Interactive Visual Data Analysis

10 – Basic picking, selection, and zooming

Objectives

- How can we identify data and interface items on the display: Learn concepts and strategies of fundamental picking
- How can relevant information be emphasized in a visual representation: Learn about selection and accentuation

Overview

- **Today: Basic techniques**
  - Fundamental picking
  - Interactive selection and accentuation
  - Navigating zoomable visualizations
- Next lectures: Advanced techniques

Motivation

We learned about **interaction intents** in Lecture 9.

Interaction intents (Yi et al., 2007)

- Fundamental picking, interactive selection, and accentuation
  - *Mark* something as interesting
  - *Show me* something conditionally
- Navigating zoomable visualizations
  - *Show me* something else
  - *Show me* more or less detail

Fundamental picking

- Before we can do any graphical interaction, we need to determine what data or interface elements are under the cursor
  - **Given:**
    - Geometry of data and interface $G = \{g_1, \dots, g_n\}$
    - Position in screen coordinates $p = (x, y)$
  - **Find:**
    - Geometry $G_p \subset G$ at p
  - **Result:**
    - $|G_p|$ = 0, if no match could be found
    - $|G_p|$ = 1, for a unique match
    - $|G_p|$ > 1, for multiple matches

Fundamental picking

- Key problem when picking data elements
  - **Inverse mapping** of visual representation to data (not always a unique solution)
  - Picking must be **as fast as possible** to reduce temporal separation

We discussed **temporal separation** in Lecture 9.

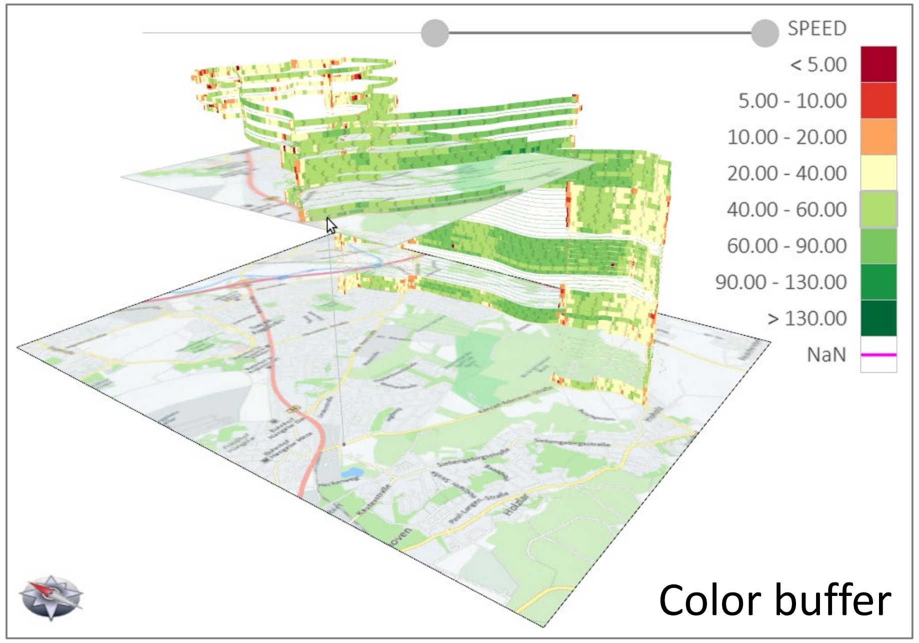
- Implementation possible at different stages of visualization pipeline
  - **Pixel picking** at the image stage
  - **Geometry picking** at the geometry stage
  - **Direct picking** at the mapping stage

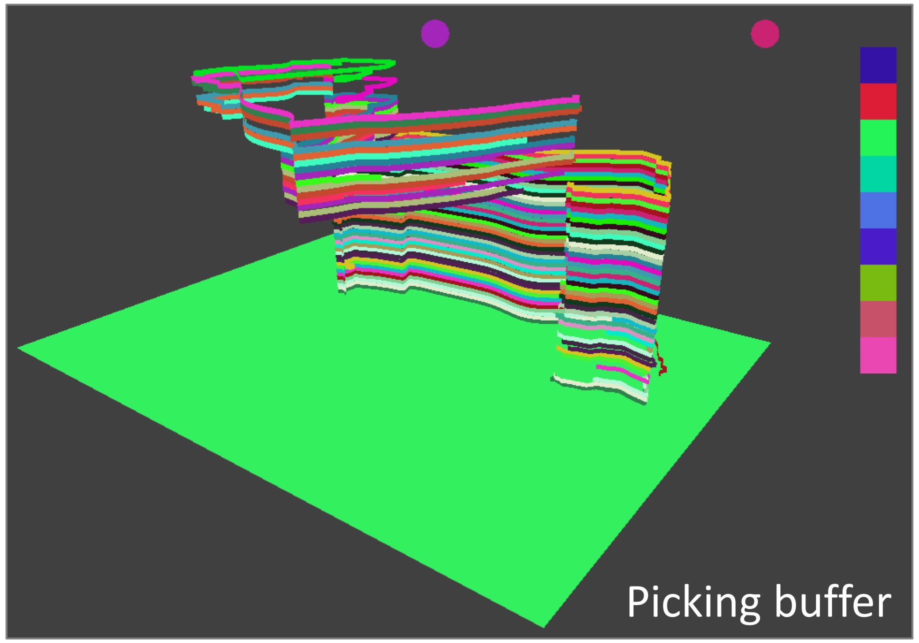
Fundamental picking

Pixel picking

- Render visualization twice
  - **Color buffer:** regular visualization colors
  - **Picking buffer:** object ids as pseudo colors
- Picking is then a simple constant-time lookup in the picking buffer → $O(1)$

- Think about: What can we learn from the two example buffers?



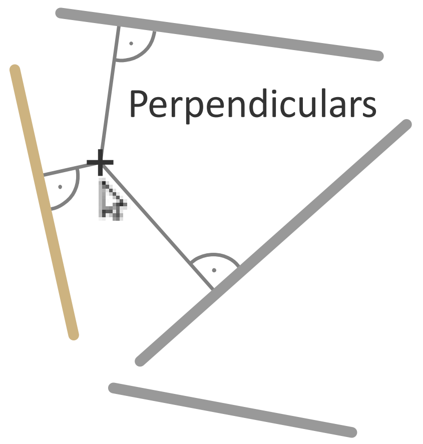


Fundamental picking

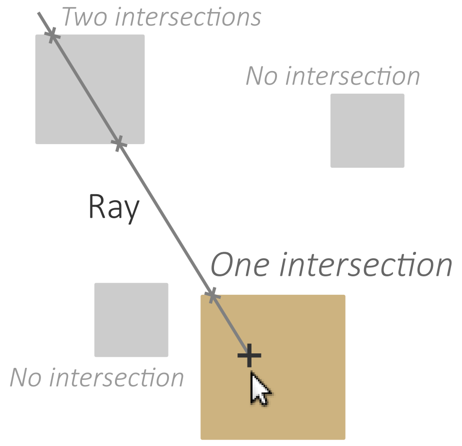
Geometry picking

- Geometric tests depending on geometry to pick → $O(n)$
- If accelerated with spatial data structures (e.g., grids, R-trees) → $O(\log n)$

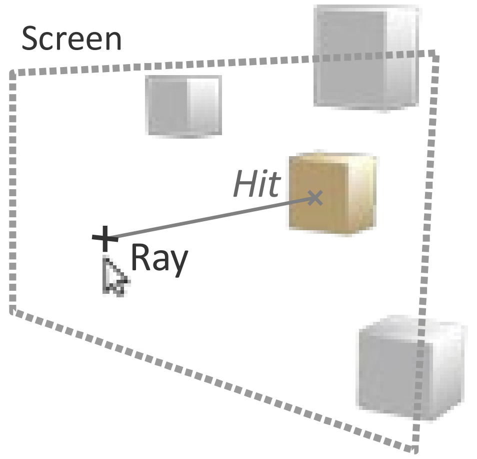
Nearest-neighbor search for 0D/1D



Inclusion test for 2D



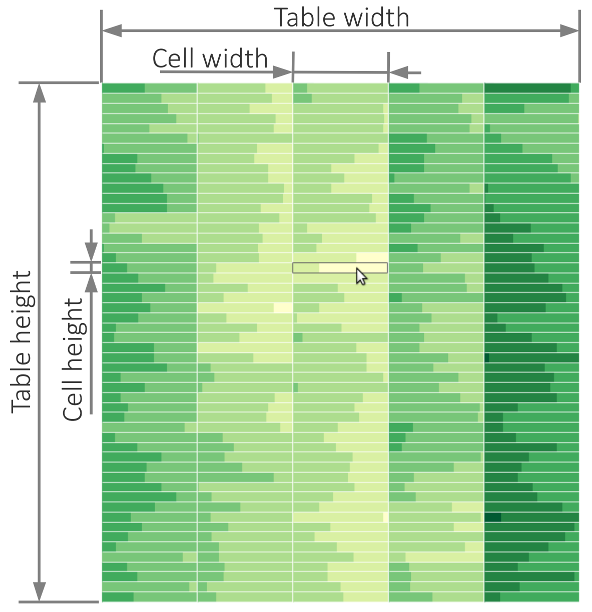
Ray intersection for 3D

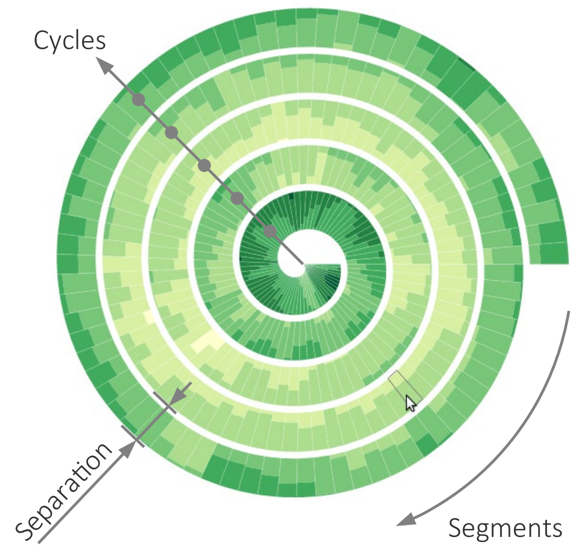


Fundamental picking

Direct picking

- Closed-form calculation → $O(1)$
- Possible for regular visual arrangements of data (e.g., tabular or spiral visualization)
- Not possible for irregular visual arrangements (e.g., force-directed node-link visualization)

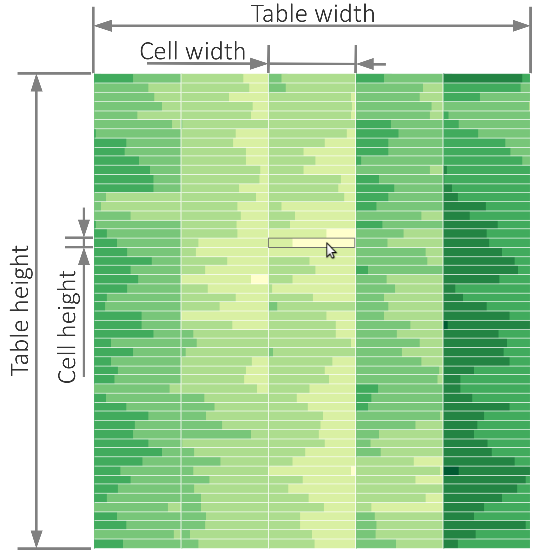




Fundamental picking

Direct picking

- Picking segments from a **table**



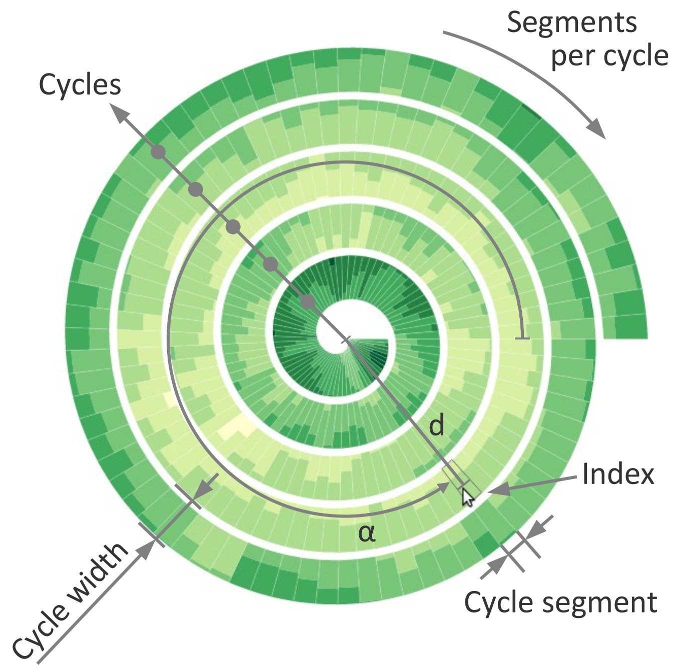
```
function pick(table, x, y) {

    // Compute cell height and width
    let cell_height = table.height / table.rows;
    let cell_width = table.width / table.columns;

    // Compute row and column at (x, y)
    let row = Math.floor(y / cell_height);
    let col = Math.floor(x / cell_width);

    return [row, col];
}
```

Fundamental picking

In Lecture 6, we learned how to draw such a spiral.

