



City Research Online

City, University of London Institutional Repository

Citation: Slingsby, A., Dykes, J. & Wood, J. (2009). Configuring Hierarchical Layouts to Address Research Questions. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), pp. 977-984. doi: 10.1109/tvcg.2009.128

This is the unspecified version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/158/>

Link to published version: <https://doi.org/10.1109/tvcg.2009.128>

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Configuring Hierarchical Layouts to Address Research Questions

Aidan Slingsby, Jason Dykes, and Jo Wood, *Member, IEEE*

Abstract—We explore the effects of selecting alternative layouts in hierarchical displays that show multiple aspects of large multivariate datasets, including spatial and temporal characteristics. Hierarchical displays of this type condition a dataset by multiple discrete variable values, creating nested graphical summaries of the resulting subsets in which size, shape and colour can be used to show subset properties. These ‘small multiples’ are ordered by the conditioning variable values and are laid out hierarchically using dimensional stacking. Crucially, we consider the use of *different layouts at different hierarchical levels*, so that the coordinates of the plane can be used more effectively to draw attention to trends and anomalies in the data. We argue that these layouts should be informed by the *type of conditioning variable* and by the *research question* being explored. We focus on space-filling rectangular layouts that provide data-dense and rich overviews of data to address research questions posed in our exploratory analysis of spatial and temporal aspects of property sales in London. We develop a *notation* (‘HiVE’) that describes visualisation and layout states and provides reconfiguration operators, *demonstrate* its use for reconfiguring layouts to pursue research questions and provide *guidelines* for this process. We demonstrate how layouts can be related through animated transitions to reduce the cognitive load associated with their reconfiguration whilst supporting the exploratory process.

Index Terms—Geovisualization, hierarchical, layout, guidelines, exploratory, notation.

1 INTRODUCTION

Hierarchical layouts can be used to generate rich views of multivariate data by conditioning datasets into multiple facets at different hierarchical levels and then presenting nested visual summaries of each facet. For example, a map of the world coloured by population in which each country contains its population age-structure as a pie-chart by age category can be considered as a two-level hierarchy. The first level (`country`) takes population data, conditions them by country and displays them in a Cartesian geographical layout using colour to show population. The second level (`ageGroup`) conditions each country’s population by age group, displaying them using polar coordinates nested within the first level. If we were interested in the gender balance for each country, we might consider placing a pair of pie-charts in each country by inserting `gender` into the hierarchy with the effect of conditioning population by gender followed by age category. If we were interested in studying the geographical distribution of age categories of population, we might move `ageGroup` to the base of the hierarchy, resulting in a pie-chart whose slices contain country maps coloured by population and conditioned by the age group of the corresponding pie chart slice. Since pie-chart slices are likely to be inappropriate containers within which to show maps, we may wish to change the layout to a more appropriate tabular layout in which cell contains a map (as a small multiple) for its age group. In terms of attribute hierarchy, we have gone from `country-ageGroup` to `country-gender-ageGroup` to `ageGroup-country-gender`, choosing layouts in response to the nature of the data and angle of the research in each case.

This example illustrates how we often implicitly, yet informally use hierarchical graphics, reconfiguring layouts and hierarchies to address particular tasks that relate to particular messages or questions in hand. We propose that if we are explicit in considering these as *hierarchical representations* with the *reconfigurable properties* of hierarchy, layout (at each level) and other visual variables such as colour – we can navigate the design and data spaces using these layouts and make visualisation design decisions from a more informed perspective to address

particular research questions.

Friendly and Kwan [11] demonstrate the importance of display configurations being related to the task in hand and call for more research in this area. We respond to this call by exploring the effects of selecting layouts for addressing research questions. We develop ‘HiVE (Hierarchical Visualisation Expression)’ – a notation for describing visualisation states and their reconfiguration – and demonstrate its use through the exploration of a 1.25 million record dataset of property transactions. We propose layout guidelines based on this example and show how well-designed interactions and animated transitions can be used to support visual exploratory analysis.

2 MULTIVARIATE DATA AND HIERARCHIES

The example above uses more than one attribute of population data to condition data into subsets – e.g. `US-male-over60` (conditioned by country, then gender, then age group) – summarised using colour. This is a well-established approach to handling multivariate data and is the basis of online analytical processing using OLAP cubes [13]. The resulting conditioned subsets are often presented as trellis displays [2], multi-panelled displays in which each panel contains a conditioned subset of the data; e.g. small multiples [33], scatterplot matrices [7] or multiform matrices [22]. Where there are two conditioning variables, these are often arranged into the rows and columns of a matrix. Where more conditioning variables are in use, dimensional stacking [19] can be used in which the set of conditioning variables is treated as a *hierarchy*, each nested within its parent. This is an important characteristic of mosaic plots [14, 10, 31] and treemaps [27]. Our layouts can be considered as hierarchical trellis displays that use dimensional stacking, in which individual parts can be reconfigured independently.

3 SPACE-FILLING RECTANGULAR DISPLAYS

We focus on space-filling rectangular layouts because of rectangles’ ability to tessellate and nest to form space-efficient data dense displays but we also make reference to non-rectangular layouts (e.g. Fig. 5).

Mosaic plots and treemaps are space-filling rectangular displays in which each rectangle corresponds to a conditioned subset of the data, characterised by one or more of its dimensions, colour [10] and position [3]. Both use dimensional stacking such that the hierarchy of data subsets is represented through the nesting of rectangles – in the case of mosaic plots, conditioning variables apply alternately to the *x* and *y* axes [14]. Treemaps are most commonly used as compact representations of tree structures but can also be used for detecting broader patterns in multivariate data (e.g. [17, 36, 30]). We use treemaps as both 1D and 2D layouts are supported and consider them to be a generalised form of mosaic plot.

-
- Aidan Slingsby (sbbb717@soi.city.ac.uk), Jason Dykes (jad7@soi.city.ac.uk) and Jo Wood (jwo@soi.city.ac.uk) are at the *giCentre* (<http://gicentre.org/>) in the Department of Information Science at City University London.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

Table 1. Hierarchical Visualisation Expression (HiVE) states & operators

	STATE <i>Parameters are ordered by level with unspecified levels inheriting from the previous.</i>	OPERATOR <i>Operates on level n only (except interaction expressions).</i>	
STRUCTURE	sHier (/ , var1, var2, ...)	oInsert (/ , n, var)	Hierarchy of conditioning variables
		oCut (/ , n)	Remove level n
		oSwap (/ , n1, n2)	Swap levels n1 and n2
APPEARANCE	sLayout (/ , lay1, lay2, ...)	oLayout (/ , n, lay)	Layout presets (here we use SQ, VT, HZ, OS, SP, SA, PG)
	sOrder (/ , var1, var2, ...)	oOrder (/ , n, var)	Order/position
	sShape (/ , var1, var2, ...)	oShape (/ , n, var)	Shape
	sSize (/ , var1, var2, ...)	oSize (/ , n, var)	Size (variable or FIX)
	sColor (/ , var1, var2, ...)	oColor (/ , n, var)	Colour (variable or \emptyset)
	sColorMap (/ , r11, r12, ...)	oColorMap (/ , n, r1)	Colour rules (r1)
INTERACTION	sHighlight (elem)	oHighlight (elem)	Highlight elements (identify elem as a path; see section 5.4)
	sZoom (elem)	oZoom (elem)	Zoom to one or more elements

'/' (root) indicates that the expression applies to all branches. Section 5.3 shows how the expression can apply to an individual branch. \emptyset is NULL.

