Interactive Visual Data Analysis

09 – Models and concepts of interaction

Objectives

- Why is interaction important for visual data analysis: Learn interaction intents and action patterns
- How does interaction work conceptually: Learn basic principles of human-computer interaction
- What makes good (and bad) interaction: Learn about requirements and guidelines for interaction
- What are basic interactions for interactive visual data analysis: Learn the fundamental interaction functionality required

Overview

- Human in the loop
  - Interaction intents and actions patterns
  - The action cycle
- Requirements
  - Interaction costs
  - Directness of interaction
  - Design guidelines
- Basic operations
  - Taking action
  - Generating feedback

Human in the loop

Why is interaction with the user necessary?

Clear problem, answer can be computed, do:

Computation

Ill-defined problem, answer can hardly be computed, add:

Visualization

If single computation and visual representation do not suffice, add:

Interaction

Many real-world problems are here!

Human in the loop

Why is interaction with the user necessary?

- Think about: What computers are good at?
  - Produce crisp results
  - Memory exact rather than approximate
  - Repetitive computations without getting tired
  - …
- Think about: What humans are good at?
  - Creativity
  - Complex reasoning
  - React to the unexpected
  - …

→ Need an interplay between human and computer, the “human in the loop”!

Human in the loop

Why is interaction with the user necessary?

- Visualization allows us to **see** things that we could otherwise not see
- Interaction allows us to **do** things that we could otherwise not do

> “A graphic is not ‘drawn’ once and for all; it is **‘constructed’ and reconstructed** until it reveals **all the relationships** constituted by the **interplay of the data**. The best graphic operations are those **carried out by the decision-maker** himself.
>
> — Bertin, 1981

> “While **visual representations** may **provoke curiosity**, **interaction** provides the means **to satisfy it**.
>
> — Tominski & Schumann, 2020

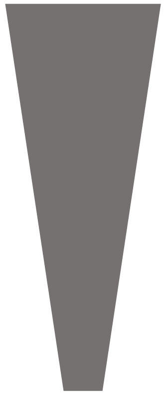
Human in the loop

Why is interaction with the user necessary?

- Required degree of interactivity varies depending on the pursued goal

Goals
- Exploration
- Description
- Explanation
- Confirmation
- Presentation

High degree



Low degree

Interaction intents and action patterns

- Obviously interaction is necessary and helpful
- But what actuates users to interact and what interactions are common?

- **High-level interaction intents**
- **More fine-granular list of action patterns**

Interaction intents and action patterns

Interaction intents (Yi et al., 2007)

- Capture *why* users interact
- **Seven main categories** of interaction intents
  - *Mark* something as interesting
  - *Show me* something else
  - *Show me* a different arrangement
  - *Show me* a different representation
  - *Show me* more or less detail
  - *Show me* something conditionally
  - *Show me* related things

Interaction intents and action patterns

Interaction intents (Yi et al., 2007)

- *Mark* something as interesting
  - Mark interesting parts in the data
  - Transient (short-term) vs. permanent (long-term) marking
- *Show me* something else
  - Usually, data only partially visible
  - Explore different parts of the data
- *Show me* a different arrangement
  - Arrange data differently to obtain different insights
  - Example: Arrange data wrt. time T or space S or attributes A

Interaction intents and action patterns

Interaction intents (Yi et al., 2007)

- *Show me* a different representation
  - Adapt the visual encoding for exploration, verification, and presentation
  - Example: Adjust color scale according to data and task
- *Show me* more or less detail
  - Get from overview to details
  - Balance conflicting demands of studying subtleties and seeing the big picture
- *Show me* something conditionally
  - Show only those data that adhere to certain conditions or search criteria
  - Dynamically filter out or attenuate irrelevant data to clear view on relevant data

Interaction intents and action patterns

Interaction intents (Yi et al., 2007)

- *Show me* related things
  - Based on already made observations, find similar or related parts of the data
  - Find, compare, and evaluate relations in the data

Interaction intents and action patterns

Interaction intents (Yi et al., 2007)

- **Two additional categories** of interaction intents
  - *Let me* go back to where I’ve been
    - Support exploratory workflows
    - Return to previous states of the data analysis
  - *Let me* change the interface
    - Adjust not only the visualization, but also the overall data analysis system
    - Configure the user interface and manage system resources

- *Intents* capture reasons for active participation of users, next we look at more concrete *action patterns*

Interaction intents and action patterns

Action patterns ([Sedig & Parsons, 2013](#))

- Capture *what* users actually do when interacting
- Two types of action patterns can be distinguished
  - **Unipolar**
    - Actions only performed in one direction
    - No natural opposite action
    - Can be reversed only with a generic *undo* action
  - **Bipolar**
    - Pairs of actions
    - One action is natural opposite of other action

Interaction intents and action patterns

Action patterns (Sedig & Parsons, 2013)

- **Unipolar**

Pattern	Description
Arranging	changes ordering, either spatially or temporally
Assigning	binds features or values to be encoded
Blending	fuses visual representations together to form one entity
Comparing	determines similarities or differences
Drilling	brings out and displays interior, deep information
Filtering	displays subsets obeying certain criteria
Navigating	moves on, through, and around the data
Selecting	focuses on or chooses either individuals or groups

Interaction intents and action patterns

Action patterns (Sedig & Parsons, 2013)

- **Bipolar**

Pattern	Description
Collapsing/ Expanding	fold in and compact visual items, or oppositely, fold them out or make them more diffuse
Composing/ Decomposing	assemble and join together to create holistic representations, or oppositely, break up into separate components
Linking/ Unlinking	establish relationships or associations, or oppositely, dissociate and disconnect relationships
Storing/ Retrieving	put aside for later use, or oppositely, bring stored items back into usage

- Think about: What other bipolar action patterns exist?