Direct picking

- Picking segments from a **spiral**



```
function pick(spiral, x, y) {

    // Compute angle α in range [-PI .. PI]
    let α = Math.atan2(y, x);
    // Map negative angles from [-PI .. 0] to [PI .. 2*PI]
    if (α < 0) α += 2 * PI;
    // Normalize angle α to from [0 .. 2*PI] to [0 .. 1]
    α /= 2 * PI;
    // Determine the segment at angle α
    let cycle_segment = Math.floor(α * spiral.segments_per_cycle);

    // Distance d from spiral center
    let d = Math.sqrt(x * x + y * y);
    // Subtract fraction of the cycle width at angle α
    d -= α * spiral.cycle_width;
    // Normalize distance to [0 .. 1]
    d /= spiral.radius;
    // Cycle at distance d (+ 1 accounts for extent of spiral band)
    let cycle = Math.floor(d * (spiral.cycles + 1));

    let index = cycle_segment + cycle * spiral.segments_per_cycle;

    return index;
}
```

Interactive selection and accentuation

- **Primary intent** when engaging in interactive visual data analysis
  - **Mark data as interesting**, temporarily or permanently
- Conceptual formulation
  - Visual representation of some dataset $V(D)$
  - Subset of current interest $D^{+} \subset D$
  - Data not currently being relevant $D^{-} = D \backslash \mathrm{D}^{+}$
  - D^{+} and D^{-} are constantly in flux during the data analysis
- Consequences
  - D^{+} and/or D^{-} must be **specified** by the user → **Selection**
  - D^{+} must clearly **stand out** in the visual representation → **Accentuation**

Interactive selection

- Question: How can users specify their interest?

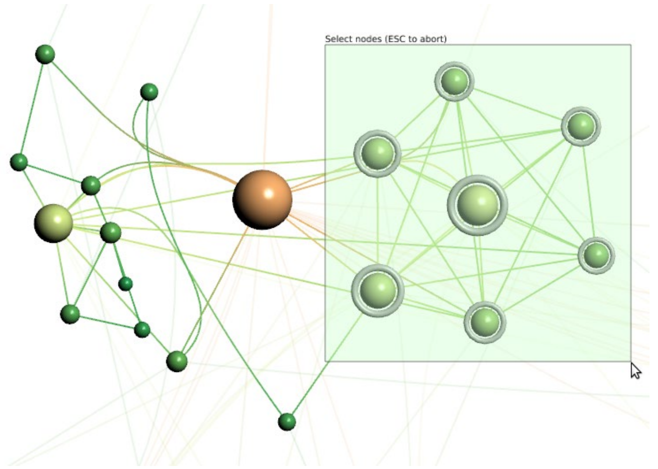
- Basic methods:
  - Select based data’s **screen location**:
    Mark something → **Interactive brushing**
  - Select based data’s **attributes**:
    Define conditions → **Dynamic querying**

Both, interactive brushing and dynamic querying should be supported, because either alone is usually not sufficient for all user needs

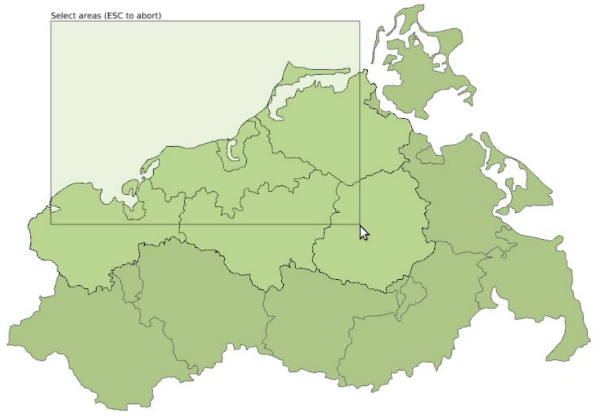
Interactive selection

Interactive brushing

- Point selection
  - Utilize basic picking for discrete picking of individual data elements
- Range selection
  - Rubberband or lasso selection
  - Variants: Inclusion or intersection
  - Think about: What are pros and cons of rubberband and lasso selection and its variants?



Data elements must be included in rubberband in order to be selected

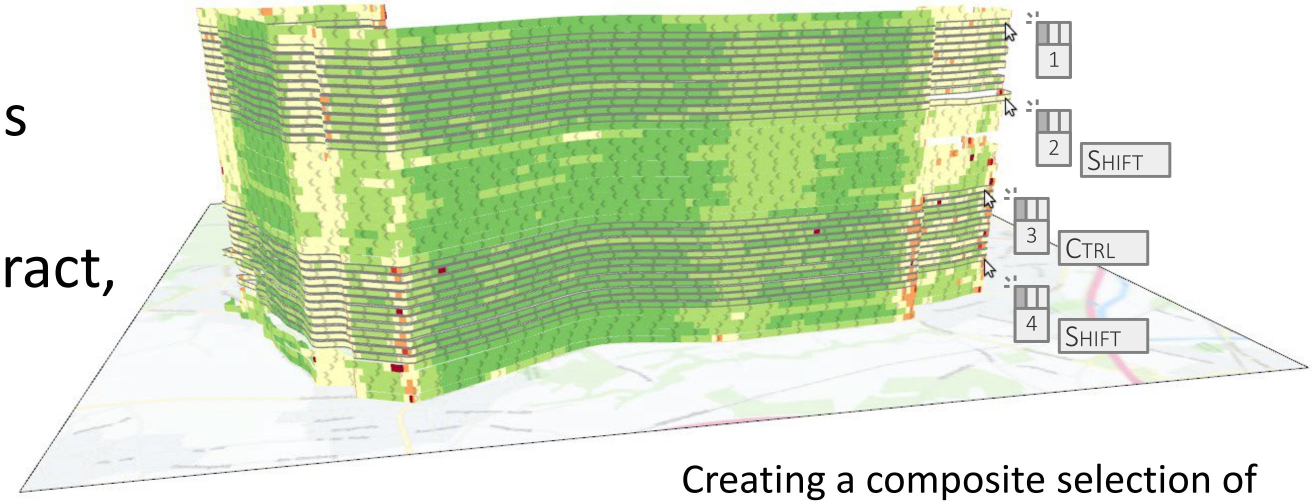


Data elements need to intersect with the rubberband in order to be selected

Interactive selection

Interactive brushing

- Composite selection
  - How to combine multiple selections
  - Theoretically, 524.288 possible combinations of add, remove, subtract, intersect, union operations ([Wills, 1996](#))
  - Today rather standard
    - CTRL modifier: Toggle selection state
    - SHIFT modifier: Select multiple items
    - Combine modifiers for complex selections



Creating a composite selection of trajectories using modifier keys

Interactive selection

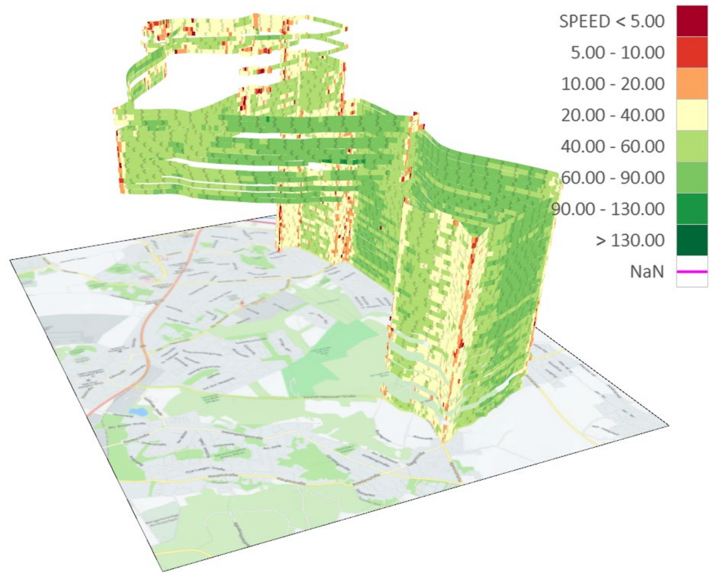
Dynamic querying

- Interactive legends
  - Legends typically show data ranges and corresponding visual encodings
  - Utilize legends for selecting the already shown data ranges
  - Direct interaction within the visualization view
- Query sliders
  - Dedicated value or range sliders facilitate selection of data ranges
  - Logic operators allow users to formulate complex query conditions
  - No direct interaction with visualization, but separate interface

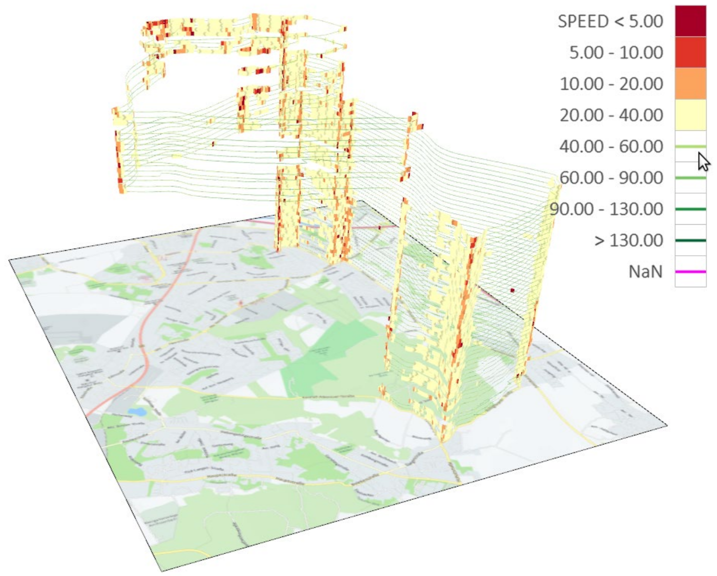
Interactive selection

Dynamic querying

- Interactive legends



Problem: How to focus on low-speed segments?

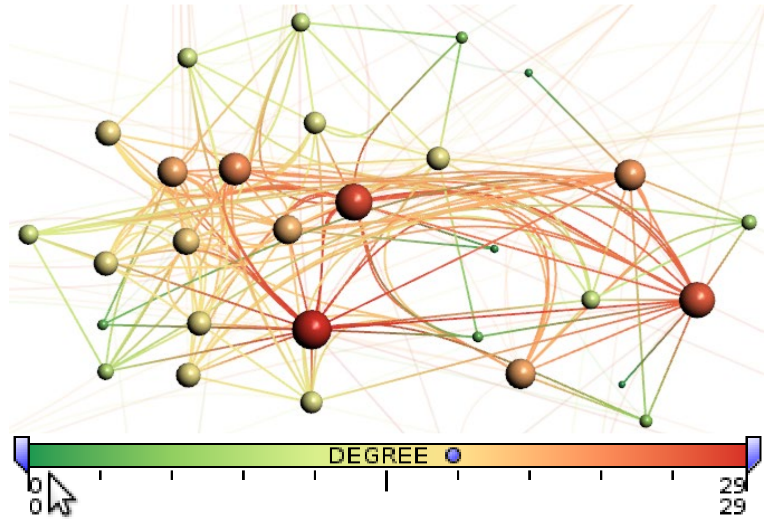


Solution: Utilize the legend to filter out high-speeds!

