A Constraint-Based Approach to Constructing Continuous Cartograms

Christopher J. Kocmoud[†]
Texas Center for Applied Technology
Texas Engineering Experiment Station

Donald H. House[‡]
Visualization Laboratory
Texas A&M University

Keywords: cartogram, value-by-area map, map transformation, anamorphosis, thematic cartography.

Abstract

We present a new constraint-based continuous area cartogram construction method that is unique in its ability to preserve essential cues for recognition of region shapes. It automatically achieves desired region areas while maintaining correct map topology. The algorithm is compared with a number of existing methods, and results are shown to be superior in both accuracy and preservation of shape recognition cues. Through hierarchical resolution, we first perform gross adjustments upon a coarsely resampled map and later refine the map at progressively higher levels of detail.

1 Introduction

The area cartogram is a useful tool for visualizing the geographic distribution of "routine" data in a variety of disciplines, including politics, social demographics, epidemiology, and business. Through the spatial transformation of map regions relative to the data, the cartogram prominently emphasizes *data distribution* instead of territorial size.

For example, the results of the popular vote in the 1996 U.S. presidential race are visualized in Figure 1a using traditional thematic mapping. There is a significant problem with this visualization. Without prior knowledge of population density, the viewer has no clear indicator as to who actually won the election. This map produces an intrinsic distortion of the data. The results would be better visualized on a map more representative of population. Figure 1b is an equal population cartogram of the same data generated using the *Constraint-Based Method* described in this paper. It clearly shows the winner.

A continuous area cartogram is one in which the topology, or connectivity, of the map regions is retained while areas are resized. Accurately resizing regions relative to a data variable while maintaining continuity and region recognition is a challenging task [8], that has not been solved effectively to date. This paper presents a new approach to this problem, that has promise to make the continuous area cartogram a common tool for cartographers.

2 Background

Several computer algorithms have already been developed to construct *continuous* area cartograms [1, 2, 4, 7, 9, 10, 11]. We present here five of the most effective of these methods and make comparisons of their results with ours at the end of the paper. We chose the names of the methods discussed here to be *descriptive* in nature, they may not necessarily be the names used by the authors.

[†] 245 Wisenbaker – M.S. 3407, College Station, Texas 77843-3407, Tel: (409) 862-3272, Fax: (409) 862-3336, c-kocmoud@tamu.edu.

[‡] 216 Langford – M.S. 3137, College Station, Texas 77843-3137, Tel: (409) 845-3465, Fax: (409) 845-4491, house@viz.tamu.edu.