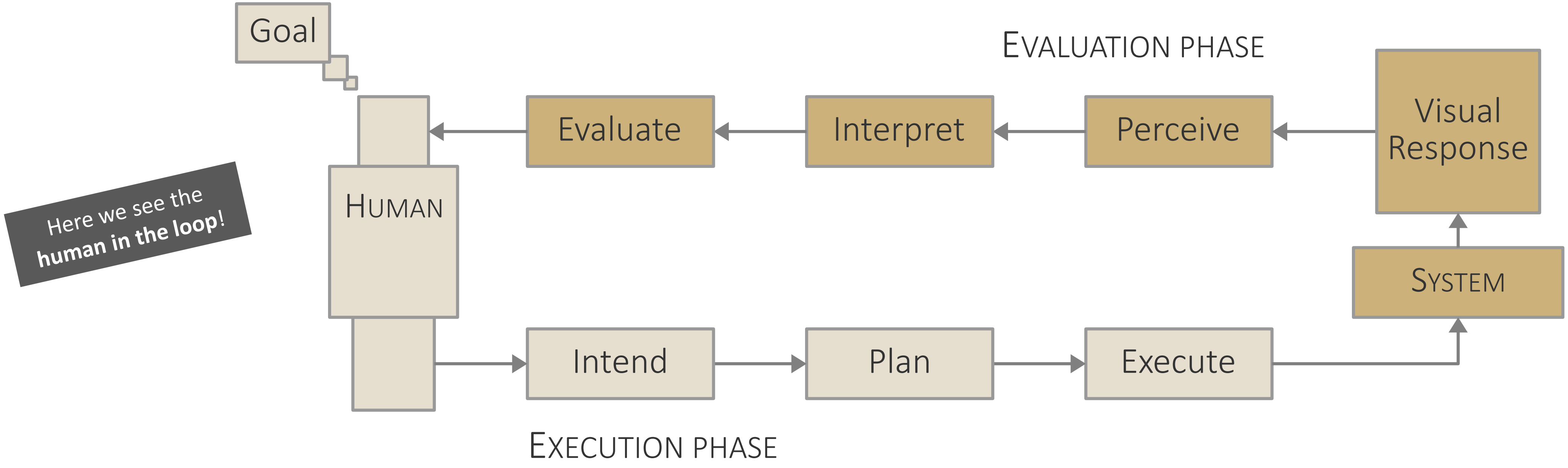
Interaction intents and action patterns

- We know now about the *why* (interaction intents) and the *what* (action patterns) of interaction for visual data analysis
- Let's next look at the *how*

- **Norman's action cycle**

The action cycle

- General interaction model by Norman ([1988 and 2013](#))
- Here adapted to interaction with a computer system



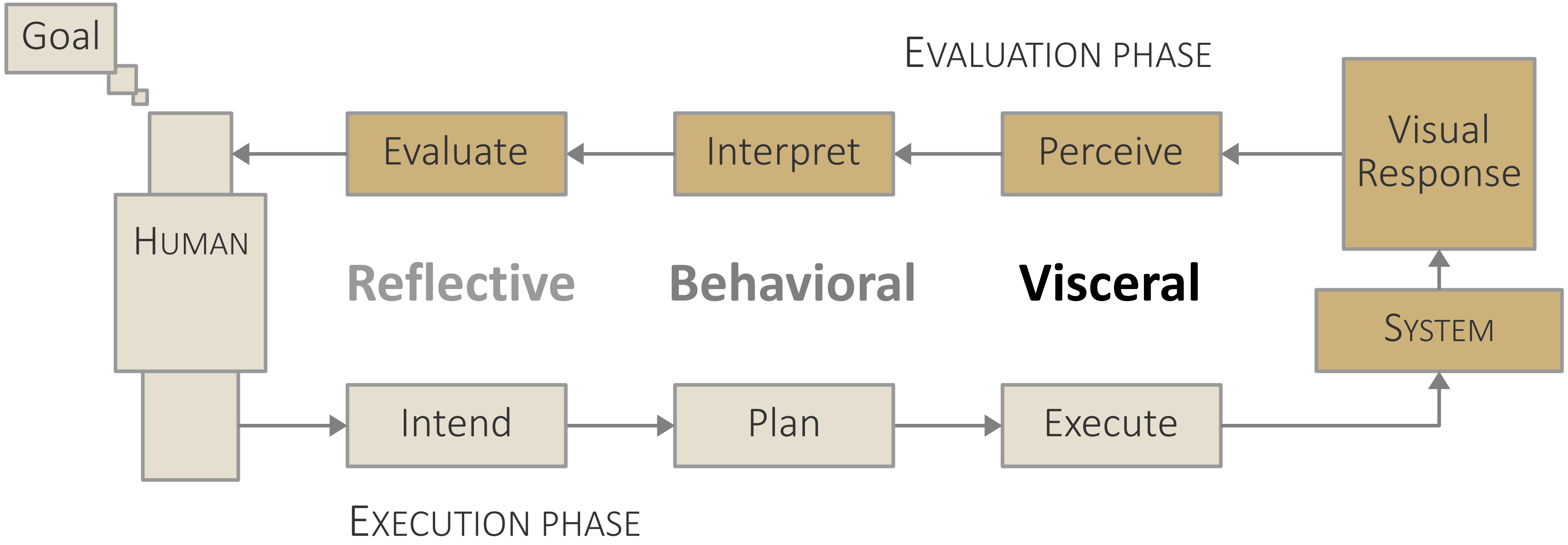
The action cycle

Phases and actions

- Goal: Overall goal for interacting
- **Execution phase**
  - Intend: Develop desire to change something
  - Plan: Specify actions to accomplish desired change
  - Execute: Perform the planned actions
- **Evaluation phase**
  - Perceive: See the response of the system
  - Interpret: Understand the response of the system
  - Evaluate: Compare response with desired outcome (re-run loop if necessary)

The action cycle

Levels of processing

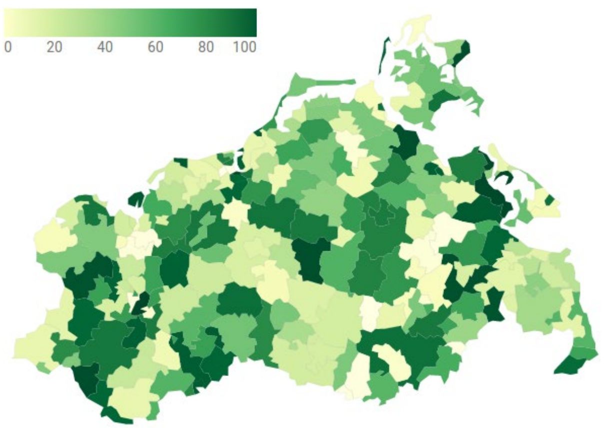


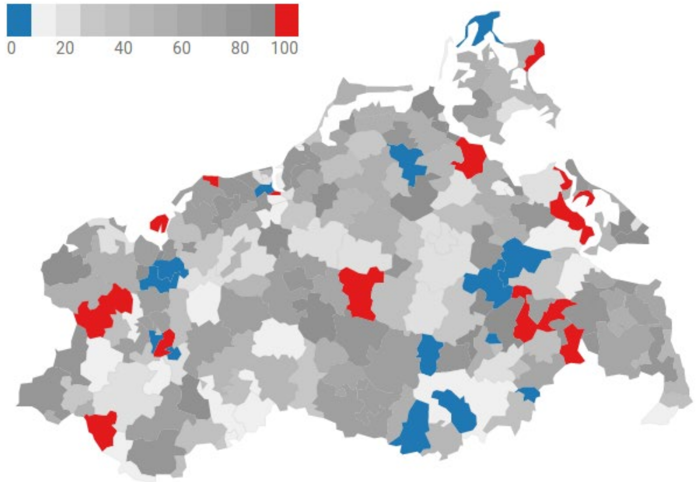
The action cycle

Levels of processing

- Visceral (instinctive)
  - Quick, basic mechanisms, subconscious, without awareness and control
- Behavioral
  - Quick, learned skills, largely subconscious, little awareness, controlled
- Reflective
  - Slow, deep analysis, conscious, full awareness and control

