Spatially Ordered Treemaps

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

Abstract—Existing treemap layout algorithms suffer to some extent from poor or inconsistent mappings between data order and visual ordering in their representation, reducing their cognitive plausibility. While attempts have been made to quantify this mismatch, and algorithms proposed to minimize inconsistency, solutions provided tend to concentrate on one-dimensional ordering. We propose extensions to the existing *squarified* layout algorithm that exploit the two-dimensional arrangement of treemap nodes more effectively. Our proposed *spatial squarified* layout algorithm provides a more consistent arrangement of nodes while maintaining low aspect ratios. It is suitable for the arrangement of data with a geographic component and can be used to create tessellated cartograms for geovisualization. Locational consistency is measured and visualized and a number of layout algorithms are compared. CIELab color space and displacement vector overlays are used to assess and emphasize the spatial layout of treemap nodes. A case study involving locations of tagged photographs in the Flickr database is described.

Index Terms—Geovisualization, treemaps, cartograms, CIELab, geographic information, tree structures.

1 Introduction

The use of treemaps, first proposed by Shneiderman [21], to represent hierarchical data has received wide attention in the information visualization community [22]. Their compact use of graphical space, reduced graphical complexity, relative ease of computation [24] as well as some high profile examples (e.g. [25]) have all contributed to their popularity. Yet they have also received criticism for their lack of cognitive plausibility [9], poorly perceived aesthetic qualities [6] and poor task-driven performance [1, 6].

In this paper we address some of the weaknesses of existing treemap layout algorithms and presentation conventions by focussing on node placement. Our aim is to use location (a ratio-scale property) to represent relationships within hierarchical levels to produce 'richer and less opaque' representations and address concerns relating to the cognitive plausibility of treemaps [24]. In doing so, we produce treemaps that may be used more effectively for answering queries that involve identifying relationships and trends within datasets.

2 Ordered Layouts

Nodes in a treemap represent individual data items in some dataset and their size, color and text label can be used to represent attributes of the data item. The topological relationship with higher level containing nodes is used to show the item's position in the hierarchy. However in most treemaps, the node's position does not precisely represent any charactersitic of the data. This is a potential waste of the information carrying capacity of the treemap and can also reduce the clarity of the representation by violating the distance-similarity metaphor [10] (the same data can be represented in arbitrarily different looking treemaps depending on the ordering of nodes).

The problem is illustrated in Figure 1. Here, 256 nodes of unit size are arranged using the *squarified* layout [5] that attempts to minimise aspect ratios. Nodes are approximately ordered from top-left to bottom-right. To search for a node early in the sequence (dark green), we would need to look somewhere towards the top or left. However the relationship between node order and distance from the top-left is not a simple one in this layout (Figure 1b). In this example nodes 0-15 show a consistent linear relationship between order and distance. However the next node in sequence, node 16 is as close to the corner as node 1. These large jumps in distance-node order relationship make locating a given node more difficult. Spatial discontinuities also make it difficult to infer node order directly from location without significant cognitive effort and so impede efforts to identify relationships and trends.

Bederson *et al* [2] attempted to address an aspect of this problem by considering various ordered layout algorthims where "*items that are next to each other in the input to the algorithm are adjacent in the treemap*" [2, p.836]. They recognized that the linear ordering of nodes could be used to emphasize trends in a dataset as well as aid navigation through it. They proposed a metric, *readability*, that attempted to quantify the ease with which an ordered sequence could be followed in a treemap. This was measured by counting the number of abrupt angular changes (of more than 6 degrees) required when moving in sequence though a set of ordered sibling nodes. While this measure identifies angular change, it does not take into account the distance of separation between adjacent nodes, nor the consistency with which position relates to order.

This is illustrated in Figure 2, which shows four layouts of 16 ordered unit-sized nodes and their respective readability scores. No angular change is required to proceed from node 1 to node 16 in the slice and dice layout, so it receives a maximum readability score of 1. The *strip layout* [2] only requires a change in direction when proceeding from one row to the next, so receives the second highest readability score. The *squarifed* [5] layout requires angular changes that increase in frequency towards the end of the node list. The fourth layout, here termed *ordered squarified* requires some form of angular change between almost all nodes, so receives the lowest readability score. However there is a consistency in this fourth layout not possessed by the strip or squarified layouts that shows a gradual decrease in node rank from top-left to bottom-right.

Tu and Shen [28], attempted to give greater importance to two-dimensional position by overlaying some known image (they used a map of the United States) on the treemap. This image was then distorted according to changes in treemap node size. They argued that knowledge of how and where the image has been distorted can be used to assess node change visually. However, this technique provides a rather loose and arbitrary coupling between node location and distorted image.

The differences in approach to layout strategies is in part a function of the fact that two-dimensional space is being used to represent a one-dimensional sequence of data items (such as time series, alphabetical ordering or size ordering). This is a specific case of a more general problem of representing one-dimensional sequences of data in two-dimensional graphical space [15]. Existing layout algorithms can tolerate this mis-match between data dimension and representation dimension if the queries they attempt to facilitate are of the form "*where is node x?*", but only then if there is some form of secondary ability to identify a node once it is found (e.g. a text label). However, queries of the form "*where is the nth node in the sequence?*" are more difficult due to the mixing of one-dimensional vertical and horizontal ordering

• *Jo Wood (jwo@soi.city.ac.uk) and Jason Dykes (jad7@soi.city.ac.uk) are based at the giCentre, School of Informatics, City University London.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mailto:tvcg@computer.org.