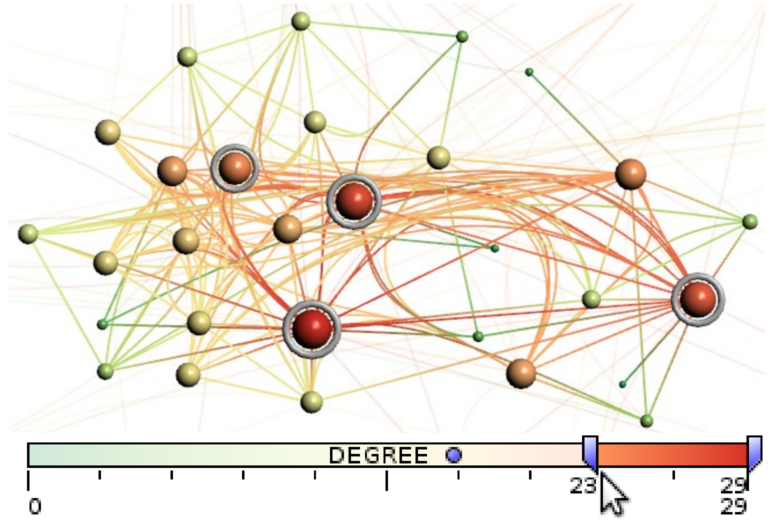
Interactive selection

Dynamic querying

- Query sliders



Problem: How to focus on high-degree nodes?



Solution: Utilize query slider to mark high-degree nodes!

Accentuation

- Given: Regular visual representation of the data $V(D)$
- Question: How to represent D^+ and D^- ?
- Conceptual answer:
  - Define additional visual encodings
    - V^+ emphasizes
    - V^- attenuates
  - Using V^+ and V^- on D^+ and D^-, we can define different accentuation strategies:
    - **Highlighting**
    - **Dimming**
    - **Filtering**

Accentuation

- **Highlighting:** $V^+(D^+) + V(D)$
  - Emphasize relevant data
  - Regular display of less-relevant data
- **Dimming:** $\mathrm{V}(D^+) + V^-(D^-)$
  - Regular display of relevant data
  - Attenuate less-relevant data
- **Filtering:** $\mathrm{V}(D^+)$
  - Regular display of relevant data
  - Omit all other data

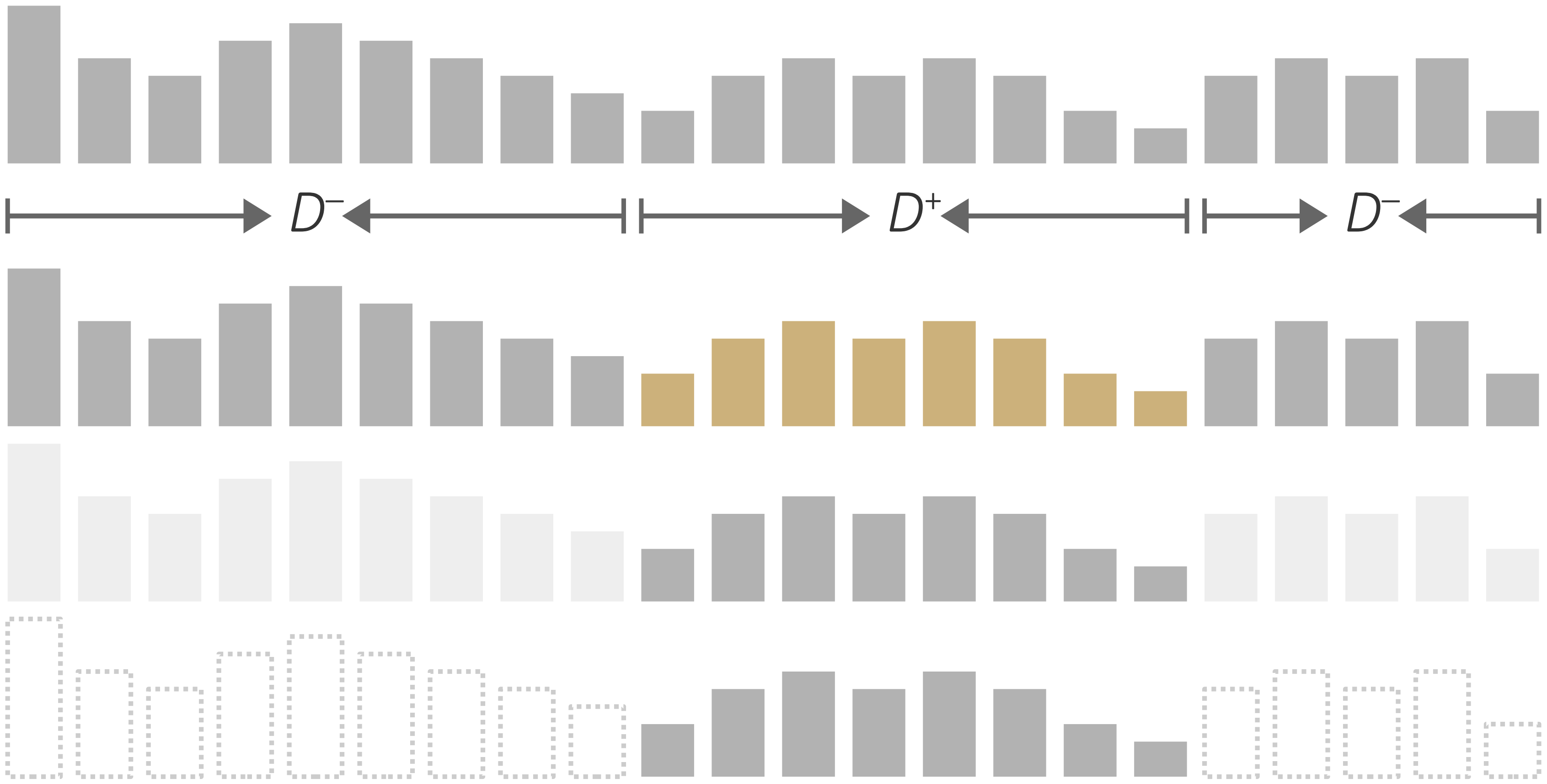
Accentuation

REGULAR VISUALIZATION
$V(D)$

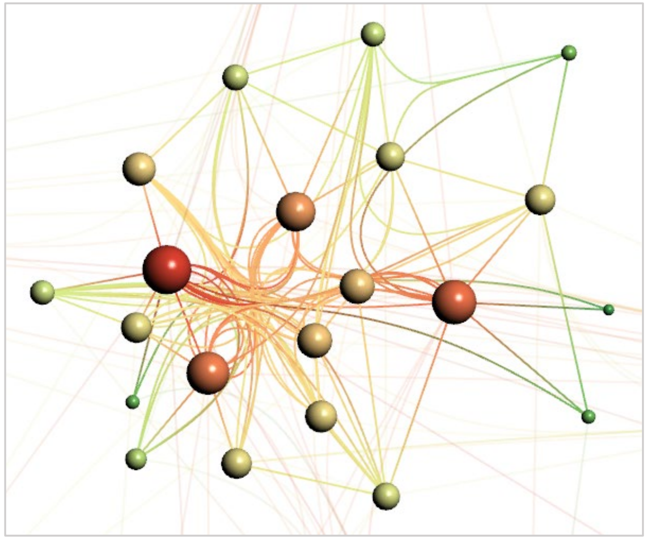
HIGHLIGHTING
$V^+(D^+) + V(D^-)$

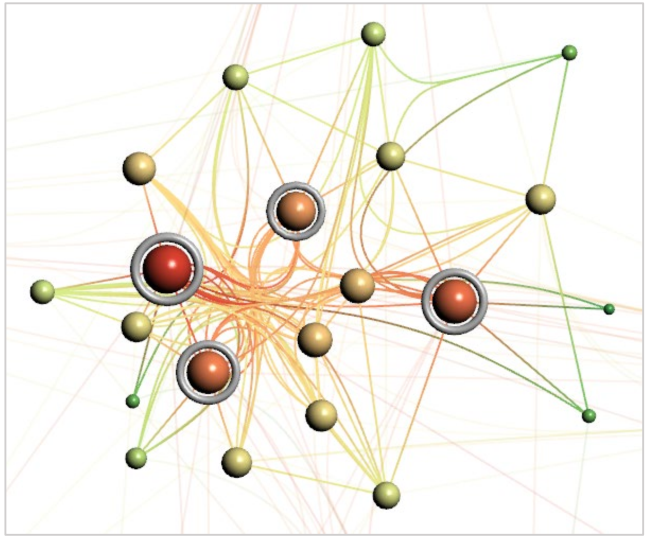
DIMMING
$V(D^+) + V^-(D^-)$

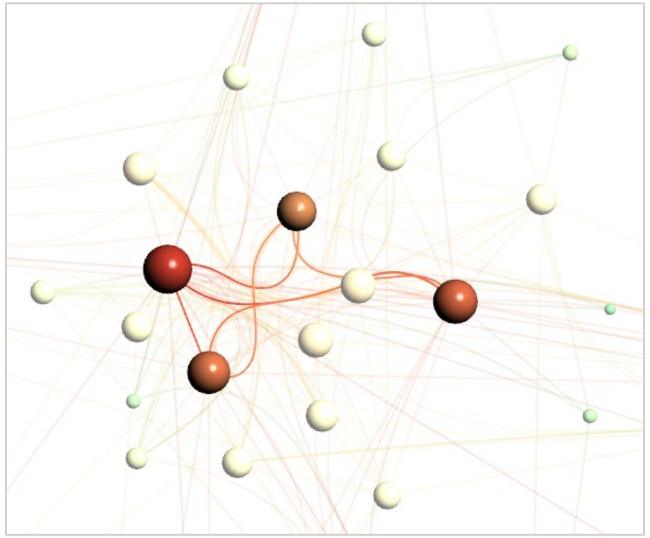
FILTERING
$V(D^+)$

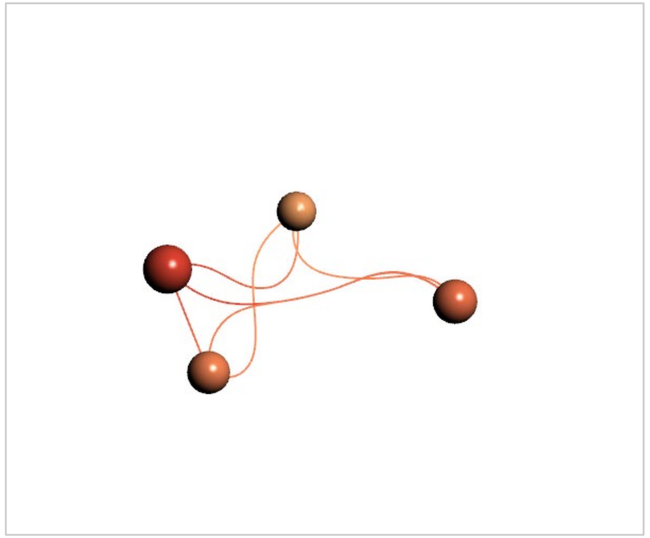


Accentuation


Regular encoding


Highlighting


Dimming


Filtering

Accentuation

- How to design V^+ and V^-?
- General rules of thumb:
  - Visual feedback should be effective: **Relevant data stands out** and can be perceived pre-attentively
  - Side effects on visualization should be minimal: Visualization can still **convey data characteristics** as intended
- Influencing factors:
  - Frequency of change: How often must visual feedback be generated?
  - Size of selection: How much of the visual representation will change?

We talked about **pre-attentive processing** in Lecture 4.