The action cycle





Example

- Goal: *Locate extrema*
- Intend: *Adjust color scale*
- Plan: *Need to operate color selector*
- Execute: *Move mouse to select alternative color scale*
- Perceive: *Visual representation changes its colors*
- Interpret: *Extrema now clearly visible in red and blue*
- Evaluate: *Success, extrema could be located*

Recall the loops of the **knowledge generation process** from Lecture 3.

In practice, the **loop usually runs many times** due to the dynamically changing interests of users during the visual data analysis!

The action cycle

Levels of interaction

- Depending on the number of operations performed
- **Low-level interaction**
    - Mapping fundamental degrees of freedom of interaction devices to basic operations of **pointing and manipulating** graphical objects
    - Interaction alphabet or syntax
- **Intermediate-level interaction**
    - Low-level interactions combined to **semantically meaningful activities**
    - Interaction vocabulary
- **High-level interaction**
    - Interaction vocabulary is employed to form longer action sequences
    - High-level **problem solving and analytical thinking**

Requirements for efficient interaction

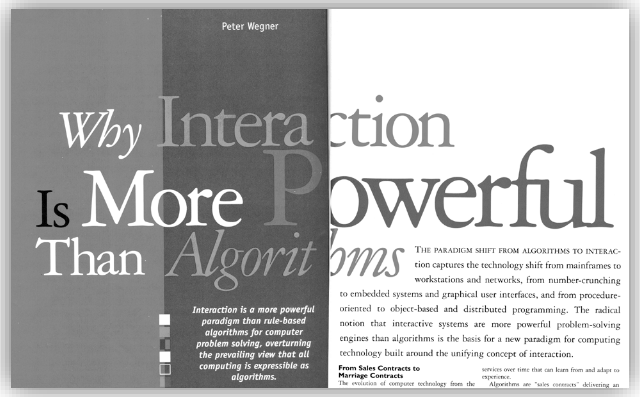
- We learned about *why*, *what*, and *how* of interaction
- We now move on to the question: *What makes good interaction?*

- **Interaction costs**
- **Directness of interaction**
- **Design guidelines**

Requirements for efficient interaction

Arguments **for** interaction

> “**Interaction is more powerful than algorithms:** Algorithms are metaphorically dumb and blind because they cannot adapt interactively while they compute. They are autistic in performing tasks according to rules rather than through interaction. In contrast, interactive systems are grounded in an external reality both more demanding and richer in behavior than the rule-based world of noninteractive algorithms.
>
> — [Wegner, 1997]


Peter Wegner

Why Interaction Is More Powerful Than Algorithms

Interaction is a more powerful paradigm than rule-based algorithms for computer problem solving, overturning the prevailing view that all computing is expressible as algorithms.

THE PARADIGM SHIFT FROM ALGORITHMS TO INTERACTION captures the technology shift from mainframes to workstations and networks, from number-crunching to embedded systems and graphical user interfaces, and from procedure-oriented to object-based and distributed programming. The radical notion that interactive systems are more powerful problem-solving engines than algorithms is the basis for a new paradigm for computing technology built around the unifying concept of interaction.

From Sales Contracts to Marriage Contracts

The evolution of computer technology from the

services over time that can learn from and adapt to experience.

Algorithms are "sales contracts" delivering an

Requirements for efficient interaction

Arguments **against** interaction

- Think about: When can interaction be harmful?

> “**Interactivity considered harmful:** […] **interactivity has even *worse* problems** than simply being a frustrating waste of time:
> - The **user has to already know *what* she wants** in order to ask for it. […] Purely interactive software forces the user to make the first move.
> - The **user has to know *how* to ask.** That is, she must learn to manipulate a machine. […]
> - Navigation implies state. Software that can be navigated is software in which the **user can get lost.** The more navigation, the more corners to **get stuck** in. The more manipulable state, the more ways to wander into a **“bad mode.”** State is the primary reason people fear computers—**stateful things can be *broken*.**
>
> — Victor, 2006

Requirements for efficient interaction

- Arguments for *and* against interaction!
- **So, what makes good interaction?**

- First of all, be aware of the **costs of interaction** (Lam, 2008)
  - Execution phase → **"Gulf of execution"**
  - Evaluation phase → **"Gulf of evaluation"**
- Costs can be
  - **Physical**
  - **Mental**

Costs of interaction

Physical costs

- Related to performing physical actions (e.g., moving fingers or forearms to control the mouse, or physically scanning the visual response)
- Mostly visceral activities with little to no dedicated attention necessary
- Examples of costly interactions
  - Many repetitive actions during exploratory data analysis
  - Deep menu structures that require long and accurate pointer movements
  - Much visual feedback distributed across screen requires much eye movement

Costs of interaction

Mental costs

- Related to mentally preparing interactions and understanding the system's response
- Reflective and behavioral activities require users pay attention
- Examples of costly interactions
  - Planning: Determine which graphical object afford which actions
  - Interpreting: Many simultaneous changes difficult to follow
  - Interpreting: Subtle changes hardly visible and interpret
  - Evaluating: Compare new visual representation against previous version from short-term memory

Costs of interaction

- Think about: Should we try to tackle each and every analysis problem interactively?
- **No!** Interaction is powerful, but not a silver bullet.
- Interaction can be a burden if seemingly simple tasks are cumbersome to accomplish due to bad interaction design
- Users may feel uncomfortable with being responsible for adjusting parameters
- Unclear whether visible features corresponds to features in the data or are just artifacts of (inappropriate) interactive adjustments

Costs of interaction

- Think of interaction in a ***less-is-more*** way
- System should be responsible for relieving user of unnecessary work
- Only as a last resort should input be requested from user, and the input should be made through a well-designed interactive interface

- Question: How can interaction costs be reduced and the gulfs of execution and evaluation be narrowed?
- Answer: Consider **high degree of directness** of interaction.

