An Optimized Rubber-Sheet Algorithm for Continuous Area Cartograms*

Shipeng Sun
University of Waterloo

This article optimizes a continuous area cartogram algorithm published in *The Professional Geographer* by Dougenik, Chrisman, and Niemeyer (DCN) in 1985. The DCN algorithm simulates a rubber sheet and is an iterative and approximate solution of cartogram construction. Although it remains popular because of its conceptual simplicity and cartographic quality, the DCN algorithm cannot completely preserve topology and its mathematical properties are inadequately explained. This article presents an optimization to the DCN algorithm, named Opti-DCN, with three improvements. First, it provides a mathematical condition for topology preservation. Second, new transformation equations that meet this condition are deduced from mathematics, which simultaneously optimize the global elasticity coefficient, a key parameter that greatly impacts the convergence rate of the rubber-sheet algorithm and the topological integrity of its generated cartograms. Last, the new algorithm simplifies the way of generating transforming forces in DCN and improves its efficiency of geometric transformation. Comparison shows that Opti-DCN is significantly faster to converge to equal-density cartograms and can mathematically and practically eliminate topological errors. **Key Words: algorithm optimization, continuous area cartogram, rubber-sheet algorithm.**

Dougenik，Chrisman，和 Niemeyer (DCN) 在 1985 年于专业地理学杂志上发表了一个连续区域统计地图的算法，本文对该算法进行了优化。DCN 算法模拟拓扑橡胶板，是构建统计地图的一种迭代和近似的解。DCN 算法因为概念简单和制图质量仍然受到欢迎，但是该算法不能完整地保存拓扑关系，而且它的数学特性没有得到充分的解释。本文介绍了一种名为 OPTI-DCN 的优化算法，对 DCN 进行了三点改进。首先，它提出了一个保存拓扑关系的数学条件。其次，从数学上推导出满足这个条件的新的转换方程，同时优化了整体的弹性系数，该系数极大地影响了橡胶板算法的收敛速度和统计地图的拓扑完整性。最后，新的算法简化了 DCN 中的转化方式，提高了几何变换的效率。比较表明，OPT-DCN 使收敛到等密度统计地图的速度明显加快，并能在数学上和实际上消除拓扑错误。**关键词：算法优化，连续区域统计地图，橡胶板算法。**

En este artículo se optimiza un algoritmo de cartograma de área continua publicado en *The Professional Geographer* por Dougenik, Chrisman y Niemeyer (DCN) en 1985. El algoritmo de DCN simula una lámina de caucho y es una solución iterativa y aproximada para la construcción de cartogramas. Aunque sigue siendo popular debido a su simplicidad conceptual y calidad cartográfica, el algoritmo de DCN no puede preservar totalmente la topología y además sus propiedades matemáticas se explican de manera inadecuada. El presente artículo presenta una optimización del algoritmo DCN, identificada con el nombre Opti-DCN, al cual se le introducen tres mejoras. Primero, suministra una condición matemática para la preservación topológica. Segundo, las nuevas ecuaciones de transformación que satisfacen esa condición se deducen de la matemática, lo cual simultáneamente optimiza el coeficiente de elasticidad global, parámetro clave que impacta fuertemente la tasa de convergencia del algoritmo lámina de caucho y la integridad topológica de los cartogramas que genere. Por último, el nuevo algoritmo simplifica la manera de generar fuerzas transformadoras en el DCN y mejora su eficiencia de transformación geométrica. La comparación permite ver que el Opti-DCN es significativamente más rápido para converger en cartogramas de igual densidad y puede eliminar errores

*This work is supported in part by the National Aeronautics and Space Administration New Investigator Program in Earth-Sun System Science (NNX06AE85G) and the University of Minnesota. The author gratefully acknowledges the efforts of the editor and three anonymous reviewers for their valuable comments and constructive suggestions. As always, the author takes full responsibility for any remaining errors or omissions.

The Professional Geographer, 65(1) 2013, pages 16–30 © Copyright 2013 by Association of American Geographers.
Initial submission, January 2011; revised submissions, April and June 2011; final acceptance, June 2011.
Published by Taylor & Francis Group, LLC.

topológicos de manera matemática y práctica. **Palabras clave: optimización algorítmica, cartograma de área continua, algoritmo de lámina de caucho.**

Cartograms are transformed maps in which areas or distances represent values having little or partial relevance to the original map geometries. After more than forty years of development, geographers, along with researchers in other fields, have achieved remarkable breakthroughs in computer cartograms (Tobler 2004). Not only have new forms of cartograms, such as circular cartograms and value-by-alpha maps, been introduced but, more noticeably, efficient cartogram algorithms have been developed and numerous cartograms have been widely disseminated to the public (Dorling 1994, 1996; Roth, Woodruff, and Johnson 2002). Cartograms produced with the Gastner–Newman algorithm, for instance, drew much attention in academia and in the 2004 and 2008 U.S. presidential elections (Gastner and Newman 2004; Gastner, Shalizi, and Newman 2005; Webb 2006). A dedicated Web site (www.worldmapper.org) publishes nearly 700 high-quality cartograms using this algorithm (Dorling, Barford, and Newman 2006; Dorling 2007).

Continuous area cartogram algorithms can be technically classified into three categories: cartographic and computational geometry methods, math-physical process simulation, and artificial intelligence. Despite advancements evidenced by a variety of cartogram algorithms, they have various limitations in efficiency, accuracy, and conceptual and computational simplicity, as well as cartographic and aesthetic qualities. For instance, the ultimate, absolute equal-density map produced by the Gastner–Newman diffusion algorithm would resemble circles in theory. This "ballooning" effect, however, negatively impacts map readers perceiving cartograms (Kocmoud 1997; House and Kocmoud 1998; H. Sun and Li 2010). It is therefore necessary to pursue new cartogram algorithms and to improve existing ones to create cartograms having different and better features.

This article reexamines one of the earliest computer algorithms for continuous area cartograms developed by Dougenik in the late 1970s and published by Dougenik, Chrisman, and Niemeyer (DCN) later in *The Professional Geographer* (Dougenik, Chrisman, and Niemeyer 1985). This DCN algorithm simulates a rubber sheet and gradually transforms a map to equal-density status. It calculates force vectors pointing from polygon vertices to their desired positions and generates a vector field to transform space. Although the DCN has difficulty in transforming complex maps composed of polygons with diverse shapes, sizes, and statistical values, it retains much popularity due to its simplicity and ability to preserve shapes (Tobler 2004; Henriques, BaÇão, and Lobo 2009). Two of the three continuous area cartogram programs written in Arcscripts™ and publicly available on the Internet use the DCN algorithm, the other being Gastner and Newman's diffusion algorithm. That said, the DCN algorithm cannot guarantee topological integrity and its mathematical properties are not well explained and thus not well utilized to optimize the algorithm (Gastner and Newman 2004; Tobler 2004). This article optimizes the DCN algorithm by providing a more formal mathematical foundation, adopting a new and uniform force generation equation, and proposing a new mathematically deduced parameter—the global elasticity coefficient—that rescales the transforming forces in a way that ensures topological integrity and optimizes convergence rate. The purpose of the optimized rubber-sheet algorithm (named Opti-DCN) is to make the DCN mathematically rigorous, algorithmically more efficient, and possibly open for further improvement. To simplify the description, in this article, cartograms are continuous area cartograms unless otherwise noted, and the statistical values used to determine the desired sizes of areas in cartograms are called the *population*.

