Scale Dependent Rendering with Knowledge Base

Jinmu Choi
Dept. of Geosciences
Mississippi State University

Abstract

Geometric representations of geographic features depend significantly on the visual scale. A building may be a point in a small scale but a polygon in a large scale. This paper describes a methodology that allows scale dependent rendering with a symbol knowledge base. To render features at a particular scale, we first specify scale ranges and symbol rules for each feature and then store them in a symbol knowledge base. The main contribution of the paper is an algorithm for feature geometry reconstruction based on the scale for adequate symbolization. Another useful methodology is the use of knowledge base for symbolization that can be accumulated and adapted for various mapping applications.

Keywords: Scale Dependent Rendering, Symbol Knowledge Base, Geometry Reconstruction Algorithm.

1. Introduction

Visualization plays an important role in the exploration and understanding of geographic phenomena. It makes it easier to better understand relations of features and complex processes of phenomena. Visualization is the process of representing information for exploration and communication (Butterfield and Mackaness, 1991; Peterson, 1994).

Advances in the Internet and spatial monitoring technology such as remote sensing and global position system (GPS) have ushered in an explosion in the production of spatial data. Now many countries have built spatial data clearing house to provide various types of spatial data to the public. The rapid growth of the Internet and the stronger utilization of geographic information systems (GIS) accelerate the use of spatial data in the form of digital maps (Longley et al., 1999).

Maps have been used for the visualization method of spatial data for a long time. To communicate information clearly, a map should be designed correctly based on a rigorous map making process. Cartography provides the guidelines for the representation of visual variables in this process (Bertin, 1983; Dent, 1990). The visual variables include feature selection, types of symbols, and map scale. The principles in conventional cartography have focused on the design of a paper map. A paper map is static, and requires a selection of feature geometry and symbols at a fixed map scale (Goodchild, 1999).

With the development of improved hardware for graphic display, there is now a high demand for digital maps (Elzakker van, 2001). Digital maps allow interaction between map and user. The interactive map allows a user to display temporal or dynamic processes to create maps temporarily. The user can therefore produce their own maps based on their requirements (Gower et al., 1997). The rapid growth in the Internet usage is another reason for the strong growth of the digital map. Peterson (2001) noted that the Internet had an enormous impact on the use of maps, map generation processes, and their distribution. In contrast to paper maps, digital maps can be personalized, updated easily, and interactive.

A digital map is dynamic. Interactive zooming enables changes in map scale, which requires that feature geometry types and their symbols be adjusted automatically. The purpose of this paper is to provide a method for scale dependent rendering for the dynamic display of such spatial data. It requires construction of rules for automatic adjustment of feature geometry types and their symbols along a range of visual scales. A symbol knowledge base is used to store those rules for later applications.

Section 2 below discusses how feature geometries are related to the visual scale including the process of geometry conversion. Section 3 discusses the process of feature visualization using symbol knowledge base along with the changes of visual scale. Section 4 describes the implementation of a scale dependent rendering tool. Section 5 shows simple applications of multiscale rendering. Key ideas are summarized in the conclusion.

2. Feature Geometry Generalization with Visual Scale

In digital cartography, the process of reducing the amount of data and adjusting the information to a given scale is called cartographic generalization (Muller, 1991; McMaster and Shea, 1992; Weibel and Dutton, 1999). Particularly, the process of changing geometry types is called collapse, which is one of the available generalization methods (ESRI, 1996). Collapse is the process of reducing a feature dimension; for instance, changing area features to linear or point features (Table 1). To change a feature dimension, the properties of the feature such as area size or width can be used as criteria. The change of geometry types requires on-the-fly generation according to the changes of visual scale (Jones et al., 2000; Feringa, 2001; Kraak and Brown, 2001).