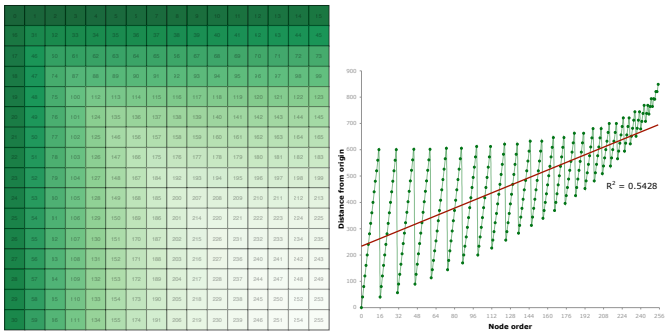

Fig. 1. *Squarified* layout of 256 ordered nodes of unit size colored by order. While there are sequences of graphical order following node order, there are also large jumps. Relationship between node order and distance from top-left (the origin) itself varies with distance from origin.

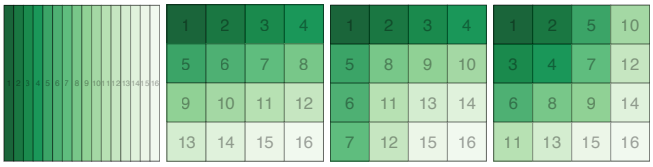

Fig. 2. *Slice and dice* [21], *strip* [2], *squarified* [5] and *ordered squarified* layouts of 16 ordered nodes of unit size. The *readability* scores of the four layouts are 1.0, 0.625, 0.375 and 0.125 respectively (1 indicates no angular change, 0 indicates every jump between sequential nodes requires an abrupt angular change).

in two-dimensional space. This in turn makes it more difficult to support queries that relate node order to some other variable attached to each node (e.g."*What is the relationship between stock value and recent stock growth*" in a treemap that orders and sizes nodes by stock value and colors them according to change over time [25]).

We therefore propose a new layout algorithm, the *ordered squarified* layout that attempts to order nodes with two-dimensional consistency by relating node order to Euclidean distance from the parent node's top-left corner (here termed its *origin*). It is based on the *squarified* layout algorithm of Bruls *et al*[5], but additionally associates a two-dimensional location with each node. For a sorted set **s** of **n** ordered nodes that must be laid out inside a containing rectangle **r**, each node is given a location according to the algorithm *AllocatePosition*:

```
Function AllocatePosition (s,r) {
  float d ← sqrt(r.area / n);
  boolean isHorizontal ← (r.width < r.height);
  List positions;

  for i←0 to < n {
    if (isHorizontal) {
      x ← r.x + mod(i*d, r.width);
      y ← r.y + floor(i*d / r.width)*d;
    } else {
      x ← r.x + floor(i*d / r.height)*d;
      y ← r.y + mod(i*d, r.height);
    }
    positions.add(x,y);
  }
  sortByDistance(positions);

  for each node in s {
    node ← positions(i++);
  }
}
```

The process of allocating locations that are exactly equally spaced within **r** yet cover it comprehensively is a non-trivial one, as it is essentially a two-dimensional circle packing problem [31]. However, since the algorithm only requires rank order of location sorted by distance from origin, and since nodes of different sizes will only be approximately placed at their nominal location, an approximate tessellation proves adequate by calculating the average distance d between nodes. The function *sortByDistance*() simply sorts the newly created positions according to their Euclidean distance from the origin $(r.x, r.y)$. An example applied to 10 nodes within a square is shown in Figure 3 .

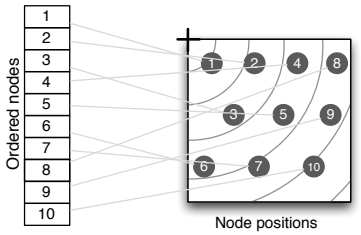

Fig. 3. Ten nodes located after spacing them within their containing rectangle. Nodes are sorted by distance from the containing rectangle's origin (the first node is closest).

The *orderedSquarified* layout proceeds recursively as in the original *squarified* layout, but instead of selecting each node in turn from an ordered list of nodes, it selects the node closest to the current position in the enclosing rectangle. Every time a new node is added, the current position is moved d units right or down depending on whether nodes are being laid out horizontally or vertically. After each call to *layoutrow*() in Bruls' *squarified* algorithm, *AllocatePosition()* is called again to reposition the remaining nodes within the remaining rectangular space.

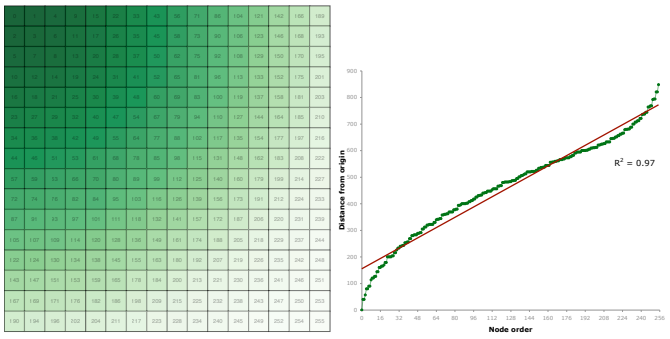

Fig. 4. *OrderedSquarified* layout of unit size nodes colored by order. Euclidean distance from the top-left corner (origin) is approximately linearly proportional to node order. Variations from linearity are due to forcing a circular distribution into an enclosing rectangle.

The layout applied to nodes of equal size inside a square parent is shown in Figure 4. Summary statistics for four layout algorithms applied to 100 equally sized nodes are shown in Table 1. The metric *distance correlation* is simply the R^2 Pearson Product-Moment correlation coefficient between node order and node distance from the origin. It gives an indication of the order-distance consistency of nodes, although it must be recognized that this relationship is likely to be a non-linear one, so the measure only gives an approximate indication of consistency. Compared with the squarified layout of the same set of nodes (Figure 1), there is greater positional consistency while low aspect ratios are retained. The slice and dice layout has greater consistency still, but as has been widely recognized, the poor aspect ratios it produces can make visual comparison difficult [2, 5, 23].