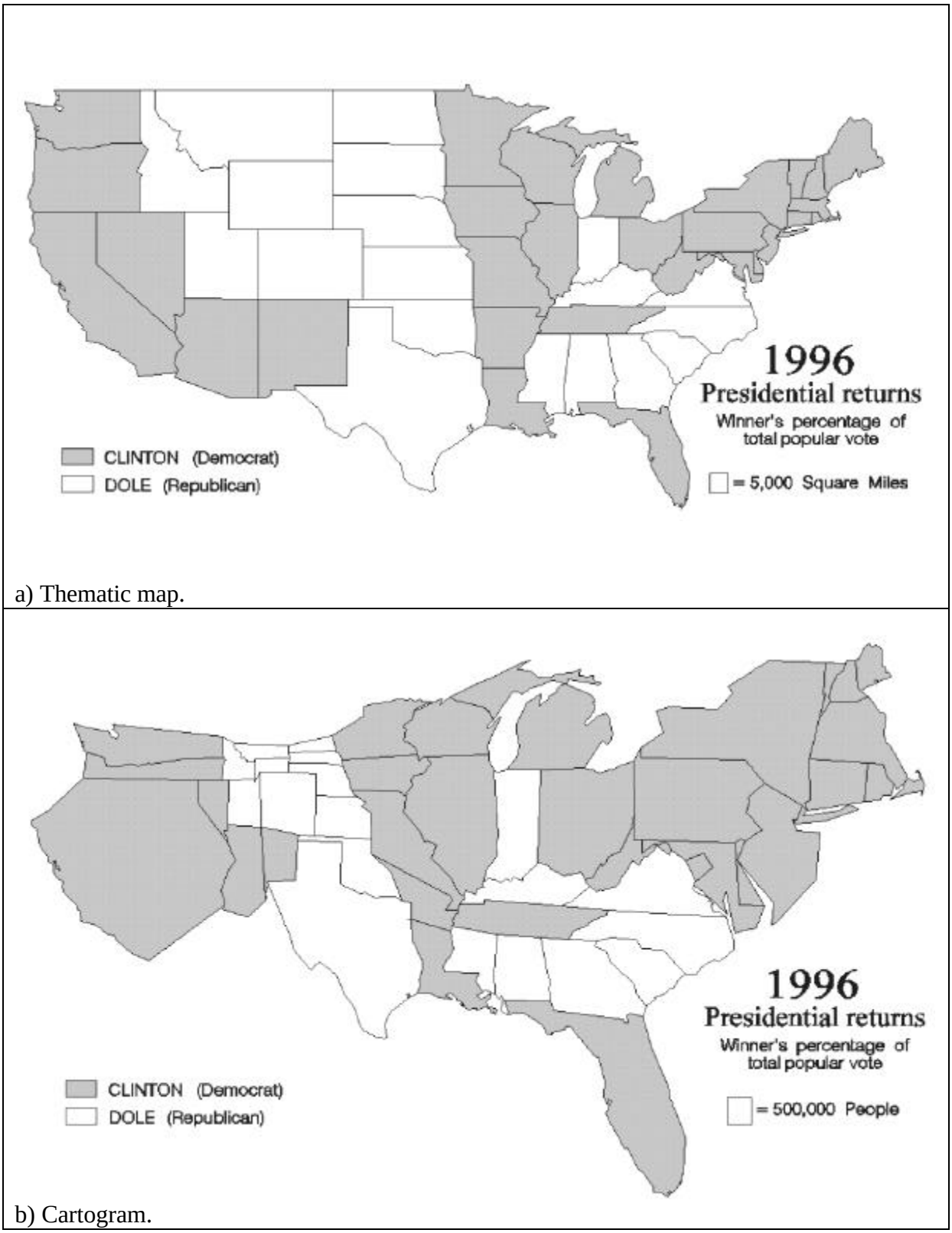


Figure 1: 1996 U.S. presidential return visualizations (Data Source: Federal Election Commission).

2.1 Continuous cartogram methods

Tobler's *Pseudo-Cartogram Method* creates an equal density approximation by compressing or expanding the lines of latitude and longitude until a least root mean square error solution is obtained [10]. This method provides an effective way to "preprocess" a map prior to cartogram construction but the cartograms produced can contain extensive area error. Dorling's *Cellular Automaton Method* is adapted from the "Game of Life," where a map has a grid superimposed on it and individual grid cells are traded until every geographic region obtains its desired number of cells [1]. While this method is very effective at achieving area, regions tend to lose their unique contours and acquire a shape reflecting the grid.

The three other methods are radial in nature. The *Radial Expansion Method* of Selvin et al. applies radial transformations from each region upon all map vertices such that the selected region expands or shrinks while leaving the area of all other regions unchanged [7]. The *Rubber Sheet Method* of Dougenik et al. exerts radial forces from each region upon all map vertices at a magnitude proportional to region area error and inversely proportional to distance [2]. Gusein-Zade and Tikunov's *Line Integral Method* applies radial transformations such that the density of a selected cell is made uniform while leaving all other cells unchanged, with the vector sum of transformations applied as a line integral around each of the region boundaries [4]. While the radial methods produce reasonable results in terms of area error, they produce both a "ballooning" effect that can render regions unrecognizable and a "pinching" of originally rectangular region corners.

Characteristics	Radial Expansion	Rubber Sheet	Pseudo-Cartogram	Cellular Automaton	Line Integral
1. Independent of region traversal order		✓	✓	✓	✓
2. Independent of coordinate axes	✓	✓		✓	✓
3. Conformal mapping	✓	✓	✓		✓
4. Global displacements per iteration		✓	✓		✓
5. Intersection prevention			✓	✓	
6. Ability to fix (pin down) points				✓	
7. User controls on *area* vs. *shape*					

Table 1: Desired cartogram method characteristics.

2.2 Desired cartogram method characteristics

In Table 1 we have expanded upon a previous review of methods by Gusein-Zade and Tikunov [5] and quantified seven characteristics that we feel, if attained, would lead to a comprehensive and versatile cartogram algorithm. Independence of results from vertex traversal order and the coordinate axes enables reproducible results, generating the same cartogram regardless of the organization of the map's source data base and orientation. In order to be an effective communication tool, the method should be conformal in its preservation of angles locally so that detailed areas on the cartogram are similar to the original map. Globality assures that *all* regions influence every map vertex, generally resulting in faster convergence to a solution. In order to guarantee quality results, the algorithm should also prevent the intersection and self-overlapping of regions. The last two characteristics provide user control of aesthetics by allowing

vertex locations to be pinned down or modified and by enabling the user to make shape versus accuracy trade-offs. None of the methods described above possesses all of the first five technical requirements, and none, with the exception of the *Cellular Automaton Method*, pay any attention to aesthetic controls.