Accentuation

- If D^+ is small (e.g., mouse hovering) → Highlighting
- If D^- is small (e.g., remove outliers) → Dimming

- Filtering removes D^- entirely
  - Pro: Clear view on D^+
  - Con: No awareness of D^-
  - Filtering suitable when selection is relatively stable during a subtask

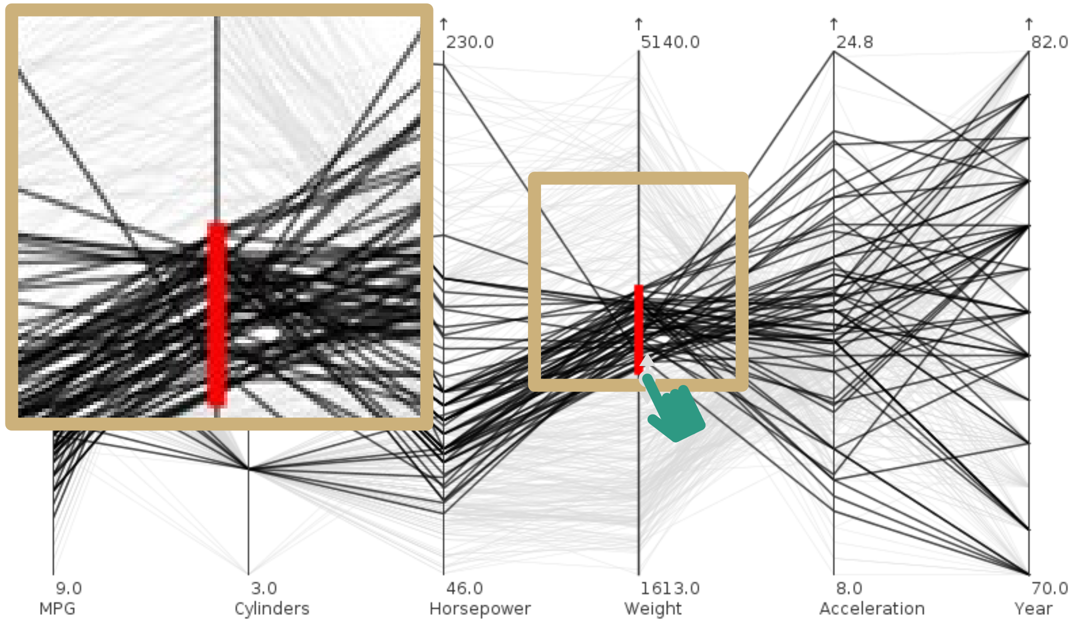
- Examples: Highlighting in CGV vs. dimming in Gephi

Enhanced selection support

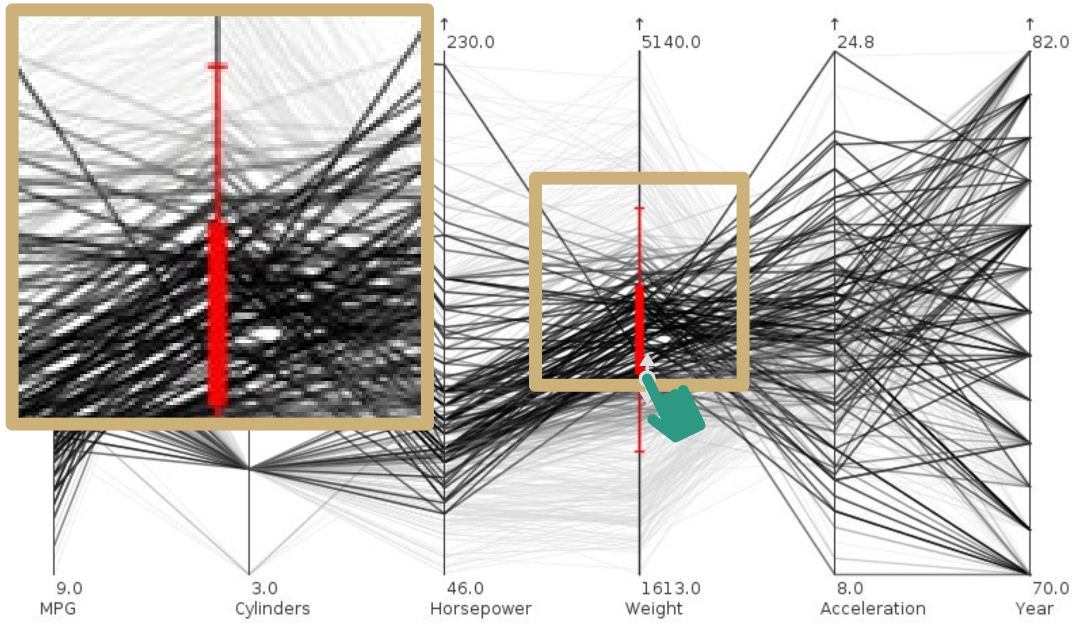
- **Smooth brushing**
  - Instead of binary-state selections, fuzzy *selectedness* in range [0,1]
- **Brushing & linking**
  - Selection in one view automatically propagated to all other views
- **Automatic selection support**
  - Auto-complete selections to reduce interaction costs

Enhanced selection support

Smooth brushing (Martin & Ward, 1995; Doleisch & Hauser, 2002)



Standard binary brushing



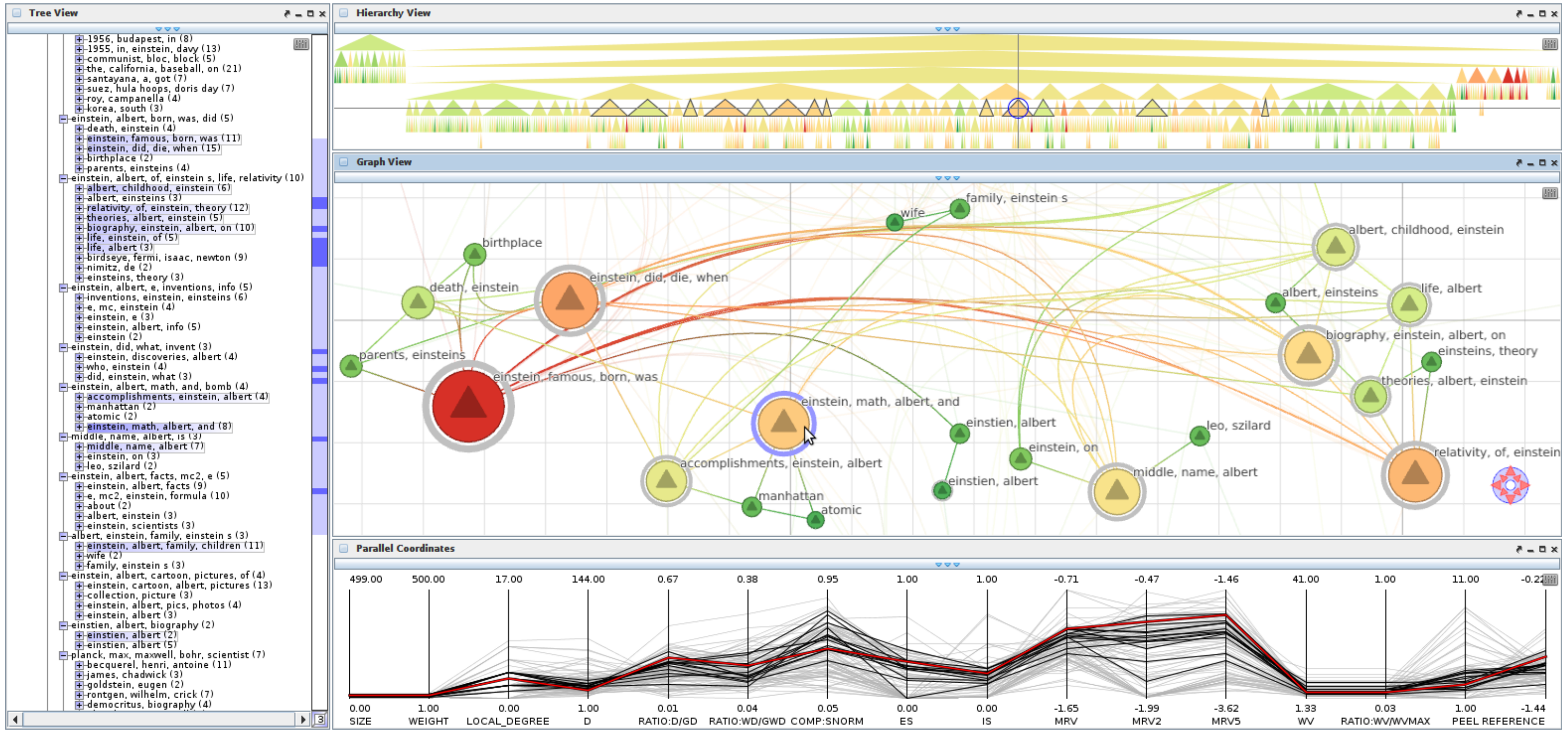
Enhanced smooth brushing

Enhanced selection support

Brushing & linking (Becker & Cleveland, 1987; Buja et al., 1991)

"Multiple views, however, should not be regarded in isolation. They need to be linked so that the information contained in individual views can be integrated into a coherent image of the data as a whole.

