Journal of the Korean Geographical Society, Vol. 43, No. 2, 2008(235~252)

Constructing Area Cartogram Using a GIS Based Circular Cartogram Technique*

YoungHoon Kim**

GIS 기반 원형 카토그램 기법을 이용한 카토그램 제작 방법 연구*

김영훈**

Abstract : Many cartographers have for many years searched for a way to construct cartograms in which the sizes of geographic areas such as states, counties or census tracts are rescaled in proportion to their population or some other socio-economic properties. While many techniques and algorithms for creating cartograms have been proposed, some of them are still extremely complex to generate in a proper manner, and many of them suffer either from this lack of readability or from seamless integration with GIS software. This paper, therefore, presents a simple population cartogram technique based on the Circular Cartogram Algorithm(CCA) by Dorling(1996) to tackle these drawbacks by drawing the areas as simple circles for use as a base map and linking the construction with GIS mapping processes. For an automated approach in the cartogram generation, this paper proposes a close coupling method of ArcView GIS 3.3. package in order for users to access to the cartogram algorithm. Then, they will be available through an interface that the ArcView GIS system allows user-written routines to be accessed easily. The CCA and its coupling architecture ensure to improve the potential applicability of the use of cartograms to census mapping at practical levels. As the cartogram examples, cartograms of population and property types in 2005 Korea census data sets are illustrated in the end, by which viewers can easily identify the residential concentration and their relative ratio in Seoul metropolitan area.

Key Words : cartogram, circular cartogram algorithm, close coupling, census mapping

요약 : 지리정보시스템(Geographic Information System, GIS) 활용이 확대됨에 따라서 다양하고 복잡한 사회경제적 변수의 시각적 전달은 공간 분석과 더불어 중요한 연구 주제라고 할 수 있다. 그러나 시각화 기술 자체의 복잡성과 GIS와의 연계성 부족으로 인해 지도화 및 시각화 기법이 제공하는 여러 장점들이 제대로 전달되고 있지 못하고 있다. 이에 대하여 본 연구에서는 카토그램(cartogram) 기법을 적용하여 다양한 인구 관련 변수의 공간적 관계를 효과적으로 지도화할 수 있는 방법을 논의하고자 한다. 이를 위하여 본 연구에서는 범용 GIS 프로그램에서 카토그램이 쉽게 제작될 수 있는 환경을 제안하고, 일반인이나 GIS 초급자들도 손쉽게 구현할 수 있는 과정을 제시하였다. 또한 카토그램의 시각적 정보 전달 및 활용성 증대을 위해서 본 연구에서는 Dorling (1996)이 개발한 원형 카토그램 알고리즘(Circular Cartogram Algorithm, CCA)과 ArcView GIS 3.3. 프로그램의 내적 결합 방법을 적용하여 CCA 기반의 원형 카토그램이 범용 GIS 프로그램내에서 효과적으로 구현될 수 있는 사례를 제시하였다. 마지막으로 실질적인 카토그램 제작 사례로써 2005년 인구 센서스 자료를 대상으로 인구와 가구 변수의 카토그램 지도를 통한 관련 센서스 변수들과 지리적인 공간 분포 패턴의 시각적 분석의 가능성을 제안하였다.

주요어 : 카토그램, 원형 카토그램 알고리즘, 내적 결합, 센서스 지도화

* This work was funded by Korea National University of Education (RI20070060)

** Assistant Professor, Department of Geography Education, Korea National University of Education, gis@knue.ac.kr

1. Introduction

Visualizing data on a traditional map inherently emphasizes 'regions' with the greatest geographic boundaries such as political or enumeration units (e.g. states, counties or census tracts), while making the presentation of very small boundaries are quite difficult. For example, when large numbers of volumes or significant values are concentrated on small sized regions or areas, the way of the traditional mapping is to plot the data on the regions with some color codes or similar representation methods such as thematic mapping. In such map, however, misperception can also arise when the geographical data and quantitative data are not correlated, and without prior knowledge of the geographic data of the each area, the viewer has no clear indication of the spatial correlation (Gastner and Newman, 2004). A common technique for overcoming these problems is to construct cartogram, in which the areas of map regions are rescaled and constructed proportional to their population, aggregate income or retail sales volumes, rather than their geographic size (Dent, 1999, 207-220), and several methods and techniques for a ways of constructing cartograms have been proposed (Dougenik, *et al.*, 1985; Tobler, 1986; Dorling, 1996; Kreveld and Speckmann, 2007).

As a common cartogram type, area cartogram can be drawn so that areas on the paper represent places in proportion to a specific chosen aspect of these places. In area cartogram, areas are proportion to the number of people in each place for which a particular exaggeration is deliberately chosen. An area cartogram is an appropriate basis for looking how something is distributed spatially across groups of people and it is an adequate mapping approach for analyzing spatial correlation with geographical areas and their inherently imbedded socio-economic data (Dorling 1994; 1995b). It is an illustration of spatial distribution across groups of people, and for this reason, area cartograms can be applied for creating socio-economic atlas of geographical areas (Dent, 1999) such as visualizing disease incidence (Selvin, et at., 1988), census geography (Dorling, 1995a), producing social atlas (Dorling and Thomas, 2004). While various cartograms including population cartograms have visualization potential, the generation of rigorous cartograms has been traditionally perceived as a complex, non-intuitive task so that cartograms are difficult to construct effectively (Selvin *et al.*, 1988, 1992). Recent advances in GIS, however, permit their automation and it would be a great aid to geographers and GIS users when the cartogram construction is integrated within GIS visualization process.

For this motivation, this paper presents an integration system of area cartograms using the Circular Cartogram Algorithm developed by Dorling (1996) with population census for which a tight coupling with ArcView GIS software are proposed. The GIS integration demonstrates the utility of population cartograms to make various social atlases, and the atlas of each spatial unit also illustrates spatial relationship between spatial structure and scale of area units used as observation. As attribute datasets and spatial scale, 2005 population census of Korea (KOSIS) and 'Si-Gun-Gu' and 'Eup-Myeon-Dong' administrative boundary data are used. The social atlas of each of the spatial level shows how many people live in each part of Korea and in what social conditions are related.

As a computer solution, cartograms are effective to visualize population and households (Dorling, 1993, 1995b), this paper, as a research motivation, proposes a close coupling approach with the Circular Cartogram Algorithm using Avenue scripts which is a programming language of ArcView GIS 3.3. It notes that ArcView GIS software permits the use of less expensive GIS mapping tool requiring less training time rather than the more complex and expensive Arc/Info, and it is more available within low computing power compared to ArcGIS. This paper also discusses the seamless processes of the population area cartogram construction, such as how to link the cartogram program and run into the GIS software package effectively. Lastly, the application potentials of the cartograms for the creation of social atlas of census are examined with several cartograms illustration of the Korea census.

2. Circular Cartogram Algorithm

According to Dent (1999), there are two types of cartograms widely used: contiguous and noncontiguous. A contiguous cartogram maintains boundary and orientation relationship, strengthening the link between the cartogram and true geographical space. Thus, the shape of the total study area can be more easily preserved than noncontiguous cartogram so that readers of the cartogram need not mentally supply missing area to complete the total form or outline of the

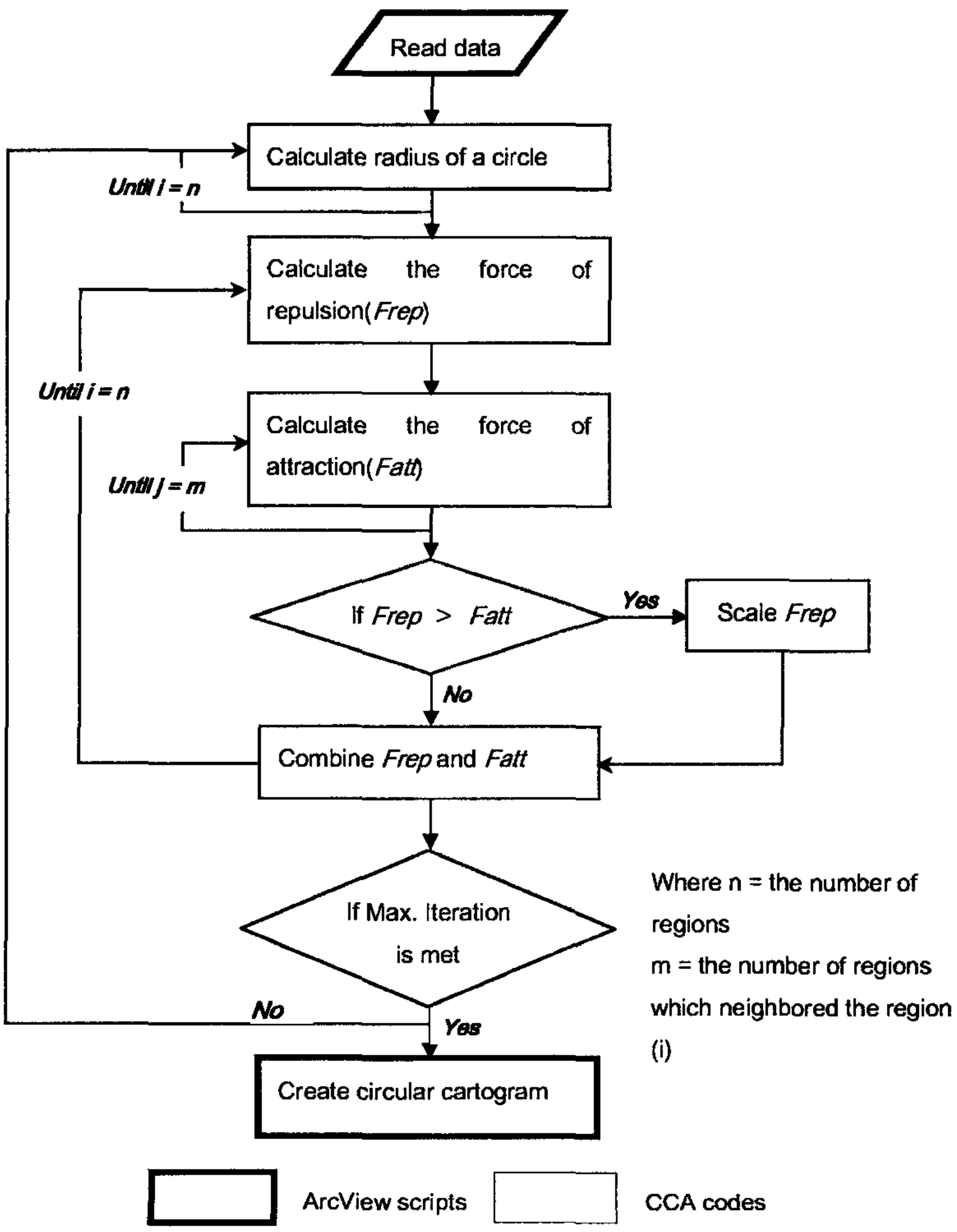

Figure 1. Procedure process of the Circular Cartogram construction

map. For mapping the general characteristics of population factors, however, there are some advantages in adopting noncontiguous cartograms (Dorling, 1994). For example, area units can be represented by desirable shapes such as circles. Moreover, it is easier to measure the area and proportion of each unit in visual manner. These simple shapes including circles can be also extensively manipulated to produce sophisticated pictures of population characteristics. Circle population area based cartograms can be used with noncontiguous constraints.

