F4Carto®

A Fast and Flexible Force-based Space Transformation Framework for Cartograms

And A Systematically Improved Carto3F Upgrade

Developed by Shipeng Sun

Introduction

F4Carto®, i.e., Force fo(u)r Carto(gram), is a program for equal-density contiguous area cartogram production. The design goal of F4Carto is fast and flexible. It takes a standard polygon Shapefile as input and exports a series of transformed cartograms. In fact, the program is so fast for cases like US States that most time consumed is actually for data reading and writing, not for shape transformation.

F4Carto is an upgrade to the original Carto3F V 1.0, thus can be viewed as Carto3F V 2.0. The user interface has been greatly simplified by automatically recommending and applying most parameters. And users now can choose a Shapefile and a field for the sizes or "population". Many improvements have also been implemented inside the program, including mitigation to shape deformation, further speed improvement, program stability, and optimization to the internal parameters.

Installation

Use the standard Windows 64x setup program F4CartoInstaller.msi available at http://sunsp.net website to install the program, documentation, and example data. Or just simply download, unzip the zipped programs and run without installation. This program works on Windows 64-bit platforms (7 & 10 verified). It is built on Windows 10 SDK 10.0.17763.0 using Visual Studio Community 2017 (VC++ 14). If your computer does not have the running environment for the program, please run the setup.exe to install prerequisite packages.

The Win32 version programs is available upon request.

Inputs and Outputs

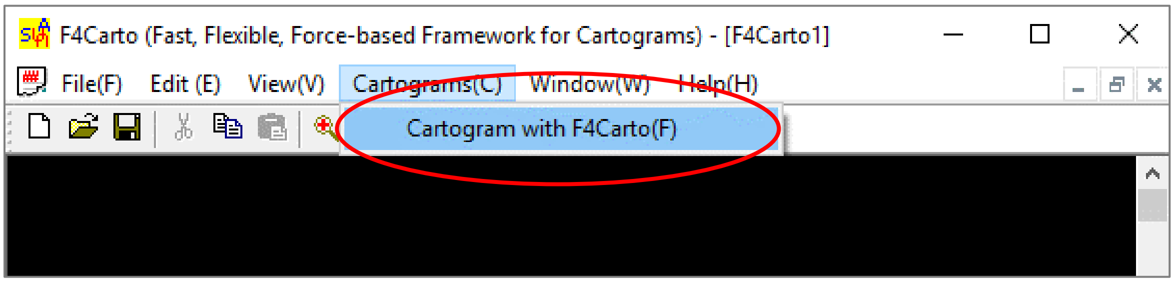
The program requires a well-formatted Shapefile (note that a Shapefile contains at least three files, shp, shx, and dbf). With users' input, the program will take one numeric (integer or double/float) field as "population" or the size of the polygons. All text/string fields will be ignored. Users also need to set a few basic parameters while advanced parameters have been optimized inside the program (More on this in the following section).

The outputs of the program are a series of Shapefiles, named with XXX_0, XXX_1. They will show the intermediate transformations. In many cases, the final step may not be the "best" result. So, it is always desirable to check all those Shapefiles between. Additionally, these files can be combined to develop animated graphs or videos. High-resolution EMF files will also be exported for illustration. These files are vector-format graph/image files.

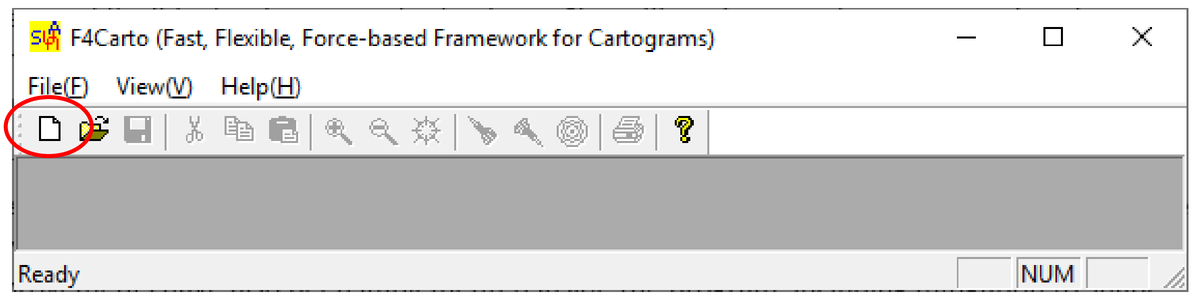
As the program is fast, users are encouraged to play different parameter settings to get the best results.

Program Interface

When the program starts correctly, the following interface is expected. The Cartogram function is in the Cartograms Menu. Click on “Cartogram with F4Carto” to start the process.

