Hierarchical Aggregation for Information Visualization: Overview, Techniques and Design Guidelines

Niklas Elmqvist, Member, IEEE, and Jean-Daniel Fekete, Member, IEEE

鄧雅文 清大服科所103078507

Objective

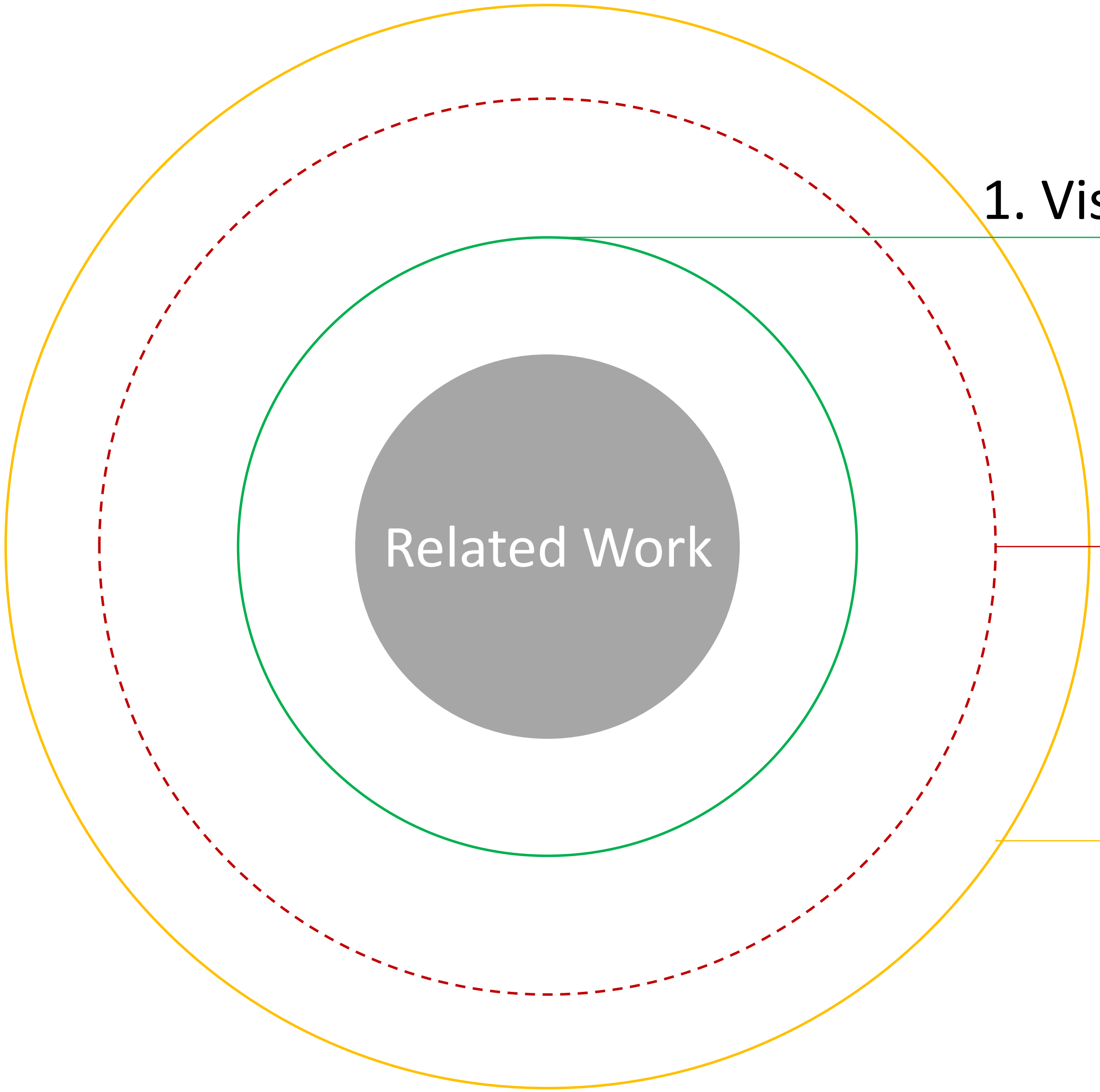
Overview is becoming increasingly difficult to effectively achieve with the ever increasing size of real-world datasets.

New visualization techniques, such as dense pixel displays, have been proposed for dealing with large datasets, but most of these approaches still attempt to draw each item in the dataset. This is not practical for massive datasets.

Problem : data overload

Present a model for transforming virtually any visualization technique into a multiscale structure using hierarchical aggregation.
Make visual representations more visually scalable and less cluttered.