Laying out nodes of equal size, especially when the number of nodes is a perfect square, provides a best-case for both aspect ratio and distance consitency. Most real-world treemaps size nodes according to some interval or ratio-scale measurement, so to test the suitability of

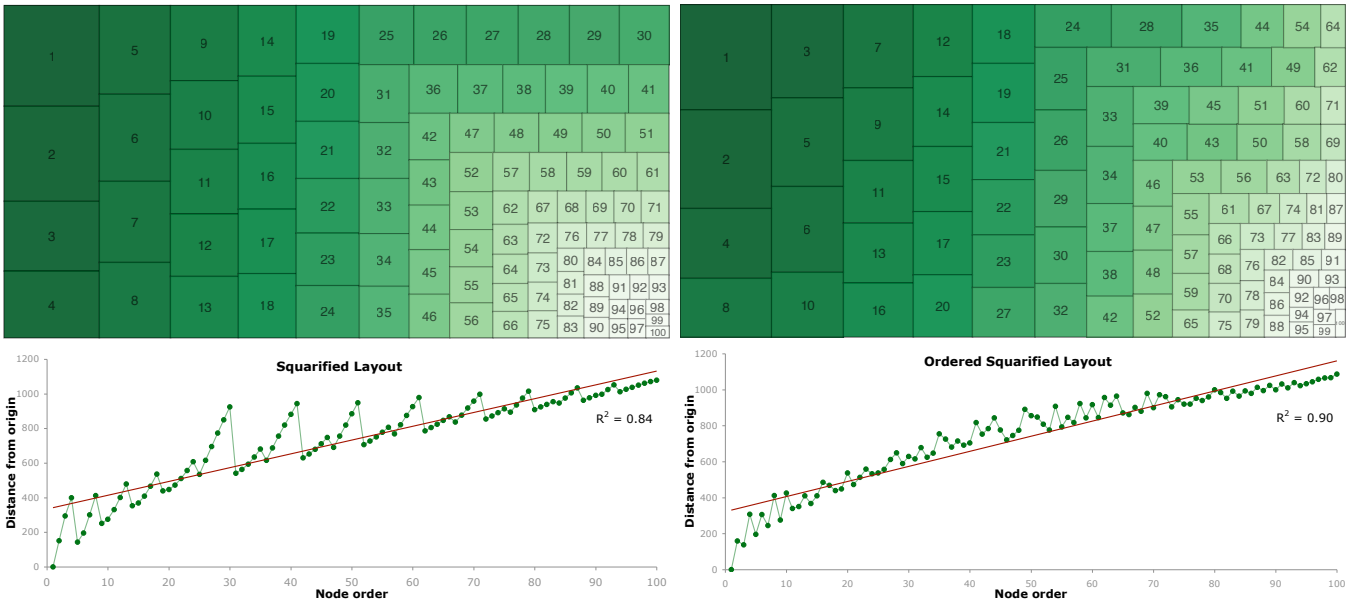

Fig. 5. *Squarified* (left) and *ordered squarified* (right) layouts of 100 ordered nodes of random sizes. Nodes are colored by order (ordered by size). Tthe *squarified* layout appears to change approximately half way along its length (node 24) as the aspect ratio in which to fit remaining nodes changes from a horizontal rectangle to being approximately square.

Table 1. Layout statistics for various layouts of 100 equally sized nodes.

Layout	Aspect ratio	Readability	Distance correlation
Slice & dice	99.02	1.00	1.00
OrderedSquarified	1.00	0.02	0.97
Squarified	1.00	0.66	0.56
Strip	1.00	0.82	0.57

Table 2. Layout statistics for various layouts of 100 randomly sized nodes. Node size follows a log-normal distribution. Statistics are means of 1000 realizations.

Layout	Aspect ratio	Readability	Distance correlation
Slice & dice	265.88	1.00	0.86
OrderedSquarified	1.28	0.05	0.86
Squarified	1.16	0.54	0.81
Strip	1.27	0.84	0.68

the *ordered squarified* layout, sets of randomly sized nodes were created. Each treemap consisted of 100 nodes each given a random size drawn from a log normal distribution, consistent with the simulations reported in Table II of [2]. Nodes were laid out using the *squarified*, *orderedSquarified*, *slice and dice* and *strip* (with lookahead) algorithms, and the layout statistics calculated. The simulation was repeated 1000 times, taking the mean layout statistic for all realizations. The results are summarized in Table 2. An example set of nodes from this simulation, laid out with the *squarified* and *orderedSquarified* algorithms inside a rectangle of aspect ratio 2 is shown in Figure 5.

Table 2 and Figure 5 reveal that the *orderedSquarified* layout results in greater position-order consistency than both the *squarified* and *strip layouts*. Its readability score is significantly lower, so it may not be a suitable layout for queries of categorical data in the form of "*where is node x?*", but it may be more suitable for queries that are concerned with ordered trends and general comparison between node positions. Figure 5 also illustrates a problem with the *squarified* layout where an arbitrary change in positioning of nodes occurs at the point when the space remaining in an enclosing rectangle changes from a rectangular to square aspect ratio. Nodes 1-24 in the *squarified* layout are ordered in adjacent vertical columns, nodes 25-100 follow an alternating horizontal and vertical arrangement. This can create the false impression of a bimodal distribution of sizes. The *orderedSquarified* layout shows a more continuous transition across this boundary, thus avoiding this problematic artifact of the squarified layout.

3 Spatial Layout and Displacement

The *orderedSquarified* layout is an attempt to provide a more consistent mapping of one-dimensional ordering into two-dimensional space. But potentially of more use is a layout that maps two-dimensional orderings into two-dimensional space. In particular, the mapping of hierarchical spatial data. We propose here a new *spatial layout* that attempts to position each node as closely as possible to its geographic location while minimizing its aspect ratio.