Cartograms are maps whose cartographic coordinate space is distorted to accommodate some non-geometrical property of geographical data (e.g. population). The Gastner cartogram [12] in Fig. 5D sizes geographical areas by the number of properties sold whilst attempting to preserve shape. Rectangular cartograms [24, 34] use rectangles in order to make relative size comparison easier.

4 DATA

Individual property transactions in London between 2000 and 2008 are the focus of our analysis. The 1.25 million records contain property type (flat, terraced, semi-detached or detached), price, location and date of sale. We aggregate these spatio-temporal data into spatial units of varying resolution and geometry [1]. The spatial units are boroughs (administrative units; \$br), wards (smaller administrative units that nest inside boroughs; \$wd) and 4km² grid cells (\$gd). Temporal variables are derived by aggregating into years (\$yr), months of any year (e.g. July; \$mn) and months of a particular year (e.g. May 2002; \$my). We summarise these subsets using number of sales (\$sal), average price (\$prc) and coefficient of variation of price (\$vpr). We also calculate signed chi-statistics for the number of sales (\$xsl) and average price (\$xpr) – elaborated upon later. \$abr represents the area of a borough and is used in Fig. 5. The data were obtained from <http://www.houseprices.co.uk/> (used with permission) and prices have been standardised to remove the effects of inflation. The rectangular space-filling layouts are produced by *treeMappa* (<http://www.treemappa.com/>).

5 PROPERTIES OF HIERARCHICAL LAYOUTS

We have developed HiVE (Table 1), a notation for describing hierarchical visualisation *states* ('s' prefix) and *operators* ('o' prefix) for reconfiguring these states. As such, HiVE goes beyond encoding the layouts themselves; it provides operators that can be used as part of the data exploration process. States are described for part A of each figure and operators are used for subsequent parts. However, each figure part can be described by a state as shown in the accompanying video (http://gicentre.org/hierarchical_layouts/). HiVE is not intended to be comprehensive at this stage – rather to make the design choices in our hierarchical views explicit, to enable us to compare designs, to consider the scope of the design space and to move through it as we explore design alternatives.

5.1 Structure

The order in which variables are dimensionally stacked (attribute hierarchy) has a strong effect on the perception of patterns and trends [10]

because it controls the way in which the dataset is conditioned. For example, in Fig. 1A, average price by year is available per property type, but not all types by year. Some sets of variables have an inherent hierarchy of granularity; e.g. \$yr, \$mn (temporal) or \$br, \$wd (spatial; Fig. 5B), but most do not. The attribute hierarchy also affects the size and visual prominence of elements; e.g. in Fig. 1A, property types are prominent because they are at the root of the hierarchy, whereas months are harder to resolve. Rectangle colour can usually only be used effectively at the hierarchy leaves in space-filling representations – the ability to interactively switch and change the depth of hierarchies are ways to address these issues [17, 30].

The attribute hierarchy is specified using sHier (Table 1) and the oInsert, oCut and oSwap operators modify the hierarchy at the specified hierarchical level (i.e. conditioning attribute).

5.2 Appearance

Appearance can be described in terms of Bertin's visual variables [4] and reconfiguring these for appropriate parts of the display is key to our work. Informed choices for doing this should strive to produce 'cognitively plausible' [29] layouts which exploit similar image schemata to those used in human reasoning [21, ch. 4]; e.g. *containment* for categories, *part-whole* and *up-down* schemas for hierarchies and *linear order* schemas for ordered data [18, p283]. The use of cartographic principles are advocated to produce cognitively plausible layouts through the use of spatial metaphors [28, 29] and Tobler's First Law of Geography [32] that the relatedness of objects is proportional to their spatial proximity. Some properties of hierarchical relationships – such as the arbitrary nature of partition adjacency at different levels of the hierarchy (e.g. 2006 flats and 2002 semi-detached in Fig. 1) – violate the distance-similarity metaphor [29]. Using gaps or borders to separate hierarchical levels [14], using more appropriate layouts [36] and using interaction [30] help address these problems. There is a danger that introducing different layouts at different levels of the hierarchy may increase the cognitive load of the user. However, we argue that the cognitive plausibility comes from the well-understood concept of containment [18, p283] and the use of layouts that reflect the nature of the variation in the data. For example, in Fig. 4, a spatial layout is used for the boroughs within which are 'calendar views' (section 5.2; months ordered top to bottom within years ordered left to right). This uses spatial and temporal layout at different levels of the hierarchy, but the logical ordering allows us to detect temporal patterns with a focus on changes at an annual resolution (see Fig 3B for an alternative layout that does not focus on annual trends).

5.2.1 Layout

We use the following seven layouts (the first five are space-filling):

- VT & HZ: The 'slice-and-dice' algorithm [27] partitions space into horizontal (HZ) or vertical (VT) rectangles retaining 1D order (e.g. Fig. 1B), but may result in narrow rectangles that are difficult to resolve visually.
- SQ: The 'squarified' algorithm [6] (e.g. Fig. 1A) has rectangles with aspect ratios close to 1 (easier to visually resolve). They are ordered from top left to bottom right in descending order of size by alternately filling vertical and horizontal strips from the top left. Particularly appropriate when size-ranking is of interest.
- OS: The 'ordered squarified' algorithm [36] (e.g. Fig. 3B) produces rectangles whose aspect ratios are close to 1 and retains 1D ordering using a distance-based measure from top left (more consistent ordering than in SQ).
- SP: The 'spatially-ordered' algorithm [36] attempts to retain a 2D ordering (with good aspect ratios), designed to preserve the geographical configuration of elements, producing space-filling hierarchical rectangular cartograms (e.g. Fig. 5B).
- SA: Uses SP, but shifts the rectangles to their absolute geographical locations (Fig. 5C) resulting in overlap.
- PG: Non-space-filling polygon-based layout that attempts to retain the given shape for each element (e.g. Fig. 5E).

