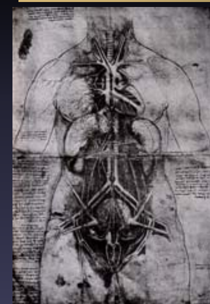
Automating the Design of Visualizations

Maneesh Agrawala

August 2, 2002

Stanford University

Visualization: Explore & Present Data



Principal Organs & Vascular System
[Leonardo da Vinci ca. 1490]

Strange Immersion of Torus in
3-Space [Curtis 92]

Good Design Improves Usability

London Underground [Beck 33]

Geographic Version of Map

- Visualizations are common
  - Newspapers, textbooks, training manuals, scientific papers, …
- Creating effective designs is time-consuming

Challenge

- Best visualizations are designed by humans
- Computing becoming ubiquitous
  - Data collection / dissemination getting faster
  - Most displays computer generated
- Therefore: Visualizations are regressing
- **Can we build automated systems capable of designing effective visualizations?**

Automation Allows Customization

- **Purpose:** Present data relevant to specific goals
- **Device:** Adapt to capabilities of display
- **Situation:** Update as data / goals change
- **Person:** Adapt to knowledge of user
- Customization increases effectiveness

Emulating Artistic Rendering Styles

- Artistic rendering can improve perception

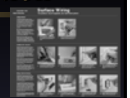
Mouse [Lum & Ma 02]

House [Winkenbach & Salesin 94]

- **High-level design still specified manually**

Automated Design as Optimization

- Page design
  - TeX [Knuth 81], GRIDS [Feiner 88], LayLab [Graf 92], [Weitzman & Wittenburg 94], [Borning et al. 97, 00]



- 3D object visualization
  - APEX [Feiner 85], IBIS [Seligmann & Feiner 91], WIP [Rist et al. 94]

- Data graphics presentations
  - APT [Mackinlay 86], SAGE [Roth et al. 94, 96], SYSTAT [Wilkinson 99]

- UI layout, Label layout, VLSI design, Camera planning, 2D/3D packing, Graph drawing, …
- **Need domain specific constraints**

Contributions

- **Analysis**
  - Identify design principles
    - Route maps
    - Assembly instructions
- **Synthesis**
  - Automated design systems

Outline

- Motivation
- Automated Route Map Design
- Framework for Automated Design
- Automated Assembly Instruction Design
- Future Directions

Visualizing Routes

- Standard online route maps difficult to use

A Better Visualization

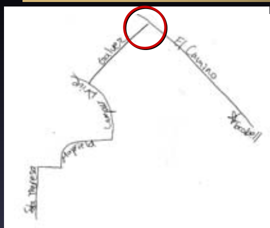
- Hand-drawn maps much easier to use

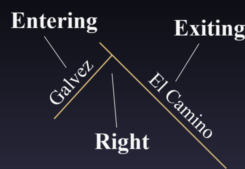
Communicative Intent of Route Maps

- Route is a sequence of turns [Tversky 92] [MacEachren 95]
  1. Start at 100 Serra
  2. Turn Right on University
  3. Turn Left on El Camino
  4. Turn Right on San Antonio

  …
  - Verbal directions emphasize turns [Denis 97]
  - Hand-drawn maps highlight turns [Tversky & Lee 99]
- **Maps must communicate turning points**

Depicting Turns





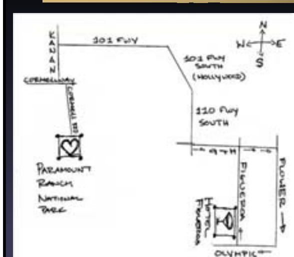
- Pair of roads (entering / exiting the turn)
- Turn direction (left / right)
- **These graphic elements must be visible**

Context can Facilitate Navigation

- Local context
  - Consistency checks
    - Cross-streets
    - Landmarks along route
    - Distance along each road
- Overview context
  - Orient route to geography
    - Large area landmarks
    - Overall shape & heading