Research Framework



1. Visual aggregation as a method of data abstraction.

2. Hierarchical aggregation in visualization techniques.

3. Yields a set of guidelines, exemplify the model and the design guidelines by showing how to add aggregation to a standard zoomable treemap.

Related work

Data abstraction for the purposes of reducing visual clutter and dataset size is not a novel idea.
Many authors fail to acknowledge the benefit of providing a link between abstraction methods in data space and their representation in visual space.

Drawing visual representations of abstractions performed in data space allows for creating simplified versions of a visualization while still retaining the general overview

By dynamically changing the abstraction parameters, the user can also retrieve details-on-demand.

Hierarchical Aggregation for Visualization

“overview first, zoom and filter, then details on demand.”

Our model for hierarchical aggregation in visualization is based on aggregation in data space and corresponding simplified visual representations of the aggregates in visual space.

Hierarchical Aggregation for Visualization-Terminology

Aggregate hierarchy	Mean a grouping of the original data items into a hierarchical structure of data aggregates, each representing their children.
Data aggregates	Data items into a hierarchical structure.
Visual aggregates	The graphical depictions of data aggregates.
Visual items	Graphical representations of data items.
Visual entity	As a common term for both aggregates and items, we use visual entity.
Visual entity budget	An upper limit of the number of visual entities to render for a particular hierarchically aggregated visualization.

Hierarchical Aggregation for Visualization- Aggregation

Given a set of data items, hierarchical aggregation is based on iteratively building a tree of aggregate items either bottom-up or top-down.
Each aggregate item consists of one or more children; either the original data items (leaves) or aggregate items (nodes). The root of the tree is an aggregate item that represents the whole dataset.
There are several specific algorithms to perform aggregation.
The most common are clustering approaches, such as graph-based or k- means clustering.

The aggregate tree becomes a multiscale structure for controlling the current level-of-detail of the visualization on the screen. Depending on the visualization technique, the visual aggregate can also convey information about the underlying data items

Visual Representation

Standard visualization	Define a visual representation for **individual data items** (henceforth called visual data items)
Hierarchical aggregated visualization	Must also define a visual representation for **data aggregates**. This visual aggregate should convey something about the underlying data aggregation that the entity captures. It should also be distinguishable from visual data items.

Visual Representation

Here are some examples of information that can be conveyed through visual aggregates:

1. **Count/sum**: The number or the sum of the aggregated data items.
2. **Average:** The arithmetic mean of the underlying data items.
3. **Mode:** The most frequent value in the underlying data set.
4. **Extents:** The extrema (minimum and maximum) of the underlying data items.
5. **Median:** The value that divides the underlying data into two,equal-sized subsets.
6. **Percentiles:** The 25th, 50th (median), and 75th percentiles for the underlying data.
7. **Distribution:** The full distribution of the underlying data.

Visual Representation

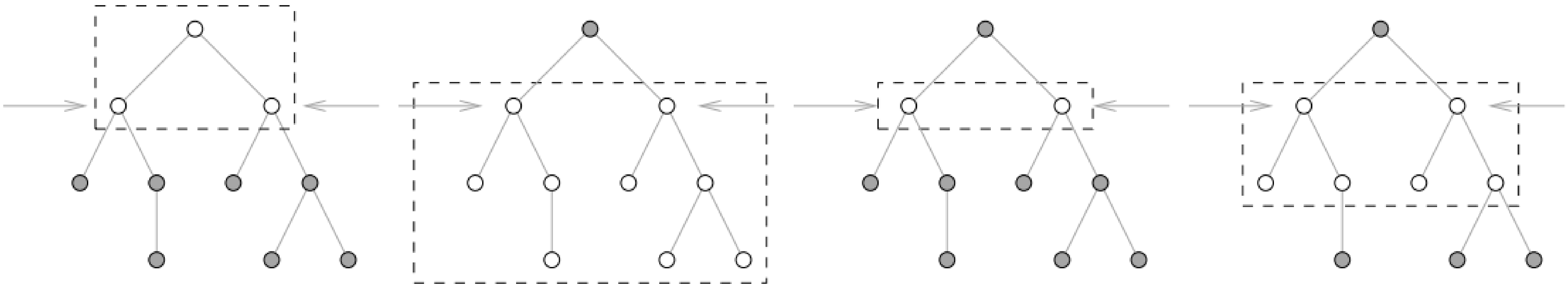
An investigation into ways to rebalance CitiBike stations throughout New York City.

Red : Destination.

Blue : Origin.