Directness of interaction

- **Directness** with which interaction is carried out determines how smoothly and efficiently the action cycle can run

- **Direct manipulation** paradigm:

> "Are we analyzing data? Then we should be **manipulating** the **data themselves**;
> or if we are designing an analysis of data, we should be **manipulating** the **analytic structures themselves**.
>
> — Hutchins et al., 1985

Directness of interaction

- **Direct manipulation** is *the* preferred interaction paradigm for interactive visual data analysis (Shneiderman, 1983)
  - **Objects** and **actions** of interest are **presented continuously** using meaningful visual metaphors
  - The user's **requests** are **expressed** through **physical actions**, rather than complex syntax
  - **Actions** are **rapid**, **incremental**, and **reversible**, and their **effect** is **immediately visible**

> "[Direct manipulation] can be defined as the use of **continuous physical motions** (of one's hand) to interactively manipulate **persistent visual representations** of objects, with **continuously updated feedback**, and with the ability to **undo actions** by simply reversing physical motions.
>
> — McGuffin & Fuhrman, 2020

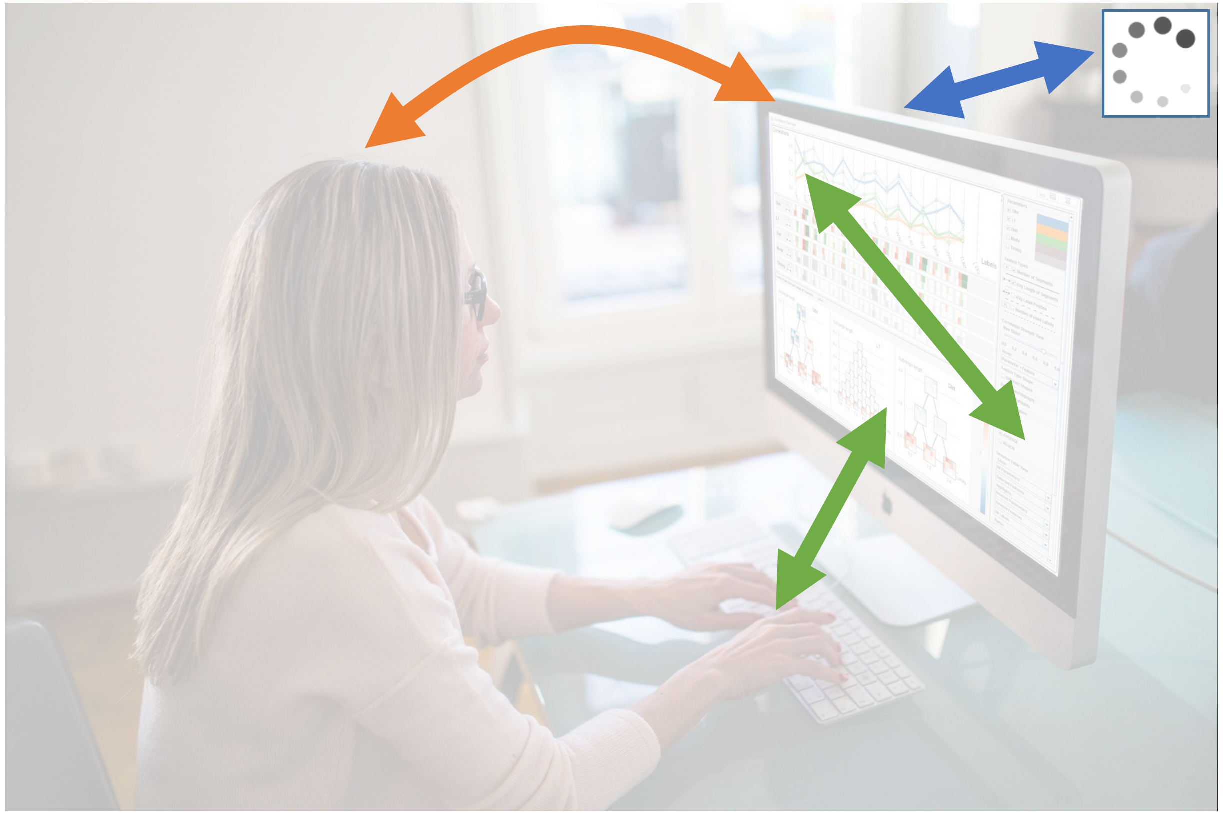
- But what does directness actually mean in terms of designing interactive visual data analysis solutions?

Directness of interaction

- Look at directness from an opposite point of view: **Directness** is **inversely proportional** to the degree of **separation** of human actions and system responses
- Different types of separation
  - **Conceptual separation**
  - **Spatial separation**
  - **Temporal separation**

Directness of interaction

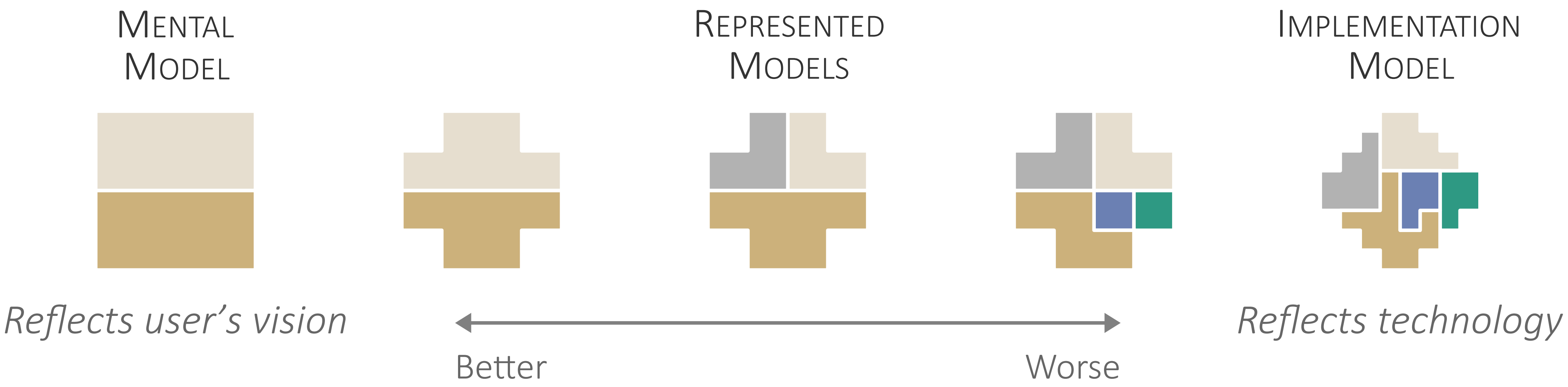
- Conceptual separation
  - Model discrepancy
- Spatial separation
  - Distance
- Temporal separation
  - Latency



