

City Research Online

City St George's, University of London

Citation: Bouts, Q. W., Dwyer, T., Dykes, J., Speckmann, B., Goodwin, S., Henry-Riche, N., Carpendale, S. & Liebman, A. (2016). Visual Encoding of Dissimilarity Data via Topology-Preserving Map Deformation. IEEE Transactions on Visualization and Computer Graphics, 22(9), pp. 2200-2213. doi: 10.1109/tvcg.2015.2500225

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/15023/

Link to published version: https://doi.org/10.1109/tvcg.2015.2500225

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, 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. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

Visual Encoding of Dissimilarity Data via Topology-Preserving Map Deformation

Quirijn W. Bouts, Tim Dwyer, Jason Dykes, Bettina Speckmann, Sarah Goodwin, Nathalie Henry Riche, Sheelagh Carpendale, Ariel Liebman

Abstract—We present an efficient technique for topology-preserving map deformation and apply it to the visualization of dissimilarity data in a geographic context. Map deformation techniques such as value-by-area cartograms are well studied. However, using deformation to highlight (dis)similarity between locations on a map in terms of their underlying data attributes is novel. We also identify an alternative way to represent dissimilarities on a map through the use of visual overlays. These overlays are complementary to deformation techniques and enable us to assess the quality of the deformation as well as to explore the design space of blending the two methods. Finally, we demonstrate how these techniques can be useful in several—quite different—applied contexts: travel-time visualization, social demographics research and understanding energy flowing in a wide-area power-grid.

Index Terms—Dissimilarity, Maps, Cartographic Visualization, Multidimensional Scaling, Deformation

1 Introduction

Using map deformation to show quantitative data (such as population or GDP) is a well-known technique in the fields of Geographical Information Science and Information Visualization. For example, so-called *contiguous area cartograms* [17], [28], [38] seek to deform the map such that the area of each geographic region corresponds to a single underlying quantitative value. Such cartograms try to preserve the adjacencies and the shape of the underlying geographic regions (i.e. their topology) while modifying the area. Cartograms have been used effectively to show a variety of attributes at global and more local scales (e.g. [10]).

Contiguous (or continuous) area cartograms are probably the most well-known use of deformation techniques to present data in the context of a map, though we review some others in Section 2. They have some obvious limitations in terms of how faithfully they represent the data. Current techniques cannot guarantee that the ratios of areas between any pair of regions precisely match the ratios of their data quantities. Furthermore, when the map is highly deformed the original geography can be difficult to recognize. Nevertheless, cartograms have a certain popular appeal since they create a juxtaposition between data and geography that may be quite surprising. Here we present a map deformation technique which—like continuous cartograms—preserves topology and balances preserving geographic shape with conveying data. However, while cartograms represent simple scalar values associated with regions, our method takes a complete weighted graph between geographic locations and moves the locations such that the distance between them corresponds as closely as possible to the weights associated with the edges of the graph (see Fig. 1). In particular, we encode the dissimilarity between the given locations as weights in the complete graph and hence the distance of the locations in the deformed map is related to data dissimilarity. This allows us to compare distances between multiple locations and data attributes concurrently. We describe our algorithm for computing such a deformation in Section 3.

In Section 4 we evaluate the speed of computation and the quality of the deformations. We find that the algorithm is fast enough to allow the deformation to be computed in response to interaction. This allows the composition of the dissimilarity metric or the data source itself to be modified in real time. In Section 5 we acknowledge the limitations of showing complex dissimilarity data through deformation alone, and explore the design space of augmenting the maps with visual overlays to overcome these limitations.

In Section 6 we describe two case studies where we use our deformation to analyse complex real-world multivariate data (socioeconomic data from the UK) and to convey the structure of a dynamic weighted network (a power-grid from Australia). One of our observations in developing these techniques is that map deformation is most useful in an interactive context. So please see the accompanying video for real-time captures of our interactive system.[1]

2 Related work

The idea to deform a map in a way that shows underlying data has precedent in contiguous cartograms, which represent scalar values via the areas of their deformed regions. There have been many papers dealing with the computational aspects of creating such cartograms. The most frequently used method was proposed by Gastner and Newman [17] and is based on diffusion models. An extensive

- *Q. W. Bouts and B. Speckmann are with TU Eindhoven, The Netherlands. E-mail: [q.w.bouts, b.speckmann]@tue.nl.*
- *Tim Dwyer, Sarah Goodwin and Ariel Liebman are with Monash University, Australia. E-mail: [tim.dwyer, sarah.goodwin, ariel.liebman]@monash.edu.*
- *Jason Dykes is with City University London, UK. E-mail: j.dykes@city.ac.uk.*
- *Nathalie Henry Riche is with Microsoft Research. E-mail: nath@microsoft.com.*
- *Sheelagh Carpendale is with The University of Calgary, Canada. E-mail: sheelagh@cpsc.ucalgary.ca.*

Manuscript received ...

1. http://tinyurl.com/mrjt3ea

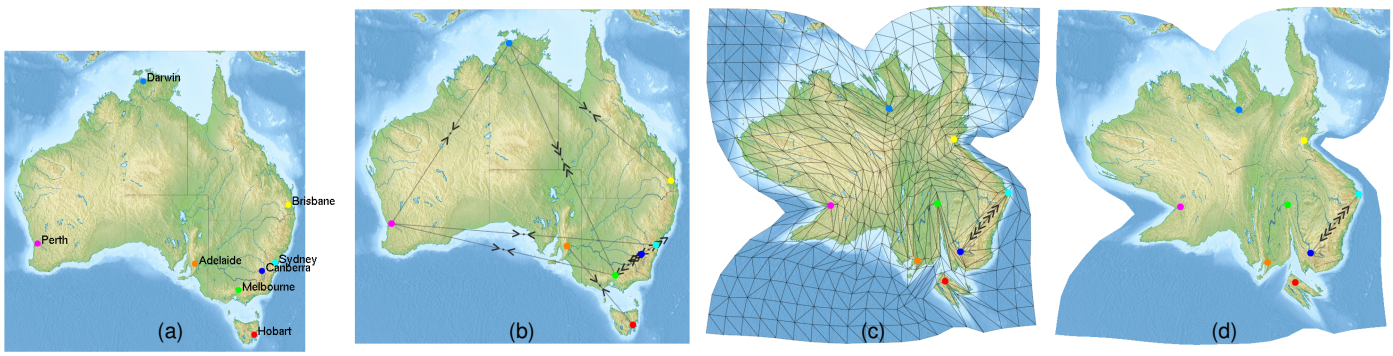

Fig. 1. House prices in Australia: (a) geographic input map; (b) glyphs showing significant dissimilarity errors (full dissimilarities are a complete graph); (c) deformed map with mesh overlay; (d) deformed map without mesh where all but one edge length are well realised.

set of examples can be found at `www.worldmapper.org`. Another family of solutions—conceptually closer to our method—is based on the "rubber-sheet" paradigm (see e.g. Sun *et al.* [37]) where a continuous transformation function from base map to area-optimized cartogram is iteratively inferred. It is made efficient through a quadtree decomposition that gives more detail in complex regions. In contrast to such techniques, which derive a continuous transformation function, in the next section we describe an approach based on deformation of a tessellated image that is then easily applied using hardware accelerated texture mapping.

Our input is a complete weighted graph between geographic locations. Deforming a map such that the edge weights can be realized as edge lengths has extensively been studied in the special case that all locations are nodes of a transportation network and edge weights correspond to travel times. The resulting maps are often called *space-time maps* or *travel time maps*, but are also referred to as *distance cartograms* or *linear cartograms*. Solutions exist for the case that the whole graph is realized (e.g. [1]), or only a planar subset (e.g. [35]), or only a star subgraph, i.e. travel times from one source only (e.g. [3]). In the single source case another variant are so-called *isochrones* which deform maps such that the Euclidean distance corresponds to travel time (e.g. [23], which includes a user study). In all cases it is important to preserve the topology of the map and keep a strong resemblance to the input. Many methods are able to (mostly) avoid topology violations and, in particular for single source scenarios, some resemblance to the original map can be kept. However, this often comes at high computational costs. Furthermore, transportation networks align comparatively well with the underlying geography and exhibit some sort of transitivity. We consider arbitrary graphs, possibly with high and unequally distributed weights and still guarantee topology preservation. In addition, by combining a fast multidimensional scaling and texture mapping we achieve interactive speeds (see video).

Another geographic deformation method is offered by Böttger *et al.* [6]. They deform the map so that a schematized metro-map can be overlaid such that the locations of the stations in the metro map correspond to the appropriate locations in the deformed geography. Their grid-warping method guarantees a topology preserving deformation; however, their figures take hours to produce. Furthermore, their target metro map configuration is both static and topologically fairly consistent with the original geography.

Haunert and Sering [20] suggest drawing road networks with focus regions such that deformation of detailed parts of the map is minimized. This is achieved by overlaying the map on a regular grid and generating constraints to prevent the grid topology from being violated. Although the details are different, their method is related to our approach, in that they use sequential quadratic programming with the constraints generated between iterations. However, the nature of the constraint model and the use of a standard solver (CPlex) mean that their method requires several seconds to solve a single instance prohibiting any interactivity. While their application is different and arguably simpler than our goal of deforming the map to show arbitrary dissimilarity relationships, we are inspired by Haunert and Sering to employ sequential quadratic programming techniques to achieve impermeable topological boundary constraints.