The rest of the article is organized as follows. The next section briefly reviews major cartogram construction algorithms developed in the last forty years. I then lay out a mathematical foundation for the Opti-DCN algorithm and propose new equations and endogenous parameter-setting mechanisms. After that, I present the U.S. continental population cartograms produced by the Opti-DCN algorithm and compare them with the original rubber-sheet algorithm. The article concludes with discussion of the value of the optimized algorithm and its mathematical foundation, as well as some directions for further improvement.

Continuous Area Cartogram Algorithms

From a technical viewpoint, the nearly fifty years of continuous area cartogram algorithm development roughly diverged into three methodological families: cartographic and computational geometry, math-physical process simulation, and artificial intelligence. Although each family of methods has its unique characteristics and advantages, all algorithms come with certain limitations. This article only briefly summarizes some features of these three categories. For more general and detailed reviews on cartogram and related algorithms, see Dorling (1996), Kocmoud (1997), Skupin and Fabrikant (2003), Tobler (2004), and Henriques, BaÇão, and Lobo (2009).

As its name implies, a cartogram is initially conceptualized and created by cartographers (Friis 1974). One of the earliest examples is a rectangular cartogram developed by Raisz (1934, 1936). In the early stage of cartogram development, cartographers and artists manually plotted "equal-density" maps or "perceptual" maps based on cartographic principles. With the availability of new information technology, especially geographic information systems, computerized programs captured the core procedure of these manual cartogram creation methods and made new improvements based on computational geometry (to continue the case of rectangular cartograms, see Heilmann et al. 2004; Keim, North, and Panse 2004; Florisson, van Kreveld, and Speckmann 2005; van Kreveld and Speckmann 2007). In recent years, researchers have utilized more general and more advanced computational geometry concepts and measures, such as medial axis and triangulation, to advance cartogram algorithms (House and Kocmoud 1998; Keim, Panse, and North 2005; Inoue and Shimizu 2006).

Simulating mathematical and physical processes facilitated developing the earliest and lately some of the best computer cartograms. Preserving topology or, in mathematical terms, maintaining topological equivalence between the original and transformed spaces, is one fundamental requirement for a cartogram algorithm. In mathematical studies of topology, a rubber sheet is most often used to illustrate abstract concepts and the discipline of topology is therefore informally referred to as "rubber-sheet geometry" (Johnson and Glenn 1960; Sauvy and Sauvy 1974). This metaphor also helped develop cartogram algorithms. As a pioneer on computer cartograms, Tobler developed rubber-map and pseudo-cartograms by simulating rubber sheet on a regular grid (Tobler 1973, 1986). The DCN algorithm also simulates similar systems, where each vertex generates forces with distance decay. These forces form a vector field and transform the space. This type of process usually cannot achieve the ideal shapes with single iteration; instead, it requires multiple steps to approach a near-equilibrium status, in which forces at any locations are no longer large enough to mobilize any points.

In addition to these approximate force-field methods, more complicated yet mathematically accurate algorithms are developed for cartograms. Gusein-Zade and Tikunov (1993), for example, used line integral to deduce the mathematical equation for cartogram transformation. Their method divides the space into small surface elements and assigns them new sizes according to the population. After integrating these changes from one side to the other, the equation can calculate all coordinates in one step. Besides pure mathematics, intuitive physical processes can also help produce cartograms. Diffusion is one such physical process. In an ideal diffusion, fluid moves from high-density to low-density areas until it reaches an equal-density status, which is exactly the goal of cartograms. Diffusion theory has been well studied in physics and its application to cartograms turns out to be a success in terms of achieving desired areas and preserving topology (Gastner and Newman 2004; Gastner, Shalizi, and Newman 2005; Dorling 2007).

Artificial intelligence is another direction of cartogram algorithm development. Cellular automata (CA) and self-organizing maps (SOMs) have been applied to generate cartograms. Dorling (1996) used a CA machine to make cartograms. The space is divided into a regular grid and each grid cell adjusts itself to approach the desired size without violating topological relationships with neighbors. Accompanying the advancement of general artificial intelligence technologies, new methods are developed for cartogram applications, of which SOMs, a neural network technology, are the most noticeable (Henriques, BaÇão, and Lobo 2009). SOM essentially maps an input space to an output space and preserves "topology" because close entities are kept close throughout the mapping process. For cartograms, SOM uses points in a regular grid as input. The training set is composed of points that are randomly generated within each polygon according to its density. SOM trains the input points so that they mimic the density-controlled points. Through a reverse labeling process, these input points build the skeleton of the cartogram, with which any points in the space can be interpolated.

As many researchers in this field have observed, there is no "silver bullet" solution for cartograms to date. It seems that no one algorithm is universally better than others (Henriques, BaÇão, and Lobo 2009). The cartogram algorithms already discussed have advantages and disadvantages in simplicity, efficiency, effectiveness, cartographic quality, topology preservation, shape preservation, and many other aspects (Kocmoud 1997; Henriques, BaÇão, and Lobo 2009). Although developing new algorithms is critically important, improving existing ones is also beneficial, not only to cartograms but to relevant spatio-temporal transformation and geovisualization research as well (e.g., Ahmed and Miller 2007). The DCN algorithm is easy to understand, simple to implement, and can produce quality and readable cartograms. Improving the efficiency of the DCN algorithm and eliminating its problem of topological errors would greatly expand its applicability to cartograms and to geovisualization as well.

Opti-DCN Algorithm

Mathematical definitions of the cartogram problem have been provided by previous studies (for recent ones, see Henriques, BaÇão, and Lobo 2009; Lekien and Leonard 2009; H. Sun and Li 2010). Unlike those definitions where the basic transformation units are polygons, the DCN algorithm can actually be defined by point displacements. Define a point set of vertices of all original polygons $S^o_{pnt} = \{P^o_i(x^o_i, y^o_i), i = 1, 2, \ldots, n\}$. According to the relationship between the area of a polygon and its population, define a point set for the transformed polygons $S^t_{pnt} = \{P^t_i(x^t_i, y^t_i), i = 1, 2, \ldots, n\}$, where P^t_i is the new position of the original point P^o_i, so that $Area(Polygon^t_m) = \frac{Population_m}{Density}$. The DCN algorithm is essential to find a mathematical transformation $T : S^o_{pnt} \rightarrow S^t_{pnt}$.

Like many multidimensional data visualization algorithms, the DCN algorithm simulates a system of springs that exert forces generated by the vertices of polygons to transform a rubber sheet. In the original DCN algorithm, the forces are calculated using the coordinates of polygon vertices and their new coordinates in an assumed situation where polygons are independently and linearly transformed according to their population. The magnitudes of the forces are specified using a continuous, differentiable function with distance decay and their directions are parallel to the line from the polygon centroids to the vertices. In each step, the forces mobilize points in the space toward desired locations. As points gradually approach their new position, the forces become weaker and will finally reach an insignificant level. In that status, the transformed areas will be very close to desired sizes. In the whole process, a force reduction factor is calculated using the size error to make vertices move toward the right direction and to avoid polygon overlapping possibly caused by "too strong forces" (Dougenik, Chrisman, and Niemeyer 1985).

The optimization presented here keeps the iterative logic of DCN but extends this rubber-sheet algorithm in three aspects. First, with such a mathematical definition of the problem, a condition that can preserve topology, namely, avoid any line intersecting and polygon overlapping, is proposed. Second, a set of force generation functions that satisfy the condition is designed. Finally, using these functions, a simplification to the original DCN force generation mechanism is proposed, which improves the efficiency of the DCN algorithm and retains its effectiveness.