Directness of interaction

Conceptual separation

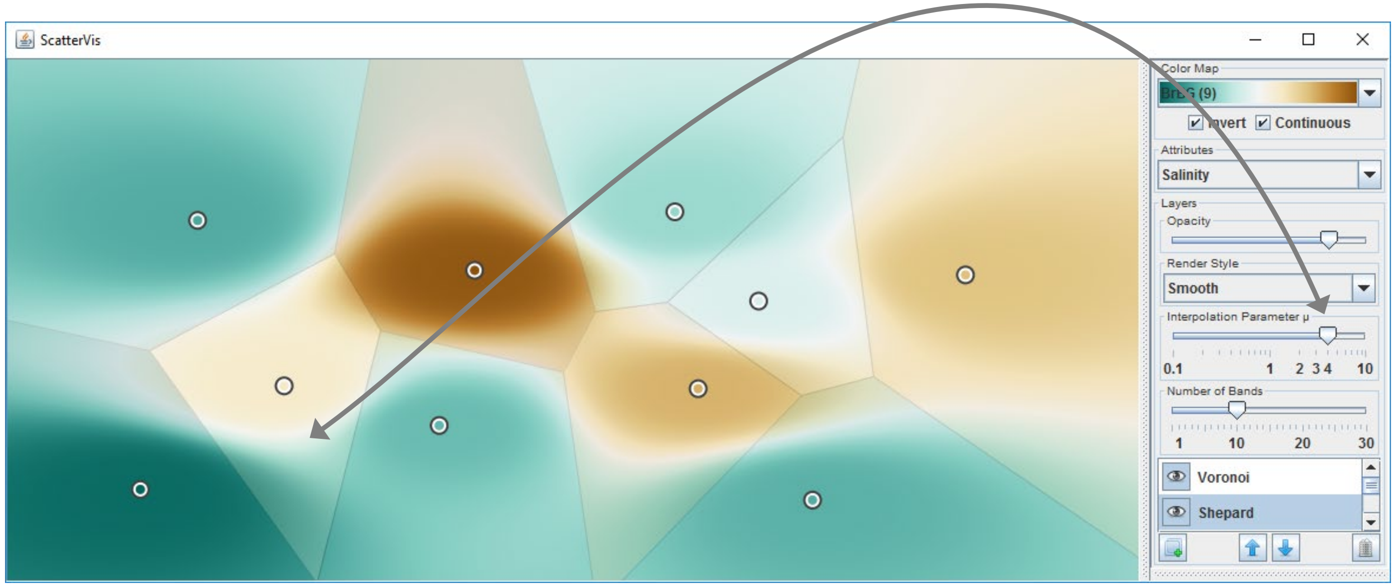
- Due to differences between user's mental model and system's represented model



Directness of interaction

Spatial separation

- Due to the spatial distance to be covered during interaction (e.g., pointer movements and eye movements)