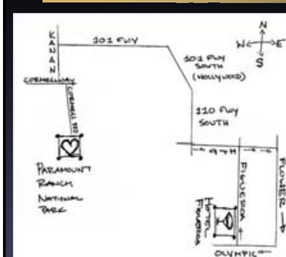
- **Context is secondary to turning points**

Geometric Properties Distorted



- Geometry *not* apprehended accurately [Tversky 81]
- Geometry *not* drawn accurately [Tversky & Lee 99]
  - Topology is accurate

LineDrive: Route Map Design System



Hand-drawn Route Map

LineDrive Route Map

Automating Route Map Design

- Layout problem
  - Set of graphic elements
    - Roads
    - Labels
    - Cross-streets
  - Choose visual attributes
    - Position
    - Orientation
    - Size
- Distortions increase choices
- Large space of possible layouts

Layout as Search-Based Optimization

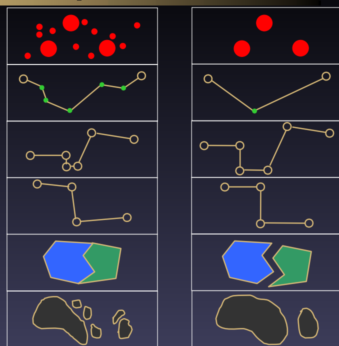
- Hard constraints
  - Required characteristics
- Soft constraints
  - Desired characteristics

Space of possible layouts
quality of layouts / layout space

- **Challenge: Develop relevant constraints**
- Simulated annealing
  - *Perturb:* Form a layout
  - *Score:* Evaluate quality
  - Minimize score

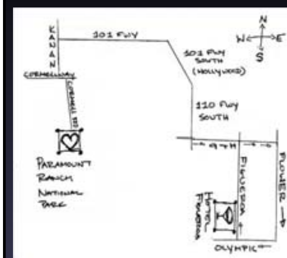
Cartographic Generalization

- Selection
- Simplification
- Exaggeration
- Regularization
- Displacement
- Aggregation



[Monmonier 96], [MacEachren 94], [DiBiase 91]

Three Generalizations for Route Maps



- Our observations from hand-drawn examples:
  - Exaggeration
    - Road length
  - Regularization
    - Turning angle
  - Simplification
    - Road shape
- **Generalizations emphasize turning points!**

Exaggeration: Length Generalization

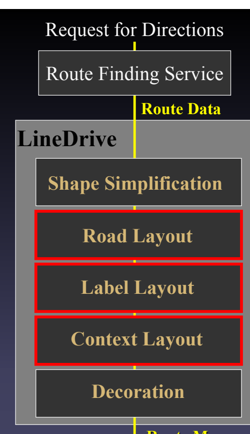
- Grow short roads, shrink long roads
  - Ensures all roads visible
  - Maintain relative ordering by length

Regularization: Angle Generalization

- Regularize turning angles
  - Reduces visual complexity
  - Maintain consistent turn direction

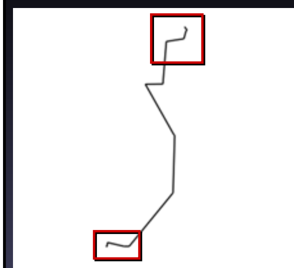
Simplification: Shape Generalization

- Simplify roads to straight lines
  - Differentiates roads and turning points
  - Maintain overall shape of route

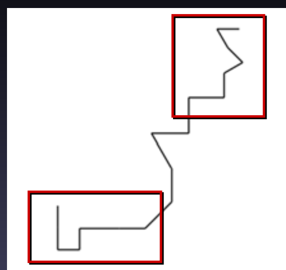


Stage 2: Road Layout

- Goal: Choose road lengths & orientations



without road layout



with road layout

Road Layout Search

- Initialize
  - Uniformly scale route to fit given viewport
- Perturb
  - Pick random road
  - Either
    - Rescale by random factor
    - Reorient by random angle
  - Rescale entire route to fit viewport
- Hard Constraints
  - Must fit in viewport
  - Must maintain consistent turn direction

Designing Soft Constraints

- Challenges
  - Choose desirable characteristics
  - Express as numerical score function
  - Balance constraints, deal with conflicts
- Desired characteristics for road layout
  - All roads visible
  - Prevent excessive distortion