Lin *et al.* [29] offers a different method for road networks. Their algorithm uses a labeled graph representation of the road network. Different layouts for a small subset of the roads are supplied as input, this *mental map* then guides the deformation. A grid mesh is overlaid and an iterative error minimization process fits the network to the mental map. The error metric can use different weights for the roads allowing main roads to behave more rigidly than secondary roads. In contrast to the method of Haunert and Sering it does not impose any hard topological constraints. In Section 6.1 we demonstrate our technique's ability to perform a similar transport-network-based deformation but strictly respecting topology and in a fraction of the time required by [20].

Weng *et al.* [40] propose a method for 2D shape deformation based on nonlinear least squares optimization. Similar to our method they triangulate the shape and allow the users to move vertices as control points. As the control points are moved they try to preserve both area constraints on the triangles and the Laplacian coordinates on the boundary curve. Igarashi *et al.* [25] propose a shape deformation method called ARAP. The deformation is done by solving two least-squares problems. The first minimizes an error metric that prevents shearing and non-uniform stretching but permits rotation and uniform scaling. The second takes the result and scales the triangles. They "solve" collisions by assigning a depth to overlapping triangles such that the shape looks natural. Both shape deformation methods work

at interactive speeds. Since our maps do not have a boundary curve defined and we do not allow self-intersections these methods can not be applied in our setting.

Lipman [30] proposes a constructive definition of generic convex spaces of piecewise linear mappings with guarantees on the maximal conformal distortion, as-well as local and global injectivity of their maps. They use this to provide bounded distortion versions of algorithms like the ARAP method mentioned above, hence addressing the self intersection problems. Their techniques offer mesh deformation with minimal distortion at (near) interactive speed.

The method introduced in the following section can be thought of as an application of mesh deformation techniques such as these to deformation of a map to optimize a point-to-point dissimilarity measure while preserving topology. In contrast to other image warping techniques our method warps subject to constraints that minimize stress for dissimilarity data. Thus, it is a new combination of multidimensional scaling techniques and mesh deformation.

3 Topology Preserving Multidimensional Scaling

Multidimensional Scaling (MDS) [5, Chap. 8] is a method for obtaining two-dimensional plots representing a set of high-dimensional data points. This is achieved by minimizing a *stress* function over positions of the data points. A plot with minimal stress has its points arranged such that the distance between them is (as much as possible) proportional to the dissimilarity between corresponding data elements in their parameter space. Minimizing stress subject to rigid constraints that, for example, require a theoretical model to be satisfied by the geometry of the plot, is also fairly well known [5, pp. 230–236]. More recently, Dwyer *et al.* [12], [13] explored constrained stress minimization in the presence of geometric constraints useful in interactive graph-drawing and diagramming applications. In this paper, we develop techniques that extend these concepts to provide multidimensional scaling of data points in a deformable mesh onto which the original map imagery is mapped through piecewise-affine transformations. We allow the mesh to be deformed, but use constraints to require that mesh and data vertices cannot pass through mesh edges.

Given a set of n data elements and a measure d_{ij} indicating the dissimilarity between any two elements $i, j \leq n$, the goal of multidimensional scaling is to find a visualizable (2- or 3-dimensional) arrangement of points (corresponding to data elements) which minimizes the *stress* function:

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} \left(d_{ij} - \mathrm{dist}(i,j)\right)^2 \tag{1}$$

where $\mathrm{dist}(i,j)$ is the Euclidean distance between points i and j in the 2D arrangement and w_{ij} is a weightwhich can be used to control the influence of a particular dissimilarity.

As an illustrative example, consider Fig. 2. We mark the Australian capital cities on the map and (in brackets) the average percentage increase in house prices in those cities in 2013. We analyse the similarity of these changes using MDS by choosing the ideal distance (d_{ij}) between a pair of cities to be the difference between these values. Since the ideal distance between Melbourne and Darwin is $6.8 - 6.0 = 0.8$ while between Sydney and Canberra we have $11.4 - 0.6 = 10.8$, we would expect an MDS plot to place Sydney and Canberra much further apart than Melbourne and Darwin. As we see in Fig. 2b this is exactly what happens.

This is a trivial univariate example but we could just as easily derive ideal distances between high-dimensional data elements. That is, we could define the ideal distance to be inversely proportional to correlation between a series of price increases over multiple time periods. Section 6 describes more complex multivariate data applications.

In this paper we try to combine the views in Figs. 2 and 2b. The first challenge is to deform the map to follow points as we move them to show dissimilarities. In the next section we show how *stress* can serve as a quality measure for fitting points to dissimilarities and also to model an approximation of elastic "stress" in a rubber-sheet-like mesh.

The resulting method is very fast compared to existing deformation techniques used in continuous cartogram techniques (see Section 2) and it is also the first topology-preserving technique applied to multi-source dissimilarities. It is fast enough to support real-time changes in the underlying data or choice of dissimilarity metric and see the resulting deformation immediately (a fraction of a second).

3.1 Decomposing Map Imagery with a Mesh and Modelling Deformation with MDS

In Fig. 3a we overlay an image containing a topographic map of Australia with a triangular mesh. The vertices of the mesh are a combination of geographic locations (the capital cities of Australia) and "helper points" chosen to provide a reasonable coverage of the map. A Delaunay triangulation over these vertices gives us the edges of the mesh.

The vertices should be sufficiently distributed that the resulting mesh can be deformed reasonably freely without individual triangles becoming too angular and introducing noticable discontinuities. The helper points not only add bendpoints to the mesh but the stress their edges contribute when the mesh is deformed acts as a regularizer and helps to preserve the geography. Many strategies are possible for placing the helper points. We considered both uniform (grid) and non-uniform placements.In practice we find a regular grid of helper points works quite well. Placing points intelligently with respect to geographic features might give better results but this only works if their geometry is known. A disadvantage of non-uniformly placed points is that due to the quadratic nature of the stress function the optimal position of the data points can change if many helper points with short edges surround it. This makes the relation between the dissimilarity and the deformation less clear. Conceptually, our "rubber-sheet" model is achieved by treating mesh edges as springs whose relaxed length is the same as in the starting mesh. Our goal is that any forced displacement of a vertex will be evenly interpolated by displacements of vertices in the surrounding mesh. This is achieved in the *stress* model of Eq. 1 by includingour helper points. Then, for any pair of points i, j connected by a mesh edge, we set the ideal separation d_{ij} to the length of that edge in the undeformed map.

These mesh-edge separation terms together with the usual stress over data points gives us a stress function

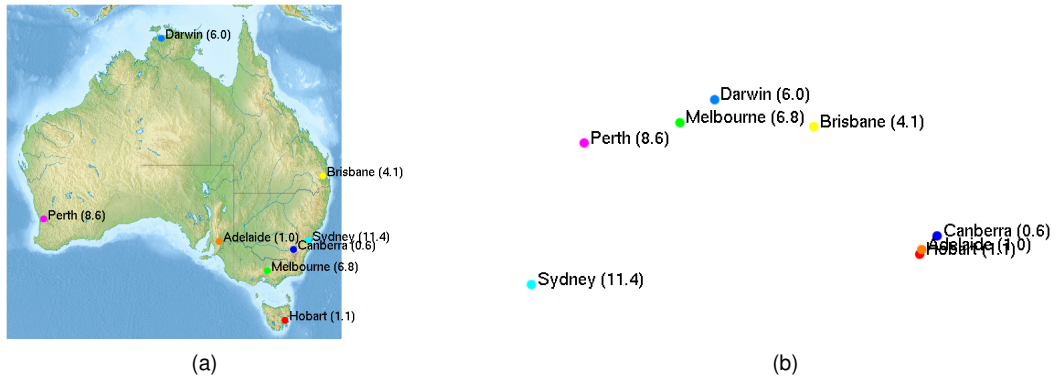

Fig. 2. A simple univariate example using MDS to visualize dissimilarities between geographic locations. Here dissimilarity is simply difference between data values at each location. The data is percentage increase in house prices in Australian capital cities in 2013 (`http://www.abs.gov.au/`). a) Australian capital cities. b) In the MDS plot, two distinct clusters and one outlier become clear. There are distinct differences between this "data landscape" and the original geography.

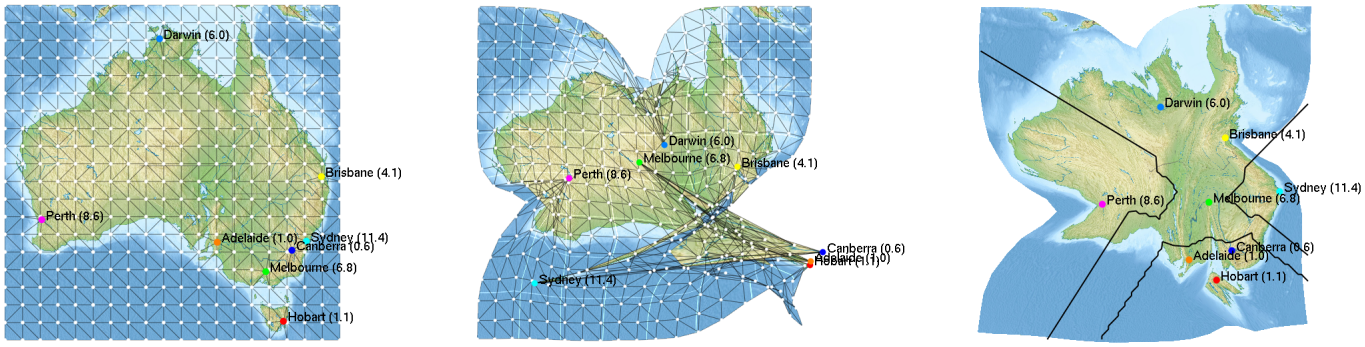

Fig. 3. Segmenting map imagery with a triangular mesh allows us to deform the mesh and use affine texture mapping to have the image follow. a) Triangular mesh over Australia, data points are capital cities. b) Using dissimilarity weights between locations in addition to stress. Minimizing stress reproduces the MDS arrangement from Fig 2 and folds the map almost beyond recognition. c) Using topology preserving constraints, which prevent the outlier Sydney from migrating to the West Coast. Clusters are still able to move together. We add isolines to highlight areas with similar house-price increase.

that models both the degree of fit of the data points to their ideal separation and the degree of deformation of the mesh. We can set different weights (w_{ij}) for the mesh-edge terms versus the data-dissimilarity terms. Assigning higher weights to data points gives a better fit to the dissimilarities whereas higher weights on the mesh edges imposed by the helper points gives less distortion of the geography.