— Buja et al., 1991

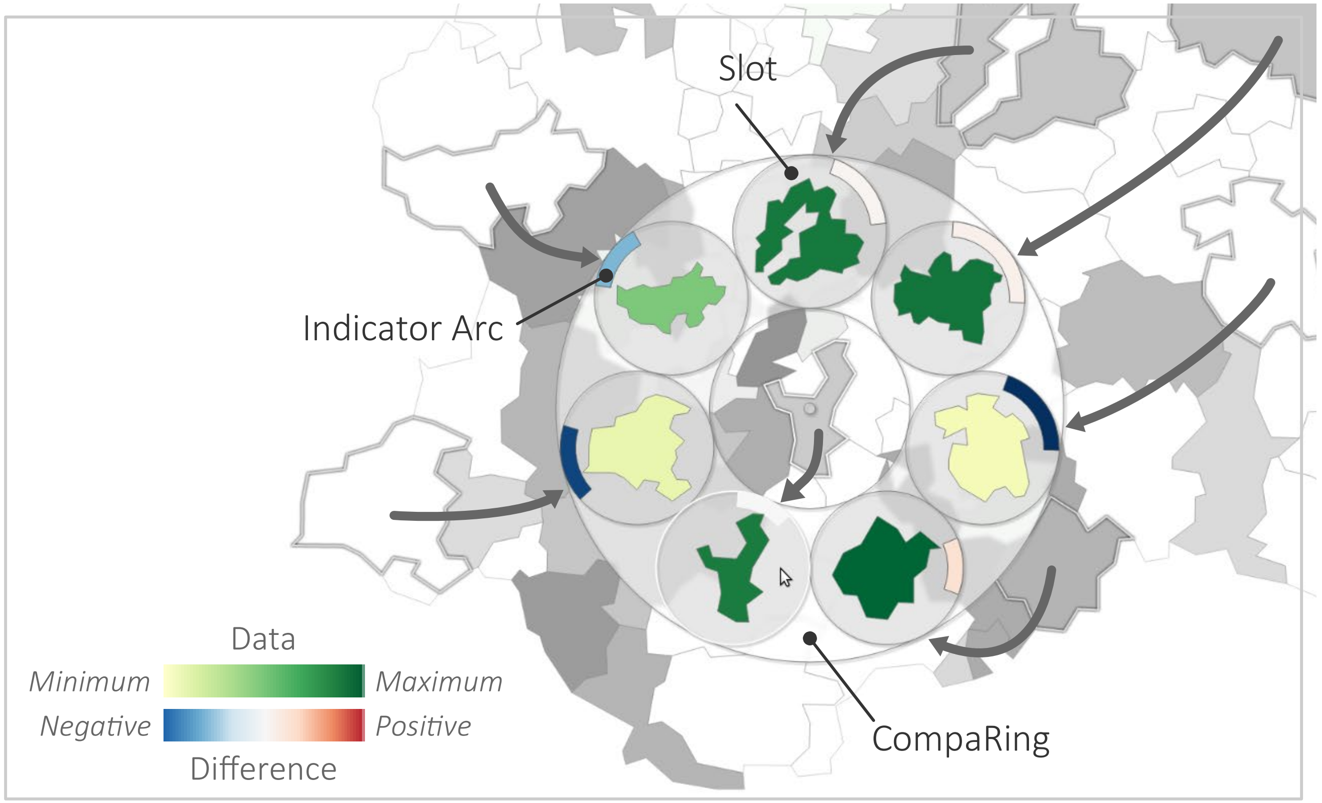


Tree View
1956, budapest, in (8)
1955, in, einstein, davy (13)
communist, bloc, block (5)
the, california, baseball, on (21)
santayana, a, got (7)
suez, hula hoops, doris day (7)
roy, campanella (4)
korea, south (3)
einstein, albert, born, was, did (5)
death, einstein (4)
einstein, famous, born, was (11)
einstein, did, die, when (15)
birthplace (2)
parents, einsteins (4)
einstein, albert, of, einstein s, life, relativity (10)
albert, childhood, einstein (6)
albert, einsteins (3)
relativity, of, einstein, theory (12)
theories, albert, einstein (5)
biography, einstein, albert, on (10)
life, einstein, of (5)
life, albert (3)
birdseye, fermi, isaac, newton (9)
nimitz, de (2)
einsteins, theory (3)
einstein, albert, e, inventions, info (5)
inventions, einstein, einsteins (6)
e, mc, einstein (4)
einstein, e (3)
einstein, albert, info (5)
einstein (2)
einstein, did, what, invent (3)
einstein, discoveries, albert (4)
who, einstein (4)
did, einstein, what (3)
einstein, albert, math, and, bomb (4)
accomplishments, einstein, albert (4)
manhattan (2)
atomic (2)
einstein, math, albert, and (8)
middle, name, albert, is (3)
middle, name, albert (7)
einstein, on (3)
leo, szilard (2)
einstein, albert, facts, mc2, e (5)
einstein, albert, facts (9)
e, mc2, einstein, formula (10)
about (2)
albert, einstein (3)
einstein, scientists (3)
albert, einstein, family, einstein s (3)
einstein, albert, family, children (11)
wife (2)
family, einstein s (3)
einstein, albert, cartoon, pictures, of (4)
einstein, cartoon, albert, pictures (13)
collection, picture (3)
einstein, albert, pics, photos (4)
einstein, albert (3)
einstien, albert, biography (2)
einstien, albert (2)
einstien, albert (5)
planck, max, maxwell, bohr, scientist (7)
becquerel, henri, antoine (11)
james, chadwick (3)
goldstein, eugen (2)
rontgen, wilhelm, crick (7)
democritus, biography (4)
Hierarchy View
Graph View
family, einstein s
wife
birthplace
albert, childhood, einstein
death, einstein
einstein, did, die, when
albert, einsteins
life, albert
biography, einstein, albert, on
parents, einsteins
einsteins, theory
einstein, famous, born, was
theories, albert, einstein
einstein, math, albert, and
einstien, albert
leo, szilard
einstein, on
relativity, of, einstein
accomplishments, einstein, albert
middle, name, albert
einstien, albert
manhattan
atomic
Parallel Coordinates
499.00
500.00
17.00
144.00
0.67
0.38
0.95
1.00
1.00
-0.71
-0.47
-1.46
41.00
1.00
11.00
-0.21
0.00
1.00
0.00
1.00
0.01
0.04
0.05
0.00
0.00
-1.65
-1.99
-3.62
1.33
0.03
1.00
-1.44
SIZE
WEIGHT
LOCAL_DEGREE
D
RATIO:D/GD
RATIO:WD/GWD
COMP:SNORM
ES
IS
MRV
MRV2
MRV5
WV
RATIO:WV/WVMAX
PEEL REFERENCE

Enhanced selection support

Automatic selection support

- Example: CompaRing
  - Manual initial selection
  - Automatically select further data elements based on similarity
  - Reduce selection costs from $O(n)$ to $O(1)$



Outline

We finished:

- Interactive selection and accentuation
  - *Mark* something as interesting
  - *Show me* something conditionally

Next we will be concerned with:

- Navigating zoomable visualizations
  - *Show me* something else
  - *Show me* more or less detail

Navigating zoomable visualizations

- How to navigate the data?
  - *Show me* something else
  - *Show me* more or less detail

The **information seeking mantra** is fundamental in interactive visual data analysis!

- Conceptual answer: **Information seeking mantra**
  - Start with overview (big-picture, but no details)
  - Zoom into subsets (details, but no overview)
  - Return to overview and navigate to different subsets at different scales
  - Repeat until satisfied

“Overview first, zoom and filter, then details-on-demand.

— Shneiderman, 1996

- Practical answer: **Zoomable interfaces** (Bederson, 2011)

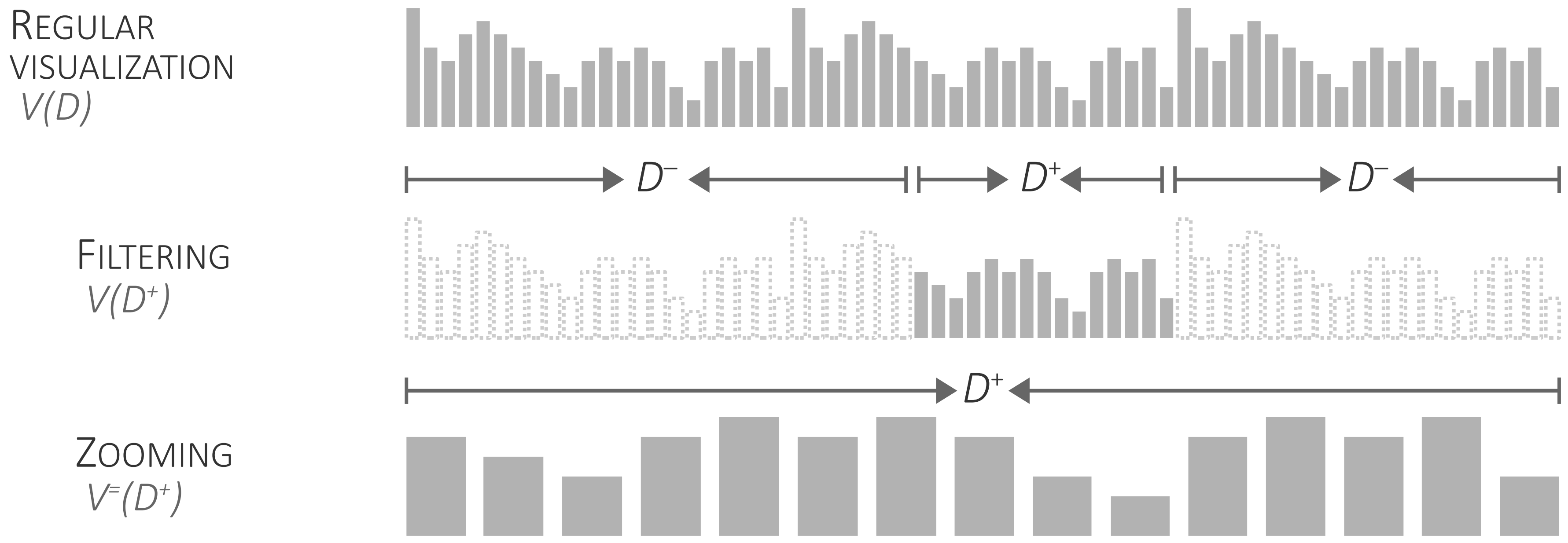
Navigating zoomable visualizations

- Information seeking mantra distinguishes **filter** and **zoom**
- What is the difference though?

- **Filtering:** $\mathrm{V}(D^{+})$ only shows the relevant data using the regular visual encoding
- **Zooming:** $V^{=}(D^{+})$ only shows the relevant data, but with a special visual encoding $V^{=}$
  - $V^{=}$ utilizes the available display space that is freed by dropping D^{-} to scale up the visual representation of D^{+}

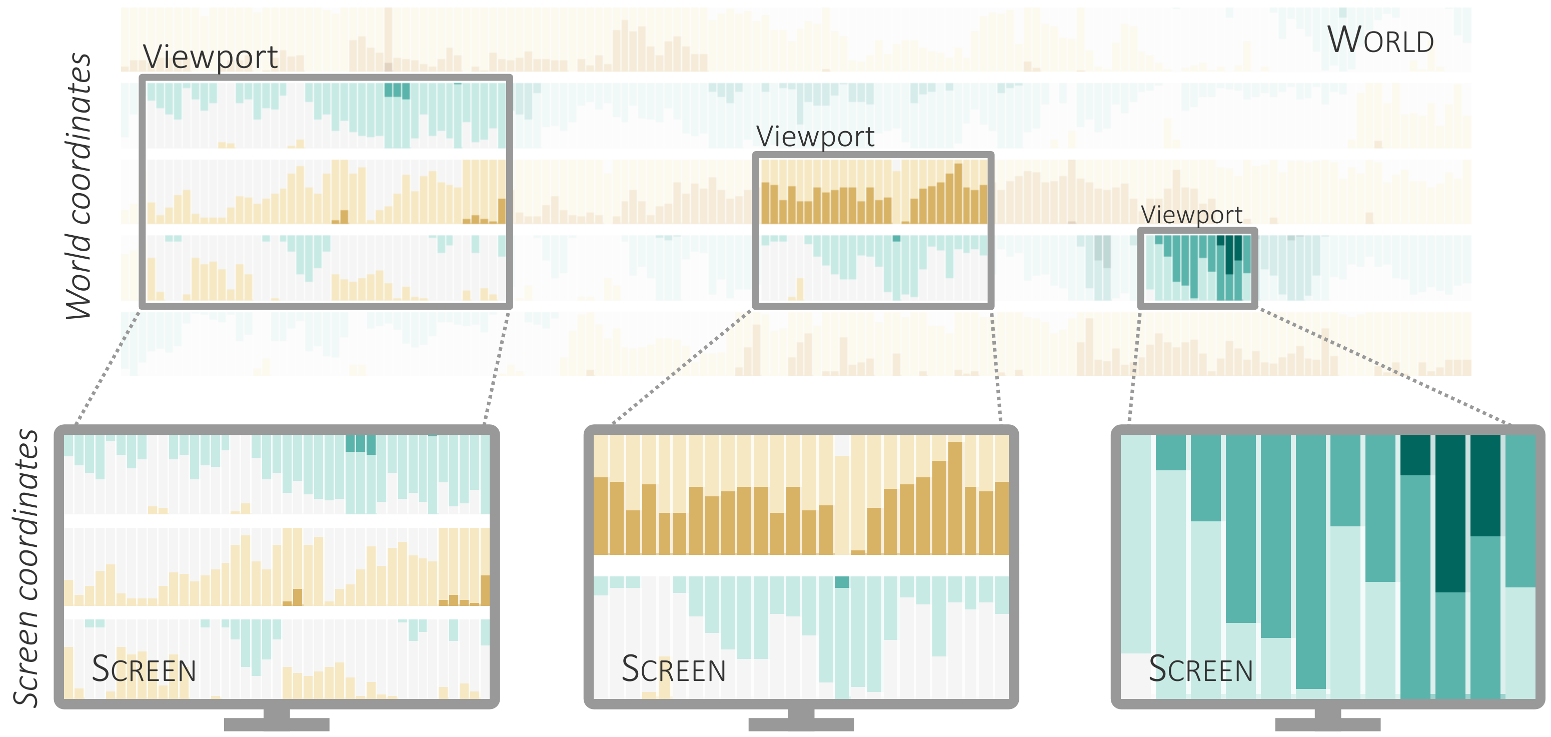
Navigating zoomable visualizations

- Information seeking mantra distinguishes **filter** and **zoom**
- What is the difference though?