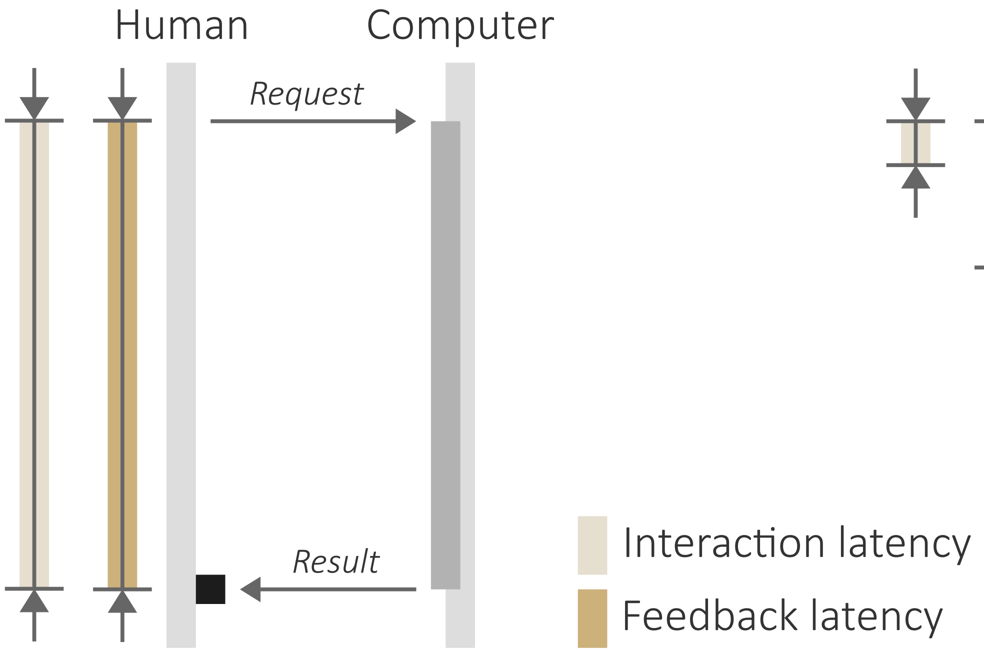


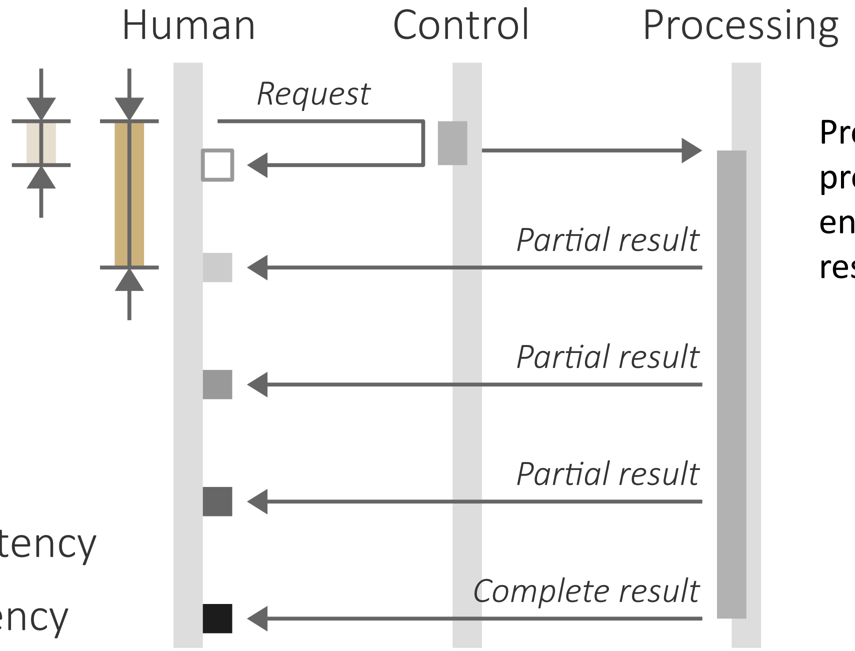
Directness of interaction

Temporal separation

- Due to the latency between user's actions and system's response

Long-running computation may block interaction cycle





Progressive processing ensures system responsiveness

Directness of interaction

Scenarios of different directness

Example interaction: Zoom in to look at details of interesting nodes in a graph

- Source code editing
  - Edit source code and compile (very low degree of directness)
- Scripting commands
  - Execute scripting commands (moderate degree of directness)
- Graphical interface
  - Control zoom via graphical controls (good degree of directness)

Directness of interaction

Scenarios of different directness

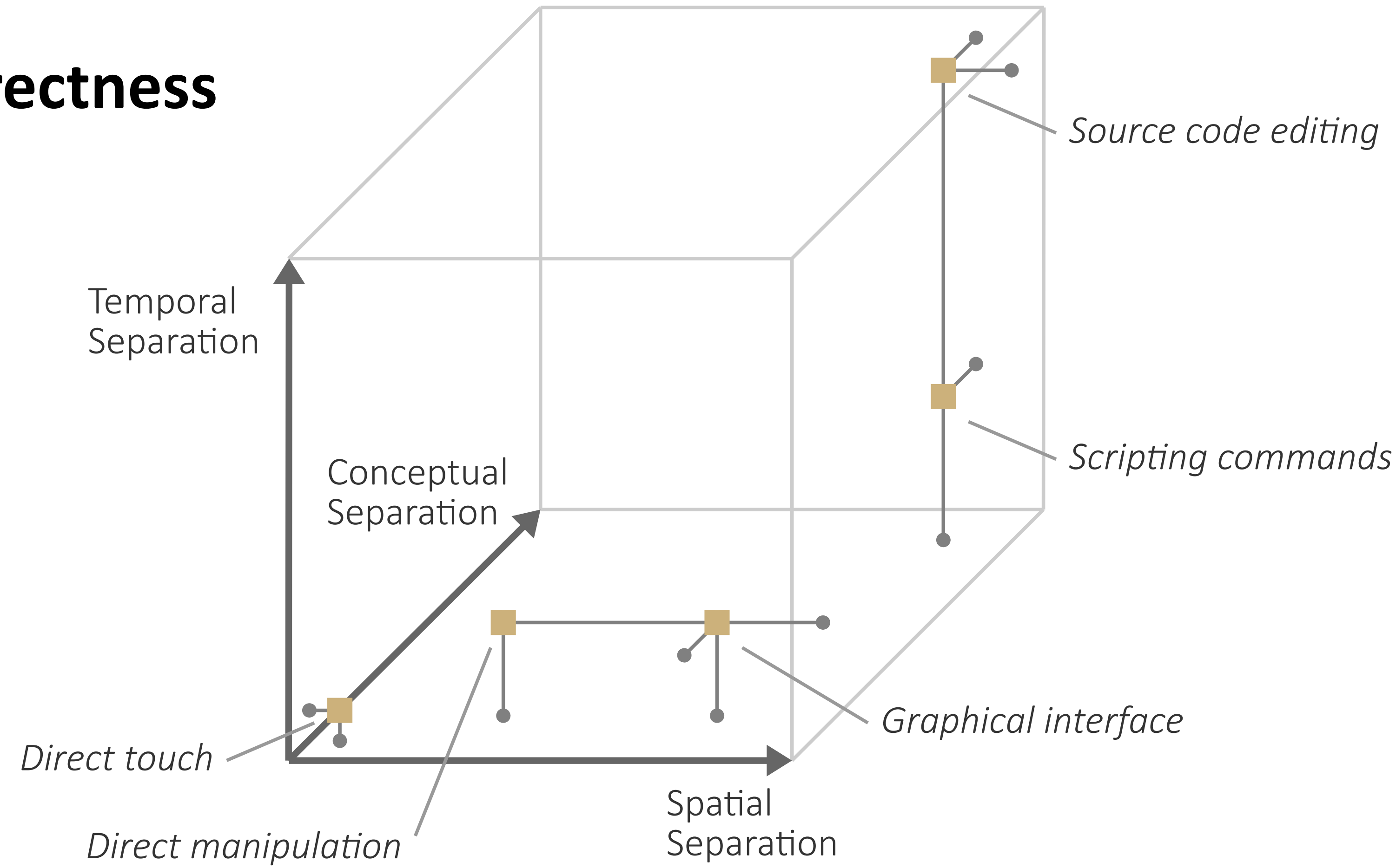
- Direct manipulation
  - Use mouse to draw elastic rectangle around nodes (high degree of directness)
- Direct touch
  - Use fingers to draw elastic rectangle (yet higher degree of directness)

- Despite the seemingly clear preference for direct touch, each scenario has its advantages and disadvantages
- Think about: What are these (dis)advantages?

Directness of interaction

Scenarios of different directness

Summarizing view with approximate degrees of separation



Design guidelines

- We learned that reducing separation leads to direct interaction, which in turn is good for reducing interaction costs

- In addition to striving for directness, there are further design guidelines to be taken into account
  - **Golden rules**
  - **Fluid interaction**

Design guidelines

Golden rules (Shneiderman & Plaisant, 2010)

1. **Strive for consistency.** Consistent actions should be required in similar contexts and be responded to consistently.
2. **Cater to universal usability.** The system should be usable for novices, casual users, and experts alike.
3. **Offer informative feedback.** For each possible action, there should be informative feedback appropriate to its importance.
4. **Design dialogs to yield closure.** Action sequences should have a well-defined beginning, middle, and end.

Design guidelines

Golden rules (Shneiderman & Plaisant, 2010)

5. **Prevent errors.** The system should prevent serious errors and be able to recover from minor problems.
6. **Permit easy reversal of actions.** Actions should be reversible to allow for undoing accidental actions and to encourage exploration.
7. **Support internal locus of control.** Users should be given the feeling that they are in charge, not the computer.
8. **Reduce short-term memory load.** Short-term memory load should be limited to seven plus minus two chunks of information.