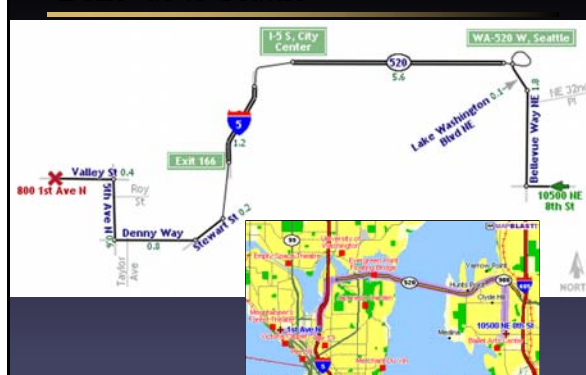
Constraints

- **Length**
  - Ensure all roads visible $((L_{min} - l(r_i))/L_{min})^2 * W_{small}$
  - Maintain ordering by length $W_{shuffle}$
- **Orientation**
  - Maintain original orientation $|\alpha_{curr}(r_i) - \alpha_{orig}(r_i)| * W_{orient}$
- **Topological errors**
  - Prevent false $\min(d_{origin}, d_{dest}) * W_{false}$
  - Prevent missing $d * W_{missing}$
  - Ensure separation $\min(d_{ext}, E) * W_{ext}$
- **Overall route shape**
  - Maintain endpoint direction $|\alpha_{curr}(v) - \alpha_{orig}(v)| * W_{enddir}$
  - Maintain endpoint distance $|d_{curr}(v) - d_{orig}(v)| * W_{enddist}$

Balancing Soft Constraints

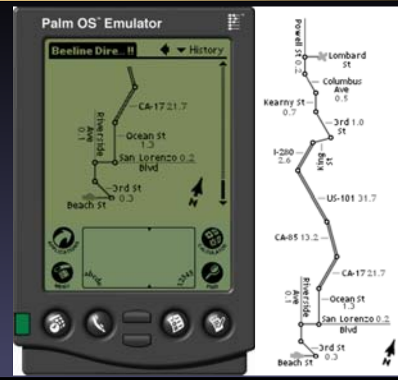
- Prioritize scores by importance
  1. Prevent topological errors
  2. Ensure all roads visible
  3. Maintain original orientation
  4. Maintain ordering by length
  5. Maintain overall route shape
- Informal usability engineering
  - Consider maps containing errors
  - Rate which errors most confusing

Bellevue to Seattle



Cross-Country Route

Limited Resolution: Palm



User Response

- Beta publicly accessible Oct 00 – Mar 01
- 150,000 maps served
- 2242 voluntary responses
  - Should replace standard maps 55.6 %
  - Use along with standard maps 43.5 %
  - Standard maps preferable 0.9 %
- Most common suggestion
  - Choose better routes (not a LineDrive issue)
  - More context in unfamiliar areas

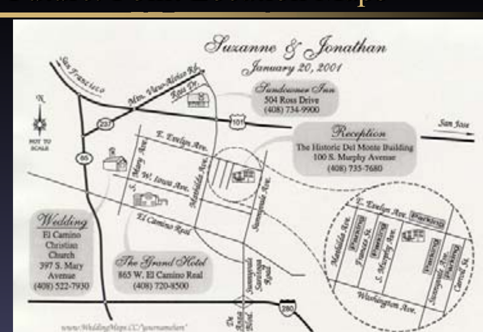
Current Status

- Default rendering style
  www.mapblast.com
- 250,000 maps/day

Next Steps

- Map enhancements
  - Cross-street after turning point
  - Large area landmarks
- In-depth user study
  - Watch users following LineDrive maps

Future: Point Location Maps



Hand-designed Wedding Map [www.WeddingMaps.CC 01]

Outline

- Motivation
- Automated Route Map Design
- Framework for Automated Design
- Automated Assembly Instruction Design
- Future Directions

Two-Step Approach

Analyze cognitive science research and examples of most effective hand-designed visualizations