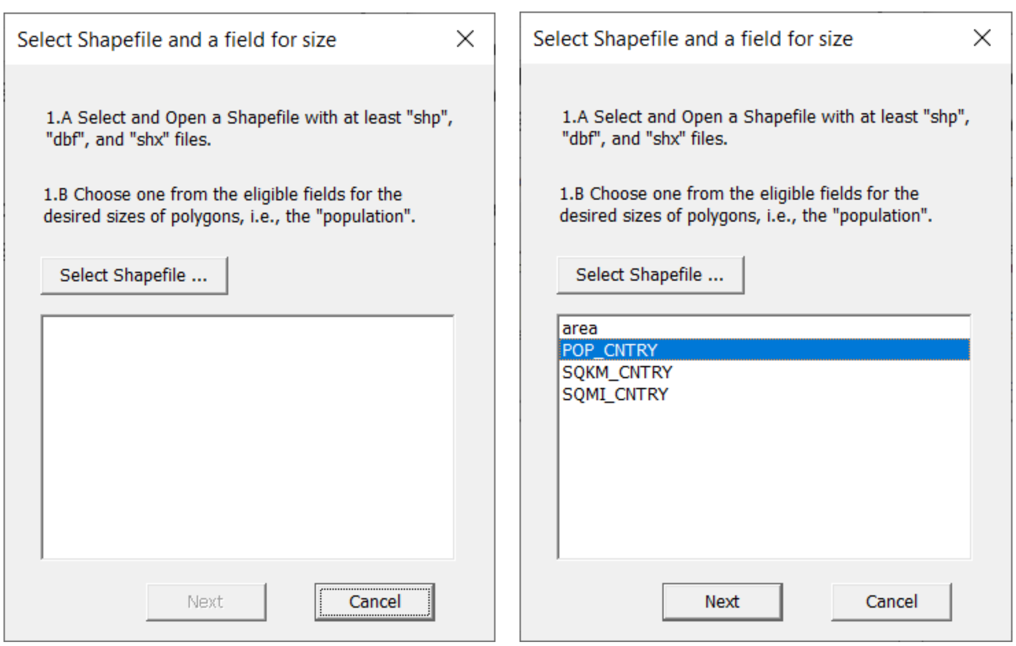


If you don’t see the above interface but the following one. Using File – New, start a new document.