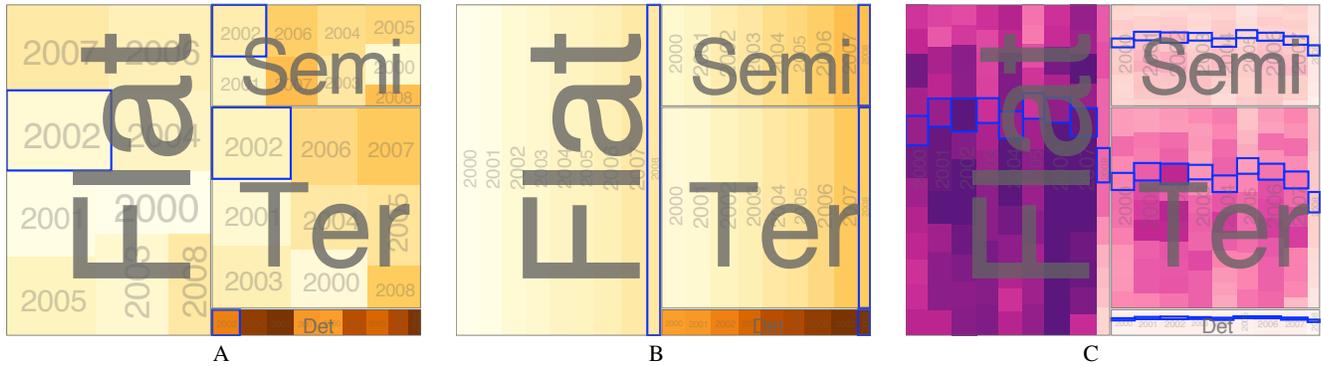


Fig. 1. A: Hierarchical space-filling rectangular layout coloured by average price, conditioned by property type then year (2002 highlighted) and sized by number of sales: `sHier(/,$ty,$yr)`; `sLayout(/,$Q)`; `sSize(/,$sal)`; `sColor(/,$prc)`; `sHighlight(/*/2002/)`. B: Using temporal ordering (2008 highlighted): `oLayout(/,2,VT)`; `oHighlight(/*/2008/)`. C: Using calendar views (May highlighted): `oInsert(/,3,$mn)`; `oLayout(/,3,HZ)`; `oColor(/,2,$)`; `oColor(/,3,$sal)`; `oHighlight(/*/*/May/)`. ‘`*/*/May/`’ refers to all values of `$ty`, all values of `$yr` and the ‘May’ value of `$mn`.

These can be considered as *layout presets* that encompass Bertin’s visual variables [4] of ‘position’ (`sOrder`) and ‘shape’ (`sShape`), with ‘size’ and ‘colour’ described below. Other visual variables could be supported with additional states and operators in HiVE (e.g. `sOrientation` and `sTexture`). In all our examples, the ordering of rectangles is derived from the conditioning variable (except for `SQ` where it is based on size), but `sOrder` can be used to specify an alternative order.

‘Calendar views’ refer to the layout `sHier(/,$yr,$mn)`; `sLayout(/,VT,HZ)`, where years are in vertical strips and months are contained within these as horizontal strips, as in Fig. 1C.

5.2.2 Size

In the majority of our examples, we base the size of elements on the number of sales (`$sal`). This value accumulates through the hierarchy (e.g. sales in 2002 is the sum of its monthly sales in Fig. 1C) and as such, size is comparable *between* hierarchical levels as well as within. Where this is not the case, interpretation may be difficult. We fix the size of elements in some of the figures (using `FIX`) where we wish to produce consistent small-multiple-like arrangements (e.g. Fig. 6) or to show each element with equal prominence (e.g. Fig. 2C).

5.2.3 Colour

`sColor` specifies the variable which is mapped to a colour through the `sColorMap`. We use the sequential ColorBrewer [5] colour schemes of ‘`RdPu`’ for `$sal`, ‘`YlOrBr`’ for `$prc`, ‘`OrRd`’ for `$vpr` and the diverging colour schemes of ‘`RdBu`’ for `$xsl` and ‘`BrGB`’ for `$xpr` consistently throughout the paper, with logarithmic scaling for `$sal` and linear scaling otherwise, to appropriate minimum and maximum values for the view. These aspects are controlled by `sColorMap`, omitted from the figures for brevity.

5.3 Applying expressions to individual branches

HiVE can be applied to individual branches of the hierarchical tree by replacing the ‘`/`’ with a path (file path analogy) to a particular element. For example, the whole of Fig. 1 can be considered a multipanel display with `$panel` as a conditioning attribute at the base of the hierarchy with the values ‘A’, ‘B’ and ‘C’. `sHier(/,$panel)`; `sLayout(/,VT)` describes the three panels. Branch ‘A’ can be specified thus – `sHier(/A/,$br,$yr)`; `sLayout(/A/,$Q)`; `sSize(/A/,$sal)`; `sColor(/A/,$prc)`; `sHighlight(/A/*/2002/)` – with other branches specified similarly.

5.4 Relating layouts using interaction and animation

We advocate the exploration of data through reconfiguring hierarchical displays as suggested by various authors (e.g. [20, 26, 38, 8]). Our operators can be used to define Yi *et al*’s interactions [38] such as

‘explore’ (show different subsets of data; `oCut` and `oInsert`), ‘reconfigure’ (reordering elements; `oOrder`), ‘encode’ (change to a different visual encoding; e.g. `oLayout` to change between rectangular and non-rectangular displays as in Fig. 7C).

Cook *et al*’s [8] ‘projection pursuit guide tour’ uses animated transitions to move through different projections of multivariate space for data exploration. Heer *et al* [15] found that some types of animation are effective means for showing how layouts relate and Robertson *et al* [25] found that although animation is a poor means of trend discovery, it is an effective way to demonstrate change. For operators that change geometrical properties of the layout, we use simple transitions that morph between these states as suggested by Florisson *et al* [9] accompanied by gradual colour blending for changes in colour. In all other cases we use fading as illustrated in the accompanying video.

Highlighting similar elements across a hierarchical level [30] enables comparison. For example, the 2002 subsets for each year are highlighted in Fig. 1A, enabling 2002 sales to be compared for each property type. The `sHighlight(/*/2002/)` expression in Fig. 1 specifies elements as described in section 5.3 using ‘`*`’ to refer to all values at that hierarchical level.

6 EXPLORING LAYOUTS

We explore the effects of applying the various hierarchical display configurations supported by HiVE to a dataset of property transactions for addressing research questions. We then propose guidelines (section 7) from the issues identified. The exploration process is reflected in the sequence of figures in this paper and the accompanying video.

6.1 Layouts with size-based orders

The ‘squarified’ layout (`SQ`) orders rectangles in decreasing order of size from the top left. The double encoding inherent in size-based ordering is appropriate for research questions based on ranking, such as identifying the boroughs with highest sales (Fig. 2 shows this to be Wandsworth) and years with the least sales (Fig. 1A shows that this is 2008). In the latter example, the lack of consistency of year positioning makes it hard to compare specific years between property types. Highlighting these rectangles helps address this difficulty [30].