The idea of using circle as the simplest of all shapes to the creation of population cartograms can be seen in work by Härö (1968 in Dorling, 1996), and Johnston et al., (1988 in Dorling, 1996) and Dorling (1995a) also showed examples of cartograms which define areas as simple shapes for the use of social atlas as a base map. There is a long tradition of mapping with circles as symbols in cartography and the advantages of using this simple shape are well known (Dent, 1972). Figure 1 shows the flowchart of the circular cartogram algorithm which creates cartograms where regions are represented by circles. Note that in this process, CCA codes are used to compute radius and forces of gravity of the regions whilst ArcView Avenue scripts are used to build topological relationship after reading the spatial information from original data sets (e.g. boundary and its census data) and then visualize a circular cartogram after calculating circularities by the CCA codes.

The cartogram algorithm starts by firstly positioning the areal units (e.g. 'Si-Gun-Gu' or 'Eup-Myeon-Dong' boundary in Korea census level) correctly on a map, and then an iterative procedure is applied to create a cartogram with the desired census characteristics (i.e. population

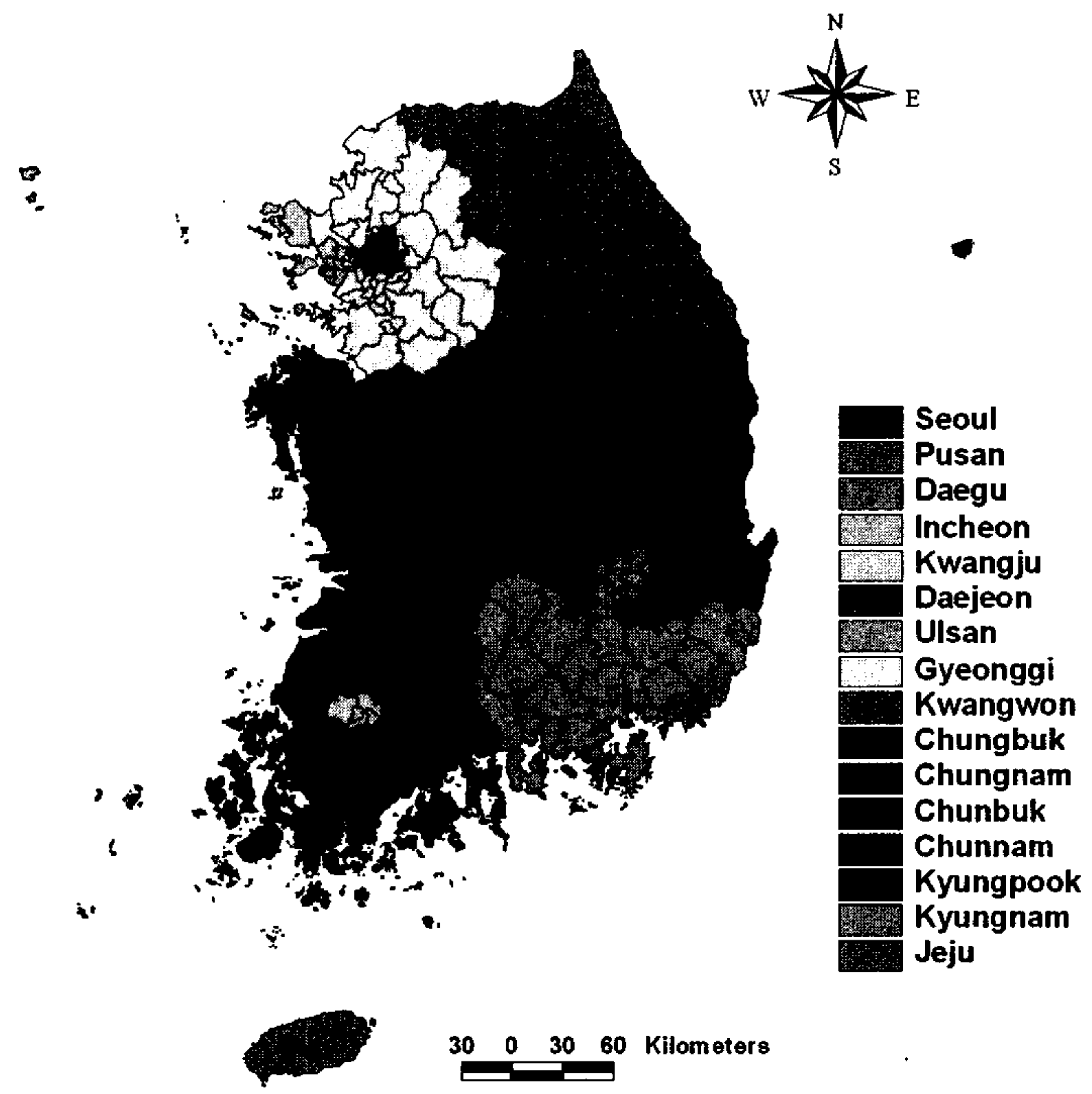

Figure 2. Land map of Si-Gun-Gu boundaries.

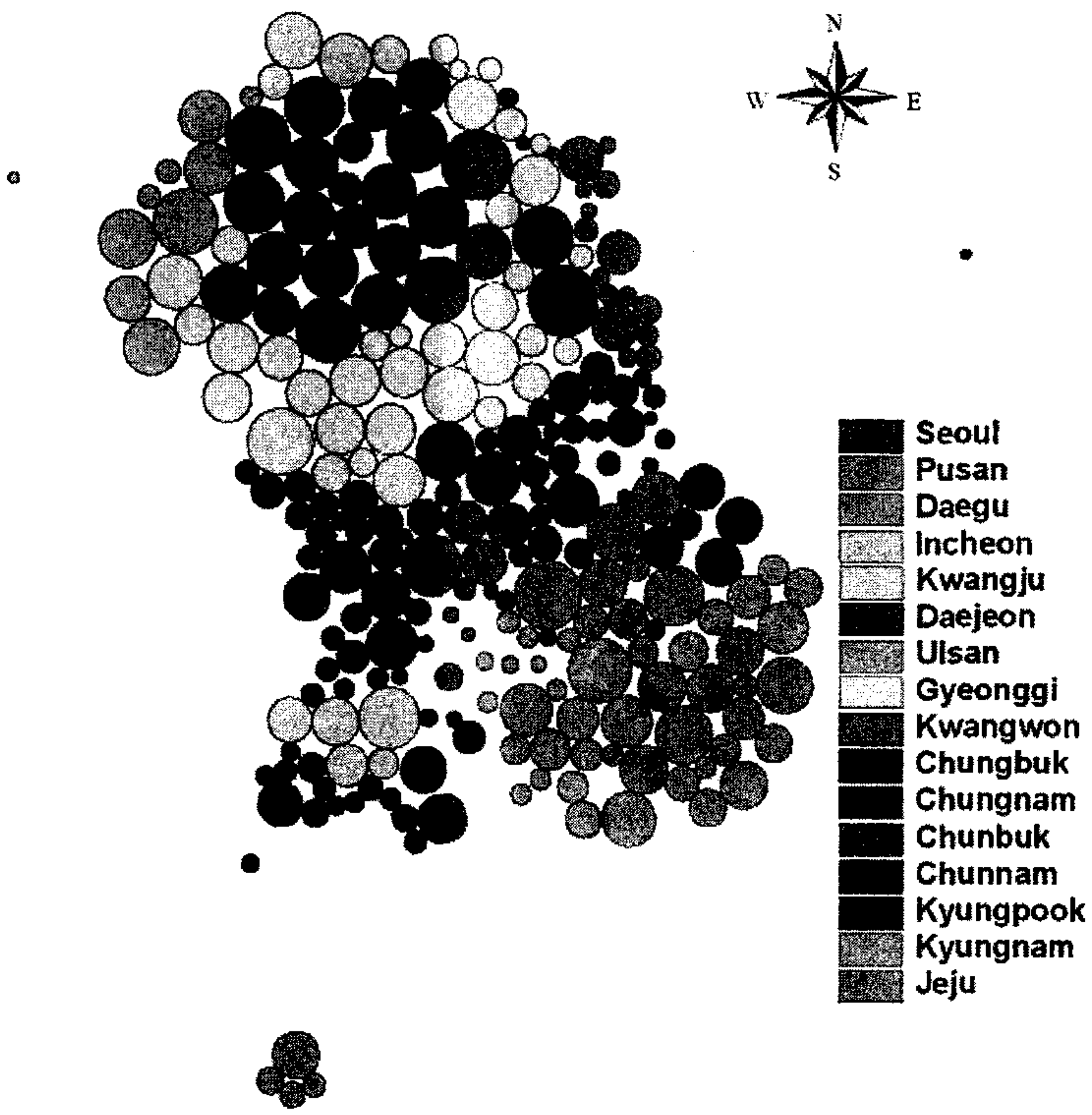

Figure 3. A Circular population cartogram of total population of 2005 on Si-Gun-Gu boundary.

or socio-economic data in each Dong boundary). During the iterations, each region is drawn as a circle with its area in proportion to its population value and is then treated as an object in a gravity model which is repelled by other circles with which it overlaps, but is attracted to circles which were neighboring regions on the original map. Under the cartogram computation routine, the forces of attraction are applied in the directions of their original neighbors in proportion to their length of boundaries between that region and each of its neighboring regions, aiming of preserving the original topology whenever possible. Figures 2 and 3 show a circular cartogram being creating based on Si-Gun-Gu boundary levels. Figure 2 is land map of South Korea Si-Gun-Gu boundaries and Figure 3 shows the completed population cartogram in which each Si-Gun-Gu is drawn as a circle with its area proportional to population.