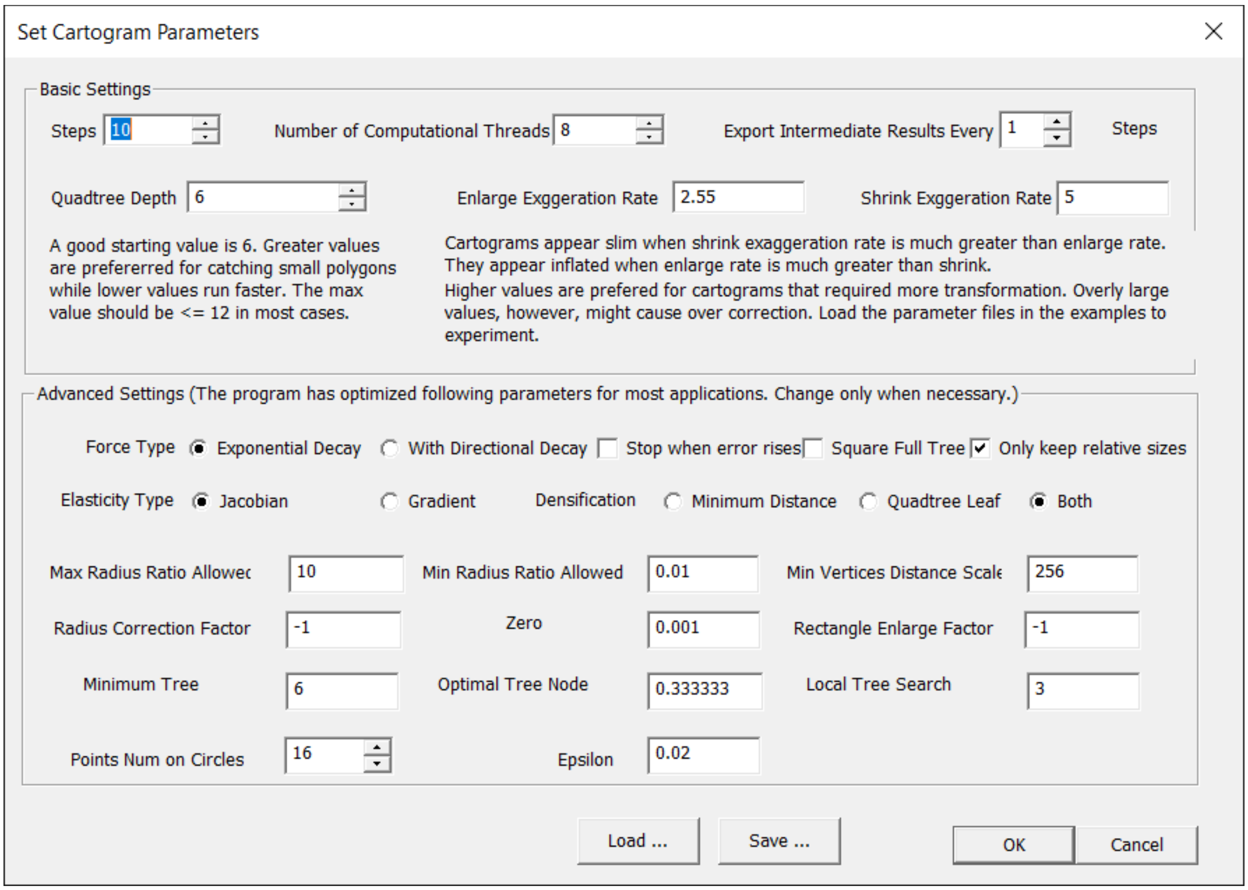
General Process

First, users should select a Shapefile and select a field for the “population”.



Make sure you have the writing permission to the folder where you choose the Shapefile. The program will write all results in the same folder as the Shapefile. Note that the program only lists integer, float/double, i.e., numeric types of fields. Double-click the field or click “Next” for the next step.

Second, set parameters for the program. All parameters can be set using the following interface. Users are strongly recommended to try the installed examples, which contain the Shapefile and associated parameter file. These examples will give you a good sense how different parameters might impact the outputs.



Parameters can be loaded from an ASCII (text) file or saved to such a file by using the “Load …” and “Save …” buttons. Example parameter-files with “cartoparam” extensions are available with the program installation. Users can just load them in. F4Carto has optimized most parameters and generally users just need to set the basic parameters of steps, quadtree depth, enlarge and shrink exaggeration rates. Detailed explanation of parameters is available below.

Parameters

Basic Settings

Steps: The maximum iterations allowed. The program will stop once its iteration reaches this number. If the "Stop when error rises" is checked on, however, the program may stop early. Due to over-correction, size errors might go up in some situations. Over correction is when, for example, we want to enlarge a polygon from 2 to 4, if the resultant polygon is 4.1 instead of 3.95, that is an over-correction.

Number of Computational Threads: The number of computing threads used to do the calculation. The program will automatically detect the number of cores/threads of the CPU and use all of them by default.

Export Intermediate Results Every [X] Steps: this controls how often the intermediate results will be saved to the hard drive. For exploratory works, one is recommended to examine the maps in every iteration. If the value is bigger than one, say 3, the program will save results for iteration 2, 5, 8, … etc. with 0 being the number one step. The last step will always be saved regardless of the parameter.

Quadtree Depth: The maximum depth of the quadtree (should not go over 12 in most cases). Depending on the details that the program needs to catch, values between 5 and 10 are recommended. Deeper trees can catch more details in the data but also require more processing times. If 0 is used for tree depth, the program will try to figure out an optional depth, which tends to be moderately or significantly bigger than needed. This is to guarantee small polygons being handled nicely but could slow the program down. A good practice is to start the quadtree depth with 6. If the tree needs to catch more details for small polygons, increase the tree depth gradually. If it needs to transform shapes faster, decrease it.

Enlarge Exaggeration Rate & Shrink Exaggeration Rate: The extra exaggeration parameters for enlarging and shrinking shapes. Both should be greater than 1. Greater values will produce stronger enlarging or shrinking forces to transform the shapes. When enlarge parameter is much bigger than the shrink parameter, the resultant map tends to be more rounded or inflated, i.e., the balloon effect. When the shrink parameter is much bigger than the enlarge parameter, the resultant map tends to be thinner and slimmer. If the Shrink exaggeration rate is set 0, the program will try to calculate an optional rate, which aims to maintain the program efficiency and to minimize overall distortion. Maps that need more enlargement or shrinkage, for example 400 times larger and 300 time smaller at maximum, should have higher rates. By contrast, maps only need to enlarge say 10 times at maximum should have lower rates.

Advanced Settings

Setting these parameters requires a good understanding to the algorithm itself as well as its implementation. Please read the publications before changing these settings.