3 The Algorithm

In designing our algorithm, we recognized that the process of creating continuous cartograms can be broken down into two distinct but conflicting tasks: adjusting region sizes and retaining region shapes. The algorithm, as shown in Figure 2, iterates over a map sampled at consecutively higher levels of detail [6]. Within a level of detail, it achieves desired areas without regard to shape, and then restores shape while attempting to hold the areas fixed. These tasks are performed within the modeling paradigm of a constraint-based physical system.

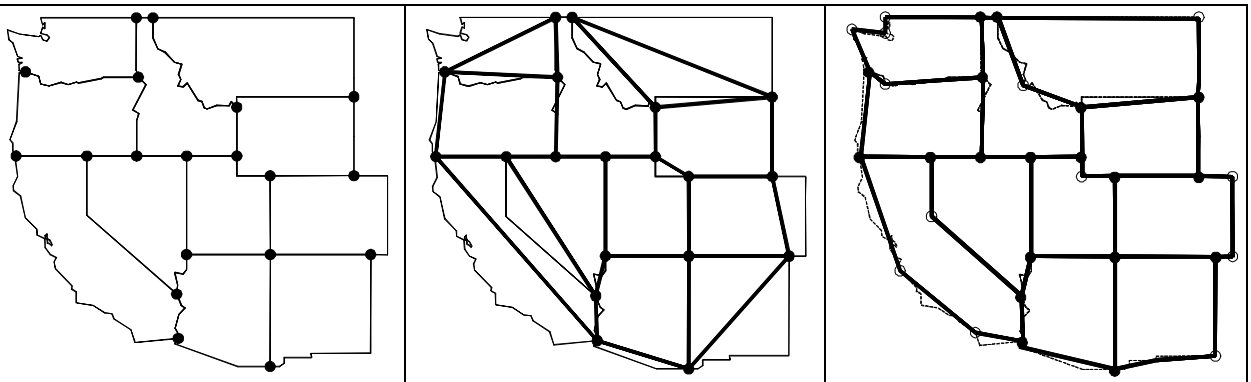
The map vertices and region data are loaded or obtained via GIS database linkages. The map is initially acted upon at a coarse resolution and refined later at progressively higher levels of detail. At the heart of our method is a repetitive "relaxation process." We alternate between the two goals of resizing regions to their correct areas and restoring region shapes while attempting to hold their areas fixed, switching goals when the solution "stagnates."

```
Regions := LoadFullResolutionMap (MapDataFile);
coarseness := max_coarseness;
AreaTargets := CalculateDesiredArea (Regions);
repeat
        SimplifyMap (Regions, coarseness);
        repeat
            AchieveAreas (Regions, AreaTargets);
            RestoreShapeWhileMaintainingArea (Regions);
        until adverse effect of area upon shape increases;
        ReconstructMapToFullResolution (Regions, coarseness);
        coarseness := coarseness / 2;
until reached desired level of detail;
```

Figure 2: Pseudocode of our continuous cartogram algorithm.

3.1 Hierarchical resolution

The first step in the simplification of map resolution is the identification of certain shared vertices that would cause a break in the map topology if they were simplified away. These "key points" can be defined as interior vertices shared by three or more regions and as perimeter vertices shared by two or more regions. The key points of a western U.S. map are highlighted in Figure 4a. Simply connecting the key points, as demonstrated in Figure 4b, introduces gross shape errors. Instead, we simplify between key points at a resolution relative to the size of the region, as shown in Figure 4c.



a) Identification of "key" shared points in the western U.S.

b) Direct simplification between key points.

c) Simplification at a resolution relative to region sizes.

Figure 4: Identification of map key points and example simplifications.

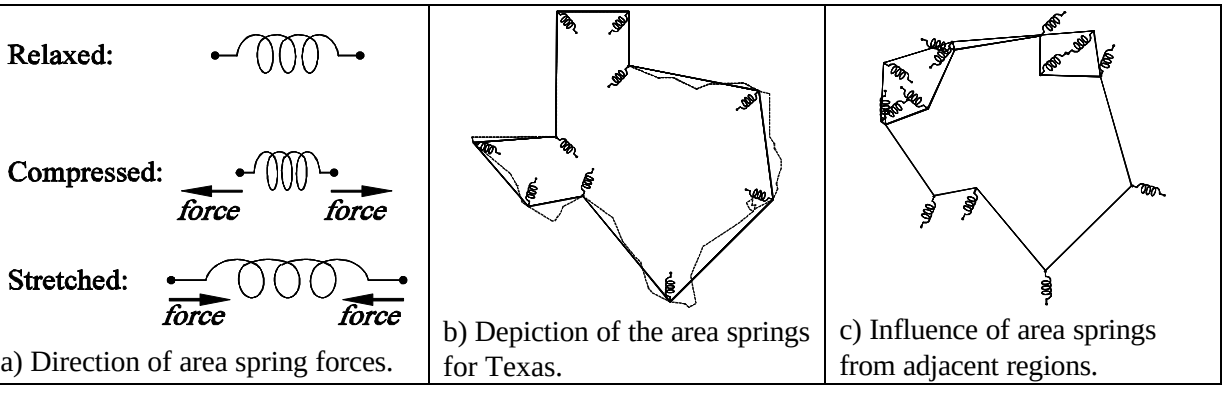
A minimum number of simplified edges are constructed between two key points such that the distance between any vertex and its simplified edge is within an allowed offset distance, computed as a percentage of the length of the region's bounding box diagonal. In this manner the simplification is based upon a collection of resolutions custom-scaled to each region, providing a balanced simplification of region details that is independent of region size.

