

# Cartograms with Topological Features

Chao Tong<sup>1</sup>, Liam McNabb<sup>1</sup>, and Robert S. Laramée<sup>1</sup>

<sup>1</sup> Visual and Interactive Computing Group, Swansea University

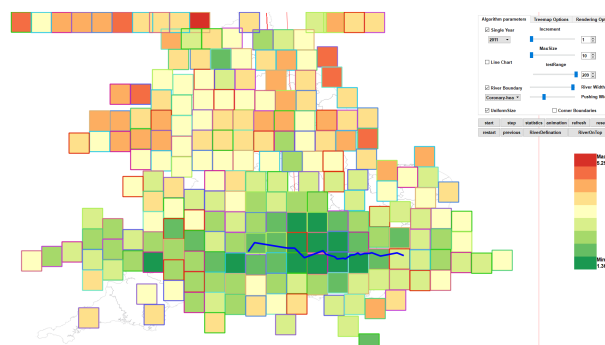
## Abstract

Cartograms are very popular and useful for depicting data on a map. Dorling style and rectangular cartograms are very good for facilitating comparisons between unit areas. Each unit area is represented by the same shape such as a circle or rectangle, and the uniformity in shapes facilitates comparative judgment. However, the layout of these more abstract shapes may also simultaneously reduce the map's legibility and increase error. When we integrate univariate data into a cartogram, the recognizability of cartogram may be reduced. There is a trade-off between information recognition and geo-information accuracy. This is the inspiration behind the work we present. We thus attempt to increase the map's recognizability and reduce error by introducing topological features into the cartographic map. Our goal is to include topological geographic features such as a river in a Dorling-style or rectangular cartogram to make the visual layout more recognizable, increase map cognition and reduce geo-spatial error. We believe that compared to the standard Dorling and rectangular style cartogram, adding topological features provides familiar geo-spatial cues and flexibility to enhance the recognizability of a cartogram.

## 1. Introduction and Motivation

Cartograms are very popular and useful for depicting data on a map. A cartogram can be defined as, "a technique for displaying geographic information by resizing a map's regions according to a statistical parameter in a way that still preserves the map's recognizability." [CCM15] According to Nusrat and Kobourov [NK16], cartograms can be categorized into four types: contiguous, non-contiguous, Dorling and rectangular. Dorling style and rectangular cartograms are very good for facilitating comparisons between unit areas. Each unit area is represented by the same shape such as a circle or rectangle, and the uniformity in shapes facilitates comparative judgment. However, the layout of these more abstract shapes may also simultaneously reduce the map's legibility and increase error. When we integrate univariate data into a cartogram, the recognizability of cartogram may be reduced. There is a trade-off between information recognition and geo-information accuracy. This is the inspiration behind the work we present. We thus attempt to increase the map's recognizability and reduce error by introducing topological features into the cartographic map.

In this paper, we use the term topology slightly different from the traditional sense as with graphs and standard cartograms. In the context of graphs and cartograms, topology normally refers to the unit areas as nodes and the edge connections between them as their topology. In this work, we adapt the notion of topology from the flow visualization literature [LHZP07]. In flow topology, vector fields are divided up into different regions of flow behavior e.g.



**Figure 1:** A cartogram with the Thames river, featuring a wide push width,  $p_e$  and a river width,  $r_e$  of 10 pixels. Color is mapped to Coronary heart disease distribution in England.

rotating flow versus linear flow. The edges or curves that separate and connect the different regions of the flow are referred to as the flow's topology. Hence, we introduce a (flow-inspired) topological feature into Dorling style and rectangular cartograms, i.e., a river. The river separates the unit areas into distinct regions and makes the cartogram more legible and reduces node layout error. This is analogous to a separatrix in the flow visualization literature [PVH\*03], a standard feature in flow topology. A separatrix is a special type

of streamline that connects two critical points and a curve that no flow crosses.

Our goal is to include topological geographic features such as a river in a Dorling-style or rectangular cartogram to make the visual layout more recognizable, increase map cognition and reduce geo-spatial error. We believe that compared to the standard Dorling and rectangular style cartogram, adding topological features provides familiar geo-spatial cues and flexibility to enhance the recognizability of a cartogram. For example, the Thames river can be converted to topological feature on a cartogram. The regions on each side of the river on the real-world map are also often divided by this topological feature on a traditional map. This helps the user to locate corresponding region groups near the Thames river easily.

Our contributions include:

1. A new way to define and add real geographic features, such as a river as a topological feature to a cartogram.
2. A novel cartographic layout algorithm that preserves a nearby region node's topological location with respect to a river.
3. The application of this cartogram design to real-world health-care data provided by the NHS, England.

The rest of the paper is organized as follows. Section 2 presents previous related work on cartograms with a special focus on Dorling and rectangular cartograms. Section 3 presents a method for river definition and describes the layout algorithm and pseudo-code for cartogram generation. Section 4 presents the results and comparison with different traditional cartograms.