6.2 Layouts for ordinal data

Most research questions benefit from using 1D orders that are independent of rectangle size. For these, slice-and-dice (`VT` and `HZ`) and ordered-squarified `OS` layouts are suitable. The choice of layout partly depends on the number of ordinal values and the aspect ratio of the space available. Ordered squarified is particularly suitable where there is a large number of values (e.g. the 108 months in each borough shown in Fig. 3B). Slice-and-dice may be more suitable where there are fewer categories. Alternating `VT` and `HZ` through

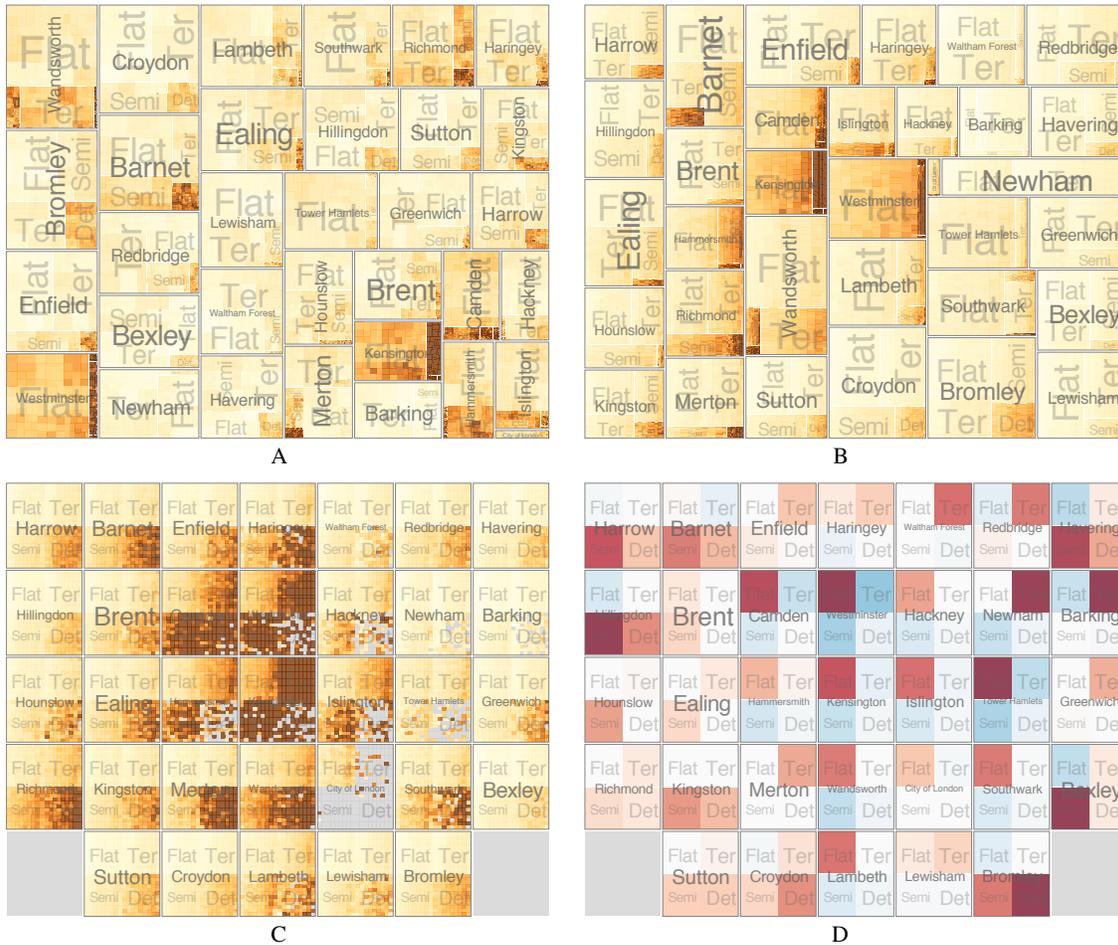


Fig. 2. A: Sized-based ordering, coloured by average price: `sHier(/, $br, $ty, $yr, $mn)`; `sLayout(/, SQ)`; `sSize(/, $sal)`; `sColor(/, 0, 0, 0, $prc)`. B: Reconfigure to a spatial and temporal layout: `oLayout(/, 1, SP)`; `oLayout(/, 2, OS)`; `oLayout(/, 3, VT)`; `oLayout(/, 4, HZ)`. C: Fix the size: `oSize(/, 1, FIX)`; `oSize(/, 2, FIX)`; `oSize(/, 3, FIX)`; `oSize(/, 4, FIX)`. D: Remove time, and colour by deviation from expected sales: `oCut(/, 4)`; `oCut(/, 3)`; `oColor(/, 2, $xsl)`.

the hierarchy can produce layouts similar to mosaic plots (and matrix diagrams if sizes are fixed). They are particularly suitable where variables have hierarchical dependencies, such as our calendar views (`sHier($yr, $mn)`).

6.3 Layouts for time-based data and questions

Temporal data can be considered as ordinal. In Fig. 1A, years are not arranged temporally; as such, temporal trends are difficult to detect. Rearranging the years into a time-based order using an ordered space-filling layout [36] (Fig. 1B) makes the increase in annual house price easier to detect. In Fig. 1C, we have added month to the hierarchy producing calendar views coloured by the number of sales. Seasonal variations in the numbers of sales are apparent for flats and terraced housing, however colour rescaling (using `oColorMap`) or using colour schemes that are local to individual parts of the hierarchy are required to detect these patterns where property types have low sales. Alternatively, colour can be used to show values as a proportion or deviation from a baseline. Appropriate baselines include those that reflect the values expected from hypotheses that we might then accept or reject on the basis of the display. For example, in Fig. 4A (calendar views), our null hypothesis is that the number of sales does not vary monthly (expected or baseline values are a twelfth of the sales for each year). The geographically-consistent seasonal trends that are apparent might cause us to reject our null hypothesis. Identifying the elements with statistically-significant levels of variation might help us make that

choice. Fig. 4B shows the deviation of price from the yearly average (accounting for inflation). Whilst prices rise steadily every year, this is not the case for 2008 where prices have dropped markedly in the final quarter, a trend not observed in Westminster.

Nesting the two temporal resolutions of year and month to produce calendar views is appropriate where we are expecting yearly and monthly patterns. However, this may obscure other temporal patterns. In Fig. 3B, we use an ordered squarified layout of all 108 months in the period ordered from the left top to bottom right (compare with the calendar views in Fig. 3A). Although both graphics show exactly the same data, the use of `$my` and the associated `OS` layout in Fig. 3B make the upward trend in prices and subsequent slump more apparent as it is a continuous trend over the entire period. The result is a more appropriate layout for research questions that relate to ongoing rather than periodic change. The additional hierarchical level used in Fig. 3A and alternative layouts are more appropriate for comparing annual patterns which are overshadowed by the longer term trend in the case of this attribute. Again, interactive colour rescaling or colouring on the basis of relative values is required to detect relative rises and falls in different boroughs.