The *HistoMap* layout of Mansmann *et al* [17] uses a variation of the pivot layout [23, 2] to place nodes according to their position relative to the pivot in their parent node. Here we propose an alternative strategy that is a refinement of the *orderedSquarified* layout. It simply replaces the function ***AllocatePosition(s,r)*** with one that allocates a position according to each node's geographic location rather than an arbitrary evenly spaced position.

```
Function AllocateGeoPosition (s,r) {
  Rectangle rg ← getMinEnclosingRectangle(s);
  AffineTrans t ← getTransform(rg,r);

  for each node in s {
    transform(node,t);
  }
}
```

$getMinEnclosingRectangle(\mathbf{s})$ finds the two-dimensional rectangle defined by the minimum and maximum coordinates of the centroids of the georeferenenced nodes in **s**, and $getTransform(rg,\mathbf{r})$ finds the non-rotational affine transformation that maps rg onto **r**. For non-leaf nodes that do not have a specific georeference, this is found by allocating the weighted mean centroid of its georeferenced children. If no georeferencing exists, ***AllocatePosition(s,r)*** is called instead.

An example of the spatial layout is illustrated in Figure 6 where the

Fig. 6. French *departements* showing conventional geographic distribution (left), the spatial treemap layout (center) and hierarchical spatial treemap (right). Each *departement* is given the same random nominal color in the first two representations. The hierarchical treemap sizes each *departement* according to its average insurance premium covering catastrophic risk (flood, windstorm etc.) and colors according to the variance in premium in response multiple simulations with various occupancy and building types. *Data courtesy of Willis Analytics' Model Sensitivity Analysis project.*

95 *departements* of France are represented as nodes with minimized aspect ratio and spatial layout. The treemap is, in effect, a space filling cartogram [27] that may be combined with non-spatial hierarchical data (as shown in Figure 6) or used to display a spatial hierarchy such as post codes or census enumeration districts.

Clearly there is some spatial distortion required to tesselate the enclosing space, but the objective of the layout algorithm is to preserve the *relative* spatial arrangement of nodes as best possible.

3.1 Coloring of Absolute Position

A spatial layout of nodes will attempt to preserve their *relative* spatial positioning, but since they are always scaled to fit inside an enclosing rectangle, shows very little of their *absolute* location. So it is possible for two sets of sibling nodes to be arranged in their respective enclosing rectangles in a similar fashion even if the absolute locations of the two sets are different. For some geographic interpretation, knowledge of absolute location may be beneficial (see Section 4 below). We therefore propose using a two-dimensional color mapping of location in addition to a spatial layout where absolute location is important.

Two-dimensional color schemes are less common than their three-dimensional counterparts (e.g. HSV, RGB, CIE, and XYZ) largely due to the trichromacy of normal human color perception [30]. Projecting color space into two dimensions while retaining a broad color range and preserving some systematic 2D color coordinate system is challenging. While guidance exists on bi-variate color schemes [3, 4], there is evidence that cognition of bi-variate color mappings of two data dimensions is problematic [29, 16, 30]. The cases where two-dimensional schemes are used tend to be reprojections of three-dimensional space for automated pattern recognition rather than human perception (e.g. scene object detection [32]; skin and face recognition [18, 13]), or for selected applications where a restricted color range is required (e.g. cartographic shaded relief [14]). However, we hypothesize that the similarity of easting and northing as data dimensions may make perception of bi-variate coloring a less cognitively arduous task. We have therefore adopted a two-dimensional transect though uniform three-dimensional color space. We propose use of the CIELa*b* color space that attempts to provide a perceptually uniform gamut [19] , holding L (equivalent to lightness) constant, and using the a^* and b^* axes to represent eastings and northings respectively.

The optimal scaling, translation and orientation of the a^* and b^* axes with respect to geographical coordinates will depend on the shape of geographical space and the most important regions of interest to be shown using the color space. The aim is to produce as discriminating a color variation as possible over the region of interest. Figure 7 shows a transformation developed for the Ordnance Survey of Great Britain National Grid. Outlying locations may be mapped to their nearest valid color value (e.g. the Orkney and Shetland islands in Figure 7).

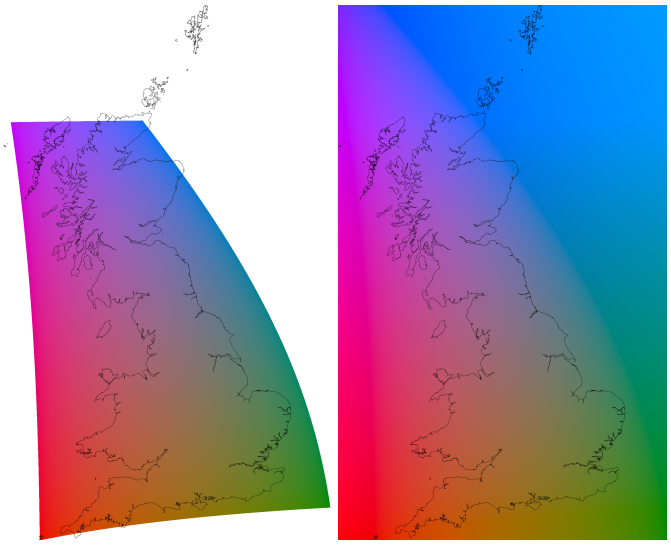

Fig. 7. CIELab colors mapped to Ordnance Survey GB locations. L is 50%, a^* represents the easting, b^* represents the northing flipped on the a^* axis. Left image shows valid RGB colors only, right image includes 'nearest' valid color for locations outside of the CIELab to RGB mapping.

The coloring scheme is illustrated in Figure 8, which shows how a uniform distribution of grid squares over the landmass of Great Britain is represented as a non-hierarchical spatial treemap using the CIELab coloring scheme. Regions of color discontinuity can be seen where the landmass is least rectangular in shape (e.g. between North Wales and South West Scotland, and the far North East of Scotland).

3.2 Identifying Displacement

Forcing most spatial arrangements of nodes into a rectangle will clearly result in some form of spatial displacement of their original georeferenced location. As has been suggested above, this displacement may be amplified if it is also a goal to produce reasonably square treemap nodes. To evaluate the spatial integrity of the spatial layout algorithm and address concerns regarding the consistency of distortion [24], a number of numeric and visual indicators of distortion may be considered.

