Midpoint Design Discussion

COLOR PALETTE GENERATION FOR NOMINAL ENCODINGS

Ketrina Yim, Simon Tan, Calvin Ardi
CS 294-10 | Fall 2008
Q: How can we produce unique, quality palettes with minimal human intervention?

A: Algorithmically (simulated annealing) using Brewer’s Color Use Guidelines as heuristics for “good palettes”

Added contributions:
- Colorblind-friendly palettes
- Color-harmonious (“themed”) palettes
- Extension of prefuse / flare
<table>
<thead>
<tr>
<th>Task</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic/evaluation function</td>
<td>Currently involves contrast, closeness of neighboring values</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>Implemented and functional with placeholder functions</td>
</tr>
<tr>
<td>User Interface</td>
<td>Overall look mostly implemented</td>
</tr>
</tbody>
</table>
Prototype

● DEMO!

Palette Assistant

Enter Palette Size (coming soon)

Choose Palette Preference:
- None
- Warm Colors
- Cool Colors

Choose Colorblindness Compatibility:
- None
- Red-Green Colorblindness
- Blue-Yellow Colorblindness

Generate
Evolution

- Definition of ‘colorblind’ discovered to have two dimensions
  - Red-green
  - Blue-yellow
- HSV believed to be best way to iterate through color space
- Put off to future work: perhaps use machine/supervised learning for computing weights in our heuristic
Questions

- Should we put work in the starting palette or let the heuristic function do all the work?
- How to display the list of generated colors when it grows too large?
Thank You

Questions?