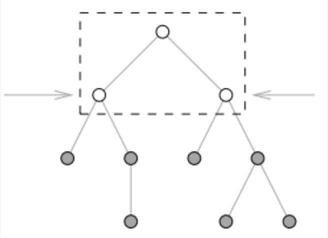
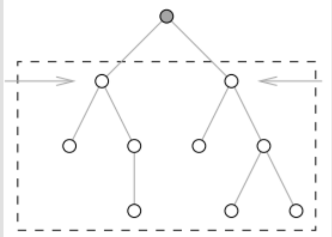
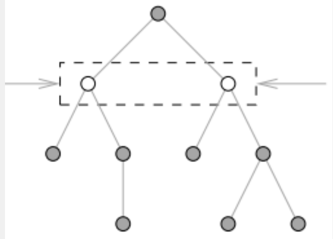
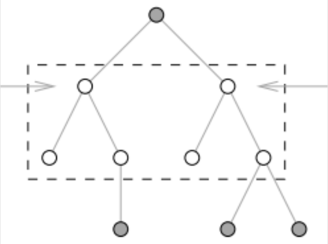
Visual aggregates

Rendering



(a) Above traversal. (b) Below traversal. (c) Level traversal. (d) Range traversal.

Rendering a hierarchical aggregated visualization amounts to traversing the visual aggregate hierarchy. The traversal should typically be breadth-first to resolve depth ordering so that higher-level aggregates are correctly occluded by lower (more detailed) levels. This also allows for aborting the rendering at any stage.
Depending on the current state of the visualization, there are four main types of rendering traversals.

 (a) Above traversal.	All nodes above (and including) the current height are rendered. Here, data items are abstracted by high level visual aggregates, hiding details to avoid perceptual and technical overload
 (b) Below traversal.	All nodes below (and including) the current height are rendered. Less common.
 (c) Level traversal.	All nodes on the same level as the current height are rendered. This gives a snapshot of the data abstraction at any specific detail level.
 (d) Range traversal.	All nodes in an interval of levels are rendered. As for level traversal, range traversals give a snapshot of the data abstraction at a specific level, as well as some additional detail on the underlying aggregate hierarchy

Interaction

One of the main benefits of hierarchical aggregated visualizations is that they can be made to support the same set of basic interaction techniques for navigating and manipulating the aggregate hierarchy.

For each of the interaction techniques in this section, we will use a combination of space-scale diagrams and the aggregate hierarchy to describe the mechanics of the interaction.

While this article mainly focuses on developing new visualization techniques that support hierarchical aggregation, the space is also open for designing new interaction techniques for interacting with them.

Interaction

Zoom and Pan – Women in science

In a way, zoom and pan operations can be seen as spatial filters that control which visual data entities to display on the screen as well as the screen space allocated to each entity.

These operations are typically used to take a closer look at parts of a visualization without actually revealing more detail in the data.

http://www.uis.unesco.org/_LAYOUTS/UNESCO/women-in-science/index.html#overview!view=map