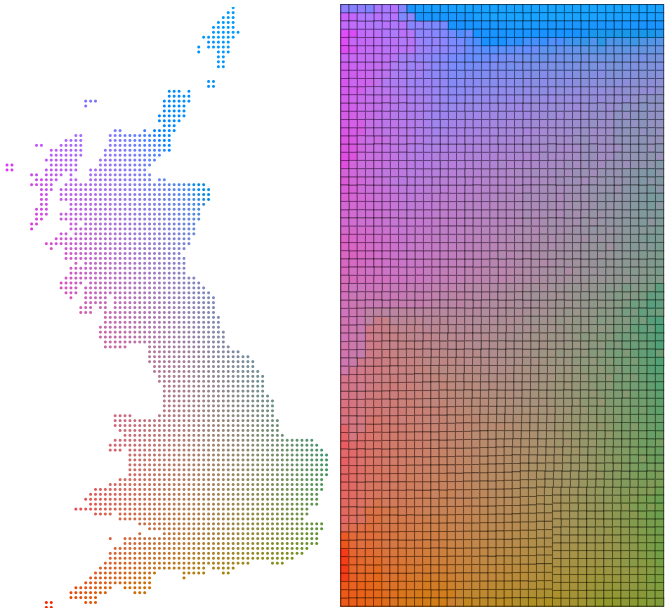

Fig. 8. Ordnance Survey National Grid square centroids in their geographic location (left) and as a non-hierarchical spatial treemap. The same CIELab color scheme is used for both images.

3.2.1 Numeric Measures

An obvious numeric metric is the average distance by which nodes have been displaced in order to tessellate their enclosing rectangle. This was calculated as follows:

$$disp_{DIST} = \frac{\sum_{i=1}^{n} d_i}{n\sqrt{A_{root}}} \quad (1)$$

Where d_i is the Euclidean distance between each node's treemap centroid and its affine transformed geographic location (to fit inside its enclosing node), n is the number of nodes and A_{root} is the area of the root node. This provides a dimensionless ratio scaled between 0 (no spatial displacement) and 1 (maximum possible displacement). d_i is always calculated relative to each node's immediate parent node so as to avoid double counting of nodes which may be displaced simply because their parent was itself displaced.

Average length of displacement hides other potentially important geographical relationships. For example, it is possible to displace two nodes by only a small amount, but to change their topological and directional relationship with each other. Likewise, nodes may be displaced by large amounts, but if many local nodes are all displaced together, their spatial relationship with each other may be preserved. Therefore, to complement the distance displacement measure, we can quantify the angular displacement between pairs of nodes. This can be calculated by taking the average angular deviation between pairs of nodes in treemap space and the same pairs in geographic space:

$$disp_{ANG} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} acos\left(\frac{\mathbf{u}_{ij}}{\|\mathbf{u}_{ij}\|} \cdot \frac{\mathbf{v}_{ij}}{\|\mathbf{v}_{ij}\|} \right) \quad (2)$$

where $\mathbf{u}_{ij}$ is the vector between each leaf node and each of its sibling leaves in treemap space, $\mathbf{v}_{ij}$ is the same vector in geographic space and n is the number of sibling leaves. The measure is scaled between 0 (no angular distortion by the treemap) and 180° (equivalent of rotating the geographic space by 180° about its centre).

Both the distance and angular metrics can be used to compare different spatial arrangements of the same set of nodes, but should be used with more caution when comparing different sets of nodes since average displacement will depend in part on how regularly spaced the geographic locations of sibling nodes are. These displacement metrics provide a useful indicator of average distortion and were used to compare the effect of minor changes to the spatial layout algorithm as well as comparison with the geographic *HistoMap* layout of Mansmann *et al* [17]. The results for the spatial layout and the HistoMap for simulated and real geographic datasets are shown in Table 3.

Table 3. Layout statistics for spatial layouts of trial datasets. 'Simulation' represents the mean of 1000 realizations of 100 log-normal randomly sized nodes with Gaussian locations; 'France' represents the 95 *departements* shown in Figure 6; 'OSGB' represents the Ordnance Survey National Grid squares shown in Figure 8; 'US Population' represents the US states sized according to population shown in Figure 9.

Dataset	Layout	Aspect ratio	$Disp_{DIST}$	$Disp_{ANG}$
Simulation	Spatial	2.66	0.21	24.3
Simulation	HistoMap	2.88	0.37	62.2
France	Spatial	1.14	0.15	18.9
France	HistoMap	1.37	0.13	12.5
OSGB	Spatial	1.02	0.19	14.1
OSGB	HistoMap	1.32	0.19	14.2
US Population	Spatial	2.26	0.17	22.1
US Population	HistoMap	7.73	0.16	17.0

Both spatial layout algorithms perform best on distributions of nodes that are more regularly spaced and evenly sized (e.g. France and OSGB). The simulation datasets were deliberately constructed to challenge the layout algorithms, with a Gaussian spatial distribution giving rise to a highly dense central region of nodes that require significant displacement to tesselate. The average aspect ratios for these data were sufficiently low to allow area-based comparisons, although the average figure does hide some small nodes with very poor aspect ratios. Distance displacement is poorer for the *HistoMap* layout than the *spatial* layout, but angular displacement much poorer for the *HistoMap* layout. This suggests that for spatial distributions with high central densities and few spatial outliers, the *spatial* layout may be more appropriate. Distance and angular displacement tends to be slightly better when applying the *HistoMap* layout to France and the US Population. This appears to be due to the fact that this layout (based on the pivot algorithm [2]) processes central nodes first and so is less affected by irregular peripheral distributions (e.g. the Brittany peninsular of NW France and the small population states of the E and NE United States).