↓

Step 1: Identify visualization design principles

↓

Low-level visualization design principles

↓

Step 2: Encode principles as constraints and algorithmically find design satisfying constraints

↓

Automated design system

Step 1: Identify Design Principles

- Cognitive science
  - How people *conceive* information
  - How people *apprehend* visual representations
- **Conception**
  - Routes conceived as sequence of turns
- **Apprehension**
  - Route geometry not apprehended accurately
- **High-level cognitive model**

Step 1: Identify Design Principles

- Analyze hand-designed visualizations
  - Identify essential graphic elements
  - Identify distortion techniques

- **Low-level visualization design principles**

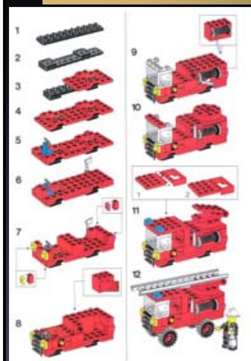
Step 2: Build Automated Algorithm

- Space of possible visualization designs
  - Graphic elements
  - Visual attributes
- Design principles → Constraints
  - Generative rules: How to vary visual attributes
  - Evaluation criteria: Measure effectiveness
  - **Main algorithmic challenge**
- Find most effective visualization design
  - Search-based optimization
  - Balance constraints
  - Efficiency

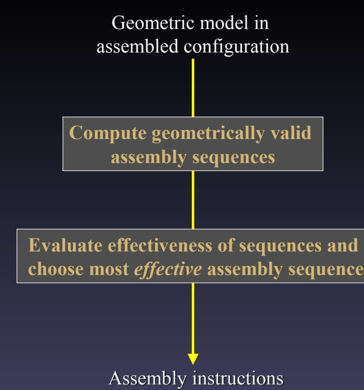
Outline

- Motivation
- Automated Route Map Design
- Framework for Automated Design
- Automated Assembly Instruction Design
- Future Directions

Assembly Instructions

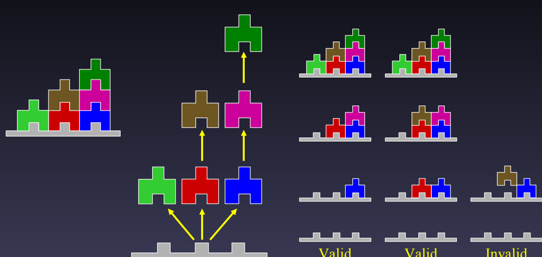


Goal: Create step-by-step instructions from 3D model



Geometrically Valid Sequences

- Robotics / Mechanical Engineering
  [DeFazio & Whitney 87] [Wolter 89] [Wilson 95] [Romney et al. 95]



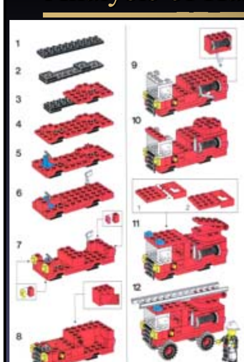
Many Geometrically Valid Sequences

- **How do we choose most effective sequence?**

Cognitive Science

- Experiments to learn how people understand assembly instructions [Heiser in progress]
- **Assemblies conceived as groupings of parts**
  - Coarse level - functional units
  - Finer levels - symmetry, similarity, proximity
- **People prefer certain assembly sequences**
  - Add *all* supporting parts then supported parts
  - Add *all* internal parts then external parts
  - Add grouped parts in same step, or in sequence
  - Add new parts onto existing parts

Analysis of Hand-Designed Examples

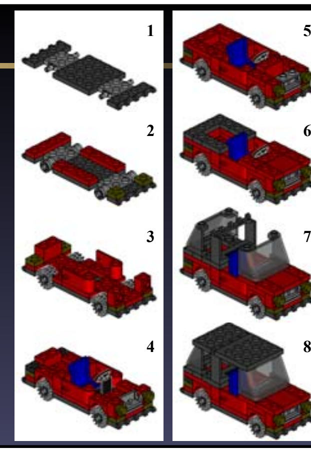


- **Essential graphic elements**
  - Parts added in step (visibility)
  - Previous parts (context)
- **Graphic design techniques**
  - Small multiples
  - Technical illustration style
  - Insets improve part visibility
  - Arrows show attachments