6.4 Geographical layouts

Spatially-ordered layouts (SP) have rectangles that are arranged according to their geographical locations. The effect of this layout can be seen by comparing the non-spatial layout in Fig. 2A with the spatial

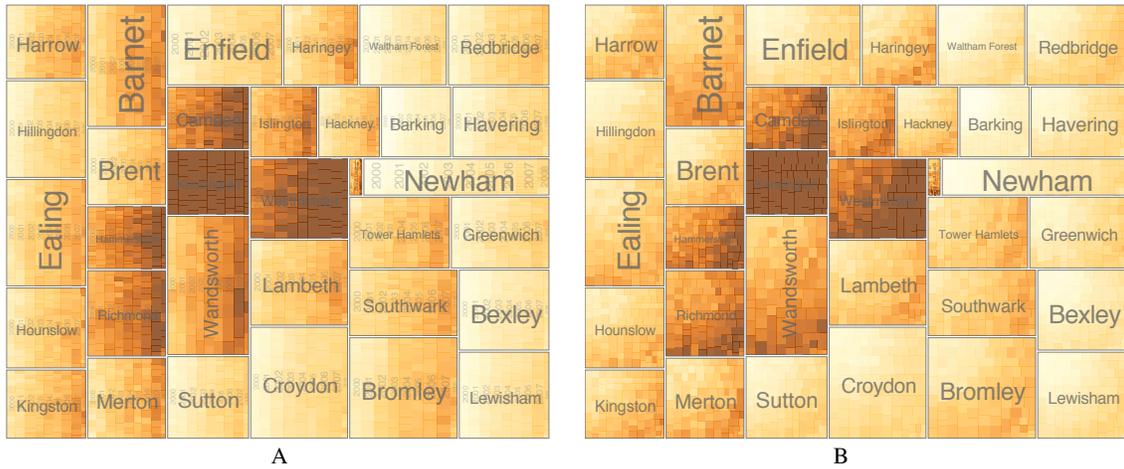


Fig. 3. Sales by borough and month, sized by the number of sales and coloured by the average price. A: Uses calendar views of time: `sHier (/,$br,$yr,$mn)`; `sLayout (/,$SP,$VT,$HZ)`; `sSize (/,$sal)`; `sColor (/,$prc)`. B: Uses all 108 months in the period ordered from the top left: `oCut (/,$3)`; `oCut (/,$2)`; `oInsert (/,$2,$my)`; `oLayout (/,$2,$OS)`; `oColor (/,$2,$prc)`.

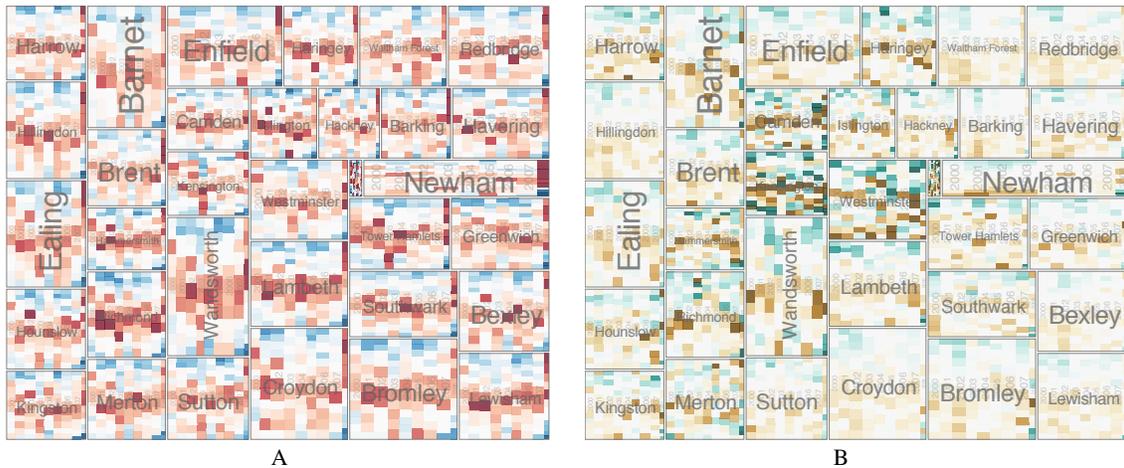


Fig. 4. Boroughs containing calendar views, coloured by deviation from 'expected'. A: Red indicates higher sales than the yearly average; blue indicates fewer sales: `sHier (/,$br,$yr,$mn)`; `sLayout (/,$SP,$VT,$HZ)`; `sSize (/,$sal)`; `sColor (/,$prc)`. B: Brown indicates higher prices than average for the year; turquoise indicates lower prices: `oColor (/,$3,$xpr)`.

layout in Fig. 2B, in which flats overwhelmingly dominate sales near Central London whereas sales of other types are proportionally higher in peripheral areas, sometimes exceeding those of flats. Fig. 3 also uses a spatial layout, facilitating the detection of spatial patterns in average price trends – south and east London have the lowest prices and central and southwestern areas have the highest prices.

Spatially-ordered layouts can also apply to multiple levels of a hierarchy. In Fig. 5B, a hierarchy of two spatial units of increasing granularity are nested and spatially arranged. High spatial variation is apparent within boroughs. For example in Lambeth, wards with the highest average price are closer to Central London, the converse is true in the case of Camden. The space-filling nature of these cartograms often results in positional inaccuracies which can be conveyed using displacement vectors [36]. Where absolute locations are required for research questions, these can be encoded using a perceptually-constant 2D colour-space [36] or by using a different layout.

We use animated transitions to relate the layouts in Figs. 5C, 5D and 5E (this method for relating layouts has been found to be effective [9]) – see video. The layouts that use absolute space show more of the spatial subtleties of the patterns, e.g. the high average house prices linearly arranged from the centre to the southwest. However, occlud-

ing layouts such as Fig. 5C are difficult to interpret on their own but may be useful when animated transitions are provided to other layouts and layouts whose geometrical elements do not fill space completely, produce less data-dense graphics when dimensionally stacked.

`oSwap` is a useful operator for OD-maps [37] which are raster-based origin-destination maps – `sHier (/,$oc,$dc)`; `sLayout (/,$SP)`; `sSize (/,$FIX)`; `sColor (/,$fl)` – in which `$oc` is the originating grid cell, `$dc` is the destination grid cell and `$fl` is the volume of flow between the given origin and destination cells. `oSwap` enables directionality in the origins and destinations to be explored. This example also illustrates that datasets may have multiple locations, both of which may be added to the hierarchy, in this case producing raster maps of destinations embedded in raster maps of origins.