A minimal stress result of the Australian house-price data used earlier is shown in Fig. 3b. Since the arrangement of the points is very different from the geographic topology, the map is distorted beyond recognition. In the next section we show how to minimize stress subject to a topology preserving constraint to better preserve the geography.

3.2 Preserving Mesh Topology

The idea behind our topology preservation constraint is simple. We require that the orientation of triangles in the mesh is preserved. Stress is a function of variables corresponding to positions of the data and helper points in our mesh. At each iteration we reduce stress by altering these variables according to gradient-related descent steps. When a constraint is violated by such a descent step we satisfy the constraint by altering variables as little as possible end hence ensure that the system will eventually converge to a configuration that satisfies the Karush-Kuhn-Tucker conditions [8, p. 224] for the stress function and hence, is a local minimum.

In practice, this means that after taking an unconstrained descent step, if we find an inverted triangle then we must solve a least-squares problem to find the minimal movement vectors that correct the triangle's orientation. It is easy to show that the points that minimally correct an inverted triangle lie along the orthogonal line of best fit for its vertices (see Fig. 4). We can also require minimum height triangles by projection of the triangle of the same base length by solving a simple procrustes problem on the three vertices, following [13]. When using a positive constraint on the minimum triangle height we not only prevent triangles from inverting, but also from collapsing. In the resulting mesh and piecewise-linear mapped image the (local) topology is preserved in the sense that every point on the map has the same neighborhood. More formally, we have a bijective map between the original and transformed map image. When only triangle inversions are prevented—and hence triangles are allowed to collapse—we have an injective map.

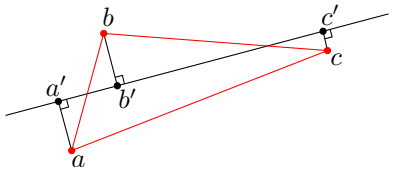

Fig. 4. A triangle is uninverted through minimal displacement of its vertices by projecting them to their orthogonal regression line. Precisely, for the triangle abc we find the points a', b', c' such that $|aa'|^2+|bb'|^2+|cc'|^2$ is minimized, subject to the constraint that a', b', c' lie on a line.

The high-level flow of this method is given below:

Algorithm *DynamicConstraintMDS*
1. **do**
2. Compute gradient info for *stress* at current config.
3. $\vec{d} \leftarrow$descent vector from gradient info
4. $\alpha \leftarrow$optimal unconstrained step-size for $\vec{d}$
5. **foreach** Point $\mathbf{p}$ in current config.
6. **do**
7. $\vec{d}_{\mathbf{p}} \leftarrow$component of $\vec{d}$ for point $\mathbf{p}$
8. $l \leftarrow$limiting factor for faces adjacent to $\mathbf{p}$
9. $\mathbf{p} \leftarrow \mathbf{p} + \min(l, \alpha)\vec{d}_{\mathbf{p}}$
10. **foreach** Triangle(a, b, c)
11. **do if** Constraint Violated
12. **then** $(a, b, c) \leftarrow$ProcrustesProject(a, b, c) [13]
13. **foreach** Edge under stress
14. **do if** Flipping topologically safe
15. **then** Flip Edge
16. **while** StressReduction $>$ epsilon

The algorithm is essentially a constrained steepest-descent method. Details for computation of gradient information, unconstrained descent vector $\vec{d}$ and optimal stepsize α (lines 2–9) for a function similar to *stress* are described in [12].

We project inverted or minimum-height violating triangles iteratively (*DynamicConstraintMDS* lines 10-12) following a procedure that was shown to be reasonably convergent in [11]. By "reasonably" we mean that the mesh is generally able to flatten itself, but if the movement deltas at each iteration are too large, the mesh can become tangled such that the iterative projection merely cicles the inversions. To prevent this we limit movement deltas of each vertex relative to the minimum size of its adjacent faces (*DynamicConstraintMDS* line 8). The result of the Australian Houseprice MDS subject to the topology constraints is shown in Fig. 3c.

3.3 Dynamically Modifying the Mesh

Because of the triangle height constraint picking the "correct" mesh can greatly influence the stress reduction achieved. Since it is very hard or, in an interactive setting, even impossible to predict where points will move we considered updating the mesh while the algorithm is running.

We considered two ways of modifying the mesh, modifying edges and adding or removing points. We found that changing the edges was sufficient to improve the results without complicating the method or introducing instability.

Edge flipping is a standard remeshing trick [24] which is often used to improve angles or change the local configuration to allow different operations. In algorithm *DynamicConstraintMDS* (lines 13–15) we use flips to allow points to move more freely. An edge flip is defined on two adjacent triangles, i.e. two triangles sharing an edge. By flipping this edge so its endpoints become its opposing points we change the direction in which minimal triangle height is measured resulting in more freedom of movement in the direction of the new edge. We illustrate a flip and its influence on vertex movement in Fig. 5.

To preserve the topology of the map we impose some restrictions on our flips (*DynamicConstraintMDS* line 14). Let $t_1 = (a, b, c)$, $t_2 = (a, b, d)$ be triangles with a', b', c' and d' the map-coordinates of vertices a, b, c and d. If (a', c', d') encloses b' or (c', d', b') encloses a' we can not flip ab.

An advantage of allowing edge flips is that one may be able to get closer to an optimal solution, but care should be taken not to create thin, very stretched triangles as these limit recognizability. A caveat in the interactive setting is that the flips can result in sudden changes in the deformation which may be distracting or hard to follow by the user.

4 Stress Reduction: Efficacy and Efficiency

In this results section we compare the three dissimilarity deformation techniques described above:
Unconstrained: MDS of the dissimilarity edges between data points (*data edges*) and mesh edges without any guarantee of topology preservation;
Constrained: As above but we introduce triangle projection to guarantee topology preservation;
Constrained Dynamic Mesh: Topology preservation and mesh modification (edge flips) to allow more movement.

We evaluate these techniques on data sets described in this paper and at various grid resolutions in terms of:
Efficacy in reducing stress of both data edges and mesh edges. To assess this we define: *Data Stress* to be the sum of stress terms across data edges and *Geographic Stress* the sum across mesh edges. **Efficiency** comparing running times for a fixed number of iterations to indicate responsiveness of interactive usage but also—as we will discuss—a variable number of iterations may be needed in a batch process.

Our experiments are shown in Table 1. The data sets have different numbers of data points. It should be noted that these data sets are quite small (at most 30 data points). Computationally larger sets can be handled, but unless the dissimilarities conform to the topology to some degree, they will be hard to overlay in a meaningful way. We discuss this limitation in more detail in our case studies in Section 6.

Fig. 6 charts the data and geographic stress by iteration in each of the three approaches tested. It is clear that introducing constraints slows data stress reduction. Essentially,

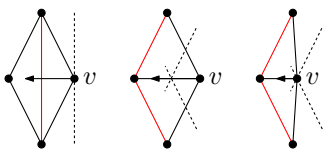

Fig. 5. Initially v is not able to move to the left because of the minimal height constraint (dotted line) imposed by the red edge in the left figure. By flipping the edge (middle figure) v can now move further to the left (right figure)

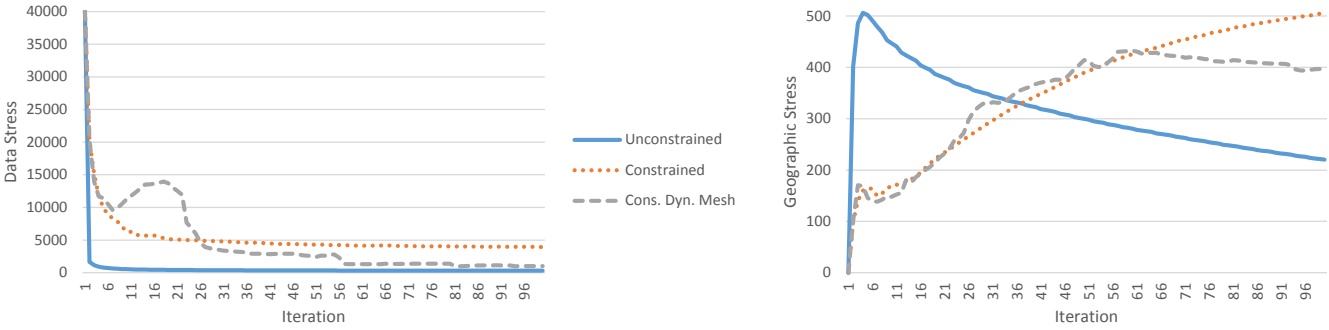

Fig. 6. Data and geographic stress levels for the Australian house price data across 100 iterations using different deformation methods.