3.2.2 Graphical Indicators

Numerical measures provide some insight into the qualities of the spatial tesselation of nodes, but they may fail to detect some systematic distortions that can result in misleading interpretations. We therefore propose using a visual indication of distance, directional, and topological distortion of geographic nodes by overlaying displacement vectors on the treemap. The displacement vector connects each treemap node to its affine transformed geographic location. In order to avoid cluttering the visual display, the quadratic Bezier arrow technique of Fekete *et al* [11] was adopted. Here the connecting vector is represented as a curve with greater curvature at the treemap node end of the line. Unlike [11], we set a single Bezier control point to 60° to the right of the vector, at a distance of 25% of the vector length giving a straighter line than Fekete *et al* proposed. This tends to keep the displacement vector within the bounds of the enclosing rectangle while still indicating the direction of the displacement. By having the maximum curvature at the treemap node end of the vector, a stronger visual indicator of any spatial clustering is given. For treemaps with relatively small numbers of nodes, these vectors can be used as additional references to aid interpretation. For those with many nodes, the vectors can be used to give a general impression of where spatial distortion is greatest and weakest. They also provide additional information on the geographic layout of data while still allowing interpretation of the treemap hierarchy [24].

Figure 9 shows the displacement vectors for a non-hierarchical car-

togram of the United States. The vectors distinguish between the western states where displacement in the treemap is uniformly towards the SE and the more complex distortion of the Eastern states where larger differences in size (population) and spatial distribution lead to some crossing vectors. In any variation of the squarified layout, relatively large nodes tend to force themselves towards the edge of their enclosing rectangle. This is because once a large node has been added, further smaller nodes added to the same row or column would have a very high aspect ratio and are therefore rejected. This is a problem for geographic patterns where the variable mapped to size is greatest towards the geographic centre of the space being mapped and significantly smaller at the periphery (see, for example, the effect of Michigan on Rhode Island, New Hampshire and Delaware in Figure 9).

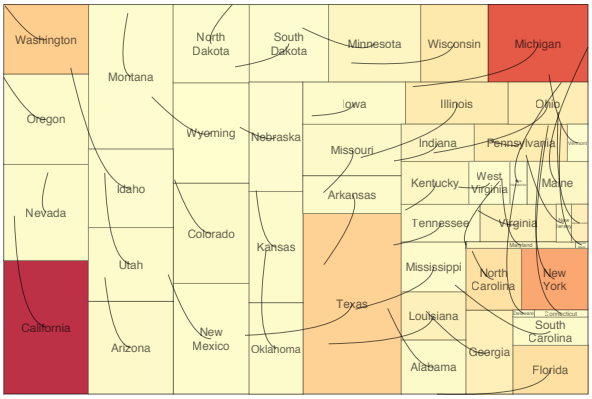

Fig. 9. US Population 2006 by State showing spatial displacement of nodes as quadratic Bezier vectors. Nodes are sized by absolute population and colored according to population change.

Figure 10 shows the 2860 landmass nodes of the OSGB 10km grid squares laid out with the *spatial* and *HistoMap* algorithms. The distortion vectors provide a visual indication of where displacement is greatest and where it is most inconsistent. Crossing vectors result in darker regions and show where there is inconsistency in spatial distortion. By combining the images with the CIELab coloring of absolute position, artifacts of the pivoting process in the *HistoMap* layout can be seen as discontinuities of color at $1/2^n$ intervals. When used in a hierarchical treemap this has the potential to be confused with genuine hierarchical classification of data.

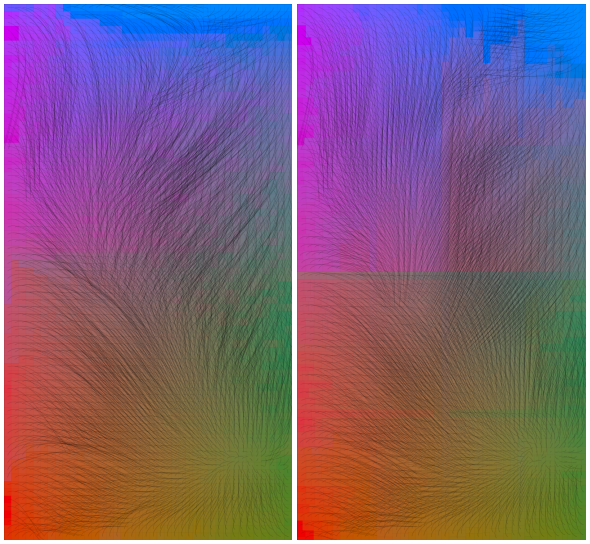

Fig. 10. OSGB grid squares showing spatial distortion of the *spatial* (left) and *HistoMap* (right) layouts. Nodes colored using the CIELab color scheme described in Section 3.1

4 Case Study: Photograph Metadata Analysis

To explore the suitability of ordered treemaps for information visualization we have applied both the *spatial* and *ordered squarified* layouts to the analysis of photographic landscape image retrieval. The work is built upon the research problem and approach identified by Edwardes and Purves [8] and Dykes *et al* [7] who investigated the metadata people choose to attach to photographic images of landscape when submitted to public image archives. The purpose was to try to identify how *place* is captured in volunteered geographic information [8, 12]. This work attempted to classify photographs according to *scene types* which were further subclassified into *scene type descriptors*, derived from the Pansofsky-Shatford facet matrix for image classification [20] and Smith and Mark's *geographical kinds* [26]. These classes were extracted by performing textual analysis on photograph metadata such as titles, descriptions, tags and comments [8]. Because each photograph was of a located scene, part of that analysis involved investigating geographic patterns in the way photographs are described.