Preserving Topology

One of the major concerns in cartograms is preserving topology, which is also an unresolved problem of the DCN algorithm, especially in the mathematical sense (Kocmoud

1997; Tobler 2004). A good cartogram algorithm should be able to prevent line segments from intersecting and make polygons free from overlapping. In the viewpoint of topology, the key to preserving topological relationships is keeping vertices in order. Mathematically, bicontinuous transformation can guarantee topological equivalence; that is, bicontinuity is a sufficient condition for topological equivalence (Sauvy and Sauvy 1974; Binmore 1980). It is easy to design a continuous transformation that smoothly maps one space to another as the original DCN algorithm does; however, it is difficult to guarantee that the reverse transformation exists and is continuous, too; that is, make the continuous transformation also bicontinuous. The core of cartogram transformation, therefore, is to choose or design a continuous transformation function with an inverse function that exists and is also continuous. Monotonically increasing or decreasing continuous functions must have inverse functions. A continuous composite function composed of differentiable elementary functions like arithmetic, exponential, logarithm, power, and four elementary operations must be continuous also if the inverse function does exist.

Based on this mathematical reasoning, we could design a continuous, differentiable transformation function using arithmetic, exponential, logarithm, power, and other elementary functions. As long as the inverse of this function exists, it should be continuous, too. This transformation, therefore, would be bicontinuous and could preserve topology. Because a monotonically increasing continuous function must have inverse function, a transformation function that is continuous and monotonically increasing would certainly lead to bicontinuity and then to topological equivalence. As a result, the essential function of designing cartogram algorithms is to find such continuous, differentiable, and invertible transformation functions.

To simplify the discussion, a one-dimensional case is used to explain the problem (Figure 1). $\forall x_i^o, x_j^o \in S^o$, $S^o \subseteq \Re^1$, and $x_i^o < x_j^o$ define a continuous, differentiable transformation $T: S^o \to S^t, S^t \in \Re^1$ and $T(x_i^o) = x_i^t, T(x_j^o) = x_j^t$, where o and t denote the original and transformed spaces, respectively. Then $x_i^t < x_j^t \Rightarrow T$ is monotonically increasing $\Leftrightarrow T' > 0 \Rightarrow T^{-1}$ exists $\Rightarrow T$ is bi-continuous, and $\Rightarrow S^o$ and S^t are topologically equivalent (for a formal definition and proof of this proposition, see Binmore 1980). This implies that, in a one-dimensional space, topology will be preserved if the order of any pair of the transformed points is the same as that of the original points (Figure 1A). In other words, a continuous and monotonically increasing function must have an inverse function. If it is a composite function composed of elementary functions, its inverse function must be continuous, too. This bicontinuous function could then guarantee topological equivalence. If the transformation function is not monotonically increasing, at least two points in the original space would be mapped to the same point in the transformed space, which implies that the space "collapses" at that point, thus violating topological equivalence (Figure 1B).

Extend this formula to a two-dimensional space. Define a continuous, differentiable transformation $T: S^o \to S^t, S^o \in \Re^2, S^t \in \Re^2$. It can be written in matrix format as $\begin{pmatrix} x^t \\ y^t \end{pmatrix} = T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} T_x(x,y) \\ T_y(x,y) \end{pmatrix}$. Its Jacobian matrix is $\mathcal{J}_T = \begin{pmatrix} \frac{\partial T_x}{\partial x} & \frac{\partial T_x}{\partial y} \\ \frac{\partial T_y}{\partial x} & \frac{\partial T_y}{\partial y} \end{pmatrix}$. It has been pointed out by many researchers that the determinant of the Jacobian matrix, $det\,\mathcal{J}_T = \frac{\partial T_x}{\partial x} \cdot \frac{\partial T_y}{\partial y} - \frac{\partial T_x}{\partial y} \cdot \frac{\partial T_y}{\partial x}$, should be equal to the ratio of global average density to local density for the purpose of producing ideal cartograms (Gusein-Zade and Tikunov 1993; Gastner and Newman 2004; Lekien and Leonard 2009). The relationship between the Jacobian and topological equivalence, however, has never been discussed in detail for cartogram algorithms. According to the inverse function theorem, if the determinant of a Jacobian matrix, $det\,\mathcal{J}_T$, is nonzero, then the transformation T has a local inverse function and therefore is locally invertible. If the determinant is greater than zero, the transformed space also has the same orientation as the original one (Nijenhuis 1974; Binmore 1980). So, for such a continuous, differentiable transformation T, $det\,\mathcal{J}_T > 0 \Rightarrow T^{-1}$ *exists locally* $\Rightarrow S$ and *T(S)* are topological equivalents. This sufficient condition literally means, to any small region around point $P^o(x, y)$ in the original space S, after such a transformation, the region will be continuously projected to another region that

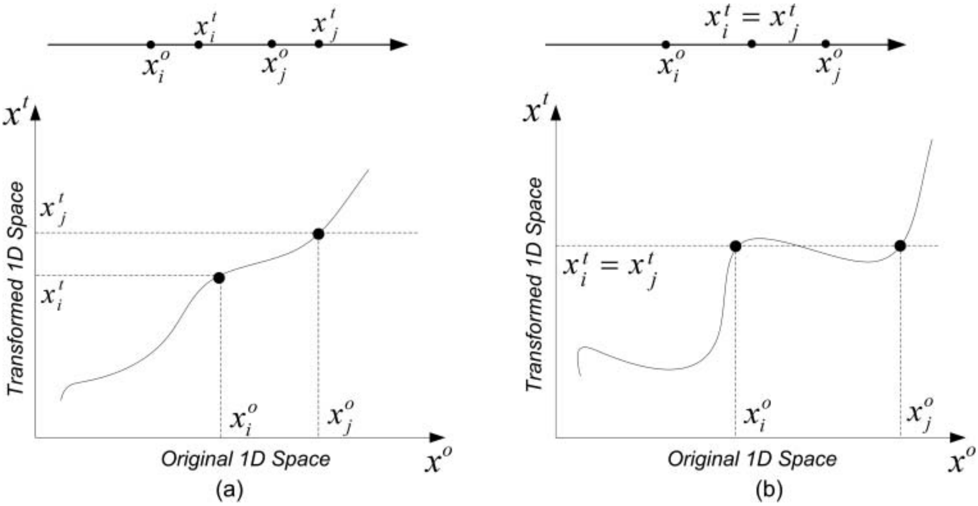

Figure 1 *Monotonic increase, inverse function, and topological equivalence in one-dimensional space. (A) Topological equivalence. (B) Point overlapping.*

has the same orientation. Because the projection is invertible, any point in the transformed region can also be continuously projected back to the original region. This implies a bicontinuous transformation and thus guarantees topological equivalence. Although this condition only applies to local regions, it is still effective when points are evenly distributed in the space.

It must be noted, however, that even with such a sufficient condition, the discrete space of vector polygons cannot completely preserve the mathematical property of topological equivalence as in a continuous space. For example, in a continuous space, the direct line of DE would become a curve to avoid overlapping with AB and BC; in the discrete space, however, DE will intersect with AB and BC (Figure 2A). Appropriate densification of vertices on polygon arcs, therefore, is necessary to preserve topology in a discrete space. If there was a point at F, for instance, the topology would be preserved under the same transformation condition (Figure 2B).