Table 1. Feature Geometry Types with the Visual Scale Range

Scale Factor	**Larger than 1k**	**1k to 24k**	**24k to 100k**	**Smaller than 100k**
Pole	Point	Point	Point	Point
Road	Line	Line	Line	Line
River	Polygon	Polygon and (or) Line	Line	Line
Building	Polygon	Polygon or Point	Point	Point
Vegetation Type	Polygon	Polygon	Aggregated Polygon	Aggregated Polygon
Political Boundary	Polygon	Polygon	Polygon	Polygon

As an example of collapse, Figure 1 shows the change of geometry types of buildings and streams. When buildings and streams are displayed at a scale smaller than 1:24K on screen, all buildings are represented as point features and streams as lines. As the visual scale gets bigger than 1:24K, large buildings are changed into polygons and small buildings remain as points. The main stream is changed into a polygon but its branches remain as lines. When the display scale is larger than 1:1K, all buildings and streams are represented as polygons. If a visual scale on screen is very large, say 1:100K, some features such as political boundaries may be eliminated from the display to reduce complexity.

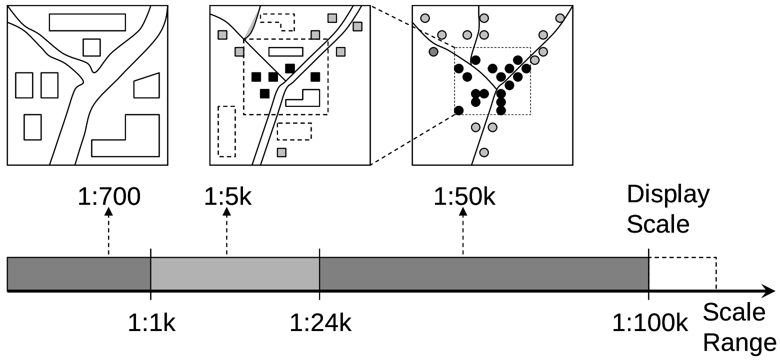

Figure 1. Scale Based Geometry Change

For collapse generalization, two dimension-reducing algorithms can be used: polygon to point and polygon to line. To represent the change of a building based on visual scale, the simplest method is the gravity center calculation algorithm. The gravity center of a polygon can be used to convert a polygon building to a point building based on the size of area when buildings are assumed as convex polygons.

To convert a polygon to a line, a medial-axis should be extracted from a polygon. There are several approaches in the media-axis transformation. A skeleton of a polygon can be derived from the Voronoi-Diagram (Chin et al., 1995). The medial-axis (skeleton) is the connection of all centers of circles inside the polygon. The circles touch the polygon in two or more points. This method creates second order lines at the intersections of skeleton lines. The medial-axis can also be derived using the Delaunay triangulation (Prasad and Rao, 1998). The skeleton is only set up of straight lines. This method has been developed to extract the skeleton of a complex natural feature such as the human body. The straight skeleton (Eppstein and Erickson, 1999) is another method of extracting the medial-axis, which is applied to construct a roof on a given ground plan. To change the geometry type of a river in this paper, the medial-axis transformation using the Delaunay triangulation is used because it is good for the morphological change of a shape.

3. Dynamic Symbolization along with Scale Change

To display geometries on screen, symbols should be specified for each feature class. Geometry changes of a feature require different symbols for representation. Even though a feature can remain in the same geometry types, the symbol for the geometry can be changed along with the changes in the visual scale. All road feature classes can be represented using the same symbols at very small scale, but the highway feature class can be represented using different symbols.

The current GIS does not have the ability to store the dynamic symbolization for feature representation. In GIS, only one set of symbols can be stored as a symbol file. To store multiple symbols for various scale ranges, a new data structure needs to be designed. In this paper, a symbol knowledge base is designed to store geometry types, symbols, and available scale ranges for each feature class.

The process of feature visualization with a symbol knowledge base is illustrated in Figure 2. For each feature class, the decision rules should be specified for geometry types and symbols based on the available scale ranges. Specified rules are stored in a symbol knowledge base. Symbols for various feature classes can be accumulated in the knowledge base, which can be a reference for other mapping applications. Every time the user zooms in or out, a map is generated for the requested scale. When a feature is considered to be a part of the display map at the visual scale, its geometry is selected, simplified, and symbolized using geometry types and symbols specified in a symbol knowledge base.