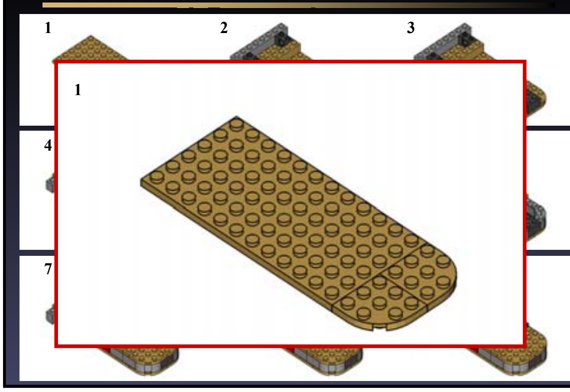
Constraints

- **Support**: All supporting parts added before supported
- **Adjacency**: All parts in step touch previous parts
- **Symmetry**: All symmetric parts added in same step
- **Linearity**: New parts added onto existing parts
- **Visibility**: If part *A* occludes *B*
  Penalty = Occlusion (A, B) * $W_{visibility}$
- **Context**: If < 25% of step *N-1* parts visible
  Penalty = Occlusion (Step N, Step N-1) * $W_{context}$

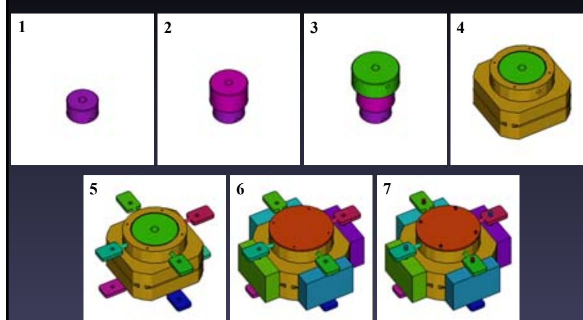
Lego Car



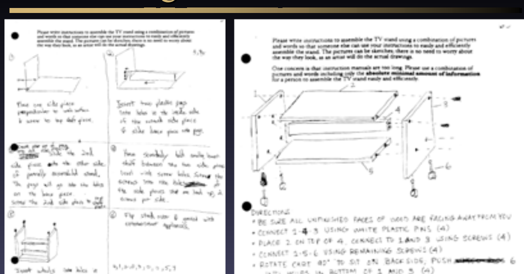
Landspeeder



Mechanical Assembly



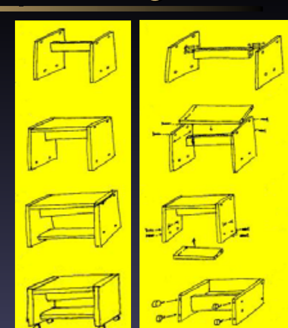
Assembling TV Stand



- Subjects assemble TV stand without instructions
- Then asked to produce clear set of assembly instructions

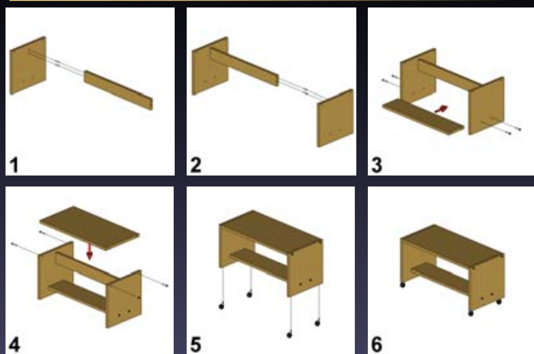
Analysis of Hand-Drawn Diagrams

- **Static:** Show object after each assembly step
- **Action:** Show operations required in each step
  - Emphasize new parts
  - Show motion of parts
  - Show alignment of parts
  - Show how fasteners attach parts