Comparing layouts where space is discretised differently is one way of studying the effect of the modifiable areal unit problem [23]. Fig. 6 shows a spatial arrangement where instead of conditioning the data by administrative unit, we use 4km² grid squares, in which we embed calendar views (`sLayout (/,$VT,$HZ)`; `sHier (/,$yr,$mn)`). Fixing the size of both the spatial units and rectangle sizes and using a spatial arrangement results in a layout that imposes a regular tessellated

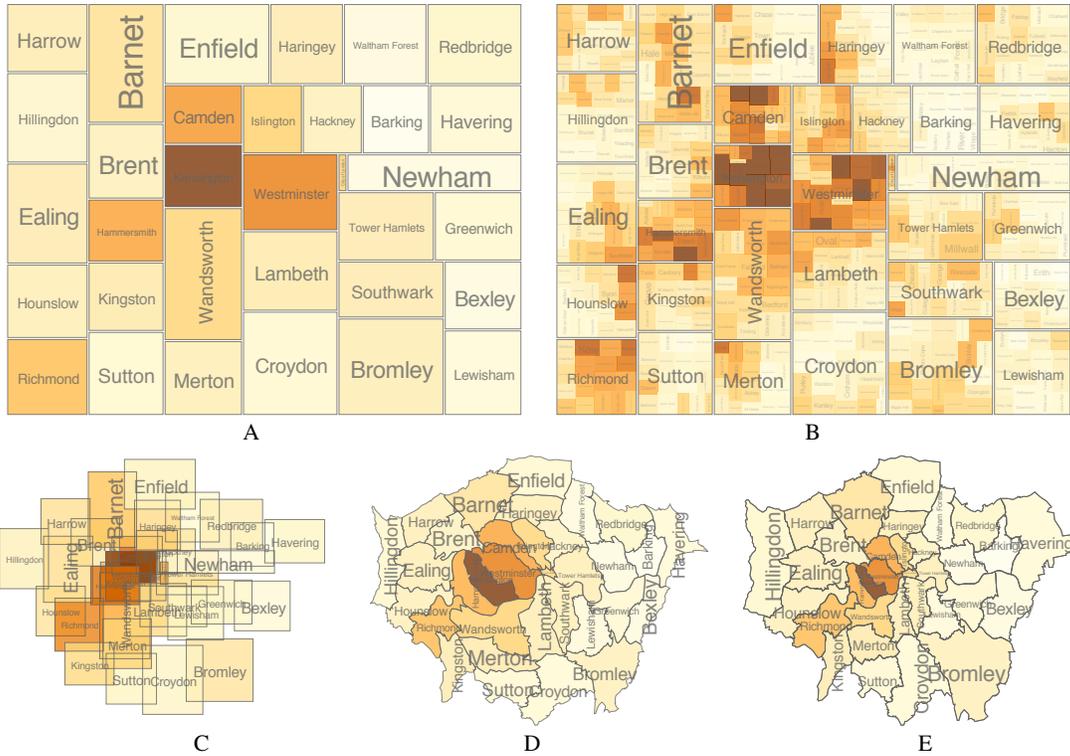


Fig. 5. Cartograms and maps. A: Rectangular cartogram: `sHier(/,$br)`; `sLayout(/,SP)`; `sSize(/,$sal)`; `sColor(/,$prc)`. B: Hierarchical rectangular cartogram: `oInsert(/,2,$wd)`; `oLayout(/,2,SP)`; `oColor(/,1,Ø)`; `oColor(/,2,$prc)`. C: As B, but using absolute positioning: `oCut(/,2)`; `oLayout(/,1,SA)`. D: Gastner cartogram (polygon layout; sized by sales): `oLayout(/,1,PG)`. E: Map (as D, but using geographical shape): `oSize(/,$abr)`. `$abr` is the borough area.

grid on absolute geographical space (at the `$gd` level) upon which geographical boundaries can be drawn.

Fixing the sizes of rectangles reduces their individual information-carrying capacity but facilitates more consistent overall layouts. It also reduces the cartogram effect, resulting in data of lower significance (low sales, therefore low sample sizes) being displayed with equal prominence. The average prices shown in row 5, col 9 of Fig. 6B correspond to low sales (see corresponding cell in Fig. 6A) but they are given more prominence in layouts where rectangles are sized by sales. As such, this (equally valid) view of the data must be interpreted slightly differently – perhaps in conjunction with a version that is coloured by the number of sales. We suggest side-by-side comparison or animated transition to help relate these views such as these.

Geography does not necessarily have to be at the base of the hierarchy. In Fig. 7, we place boroughs at the second level of the hierarchy, apply the `oSize(FIX)` operator to fix the size of rectangles, remove the final two hierarchical levels and reconfigure level 2 to map-based layouts (Fig. 7C). This small multiple map layout allows the recognisable shapes of boroughs to be preserved, but at the expense of space-efficiency and space-efficient dimensional stacking.

6.5 Layouts for nominal data

We recommend that a *consistent ordering* be used for nominal values. In Figs. 2B, 2C and 2D, we consistently order flats, terrace, semi-detached and detached types. The ordering used should be selected to reflect some ordinal sequence to encourage comparison (unlike in Fig. 2A – see Redbridge). We have ordered these by likely floor-space.

The numbers of sales vary markedly between the property types, resulting in some rectangles sizes (e.g. detached houses in the centre) being too small to be easily resolvable. In Fig. 2C, we fix the size of each rectangle (grey shows no data; there are few detached house sales in the City of London). Fixing the rectangle size may draw more attention to these than warranted and so these displays should be used

in conjunction with a version that is coloured by sales, either using a fade transition or placing side-by-side (as is the case in Fig. 6).

To investigate how relative sales of different property types vary spatially, we can form a null hypothesis that the ratio of sales between the property types are spatially invariant. To test this hypothesis, we use the average sales proportions of flats (49%), terraced (31%), semi-detached (16%) and detached (4%) for the whole area to establish a baseline and then show the deviation from this. Fig. 2D (this uses a linear and symmetrical diverging colour scheme) shows that we can probably reject our null hypothesis. Sales of flats are higher than the London average in the centre (the consistent ordering ensures flats are always in the top left), more semi-detached housing than average exists towards the periphery and no borough has the average proportion. By modifying the hierarchy (with the `oCut`, `oInsert` and `oSwap` operators), reconfiguring the layouts (`oLayout` and `oSize`), changing the colour (`oColor` and `oColorMap`) and establishing alternative baselines, alternative hypotheses can be investigated to address different research questions.

In Fig. 7 we study the consistency of price by type, space and time, by colouring layouts by the coefficient of variation of price. The instability of colour, suggests that many of the sample sizes are too small to give reliable estimations of price variation, but nevertheless colour is relatively consistent by borough and different spatial patterns can be detected for each property type. In Fig. 7C, we fix the size of the rectangles, remove the temporal attributes from the hierarchy and switch the layout to polygons. This results in small multiple choropleth maps conditioned by type (`sHier(/,$ty,$br)`; `sLayout(/,OS,PG)`; `sSize(/,FIX,$abr)`).

7 GUIDELINES FOR USING HIERARCHICAL LAYOUTS

We propose a number of guidelines based on our observations and experiences for using and configuring hierarchical layouts to address research questions.