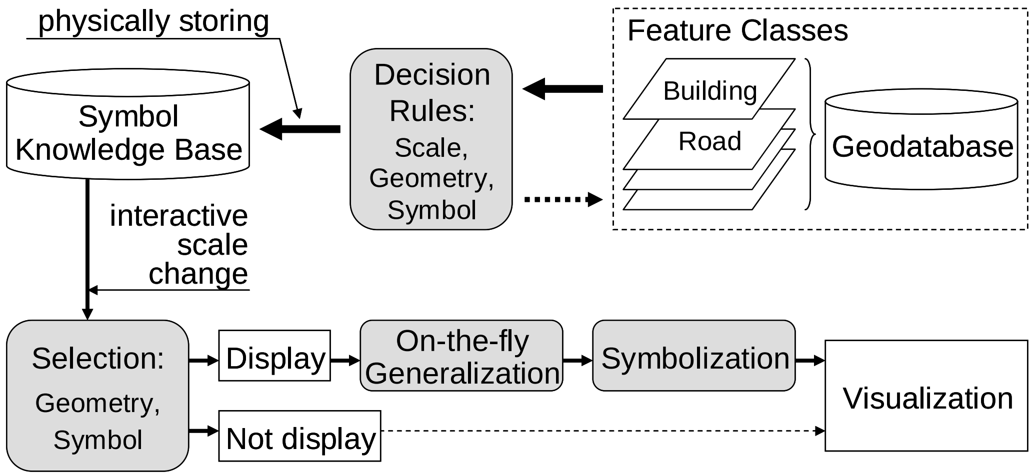

Figure 2. The Process of Feature Visualization Using Symbol Knowledge Base

4. Implementation for Scale Dependent Rendering

The scale dependent rendering (SDR) tool consists of the symbol database, symbol selector and rendering module (Figure 3). The symbol database stores symbol knowledge, which is a collection of the decision rules. The symbol selector is the user interface for the interactive selection of decision rules, which allows users to decide on feature geometry types and symbols along with the specified scale range. The rendering module reconstructs geometry types for each feature class on-the-fly and draws features using symbols in the map according to decision rules in the symbol database. The scale dependent rendering tool was developed as a plug-in in the ESRI’s ArcMap using Arc Developer’s toolkit with Visual Basic.

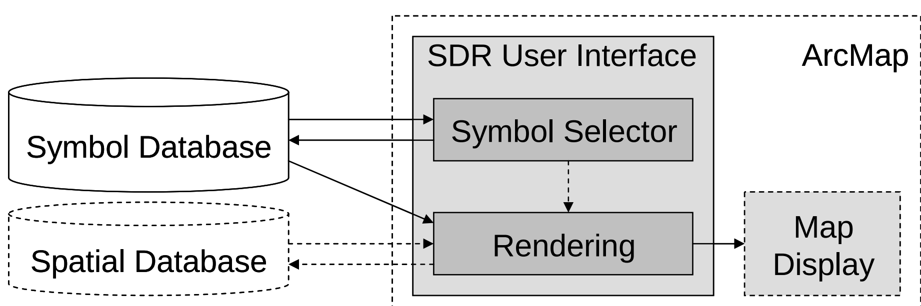

Figure 3. Design of the Scale Dependent Rendering Tool

The symbol database consists of three tables based on the feature geometry types: point, line, and polygon tables. Each table includes feature name, scale ranges, and symbols for each scale range (Figure 4). The combination of these three types of attributes decides the rules for rendering features on a display. For example, buildings can be represented as polygons in the maximum scale range, as polygons and points in the medium scale range, and as points in the minimum scale range.

Point Table

- Feature Name
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Point Style
- Point Color
- Point Size
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Point Style
- Point Color
- Point Size
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Point Style
- Point Color
- Point Size

Line Table

- Feature Name
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Line Style
- Line Color
- Line Size
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Line Style
- Line Color
- Line Size
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Line Style
- Line Color
- Line Size

Polygon Table

- Feature Name
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Use Polygon
- Polygon Style
- Polygon Color
- Outline Type
- Outline Width
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Use Polygon
- Polygon Style
- Polygon Color
- Outline Type
- Outline Width
- Use_Max_scale
- From_Max_scale
- To_Max_scale
- Use Polygon
- Polygon Style
- Polygon Color
- Outline Type
- Outline Width
- Use point (Line) for max
- Point (Line) Style
- Point (Line) Color
- Point (Line) Size
- Criteria Value
- Use point (Line) for mid
- Point (Line) Style
- Point (Line) Color
- Point (Line) Size
- Criteria Value
- Use point (Line) for min
- Point (Line) Style
- Point (Line) Color
- Point (Line) Size
- Criteria Value

Figure 4. Components of the Symbol Database

The symbols in the symbol database are simplified but can be extended to include complex symbol types such as character, picture, multi-layer symbols. Each table in the symbol database stores symbols according to the corresponding geometry types. Especially, the polygon table also includes point and line symbols to symbolize polygon features when the polygons are represented as points or lines in a specific scale range.