From figures 2 and 3, it is identical to perceive the advantage of this population cartogram over more sophisticated forms of showing quite clearly how the populations of areas vary across over South Korea, without comparing complex shapes in rural and urban areas. For example, the dominance of the seven major metropolitan city areas in the population structure of South Korea is emphasized in Figure 3, which utilizes the spatial distribution of population as a functional explanation rather than an administrative definition of cities. In turn, the integration structure of the cartogram construction program within ArcView GIS is described in the next chapter.

3. Close coupling with ArcView GIS

For cartogram integration strategies, there are essentially three different strategies for linking cartogram models or techniques with GIS software packages which can lead functional improvement of GIS. They range from loose coupling, to close coupling and full integration architectures (Wise and Haining, 1991). Loose coupling involves importing and exporting data transfer between GIS and external programs (Gatrell, 1987; Waugh, 1986). Data are exported from a GIS and transferred to an external program provided for execution. The cartogram results, then, may be sent back to the GIS for display and further analysis if desired. However, the main weakness of the loose coupling method is that users to be familiar with at least two software packages and it also require extra effort to investigate the data transfer compatibility between GIS and the external programs. Otherwise, close coupling architecture involves writing programs within the GIS environment, avoiding explicit data transfer between software packages and is undertaken by a routine which is called from within the GIS package. There are at least two ways of performing this work, either using a macro language provided by a GIS such as Avenue script of ArcView GIS, or calling a routine written in a standard computer program language such as FORTRAN or C/C++. Although full integration method is an advanced method of access for GIS users, it is still not common used in general GIS environments.

In this paper, thus, close coupling strategy is adopted because of user interface and availability. This architecture can often achieve much the good effect as Avenue script language allows user written routines to be accessed as if it is standard commands of ArcView. Avenue scripts make facilitate data generation and swap routines between ArcView GIS and the CCA program. At here, it is important to note that as Avenue script is a customization language, several Avenue scripts have been developed and proposed to be used for several types of contiguous and non-contiguous cartograms, some of the cartogram sources are downloadable from the Web sites (e.g. Jakel, 1997; Du and Liu, 1999; Huber, 2002). However, the implementation and computing speed of Avenue may not be as fast as the programs written in other languages such as C/C++ or FORTRAN (Zhang and Griffith, 1997), so that reason leads the use of external language coding using C/C++ in cartogram construction process. It is also natural to image that a loose coupling employing Avenue scripts may cause impractical computing performance to construct adequate cartograms for large data sets on Eup-Myeon-Dong levels in Korea census data. Therefore, this paper chooses C computer language to code the CCA and perform the cartogram construction as it is faster to the algorithm computation compared to Avenue scripts. The utilization of Avenue codes is applied for data transfer between the C program and ArcView, and visualizing the cartogram computation runs on the ArcView's view window. Figure 4 illustrates the close coupling structure of the circular cartogram within ArcView GIS. Note that the dotted line in Figure 4 means that interactive interface is performed within ArcView GIS package.

Note that the CCA program has two parts within ArcView GIS; cartogram computation by the Dorling's compiled C and data transfer and visualization by Avenue scripts. The Dorling's C program is compiled and run outside ArcView program to generate the results of the circular cartograms, and the output data are returned to the visualization parts in the ArcView using Avenue scripts. For more details of the CCA cartogram construction, the Avenue

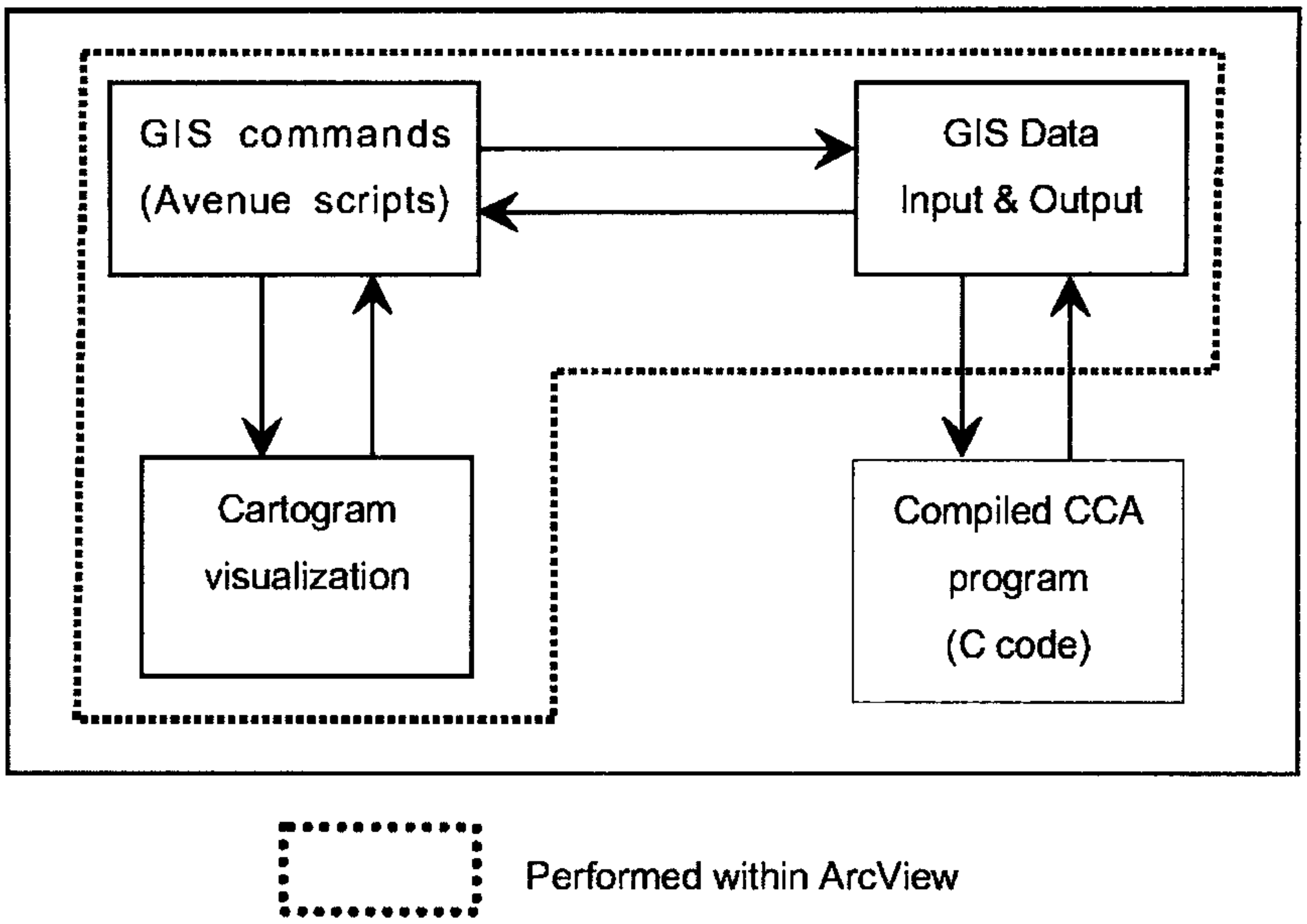

Figure 4. The close coupling structure of the ArcView's circular cartogram

implementation of the script code is listed in the Appendix.

For the data sets in this structure, the roles of the Avenue scripts are generating an input file to be used in the circular cartogram and taking an output file from the CCA code to be used in ArcView cartogram visualization. To facilitate the cartogram interface, the role of Avenue scripts is essential. The following sections describe the main parts of the Avenue scripts that achieve the circular cartogram interface. They consist of several steps to prepare input data and visualization.

- A pop-up menu that asks iteration numbers and working directory
- Create a new field of cartogram IDs in source polygon's attribute table
- Create contiguous polygon table that records all adjacent polygons in each area
- Visualization routines to create circle shapes of population cartogram from circular graphics

4. The creation process of the circular population cartogram

This section articulates the procedures of the cartogram creation within the cartogram interface. For the cartogram interface, this paper attempts to modify the CCA codes to be suitable for the ArcView interface. Before testing the link, the modification contents of the C program is described as follows.

- Increase area numbers (polygon numbers) up to 5000
- Setting iteration numbers by user's choice
- Change input and output format in the program
- Float value format of X and Y co-ordinates
- Integer format of cartogram value to include population and types of houses

Since the procedures of the cartogram interface are started, several menus appear to ask users some steps. However, the procedure is