## 2. Related Work

Related work falls into the areas of Dorling and rectangular cartograms. Cartograms can display geo-spatial information and another data attributes (such as population or disease prevalence) in one image. Some very helpful survey papers provide an overview of geo-spatial visualization e.g. by McNabb and Laramée and Tong et al. [ML17, TRR\*18]. Tobler [Tob04] and Nursat and Kobourov [NK16] survey cartograms. They present the development of value-by-area cartogram algorithms and comparing their performance in visualization.

Auber et al. [AHL\*11] propose a layout method based on a geographic map metaphor, which facilitates the visualization and navigation of a hierarchy and preserves the order of the hierarchy's nodes.

Gastner and Newman [GN04] present a diffusion cartogram for constructing value-by-area cartograms, which provides a valuable tool for the presentation and analysis of geographic data. Keim et al. [KNP04] develop a faster algorithm for cartograms. It enables display dynamic data with cartogram visualizations. These two algorithms are categorised as contiguous area cartograms. Their performance depends on the corresponding value in each area. If the given data value does not correspond to the area, the cartogram may be difficult to recognize.

Raisz [Rai34] presents the rectangular cartogram, using rectangles instead of real area shapes. Dorling [Dor11] presents the Dorling cartogram which uses circles instead of geographic area shape,

similar to the modified cartogram we present. They are categorized as non-continuous area cartograms. They can display statistical information well, regardless of original shape of area, and preserve relative position. Van Kreveld and Speckmann [vKS07] present the first algorithm for rectangular cartograms. They formalize region adjacencies in order to generate processable layouts that represent the positions of the geographic regions. They convert a rectangular cartogram to a contiguous area cartogram. Our modified cartogram does not fall into the category of continuous cartograms but resembles a cross between rectangular and Dorling cartograms [NK16].

Heilman et al. [HKPS04] propose a novel visualization technique for geo-spatial datasets that approximates a rectangular partition of the rectangular display area into a number of map regions preserving important geo-spatial constraints. They use elongated rectangles to fill the space whereas we use uniform rectangles to fill the space such that regions can easily be compared with one another.

Panse et al. [PSKN06] combine a cartogram-based layout (global shape) with PixelMaps (local placement), obtaining benefits of both for improved exploration of dense geo-spatial data sets.

Slingsby et al. [SDW09] explore the effects of selecting alternative layouts in hierarchical displays that demonstrate multiple aspects of large multivariate data sets, including spatial and temporal characteristics. They demonstrate how layouts can be related, through animated transitions, to reduce the cognitive load associated with their reconfiguration whilst supporting the exploratory process. No metric for neighborhood preservation is described in this work.

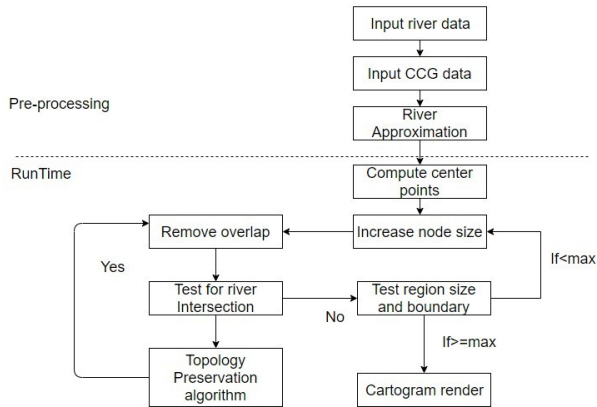
Slingsby et al. [SDW10] present rectangular hierarchical cartograms for mapping socio-economic data. They present a detailed map of 1.52 million UK unit postcodes in their spatial hierarchy, sized by population and coloured by the OAC category that most closely characterizes the population. However, no algorithm for preserving geo-spatial information is provided.

Alam et al. [AKV\*15] present a set of seven quantitative measures (Average Cartographic Error, Maximum Cartographic Error, Adjacency Error, Angular Orientation Error, Hamming Distance, Average Aspect Ratio, Polygonal Complexity) to evaluate performance of cartograms based on the accuracy of data and its readability. They compare previous cartogram algorithms based on statistical distortion, geography distortion and algorithm complexity and evaluate their performance with respect to different properties.

Eppstein et al. [EvKSS15] introduce a new approach to solve the association challenge for grid maps by formulating it as a point set matching problem. They present algorithms to compute such matchings and perform an experimental comparison that also includes a previous method to compute a grid map.

Meulemans et al. [MDS\*17] design a comprehensive suite of metrics that capture properties of the layout used to arrange the small multiples for comparison (e.g. compactness and alignment) and the preservation of the original data (e.g. distance, topology and shape). Their work focuses on geo-spatial information and neighborhood preservation.

Tong et al. [TML\*17, TRL\*17] develop a Cartographic treemap



**Figure 2:** The processing pipeline for producing a cartogram with topological features.

to integrate a modified representation of the UK based on the geo-spatial information of CCG (Clinical commissioning group) regions combined with a modified treemap to present multivariate NHS data. They also present a metric to analyze the trade-off between space-filled and geo-spatial accuracy. To the best of our knowledge, the work we present here is the first of its kind to introduce topological features to Dorling-style and rectangular cartograms.