Data Set (Initial Stress)	Data Points	Data Weight	Grid	Vertices	Unconstrained Run Time	Unconstrained Stress Data	Unconstrained Stress Geo	Constrained Run Time	Constrained Stress Data	Constrained Stress Geo	Constrained Dyn. Mesh Run Time	Constrained Dyn. Mesh Stress Data	Constrained Dyn. Mesh Stress Geo
Aus House Prices (40044)	7	5	20x20	407	440	328	262	718	4202	388	676	2615	469
			30x30	907	1000	426	379	1549	3771	556	1489	5432	436
			40x40	1607	1876	563	486	2660	10502	246	2596	17077	443
Power Grid (1355)	30	3	20x20	430	569	706	125	788	819	101	785	754	108
			30x30	930	1143	758	152	1632	986	123	1584	791	145
			40x40	1630	1974	887	131	2787	973	104	2732	944	107
UK (18462)	10	30	20x20	410	467	2429	619	765	3767	551	736	3066	630
			30x30	910	1075	2805	855	1603	7626	166	1558	7152	139
			40x40	1610	1998	3150	1068	2876	7898	358	2834	9951	82

TABLE 1
Running times and stress levels for different data sets, using different deformation methods and different grid sizes. We run 100 iterations of each method. All times are in milliseconds and are averages over 5 runs. Running time for a fixed number of iterations is quadratic in the number of data points and linear in the number of grid vertices. Increasing grid resolution does not necessarily lead to lower stress solutions while edge-flipping is worthwhile in this regard. See Section 4.

less movement is possible at each step, however the geographic stress is introduced in a much more controlled way. With a static mesh the stress decay is still quite monotonic and predictable. The dynamic mesh allows data points to move past each other more easily which leads to a strange stress curve that even backtracks noticeably before reaching a lower data-stress than is possible with the static mesh.

The effect of increasing grid-size (see Table 1) is also interesting. Increasing mesh resolution increases running time only linearly since we use sparse matrix operations in our stress minimization and our mesh is planar and thus sparse. Hence it is possible to obtain "smooth" continuous looking deformations quickly. However, larger grids do not lead to lower data stress. Rather the additional stress terms increase the contribution of geographic stress to the model. By increasing the data weight one can achieve similar visual results but this makes the numbers impossible to compare.

We find that using a dynamic mesh is a more effective way to achieve lower stress and improves performance since it reduces the number of triangle inversions and hence the number of projections required. However, it does not perform as well on large grid resolutions. This is because with smaller triangles the effect of flipping is less pronounced.

In summary, the relatively low-resolution grids work best to reduce data stress. We recommend that if a very smooth deformation is required that the grid resolution be increased later to allow improvements to proceed from gross to fine scale. Our default grid resolution is 20x20 which we adjust for non-square images to keep each grid cell square.

5 Visual Design

Use of deformation to convey data can be important because of its strong visual and perhaps emotional impact. However, we acknowledge that such a deformed map is an incomplete (approximate) representation of complex dissimilarities. In this section, we explore the design space of combining map deformation with overlays to give the user a more precise representation. We realized that this design space is rather large and contains many fruitful directions to pursue. The high-level scope of our initial exploration of this space is illustrated in Fig. 7. The dimensions we chose to explore are: from the regular undeformed map (left side) to the data deformed map (right side). The top row shows dissimilarities as *visual links* whose thickness encodes the data. The bottom shows the error in map distance using *error glyphs*.

5.1 Visual links

To convey both dissimilarity and geographical data a straight-forward visual encoding is to overlay visual links on a map. We can think of dissimilarities between locations on the map as a complete weighted graph, where the nodes are locations and the edges between them are weighted by similarity (according to an arbitrary metric). Nodes are located in their geographical context and linked with lines where a visual variable of the link—such as opacity or thickness—indicates the edge weight (and possibly the edge directionality). However, since maps are visually dense representations, integrating this additional layer of information needs to be done carefully. We formulate a set of design goals for these overlaid visual links, stating that they should:

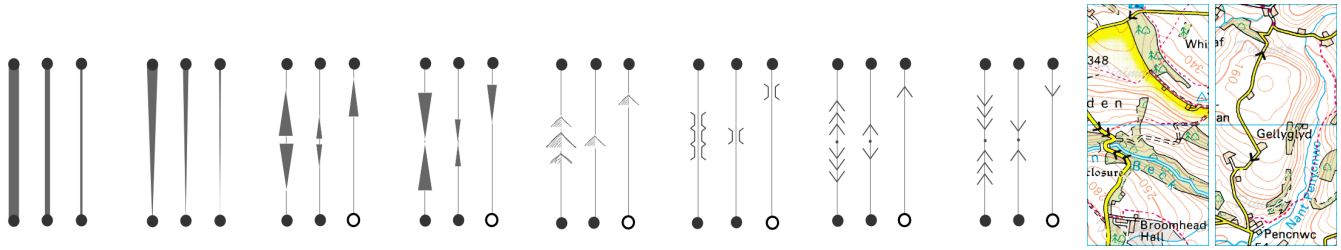

Fig. 8. Visual encoding variations: (a) first two sets are undirected and directed link encodings varying in weight; (b,c,d) the next 6 sets are variations of error glyphs we experimented with for conveying positive and negative errors (i.e. points should be further away or closer apart). Hollow circle indicates a selection. Glyphs in (b) encode error with a continuous scale, (c) with a discrete scale, using the direct metaphor of ridges and bridges in cartography, and (d) is the final encoding we selected to convey error with a discrete scale, inspired from the chevrons used to depict roads with steep gradients in (e) the Ordnance Survey 1:50,000 series.©*Crown Copyright/database right 2014. An Ordnance Survey/EDINA supplied service.*

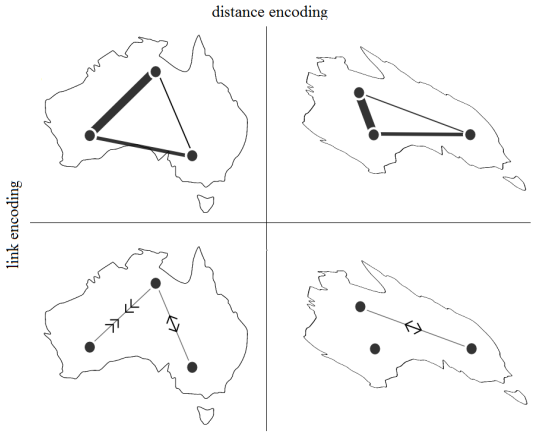

Fig. 7. Images on the left use distance to encode the geography; right images use distance to encode graph information. Top images use visual links to encode weighted graph information; bottom images use glyphs to encode the difference (error) between the distance on the map and actual distances required to accurately convey dissimilarity.

G1 be distinguishable from the background map,
G2 limit clutter of the background map,
G3 encode a weight (magnitude of (dis)similarity),
G4 and possibly encode directionality (e.g. flow as in Section 6.3),

Edges as visual links

We selected a grayscale palette as the background map may use many colors and visual encodings. To minimize clutter we opted for thin lines and pencil-like marks. For directed links we followed the recommendations of the directed graph visualization study [22], opting for a tapered encoding. To convey edge weight, we took inspiration from the glyphs introduced in Spacetree [34], varying the thickness of the non-tapered link ending. Fig. 8a shows the link encodings we selected. While these encodings satisfied our experts, more empirical studies are needed to understand how they may complement or interfere with each other.

Using this additional layer we can encode both geographical and (un)directed weighted graph data.This makes it well suited for the sparse power-grid network which we explore in Section 6.3 as it allows us to show the full network. As graphs become more dense many link crossings occur, making it hard to decode edge weight information or read map data. Clutter could be reduced by filtering links but does one filter out high- or low-weight links? Instead, we found it was preferable to show edges that are furthest from their desired distance and so next we develop a different visual encoding that highlights this *error*.

Edge as distance adjustment glyph

Fig. 8bcd shows glyphs that highlight the relative differences between data dissimilarity and spatial separation in a map. For a given edge this difference (or error) corresponds directly to the magnitude of the stress described in Section 3. Based on a user defined threshold we show glyphs for the most significant errors as showing the complete graph would create too much clutter. For a given threshold, as map deformation reduces stress the glyphs disappear giving useful feedback to the user on the quality of the deformation.

We added a fifth design goal for these glyphs:
G5 Glyphs should be additive to convey regions under a lot of stress (with conflicting edge weights).

We considered several variations to convey the concept of deformation plus overlays while giving the look and feel of error bars. Experiments during case studies with experts led to two major decisions about these variations. First, representing errors in a discrete manner (e.g. employing three bins as in Fig. 8cd instead of continuous scale as in Fig. 8b) improves readability. Second, utilizing variations of symbols that exist in published cartography (e.g. Fig. 8d instead of metaphorical glyphs in Fig. 8c) causes marginally less clutter while still providing an easy-to-remember symbol.

5.2 Deforming the map

Over the centuries, designers have used map deformation as a compelling communication tool for conveying data impact and trends. These deformation techniques rely on the viewer's knowledge of and emotional attachment to the geographical data, as well as the context conveyed by the texture or symbols on the map. Deforming the map to reflect dissimilarity between nodes allows us to represent this information with fewer overlays as described above. However, both geographical knowledge and spatial memory is imperfect in most people, so we find it necessary to modify the design of the map to convey deformation.

Visual design

We experimented with augmenting the deformed map with a *visual reference structure*. We show the result of the deformation of a regular grid as is common in geographical map projections (e.g. [26]). Since grids have a regular structure, they can be used as a reference to understand what was deformed and how (See Figs. 10 to 14).

As the map deforms grid cells are enlarged or shrunk helping viewers gain a coarse understanding of the deformation. To perceive finer deformation (where points where beforehand) we show the *displacement vector*, linking current positions to some previous state – such as the original locations – as a simple black line (See Figs. 9, 10b and 12).