3.2 Dynamics

Our method uses a dynamic system paradigm in which area and shape maintaining forces act upon the map vertices. We frequently apply strong, one-time forces upon vertices to prevent a break in map topology, which in a momentum-based Newtonian physical system would lead to oscillations. Instead, we use Aristotelian dynamics, where the *velocity* of a point is directly proportional to the total force upon it, so that a vertex only moves when a force is acting upon it.

3.3 Achieving desired areas

Region areas are achieved using *area springs*, as demonstrated in Figure 5. Area springs are situated so that they scale a region by exerting equal forces upon each vertex in a direction that bisects each interior angle, as depicted in Figure 5b. However, as demonstrated in Figure 5c, adjacent regions also exert area forces upon shared vertices, resulting in a tug of war, with the region of greatest error having the stronger hand.



a) Direction of area spring forces.

b) Depiction of the area springs for Texas.

c) Influence of area springs from adjacent regions.

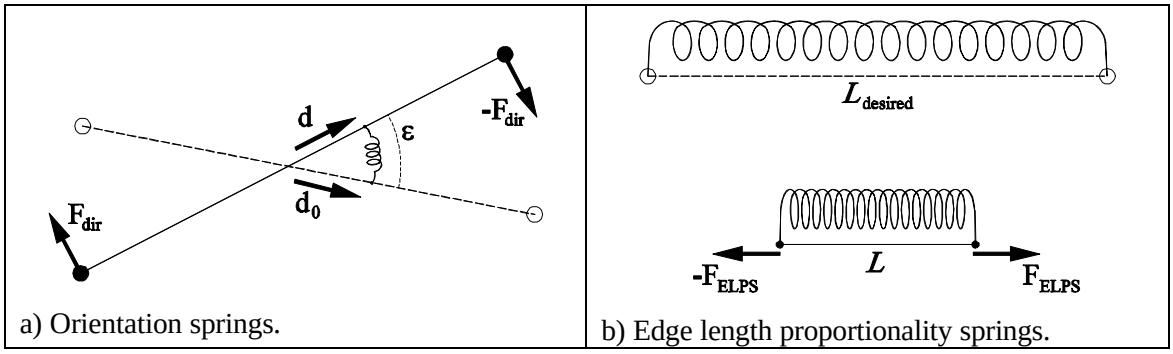
Figure 5: Examples of area springs.

The desired area of a map region is proportional to its share of the geographic quantity to be visualized. The force of each area spring is proportional to the percentage area error of the region.

The pseudocode for resizing regions is given in Figure 6. It begins by distributing the area spring forces to the region vertices. This is followed by superimposing topological constraint forces that prevent regions from inverting and intersecting. These are explained in more detail below. The net forces are then collectively applied upon the individual vertices during the dynamics procedure, thereby affecting vertex velocities which, in turn, modify point positions.

```
procedure AchieveAreas (Regions, AreaTargets);
    repeat
        DistrubuteAreaSpringForces (Regions, AreaTargets);
        DistributeTopologicalConstraintForces (Regions);
        PointDynamics (Regions);
    until average area error increases;
```

Figure 6: Pseudocode of the process to resize regions to their desired areas.



a) Orientation springs.

b) Edge length proportionality springs.

Figure 7: Orientation springs and edge length proportionality springs.

3.4 Restoring region shapes

Shape restoration is achieved by *orientation springs* and *edge length proportionality springs*. Orientation springs influence region edges to maintain their initial map orientation, as demonstrated in Figure 7a. We compute the magnitude of each orientation spring force relative to the edge's deviation from its initial orientation. Edge length proportionality springs influence region edges to remain in the same proportion to each other as they were in the original map, as demonstrated in Figure 7b. We compute the magnitude of each edge length proportionality spring force relative to the percentage difference between the edge's current and desired proportional lengths.