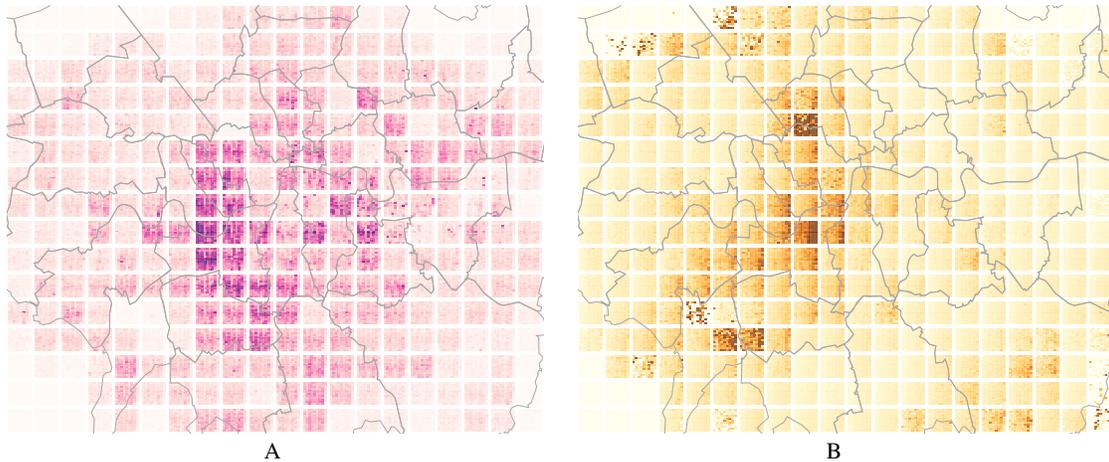


Fig. 6. The data are spatially reaggreated into 4km^2 grid squares. Absolute geographical positioning is employed because node size is fixed and the correct aspect ratio is used (borough boundaries shown for reference). A: Coloured by number of sales: `sHier(/, $gd, $yr, $mn)`; `sLayout(/, SP, VT, HZ)`; `sSize(/, FIX)`; `sColor(/, 0, 0, $sal)`. B: Coloured by average price: `oColor(/, 3, $prc)`.

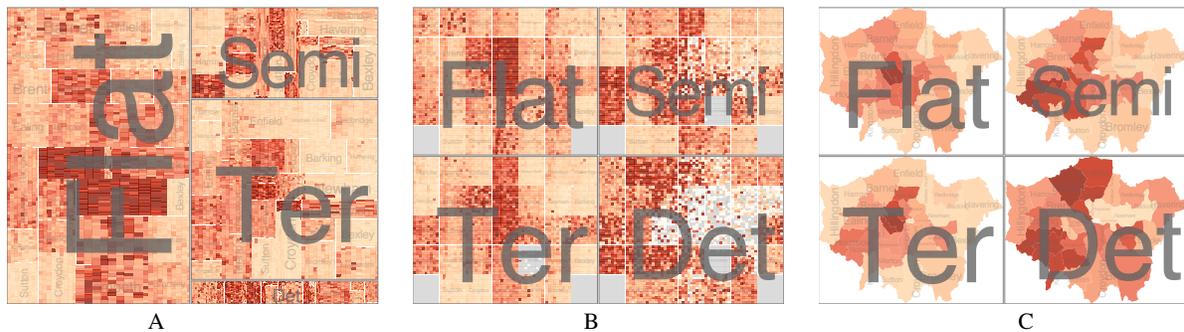


Fig. 7. Space is at level 2 of the hierarchy. Coloured by coefficient of variation of price (grey is no sales). A: `sHier(/, $ty, $br, $yr, $mn)`; `sLayout(/, OS, SP, VR, HZ)`; `sSize(/, $sal)`; `sColor(/, 0, 0, 0, $vpr)`. B: Fix rectangle size: `oSize(/, 4, FIX)`; `oSize(/, 3, FIX)`; `oSize(/, 2, FIX)`; `oSize(/, 1, FIX)`. C: Choropleth maps: `oCut(/, 4)`; `oCut(/, 3)`; `oLayout(/, 2, PG)`; `oSize(/, 2, $abr)`.

1. *Reconfigure conditioning hierarchies to explore the data space.* Use `oCut`, `oInsert` and `oSwap` to reconfigure the hierarchy to explore variation in terms of different conditioning variables. For example, placing `$br` above `$ty` in Fig. 7 allows geographical variation by property type to be explored.
2. *Use appropriate layouts to reveal structure in data.* Experiment with alternative layouts to explore the design space. `HZ`, `VT` with fixed rectangle size (see 4) can produce mosaic plots, useful where combinations of categorical variables are important. `OS` is appropriate where there is a large number of values and `VT/HZ` where there are fewer values and where the dimensions of the available space allow good aspect ratios.
3. *Preserve salient 1D or 2D ordering.* Choose appropriate ordering for ordinal, temporal and spatial variables for each hierarchical level in response to research questions and order nominal variable values consistently.
4. *Fix rectangle size at appropriate hierarchical levels to produce consistent layouts with small-multiple-like properties.* The resulting juxtaposed graphical elements with shared layout characteristics can facilitate the side-by-side comparison of graphics, minimising the work required of the eye and brain.
5. *Scale colour to data-ranges to different parts of the hierarchy to explore local and global patterns.* Scaling to data-ranges in localised parts of the hierarchy (e.g. by year in Fig. 4) addresses research questions based on localised variation, whereas scaling to the entire data-ranges draws attention to more global patterns.
6. *Condition datasets by attributes of different granularities at ad-*

acent levels of the hierarchy. In the case of time, this allows us to consider the effects of cyclical temporal patterns (e.g. `$yr`, `$mn`). In the case of space this draws attention to the effects of spatial resolution and scale.

7. *Condition by different aggregations of time and space.* This helps explore the effects of modifiable units on patterns in the data.
8. *Reaggregate spatial data to equally-sized grid cells and fix rectangle size.* This can produce consistent small-multiple-like arrangements (see 4) that retain the properties of the original geographical coordinate space (e.g. Fig. 6) and can be used to address research questions that relate to geographic variation in absolute geographical space.
9. *Use dynamic techniques to relate these various states.* For example, use *highlighting* to show items across hierarchy and *brushing* for details-on-demand. Smooth transitions between layouts can to help reduce cognitive load when relating these.

8 FURTHER AND ONGOING WORK

Although our examples and notation have focussed on space-filling rectangular layouts, the concepts are applicable to other types of layout as illustrated by our introductory example and our use of some non-rectangular layouts. HiVE was developed so that we could be systematic in describing configurations and reconfigurations in layouts and so we could describe and build interfaces for collaborative visualisation. We are extending this so that it can encode a broader set of hierarchical layouts that use dimensional stacking by adding states and operators to represent a wider range of visual variables. For example,

a stacked bar chart and pie-chart can be considered to be equivalent, except that pie charts use polar rather than Cartesian coordinates [35].

There is also scope for HiVE to be used to document the visual data analysis process and maintain a history of interactions. This could either be used to support users during data exploration (e.g. documenting insights, reverting to saved states) or used subsequently to help increase the understanding of the data visualisation process [16] and to undertake user studies.