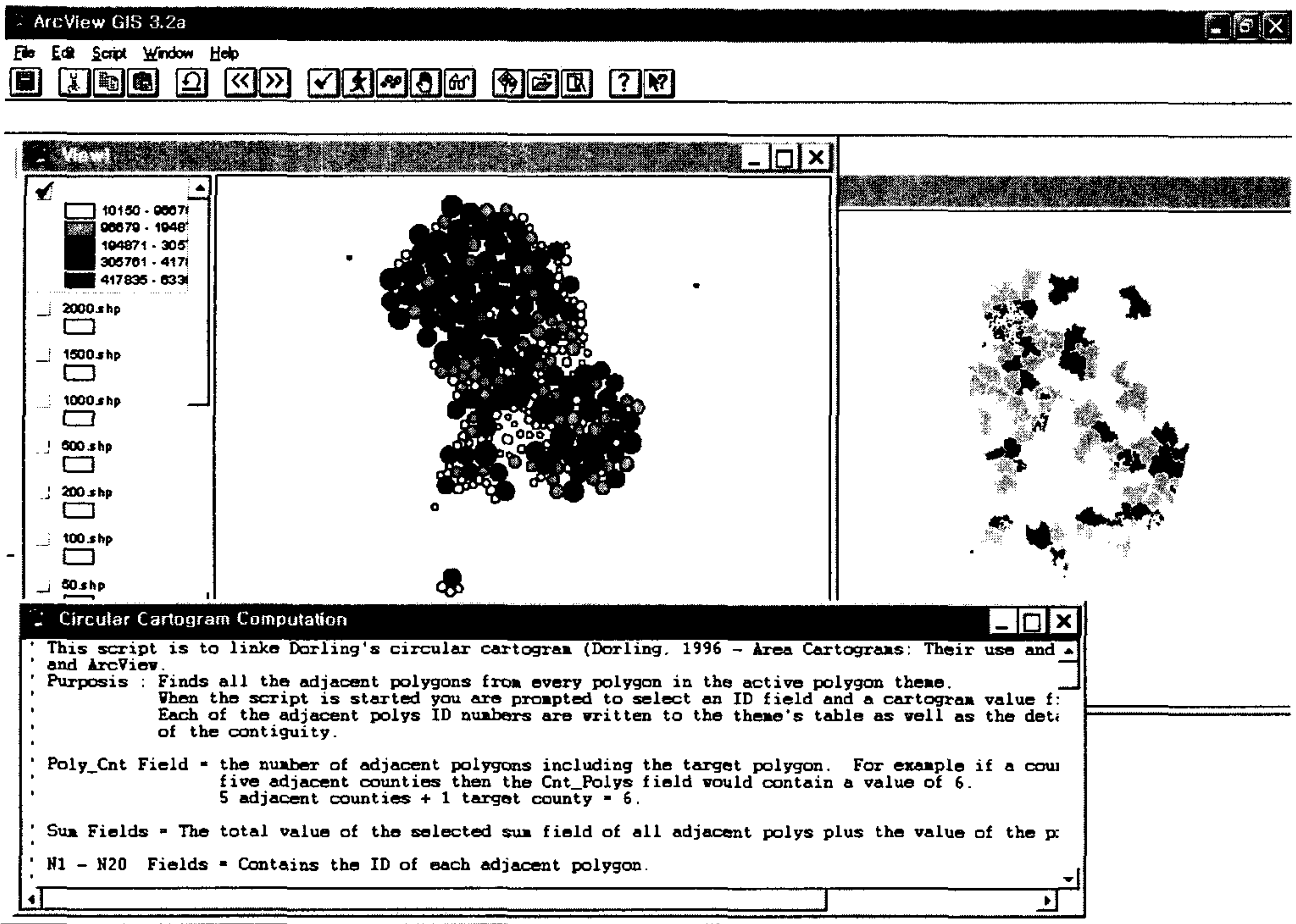

Figure 5. The Cartogram ArcView GIS

straightforwardly designed for the user having no prior knowledge of both cartograms and ArcView GIS package. Figure 5 shows a screenshot of the ArcView cartogram user interface that contains the Avenue script window and the View windows - the cartogram interface illustrates two polygon themes (Si-Gun-Gu based population census data) and its cartogram.

At first, the Avenue script creates a pop-up menu to take iteration numbers and working directory. The iteration number is used for the CCA program to control the cartogram precision and computing time taken. To provide flexible transformation condition, users can specify their preference up to 3500 iterations. The working directory sets the root directory where the cartogram source code and data files exist. As the working directory is set up, all the data sets and ArcView shapefiles are saved in the directory. Figure 6 shows the pop-up menu to type iteration numbers and the directory. Note that the working directory setting is for PCs interface environment.

After taking the parameters, the next step is to select a target ArcView theme. Users can select one theme among the theme lists opened in a View window. After selecting the theme, the Avenue script creates a new cartogram ID field in the theme table, called, 'CartID' from a pop-up menu. The field, if exists already, is deleted and overwrite when users click 'Yes' button on the

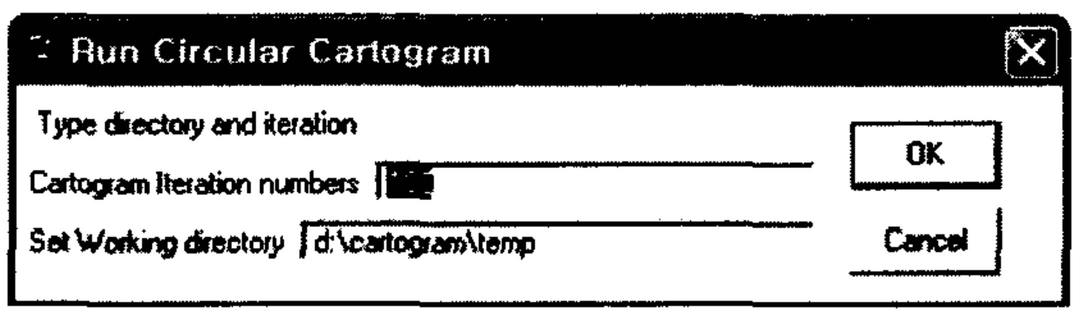

Figure 6. Parameter setting pop-up menu within the ArcView cartogram interface

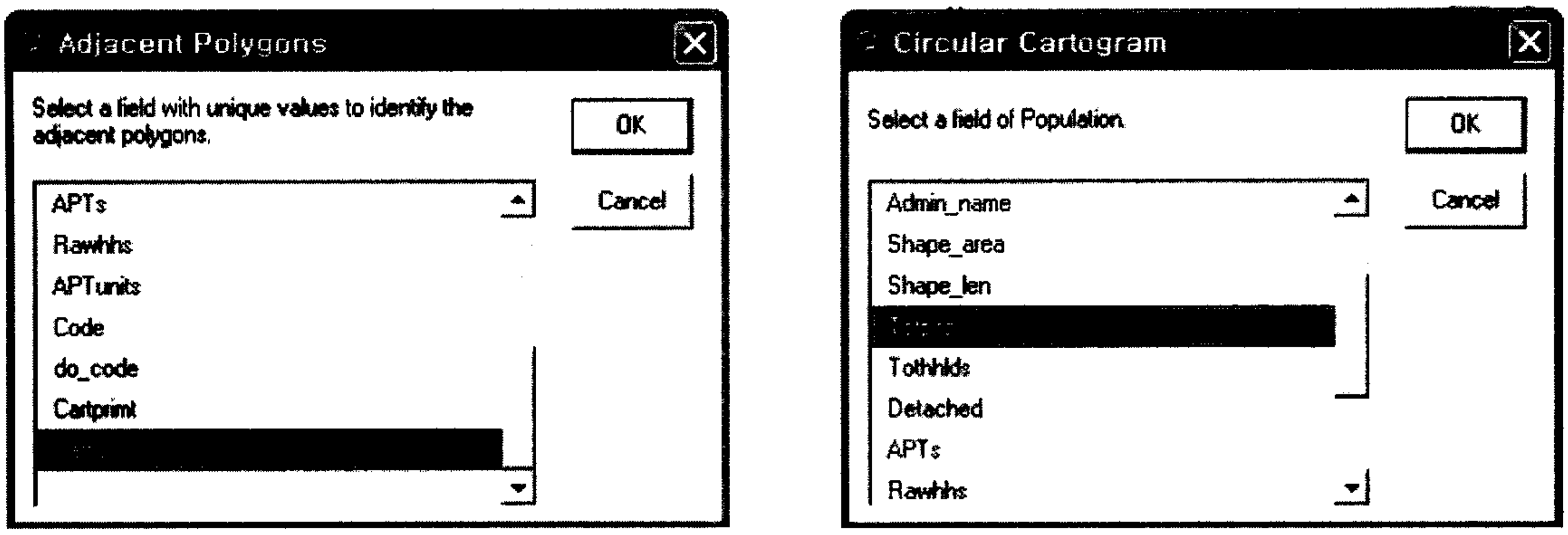

Figure 7. The pop-up menu to select cartogram ID and population value

menu. Note that if 'No' button is selected, the cartogram interface is stopped. All the time users run the script, it is necessary to update the field variable ('CartID') with a serial ID numbers (i.e. 1, 2, 3, ... , n, n = the total number of the polygons of the original regions). This is because boundary IDs of the Korea census codes (i.e. 'Si-Gun-Gu' and 'Eup-Myeon-Dong' ID codes in the Korea census boundary data from Korea National Statistics Office) need to be identical for the circular cartogram algorithm.

After selecting an active theme and updating the cartogram ID field, the Avenue script asks users to select a unique ID field (it should be 'CartID') and a field of numeric value which decides cartogram circle sizes, such as total population, households, number of properties, etc. Figure 7 shows the pop-up menu that selects

```
246 2000
1 170118 197065 455197 7 2 19014.27 4 18881.43 6 18054.40
2 135173 199386 450623 6 1 28699.94 3 21226.46 4 18881.43
3 231484 198361 447941 7 2 19014.27 4 18881.43 14 27997.61
4 325251 203555 449867 6 1 28699.94 2 19014.27 3 21226.46
5 381568 207462 449593 7 4 18881.43 6 18054.40 7 19362.65
6 367596 204427 453467 6 1 28699.94 2 19014.27 4 18881.43
7 440863 208223 454769 5 5 18218.17 6 18054.40 8 31517.79
8 460511 201998 456078 6 1 28699.94 6 18054.40 7 19362.65
9 340765 201230 460535 5 8 31517.79 10 23704.83 11 31977.2
10 352702 202748 462660 4 9 26062.74 11 31977.28 81 45886.
```

where
246 means total numbers of the regions taken for cartogram construction
2000 means iterations for cartogram generation chosen by user
1, 2, ..., 10 means the cartogram ID
170118 means the value of the polygon (area) such as population, unemployment, and so on
197065 and 455197 means the polygon centroid's X and Y co-ordinate value of original region
7 means total number of neighboring polygons of the region 1
2 and 10914.27 mean the cartogram IDs adjacent to the region 1 and the polygon perimeter value

Figure 8. The input file format generated by the Avenue script

```
CartID,X-Coord,Y-Coord,ScaledR,CartValue
1,189919.953125,517457.843750,10466.853690,170118
2,185654.234375,490983.375000,9330.093492,135173
3,191033.328125,470138.312500,12209.602032,231484
4,213943.531250,456507.500000,14472.725984,325251
5,264934.000000,455574.906250,15675.701761,381568
6,210331.046875,489847.375000,15386.024081,367596
7,239042.250000,475228.968750,16849.726082,440863
8,226376.203125,518215.562500,17221.105351,460511
9,202228.281250,539524.937500,14813.869357,340765
```