Previous implementation of the DCN algorithm mentioned the value of densification for avoiding polygon overlapping, although without much explanation (Du and Liu 1999). In fact, much topological error produced by the DCN can be effectively eliminated after inserting more points on those long

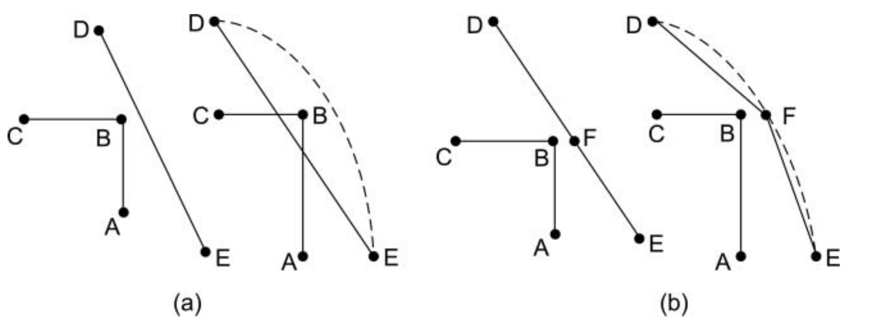

Figure 2 *The value of point densification in topology preservation. (A) Broken topology. (B) Preserved topology.*

polygons arcs. Point densification in DCN and Opti-DCN can be done by imposing an upper limit on the length of the line segment between any adjacent polygon vertices. If the distance between two adjacent vertices is greater than a maximum length, a new point would be inserted between them. In general, one two-hundredth of the map width or height, whichever is lower, is small enough for such a maximum length. In some rare cases where neighboring polygon vertices experience exceptionally strong forces in opposite directions, this maximum value might need to be set even smaller to produce a smoother transformation.

Force Generation Function

With this condition for topology preservation, the key optimization to the DCN algorithm is to design and calibrate a force generation equation that can meet such a condition. Essentially, it is to find a continuous function with a differential or determinant of its Jacobian matrix that is greater than zero. Before doing this, I first lay out the input and output of the force generation function.

To generalize the problem, a set of force-generating point pairs is taken as algorithm inputs. On a two-dimensional Cartesian surface, a set of points $P_i^o(x_i^o, y_i^o)$ is to be transformed to $P_i^t(x_i^t, y_i^t), i = 1, 2, 3, \ldots$. Each pair of points (P_i^o, P_i^t) forms a vector $\vec{P}_i = P_i^t - P_i^o$ and exerts forces on the surface, thus forming a vector field. For a cartogram application, these pairs of points could be the vertex points in the polygons to be transformed; however, they could also be other points as long as the vector force field generated by them can transform the space. The optimized algorithm proposed here essentially specifies the forces to achieve the following: (1) $T(P_i^o)$ is as close to P_i^t as possible, and ideally $T(P_i^o) = P_i^t$; and (2) the T transformed polygons retain their topology; that is, T meets the condition specified in the previous section.

I use two steps to specify a mathematical form for the forces generated by such a point pair. First, using a one-dimensional case, I give some basic properties and constraints of such forces. Then, I give some adjustment and extend it to two-dimensional situation.

In a one-dimensional space, let $\overrightarrow{F_{k \cdot i}(x)}$ be the force generated by vector $\overrightarrow{P_i} = P_i^t - P_i^o$ at $P_k(x)$, where x is the coordinate of the point k. It is obvious that the magnitude of the force should be proportional to the distance between points P_i^o and P_i^t. So, define the magnitude of the force generated by the i^{th} point pair at point k as

$$\left\| \overrightarrow{F_{k \cdot i}(x)} \right\| = \left\| \overrightarrow{P_i} \right\| \cdot f\left(\frac{\| P_x - P_i^o \|}{\left\| \overrightarrow{P_i} \right\|} \right) = d_{ot} \cdot f\left(\frac{d_{ox}}{d_{ot}} \right) = d_{ot} \cdot f(d) \quad (1)$$

where d_{ox} is the distance from the original point P_i^o to $P_k(x)$, d_{ot} is the distance from P_i^o to ${P^t}_i$, and d is a standardized distance. The direction of the force is obviously the same as that of $\overrightarrow{P_i}$. There are four requirements for the properties of $f(d)$, the basic transforming force function (S. Sun and Manson 2007). First, $f(0)=1$; that is, P_i^o should be displaced to ${P^t}_i$ after applying this force. Second, $f\prime(d)$ exists on $[0, +\infty)$; that is, the force is differentiable and thus continuous. Third, $f'(d) < 0$ when $d > 0$; that is, $f(d)$ is monotonically decreasing. This means distance decay of the forces. Finally, $T(d) = d + f(d)$ is monotonically increasing; that is, $T\prime(d) > 0$ and $f\prime(d) > -1$ when $d > 0$. This requirement is the equivalent of the condition specified earlier, which ensures topological integrity because every point being transformed will be kept in the original order. Examples of such functions include $f(d) = e^{-d}$ and $f(d) = \frac{2}{e^d + e^{-d}}$. These two functions reveal little difference. For convenience of discussion and calculation, the simple exponential form is used for optimization.

Now, extend the situation to two-dimensional with a single pair of origin and destination points. The force exerted by the ith point pair $\{P_i^o(x_{i \cdot o}, y_{i \cdot o}) \rightarrow P_i^t(x_{i \cdot t}, y_{i \cdot t})\}$ at any point $P_k(x, y)$ in the original space is

$$\overrightarrow{F_{k \cdot i}(x, y)} = F_{k \cdot i \cdot x}(x, y) \cdot \overrightarrow{\mu_x} + F_{k \cdot i \cdot y}(x, y) \cdot \overrightarrow{\mu_y},$$

where $F_{k \cdot i \cdot x}$ and $F_{k \cdot i \cdot y}$ are forces on the X and Y axis, respectively, and $\overrightarrow{\mu_x}$ and $\overrightarrow{\mu_y}$ are unit vectors (Figure 3). Then

$$F_{k \cdot i \cdot x}(x, y) = d_{i \cdot ot} \cdot f\left(\frac{d_{i \cdot ox}}{d_{i \cdot ot}} \right) \cdot \cos\left(\overrightarrow{P_i^o P_i^t} \right) = d_{i \cdot ot} \cdot f\left(\frac{d_{i \cdot ox}}{d_{i \cdot ot}} \right) \cdot \left(\frac{x_{i \cdot t} - x_{i \cdot o}}{d_{i \cdot ot}} \right)$$

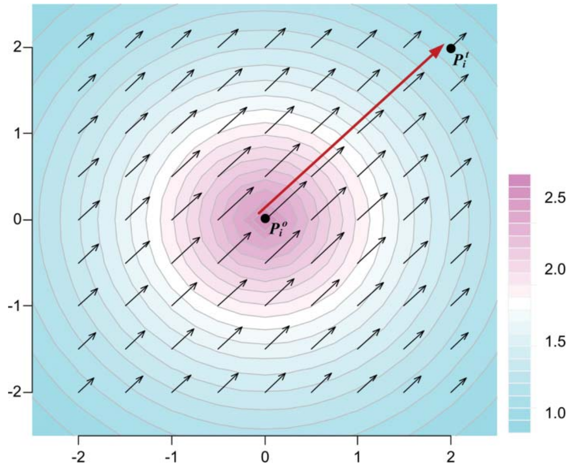

Figure 3 *Force field generated by one point pair $P^o \to p^t$. (Color figure available online.)*

$$= (x_{i\cdot t} - x_{i\cdot o}) \cdot f\left(\frac{d_{i\cdot ox}}{d_{i\cdot ot}}\right)$$
$$= (x_{i\cdot t} - x_{i\cdot o}) \cdot e^{-\frac{d_{i\cdot ox}}{d_{i\cdot ot}}}. \quad (2)$$