Navigating zoomable visualizations

- Basic notions
  - **World:** The space in which the visualization resides
  - **Viewport:** The window through which the world is seen
  - **Screen:** The display part onto which the view is projected



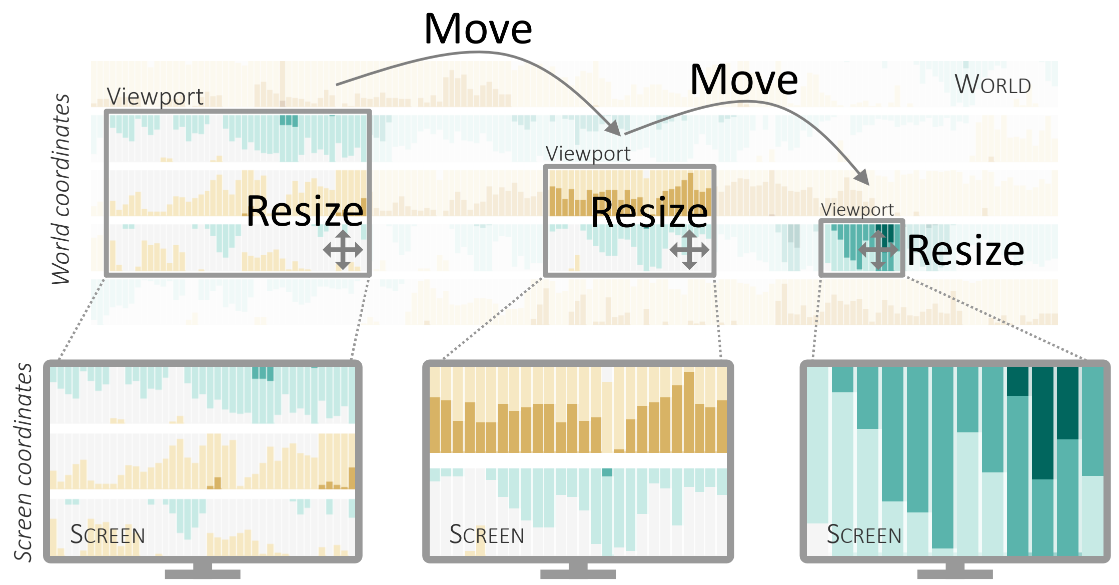
Navigating zoomable visualizations

- Basic mapping
  - **World** coordinates are **projected** to **screen** coordinates depending on the **viewport**
  - **Screen** coordinates can be **un-projected** to **world** coordinates also depending on the **viewport**

```
// Viewport (left, top, width, height) in world coordinates
// Screen (0, 0, width, height) in screen coordinates

function project(viewport, screen, wx, wy) {
    // Position in world (wx, wy) to position on screen (sx, sy)
    let sx = (wx-viewport.left) * screen.width /viewport.width;
    let sy = (wy-viewport.top)  * screen.height/viewport.height;
    return [sx, sy];
}


function unproject(viewport, screen, sx, sy) {
    // Position on screen (sx, sy) to position in world (wx, wy)
    let wx = viewport.left + sx * viewport.width /screen.width;
    let wy = viewport.top +  sy * viewport.height/screen.height;
    return [wx, wy];
}
```

Navigating zoomable visualizations

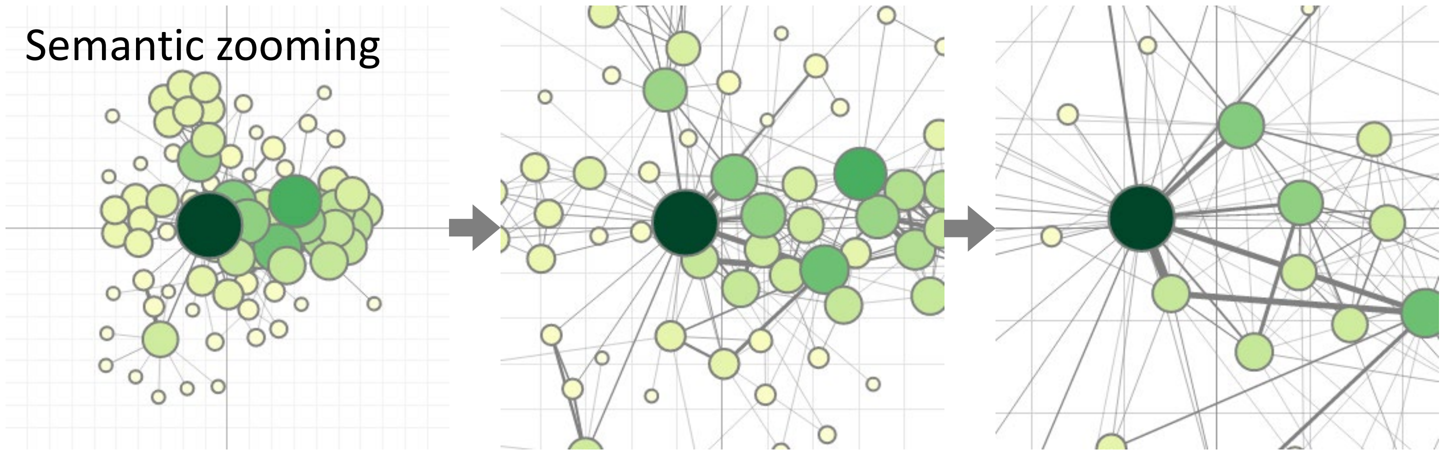
- Basic interactions
  - **Move** viewport
    - Change which part of the world is seen
  - **Resize** viewport
    - Adjust at what scale the world is seen
    - Smaller viewport → Larger zoom
    - Larger viewport → Smaller zoom
- Basic visual feedback
  - **Geometric zooming**
    - Purely geometric scaling of visual representation
  - **Semantic zooming**
    - Any kind of adjustment of the visualization

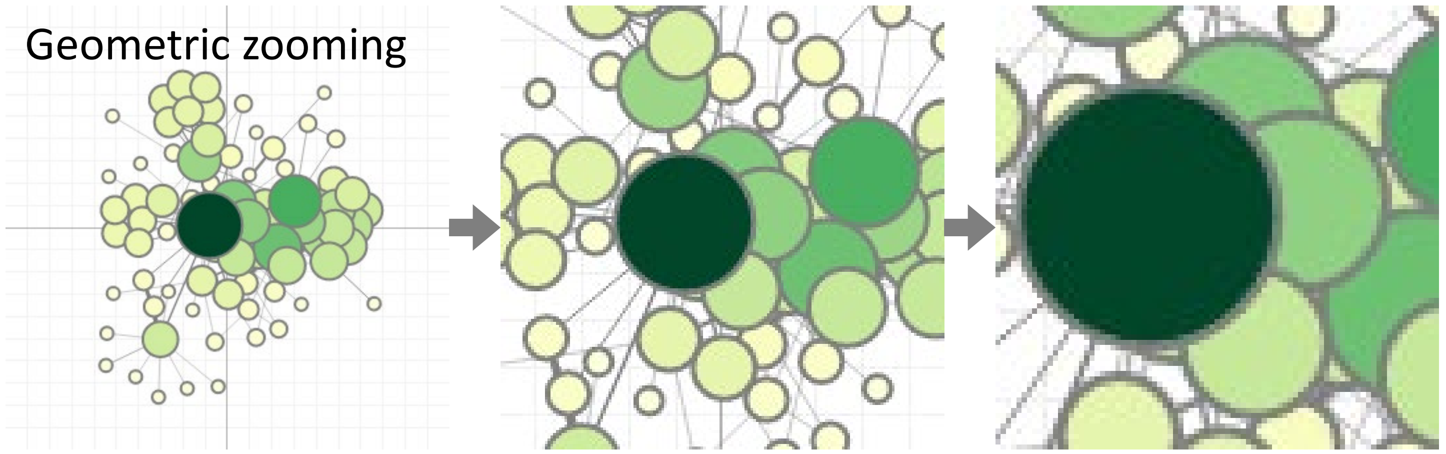


Navigating zoomable visualizations

- Think about: Which row shows
  - Geometric zooming?
  - Semantic zooming?

- Think about: What is semantic in this example?





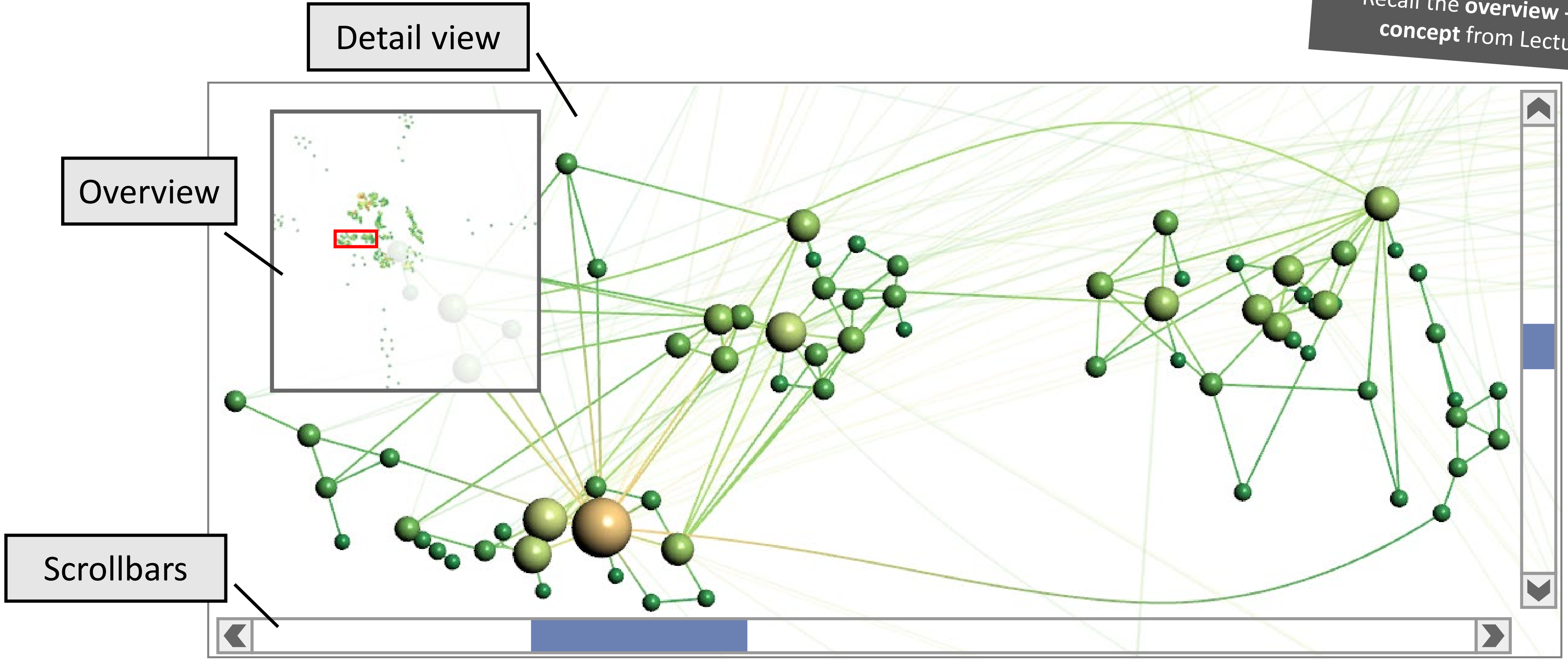
Navigating zoomable visualizations

- Basics of zooming are clear now
- Open question: How does the interface look like?

- **Visual interface** must support users in coping with three questions:
  1. **Where am I?**
     → Indicate where in the world the current viewport is located
  2. **Where can I go?**
     → Indicate what is not covered by the current viewport
  3. **How do I get there?**
     → Provide interactions to define the viewport

Navigating zoomable visualizations

Recall the **overview + detail concept** from Lecture 4.



- Think about: How are '**Where am I?**' and '**Where can I go?**' supported?

Navigating zoomable visualizations

How do I get there?

- Three operations: *Move*, *scale*, or *define* new viewport
  - **Direct interaction on the view**
    - Move view: Drag world in view (aka. *panning*)
    - Scale view: Mouse wheel, mouse drag, or pinch gesture on view
    - Define view: Draw elastic rectangle in view
  - **Manipulation of interface objects**
    - Move view: Drag scrollbars or viewport (red frame) in overview
    - Scale view: Drag edges of scrollbars or border of viewport (red frame) in overview
    - Define view: Draw elastic rectangle in overview
- Think about: What are benefits and drawbacks of the two options

Navigating zoomable visualizations

  - So far, our zoom navigation is based purely on the visual layout of the data
  - For example, panning a graph will bring us to nodes that are spatially near
  - However, what about navigating to nodes that are "near" wrt. to the graph structure or "similar" in terms of node attributes?