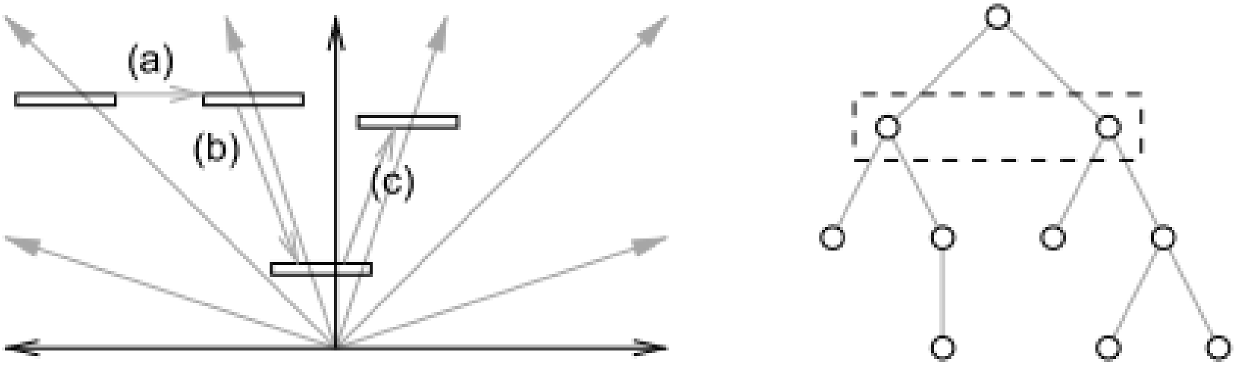
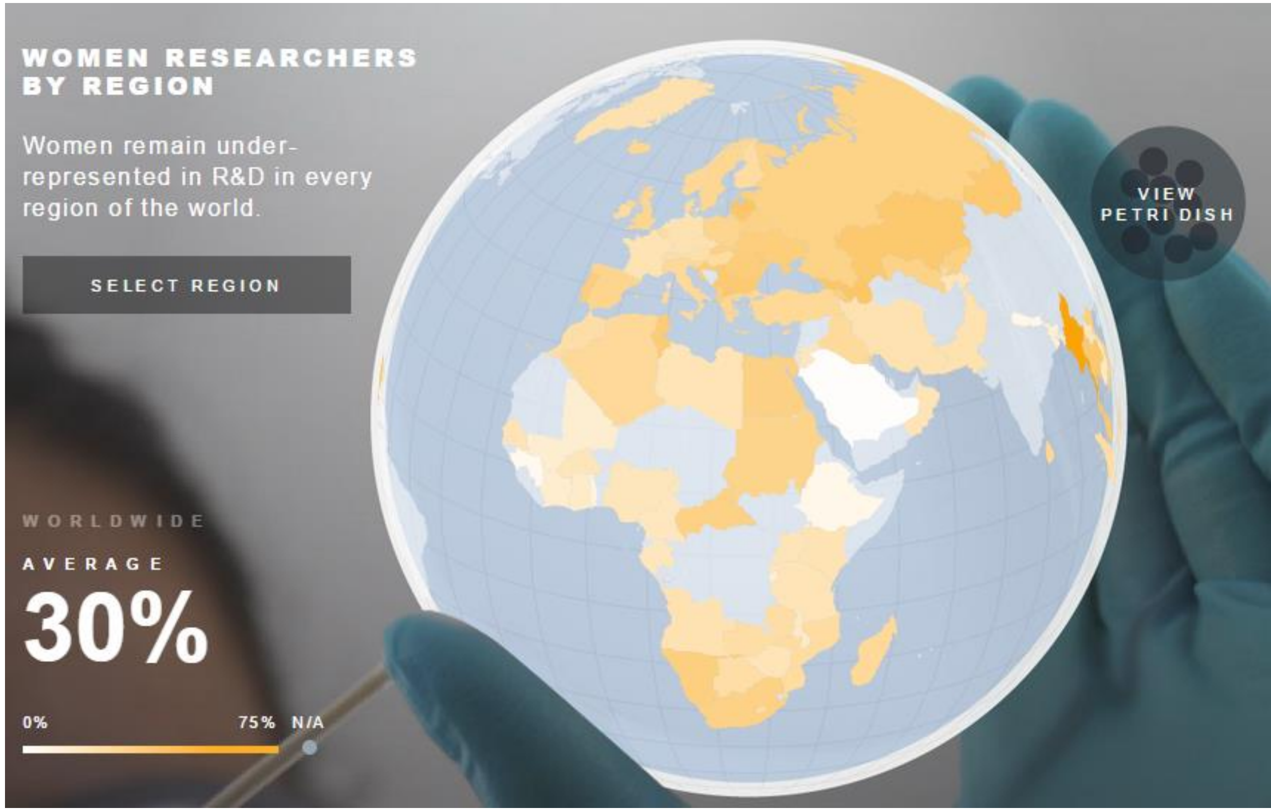


Fig. 3. Zoom and pan interaction. (a) Pure pan. (b) Zoom out to show two items. (c) Combined zoom in and pan.



Interaction

Drill-down and Roll-up-WikiGalaxy

the drill-down and roll-up operations govern the level of detail at which the data is displayed. In other words, **drill-down moves the rendering traversal deeper into the aggregate hierarchy, showing increasing amounts of detail**, whereas roll-up moves up in the hierarchy, showing less detail.