The summation of all forces at point $P_k(x,y)$ along the X axis is

$$F_{k\cdot x}(x,y) = \sum_{i=1}^{n} F_{k\cdot i\cdot x}(x,y)$$
$$= \sum_{i=1}^{n} (x_{i\cdot t} - x_{i\cdot o}) \cdot e^{-\frac{d_{i\cdot ox}}{d_{i\cdot ot}}}. \quad (3)$$

Similarly,

$$F_{k\cdot y}(x,y) = \sum_{i=1}^{n} (y_{i\cdot t} - y_{i\cdot o}) \cdot e^{-\frac{d_{i\cdot ox}}{d_{i\cdot ot}}} \quad (4)$$

Then, define $\begin{pmatrix} x^t \\ y^t \end{pmatrix} = T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} T_x(x,y) \\ T_y(x,y) \end{pmatrix} = \begin{pmatrix} x + c_k \cdot F_{k\cdot x}(x,y) \\ y + c_k \cdot F_{k\cdot y}(x,y) \end{pmatrix}$, $0 < c_k < 1$, where c_k is a local elasticity coefficient at the point $P_k(x,y)$. To meet the condition of topological equivalence specified earlier, we must find a value for c_k that makes the determinant of the Jacobian greater than zero; that is, $det\,\mathcal{J}_T = \frac{\partial T_x}{\partial x} \cdot \frac{\partial T_y}{\partial y} - \frac{\partial T_x}{\partial y} \cdot \frac{\partial T_y}{\partial x} > 0$. The four partial derivatives of T are not difficult to calculate. For instance,

$$\frac{\partial T_x}{\partial x} = 1 + c_k \cdot \frac{\partial F_{k\cdot x}(x,y)}{\partial x}$$
$$= 1 + c_k \cdot \sum_{i=1}^{n} (x_{i\cdot t} - x_{i\cdot o}) \cdot e^{-\frac{d_{i\cdot ox}}{d_{i\cdot ot}}} \cdot \frac{\partial d_{i\cdot ox}}{\partial x}$$
$$= 1 + c_k \cdot \sum_{i=1}^{n} (x_{i\cdot t} - x_{i\cdot o}) \cdot e^{-\frac{d_{i\cdot ox}}{d_{i\cdot ot}}} \cdot \frac{x - x_{i\cdot o}}{d_{i\cdot ox}} \quad (5)$$

So, $det\,\mathcal{J}_T = \frac{\partial T_x}{\partial x} \cdot \frac{\partial T_y}{\partial y} - \frac{\partial T_x}{\partial y} \cdot \frac{\partial T_y}{\partial x} = (1 + c_k \cdot \frac{\partial F_{k\cdot x}}{\partial x}) \cdot (1 + c_k \cdot \frac{\partial F_{k\cdot y}}{\partial y}) - c_k \cdot \frac{\partial F_{k\cdot x}}{\partial y} \cdot c_k \cdot \frac{\partial F_{k\cdot y}}{\partial x}$. Then this can be rewritten as a quadratic equation:

$$det\,\mathcal{J}_T = \left(\frac{\partial F_{k\cdot x}}{\partial x} \cdot \frac{\partial F_{k\cdot y}}{\partial y} - \frac{\partial F_{k\cdot x}}{\partial y} \cdot \frac{\partial F_{k\cdot y}}{\partial x}\right) \cdot c_k^2 + \left(\frac{\partial F_{k\cdot x}}{\partial x} + \frac{\partial F_{k\cdot y}}{\partial y}\right) \cdot c_k + 1 \quad (6)$$

Simply solving equation $det\,\mathcal{J}_T = \varepsilon > 0$ with a constraint of $0 < c_k < 1$ can give a value of c_k at point $P_k(x,y)$. If there is no solution for the equation or c_k is negative, make c_k one. Experiments show that 0.05 is large enough for ε to preserve topology.

Alternatively, the elasticity coefficient c_k can be calculated in a simpler but mathematically relaxed way. Topologically, if the order of points along the curve with maximum distortion remains the same after cartogram transformation, the transformed space would preserve topology. Similar to the one-dimensional cases, as long as the transformation function T is monotonically increasing along the steepest direction (i.e., the gradient), T^{-1} will exist locally, and S and $T(S)$ will be topologically equivalent. Although this condition is not as strict as the Jacobian determinant, it also largely maintains the topological integrity of area cartograms in most cases. To make sure T^{-1} exists along X, Y, and the direction of gradient,

$$c_k = \begin{cases} 1. \\ \quad \text{if } \dfrac{\partial F_{k \cdot x}(x,y)}{\partial x} \geq 0 \text{ and } \dfrac{\partial F_{k \cdot y}(x,y)}{\partial y} \geq 0 \\ (\varepsilon - 1)/\min\left(\dfrac{\partial F_{k \cdot x}(x,y)}{\partial x}, \dfrac{\partial F_{k \cdot y}(x,y)}{\partial y}\right). \\ \quad \text{if } \dfrac{\partial F_{k \cdot x}(x,y)}{\partial x} \cdot \dfrac{\partial F_{k \cdot y}(x,y)}{\partial y} < 0 \\ (1-\varepsilon)/\left(\dfrac{\partial F_{k \cdot x}(x,y)}{\partial x}^2 + \dfrac{\partial F_{k \cdot y}(x,y)}{\partial y}^2\right)^{0.5}. \\ \quad \text{if } \dfrac{\partial F_{k \cdot x}(x,y)}{\partial x} < 0 \text{ and } \dfrac{\partial F_{k \cdot y}(x,y)}{\partial y} < 0 \end{cases} \tag{7}$$

For any point to be transformed $P_k(x, y)$, although c_k is not, in the strictest sense, the value required for the existence of an inverse for the global transformation function, it is easy to calculate and actually reduces enough forces to retain topological integrity of cartograms while allowing a near optimal convergence rate.

With such mathematical deductions, the optimized force generation equation can be laid out as follows. Calculate the local elasticity coefficient c_k at $P_k(x, y)$ as in Equation (6) or (7) and the global elasticity coefficient as $c = \min(c_k)$, $k = 1, 2, 3, \ldots$. Then the new coordinates of point $P_k(x, y)$ in one single iteration are $P_k^d(x, y) = P_k(x, y) + \overrightarrow{S_k(x, y)}$. $\overrightarrow{S_k(x, y)} = \overrightarrow{S_{k \cdot x}} + \overrightarrow{S_{k \cdot y}} = c \cdot F_{k \cdot x} \cdot \overrightarrow{\mu_x} + c \cdot F_{k \cdot y} \cdot \overrightarrow{\mu_y}$, where $F_{k \cdot x}$, $F_{k \cdot y}$ are forces along the X, Y dimension at the point $P_k(x, y)$. Or in a matrix form, $\begin{pmatrix} x^t \\ y^t \end{pmatrix} = \begin{pmatrix} x + c \cdot F_{k \cdot x}(x, y) \\ y + c \cdot F_{k \cdot y}(x, y) \end{pmatrix}$.

Simplified Force Generation Mechanism

In the original DCN algorithm, the force-generating point pairs are generated using polygon centroids and vertices. When using

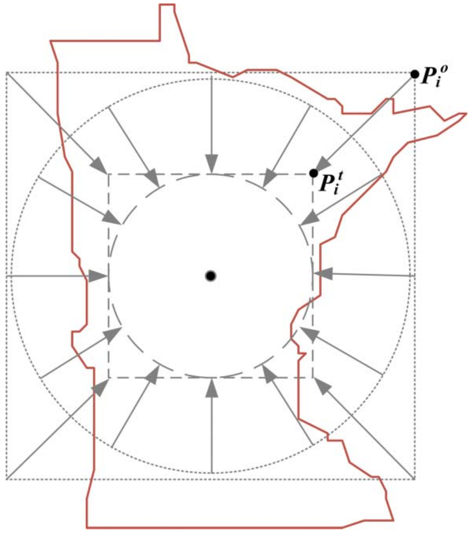

Figure 4 *A simplified force generation mechanism. (Color figure available online.)*

all vertices of polygons, levels of generalization can impact the forces calculated from the same polygons. For example, without any other changes, adding more points to a specific arc in a polygon increases the forces around that arc and causes different cartograms. Moreover, calculating forces generated by all vertices is computationally more costly than using simplified shapes. The application of Opti-DCN shows that using simplified shapes could be as efficient as using all vertices.