In our analysis in Section 6 we discovered animation to be another important tool in analysis based on map deformation. The iterative nature of our algorithm makes it easy to animate simply by redrawing after every iteration. By smoothly transitioning between true geography and deformation based on data, the viewer can switch back and forth, experiencing the impact of the graph data on the geography. While our algorithm does not ensure the reversibility of each deformation, we achieve this by enabling the viewer to save states of the deformed map during the exploration and allowing smooth transitions between them.

Interaction

Our system offers standard interactions such as selection of nodes and filtering of links. Selection allows the viewer to switch from the general deformed view to one centered on a few selected points. To convey that the current deformations only consider edges from the focus points, we designed a variation of the adjustment glyphs (Fig. 11). Filtering allows the viewer to experiment with the granularity of the displayed adjustments like ignoring those of less than 10%.

Since graph and geographical data may be encoded by different means, our system also provides a configuration panel to experiment with different combination of encodings. Combinations that support *comparisons* are perhaps the most interesting ones. For example, in the UK case study, we relied on animation between deformations to experience the change in data, displacement vectors also quantify these differences (see Fig. 10b). By experimenting with the system our analysis also discovered that link adjustment glyphs are a compelling way to prepare the viewer before applying map deformations (see Fig. 10a). Offering this preview helps guide the viewer's attention to most or least salient deformations.

6 Application Case Studies

In Section 3 we discussed a simple univariate application intended to illustrate the concepts behind our technique. In this section we offer three case studies in which we explore the use of map deformations and our overlay design in real-world applications. The first, in Section 6.1 explores an application where it is common in practice to distort geography to better show the network structure. Our technique is unique in being able to perform this distortion at interactive speeds and without introducing folds in the underlying map (topology violations).

6.1 Travel-Times in the UK

Our initial case study utilises a data set typically used in deforming map examples, as explained in Section 2. In this example we show major locations in Great Britain on a base map of the railway network (Fig. 9a). We deform the map image based on travel time between each of the 35 locations. Using our data set we first use a standard MDS approach. In Fig. 9b the map topology is not preserved; folds appear in the map, stations fall off the edge and the topology around the South and East of the map becomes barely recognisable. Fig. 9c illustrates how our topology preserving method deforms the map based on the data, but also ensures that the topological features are not eroded. In Fig. 9c the degree of displacement of the data points is less compared to Fig. 9b, but it is still clear that the remote towns at Thurso, Pembroke and Penzance are pushed outward and hence are not as well serviced in terms of train-travel time as population centres such as Liverpool (pulled inwards).

6.2 Socioeconomic Data in the UK

Classifying the places where people live into geodemographic groups [19] based on similar multivariate characteristics is popular in the case of the UK census. The *OAC* output area classification [39] for example uses 41 variables to generate a hierarchy of 7 widely-used clusters that subdivide into two higher resolution sub-sets. Geodemographics are often used to inform and support policy, investment and service provision [19].

To ensure the combined data sets produce discriminating profiles, the selection of the variables used and the consequent effects of this choice are important considerations. Suitable selections may vary geographically [33] or according to the domain [36]. The effects associated with the inclusion of individual variables may be either global or geographically focussed. We are working to provide visualization support to assess such effects of geography and scale for geodemographic variable selection [18]. We focus on the energy domain, where consumer profiles may help to understand energy use and perhaps influence behavior. There is an opportunity to use topology preserving map distortion in our work. We offer three reasons.
(1) The degree of similarity between places—which may inform understanding and policy—maps naturally to distance: "*people believe closer things to be more similar than distant things*" [15].
(2) Comparing distorted maps that show similarities determined by different combinations of variables may allow us to evaluate the effects of these combinations synoptically – *in multiple places concurrently*. This approach has potential for comparing the effects of individual variables and their weightings or groups of variables from particular domains.
(3) Comparing such solutions using animated transitions [21] to show how these different combinations of variables 'nudge' places closer together and further apart may emphasize the nature of this change in a manner that is compelling.

In Figs. 10, 11 and 12 we show points representing the ten high-level regions used to aggregate the populations of England and Wales. Fifteen normalised variables (listed in Fig. 10d) are used to determine similarity between regions. Nine – relating to demographics and housing – are from

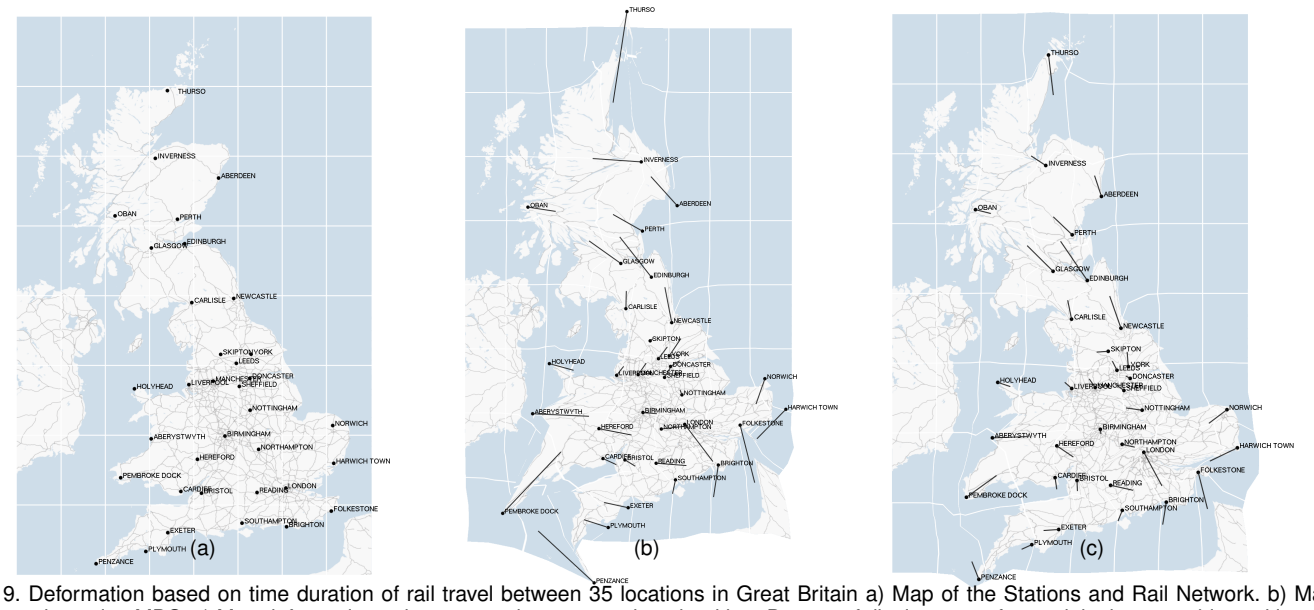

Fig. 9. Deformation based on time duration of rail travel between 35 locations in Great Britain a) Map of the Stations and Rail Network. b) Map deformation using MDS. c) Map deformation using our topology preserving algorithm. Degree of displacement from original geographic positions is shown by the vectors. Data courtesy Matt Griffin.

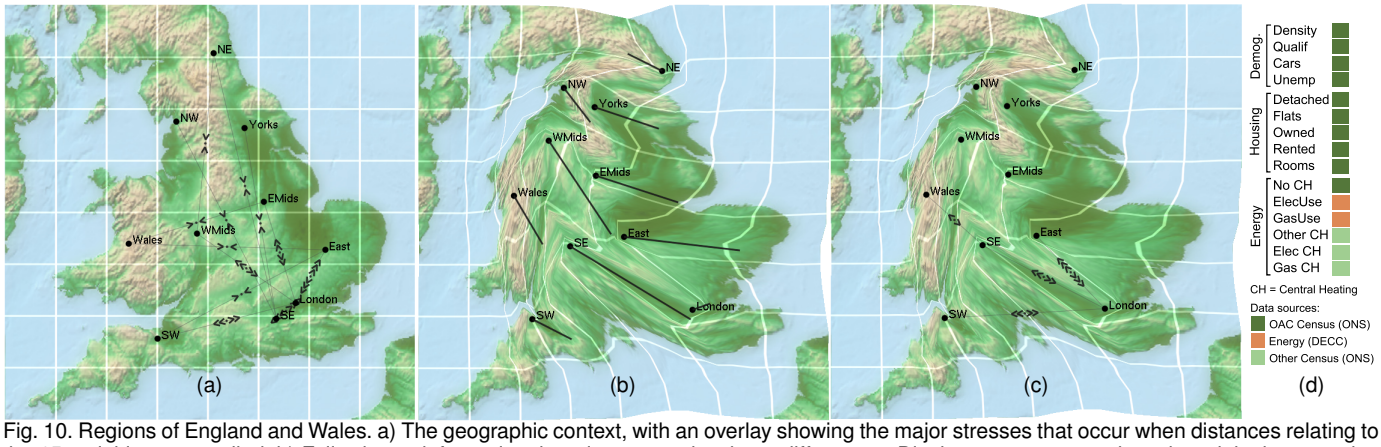

Fig. 10. Regions of England and Wales. a) The geographic context, with an overlay showing the major stresses that occur when distances relating to the 15 variables are applied; b) Following a deformation that aims to resolve these differences. Displacement vectors show the original geography; c) Following the deformation, with an overlay showing the remaining stresses. London should be further from the southern regions; d) Variable weights used to compute dissimilarities in (a)-(c) are equal across all dimensions (CH = Central Heating).