### 3. Adding Topological Features to Cartograms

This section describes the cartogram construction algorithm starting with an overview. The processing begins with reading the UK geo-spatial information. The algorithm summary is as follows:

1. Acquire and input selected river data.
2. Input geo-spatial data for cartogram generation.
3. Define a river approximation and add it to the geo-spatial data set that the cartogram is based on.
4. Compute unit area (or region) center points: We use the QGIS [QGI] tool to calculate the center points of each CCG (Clinical Commissioning Group) region. The center points are the starting positions of the rectangular region nodes.
5. Update node size: We start with a unit square to represent each CCG region as a node in the cartogram and gradually increase the size of each node.
6. Update cartographic layout: During the region growing process, regions may not cross topological features.
7. Test for river intersection. When a region intersects a topological feature. The layout algorithm returns this region to its previous position.
8. Remove overlap between nodes.
9. Test boundary conditions.
10. Render the cartogram with features.

Figure 2 shows an overview of the algorithm pipeline.

#### 3.1. Input River Data

We search for and obtain the Thames river geo-spatial information from OpenStreetMap [Ope] for river shape, name and type. Then we use Overpass Turbo API [API] to build a query based on the search result and export the Thames river as a shapefile. We then input the Thames shapefile into QGIS [QGI] described in section 3.2 and combine it with a CCG map, and proceed to river approximation. The geo-spatial information is from a separate data source than our cartogram data for this special application which involves population healthcare prevalence values.

#### 3.2. Input CCG Data

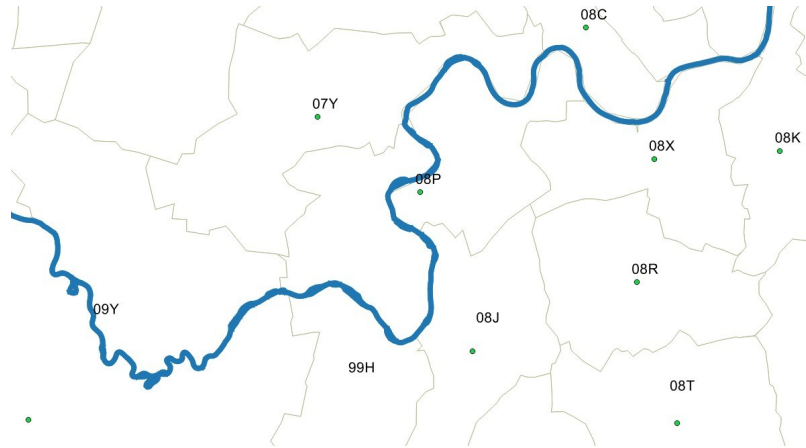
We study open NHS healthcare population data as a case study for our topology-based cartograms. The UK government collects yearly diagnoses of region-specific population healthcare data. The public health profiles website [NHSa] is used for publishing the latest national healthcare data in England, UK. The data archive is designed to support GPs, clinical commissioning groups (CCGs), and local authorities to ensure that they provide and commission effective and appropriate healthcare services. Typically this data is displayed using standard line charts, bar charts and pie charts. The standard visualizations do not usually feature any geo-spatial information. The dataset consists of 14 Excel files of around 10 Megabytes in total together with a CCG region map containing 209 regions. There are more than 60,000 rows and an average of 100 columns in each file.

Our goal is to visualize this CCG data with cartograms, and make it more recognizable and reduce layout error by adding topological features. We use the NHS healthcare data and the NHS healthcare regions map as input.

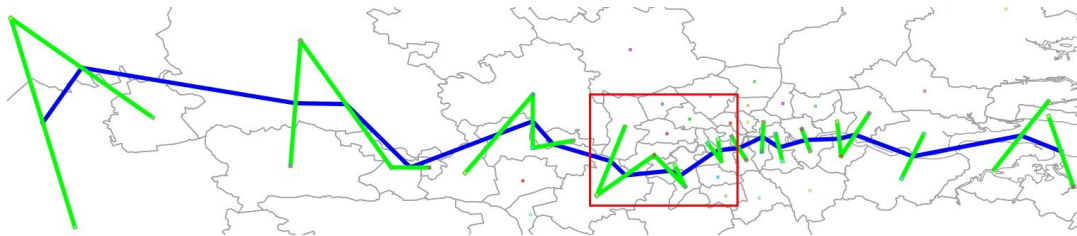
#### 3.3. River Definition and Approximation

Our goal is to include a topological geographic feature such as a river in a dorling-style cartogram to make the visual layout more recognizable, increase map cognition and reduce geo-spatial error introduced by the layout procedure. We approximate a river to match the cartogram style and to simplify testing for edge intersections (rather than using the original.) At first, a river approximation can be converted from geographic map positions to topological positions that separate the CCG regions on either side of the river. Our process for deriving a river approximation is as follows:

1. We overlay the river shapefile [Ope] and combine it with the CCG region map [NHSb].
2. We identify the CCG regions on both sides along its full length.
3. For each CCG region adjacent to the river, we couple corresponding CCGs on opposite sides of the river and save matching pairs of region nodes based on closest centroids. See Figure 3.
4. For each pair of corresponding CCG regions, we connect their centroids with an edge,  $e(c_1, c_2)$ .
5. We add the mid-point of each edge  $e(c_1, c_2)$  as a vertex  $v(r)$  on a polyline representing the river,  $r(v_0, \dots, v_n)$ .
6. Connect all the derived river vertices,  $v(r)_0, \dots, v(r)_n$  and add the river to the cartographic map. See Figures 3 and 4 for an illustration of this process.