Point number on circles: the number of points on the approximate circles.

Epsilon: the epsilon (the indefinitely small number) value

Force Type: The transforming force type: 0 exponential distance decay (default) 1 with directional decay. Normally, 0 should be used.

Stop when Error goes up: whether to stop the iteration if the error metrics go up. If false, the program is essentially controlled by Steps parameter.

Full Square Tree: when using full quadtree to generate transforming points, the program could use a square coverage or use a rectangle coverage. The rectangle coverage will reduce the number of points.

Only Keep Relative Size: whether the sizes are controlled by relative sizes or absolute sizes. If relative, the ratio between the polygons and "blank" areas is not maintained.

Elasticity Type: method of calculating elasticity coefficient 0 Jacobian 1 gradient. Jacobian is recommended for complete topological equivalence.

Densification Type: 0: using minimum distance; 1: using intersections with quadtree edges and diagonals (grid equivalent), guarantee no-error in theory; 2: both methods

Rectangle Enlarge Factor: The method to generate approximate rectangles, value > 0. If <= 0, no point pairs will be generated from rectangles/squares.

Maximum radius ratio allowed & Minimum radius ratio allowed: All ratios will be scaled according to this max and min ratios in order to avoid extreme values.

Min Vertices Distance Scale: scale used to calculate the maximum distance between vertices, max distance = min(height, width)/this scale.

Radius Correction Factor: radius correction parameter used to tackle the within-polygon force cancellation. -1 means no correction at all, > 0 this will be multiplied to change the radius for smaller than average circles.

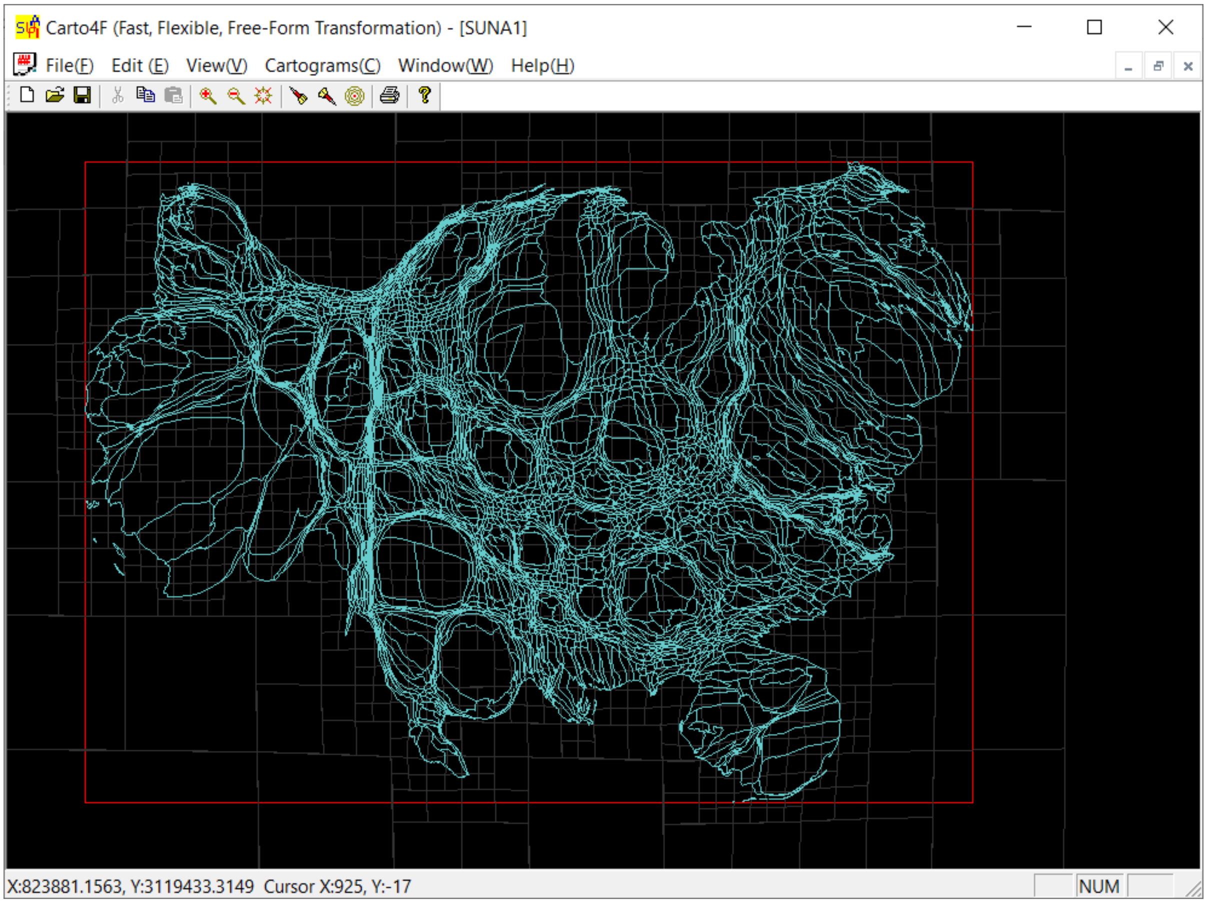
Zero Tolerance: For Shapefile with low precision, if the distance between two points is below 1E-6, they will be treated as one point and causes self-intersection error. This is not our program's problem. It is ESRI's data model problem. In total, the number can have 15 digits, to be safe let it be 12. max(x,y)=z. (int)(log(z)/log(10)) + 1 digits for the integral parts.