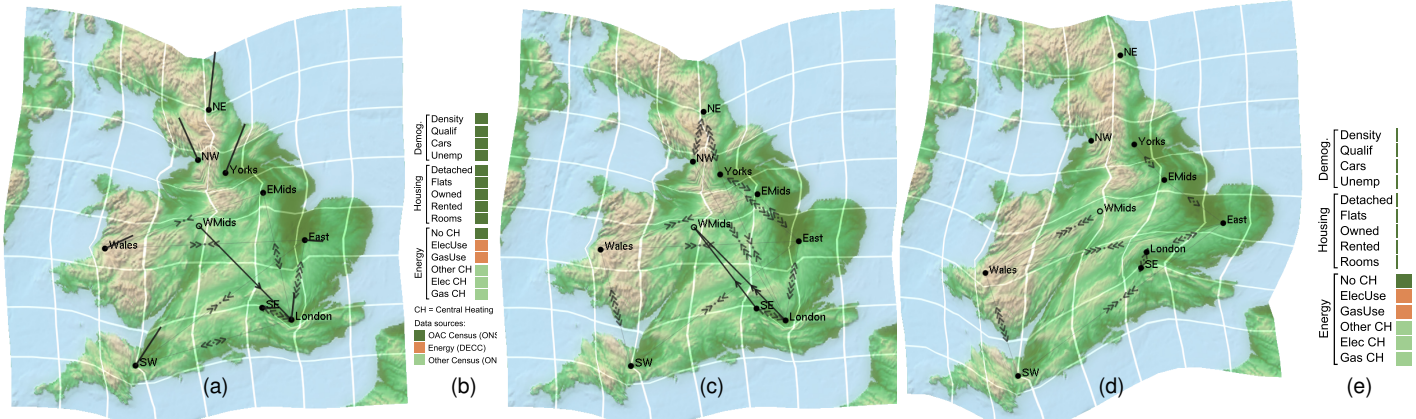

Fig. 11. A local solution in which the deformation shows accurate distances from the West Midlands (*WMids*). a) For all 15 (equal weighted) variables (as per b), with displacement vectors relating locations to the original geography and an overlay highlighting major *local* remaining stresses. Those relating to the West Midlands (focussed) are in a dark shade, showing that London is located closer than it should be given the dissimilarity distance; c) With deformation as per a, but with an overlay showing stresses associated with a *local* solution that only considers the 6 energy variables (with dimension weights as shown in e). London and *SE* are not as distant from *WMids* in terms of the energy variables; c) After deformation for the 6 energy variables, as per e, with an overlay revealing that significant *global* stresses remain.

the 2011 UK Census of Population [32] and are variables in OAC [14]. Six relate specifically to energy usage – four from the Census, related to forms of central heating, and two measure gas and electricity consumption as provided by the UK Department of Energy & Climate Change (DECC) [9].

Fig. 10a shows a regular geographic map, projected to the British National Grid, which ensures that distance measurements are accurate across the area shown. We use our error glyph overlays (Section 5.1) to represent the differences between the distances on the map and a dissimilarity measure calculated using the 15 attributes – a weighted 15 dimensional euclidean distance. Other possibilities exist, but we found this simple measure worked well in conjunction with a "mixing panel" of sliders to control the weights on each dimension. The figures in this section include charts showing the relative weights of the various dimensions.

In Fig. 10a dissimilarities are shown using overlays only. In Fig. 10b we also deform the map, using displacement vectors (Section 5.2) to highlight the movement of each region from its original position. This shows the considerable geodemographic differences between London and the other regions in a dramatic fashion. These differences are even more significant than Fig. 10b suggests, as is apparent when the stresses that remain after computing a solution are added in a glyph overlay. Fig. 10c shows that London should be even further from its neighbours in a perfect solution.

Helping energy consumers consider their characteristics in relation to others locally has proven to be effective in changing behaviour [2], [4]. The deformation algorithm resolves very successfully to a local solution whereby stresses are only applied between a single location and all others. This enables us to produce a local map in which the distances between any selected point and all others relate very closely to attribute dissimilarity. A local solution for the *West Midlands* (Fig. 11a) shows this region to be more like its northern neighbours and less like its southern neighbours than true geographic distances might suggest. Once again our deformed map shows this dramatically.

We explore the effects of the variables used in calculating dissimilarities in this local case. We recompute dissimilarities relative to the West Midlands in Fig. 11b, but now only the 6 energy variables are considered. The resulting stresses are shown in the overlay, with those relating to the West Midlands emphasized in a dark shade. It is clear that there are differences between the solutions according to the variables used. These are likely to be especially acute when variables are selected or deselected according to a particular domain as is the case here. But these differences also appear to be geographic, with regions to the South East being 'closer' to the West Midlands in terms of their energy attributes than their other characteristics, while most of the other regions are more dissimilar than is the case with the 15 variable solution. London and *SE* 'nudge' towards the West Midlands when these characteristics are the focus of the comparison. The local, energy-based, deformation is implemented in Fig. 11c. The distances between the focal location – *WMids* – and all others are relatively accurate, but this accuracy comes at the expense of a poor global solution. The glyphs in the overlay show significant stresses between locations other than the West Midlands.

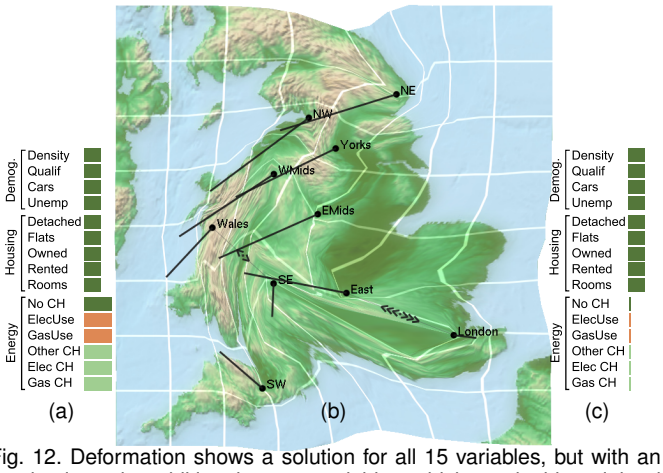

Fig. 12. Deformation shows a solution for all 15 variables, but with an emphasis on the additional energy variables, which are doubly weighted as per (a). Remaining stresses are shown with glyphs. Displacement vectors show the locations resulting from a deformation when energy variables are not considered in calculating distances, as per the weighting shown in (c).

Producing a map with such strong distortion as that shown in Fig. 11 is somewhat questionable. But the visual effect achieved may be useful in helping individuals with a focus on one place – *their* location – consider its attribute relationships with other places *concurrently* and *geographically* as they move closer together and further apart. This is not possible with more abstract and less spatial graphics. This idea of places moving in relation to their dissimilarities with one another gives rise to a metaphor with *behaviour nudging*, which has been shown to occur in response to visual stimuli [27] and is being explored as a means of effecting behaviour change in the energy domain [2], [4], [7].

We can explore the effects of using different variables and changing the weights applied to them through the deformation. For example, Fig. 12 shows a globally deformed map in which the six energy variables have twice the weight of the housing and population characteristics in determining the dissimilarities. The displacement vectors show how these locations differ from an alternative solution, one that uses the nine other variables in this case. Here we see deep differences between *London* and the other regions in the energy-based solution to which we deform the map. But, through their dispersion from locations associated with the nine variable solution, the displacement vectors show that when energy data are emphasised in multivariate analyses at this resolution better differentiation between regions may be achieved. This finding supports our efforts to develop energy based geodemographic profiles that are nationally discriminating.

We consider the global and local deformations presented here sufficiently persuasive to demonstrate the communicative properties and affect of the deformations and subsequent 'visual nudging' through interaction and animated transition. The global deformation could also be applied at a neighbourhood level for nudging small numbers of locations closer together and further apart under different conditions (as shown in Fig. 12). This would help us determine the influence of these variables over differing spatial scales. In the behavioural context 'local nudging', in which

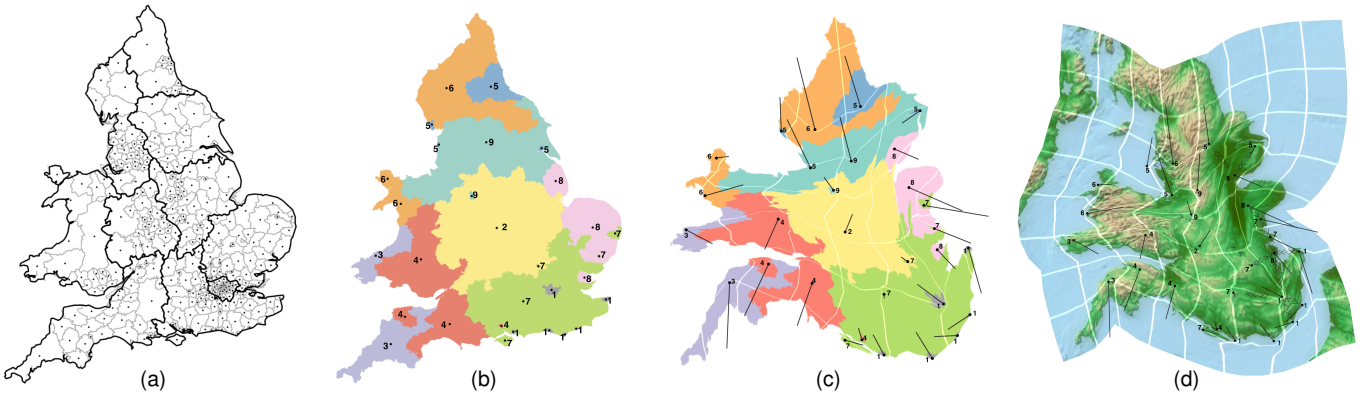

Fig. 13. Scaling example using a) 348 Local Authority Units associated with the 10 NUTS regions of England and Wales, b) spatial clustering results in 9 clusters (labelled 1-9) as 29 regions, c) deformation is based on the mean value for the 15 variables for each of the 29 regions with the cluster base map and d) again with the topological map, where remaining stresses are shown with glyphs. Displacement vectors show the locations resulting from a deformation in (c) and (d).

energy consumers are compared to their neighbours, is also known to be particularly effective in reducing energy usage in high-consuming households [2].