In the Opti-DCN algorithm, only circles and squares are used to approximate polygons (Figure 4). For each polygon, Opti-DCN creates a circle with the polygon centroid being its center and the polygon size being its area. A square is then created around the circle. On the circle, sixteen points are evenly sampled. According to the population of the polygon, the intended coordinates of these origin points on the circle and the square can be easily calculated. It is proved that such extreme simplification works well for geometric shapes of U.S. continental states. Of course, such simplification would reduce the transforming forces because, compared with all polygon vertices, the simple circles and squares generally have fewer force-generating point pairs. An exaggeration mechanism therefore is applied to enlarge the

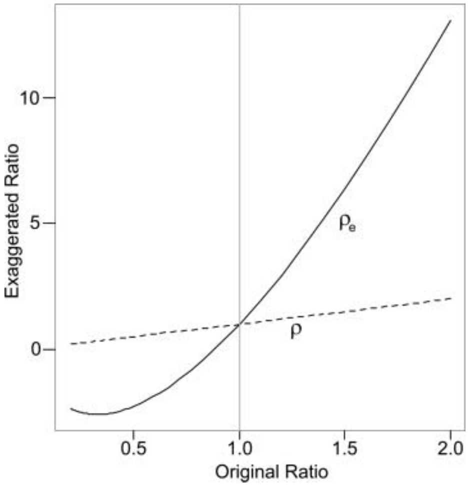

Figure 5 *Force exaggeration.*

transformation and to recover the lost forces. Suppose the ratio between the desired and the original radii of a circle is $\rho = r^d/r^o$. The exaggeration mechanism should, in a continuous manner, make the exaggerated ratio ρ_e smaller than ρ when ρ is less than one but make ρ_e bigger when ρ is greater than one. An example of such exaggeration is $\rho_e = \rho \cdot [1 + \frac{8 \cdot \ln(\rho)}{N_{step:}}]$ (Figure 5). The new radius of the desired circle is $r_e^d = \rho_e \cdot r^o$. This mechanism essentially further shrinks the desired circles and squares that should shrink according to the population of the polygon being approximated; accordingly, it also further enlarges the circles and squares that should be enlarged. The exaggerated circles and squares therefore generate stronger transforming forces, recover some lost forces due to using fewer transforming point pairs, and help prevent Opti-DCN from losing algorithmic convergence efficiency.

It should be noted that the number of sampled points on the circle and the specification of the exaggeration function are flexible. Changing the number of points sampled or the form of the force exaggeration formula would slightly change the convergence rate and the shape of the resultant cartogram, which could help develop variants of the Opti-DCN algorithm. Nevertheless, one advantage of Opti-DCN is its ability to preserve topological relationships under various specifications as long as the determinant of the Jacobian matrix of the transformation function is greater than zero.

In summary, Opti-DCN keeps the algorithmic logic and main steps of the original rubber-sheet algorithm and it optimizes DCN in three critical ways. First, the points used to generate forces are changed from all vertices to points on circles and squares. Second, the force generation equation in Opti-DCN is a scaled negative exponential function. Last, a global elasticity coefficient is calculated from the Jacobian matrix of the transformation function and can largely guarantee topological integrity and optimal convergence rate. Because the differential of an exponential function is itself, calculating the new, optimized global elasticity coefficient adds little computational burden to Opti-DCN.

Results and Comparison

Visually comparing the 2000 U.S. population cartograms produced by the original and optimized DCN algorithms further illustrates the value of the optimization. Because Opti-DCN is essentially a variant of the DCN algorithm, conclusions from previous comparisons between the DCN algorithm and other cartogram algorithms similarly apply to Opti-DCN as well. The main improvement to the DCN algorithm is that Opti–DCN is able to preserve topology with point densification (i.e., adding new vertices if the distance between two neighboring vertices is greater than a predefined criterion; in the case of the continental United States, for example, one two-hundredth of the minimum dimension of the map). In addition, Opti-DCN is algorithmically faster converging to the ideal equal-density shapes.

Specifically, the Opti-DCN algorithm creates almost the same, if not better, cartograms in terms of map aesthetics and readability, despite the fact that it only utilizes circles and squares to approximate the original polygons (Figure 6). Visually inspected, cartograms produced by Opti-DCN better preserve shapes. The square shape of Wyoming, for example, is transformed into a star shape with DCN; with Opti-DCN, Wyoming is largely kept in a square shape with much less distortion at the