Figure 9. The output file format generated by the C program

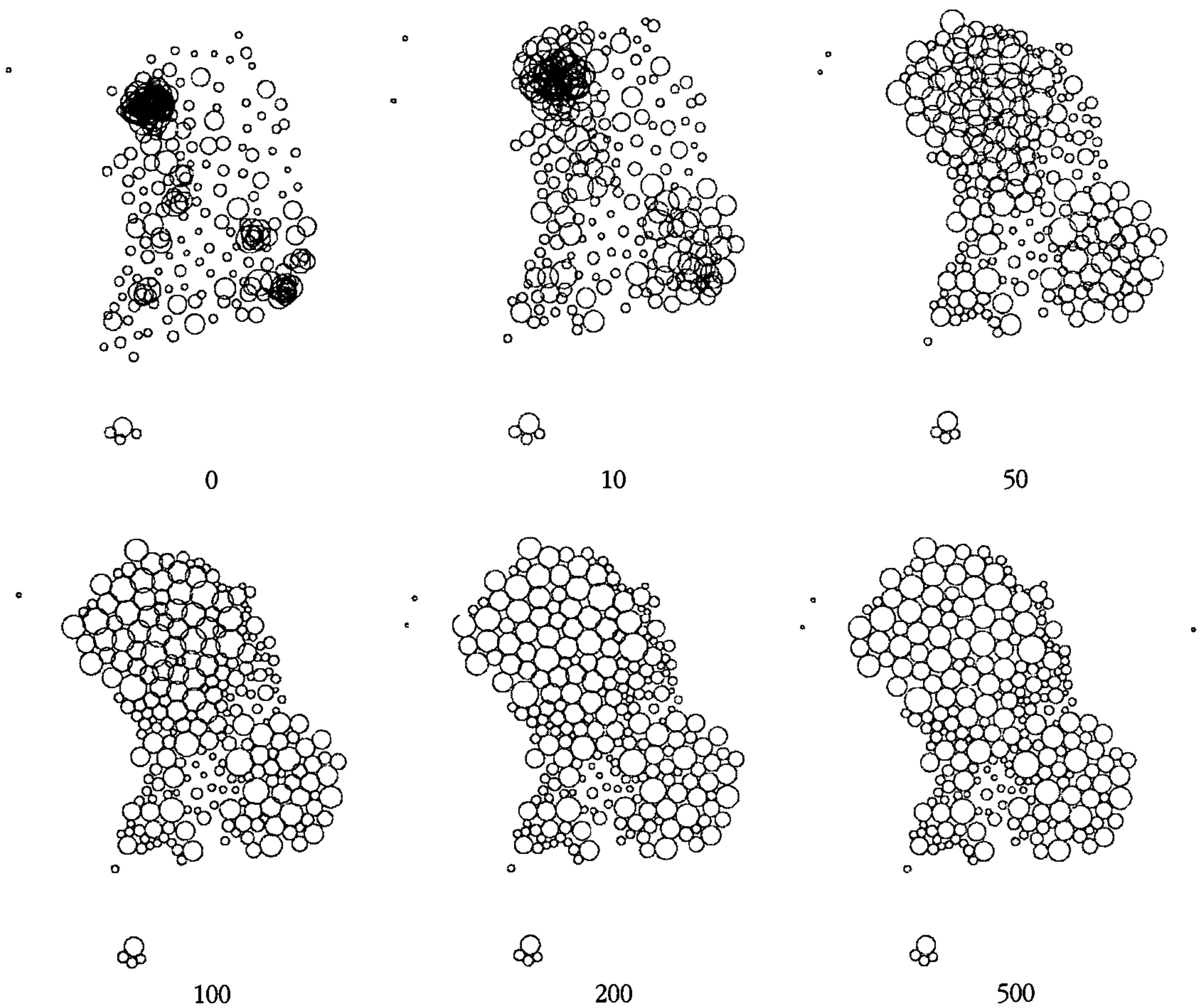

Figure 10. South Korea Si-Gun-Gu boundaries transformed after 500 iterations of the circular cartogram algorithm

the cartogram ID and the population variable.

The next step is to identify adjacent polygons of each area in the map. For detecting polygon contiguity, the Avenue script is applied to record the contiguous cartogram IDs of each polygon and the perimeters of the contiguous polygons into a dummy table that adds the polygons information in the shape file's table temporarily. In ArcView, the attribute table handling script can add or remove additional table fields directly without any losses of tabular data sets. This is one of the most efficient Avenue programming advantages for data handling and retrieving. Using the table handling technique, an input data is created, keeping the contiguity information, called 'dcartout.txt'. Figure 8 shows the input file variables and data format which are exported to the CCA program to create circular cartogram.

Note that in the Avenue program, the contiguity polygon numbers are limited up to 20 for the practice purpose. Thus, the Avenue script assumes the polygons that have over 20 contiguous areas around each original region do not provide precise cartogram results and it may influence the solution quality. However, to aware the overnumbering, a message box popped up to notice the polygon IDs. After exporting the input data by the Avenue script, the C program runs the cartogram routines and generates an ASCII output file, called dcartout.txt, in the directory the user specified in the ArcView menu shown in Figure 7. Figure 9 shows the output file structure, which is imported to the ArcView GIS system to draws circular cartograms. Compared with the input file, the output file provides new sets of X and Y co-ordinates for the polygon and the scaled radius results, which are calculated by the CCA program. The 'CartValue' is the population variable' value selected in the ArcView.

Figure 10 illustrates the process of the population cartogram generations on Si-Gun-Gu boundaries that transformed after 500 iterations of the circular cartogram algorithm as there is no differences after 500 iteration up to 2005.

The output file, then, is sent back to ArcView to draw the cartogram circles on the View window of the ArcView for which Avenue scripts are used. The following figures show the examples of the circular cartograms with the 2005 census data of South Korea.

5. Looking at the cartograms in the census

Here are two pictures; Figure 11 is a typical choropleth map of population density illustrating area population density of 2005 census data in Korea, and this map looks familiar while Figure 12 looks strange. These figures show the geographical relationship between people and land. Figure 11 is a Eup-Myeon-Dong land map, with each boundary shaded to show its density of population. Eup-Myeon-Dongs are the smallest administrative boundary on which census data is publicize in Korea.

The population density of a Eup-Myeon-Dong is the number of people living in that boundary divided by its land area. The key shows that in Korea this ranges from under 2 to over 20,000 people per square kilometer. It is easy to observe on the map (Figure 11) the areas where the density of population is low, however there is a problem with areas of high density. The Eup-Myeon-Dongs are often too small to see on a map of this size. This is a problem if it is analysts or people who are of interests to observe the spatial relationship or patterns in those areas because a great number of people live in those Eup-Myeon-Dong areas. To overcome this problem and search for effective visualization ways, this paper proposes cartogram based mapping in which the area of maps are rescaled in proportion to their population rather than land size.

It is easy to identify that almost half the people of Korea(49.5%) live on only 1.1 % of the land shown in Figure 11, and 49.5 % of the people live on highly populated density areas such as over 10,000 per square kilometer in Eup-Myeon-Dongs shown in Figure 12.