The Flickr photo sharing service (www.flickr.com) was used to extract the metadata for all photos that had been geolocated in the British Isles and contained at least one of the following scene types as tags: *mountain*, *hill*, *village*, *beach*. Photos were then subclassified according to scene types divided into the following classes: *elements* (nouns such as peak, church, sand), *qualities* (usually adjectives such as cold, green, rural), and *activities* (verbs such as walking, surfing, fishing). Selecting only photos with a geolocation accuracy of approximately 5km or better (Flickr accuracy levels 13-16), and filtering out those with 'tag spam', resulted in a set of $\tilde{5}$0,000 photographs tagged by the four scene types. Figure 11 shows the spatial treemap of these data selecting the 10 most frequent scene type descriptors for each category of scene type descriptor in each scene type. This yields a tree structure of depth 3, with 4 categories at the first level, 12 at the second and 120 at the third.

Positioning of non-leaf nodes gives a general view of relative geographical patterns in subject matter and tagging behavior. So, for example, photos tagged with 'mountain' tend to be further north-west than those tagged with 'hill'. Color can be used to identify the degree to which such relationships exist, for example that 'beach/surfing' tends to have a greater proportion of photographs in the SW than 'beach/waves'. The displacement vector overlays in this context indicate the geographic concentration of photographs. This is most clearly seen in the 'sea' nodes where photos are inevitably concentrated around the UK coastline. The 'beach/pier' node shows the dominance of Brighton pier in the south-east. The size of non-leaf nodes gives an indication of relative popularity of tag styles. So for example, elements are more common than qualities and activities for all scene types, with the contrast being strongest in photographs tagged with 'beach'. Activity tags are more common than quality tags for hills and mountains but not for villages and beaches. The combination of vector overlay and coloring of leaf nodes is useful in identifying where individual contributions or events can dominate a pattern (and could therefore be filtered out in further analysis). For example, the brown 'car', 'hillclimb', 'racing' and 'carracing' tags in the 'hill' scene type are dominantly taken by an individual at the Prescott Hill motor racing circuit in Southern England.

It is possible that direct analysis of submitted photographs in addation to their volunteered metadata may help to identify what it is that contributors use to define place. Figure 12 shows the mean image color of each of the 50,000 photographs classified by scene type and scene type descriptor. Using the spatial layout it is possible to explore whether there are any geographic patterns in this color variation. Figure 12 suggests that scene type descriptor is probably more strongly correlated with photo color than geography (e.g. hills are greener than any other scene type; most color quality tags are associated with the color they describe, but 'white' and 'light' tagged photos appear darker than 'black' tagged photos. Where spatial layout does play a useful role is in identifying spatially clustered photos of a similar average color. These tend to be multiple photographs submitted by the same contributor of the same event.

Some caution needs to be exercised in assessing assemblages of

Fig. 11. UK Flickr photos categorised by scene type (beach, hill, mountain, village) and scene type descriptor (e.g. sky, blue, winter, surfing) with absolute location shown with spatial displacement vectors and color.

colored pixels though, as the ordering itself can affect the impression of the distribution of colors. The six treemaps shown in Figure 13 all show exactly the same data, but re-ordered according to different criteria and layout algorithms. The top row of Figure 13 shows a subgraph of the tree where leaf nodes have been ordered using the *orderedSquarified* layout. In each of the three examples, the same set of mean colors have been ordered according to the 3 principal components of the RGB color space. The first component is approximately a transect though color value, the second along a blue-orange transect and the third along a green-magenta transect. A very different visual impression of the same set of colors can be given simply by changing the (arbitrary) ordering of colored nodes. The bottom row shows the same nodes ordered by just the first principal component of color, but laid out using the *squarified*, *pivot by middle* and *strip map* algorithms. In each of these cases, discontinuities in color can be seen that don't reflect properties of the data, but rather artifacts of the layout algorithm. These include localized clusters of orange and blue nodes, diagonal clusters of dark pixels and apparently nested square clusters that simply reflect the pivot points used in the layout algorithm.

5 Conclusion

We have proposed a pair of new algorithms that attempt to increase the cognitive plausibility of treemap layouts by relating the two-dimensional positioning of nodes in a treemap more closely to the properties of the data they represent. While attempts to do this have been made in the past, most notably by Bederson *et al* [2], they have tended to focus on the problem of identifying a particular node within an ordered list. In our work, we have attempted to lay out nodes to allow trends and comparisons between nodes to be made. The geography of data is one obvious example, exploited by our *spatial* layout, where location is an important property that should be reflected in the information graphic. Where geographic information is not available, we argue that the *ordered squarified* layout follows the distance-similarity metaphor more closely by minimizing arbitrary spatial discontinuities that do not reflect properties of the data.

We have considered a number of metrics that might be used to measure the success of a layout algorithm. We argue that *readability*, while summarizing the cognitive effort required to follow an ordered sequence of nodes, does not necessarily reflect the effort required to assess trends or comparisons between nodes. Instead we have used correlation between node order and distance from the origin of a parent node. For spatial layout of data with a geographic component, measures of distance and angular displacement can be used to assess the degree to which the treemap reflects the spatial properties of the data it represents. This has allowed us to make comparisons between our *spatial* layout and the *HistoMap* layout [17], identifying the types of spatial pattern that are best represented by each layout. Yet summary statistics of overall spatial distortion or consistency fail to detect the impact of discontinuities in layout. These may be better reflected by graphical means such as displacement vectors and spatial coloring. We have used these techniques to identify the spatial patterns and complex geographies of volunteered photographic metadata as well as drawing attention to the advantages of the *spatial* layout over pivot-based algorithms.

Further developments of this work include the identification of more discriminating metrics of layout inconsistency. In particular, measures that identify systematic but arbitrary discontinuities in layout. These might include geometric inconsistencies as well as topological ones. Suitable metrics may help in refining the layout algorithms to better reflect the geographic distributions of the data they represent.

Fig. 12. Categorized UK Flickr photos with color representing the mean color of the photograph represented by each leaf node.