The shape restoration pseudocode, given in Figure 8, begins with the distribution of shape forces, either from the orientation springs or the edge length proportionality springs. We have found that isolating the two shape mechanisms, whose forces are often contradicting, enables each to retain their ground more effectively against the subsequent adversarial procedure, the area

constraint. The looping construct halts when the average shape error, computed as a weighted sum of degrees of orientation error and unit lengths of edge proportion error, ceases to decrease.

```
procedure RestoreShapeWhileMaintainingArea (Regions);
  useOS := true;
  repeat
      if useOS then
          DistrubuteOrientationSpringForces (Regions);
      else
          DistributeEdgeLengthProportionalitySpringForces (Regions);
      useOS := not useOS;
      DistributeAreaConstraintForces (Regions);
      DistributeTopologicalConstraintForces (Regions);
      PointDynamics (Regions);
   until average shape error increases;
```

Figure 8: Pseudocode of the shape restoration process.

3.5 Maintaining area

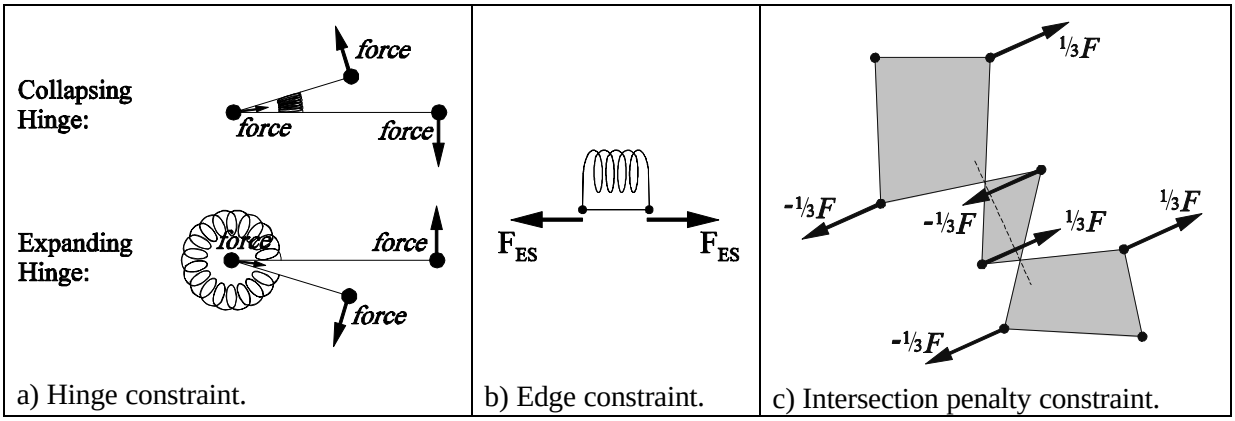
The key component of the shape restoration process, and our entire method, is holding the region *areas* fixed while their *shapes* are readjusting. This is enabled through dynamic area constraints, which attempt to cancel those components of the shape forces that would cause a change in area. The resulting constrained dynamic environment enables shape adjustments to occur without significant loss of accuracy in area. We utilize constrained particle dynamics, as detailed in tutorial form by Witkin in [12], to ensure that our regions obey a specific area constraint. We modify Witkin's formulation, as explained in [6], to utilize Aristotelian dynamics. As shown in the pseudocode in Figure 8, the area constraining forces are computed *following* the distribution of the shape restoration forces. This is necessary since the area constraints serve as a mediator, canceling shape forces as necessary to keep the region areas fixed. Once the constraint forces are computed and applied to the vertices, topological constraint forces are applied to ensure the integrity of the map topology.

3.6 Maintaining topological integrity

The map topology is maintained by *hinge constraints*, *edge constraints*, and *intersection penalty constraints*. The hinge constraint restricts the angle between two adjacent edges from opening or closing beyond their limits, as demonstrated in Figure 9a. The hinge gradually applies force only when the angle nears 0° or 360°, at a magnitude inversely proportional to the hinge angle. The edge constraint prevents an edge from flipping to negative length, as demonstrated in Figure 9b. Force is gradually exerted as the edge approaches zero length, at a magnitude inversely proportional to the edge length.