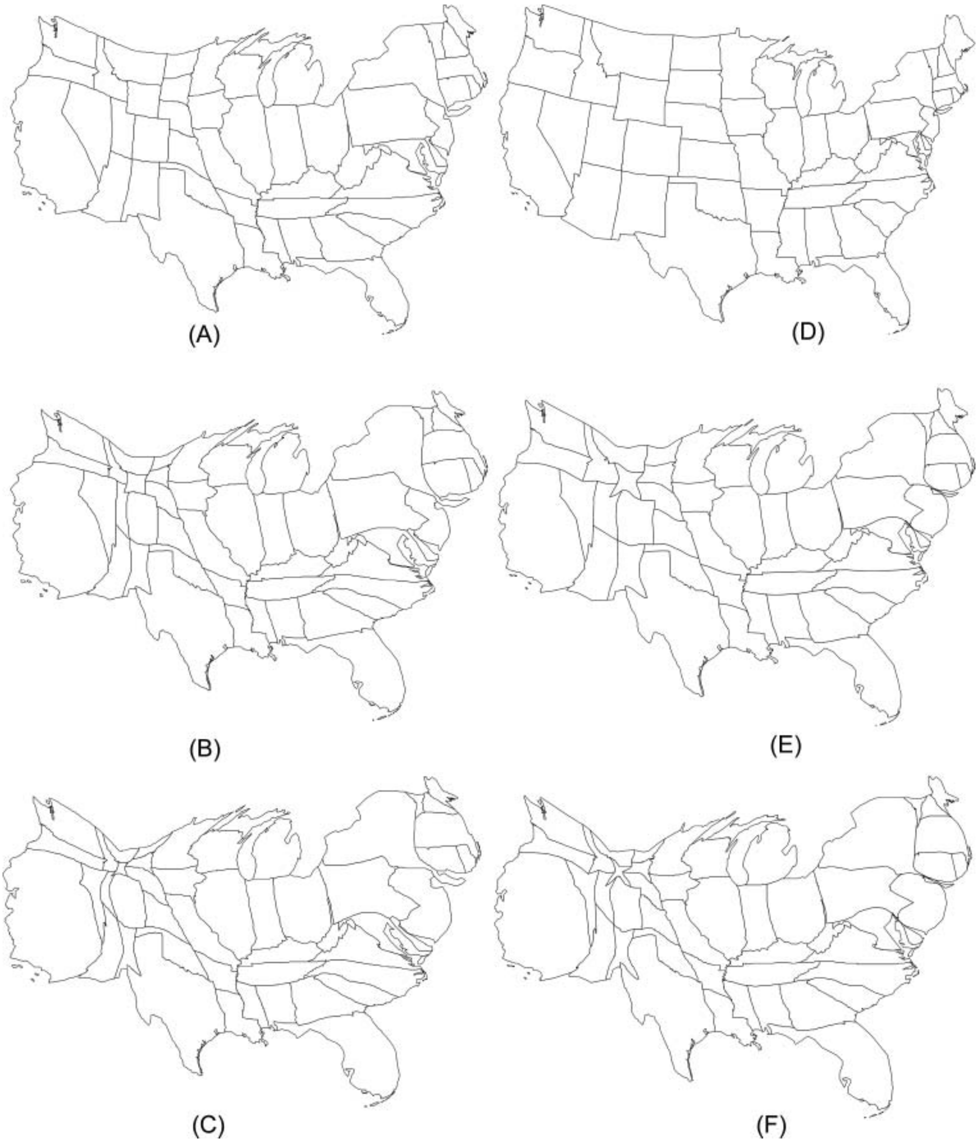

Figure 6 *Algorithmic convergence comparison. Opti-DCN iteration 1 (A), 2 (B), 3 (C); original DCN iteration 1 (D), 4 (E), 7 (F). Note: Opti-DCN only needs one third of the iterations as the original DCN algorithm.*

four corners. Another big difference between shapes generated by DCN and Opti-DCN is evident in viewing Long Island, New York. The Opti-DCN algorithm well preserves its original shape, whereas DCN compresses this area to an unrecognizable narrow belt (Figure 6).

With optimization, the Opti-DCN algorithm works much faster than the original algorithm toward equal-density shapes. To measure the global area errors of cartograms, three measures are calculated using similar metrics as in Keim, North, and Panse (2004) and

Henriques, BaÇão, and Lobo (2009). First, define $e_i = \frac{|S_i^d - S_i^t|}{S_i^d + S_i^t}$ as the size error of the polygon i, in which S_i^t is the area of the ith transformed polygon and S_i^d is the ideal area calculated from population. Then, mean quadratic error is $mqe = \frac{1}{N}\sqrt{\sum_{i=1}^{N} e_i^2}$, weighted mean error is $wme = \frac{1}{S}\sum_{i=1}^{N}(e_i \cdot S_i^t)$, and simple average error is $se = \frac{1}{N}\sum_{i=1}^{N} e_i$, where N is the total number of polygons and S is the total area of all polygons. In general, the Opti-DCN algorithm can achieve the same transformation rate in one step that would have required the original DCN algorithm about three steps. For example, Opti-DCN only needs two steps to get the mean quadratic error to be lower than 2 percent, whereas DCN takes six steps to achieve that (Table 1).

Most important, the Opti-DCN algorithm can theoretically guarantee topological equivalence for continuous space. With point densification using a maximum length of one two-hundredth of the minimum map dimension, the Opti-DCN algorithm does not create any topological errors, even for areas that have complex spatial patterns, such as narrow bays, rivers, islands, and peninsulas. At San Juan Islands of Washington, Potomac River, and Delaware Bay, Opti-DCN well preserves topologies (Figure 7). As some have acknowledged, the DCN algorithm can similarly benefit from point densification, with which DCN can reduce many and even completely avoid all topological errors (Du and Liu 1999).

Discussion and Conclusion

This article presents an optimized algorithm —Opti-DCN—for continuous area cartogram construction, based on the rubber-sheet algorithm developed by Dougenik, Chrisman, and Niemeyer (1985). Opti-DCN gives the original DCN algorithm a more formal mathematical treatment. It specifies a sufficient condition for topological equivalence; that is, a continuous, differentiable, and invertible transformation function. The essence of Opti-DCN is to specify and calibrate such a transformation function through manipulating the partial derivatives of the basic negative exponential force equation. The transformation function specified in Opti-DCN also endogenously optimizes the global elasticity coefficient, which is closely related to topological integrity and convergence rate of the rubber-sheet algorithm. Additionally, Opti-DCN simplifies the transforming force generation mechanism. Using circles and squares to create force-generating point pairs simplifies the algorithm and greatly improves its efficiency without sacrificing its ability to transform space and to preserve shapes.

Comparison with the original DCN algorithm shows that Opti-DCN can completely preserve topology for the case of the continental United States and is algorithmically faster to converge to equal-density maps. More important, Opti-DCN provides a solid mathematical foundation and a flexible framework for the application of rubber-sheet algorithms to geographic transformation. Most elements in this

Table 1 *Transformation efficiency and global area error*

	Opti-DCN algorithm			DCN algorithm		
Steps	**Mean quadratic error**	**Weighted mean error**	**Simple mean error**	**Mean quadratic error**	**Weighted mean error**	**Simple mean error**
0	0.071	0.375	0.416	0.071	0.375	0.416
1	0.040	0.191	0.226	0.062	0.305	0.357
2	0.019	0.060	0.095	0.052	0.234	0.293
3	0.014	0.032	0.052	0.042	0.169	0.229
4	0.015	0.036	0.058	0.033	0.115	0.169
5	0.009	0.024	0.037	0.024	0.074	0.118
6	0.009	0.021	0.033	0.017	0.047	0.079

Note: DCN = Dougenik, Chrisman, and Niemeyer.

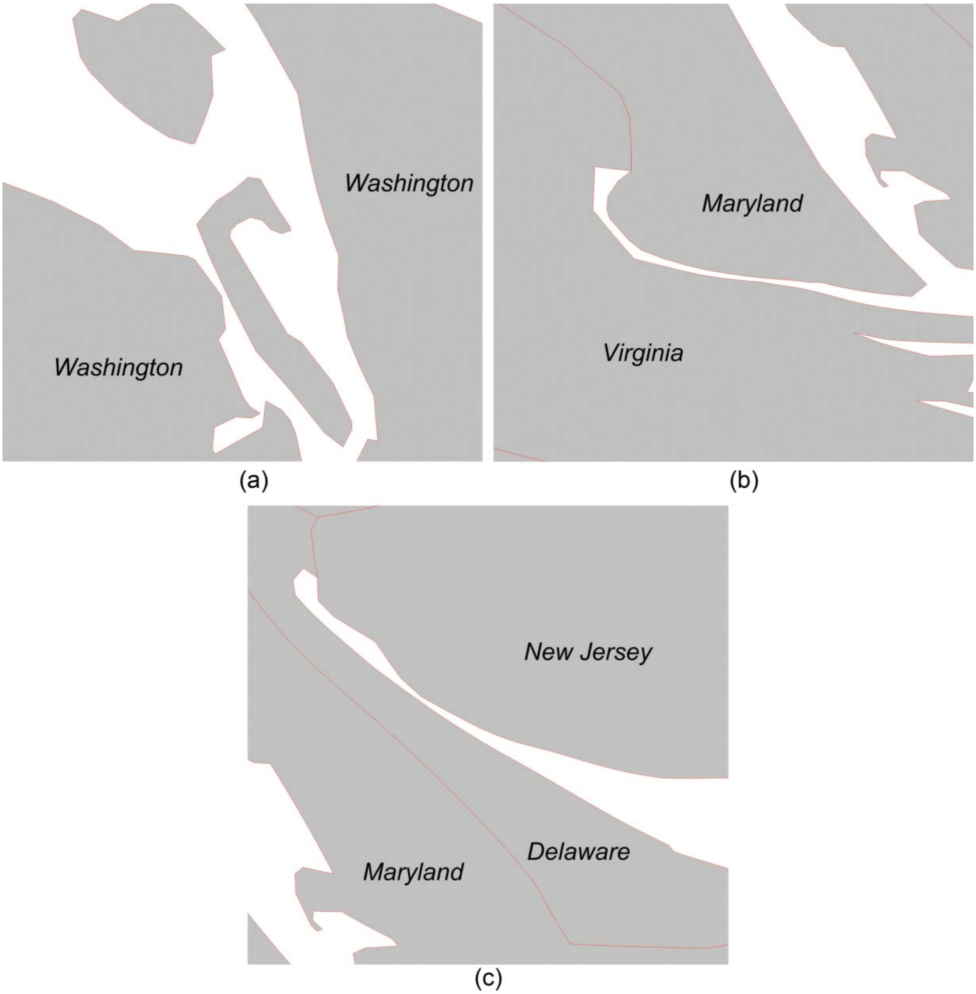

Figure 7 *Topology preservation with Opti-DCN. (A) San Juan Islands of Washington, (B) Potomac River, and (C) Delaware Bay. (Color figure available online.)*