9 CONCLUSION

Many graphical techniques in common use for representing multivariate data are hierarchical. Explicitly acknowledging this hierarchy draws attention to reconfigurable properties, including attribute hierarchy, layout and colour. Each strongly affects the salient properties of the graphic, the patterns and trends revealed and the research questions that can be addressed.

Our Hierarchical Visualisation Expression (HiVE) notation describes the hierarchical data and design space, allowing these to be explored comprehensively and systematically. Independently reconfiguring layouts for different parts of the hierarchy as an aspect of the data exploration process is key to our approach. These configurations should correspond to the data types being represented and the questions being asked of the data. HiVE embeds this approach, not only describing the configuration of graphics but also the operators required to explore data using these layouts.

We illustrate this by visually exploring a spatio-temporal dataset of 1.25 million property transactions, in which we have found temporal and spatial patterns in property sales. Using HiVE enables us to recognise large-scale patterns (e.g. the 2008 slump), assess their spatial variability (e.g. the slump in price was not observed in Westminster) and identify new lines of enquiry (e.g. investigate whether the high Westminster property prices in 2008 apply to all housing types and price-bands at a range of spatial and temporal scales). We propose a number of guidelines based on this example for choosing layouts that address research questions as part of the interactive data exploration process.

ACKNOWLEDGMENTS

The house price data are Crown copyright (used with kind permission of HM Land Registry) and the spatial boundary data are © Crown Copyright/database right 2009 (an Ordnance Survey/EDINA supplied service). The authors wish to thank participants of the GeoViz Hamburg workshop and the InfoVis reviewers for the useful feedback.

REFERENCES

- [1] G. Andrienko, N. Andrienko, J. Dykes, S. I. Fabrikant, and M. Wachowicz. Geovisualization of dynamics, movement and change: key issues and developing approaches in visualization research. *Information Visualization*, 7:173–180, 2008.
- [2] R. A. Becker, W. S. Cleveland, and M. Shyu. The visual design and control of trellis display. *Journal of Computational and Statistical Graphics*, 5:123–155, 1996.
- [3] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.
- [4] J. Bertin. *Sémiologie Graphique*. Editions Gauthier-Villars, Paris, 1967.
- [5] C. Brewer, G. Hatchard, and M. Harrower. ColorBrewer in print: A catalog of color schemes for maps. *Cartography and Geographic Information Science*, 30(1):5–32, 2003.
- [6] M. Bruls, K. Huizing, and J. J. van Wijk. Squarified treemaps. In *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, Springer Computer Science, pages 33–42, 1999.
- [7] J. Chambers, W. S. Cleveland, B. Kleiner, and P. Tukey. *Graphical methods for data analysis*. Duxbury Press, 1983.
- [8] D. Cook, A. Buja, J. Cabrera, and C. Hurlley. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4:155–172, 1995.
- [9] S. Florisson, M. van Kreveld, and B. Speckmann. Rectangular cartograms: construction & animation. In *Proceedings of the 21st annual symposium on Computational geometry*, pages 372–373, Pisa, Italy, 2005. ACM.
- [10] M. Friendly. Mosaic displays for Multi-Way contingency tables. *Journal of the American Statistical Association*, 89(425):190–200, Mar. 1994.
- [11] M. Friendly and E. Kwan. Effect ordering for data displays. *Computational Statistics and Data Analysis*, 43(4):509–539, 2003.
- [12] M. T. Gastner and M. E. J. Newman. Diffusion-based method for producing density-equalizing maps. *Proceedings of the National Academy of Sciences of USA*, 101(20):7499–7504, May 2004.
- [13] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venktrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing Group-By, Cross-Tab, and Sub-Totals. *J. Data Mining and Knowledge Discovery*, 1(1):2953, 1997.
- [14] J. Hartigan and B. Kleiner. Mosaics for contingency tables. In W. Eddy, editor, *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface*, pages 286–273, 1981.
- [15] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1240–1247, 2007.
- [16] T. Jankun-Kelly, K. Ma, and M. Gertz. A model for the visualization exploration process. In *IEEE VIS 2002*, pages 323–330, 2002.
- [17] E. Kolatch and B. Weinstein. CatTrees: dynamic visualization of categorical data using treemaps. Project report, 2001.
- [18] G. Lakoff. *Women, fire and dangerous things*. University of Chicago Press, Chicago, 1987.
- [19] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proceedings of the 1st conference on Visualization '90*, pages 230–237, San Francisco, California, 1990. IEEE Computer Society Press.
- [20] A. MacEachren. Time as a cartographic variable. In H. M. Hearnshaw and D. J. Unwin, editors, *Visualization in Geographical Information Systems*, pages 115–130. John Wiley & Sons, Mar. 1994.
- [21] A. MacEachren. *How maps work*. The Guilford Press, New York, 1995.
- [22] A. MacEachren, D. Xiping, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 31–38, 2003.
- [23] S. Openshaw. *The Modifiable Areal Unit Problem*. Geo Books, Norwich, UK, 1984.
- [24] E. Raisz. The rectangular statistical cartogram. *Geographical Review*, 24(2):292–296, Apr. 1934.
- [25] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1325–1332, 2008.
- [26] I. Shepherd. Putting time on the map: Dynamic displays in data visualization and GIS. In P. Fisher, editor, *Innovations in GIS 2*, pages 169–187. Taylor & Francis, 1 edition, May 1995.
- [27] B. Shneiderman. Tree visualization with tree-maps: 2D space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.
- [28] A. Skupin and B. P. Buttenfield. Spatial metaphors for visualizing information spaces. *Proceedings of ACSM/ASPRS Annual Convention and Exhibition*, pages 116–125, 1997.
- [29] A. Skupin and S. I. Fabrikant. Spatialization methods: A cartographic research agenda for non-geographic information visualization. *Cartography and Geographic Information Science*, 30:99–119, Apr. 2003.
- [30] A. Slingsby, J. Dykes, and J. Wood. Using treemaps for variable selection in spatio-temporal visualisation. *Information Visualization*, 7(3-4):210–224, 2008.
- [31] M. Theus. Interactive data visualization using mondrian. *Journal of Statistical Software*, 7:1–9, 2002.
- [32] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(2):234–240, 1970.
- [33] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.
- [34] M. van Kreveld and B. Speckmann. On rectangular cartograms. *Comput. Geom. Theory Appl.*, 37(3):175–187, 2007.
- [35] L. Wilkinson. *The Grammar of Graphics*. Springer, 1 edition, Aug. 1999.
- [36] J. Wood and J. Dykes. Spatially ordered treemaps. *Visualization and Computer Graphics, IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355, 2008.
- [37] J. Wood, J. A. Dykes, A. Slingsby, and R. Radburn. Flow trees for exploring spatial trajectories. In *Proceedings of GIS/UK*, pages 31–34, 2009.
- [38] J. S. Yi, Y. ah Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.