- Additional support needed: **Interaction aids and visual cues**
  - **Off-screen visualization:** Show (selected) information that is currently off-screen
  - **Navigation shortcuts:** Enable quick and effortless navigation to distant targets
  - **Animated view transitions:** Make view changes understandable

Interaction aids and visual cues

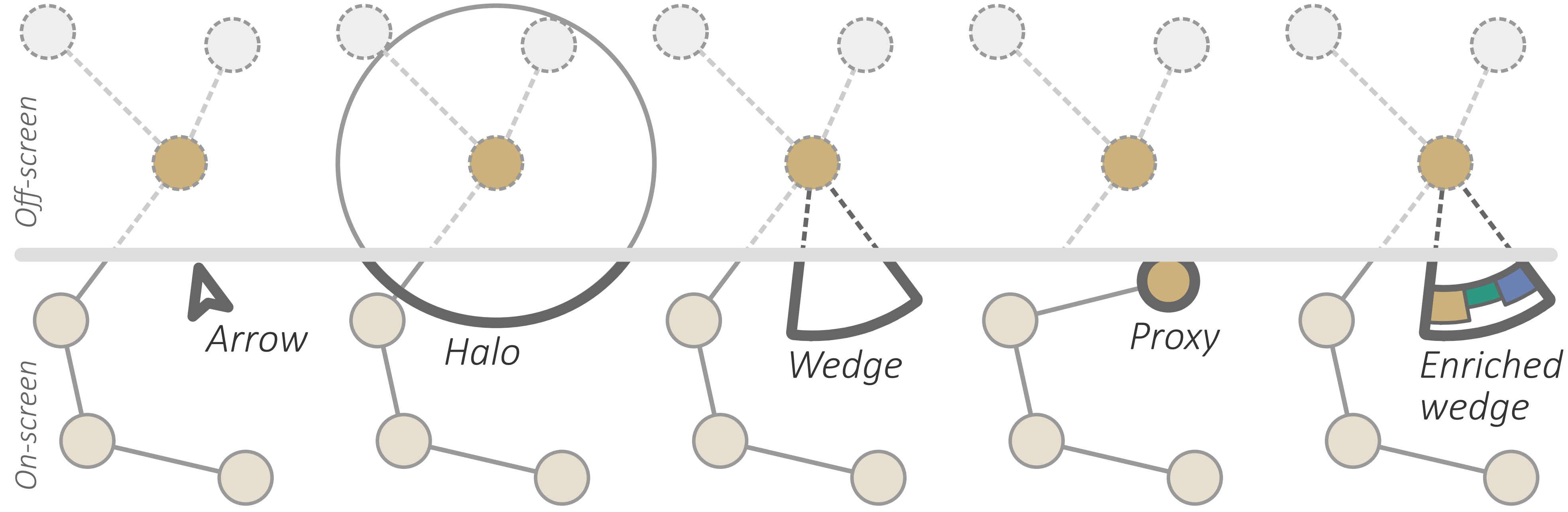
Off-screen visualization

- Focus+context approach to hint at subset of off-screen data
- Two-step procedure
  1. Determine subset of **data of interest** to be hinted at
  2. Embed **visual cues** to indicate
     - **Where:** Direction + distance where data are located
     - **What:** Attribute values of data elements
     - **Why:** Relevance of data elements

Interaction aids and visual cues

Off-screen visualization

- Examples of off-screen visualization techniques

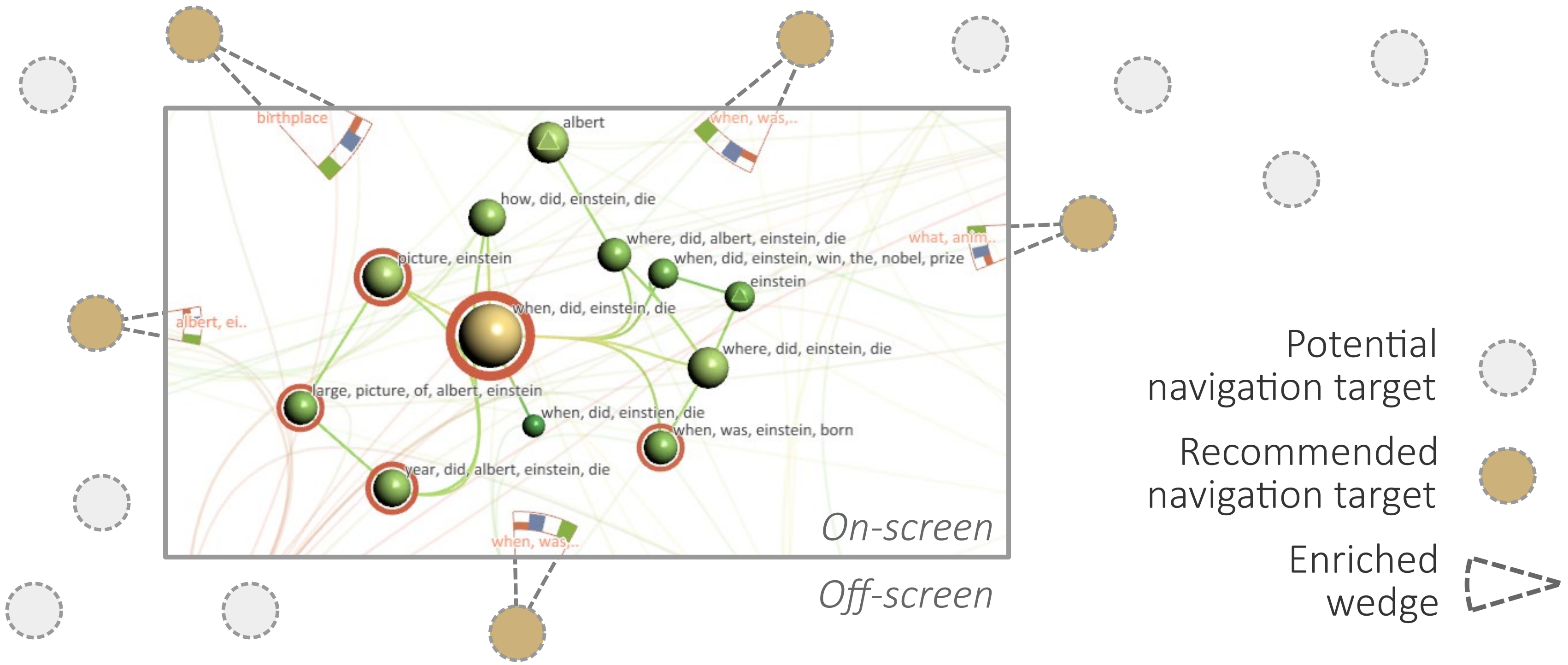


- Think about: What can the individual techniques convey (where, what, why)?

Interaction aids and visual cues

Off-screen visualization

- Navigation recommendations for graph exploration (Gladisch et al., 2013)


birthplace
albert
when, was,..
how, did, einstein, die
where, did, albert, einstein, die
what, anim..
when, did, einstein, win, the, nobel, prize
picture, einstein
einstein
when, did, einstein, die
albert, ei..
where, did, einstein, die
large, picture, of, albert, einstein
when, did, einstien, die
when, was, einstein, born
year, did, albert, einstein, die
when, was,..
On-screen
Off-screen
Potential navigation target
Recommended navigation target
Enriched wedge

Interaction aids and visual cues

- Off-screen visualization helps us see where interesting data are located, but how to get there? Manual navigation works, but can be costly for longer distances.
- Better solution:

Navigation shortcuts

- Enable quick and effortless navigation to distant targets
- Shortcut defined by
  - Target (data elements)
  - Viewport (usually centered on target and large enough to include some context)
  - Visual representation of shortcut (utilize visual cues for off-screen visualization)

Interaction aids and visual cues

Navigation shortcuts

Combine off-screen vis. and nav. shortcuts

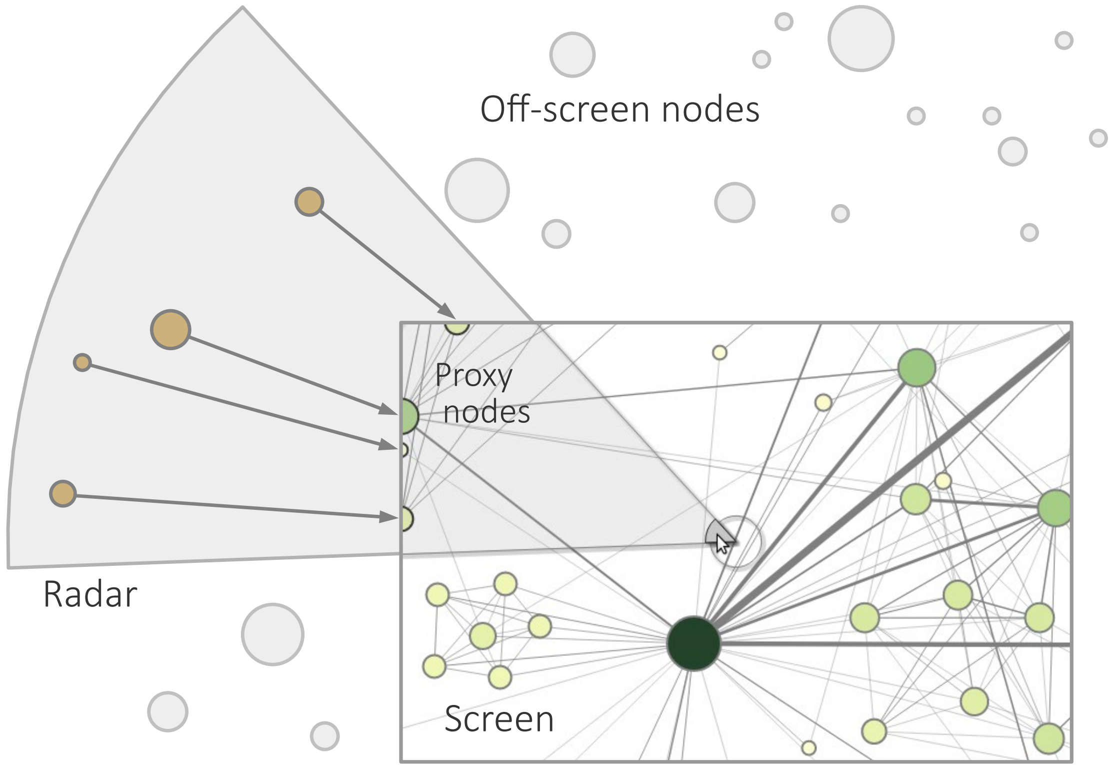
→ ***Bring & Go*** (Moscovich et al., 2010 and Tominski et al., 2010)

- **Bring**
  - Arrange off-screen data of interest so that they are visible
- **Go**
  - Enable direct navigation to the data's original location in the visualization

Interaction aids and visual cues

Navigation shortcuts

- *Bring & go example: Radar view* (Tominski et al., 2010)



Interaction aids and visual cues

Navigation shortcuts

- With navigation shortcuts, we can now easily cover larger distances
- But what type of visual feedback should we offer?
  - Static feedback: Instantaneous switch to new viewport
  - Animated feedback: Smooth transition from current to new viewport
  - Think about: What are pros and cons of the two options for viewport changes?

We introduced these two **types of visual feedback** in Lecture 9.

- Animated feedback would be helpful,
  but how to animate viewport changes?

Interaction aids and visual cues

Animated view transitions

- **Naïve approach:** Default timed animation
  - Interpolate between current viewport and new viewport
  - May include easing functions
  - Drawbacks
    - No consideration of the distinct meanings of viewport size (scale) and viewport position
    - No consideration of the distance covered by the viewport change
- **Better approach:** Dedicated smooth and efficient viewport animation