**Figure 3:** This figure illustrates how we select pairs of CCG regions spanning the Thames from the QGIS file. Showing a subset of the Thames first shows CCG 08C and 08X on the border. We identify 08C and 08X as a corresponding pair for river definition and approximation. The river continues between 07Y and 08P, we identify 07Y and 08P as the next pair. If the river flows directly in the middle of one region, such as 08P, a nearby CCG region for this segment of the river is selected. In this case we add 07Y and 08J as a third pair.



**Figure 4:** This figure illustrates inserting the river polyline on the cartographic map by connecting all the derived river vertices. Green lines show edges connecting pairs of CCG regions, and the river line is formed by connecting the mid-point of all green lines. The red rectangle highlights the parts of detail view of Figure 1.

### 3.4. Compute Region Center Points

To generate a Doring style or rectangular cartogram, we start with small rectangles to represent each CCG region, and increase the size of each rectangle until they gradually reach the maximum size of which the user has control. Initially we input the original CCG map from NHS website [NHSb] into QGIS [QGI] and use the centroid calculation tool to calculate the center points of each CCG region. We export the position of center points to a CSV file and use it as an input to our cartogram as the starting positions of the rectangular region nodes. Sample center points can be seen in Figures 3 and 4.

### 3.5. Update Node Size and Remove Overlap

We start with a unit square to represent each CCG region as a node in the cartogram and increase the size of each node gradually. During the region growing process, one region may shift adjacent neighboring regions in order to remove overlap and preserve relative position. We use the fast overlap removal algorithm [DMS06, DMS07] incrementally for this process. When all regions reach their maximum size or one of the regions reaches the boundary of map, the cartogram layout stops. In each step we increase the

size of all nodes by 1 pixel (if they have not reached their maximum size) to make them grow smoothly and gradually.

---

#### Algorithm 1 TestRiverIntersection

---

*Input*

*CCGList* : list of CCG regions

*river*: list of river edges

*Local Variables*

*Edge* : a line segment connecting current position and previous position of each CCG region

*Output*

*intersectionList*: a list of CCGs that intersect the rivers

```

1: procedure TESTRIVERCROSSING (CCGList , river)
2:   intersectionList
3:   for index = 0; index < CCGList.size(); index++ do
4:     Line Edge = Line(CCGList.at(index).getPosition(),
5:       CCGList.at(index).getPreviousPosition())
6:     if TestIntersection ( Edge, riverEdge) == TRUE then
7:       intersectionList.append(CCGList.at(index))
8:     endIf
9:   endFor (index)
10:  return intersectionList

```

---

### 3.6. Test For River Intersection

We also incorporate a river intersection test algorithm to confirm that regions do not cross this topological feature. When the size of each region increases, we connect the current centroid with its previous centroid. If the connected edge intersects one of the river edges, we mark this region as an intersecting region. See algorithms 1 and 2. For the intersection test of two edges, we first test whether the bounding box of two segments intersect. If not, then the two segments do not intersect. Then we use a vector cross product to test whether two points of one segment are on the opposite side of the other segment. If they are both true, the two segments intersect [Eri04].

---

#### Algorithm 2 LayoutNode

---

Input–

*CCGList* : list of CCG regions

*river*: list of river edges

Local variables

*MaxSize* : the maximum size of CCG regions set by user

*crossing*: TRUE if a CCG region crosses river

*stalemate*: TRUE if a CCG region is placed back in its previous position

*reverseDirection*: the reverse direction that lead neighboring CCG regions to stagnation

```

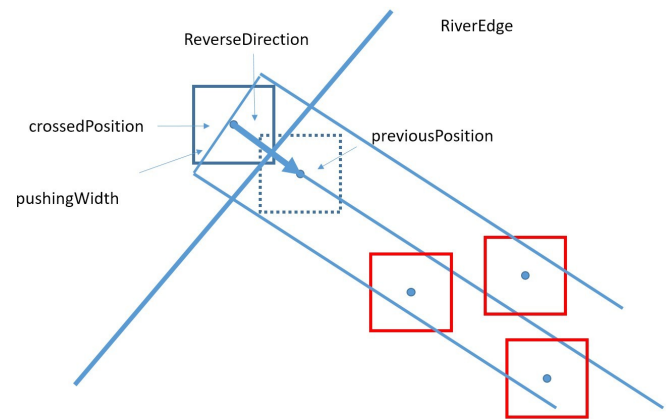
1: procedure LAYOUTNODE ( CCGList , river )
2:   while CCGList.getMaxSize() < MaxSize do
3:     intersectionList = TestRiverIntersection( CCGList, river )
4:     if intersectionList.getSize() > 0 then
5:       intersectionList.saveCrossedPositions()
6:       intersectionList.setPositions (
7:         intersectionList.getPreviousPositions()
8:       )
9:       boolean stalemate = intersectionList.isStalemate ()
10:      if stalemate == TRUE then
11:        intersectionList.deriveReverseDirections ()
12:        // push neighboring CCGs
13:        CCGList.counterMovement ()
14:      endIf // Stalemate
15:    else // No intersections
16:      CCGList.setSize ( CCGList.getSize()+ )
17:      CCGList.savePreviousPosition()
18:    endIf // Intersections
19:    RemoveOverlap()
20:    UpdateLayout()
21:  endWhile