Design guidelines

Fluid interaction (Elmqvist et al., 2011)

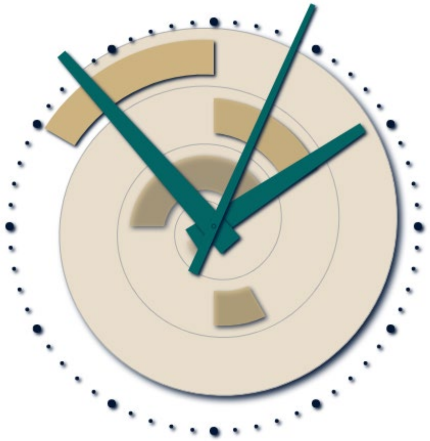
- Promote flow
- Support direct manipulation
- Minimize the gulfs of execution and evaluation

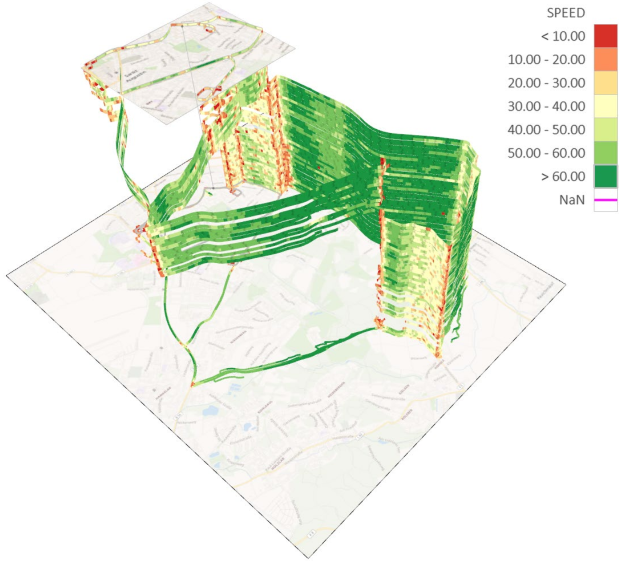
> "To be **most effective**, visual analytics tools must support the **fluent and flexible** use of visualizations at rates resonant with the **pace of human thought**.
>
> — Heer & Shneiderman, 2012

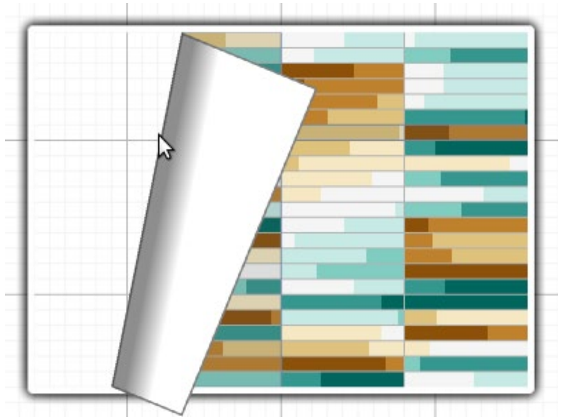
Design guidelines

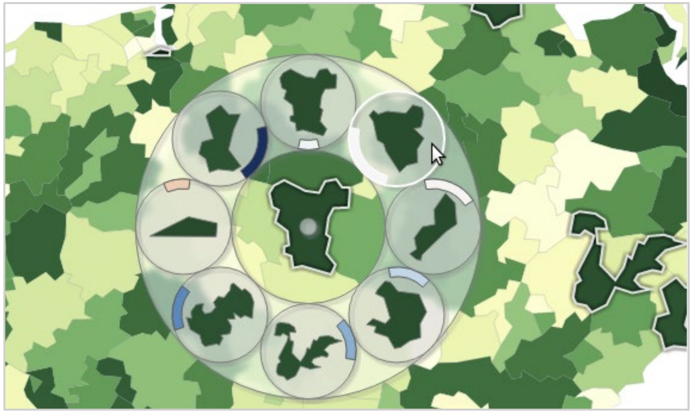
Fluid interaction (Elmqvist et al., 2011)

- Demos
  - SpiraClock
  - TrajectoryVis
  - FoldableVis
  - Bring’N’Compare









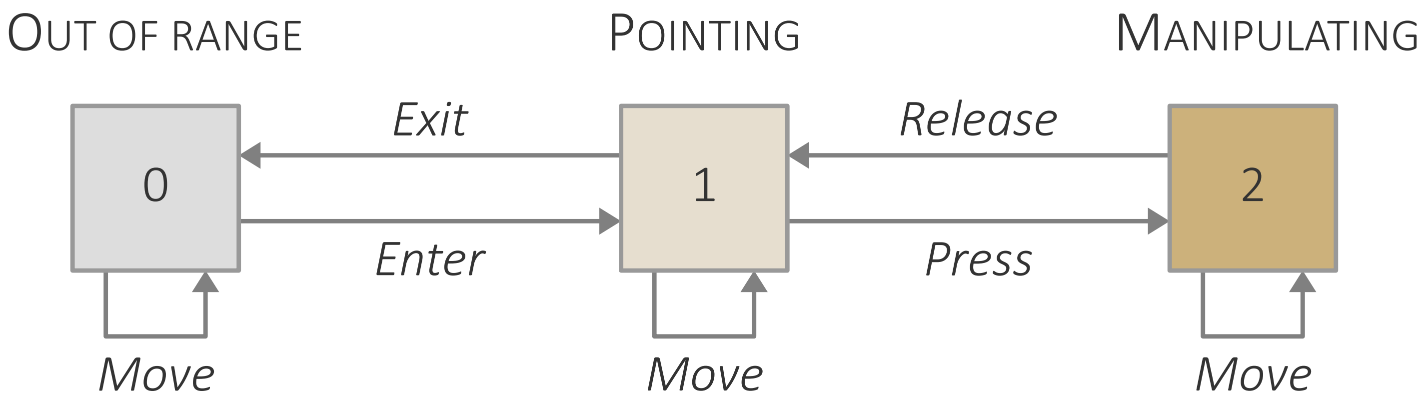
Basic operations

- So far for the conceptual background of interaction for visual data analysis
- Now we move on to the **basic operations**
  - **Taking action**
  - **Generating feedback**

Basic operations

Taking action

- *Three-state model* by Buxton (1990)
- **Point:** *Where* should the interaction take effect
- **Manipulate:** *What* should the effect be