The symbol selector retrieves existing symbols for feature classes from a symbol database and, if necessary, modifies the symbols. It also can be used to set a new symbol when a new feature class is added in the symbol database. The user interface of the symbol selector consists of the layer table and range symbol selector (Figure 5). The layer table includes all feature class layers used in the current display. The range symbol selector is used to specify symbols for each feature class in the layer table.

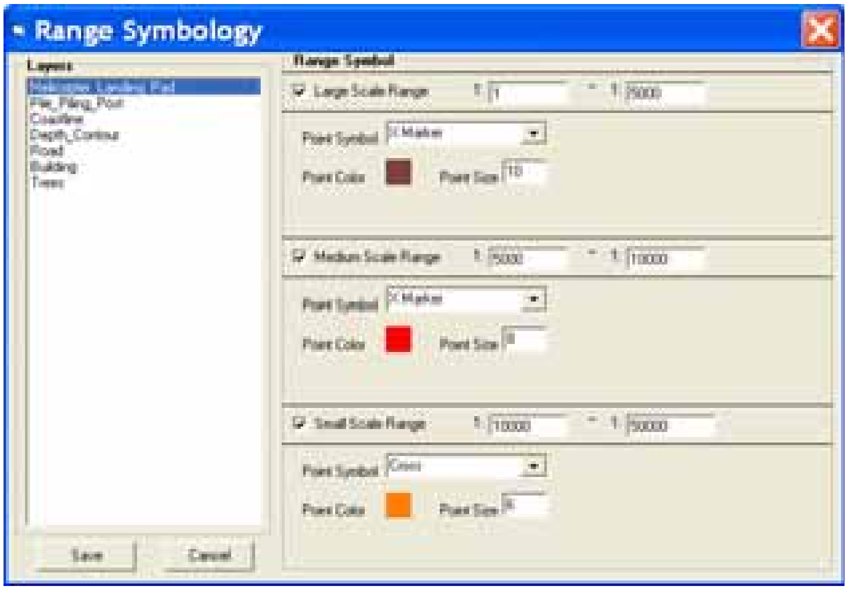

Figure 5. The User Interface of a Symbol Selector for Point Feature

The rendering tool applies symbols to the display. The process needs four steps: geometry type checking, scale checking, on-the-fly geometry reconstruction, and symbolization (Figure 6). First, it retrieves all symbol rules from a symbol database for each feature class based on the current geometry type of each class. Second, it compares current visual scale to the scale range in the symbol database in order to identify geometry types and symbols to be used. Third, based on the geometry type rule with the visual scale, the generalization is executed. Finally, newly generated geometry is visualized using specified symbols.