```

---

### 3.7. Topology Preservation Algorithm

After testing river intersection for each region, we select the crossing regions and move them back to their previous position (those positions are stored because a unit area's centroid is not allowed to cross a river edge). This backward transition may cause overlap between regions, so we rerun the overlap removal algorithm to remove overlap. In some cases, a region may shift repeatedly between the same two positions by the topology preservation algorithm and overlap removal algorithm, which we identify as a stalemate. When a stalemate occurs, we move the affected region back to its previous position again, and push all other overlapping regions in the reverse push region by same amount. Our first attempts at this algorithm used only a reverse push line segment. This alone was not enough to prevent a stalemate. Thus we introduced a reverse push



**Figure 6:** This figure illustrates the configuration when a CCG region (blue) crosses the river and is placed back to its previous position (dashed outline). A reverse direction is derived used to push all neighboring CCG regions (red) in the reverse direction. The reverse direction is turned into a reverse region by introducing a width,  $p_\epsilon$ .

region. See Figure 6. The width of the reverse push direction region,  $p_\epsilon$ , may also be adjusted by user. See Algorithm 2 for a complete procedural description.

### 3.8. Test Region Size and Domain Boundaries

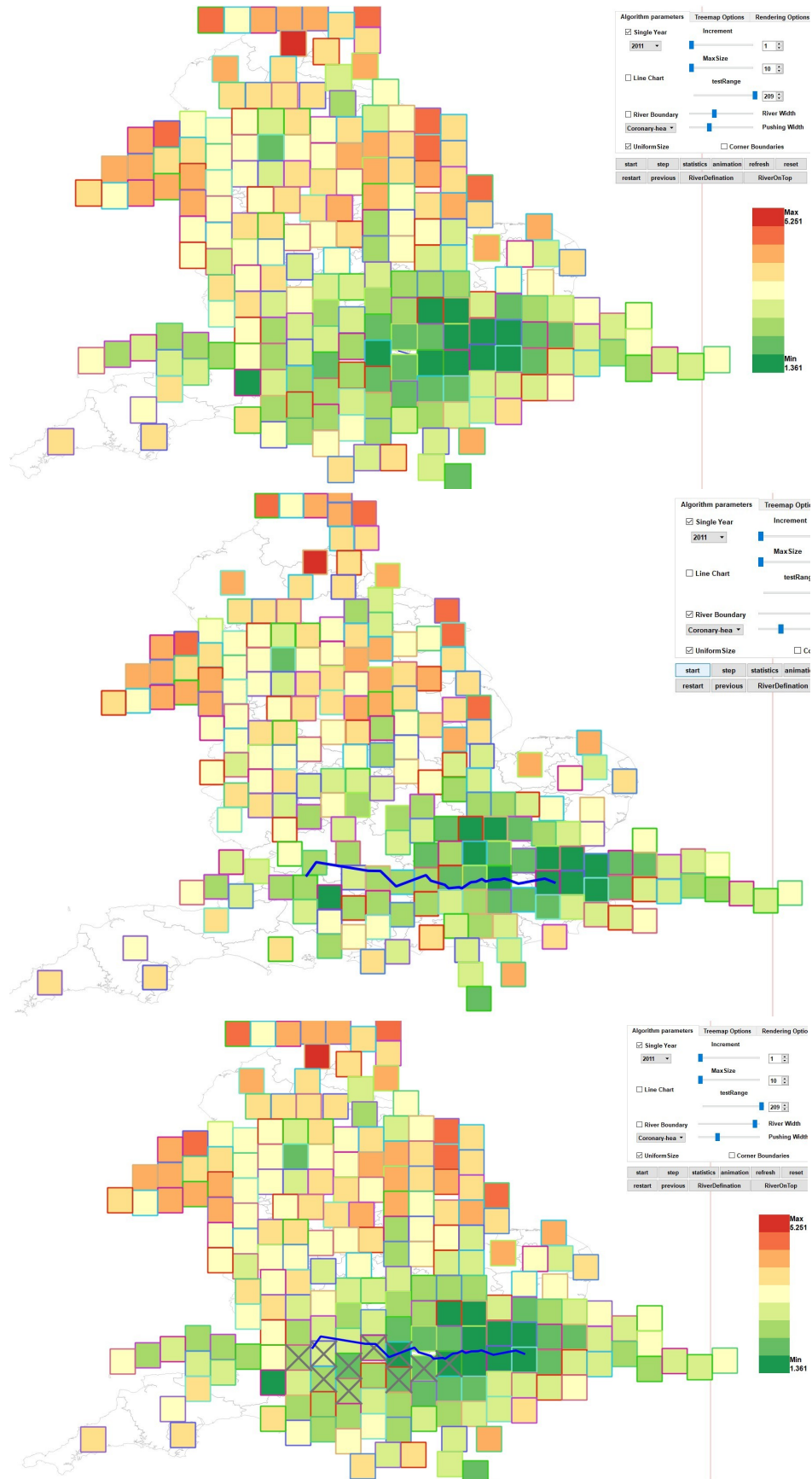
We use maximum size as a control for cartogram node size. The cartogram generation algorithm stops when all the region nodes reach a maximum size. It is same size when user choose uniform size nodes or the maximum size node when user chooses unit area size mapped to area population.