particular algorithm such as the approximate shapes, point densification, force exaggeration and cooling, and even the force formula itself, could be altered to achieve alternative algorithmic features. Opti-DCN well illustrates that as long as the sufficient condition of topological equivalence is mathematically met within the framework laid out in this article, the method of generating transforming forces and specifying a rubber-sheet algorithm could be rather flexible.

With such a mathematical framework, further improvement to Opti-DCN is possible. Transformation functions with better mathematical properties can be designed to make the calculation of forces and their partial derivatives even more efficient. The simple circles and squares used in Opti-DCN can be repositioned to make force-generating points as close to polygon vertices as possible. They could also be replaced by geometries that better describe the original polygons using automatic cartographic generalization, which could further improve its ability to preserve shapes. ■

Literature Cited

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

Binmore, K. G. 1980. *Topological ideas. Vol. 2. The foundations of analysis: A straightforward introduction.* New York: Cambridge University Press.

Dorling, D. 1994. Cartograms for visualizing human geography. In *Visualization in geographical information systems*, ed. H. H. Hearnshaw and D. J. Unwin, 85–102. Chichester, UK: Wiley.

———. 1996. *Area cartograms: Their use and creation, concepts and techniques in modern geography*. Norwich, UK: School of Environmental Sciences, University of East Anglia.

———. 2007. Worldmapper: The human anatomy of a small planet. *PLoS Medicine* 4 (1): 13.

Dorling, D., A. Barford, and M. Newman. 2006. Worldmapper: The world as you've never seen it before. *IEEE Transactions on Visualization and Computer Graphics* 12 (5): 757–64.

Dougenik, J. A., N. R. Chrisman, and D. R. Niemeyer. 1985. An algorithm to construct continuous area cartograms. *The Professional Geographer* 37 (1): 75–81.

Du, C., and L. Liu. 1999. Constructing contiguous area cartogram using Arcview Avenue. Paper presented at the Proceedings of Geoinformatics' 99 Conference, San Diego, CA.

Florisson, S., M. van Kreveld, and B. Speckmann. 2005. Rectangular cartograms: Construction & animation. Paper presented at the Proceedings of the Twenty-First Annual Symposium on Computational Geometry, Pisa, Italy.

Friis, H. R. 1974. Statistical cartography in the United States prior to 1870 and the role of Joseph C. G. Kennedy and the U.S. Census Office. *Cartography and Geographic Information Science* 1 (2): 131–57.

Gastner, M. T., and M. E. J. Newman. 2004. Diffusion-based method for producing density-equalizing maps. *Proceedings of the National Academy of Sciences, USA* 101 (20): 7499–7504.

Gastner, M. T., C. R. Shalizi, and M. E. J. Newman. 2005. Maps and cartograms of the 2004 U.S. presidential election results. *Advances in Complex Systems* 8 (1): 117–23.

Gusein-Zade, S. M., and V. S. Tikunov. 1993. A new technique for constructing continuous cartograms. *Cartography and Geographic Information Science* 20 (3): 167–73.

Heilmann, R., D. A. Keim, C. Panse, and M. Sips. 2004. Recmap: Rectangular map approximations. Paper presented at the IEEE Symposium on Information Visualization, Austin, TX.

Henriques, R., F. BaÇão, and V. Lobo. 2009. Carto-Som: Cartogram creation using self-organizing maps. *International Journal of Geographical Information Science* 23 (4): 483–511.

House, D. H., and C. J. Kocmoud. 1998. Continuous cartogram construction. Paper presented at IEEE Visualization'98, Research Triangle Park, NC.

Inoue, R., and E. Shimizu. 2006. A new algorithm for continuous area cartogram construction with triangulation of regions and restriction on bearing changes of edges. *Cartography and Geographic Information Science* 33 (2): 115–25.

Johnson, D. A., and W. H. Glenn. 1960. *Topology: The rubber-sheet geometry*. Georgetown, CT: Webster House.

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

Keim, D. A., C. Panse, and S. C. North. 2005. Medial-axis-based cartograms. *Computer Graphics and Applications, IEEE* 25 (3): 60–68.

Kocmoud, C. J. 1997. *Constructing continuous cartograms: A constraint-based approach.* Master's thesis, Texas A&M University, College Station, TX.

Lekien, F., and N. E. Leonard. 2009. Nonuniform coverage and cartograms. *SIAM Journal on Control and Optimization* 48 (1): 351–72.

Nijenhuis, A. 1974. Strong derivatives and inverse mappings. *The American Mathematical Monthly* 81 (9): 969–80.

Raisz, E. 1934. The rectangular statistical cartogram. *Geographical Review* 24 (2): 292–96.

———. 1936. Rectangular statistical cartograms of the world. *Journal of Geography* 35 (1): 8–10.

Roth, R. E., A. W. Woodruff, and Z. F. Johnson. 2002. Value-by-alpha maps: An alternative technique to the cartogram. *The Cartographic Journal* 47 (2): 130–40.

Sauvy, J., and S. Sauvy. 1974. *The child's discovery of space: From hopscotch to mazes: An introduction to intuitive topology.* London: Penguin Education.

Skupin, A., and S. I. Fabrikant. 2003. Spatialization methods: A cartographic research agenda for non-geographic information visualization. *Cartography and Geographic Information Science* 30 (2): 99–119.

Sun, H., and Z. H. Li. 2010. Effectiveness of cartogram for the representation of spatial data. *Cartographic Journal* 47 (1): 12–21.

Sun, S., and S. M. Manson. 2007. Distance cartogram: Concept, approximate algorithm, and examples. Paper presented at the 2007 International Workshop on Geographic Information Science, Beijing.

Tobler, W. R. 1973. A continuous transformation useful for districting. *Annals of the New York Academy of Sciences* 219 (1): 215–20.

———. 1986. Pseudo-cartograms. *Cartography and Geographic Information Science* 13 (1): 43–50.

———. 2004. Thirty five years of computer cartograms. *Annals of the Association of American Geographers* 94 (1): 58–73.

van Kreveld, M., and B. Speckmann. 2007. On rectangular cartograms. *Computational Geometry* 37 (3): 175–87.

Webb, R. 2006. Cartography: A popular perspective. *Nature* 439 (7078): 800.

SHIPENG SUN is a Postdoctoral Fellow in the School of Planning, Faculty of Environment at the University of Waterloo, 200 University Avenue West, Waterloo, ON N2L3G1, Canada. E-mail: s27sun@uwaterloo.ca. His research interests include network analysis, geovisualization, agent-based modeling, intraurban migration, and land use and land cover change.