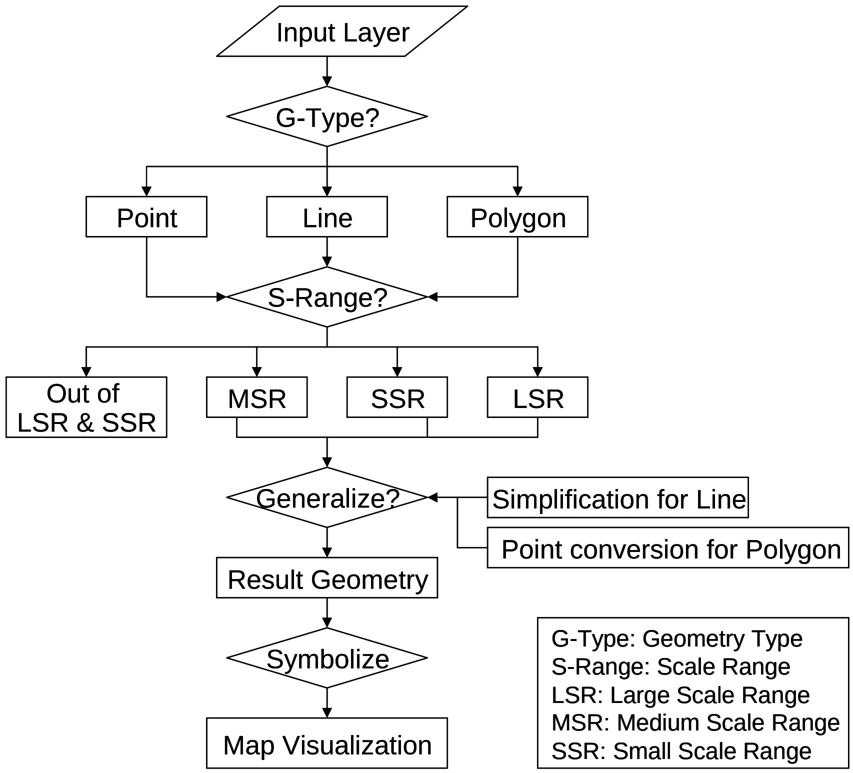

Figure 6. Dynamic Symbolization Process along with Visual Scale

For on-the-fly geometry reconstruction, two type-conversion algorithms are implemented. For the conversion of a polygon to a point, the gravity center of a polygon is extracted to represent a feature. To represent the change of a river, the partial change of geometry is necessary (Haunert and Sester, 2004). A part of a river can be modeled as a polygon or a line depending on its width. The algorithm implemented in this study for the partial geometry change needs four steps (Figure 7). First, the Delaunay triangles inside a given polygon are created. Second, the inner nodes of the medial-axis are extracted. The inner nodes are the line center points of the sides of the inner triangle, which are inside the polygon. Third, the medial-axis is created by connecting the inner nodes to the start nodes. The start node is a vertex of a Delaunay triangle that has only one side located inside the polygon. Finally, a partial geometry change is completed by combining a medial-axis to the polygon based on the width of the polygon.

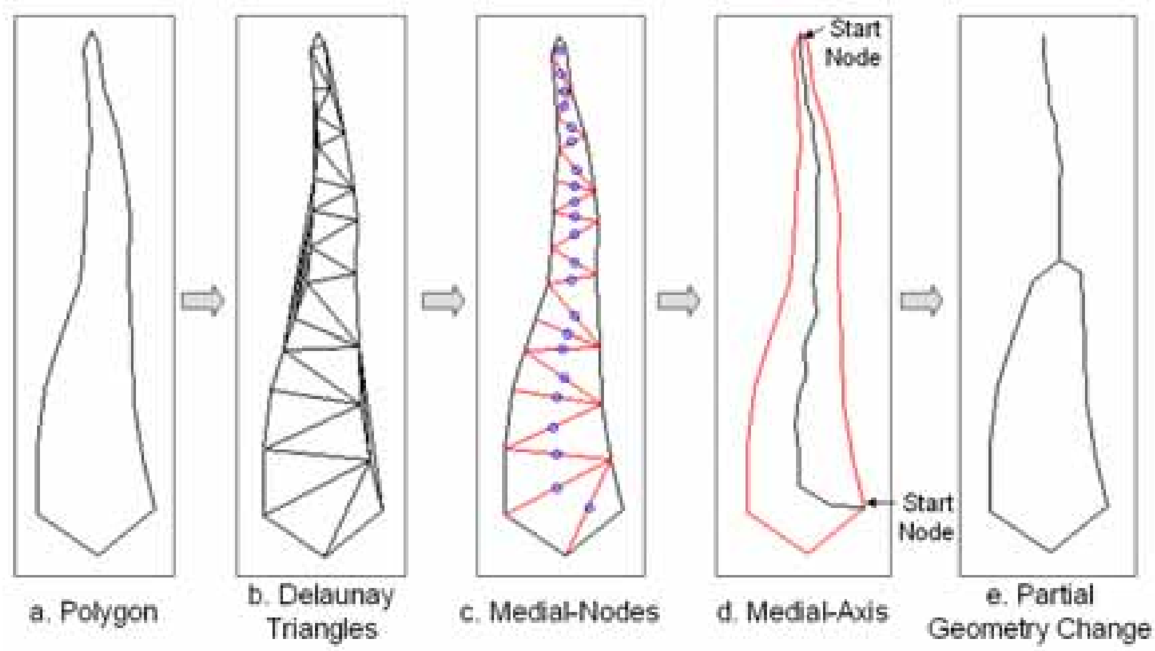

Figure 7. The Process Executing a Partial Geometry Change using a Medial-Axis

5. Multiscale Rendering for Buildings and Rivers

The study area of the applications is the Camp Lejeune Marine Corps Base, North Carolina (Figure 8). To test the implementation, the application focused on the two feature types: buildings and rivers. Symbol rules for buildings and rivers are specified based on the visual scale ranges (Table 2).