http://wiki.polyfra.me/#

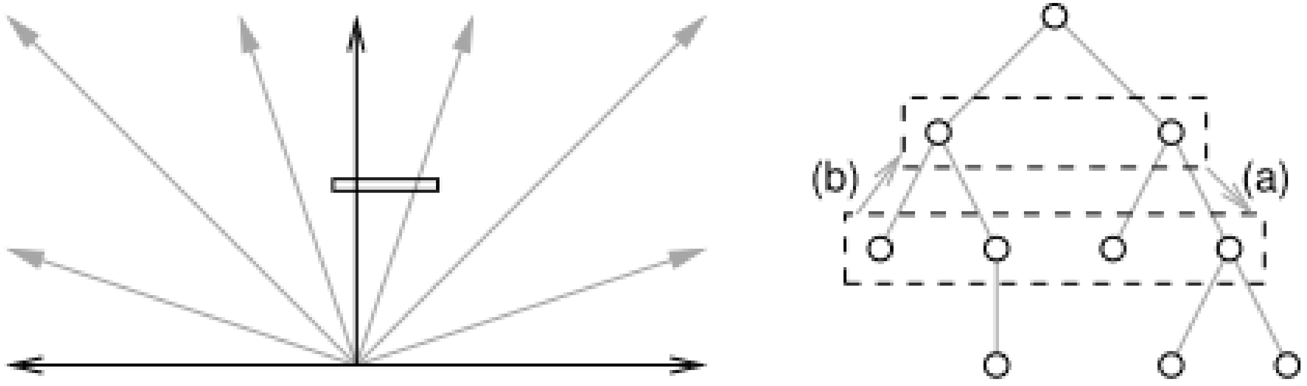
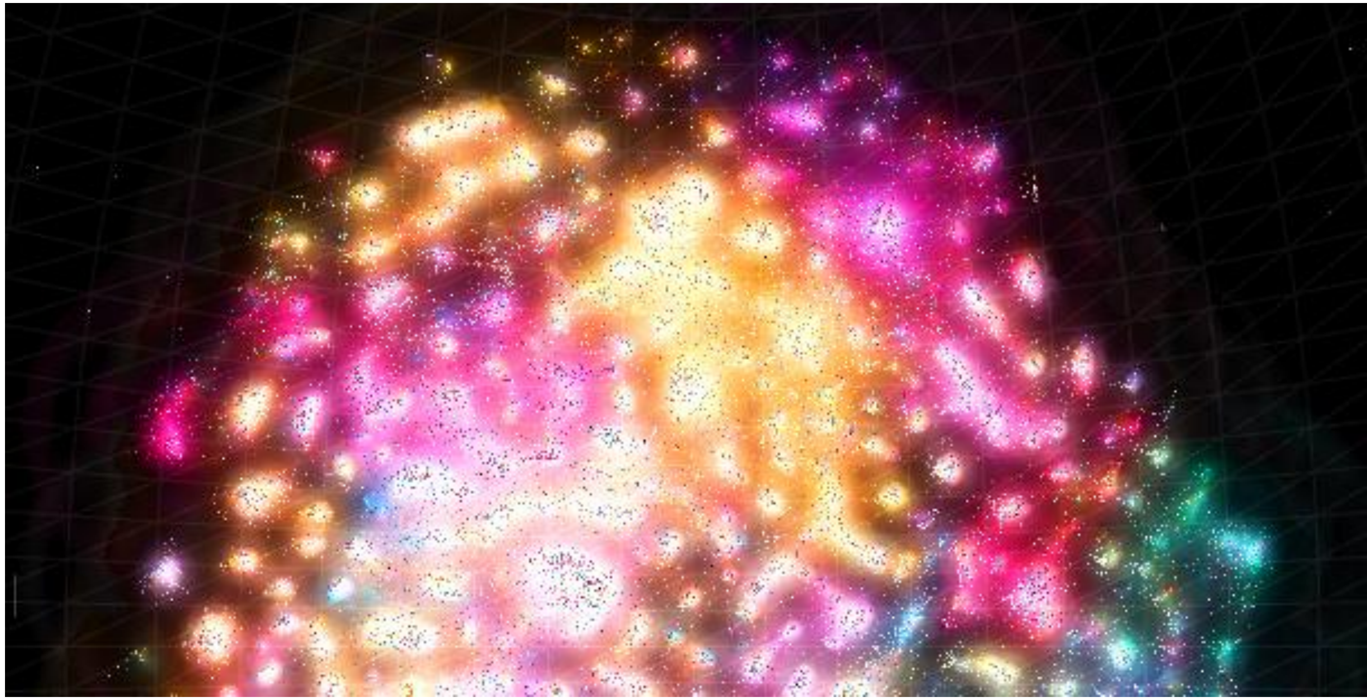


Fig. 4. Drill-down and roll-up interaction. (a) Drill-down from level 1 to level 2. (b) Roll-up from level 2 to level 1.



Interaction

Local Aggregation Control-
Most Valuable Players

As discussed in Section 3.4, it is sometimes useful to perform unbalanced rendering of the aggregate hierarchy to reveal different amounts of detail for different regions of the visualization **in a focus+context fashion**. Local aggregation control gives the user control over this process in different ways.

Fading out unselected regions in favor of selected regions.