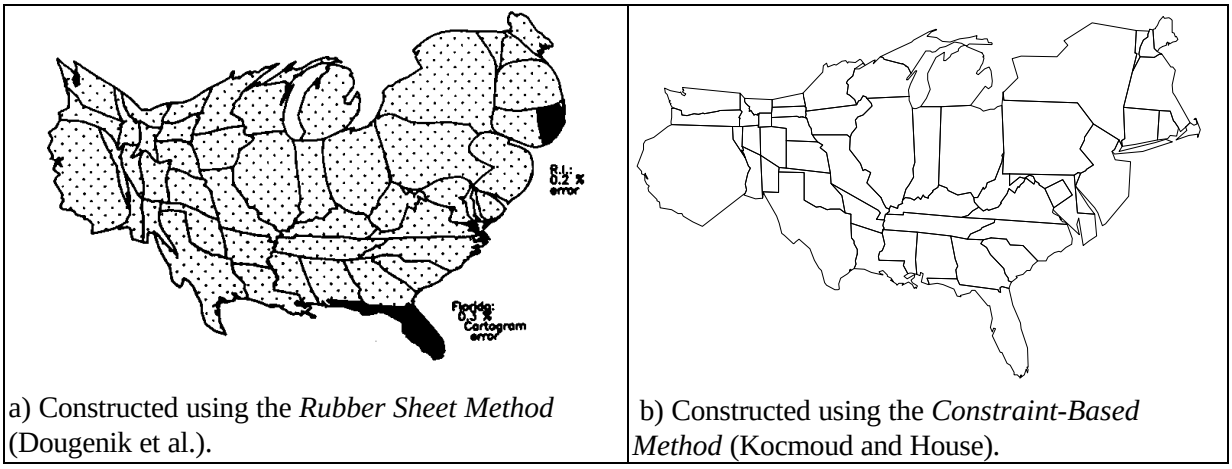
Polygonal regions may either cross over each other or self-intersect. To prevent a break in the map connectivity, we employ the additional topology maintaining mechanism of *intersection penalty constraints*. These exert forces, as demonstrated in Figure 9c, to correct situations where

a region has overlapped itself or others. Intersections are detected using the parametric line clipping method in [3].



a) Hinge constraint. b) Edge constraint. c) Intersection penalty constraint.

Figure 9: Examples of topological constraints.



a) Constructed using the *Rubber Sheet Method* (Dougenik et al.).

b) Constructed using the *Constraint-Based Method* (Kocmoud and House).

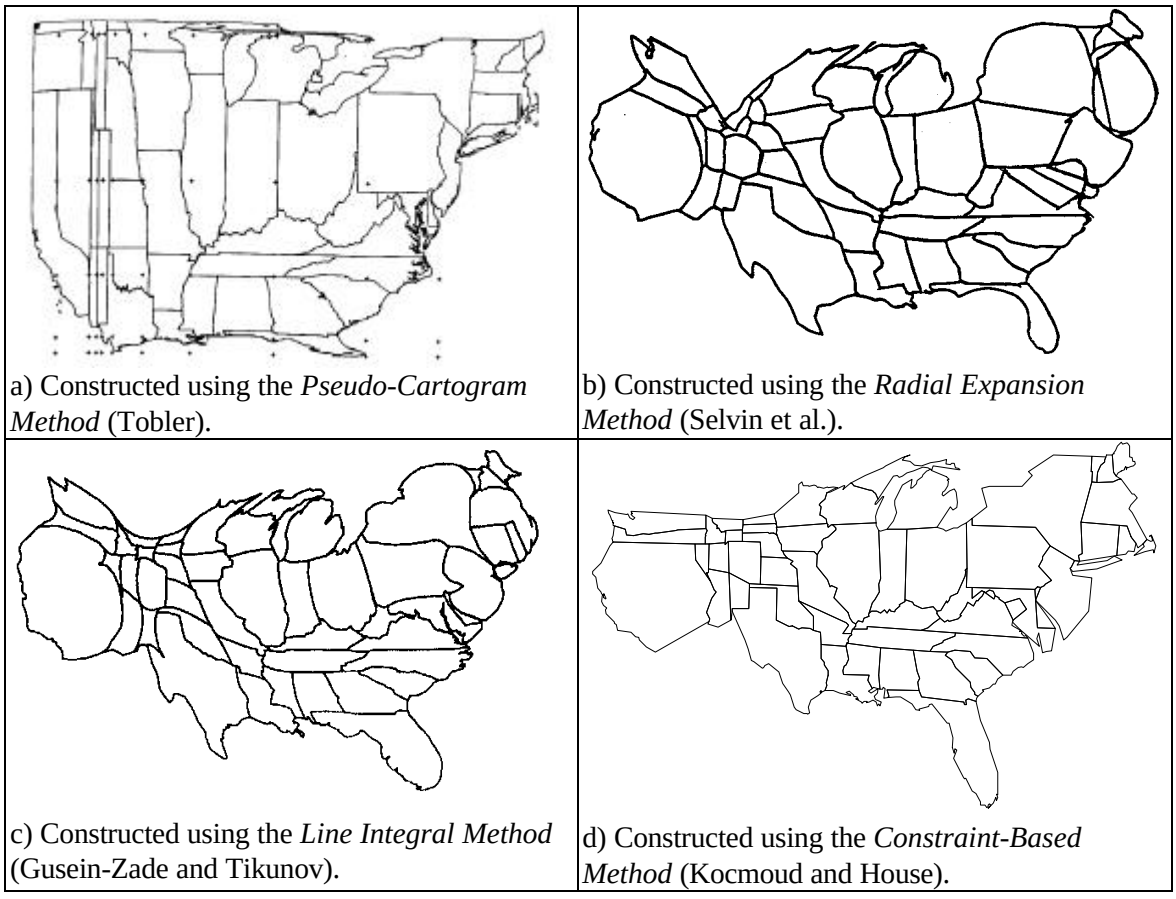
Figure 10: 1960 U.S. population cartograms (a: Adapted from and reproduced with permission from [2], page 80, Figure 6).