Figure 12 shows is a Eup-Myeon-Dong population cartogram where each Eup-Myeon-

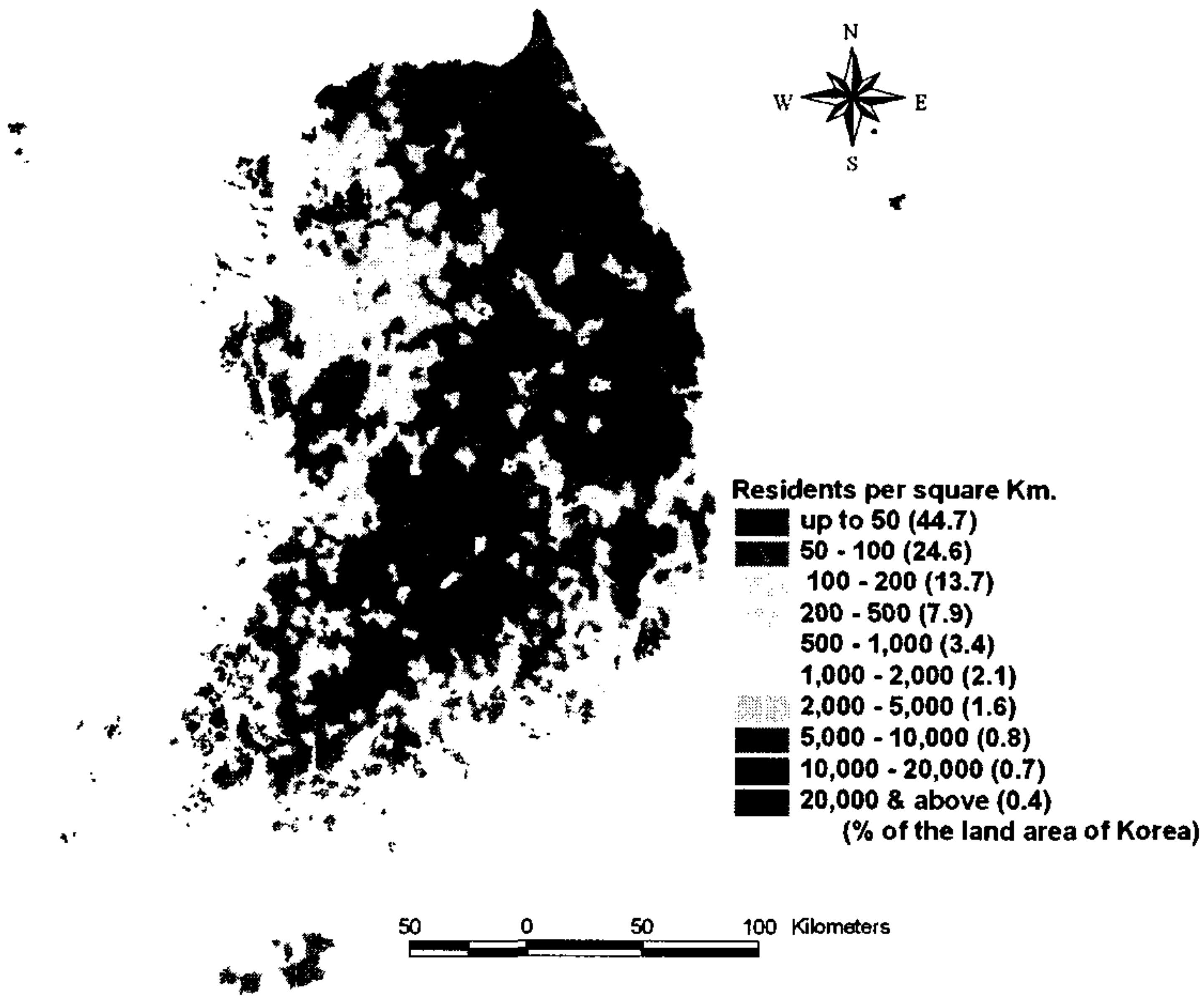

Figure 11. Area population density 2005

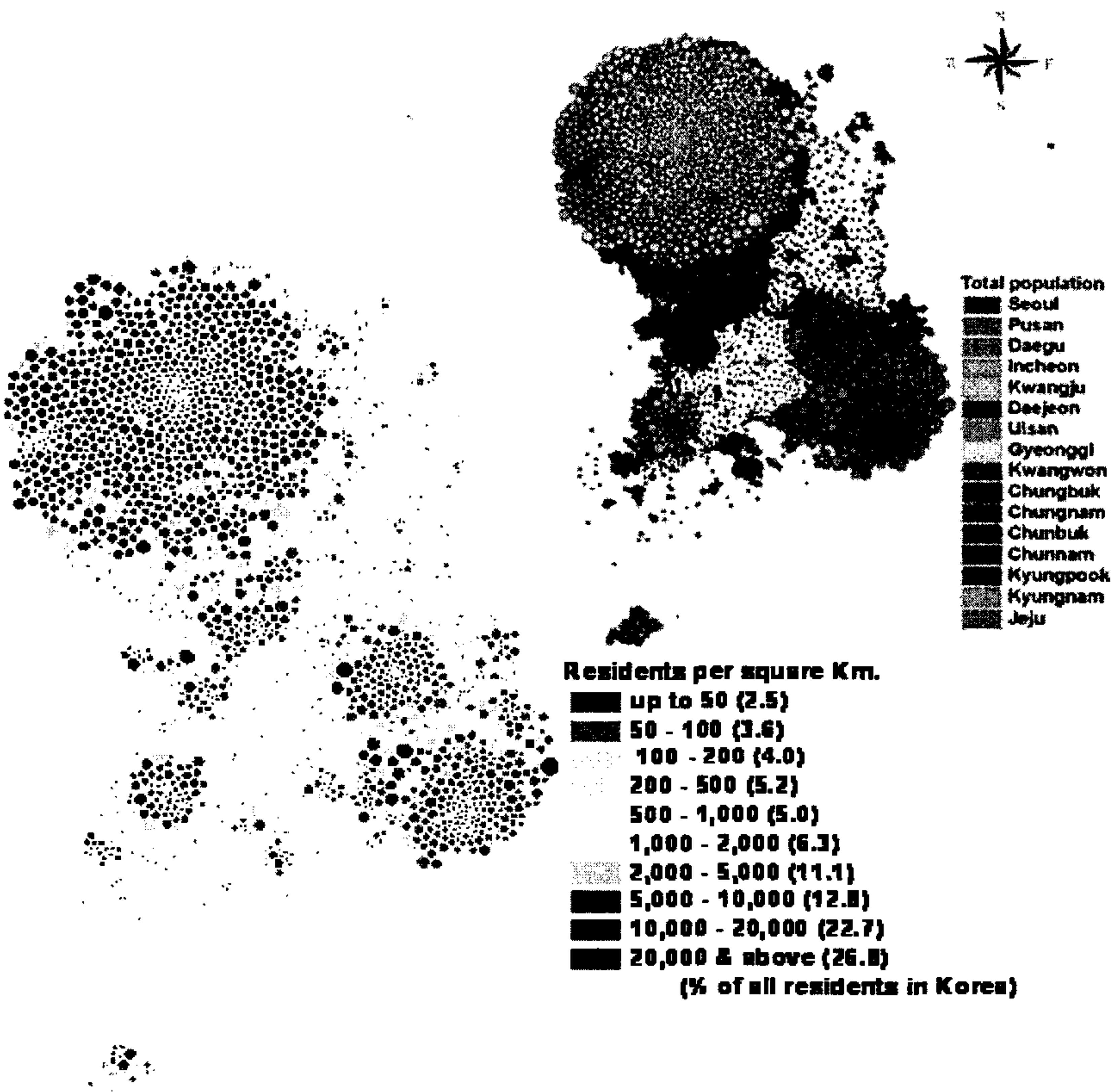

Figure 12. Population concentration 2005

Dong is shown as a circle of size proportion to its population. As it is not possible to draw Si-Gun-Gu or Do boundaries on the Eup-Myeon-Dong population cartogram, the small sized population cartogram that Do boundaries colored is attached on Figure 12's upper-right handed side for viewer's reference. This is a key to identify the regional proportion to its population concentration.

On Figure 12, each Eup-Myeon-Dong boundary is shaded to show the density of population, as in Figure 11. It uses the same key with the same colors. Figure 11 shows that most of the country has below 500 people living in this country. Figure 12 shows that most of the people live in places where many other people live which is generally urban areas in Korea.

It is a clear that the first map (Figure 11) focuses on the land and the second (Figure 12) focuses on the people. As this paper is concerned with people and their geographical distribution, so cartograms are used throughout. This is because population cartograms present prominence to people depending not on how much land they occupy but on their numbers such as population, household, properties, and so on. Next two maps, a thematic map and cartogram, illustrate housing circumstances using dwelling type information from the census data

The type of housing where people live is often a good guide to their general level of life styles and affluence. Figure 13 shows the land area map and population cartogram of Korea; which type of dwelling is most common in each Eup-Myeon-Dong. The land area map illustrates an important difference between the dwelling types, which detached houses predominate in rural areas (Eup-Myeon in Gun administrative boundary) and so they dominate this image. The detached houses as illustrated in Figure 13 constituted 90% of the land of Korea and contained almost 40% of all residents. The cartogram shows how most people live in Eup-Myeon-Dongs where the predominant dwellings are either a detached house or an apartment building. It is supposed that Apartments are dominant in large numbers in

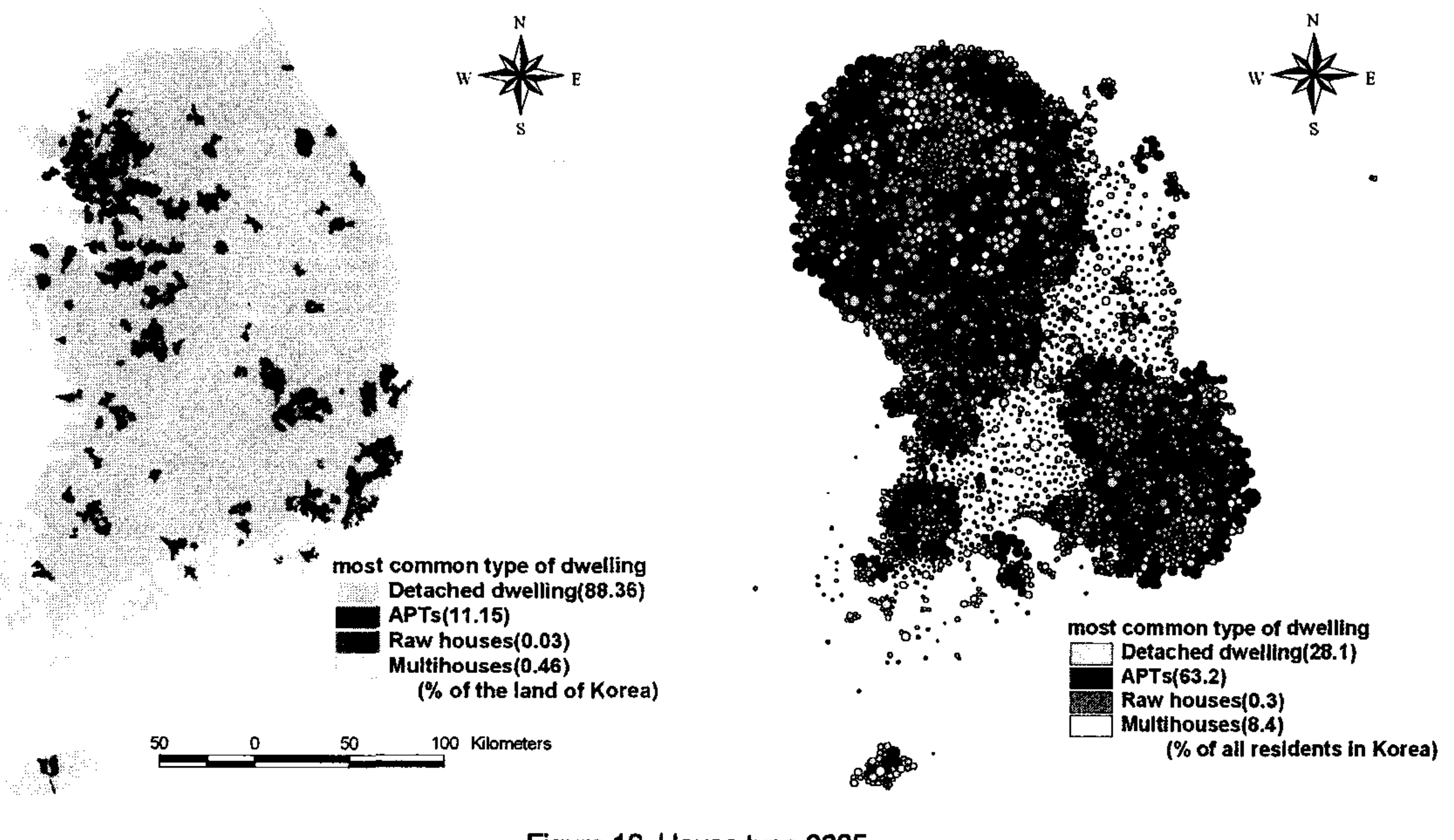

Figure 13. House type 2005

urbanized areas and the outskirts of the metropolitan areas of Seoul and other major cities in Korea.