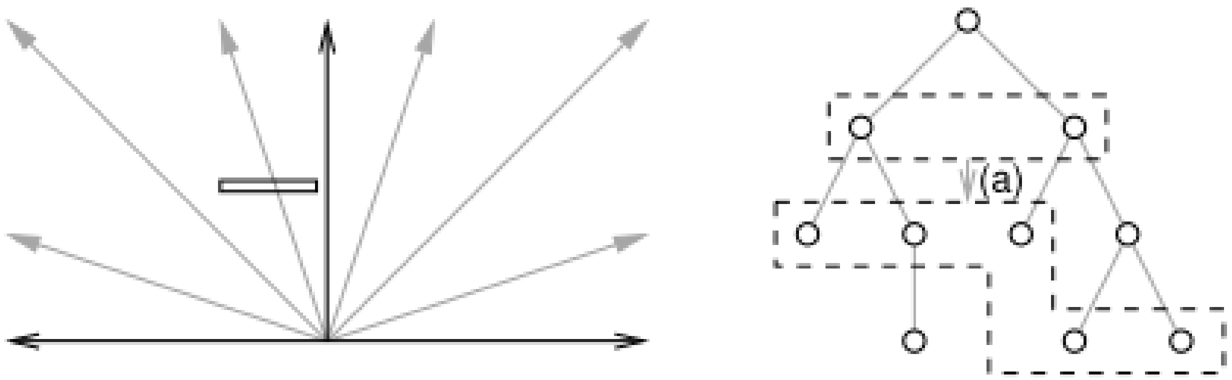
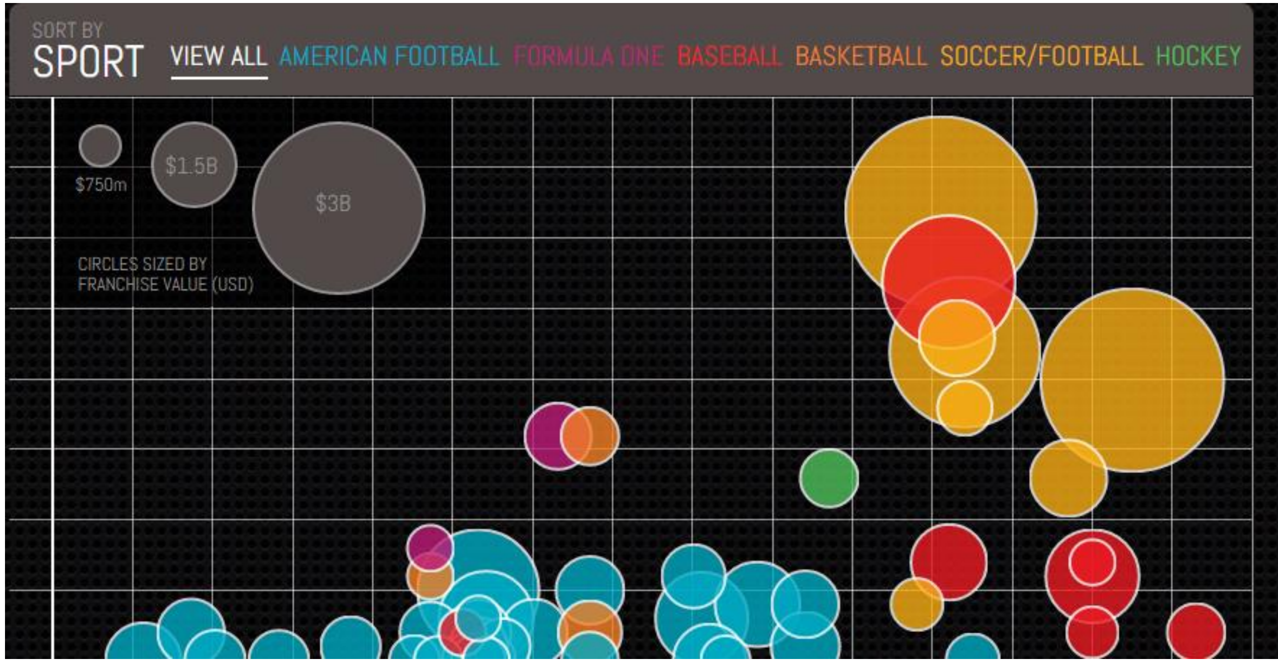


Fig. 5. Local aggregation control. (a) Unbalanced drill-down.



http://mvp.columnfivemedia.com/

Interaction

Flipping- Secret life of the cat

Fipping lets the user visit neighboring siblings in the aggregate hierarchy. Note that the effect of a flip operation is a geometrical pan because **the operation changes the view of the visualization and not the aggregation level.**