4 Result Comparisons

The results of our cartogram algorithm and previous algorithms are presented here. We begin with a 1960 U.S. population cartogram comparison with the *Rubber Sheet Method* in Figure 10. While the *Rubber Sheet* cartogram retains a better likeness of the western coastline, it also produces a severe pinched distortion of the non-coastal western states that inhibits recognition. The overall appearance of the northeastern states is better represented in our cartogram when compared to the generalized "ballooning" appearance in the other method. Our map contains only 2.2% average area error, half a percent away from the other map's 1.7% area error.

In Figure 11 we show 1980 U.S. population cartograms generated from various methods. While the shapes are preserved well in the *Pseudo-Cartogram* in Figure 11b, it has a 60% area error. The *Radial Expansion* cartogram in Figure 11c suffers most from circular shape

generalizations and loss of state contour details. Overall, the *Line Integral* cartogram in Figure 11d stands out among the radial algorithms, displaying more controlled "ballooning" than the *Radial Expansion* cartogram and reportedly approaching near 1% area error. However, the ballooning artifact still exists, and pinching of rectangular corners is still a problem. By contrast, our method is successful in preserving the distinctive shapes of nearly all the states, while still achieving a high accuracy of 4.4% average area error. Accuracy data is not available for the *Radial Expansion* method.



a) Constructed using the *Pseudo-Cartogram Method* (Tobler).

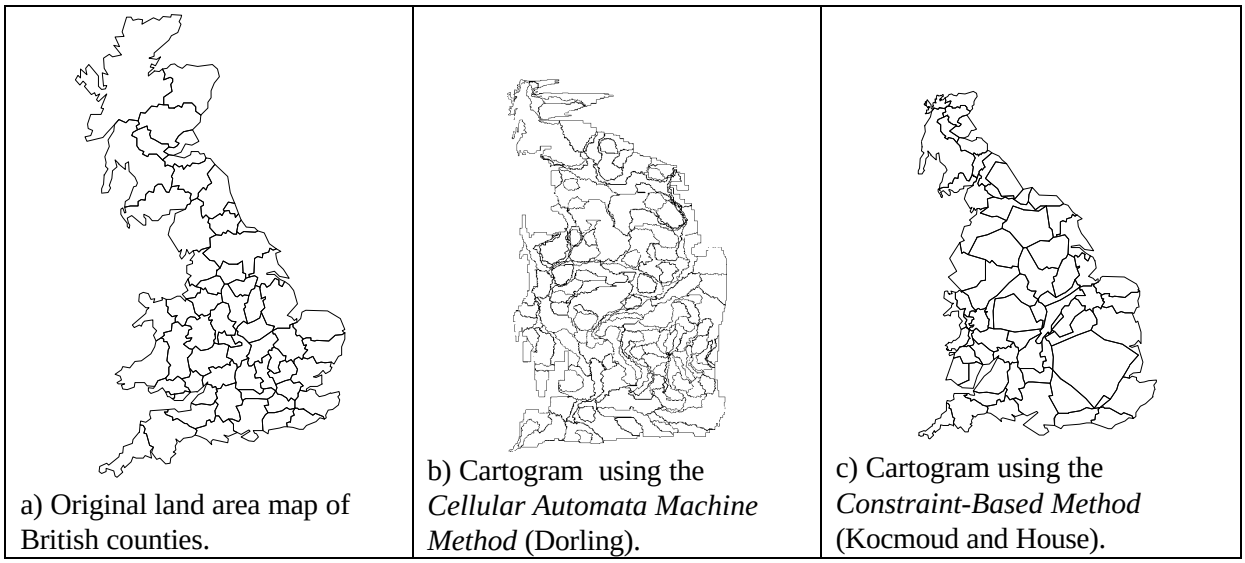
b) Constructed using the *Radial Expansion Method* (Selvin et al.).

c) Constructed using the *Line Integral Method* (Gusein-Zade and Tikunov).

d) Constructed using the *Constraint-Based Method* (Kocmoud and House).

Figure 11: 1980 U.S. population cartograms (a: Reproduced with permission from [10], page 49, Figure 8, © 1986 American Congress on Surveying and Mapping; b: Reproduced with permission from [7], page 21, figure 3; c: Reproduced with permission from [4], page 172, Figure 1, © 1993 American Congress on Surveying and Mapping).