The domain boundaries are set to ensure each node does not move outside the screen. When a region node reaches the north/south boundaries, it will be shifted to west/east direction which enables the cartogram generation algorithm to continue. When a region node reaches west/east boundaries, the algorithm terminates. This decision is made due to the elongated shape of the UK in the north/south orientation.

## 4. Results and Discussion

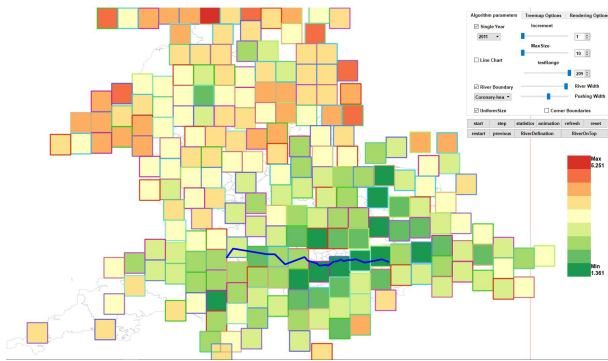
For a video demonstration of the algorithm and more results, please see the accompanying video. In this section, we present our cartograms with and without topological features. Figure 5 (top) shows the typical cartogram generated without any topological features. Figure 5 (middle) presents the output of the cartogram with the Thames river topology. The reverse pushing width is set to  $p_\epsilon = 40\%$  of a unit region width. Figure 8 shows the cartogram with Thames river and a larger pushing width  $p_\epsilon = 100\%$  of a unit region width. Figure 5 (bottom) shows the regions that cross the river topology if no intersection test is performed. In other words, topologically incorrect region nodes.





submitted to EG UK Computer Graphics & Visual Computing (2018)

**Figure 5:** The top shows the basic cartogram without topological features. The middle shows the cartogram with Thames river and a narrow push width,  $p_{\epsilon}$ . The bottom shows regions marked with a gray cross are those that cross the Thames river if the topology is not preserved. Color is mapped to Coronary heart disease distribution in England.



**Figure 8:** This figure shows the cartogram with Thames river and a wide pushing width,  $p_{\epsilon} = 100\%$ . Color is mapped to Coronary heart disease distribution in England.

Also, instead of using uniform size, we can map the size of node to each regions' population. Figure 7 (top) shows the standard cartogram with size mapped to population. Figure 7 (middle) shows the cartogram with Thames river feature and mapped to population. Regions in Figure 7 (bottom) marked with a gray cross are those that cross the Thames river. If the topology is not preserved. There are 10 error regions crossing the river in the figure, which are Oxfordshire, Chiltern, Windsor Ascot & Maidenhead, Hillingdon, Slough, Ealing, Hammersmith Fulham, Hounslow, Richmond and Wandsworth.

We are using color from dark green to dark red (from ColorBrewer [Col]) to illustrate a user-chosen health care prevalence from minimum to maximum. In the figures we present, the color is mapped to coronary heart disease distribution in England. Users also can select any healthcare prevalence through a menu and generate different cartograms. We also enable the user to select the maximum size of a region to control the trade-off between space-filled and map accuracy.

We also offer zooming and panning tools that help users to zoom close-up for details of the cartogram.

To increase the river recognizability, as a user option, we also present a river width setting,  $r_{\epsilon}$ . A region is marked as crossing a river not just when its center point crosses the river polyline, but when the distance between its center point and any river polyline segments is less than a user specified distance. See Figure 1. For a supplementary video showing further results, please see the accompanying video.

## 5. Conclusion and Future Work

This paper presents cartograms with topological features which increase the recognizability of a cartographic map and reduce layout error. We convert real geographic information such as a river into a topological feature on the cartogram using a definition and approximation algorithm. Then we implement a Dorling style cartogram incorporating this topological feature constraint and display CCG population and healthcare data. Several interactive user options are available to explore and present the results focusing on different

user requirements for further exploration, analysis and comparison. Future work includes investigating more topological features, e.g. including more major rivers and a more in-depth user feedback study.