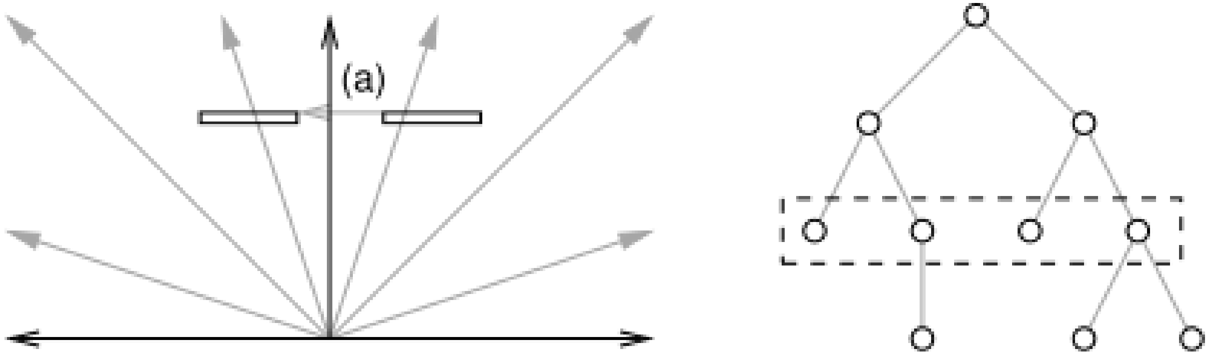


Fig. 6. Flipping interaction. (a) Flip between adjacent siblings.

http://www.bbc.com/news/science-environment-22567526

Interaction

Coupled Zooming and Drilling- Flooding and flood zones.

Coupled zooming and drilling operation ensures that zooming into a specific part of a visualization will cause the rendering to traverse deeper into the aggregate hierarchy, **yielding more detail.**
User or the system maintains a size tolerance for visual entities, and traverses deeper into the **tree until each visual entity is within this tolerance.**

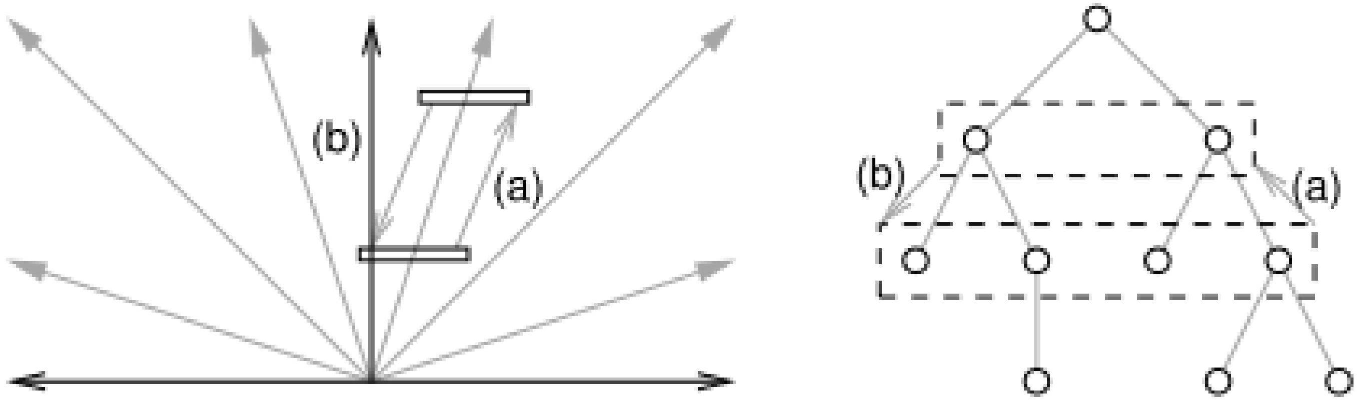
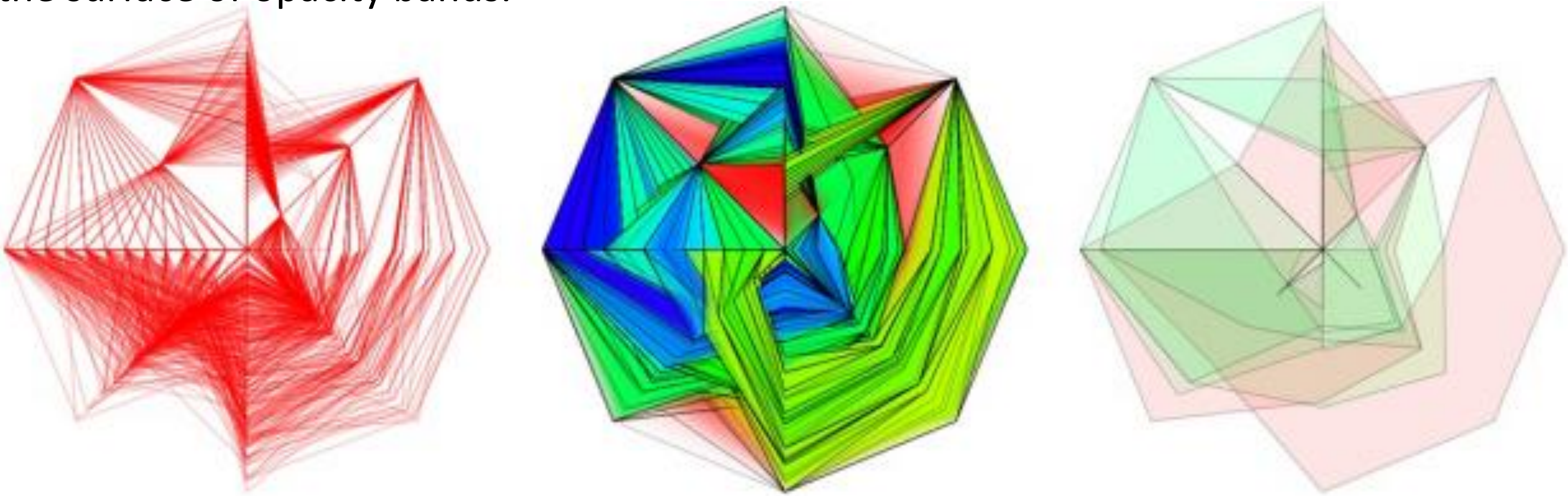