The default resolution/precision in ArcGIS is 0.0001 meter (ArcGIS will translate other units into meters). And the default tolerance is 0.001 meter. So, any points that are closer to 0.001 meter will be treated as one point.

Min Tree Node Depth: The minimum tree node depth when using the adaptive tree. All nodes must have at least this depth. For complete topological integrity in any situation, this should be the same as tree depth. If this value is less than the tree depth, the program runs faster but might cause self-intersection and overlapping in some conditions.

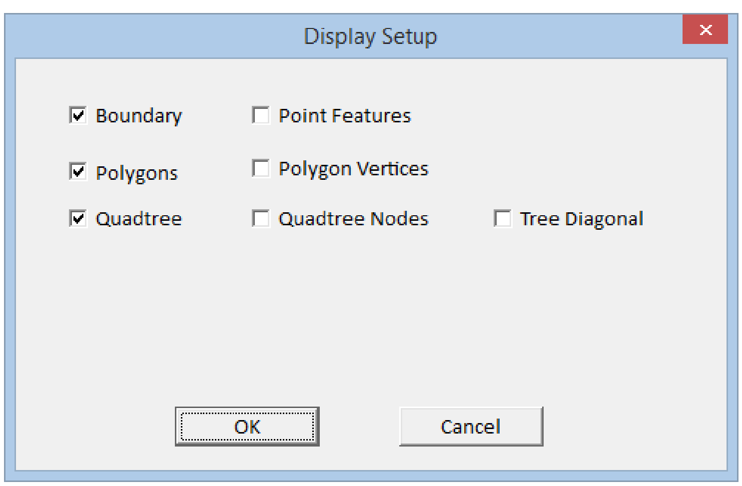
Optimal Tree Node Size Scale: The scale factor used to determine the optional tree node depth. If a polygon is small, its points should have higher depth to capture its geometry. Big polygons do not need fine quadtree nodes. Node size will be roughly min(polygon width, height)*optimalTreeNodeSizeScale*Poly_Area/MBR_Area

Local Search Depth: Because vertices in polygons are 'clustered', so local search should be tried when finding the tree node that contains a point. This parameter defines how far go up to parent, grandparent nodes.



USA County Cartograms with 2010 Census Population

F4Carto also provides a basic visualization function to review results, particularly the quadtree. In the menu: View – Set Parameters, different display options can be specified.



Version and Control

F4Carto, (Force fo(u)r Carto(gram), or Carto3F V 2.0 for Public Release on Earth Day, April 22, 2019.

Copyright and Contact

 For those who use the program free, please acknowledge using "Cartogram Created with F4Carto (Carto3F V 2.0) Developed by Dr. Shipeng Sun" or similar expression. For those who post cartograms created by Carto3F online, please also add or embed link http://sunsp.net or http://sunsp.net/portfolio.html to your web pages.

Should you have questions, feedbacks, suggestions, or want to use Carto3F for commercial purposes, please contact Dr. Shipeng Sun at sunsp.gis@gmail.com or cartogram@outlook.com.

For academic users, please cite the following publications where and when appropriate and possible.

Sun, Shipeng (2013). A Fast, Free-Form Rubber-Sheet Algorithm for Contiguous Area Cartograms. *International Journal of Geographic Information Science* 27 (3): 567-93.

Sun, Shipeng (2013). An Optimized Rubber-Sheet Algorithm for Continuous Area Cartograms. *The Professional Geographer* 16 (1): 16-30.

Sun, Shipeng (2015). Carto3F Program: A Fast, Free-Form Algorithm Implementation for Area Cartograms. Available at http://sunsp.net.

Thank you for your interest in the F4Carto (Carto3F V 2.0) program and hope you find it useful.