In Figure 12 we compare our method with Dorling's *Cellular Automaton Method.* Note that he has included major cities in his 1981 equal population cartogram of British counties, whereas we have not. We see that the cellular approach severely distorts region shapes, rendering many of them unrecognizable. Our cartogram, however, is again successful in maintaining region shape. The bloated average area error of 28% in our cartogram is partly due to two sparsely populated counties, with 196% error each, that cannot shrink any further without compromising our specified level of shape preservation.



a) Original land area map of British counties.

b) Cartogram using the *Cellular Automata Machine Method* (Dorling).

c) Cartogram using the *Constraint-Based Method* (Kocmoud and House).

Figure 12: 1981 equal population cartograms of Britain (a, b, c: Reproduced with permission from [1], pages 20-22, Figures 14b,14c, and 14d).

5 Discussion

Our method, as implemented, includes all seven of the desirable cartogram algorithm features listed in Table 1. It is structured so that all forces are accumulated and applied at once, so results are independent of traversal order, and all calculations are coordinate frame independent. It preserves angles locally as much as possible, and prevents topology changes through edge intersections. It is also possible for vertices to be fixed and for the area/shape tradeoff to be adjusted. Although we do not, as yet, perform global displacements when adjusting areas, our area constraint forces during the shape restoration process are the direct result of a global calculation and effectively hold all areas once they are attained.

Our method was implemented in the MicroStation Development Language, a standard C programming language that compiles and executes within the MicroStation CAD program (Bentley Systems, Inc.). The map input data can be loaded from a formatted text file or converted directly from GIS data. Our examples were created on a 120MHz Toshiba Satellite Pro 430 notebook computer with 32MB RAM. Transforming the U.S. map with its 744 vertices took 18 hours for the 1960 population cartogram and 16 hours for the 1980 cartogram. The 1981 population cartogram of British counties contains 983 vertices and ran for 24 hours.

A significant feature of our method is the ability to incorporate interactive aesthetic control, enabling the user to fix small problems at any time by stiffening particular springs or by modifying or pinning down vertex locations. The user can also adjust the level of area accuracy with respect to sacrifice in shape at any time. Since we wished to stress the automatic nature of our algorithm, we have not hand-adjusted any of our examples shown here. However, these features could easily be integrated into an attractive interface.

6 Conclusion

We have demonstrated a new method for automatically generating continuous area cartograms that appears to be a significant improvement over previous methods. The technique offers easy map reproducibility as well as the opportunity for interactive aesthetic control. It runs on commonly available computers at speeds that could make it a useful tool for general individual use in geography.

References

[1] Dorling, D. *Area Cartograms: Their Use and Creation*, Department of Geography, University of Newcastle upon Tyne, Newcastle upon Tyne, England, August 1995.

[2] Dougenik, J.A., N.R. Chrisman and D.R. Niemeyer. "An Algorithm to Construct Continuous Area Cartograms," *The Professional Geographer*, Vol. 37, No. 1, 1985, pp. 75-81.

[3] Foley, J.D., A. van Dam, S.K. Feiner and J.F. Hughes. *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[4] Gusein-Zade, S.M. and V. Tikunov. "A New Technique for Constructing Continuous Cartograms," *Cartography and Geographic Information Systems*, Vol. 20, No. 3, 1993, pp. 167-173.

[5] Guseyn-Zade, S.M. and V.S. Tikunov. "Numerical Methods in the Compilation of Transformed Images," *Mapping Sciences and Remote Sensing*, Vol. 31, No. 1, 1994, pp. 66-85.

[6] Kocmoud, C.J. *Constructing Continuous Cartograms: A Constraint-Based Approach*, Masters Thesis, Texas A&M Visualization Laboratory, Texas A&M University, College Station, Texas, 1997.

[7] Selvin, S., D. Merrill, S. Sacks, L. Wong, L. Bedell and J. Schulman. *Transformations of Maps to Investigate Clusters of Disease*, Lawrence Berkeley Laboratory, University of California, No. LBL-18550, 1984.

[8] Tikunov, V.T. and S. Gusein-Zade. "Map Transformations," *Geography Review*, Vol. 9, No. 1, 1995, pp. 19-24.

[9] Tobler, W.R. "A Continuous Transformation Useful for Districting," *Annals of the New York Academy of Sciences*, Vol. 219, No. 9, 1973, pp. 215-220.

[10] Tobler, W.R. "Pseudo-Cartograms," *The American Cartographer*, Vol. 13, No. 1, 1986, pp. 43-50.

[11] Torguson, J.S. *Cartogram: A Microcomputer Program for the Interactive Construction of Value-By-Area Cartograms*, Masters Thesis, Department of Geography, University of Georgia, Athens, Georgia, 1990.

[12] Witkin, A. "Constrained Dynamics," *ACM SIGGRAPH 97 Course Notes* (Course 19: Physically Based Modeling), 1997, pp. F1-F12.