Fig. 7. Coupled zoom and drill interaction. (a) Drill-down and zoom in. (b) Roll-up and zoom out.

http://project.wnyc.org/flooding-sandy-new/#12.00/40.7794/-73.9613

Examples- Starplot visualization

Both lines and bands can be trivially extended to starplot diagrams. Recent work on starplot diagrams introduced the of color histograms showing the data distribution on the surface of opacity bands.



(a) Starplot visualization. (b) Opacity band aggregation (above-traversal). (c) Opacity band aggregation (level-traversal).

Fig. 10. Starplot visualization of an 8-dimensional dataset with opacity band [34] visual aggregations. Transparency indicates the extents and the averages for each opacity band.

Design Guideline

We now derive a set of general guidelines for designing and implementing such techniques.

G1	Entity budget	Maintain an entity budget.
G2	Visual summary	Aggregates should convey information about underlying data.
G3	Visual simplicity	Aggregates should be clean and simple.
G4	Discriminability	Aggregates should be distinguishable from data items.
G5	Fidelity	Beware that abstractions may lie.
G6	Interpretability	Aggregate items only so much so that the aggregation is still correctly interpretable within the visual mapping.

Entity budget - Maintain an entity budget.
The added benefit of a visual item budget is that it limits time spent on rendering and guarantees a lower bound on the framerate.

Visual summary - should convey information about underlying data.
This is one of the most basic features of a hierarchically aggregated visualization: the viewer can gain a reasonably detailed overview of a dataset while not being overloaded by the full data.

Visual simplicity - Aggregates should be clean and simple.
The purpose of aggregated visualizations is to reduce clutter in order to improve overview. Remember that even visual aggregates may be rendered in large quantities, so choosing a visual representation that is too complex or attempts to convey too much information may cause clutter in the visualization all over again.

Discriminability - Aggregates should be distinguishable from data items.
Visual aggregates and visual data items must be easy to differentiate in order to not convey the wrong information to the viewer. While using the same visual representation for both aggregates as for items is often easiest, it may give the viewer the wrong idea about the data.

Fidelity - Counteract fidelity problems in visual aggregates.
Data abstraction operations simplify datasets, but inherent in this process is also a loss of fidelity—the visualization may, in effect, lie about the size of the effect.
Or, as Andrienko and Andrienko remark, "the mean surface temperature on the Moon may seem quite comfortable, but the actual temperature ranges from -230°C to $+130^\circ$C, and this should be taken into account in designing clothes for astronauts."

Interpretability - Aggregate items only so much so that the aggregation is still correctly interpretable within the visual mapping.
Also relevant in the interpretability discussion is the fact that visual aggregates should be designed so that they communicate to the viewer that there is more to see by expanding aggregated nodes.

Conclusion and Future work

Conclusion

1. Present a model for hierarchical aggregation in information visualization for the purpose of improving overview and scalability of large-scale visualization.
2. Described a set of general interaction techniques for manipulating aggregated visualizations.
3. Formulate a set of common design guidelines for building new aggregated visualization techniques as well as extending existing ones with hierarchical aggregation.

Future work

Refining the model further and deriving new interaction techniques that are common to this whole class of aggregated visualizations.

The End.