6.2.1 Scalable and more accurate dissimilarity deformation by clustering over data as well as geography

Geodemographic profiles are usually developed at high resolutions (neighbourhood rather than region). In Figs. 9–11 the control points for our map deformation were the centres of the 10 "Territorial Units" (NUTS) of England and Wales mandated by the European Office for Statistics [31]. The boundaries of these regions can be rather arbitrary with respect to the underlying pair-wise similarities of a given dataset. To get a more accurate picture of the underlying data one could attempt a deformation using control points at the finest level of granularity, but in practice—as discussed in Section 4—one runs into issues of both algorithmic complexity for the deformation, and severe clutter with respect to the overlays. Instead, we can derive a different clustering of regions based on both data similarity and geographic distance. By computing such a clustering as a preprocessing step, we can apply our map deformation and overlays efficiently and clearly to *any number* of input data points.

We demonstrate this idea for recovering such local detail in Fig. 13. Instead of the 10 NUTS regions (bold borders) we use the 348 local authority units (LAU) as shown in Fig. 13a. For the dissimilarity matrix D of the 15 data variables and the matrix E of the squared euclidean distances between each LAU centroid, we apply k-means clustering to $D \circ E$. We chose $k = 9$ such that the 348 data points are reduced to 9 clusters. Where these clusters are geographically non-contiguous we split them again leaving 29 distinct regions whose geographic centres become control points for our deformation (see points and cluster labels in Fig. 13b). The ideal-lengths of the edges between these points are recomputed from the underlying data of their constituent LAU.

The subsequent deformation (Figs. 13c and d) shows that where data points are within the same cluster they are pulled together, whilst different clusters are pushed apart. Comparing Fig. 10c (equal weighted) to Figs. 13c and d, we see that Central London, for instance, is in the same cluster (1) as five smaller areas on the south and south-east coastline, all are densely populated regions. These areas are pulled together instead of all regions being pushed away from London as noted previously from Fig. 10c, indicating a truer extent of the homogeneous socio-economic region around London. Some similarities are also noted, for example Cornwall to the southwest and the west of Wales are combined in the same cluster (3) and are pulled together as *Wales* and *SW* are slightly in Fig. 10c. The difference in Figs. 13c and d is that *Wales* split into three distinct clusters (3, 4 and 6) relating to the underlying data and each are morphed in different directions. Thus, more detail of the underlying data dissimilarity is revealed in deformation based on this data-derived clustering compared to deformation using the arbitrary NUTS region centres as control points.

6.3 Power Grid Data in Australia

Power systems are often referred to as the most complex human-engineered systems ever built. A power-grid is a meshed network of transmission lines connecting a relatively small number of power stations with a large number of consumer loads. Because electricity is not (or—to date—has not been) economically storable in substantial amounts, the demand from consumers must be met instantaneously every moment in time. The economics of the supply side of electricity results in power systems being traditionally structured with a small number of very large power generators built close to resources (typically lakes on large rivers or fossil fuel sources such as coal mines and gas fields).

In Fig. 14 we use deforming maps to show the dynamic behavior of a power-grid (conforming to an IEEE 30-bus standard test system) modelling a hypothetical scenario in-which conventional power-sources are augmented by renewables. In this scenario engineers need to study the dynamics of the system, with particular attention to parts of the network at or near capacity at different times of the day. Our power network is a graph with nodes corresponding to generators and consumption points and edges corresponding to physical transmission lines between those nodes. The graph with its undeformed geographic embedding is shown in Fig. 14a. Our intention is to deform this map to better

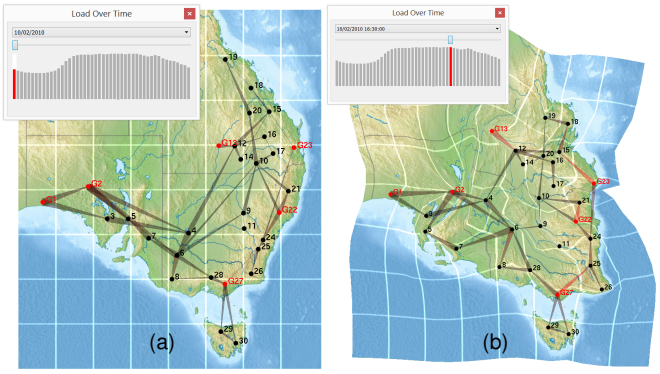

Fig. 14. Power-grid network in its geographic context: (a) undistorted for the baseline load at midnight and (b) deformed to highlight peak-load. Generator nodes are red. Tapering of edges indicates direction of flow. Thickness of the edge indicates total flow on the edge. Edges are coloured red when they are close to capacity.

show the network structure and relative line flows at each point in time. We also need to be able to dynamically morph the map to transition and compare between samples. To achieve this aim we set data edges in our two part MDS stress model as follows. Data edges corresponding directly to power-grid network edges have their ideal distance set proportionally to load. Data edges for which there is no source network edge have their ideal distances set according to the sum of loads on edges across the shortest paths between their source and target. This follows the Gansner *et al.* [16] model for adapting stress-minimization to achieve clear graph layout.

To allow engineers to study the dynamic behavior of the system we allow them to interactively switch between samples and hence between different sets of ideal distances among nodes. Our deformation method is fast enough to morph the map in real time as the engineer uses a slider to "scrub" through the data at all time samples. Fig. 14 shows this slider being manipulated to compare loads between midnight and midday. A histogram is coupled to the slider to show total network load over time.

In our initial design for this visualization we followed the MDS practice of trying to place highly-linked (more-similar) data points more closely. That is, we set desired edge-distance to be proportional to the inverse load so that, as load increased across certain edges, those edges would become shorter. Hence, geographic regions with heavily loaded connections would be drawn together. When presenting this design to power engineers we found that it was contrary to their needs and expectations. Rather, they wanted problem areas of the map (those where load is high) to be enlarged both to attract more attention and to support more detailed examination.

We can see this mapping of load to edge-length taking effect in the figures. In particular, at midday (Fig. 14b) the load is significantly higher than at midnight (Fig. 14a) and parts of the map are distinctly larger. Originally, our map was always distorted to show absolute load. In our interviews with Engineers, we realised that subtle changes from one highly distorted figure to another were easily missed. Thus, our final design for the system uses a baseline load for the undistorted map, for example in Fig. 14a midnight is taken as the baseline. Then, distortion is based on difference deltas from this baseline. Fig. 14b shows the maximally distorted view when the load is peak at 4:30pm. The baseline can be reset to any time, thus allowing for small changes to be examined in detail.

In our system (see accompanying video) the deformation is computed continuously in a background thread leaving the UI (the slider) responsive at all times. We render each iteration of the deformation to provide smooth animated transitions, typically in excess of 30 frames per second. When the slider is not manipulated the layout converges in about half a second (see timings in Section 4). Thus, the technique is responsive enough to be used to monitor data live as it is received during operation of a real power-grid system.

7 Limitations

It is difficult to say that the method scales to a particular number of data points. The examples presented here, although they have limited numbers of data points, are embedded in a mesh subdivision of the map with many hundreds of vertices as seen in Fig. 3a. In principal, we could have data encoded at each of these vertices. In Section 6.2.1 we demonstrated a clustering over the combined data and geographical spaces that allows for deformation showing high-level dissimilarities between any number of underlying data points.

A much harder problem to deal with than making the algorithm scale to large numbers of data-points—and a much more pragmatic limitation to scalability—are complex dissimilarities between those points. That is, when dissimilarities vary greatly and confoundingly from geographic distance, similar points may only be brought close together through a complex folding of the map. As illustrated in Figs. 3b and 3c, any subsequent deformation of the map is a compromise between faithfulness to the data and faithfulness to geography while maintaining readability.

As with any visualisation technique, an alternative to visualising every single data point when doing so may lead to too much clutter or visual complexity, is to instead aggregate the data. In our UK case study of Section 6.2 the underlying dataset included 15-dimensional data of 348 geographic regions.

8 Conclusion

We have presented a fast algorithm for deforming a geographic map such that distances more closely correspond to underlying data distances. We also presented case-studies applying this technique to study multivariate data-dissimilarities and to show a weighted graph structure. We believe that this is the first exploration of visualizing multivariate dissimilarity data between many geographic points through map deformation. Previous work on map deformation (see Section 2) has only looked at single-variate data (cartograms); deformation to show dissimilarity with respect to a single point (a much simpler problem); or to show dissimilarities that are less decoupled from geography, for example, travel times or metro maps, which was done with diffusion based methods much slower than ours.

In exploring these applications we came to realize that when the data topology is very different from its geography, there is a severe trade-off in terms of how far you can distort the map before it becomes unrecognisable. As a result, the visualisations presented here double-encode dissimilarity with visual overlays next to the map deformation.