## 6. Acknowledgment

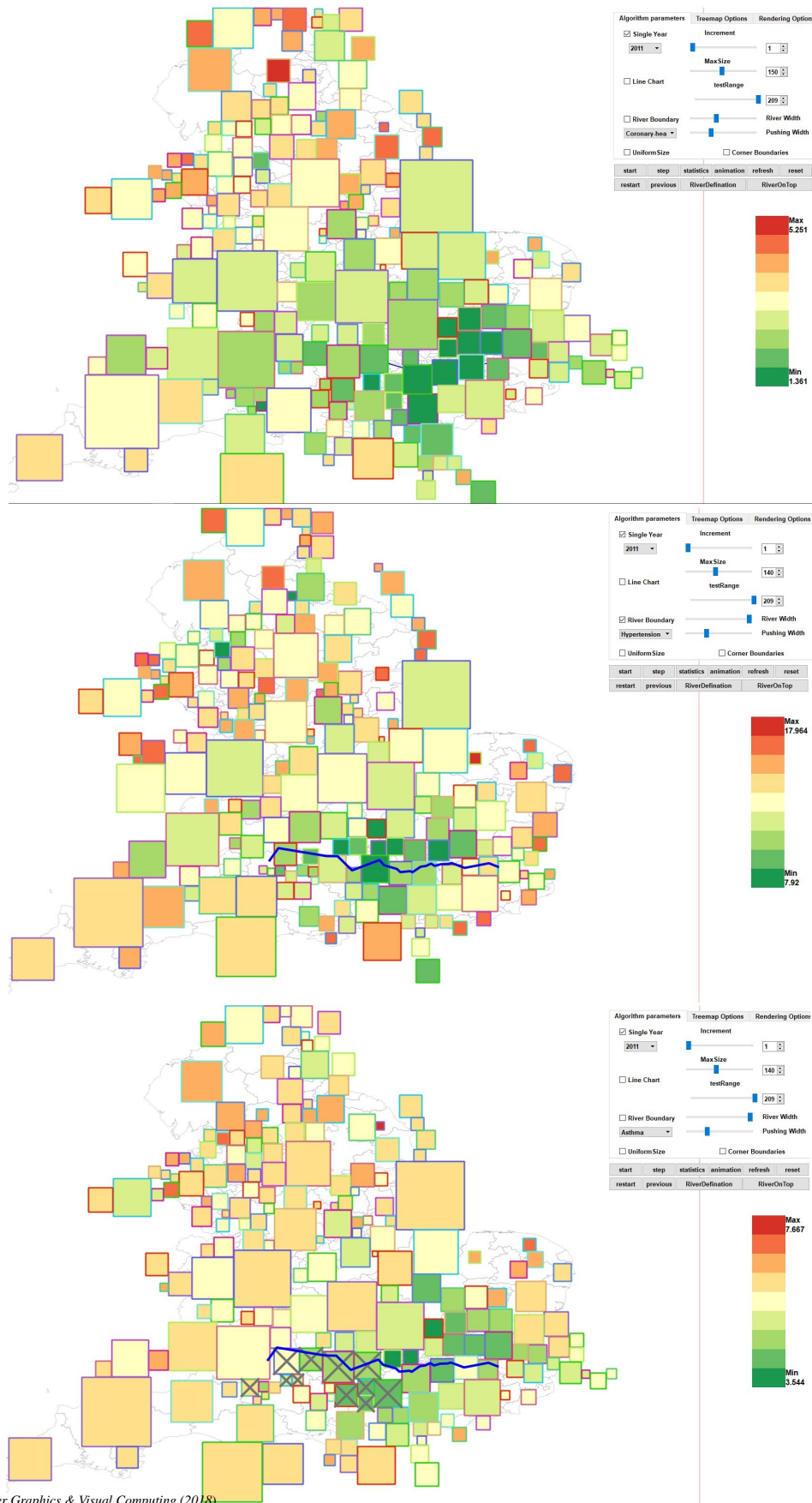
Thanks to Richard Roberts, Liam McNabb, Sean Walton and Dylan Rees for proofreading this paper.