Compared to the thematic map, it is clearly identified of the spatial concentration of population and properties on small sized Eup-Myeon-Dong areas in Seoul and Gyeonggi region. With no prior knowledge of geographical distribution, people can identify the ratio of the population concentration between the land areas and population cartogram. For example, apartments are concentrated in Seoul and Gyeonggi regions and seven metropolitan city areas, especially in suburban areas with unevenly distribution of the people in this country. Therefore, the population cartogram supports the viewers an easy identification of the correlation of spatial location and the ratio of the dwelling types, which is almost not possible to perceive this correlation with traditional thematic mapping approaches.

6. Conclusions

This paper has presented a general framework for constructing area cartograms with a close coupling strategy, which provides an invaluable tool for visualizing spatial concentration of geographical data. The coupling architecture also demonstrates a practicability well suited to automatically creating population cartogram throughout seamless data exchange between the CCA and ArcView GIS. The CCA offers easy to create population cartograms as well as the opportunity for rapid calculation for large volume of data sets. This practicability shows promises for making it flexible for a wide variety of applications and also envisaging fruitful ways in which GIS analysts and social scientists might collaborate in order to make a new GIS disciplinary area. This research remains further research works to be undertaken. Recently, various cartogram techniques have been developed in the purposes of exploring social structures in a society by making social atlases with adequate cartograms. This ensures the interests of census data and its related socio-economic data in geographical regions for the use of wide cartogram applications to the census atlas visualization. Thus, it requires adequate cartogram construction methods to promote the geographical understanding of social structures, by which various cartogram examples of Korea census data are able to be generated automatically without much efforts of visualization. While this paper has applied a close coupling architecture to promote the construction efficiency of the cartogram algorithm, the utilization of the cartograms should be diversified into the explanation of spatial dimensions of social structures that exists underneath a society which is hard to perceive the structures without the use of an adequate visualization technique. For computational topic, performance efficiency and effectiveness of cartogram algorithms should be analyzed and evaluated to explore adequate cartogram techniques suitable for Korea census atlas. For this research, the computing performance of contiguous areal cartograms against CCA based cartogram should be compared and evaluated on both of advantages and disadvantages.

Reference

Dent, B.D., 1972, A note on the importance of shape in cartogram communication, *The Journal of Geography*, 71(7), 393-401.

Dent, B.D., 1999, *Cartography: The Thematic Map Design*, WCB McGraw-Hill, Iowa.

Dorling, D., 1993, Map design for census mapping, *The*

Cartographic Journal, 30, 167-183.

Dorling, D., 1994, Cartograms for visualizing human geography, in Hearnshaw, H.M. and Unwin, D. (eds.), *Visualization in Geographic Information Systems*, Wiley, New York, 85-102.

Dorling, D., 1995a, *A New Social Atlas of Britain*, John Wiley & Sons, West Sussex, England.

Dorling, D., 1995b, Visualizing changing social structure from a census, *Environment and Planning A*, 27, 353-378.

Dorling, D., 1996, Area Cartograms: *Their Use and Creation*, CATMOGS 59, University of East Anglia, England.

Dorling, D. and Thomas, B., 2004, *People and Places: A 2001 Census Atlas of the UK*, The Policy Press, Bristol, England.

Dougenik, J.A., Chrisman, N.R., and Niemeyer, D.R., 1985, An algorithm to construct continuous cartograms, *The Professional Geographer*, 37, 75-81.

Du, C. and Liu, L., 1999, Constructing contiguous area cartogram using ArcView Avenue, *ESRI International Conference*, Paper No. 489, San Diego, CA.

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

Gatrell, A.C., 1987, *On putting some statistical analysis into geographic information systems: with special reference to problems of map comparison and map overlay*, Northern Regional Research Laboratory Research Report Number 5, Lancaster, U.K.

Huber, W., 2002, *Cartogram!* (QD Version), http://arcscripts.esri.com/details.asp?dbid=11804, Also available online at: access 13th Feb. 2008.

Jackel, C., 1997, Using ArcView to create contiguous and noncontiguous area cartograms, *Cartography and Geographic Information Systems*, 24(2), 101-109.

Kreveld, M. and Speckmann, B., 2007, On rectangular cartograms, *Computational Geometry*, 37, 175-187.

Selvin, S., Merrill, D., Schulman, J., Sacks, S., Bedell, L., and Wong, L., 1988, Transformation of maps to investigate clusters of disease, *Social Science and Medicine*, 26(2), 215-222.

Selvin, S., Schulman, J., and Merrill, D., 1992, Distance and risk measures for the analysis of spatial data: A study of childhood cancers, *Social Science and Medicine*, 34(7), 767-777.

Tobler, W.R., 1986, Pseudo-Cartograms, *The American Cartographer*, 13(1), 43-50.

Waugh, T.C., 1986, The Geolink system: interfacing large systems, in Blackmore, M. (ed.), *Proceedings of Auto-Carto*, RICS, London, 76-85.

Wise, S. and Haining, R., 1991, The role of spatial analysis in geographic information systems, *Annual Conference of the Association for Geographic information*, Birmingham, U.K., 3(24), 1-8.

Zhang, Z. and Griffith, D.A., 1997, Developing user-friendly spatial statistical analysis modules for GIS: An example using ArcView, *Computers, Environment and Urban Systems*, 21(1), 5-29.

http://gis.esri.com/library/userconf/proc99/proceed/papers/pap489/p489.htm, Also available online at: accessed 13th Feb. 2008.

Correspondence: YoungHoon Kim, Department of Geography Education, Korea National University of Education, San 7, Darak-ri, Ganae-myeon, Cheonwon-gun, Chungbuk, 363-791, South Korea(e-mail: gis@knue.ac.kr)

교신: 김영훈, 363-791, 충북 청원군 강내면 다락리 산 7 한국교원대학교 지리교육과(이메일: gis@knue.ac.kr)

