Content Creation
Last time

Successful programming tools shift our cognitive problem representations to make the task more readily solvable.

Tools for learning programming help externalize our cognition to better understand what code is doing (or ought to be doing).

Programming tools often either aim to reduce the threshold or increase the ceiling — how depends on which one we’re pursuing.

Threshold: Difficulty to use (semantic distance, often in gulf of execution, sometimes in gulf of evaluation)

Ceiling: Sophistication of what can be created (higher expressivity)
Creativity Support Tools
Every tool supports creativity

Is there a tool that does not support creativity?

We can use any tool in creative ways
Content creation tools

Content comes in many forms (illustration, film, animation, photographs, podcasts, screenplays, …)

The workflows content creators use differ by content.

How might we develop tools to facilitate creation workflows while ensuring that the user feels in control of the creative decisions?
Today

Design principles for visual communication

Digital Illustration

Video and Audio

Instructions and Exploded Views
Design principles for visual communication
Design Principles

Users’ task:
Understand how to get from point A to point B

Important information:
Sequence of stops/interchanges along each line

Design principles:
Straighten lines & evenly space stops to emphasize sequence
De-emphasize geographic shape of subway lines

Techniques used to emphasize/de-emphasize information
Cognition of Route Maps

[Tversky 1992, 1999]

Essential information
Turning points
Route topology

Secondary context information
Local landmarks, cross streets, etc.
Overview area landmarks, global shape

Exact geometry less important
Not apprehended accurately
Not drawn accurately
Design Principles

1. Exaggerate road length
2. Regularize turning angles
3. Simplify road shape
From principles to algorithms

[Agrawala and Stolte 2001]
From principles to algorithms

[Agrawala and Stolte 2001]

Techniques to:

Simplify the shapes in the original route map

Grow short roads to emphasize them

Layout graphic elements (e.g., roads, labels) by stochastically searching over possible visual attributes (e.g., position, orientation, size)

Evaluate/score layout based on alignment with design principles

Algorithm: simulated annealing — a “try, score, and perturb” loop
From principles to algorithms

[Agrawala and Stolte 2001]

Road layout

Labels
Design principles for visual communication

[Agrawala, Li, and Berthouzoz 2011]

Step 1: Identify design principles

Analyze most effective visualizations within domain (consider user's task) and look for techniques they frequently use to emphasize/de-emphasize information.

Examine prior work in cognitive psychology that has considered how people understand a domain to determine information that is important/unimportant to task.

Perform new experiments in perception or cognition to determine information that is important/unimportant to task.
Design principles for visual communication

[Agrawala, Li, and Berthouzoz 2011]

Step 2: Instantiate design principles

- Encode design principles into algorithms and interfaces
- Constrained optimization, controls that match the user's mental models

Step 3: Evaluate/validate design principles

- Measure improvements in task performance, quality of results, etc.
Digital Illustration
Draco: kinetic textures

[Habib et al. 2014]
From principles to design

Via an inductive study of animations on YouTube and interviews with animators, found that common approaches:

- Particle systems, flocking behavior, and stochastic motion

**System goal:** author these effects without a technical background
Wow! @autodesk Sketchbook Motion (AKA @rubaiat et al, Draco, CHI 2014), was chosen by Apple as iPad App of the Year. sketchbook.com/motion
Visual blends

[Chilton, Petridis, and Agrawala 2019]

Combinations of visual concepts, suggested by algorithm

Design principles:

Two concepts, two objects, integrated into one object

Retain the most salient visual signals (semiotics) of each object
Sketchpad
[Sutherland 1962]
First use of light pen
First use of GUI
windows
Rubberband lines
Constraint-based
drawing
Obj. oriented
Sketchpad
[Sutherland 1962]
First use of light pen
First use of GUI windows
Rubberband lines
Constraint-based drawing
Obj. oriented
Video and Audio
Frame-based representation forces users to navigate and edit video by timecode.
Design principle:
For dialogue-heavy video, editors think in terms of the words being spoken (the transcript)

Interface should:
Empower editors to directly edit scripts, not video, and smooth the cuts automatically

Algorithm:
frame similarity graphs
Design principle:

For dialogue-heavy video, editors think in terms of the words being spoken (the transcript)

Interface should:

Empower editors to directly edit scripts, not video, and smooth the cuts automatically

Algorithm:

frame similarity graphs

Jump cuts (in red)
Design principle: For dialogue-heavy video, editors think in terms of the words being spoken (the transcript).

Interface should: Empower editors to directly edit scripts, not video, and smooth the cuts automatically.

Algorithm: frame similarity graphs

Our result: hidden transitions in blue, pauses in green

Comp. time:
- clusters: 22m
- hidden: 5s
- pauses: 9s

[Berthouzoz, Li, Agrawala 2012]
On good sound systems you can do basic editing, but nothing too risky. And there’s lots of good software that lets you edit sounds on a normal home computer (see page 199).
Example musical underlay
from This American Life #441: "When Patents Attack!"

Underscore
[Rubin et al., CHI 2014]
"Great Expectations"
Charles Dickens

Underscore
[Rubin et al., CHI 2014]
"Tenuous Gears"
Damiak

Underscore
[Rubin et al., CHI 2014]
Underscore

[ Rubin et al., CHI 2014]
Story: Charles Dickens – "Great Expectations"
Read by Mark F. Smith [librivox.org]

Music: Damiak – "Tenuous Gears"

Underscore
[Rubin et al., CHI 2014]
**Story:** David Sedaris – “Go Carolina”  
Read by David Sedaris [Hachette Audio, 2001]  
**Music:** El Chicano – “Viva Tirado Pt. 1”  

**Underscore**  
[Rubin et al., CHI 2014]
Instructions and Exploded Views
Assembly instructions

[Agrawala et al. 2003] [Heiser et al. 2004]

Design principles:

Depict subassemblies first, then combine those subassemblies together.

Annotations and step-by-step diagrams highlight changes.

All changes in a given step must be in plain view, while keeping the viewpoint static when possible.
Assembly instructions

[Agrawala et al. 2003]
Exploded view diagrams

[Li et al. 2008]

Design principles:

Explode parts in directions that do not occlude (block) other parts, while minimizing distance from their original position.

When parts are nested inside a container, explode out from the center of the container.
Exploded view diagrams

[Li et al. 2008]

Algorithmically generated diagrams:
Interactive Exploded Views

[Li et al. 2004]

Design Principles:

Clarify spatial relationships
- Direct manipulation [Shneiderman 83]
- Animated transitions [Woods 84] [Robertson 91] [Grossman 01]

Reduce visual clutter
- Interactive filtering [Shneiderman 96] [MacEachren 97]
- Highlight most important information [Tufte 83] [MacEachren 97]
Authoring Pipeline

<table>
<thead>
<tr>
<th>Input</th>
<th>Segment</th>
<th>Stack</th>
<th>Fragment</th>
<th>Assign ordering</th>
<th>Annotate</th>
</tr>
</thead>
</table>

Diagram showing stages of the Authoring Pipeline.
Interactive Viewing
Summary

Design principles provide strong guides for content creation tools: (1) **identify design principles** in expert output based on **cognition/perception**, and (2) **instantiate them into algorithms** to aid content creators, and (3) **evaluate principles** through user studies.

Approach generalizes across a wide range of categories, ranging from digital illustration to audio, video, instructions and exploded views.
References


References


Sutherland, Ivan E. "Sketch pad a man-machine graphical communication system." Proceedings of the SHARE design automation workshop. 1964.