Interaction aids and visual cues

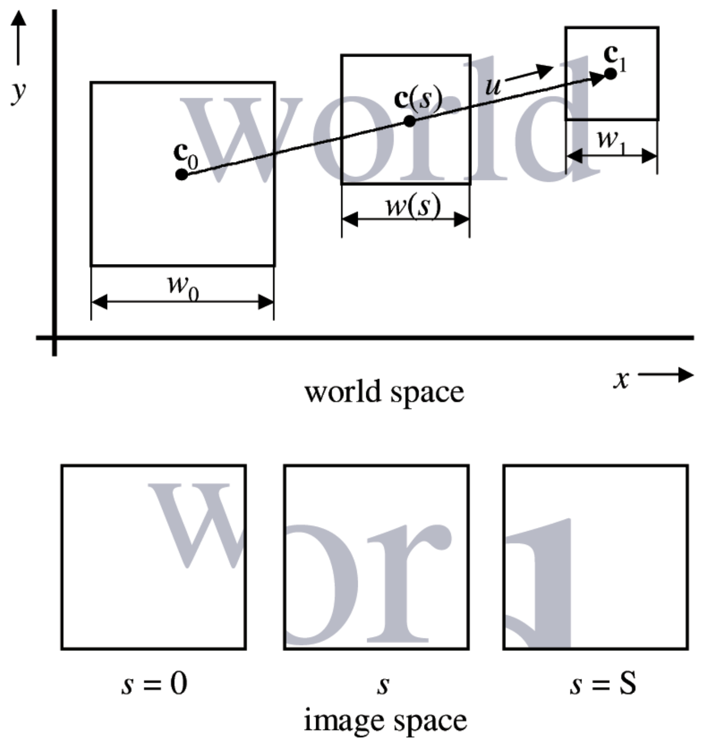
Animated view transitions

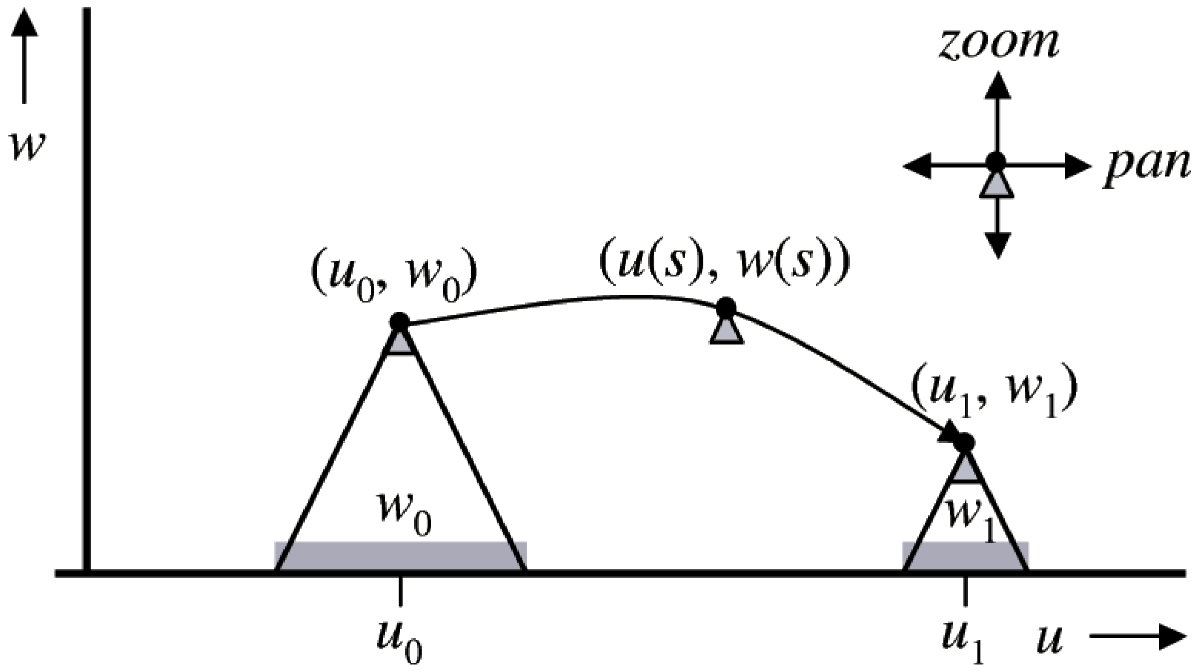
- **Smooth and efficient** viewport animation (van Wijk & Nuij, 2004)
- Three steps
  - *Zoom out* from current viewport
  - *Pan* toward new viewport
  - *Zoom in* on new viewport
- Instead of taking one step after the other, they are *smoothly intertwined*
- Optimizing the animation interpolation involves non-trivial math

Interaction aids and visual cues

Animated view transitions

- Smooth and efficient viewport animation (van Wijk & Nuij, 2004)





Interaction aids and visual cues

Animated view transitions

- Smooth and efficient viewport animation (van Wijk & Nuij, 2004)

```
let V = 1.0; // Controls the speed of the animation
let rho = 1.565; // Controls the ratio of zooming and panning
let rho2 = rho*rho, rho4 = rho2*rho2;
```

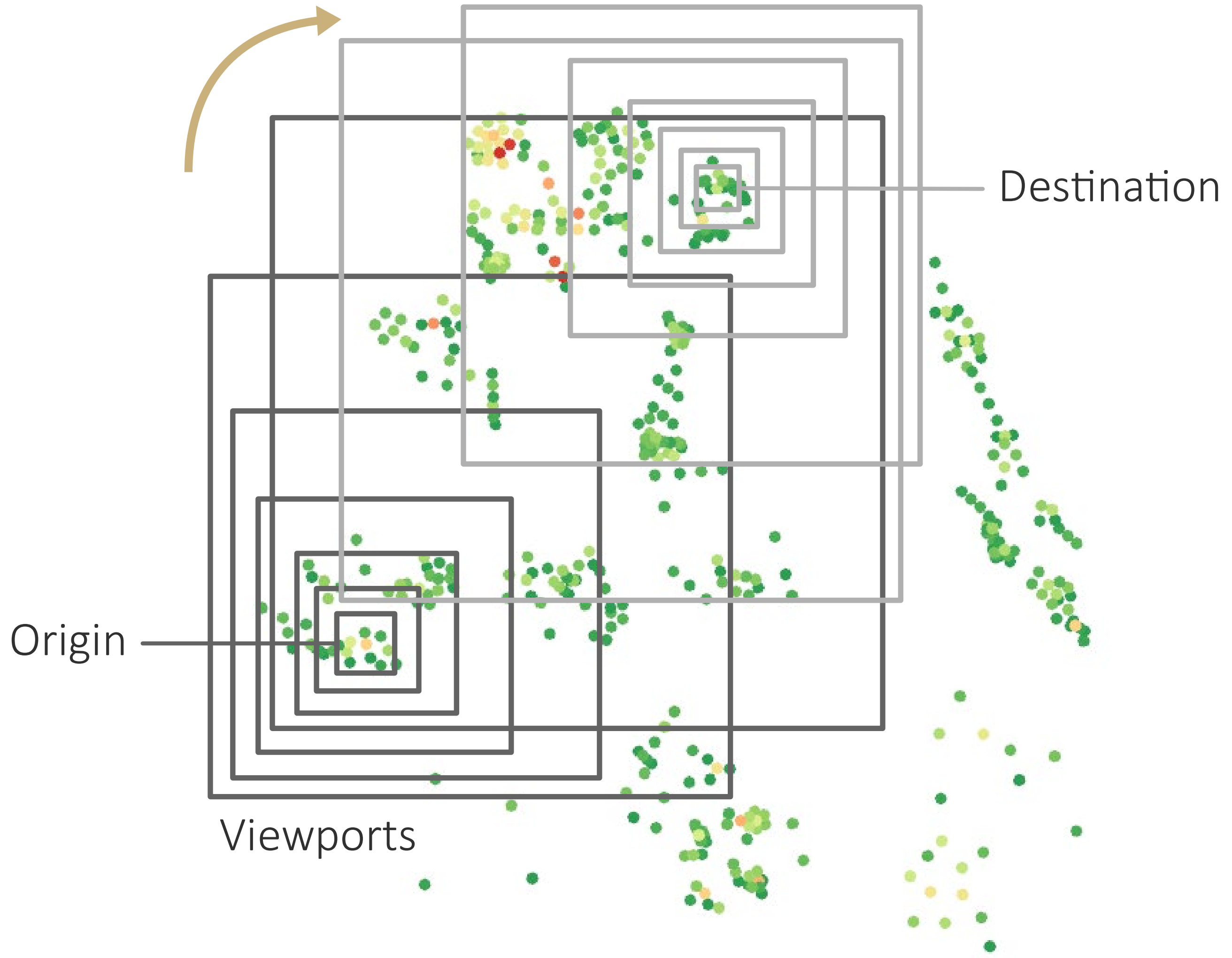
```
function Interpolator(orig, dest) {
    let dx = dest.cx-orig.cx, dy = dest.cy-orig.cy; // Distance
    let d2 = dx*dx + dy*dy, d = Math.sqrt(d2);  // orig to dest

    let u0 = 0, w0 = orig.w; // Interpolate between u0, w0
    let u1 = d, w1 = dest.w; // and u1, w1 in u-w space

    let S, i = {}; // Curve length S and interpolator object i

    if (d > 1e-6) { // Regular case
        let b0 = (w1*w1 - w0*w0 + rho4 * d2) / (2 * w0 * rho2 * d);
        let b1 = (w1*w1 - w0*w0 - rho4 * d2) / (2 * w1 * rho2 * d);
        let r0 = Math.log(Math.sqrt(b0*b0 + 1) - b0);
        let r1 = Math.log(Math.sqrt(b1*b1 + 1) - b1);
        let cr0 = Math.cosh(r0);
        let sr0 = Math.sinh(r0);

        S = (r1 - r0) / rho; // Curve length in u-w space
        i.interpolate = function (t) { // Regular case interpolation
            let s = t * S;
            let u = w0 / (rho2*d) * (cr0*Math.tanh(rho*s + r0) - sr0);
            return { // Return new viewport
                cx: orig.cx + u * dx,
                cy: orig.cy + u * dy,
                w: w0 * cr0 / Math.cosh(rho * s + r0)
            };
        }
    } else { // Special case
        ...
    }

    i.duration = S / V * 1000; // Depends on curve length and speed
    return i;
}
```

```
} else { // Special case
    let k = (w1 < w0) ? -1 : 1;
    // Curve length depends only on w0 and w1
    S = Math.abs(Math.log(w1 / w0)) / rho;
    i.interpolate = function (t) { // Special case interpolation
        let s = t * S;
        return { // Return new viewport
            cx: orig.cx + t * dx,
            cy: orig.cy + t * dy,
            w: w0 * Math.exp(k * rho * s)
        };
    }
}
```

Interaction aids and visual cues

Animated view transitions

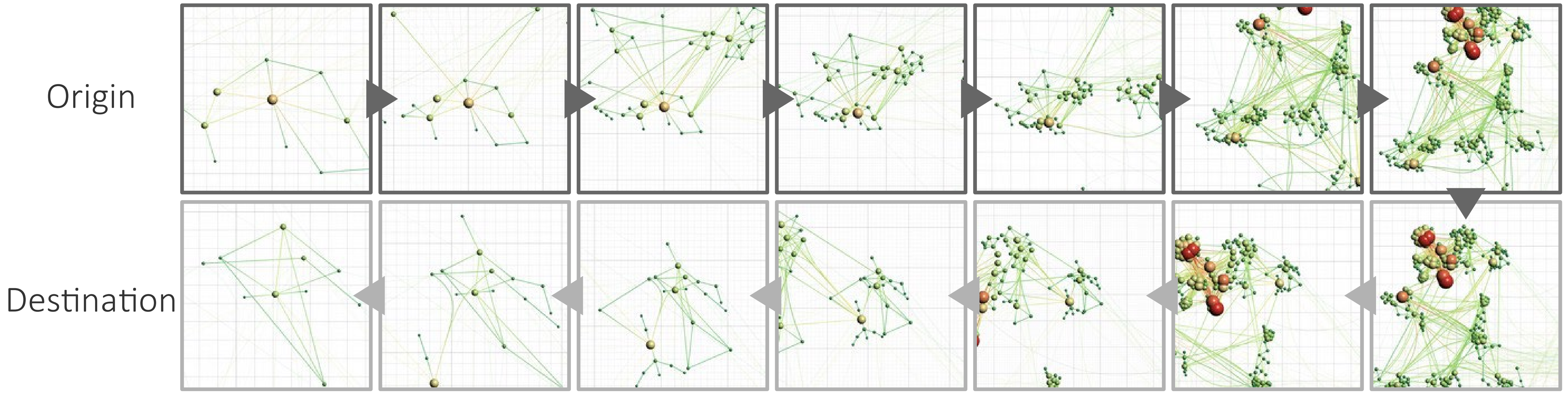
- Smooth and efficient viewport animation
  (van Wijk & Nuij, 2004)



Interaction aids and visual cues

Animated view transitions

- Smooth and efficient viewport animation (van Wijk & Nuij, 2004)



Summary

- **Basic interaction techniques to**
  - Picking, selection, accentuation
    - *Mark* something as interesting
    - *Show me* something conditionally
  - Zooming
    - *Show me* something else
    - *Show me* more or less detail

Assignments

1. Read about "The promise of zoomable user interfaces" by Ben Bederson!
2. Read about "Effective view navigation" by George Furnas!

Questions

1. What are the basic steps of pixel picking?
2. What is the computational complexity of geometry picking?
3. How can interactive selection and accentuation be modeled conceptually?
4. With respect to what characteristics do interactive brushing and dynamic querying select data?
5. Explain the difference between highlighting, dimming, and filtering!
6. What are the specifics of smooth brushing?
7. In what context is brushing & linking relevant?
8. How are the world, the viewport, and the screen related in zoomable interfaces?
9. What is semantic zooming?
10. What is off-screen visualization?
11. What is the basic idea of bring & go?
12. Indicate the three intertwined steps for smooth viewport animation!