Recieved April 11, 2008
Accepted June 18, 2008

Appendix. Avenue script of ArcView based CCA cartogram

```
coView = av.GetActiveDoc
theCartoIter = 100
thewkDir = av.GetProject.GetWorkDir.GetFullName
theInputLabel = {"Cartogram Iteration numbers","Set Working directory"}
theInputList = {theCartoIter.AsString,thewkDir}
theChoice = msgbox.MultiInput("Type directory and iteration","Run Circular Cartogram",theInputLabel,theInputList)
av.GetProject.SetWorkDir(theChoice.Get(1).AsFileName)
theWkdir = av.GetProject.GetWorkDir.GetFullName
theCheck = theWkdir.Right(1)
if ( (theCheck = "\").not) then
 theWkDir= theWkDir+"\"
end
theIteration = theChoice.Get(0)
theThemes = {}
'Select the theme
for each rec in coView.GetThemes
  if (rec.Is(theme)) then
    theThemes.Add(rec)
  end
end
coThm = MsgBox.ListAsString(theThemes,"Get a theme","Circular Cartogram")
if (coThm = nil) then
  exit
end
'check if polygon theme
if ((coThm.GetSrcName.GetSubName = "polygon").Not) then
  msgbox.Info("The selected theme is not polygon","Error")
  exit
end
coFtab = coThm.GetFtab
'Check if a field named "perimeter for Cartogram" exists
coFTab.SetEditable(true)
thePerimeterExist = (coFTab.FindField("Cartprimt") = nil).NOT
if (thePerimeterExist) then
  coFTab.RemoveFields({coFTab.FindField("Cartprimt")})
end
'Creat a new perimeter ID Field
thePerimeterFld = Field.Make("Cartprimt",#FIELD_FLOAT,12,3)
coFTab.AddFields({thePerimeterFld})
'Check if a field named "CartID" exists
theCartIDexist = (coFTab.FindField("CartID") = nil).NOT
if (theCartIDexist) then
   if (MsgBox.YesNo("Overwrite existing CartID field?","the CartID field already exist", true)) then
    'if ok to overwrite, deletect the field as it may not be defined
    'as required by this script (eg., created from another script).
    if (theCartIDexist) then
      coFTab.RemoveFields({coFTab.FindField("CartID")})
    end
  else
    coFTab.SetEditable(false)
    exit
  end 'if (MsgBox...)
end 'if
'Create a new Cartogram ID and perimeter Fields
theFld = Field.Make("CartID",#FIELD_SHORT,5,0)
coFTab.AddFields({theFld})
theCount = 0
theFieldName = coFTab.FindField("CartID")
shpFld = coFTab.FindField("Shape")
thePerimeterNameFld = coFTab.FindField("Cartprimt")
for each rec in coFtab
  theCount = theCount + 1
  coFTab.SetValueNumber(theFieldName,rec,theCount)
  thePoly = CoFTab.ReturnValue(shpFld,rec)
  coFTab.SetValue(thePerimeterNameFld,rec,thePoly.ReturnLength)
end
coFTab.SetEditable(false)
dist = 0
miles = 0
'CLOSE THE POLYGON THEME'S TABLE IF IT EXISTS.  THIS SPEEDS UP THE PROCESS.
polyTab = av.GetProject.FindDoc("Attributes of"++coThm.GetName)
if (polyTab <> nil) then
  polyTab.GetWin.Close
end
'SELECT THE POLYGON UNIQUE ID FIELD
fldList = coFtab.GetFields
idFld = MsgBox.ListAsString(fldList,"Select a field with unique values to identify the adjacent polygons.","Adjacent Polygons")
if (idfld = nil) then
  Exit
end
PopFld = MsgBox.ListAsString(fldList,"Select a field of Population.", "Circular Cartogram")
if (Popfld = nil) then
  exit
end
'CREATE A new text file
theTempFile = theWkDir+"dcartin.txt"
theFN = FileDialog.Put(theTempFile.AsFileName, "*.txt", "Adjacent file")
if (theFN = nil) then
  exit
end
theWriteFile = LineFile.Make(theFN,#FILE_PERM_WRITE)
if (theWriteFile = nil) then
  MsgBox.Info("Cannot open file:"++theFN.GetFullName,"")
  exit
end
theAdjFile = {}
'CREATE A LIST OF NUMBERIC FIELDS
numFldList = {}
for each fld in fldList
  if ((fld.isTypeShape).not) then
    numFldList.Add(fld)
  end
end
'START THE EDIT LOOP
coFTab.StartEditingWithRecovery
coFTab.BeginTransaction
'ADD THE COUNT FIELD
PrimeIDfld = Field.Make("CartoID",#FIELD_SHORT,5,0)
thePop = Field.Make("Pop",#FIELD_SHORT,12,0)
theCenterX = Field.Make("CentroX",#FIELD_DOUBLE,12,0)
theCenterY = Field.Make("CentroY",#FIELD_DOUBLE,12,0)
theCntPolys = Field.Make("CntPoly",#FIELD_SHORT,5,0)
coFTab.AddFields({PrimeIDfld,thePop,theCenterX,theCenterY,theCntPolys})
IDWidth = idfld.GetWidth
fldWidth = popfld.GetWidth
'CREATE A LIST OF ALL THE POLYGON FIELDS IN THE THEME TABLE
 PolyFldNames=
{"N1","N2","N3","N4","N5","N6","N7","N8","N9","N10","N11","N12","N
```

```
13","N14","N15","N16","N17","N18","N19","N20"}
  PeriFldNames =
{"P1","P2","P3","P4","P5","P6","P7","P8","P9","P10","P11","P12","P13","
P14","P15","P16","P17","P18","P19","P20"}
  PolyFldList ={}
  for each fldName in PolyFldNames
    fld  = Field.Make(fldName,#FIELD_CHAR,fldWidth,0)
    PolyFldList.Add(fld)
  end
  PeriFldList = {}
  for each fldName in PeriFldNames
    fld2 = Field.Make(fldName,#FIELD_LONG,fldWidth,0)
    PeriFldList.Add(fld2)
  end
coFtab.GetSelection.ClearAll 'UNSELECT ALL COUNTIES
str = " "
str = str + coFTab.GetNumRecords.AsString + " "
str = str + theIteration + " "
theWriteFile.WriteElt(str)
count = -1
'FOR EACH COUNTY IN THE COUNTY THEME
for each rec in coFtab '.GETSELECTION
   coFtab.GetSelection.Set(rec)
   primeRec = rec.Clone
   primeID = coFTab.ReturnValue(idfld,rec)
     coFtab.SetValue(primeIDFld,rec,primeID)  'WRITE THE PRIME
POLYS ID TO ANOTHER FIELD
   theADjFile.Add(PrimeID)
   coShape = coFtab.ReturnValue(coFTab.FindField("shape"),rec)
   theCtr = coShape.ReturnCenter
   theCtrX = theCtr.GetX
   theCtrY = theCtr.GetY
   PrimePop = coFTab.ReturnValue(Popfld,rec)
coFTab.Setvalue(thePop,rec,PrimePop.setformat("dd"))
   coFTab.SetValue(theCenterx,rec,theCtrX)
   coFTab.SetValue(theCentery,rec,theCtrY)
   'SELECT THE NEAREST COUNTIES TO THE PRIME COUNTY
     coThm.SelectByTheme (coThm,#FTAB_RELTYPE_ISWITHIND
ISTANCEOF, dist*miles, #VTAB_SELTYPE_NEW)
        coFtab.GetSelection.Clear(primeRec) 'UNSELECT THE PRIME
COUNTY
   coFtab.UpdateSelection
       numCo = coFtab.GetSelection.Count 'COUNT ALL ADJACENT
POLYGONS
   if (numCo > 20) then
       msgbox.info("CartID "+idfld.AsString+"Exceed 20 fields
!!!!","Warning")
   end
   coFTab.SetValue(theCntPolys,rec,numCo)
   theAdjFile.Add(PrimePop)
   theAdjFile.Add(theCtrX)
   theAdjFile.Add(theCtrY)
   theAdjFile.Add(numCo)
'******* ADD ADDITIONAL POLYGON FIELDS TO THE POLYGON
TABLE IF NECESSARY
   'Test to see if there are enough fields, if not add more
   if (count < numCo) then
     For each num in count+1..(numCo)
       coFtab.AddFields({PolyFldList.Get(num)})
       coFtab.AddFields({PeriFldList.Get(num)})
     end
     count = numCo
   end
'GET INFORMATION FOR EACH OF THE NEAREST COUNTIES
  polyCount = 0
  for each rec in coFtab.GetSelection
    ID = coFtab.ReturnValue(idfld,rec)
     coPerimeter = coFTab.ReturnValue(coFTab.FindField("Cartprimt")
,rec)

coFTab.SetValue(PolyFldList.Get(polyCount),primeRec,ID.AsString)

coFTab.SetValue(PeriFldList.Get(polyCount),PrimeRec,coPerimeter)
    polyCount = polyCount+1
    theAdjFile.Add(ID)
    theAdjFile.Add(coPerimeter.SetFormatPrecision(2))
    End
    'TOTAL EACH SUMMARY FIELD SELECTED BY THE USER
  coFtab.GetSelection.Set(primeRec)  'Select the prime county
  coFtab.GetSelection.ClearAll 'UNSELECT ALL COUNTIES
str = " "
  for each i2 in theAdjFile
    str = str + i2.AsString+" "
  end
 theWriteFile.WriteElt(str)
 theAdjFile.Empty
end
theWriteFile.Close
coFTab.removeFields(numfldList)
coFTab.EndTransaction
saveEdits = FALSE
coFTab.StopEditingWithRecovery(saveEdits)
system.execute(theWkDir+"dcart.exe")
msgbox.info("Cartogram computation is completed!!!!","Cartogram")
theOutVtab = VTab.Make("dcartout.txt".AsFileName,FALSE,FALSE)
theOutVtabFile = File.Exists("dcartout.txt".AsFileName)
if (theOutVtabFile) then
  msgbox.info("Cartogram computation is completed!!!!","Cartogram")
else
   msgbox.info("FAIL to Cartogram computation!"+NL+"Check
cartogram program","Cartogram message")
  exit
end
xField = theOutVtab.FindField("X-coord")
yField = theOutVtab.FindField("Y-coord")
RadiusField = theOutVtab.FindField("Scaledr")
CartValField = theOutVtab.FindField("Cartvalue")
theXYSrc = XYName.Make(theOutVtab,xField,yField)
theOutTheme = Theme.Make(theXYSrc)
def = av.GetProject.MakeFileName("dcart","shp")
def = FileDialog.Put(def, "*.shp", "New Theme")
if (def = nil) then
  exit
end
theSelectedGraphics = {}
theScaledRadius = {}
theSelectedCartVal = {}
msgbox.info("Vtab record number: "++theOutVtab.getnumrecords.
asstring,"")
for each rec in theOutVtab
 theXpoint = theOutVtab.ReturnValue(xField,rec)
 theYPoint = theOutVtab.ReturnValue(yField,rec)
 theRadius = theOutVtab.ReturnValue(RadiusField,rec)
 theCartValue = theOutVtab.ReturnValue(CartValField,rec)
 theCircle = Circle.Make(theXpoint@theYpoint,theRadius)
 theGraphicCircle = GraphicShape.Make(theCircle)
 theSelectedGraphics.Add(theGraphicCircle)
 theScaledRadius.add(theRadius)
 theSelectedCartVal.Add(theCartValue)
end
```

```
if (theSelectedGraphics.Count = 0) then
  MsgBox.Info("Please, select graphics to convert to shapefile","Select
Graphics")
  exit
end
if (def <> nil) then
  tbl = FTab.MakeNew(def, Polygon)
  if (tbl.HasError) then
    if (tbl.HasLockError) then
        MsgBox.Error("Unable to acquire Write Lock for file " +
def.GetBaseName, "")
    else
      MsgBox.Error("Unable to create " + def.GetBaseName, "")
    end
    return nil
  end
  fld = Field.Make("CirID",#FIELD_SHORT,5,0)
  Xfld = Field.Make("X-coord",#FIELD_FLOAT,12,3)
  Yfld = Field.Make("Y-coord",#FIELD_FLOAT,12,3)
  ScaledRfld = Field.Make("Scaledr",#FIELD_FLOAT,12,5)
  Radiusfld = Field.Make("Radius",#FIELD_FLOAT,12,5)
  CartValfld = Field.Make(Popfld.Asstring,#FIELD_FLOAT,12,3)
  fld.SetVisible(TRUE)
  tbl.AddFields({fld,Xfld,Yfld,Radiusfld,ScaledRfld,CartValfld})
  tbl.SetEditable(FALSE)
  theTheme = FTheme.Make(tbl)
  theNewFtab = theTheme.GetFtab
  theNewShapeFld = theNewFTab.FindField("Shape")
  coView.AddTheme(theTheme)
  theTheme.SetActive(TRUE)
  theTheme.setVisible(TRUE)
  coView.SetEditableTheme(theTheme)
  av.GetProject.SetModified(true)
end
if (theNewFTab = nil) then
  exit
end
theIDcount = 0
for each rec in theSelectedGraphics
  theIDCount = theIDCount + 1
  theShape = rec.GetShape
  theNewRec = theNewFtab.AddRecord
  thePolygon = theShape.AsPolygon
  theNewFTab.SetValue(theNewShapeFld,theNewRec,thePolygon)
  theNewFTab.SetValueNumber(fld,theNewRec,theIDCount)
  theX = thePolygon.ReturnCenter.GetX
  theY = thePolygon.ReturnCenter.GetY
  theRadius = thePolygon.ReturnLength
  theNewFTab.SetValueNumber(Xfld,theNewRec,theX)
  thenewFTab.SetValueNumber(Yfld,theNewRec,theY)
  theNewFTab.SetValueNumber(Radiusfld,theNewRec,theRadius)

thenewFTab.SetValue(ScaledRfld,theNewRec,theScaledRadius.Get(the
IDCount-1))
  theCartVal = theSelectedCartVal.Get(theIDCount-1)
  theNewFTab.SetValue(CartValfld,theNewRec,theCartVal)
end
theNewFTab.join(fld,coFtab,theFieldName)
theNewFTab.SetEditable(FALSE)
tbl.SetEditable(FALSE)
```