithm

- How did it work?
  - Two core concepts:
    - Collision methods
      - Parameterized by a radius value
How did it work?

Two core concepts:

- **Collision methods**
  - Parameterized by a *radius* value
- **Agglomeration**

1. Take an unexamined point from the scene and add it to the current group.
2. Add any other unexamined points within the grouping radius to the current group.
3. Repeat step 2 for each new point added, growing the group.
4. Until no unexamined points remain within the radius of any group members.
5. Repeat steps 1-4 until all points in the scene have been assigned a group.
VEGA - Visual Entity Grouping Algorithm

- What could it do?
What could it do?
VEGA - Visual Entity Grouping Algorithm

- What could it do?
● What couldn’t it do?
● What couldn’t it do?
  ○ Clear distinctions between objects in close proximity
    ■ Just select “best” radius value
● What couldn’t it do?
  ○ Clear distinctions between objects in close proximity
    ■ Just select “best” radius value
  ○ Hierarchical groups
VEGA - Visual Entity Grouping Algorithm

- What couldn’t it do?
  - Clear distinctions between objects in close proximity
    - Just select “best” radius value
  - Hierarchical groups
  - ...work!
VEGA - Visual Entity Grouping Algorithm

- How did it break?
  - Implemented as an around method for (build-vis-locs-for), which was deprecated with the API changes to support the RPC interface (version 7.11?)