Static Diagrams Action Diagrams

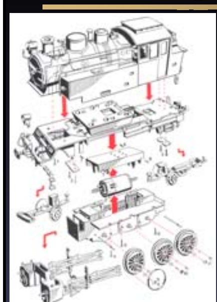
Computer-Generated Instructions



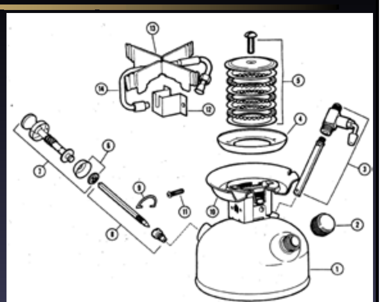
Current Agenda

- Identify more design principles
- Incorporate other graphic design techniques
  - Insets
  - Scale exaggeration
  - Cutaways
  - Sections
  - Text labels
- User studies

Future: Exploded Views

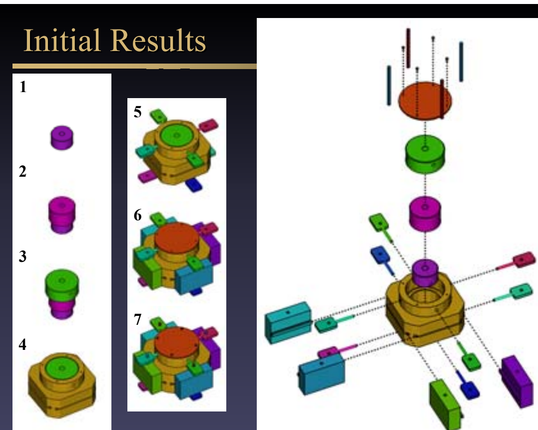


Train [from Mijksenaar 99]

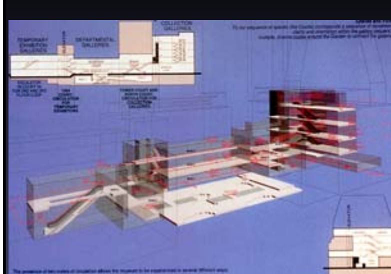


Camping Stove [from Mijksenaar 99]

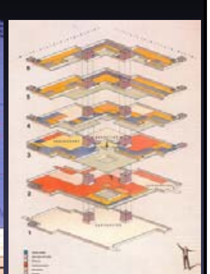
Initial Results



Future: 3D Environments



MoMA Design Entry [Tschumi 99]



IBM Building Plan [from Holmes 93]

Summary

- General two-step approach
  - Step 1: Identify cognitive design principles
  - Step 2: Encode principles as constraints and find most effective visualization
- Automated design systems
  - Route maps
  - Assembly instructions
- Benefits
  - Novices can leverage skills of experts
  - Deal with data overload

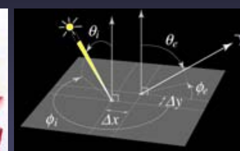
Outline

- Motivation
- Automated Route Map Design
- Framework for Automated Design
- Automated Assembly Instruction Design
- Future Directions

Many Other Domains To Consider

- Medical illustration: Complex biological organisms
- Scientific diagrams: Depict scientific concept
- Graphs and charts: Scatter plots, bar charts, etc.
- Architectural plans: Room and furniture layout
- Proof visualization: Depict complex logical statements


Medical Illustration


Scientific Diagram


Graphs and Charts

Interaction and Animation

- Interaction
  - Hide clutter, let user request details
  - Direct, intuitive, navigation controls
- Animation
  - Should add information [Hegarty 00] [Morrison 01]


Long-Term Challenge

- Current focus on *how*
  - Simulate realistic lighting, shading
  - Emulate artistic media (paint, pen & ink, …)
  - Display data using std. metaphors (bar graph, binary tree, …)
  - …
- Need principles guiding *where, what, why*
  - Where to place lights to communicate a mood?
  - What information does an artistic rendering style convey?
  - Why is a particular metaphor effective?
  - …
- **Must understand and appreciate what makes an effective visualization**

Acknowledgements

- Pat Hanrahan
- Chris Stolte
- Barbara Tversky
- Boris Yamrom
- Vicinity Corporation