9 Future Work

It is the complexity of dissimilarity analysis that makes the interactive performance of our method particularly important. In the analysis tasks described in this paper, this interaction was crucial in finding interesting geographical anomalies. There are many more possibilities for direct interaction with the mesh that defines geographic stress, for example the user could be allowed to create tears in the mesh to free the map to "open up" in ways that permit a much lower data stress solution. Another idea is to make the glyphs themselves interactively interrogable to allow, for example, the user to drill into details about the error it represents.

Currently our system only shows a single map view at a time. Much of the analyses described in our case studies involved transitioning between various stored states in order to visually compare them. Another option would be to use a small multiples display to support static comparison, just as we have used side-by-side figures to support comparison in this paper.

We explored applications that attempted serious analysis using map deformation, but perhaps another area of future exploration should be communication. That is, using these kinds of techniques to tell the story of data's place in the world. There are many other areas of further work associated with the underlying technique. For example, investigation of different mesh construction including hexagonal meshes or meshes aware of underlying geography such as coastlines, rivers or county lines. Also, there are many optimizations that are possible for the algorithm such as multilevel schemes to decrease algorithmic complexity.

References

[1] N. Ahmed and H. Miller. Time-space transformations of geographic space for exploring, analyzing and visualizing transportation systems. *Journal of Transport Geography*, 15:2–17, 2007.

[2] H. Allcott. Social norms and energy conservation. *Journal of Public Economics*, 95(9-10):1082–1095, 2011.

[3] S. Bies and M. J. van Kreveld. Time-space maps from triangulations. In *Proc. 20th International Symposium on Graph Drawing*, LNCS 7704, pages 511–516, 2012.

[4] J. Bird and Y. Rogers. The pulse of tidy street: Measuring and publicly displaying domestic electricity consumption. In *Workshop on Energy Awareness and Conservation through Pervasive Applications*, 2010.

[5] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[6] J. Böttger, U. Brandes, O. Deussen, and H. Ziezold. Map warping for the annotation of metro maps. *IEEE Computer Graphics and Applications*, 28(5):56–65, 2008.

[7] A. Boucher, D. Cameron, and N. Jarvis. Power to the people: dynamic energy management through communal cooperation. In *Proc. Designing Interactive Systems Conference*, pages 612–620. ACM, 2012.

[8] S. Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[9] DECC. Sub-National Electricity and Gas Consumption Statistics - http://bit.ly/1bCqsb9, Accessed October, 2015.

[10] D. Dorling, A. Barford, and M. Newman. Worldmapper: the world as you've never seen it before. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):757–764, 2006.

[11] T. Dwyer. Scalable, versatile and simple constrained graph layout. *Computer Graphics Forum*, 28(3):991–998, 2009.

[12] T. Dwyer, K. Marriott, and M. Wybrow. Topology preserving constrained graph layout. In *Proc. 17th International Symposium on Graph Drawing*, LNCS 5849, pages 230–241, 2009.

[13] T. Dwyer and G. Robertson. Layout with circular and other non-linear constraints using procrustes projection. In *Proc. 18th International Symposium on Graph Drawing*, LNCS 6502, pages 393–404, 2010.

[14] ESRC: Retail Research Data. Preliminary 2011 OAC - http://bit.ly/1pgSEE4, Accessed October, 2015.

[15] S. I. Fabrikant, M. Ruocco, R. Middleton, D. R. Montello, and C. Jörgensen. The first law of cognitive geography: Distance and similarity in semantic space. *Proc. 2nd International Conference on Geopgraphic Information Science*, pages 31–33, 2002.

[16] E. R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *Proc. 13th International Symposium on Graph Drawing*, LNCS 3843, pages 239–250. Springer, 2005.

[17] M. T. Gastner and M. E. Newman. Diffusion-based method for producing density-equalizing maps. *Proc. National Academy of Sciences of the United States of America*, 101(20):7499–7504, 2004.

[18] S. Goodwin, J. Dykes, A. Slingsby, and C. Turkay. Visualizing multiple variables across scale and geography. *(Accepted for) IEEE Transactions on Visualization and Computer Graphics (Proceedings of the Information Visualization 2015)*, 23(01), January 2016.

[19] R. Harris, P. Sleight, and R. Webber. *Geodemographics, GIS and neighbourhood targeting*, volume 7. John Wiley and Sons, 2005.

[20] J.-H. Haunert and L. Sering. Drawing road networks with focus regions. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2555–2562, 2011.

[21] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[22] D. Holten and J. J. van Wijk. A user study on visualizing directed edges in graphs. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 2299–2308. ACM, 2009.

[23] S. R. Hong, Y. Kim, J.-C. Yoon, and C. R. Aragon. Traffigram: Distortion for clarification via isochronal cartography. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 2014.

[24] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proc. 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 19–26. ACM, 1993.

[25] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *Proc. 32nd ACM SIGGRAPH*, 24(3):1134–1141, 2005.

[26] B. Jenny. Adaptive composite map projections. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2575–2582, 2012.

[27] V. Kalnikaite, Y. Rogers, J. Bird, N. Villar, K. Bachour, S. Payne, P. M. Todd, J. Schöning, A. Krüger, and S. Kreitmayer. How to nudge in situ: designing lambent devices to deliver salient information in supermarkets. In *Proc. 13th International Conference on Ubiquitous Computing*, pages 11–20, 2011.

[28] D. A. Keim, S. C. North, and C. Panse. Cartodraw: A fast algorithm for generating contiguous cartograms. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):95–110, 2004.

[29] S. Lin, C. Lin, Y. Hu, and T. Lee. Drawing road networks with mental maps. *IEEE Transactions on Visualization and Computer Graphics*, 20(9):1241–1252, 2014.

[30] Y. Lipman. Bounded distortion mapping spaces for triangular meshes. *ACM Trans. Graph.*, 31(4):108:1–108:13, July 2012.

[31] Office for National Statistics. Nomenclature of territorial units for statistics (nuts) / local administrative units (lau) - http://bit.ly/1zryj1u, Accessed July, 2015.

[32] Office for National Statistics. 2011 Census ONS - http://bit.ly/1dBdIUc, Accessed October, 2015.

[33] J. Petersen, M. Gibin, P. Longley, P. Mateos, P. Atkinson, and D. Ashby. Geodemographics as a tool for targeting neighbourhoods in public health campaigns. *Journal of Geographical Systems*, 13(2):173–192, June 2011.

[34] C. Plaisant, J. Grosjean, and B. B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 57–64. IEEE, 2002.

[35] E. Simizu and R. Inoue. A new algorithm for distance cartogram construction. *Geographical Information Science*, 23(11):1453–1470, 2009.
[36] A. D. Singleton and P. A. Longley. Creating open source geodemographics: Refining a national classification of census output areas for applications in higher education. *Papers in Regional Science*, 88(3):643–666, 2009.
[37] S. Sun. An optimized rubber-sheet algorithm for continuous area cartograms. *The Professional Geographer*, 65(1), 2012.
[38] W. R. Tobler. Thirty five years of computer cartograms. *ANNALS of the Association of American Geographers*, 94(1):58–73, 2004.
[39] D. Vickers and P. Rees. Creating the uk national statistics 2001 output area classification. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):379–403, 2007.
[40] Y. Weng, W. Xu, Y. Wu, K. Zhou, and B. Guo. 2d shape deformation using nonlinear least squares optimization. *The Visual Computer*, 22(9-11):653–660, 2006.



Quirijn Bouts received his MSc from Einhoven University of Technology in the Netherlands in 2013. He has since joined the Applied Geometric Algorithms group in Eindhoven where he is now working towards his PhD. His research interests include computational geometry, visualization and graph drawing.



Tim Dwyer is a Larkins Fellow and Senior Lecturer with the Faculty of IT at Monash University, Australia. He co-directs the Immersive Analytics Initiative and is a founding member of the Monash Adaptive Visualisation Lab. Since receiving his PhD from the University of Sydney he has been a Research Fellow at Monash University and Microsoft Research as well as an engineer with the Visual Studio product group at Microsoft, Redmond USA.



Jason Dykes is a Professor of Visualization in the Department of Computer Science at City University London. Co-Director of the giCentre, he uses techniques from Cartography, Information Visualization, Human Computer Interaction, Computer Science and GIScience to develop novel maps that help generate insights from data and communicate trends.



Bettina Speckmann is a full professor at the Department of Mathematics and Computer Science at the Eindhoven University of Technology (the Netherlands). Her research interests include the design and analysis of algorithms and data structures, discrete and computational geometry, applications of computational geometry to geographic information systems, graph drawing, and geo-visualization.



Sarah Goodwin received her PhD in Geographical Information Science from City University London in 2015. Her research investigates visual techniques for energy analysts and variable selection for geodemographic profiling. Her professional and academic background is in Geography, Spatial Analysis and Geovisualisation. She is currently a post-doctoral Research Fellow at Monash University with the Monash Adaptive Visualisation Lab.



Nathalie Henry Riche is a researcher at Microsoft Research since December 2008. Her interests lie in the visual exploration of graphs and networks, visualization of groups, interactive graph navigation techniques and evaluation methods for information visualization.



Sheelagh Carpendale is a Professor at the University of Calgary where she holds a Canada Research Chair in Information Visualization and an NSERC/AITF/SMART Industrial Research Chair in Interactive Technologies. She directs the Innovations in Visualization (InnoVis) research group and initiated the interdisciplinary graduate program, Computational Media Design. She has a dual background in Computer Science (Ph.D. Simon Fraser University) and Visual Arts (Sheridan College, School of Design and Emily Carr, College of Art).



Ariel Liebman is an Adjunct Senior Research Fellow with the Faculty of Information Technology in this role he is coordinating Monash University's development of an energy market modelling and analysis capability.