Basic operations

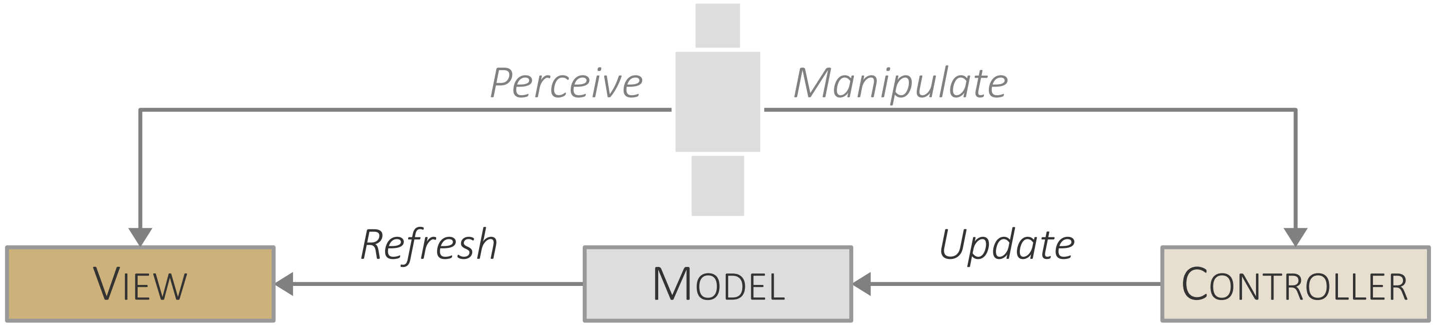
Taking action

- Modes of interaction
  - **Discrete** (or stepped) **interaction**
    - Temporarily enter state 2 (e.g., click or touch)
    - Trigger discrete change in visualization
    - Good for selecting among few alternative choices
  - **Continuous interaction**
    - Stay in state 2 for a longer time (e.g., drag slider)
    - Continuously update the visualization
    - Good for browsing larger numbers of alternative choice

Basic operations

Generating feedback

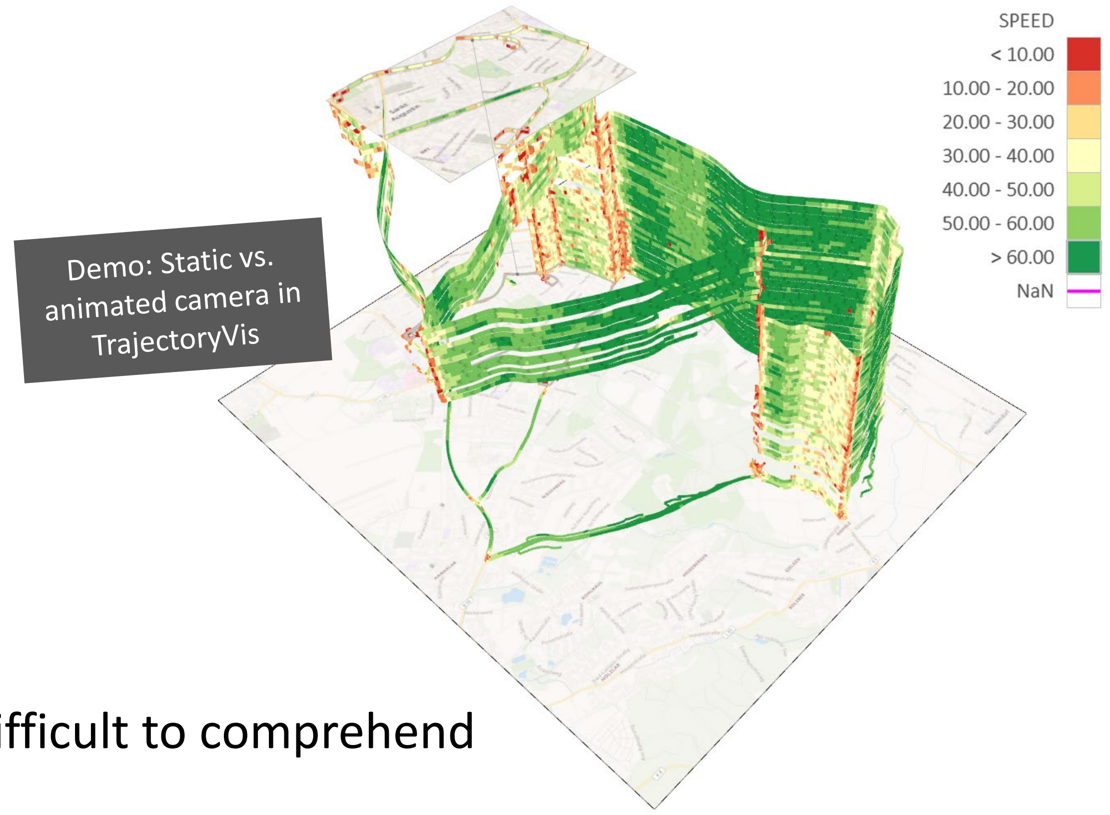
- **Update**
  - Change internal state of the visualization
  - Perform stages of the visualization pipeline as necessary
- **Refresh**
  - Present new visual representation to make internal state change visible



Basic operations

Generating feedback

- Types of visual feedback
  - **Static feedback**
    - Immediately replace old state with new state
    - Can be good to attract attention
    - But instantaneous switch can make changes difficult to comprehend
  - **Animated feedback**
    - Smoothly transition between old state and new state
    - Can help users understand the state change
    - But animation takes time, which delays the action cycle



Summary

- Human in the loop (intents, action patterns, Norman's action cycle)
- Requirements (interaction costs, directness, design guidelines)
- Basic operations (point and manipulate, update & refresh, feedback)

- Next lectures:
  - Basic interaction techniques
  - Advanced interaction techniques

Assignments

1. Experiment with the EnhancedSpiral.js and think about the modes of interaction that are provided!
2. Play with Responsive Matrix Cells and think about the types of feedback being offered!
3. Read "What is Interaction for Data Visualization" by Diamara and Perin (2020)!

Questions

1. How does the required degree of interactivity vary depending on visualization goals?
2. Name and explain 3 main categories of interaction intents!
3. What is the difference between unipolar and bipolar action patterns? Give examples!
4. Explain the phases and actions of Norman's action cycle!
5. Characterize the different levels of processing involved in the action cycle!
6. Give arguments for and against interaction for visual data analysis?
7. What interaction costs are relevant in the context of visual data analysis?
8. When does interaction reach a high level of directness?
9. Name some golden rules to be taken into account when designing interaction!
10. Sketch and explain Buxton's "three-state model"!
11. What are discrete and continues interaction, and static and animated feedback?