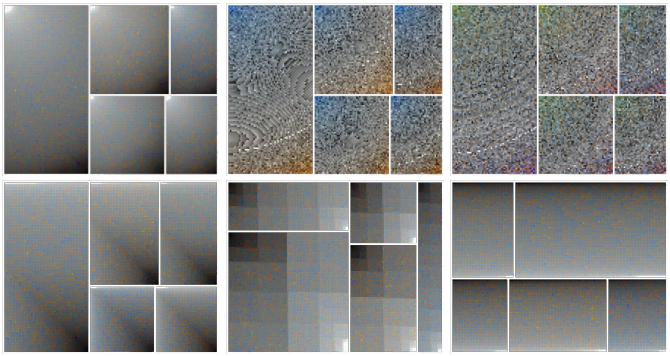

Fig. 13. Selected treemap nodes showing six orderings of mean photo color. *Top row*: Nodes ordered by the three principal components of image color arranged using the *OrderedSquarified* layout. *Bottom row*: Nodes ordered by the first principal component of color arranged using the *squarified* (left), *pivot by middle* (centre) and *strip* (right) layouts.

ACKNOWLEDGEMENTS

GB outline (Figure 7) and National Grid squares (Figures 8 and 10), crown copyright/database right 2008. An Ordnance Survey/EDINA supplied service. The authors are also grateful for insightful discussion with Ross Purves and Alistair Edwardes at the University of Zurich on the use of treemaps for photographic image retrieval.

REFERENCES

[1] T. Barlow and P. Neville. A comparison of 2-d visualizations of hierarchies. *IEEE Symposium on Information Visualization*, pages 131–138, 2001.

[2] 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*, 21:833–854, 2002.

[3] C. Brewer. Guidelines for selecting colors for diverging schemes on maps. *Cartographic Journal*, 33:79–86, 1996.

[4] C. Brewer. Selecting good color schemes for maps. *www.colorbrewer.org*, 2002.

[5] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42, 2000.

[6] N. Cawthon and A. V. Moere. The effect of aesthetic on the usability of data visualization. *11th International Conference on Information Visualization (IV '07)*, pages 637–648, 2007.

[7] J. Dykes, R. Purves, A. Edwardes, and J. Wood. Exploring volunteered geographic information to describe place: Visualization of the 'geograph british isles' collection. In D. Lambrick, editor, *GISRUK 2008*, pages 256–267, Manchester, UK, 2008. Manchester Metropolitan University.

[8] A. Edwardes and R. Purves. A theoretical grounding for semantic descriptions of place. *7th International Symposium on Web and Wireless Geographic Information Systems*, 4857:106–121, 2007.

[9] S. Fabrikant. Cognitively plausible information visualization. In J. Dykes, A. MacEachren, and M.-J. Kraak, editors, *Exploring Geovisualization*, pages 667–690, London, 2005. Elsevier.

[10] S. Fabrikant, D. Montello, M. Ruocco, and R. Middleton. The distance-similarity metaphor in network-display spatializations. *Cartography and Geographic Information Science*, 31:237–252, 2004.

[11] J.-D. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Overlaying graph links on treemaps. In *InfoVis03*, pages 82–83, 2003.

[12] M. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69:211–221, 2007.

[13] S. Jayaram, S. Schmugge, M. Shin, and L. Tsap. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. In *CVPR'04*, pages 813–818, 2004.

[14] B. Jenny and L. Hurni. Swiss-style colour relief shading modulated by elevation and by exposure to illumination. *Cartographic Journal*, 43:198–207, 2006.

[15] D. A. Keim. Enhancing the visual clustering of query-dependent database visualization techniques using screen-filling curves. *Proceedings of the IEEE Visualization '95 Workshop on Database Issues for Data Visualization*, pages 101–110, 1995.

[16] A. MacEachren, C. Brewer, and L. Pickle. Visualizing georeferenced data: Representing reliability of health statistics. *Environment and Planning A*, 30:1547–1561, 1998.

[17] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda. Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats. *IEEE Transactions on Visualization and Computer Graphics*, 13:1105–1112, 2007.

[18] V.-E. Neagoe. An optimum 2d color space for pattern recognition. In *IPCV'06*, volume 2, pages 526–532, Las Vegas, 2006. CSREA Press.

[19] P. Robertson and J. O'Callaghan. The generation of color sequences for univariate and bivariate mapping. *IEEE Computer Graphics and Applications*, 6:24–32, 1986.

[20] S. Shatford. Analyzing the subject of picture: A theoretical approach. *Cataloging and Classification Quarterly*, 6:39–62, 1986.

[21] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11:92–99, 1992.

[22] B. Shneiderman. Treemaps for space-constrained visualization of hierarchies. *www.cs.umd.edu/hcil/treemap-history*, 2006.

[23] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *Infovis01*, pages 73–78, 2001.

[24] A. Skupin and S. Fabrikant. Spatialization methods: A cartographic research agenda for non-geographic information visualization. *Cartography and Geographic Information Science*, 30:99–119, 2003.

[25] smartmoney.com. Map of the market, 2008.

[26] B. Smith and D. Mark. Geographical categories: an ontological investigation. *International Journal of Geographic Information Science*, 15:591–612, 2001.

[27] W. Tobler. Thirty five years of computer cartograms. *Annals of the Association of American Geographers*, 94:58–73, 2004.

[28] Y. Tu and H.-W. Shen. Visualizing changes of hierarchical data using treemaps. *Visualization and Computer Graphics, IEEE Transactions on*, 13:1286–1293, 2007.

[29] H. Wainer and C. M. Francolini. An empirical inquiry concerning human understanding of two-variable color maps. *The American Statistician*, 34:81–93, 1980.

[30] C. Ware. Color. *Information visualization: Perception for design*, pages 97–144, 2004.

[31] R. Williams. Circle packings, plane tessellations, and networks. *The Geometrical Foundation of Natural Structure: A Source Book of Design*, pages 34–47, 1979.

[32] T. T. Zin, S. S. Koh, and H. Hama. Optimal color space for relative color polygons. *IEICE Electronics Express*, 4:106–113, 2007.