## References

- [AHL\*11] AUBER D., HUET C., LAMBERT A., SALLABERRY A., SAULNIER A., RENOUST B.: Geographical Treemaps. *Technical Report* (2011). 2
- [AKV\*15] ALAM M., KOBUROV S. G., VEERAMONI S., ET AL.: Quantitative Measures for Cartogram Generation Techniques. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 351–360. 2
- [API] API O. T.: <http://overpass-turbo.eu/>. 3
- [CCM15] CRUZ P., CRUZ A., MACHADO P.: Contiguous Animated Edge-Based Cartograms for Traffic Visualization. *IEEE Computer Graphics and Applications* 35, 5 (2015), 76–83. 1
- [Col] ColorBrewer. <http://colorbrewer2.org/>. 7
- [DMS06] DWYER T., MARRIOTT K., STUCKEY P. J.: Fast Node Overlap Removal. In *Graph Drawing* (2006), Springer, pp. 153–164. 4
- [DMS07] DWYER T., MARRIOTT K., STUCKEY P. J.: Fast Node Overlap Removal Correction. In *Graph Drawing* (2007), Springer, pp. 446–447. 4
- [Dor11] DORLING D.: Area Cartograms: Their Use and Creation. *The Map Reader: Theories of Mapping Practice and Cartographic Representation* (2011), 252–260. 2
- [Eri04] ERICSON C.: *Real-time Collision Detection*. CRC Press, 2004. 5
- [EvKSS15] EPPSTEIN D., VAN KREVELD M., SPECKMANN B., STAALS F.: Improved Grid Map Layout by Point Set Matching. *International Journal of Computational Geometry & Applications* 25, 02 (2015), 101–122. 2
- [GN04] GASTNER M. T., NEWMAN M. E.: Diffusion-based Method for Producing Density-equalizing Maps. *Proceedings of the National Academy of Sciences of the United States of America* 101, 20 (2004), 7499–7504. 2
- [HKPS04] HEILMANN R., KEIM D. A., PANSE C., SIPS M.: Remap: Rectangular Map Approximations. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004.* (2004), IEEE, pp. 33–40. 2
- [KNP04] KEIM D. A., NORTH S. C., PANSE C.: Cartodraw: A Fast Algorithm for Generating Contiguous Cartograms. *IEEE Transactions on Visualization and Computer Graphics* 10, 1 (2004), 95–110. 2
- [LHZP07] LARAMEE R. S., HAUSER H., ZHAO L., POST F. H.: Topology-based Flow Visualization, The State of The Art. In *Topology-based methods in visualization*. Springer, 2007, pp. 1–19. 1
- [MDS\*17] MEULEMANS W., DYKES J., SLINGSBY A., TURKAY C., WOOD J.: Small Multiples with Gaps. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 381–390. 2
- [ML17] MCNABB L., LARAMEE R. S.: Survey of Surveys (SoS) - Mapping The Landscape of Survey Papers in Information Visualization. *Computer Graphics Forum* 36, 3 (2017), 589–617. 2
- [NHSa] NHS: <http://fingertips.phe.org.uk/profile/general-practice>. 3
- [NHSb] NHS: <https://www.england.nhs.uk/resources/ccg-maps/>. 3, 4

- [NK16] NUSRAT S., KOBOUROV S.: The State of the Art in Cartograms. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 619–642. 1, 2
- [Ope] OPENSTREETMAP: <https://www.openstreetmap.org/>. 3
- [PSKN06] PANSE C., SIPS M., KEIM D., NORTH S.: Visualization of Geo-spatial Point Sets via Global Shape Transformation and Local Pixel Placement. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 749–756. 2
- [PVH\*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The State of The Art in Flow Visualisation: Feature Extraction and Tracking. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 775–792. 1
- [QGI] QGIS: <http://www.qgis.org/en/site/>. 3, 4
- [Rai34] RAISZ E.: The Rectangular Statistical Cartogram. *Geographical Review* (1934), 292–296. 2
- [SDW09] SLINGSBY A., DYKES J., WOOD J.: Configuring Hierarchical Layouts to Address Research Questions. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 977–984. 2
- [SDW10] SLINGSBY A., DYKES J., WOOD J.: Rectangular Hierarchical Cartograms for Socio-economic Data. *Journal of Maps* 6, 1 (2010), 330–345. 2
- [TML\*17] TONG C., MCNABB L., LARAMEE R. S., LYONS J., WALTERS A., BERRIDGE D., THAYER D.: Time-oriented Cartographic Treemap for Visualization of Public Health Care Data. *Proceedings of the Conference on Computer Graphics and Visual Computing (CGVC), 2017* (2017). 2
- [Tob04] TOBLER W.: Thirty Five Years of Computer Cartograms. *ANNALS of the Association of American Geographers* 94, 1 (2004), 58–73. 2
- [TRL\*17] TONG C., ROBERTS R., LARAMEE R. S., BERRIDGE D., THAYER D.: Cartographic Treemaps for the Visualization of Public Health Care Data. *Proceedings of the Conference on Computer Graphics and Visual Computing (CGVC), 2017* (2017). 2
- [TRR\*18] TONG C., RICHARD R., RITA B., ROBERT S L., KODZO W., AIDONG L., WANG Y., QU H., LUO Q., MA X.: Storytelling and Visualization: A Survey. In *Proceedings of the 9th International Conference on Information Visualization Theory and Applications (IVAPP) 2018* (forthcoming, 27-29 January 2018), Funchal, Madira, Portugal. 2
- [vKS07] VAN KREVELD M., SPECKMANN B.: On Rectangular Cartograms. *Computational Geometry* 37, 3 (2007), 175–187. 2





submitted to EG UK Computer Graphics & Visual Computing (2018)

**Figure 7:** The figure shows the cartogram with unit area size mapped to population without (top) and with (middle) the Thames topology feature. Regions marked with a grey cross are those that cross the Thames river if the topology is not preserved (bottom). The color is mapped to hypertension prevalence in England.