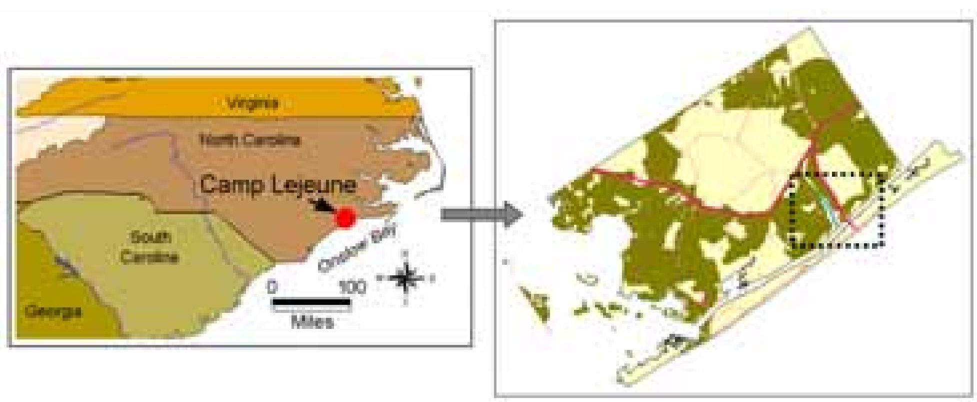

Figure 8. Study Area

Table 2. Symbol Rules for Building and River Stored in the Symbol Knowledge base

Feature Class	**Building**	**River**
Large Scale Range	1 : 1 K ~ 1 : 3 K	1 : 1 ~ 1 : 10 K
Use Polygon	Polygon	Polygon
Polygon Style	Solid Fill	Solid Fill
Polygon Color	Red	Cyan
Outline Type	Solid Line	Solid Line
Outline Width	1	1
Medium Scale Range	1 : 3 K ~ 1 : 6 K	1 : 10 ~ 1 : 24 K
Use Polygon	Polygon	Polygon
Polygon Style	Solid Fill	Solid Fill
Polygon Color	Blue	Cyan
Outline Type	Solid Line	Solid Line
Outline Width	1	1
Small Scale Range	1 : 6 K ~ 1 : 50 K	1 : 24 ~ 1 : 100 K
Use Polygon	No	No
Polygon Style	•	•
Polygon Color	•	•
Outline Type	•	•
Outline Width	•	•
Point (Line) for Large Scale	No	No
Point (Line) Style	•	•
Point (Line) Color	•	•
Point (Line) Size	•	•
Criteria Value	•	•
Point (Line) for Medium Scale	Point	Line
Point (Line) Style	Circle	Solid Line
Point (Line) Color	Dark Red	Cyan
Point (Line) Size	2	1
Criteria Value	Area Size: 50	Width: 100
Point (Line) for Small Scale	Point	Line
Point (Line) Style	Circle	Solid Line
Point (Line) Color	Black	Cyan
Point (Line) Size	1	1
Criteria Value	•	•

In the digital maps, the adaptation of the content of a map is strongly scale dependent. The geometry types of features on screen are also dependent on the display scale. In a very large scale, most features are represented as polygons. When they get smaller in visual scale on screen, feature geometry types need to be changed. Figure 9 shows the changes of geometry types and symbols for buildings along with the changes of visual scale on screen. Between 1:1K and 1:3K (large scale range for buildings), all buildings are modeled as polygon features. Between 1:3K and 1:6K (medium scale range for buildings), some buildings which are larger than 50 m^2 are mapped as polygons and the others which are smaller than 50 m^2 are mapped as points. At a smaller scale range (smaller than 1:6K) all buildings are mapped as points.

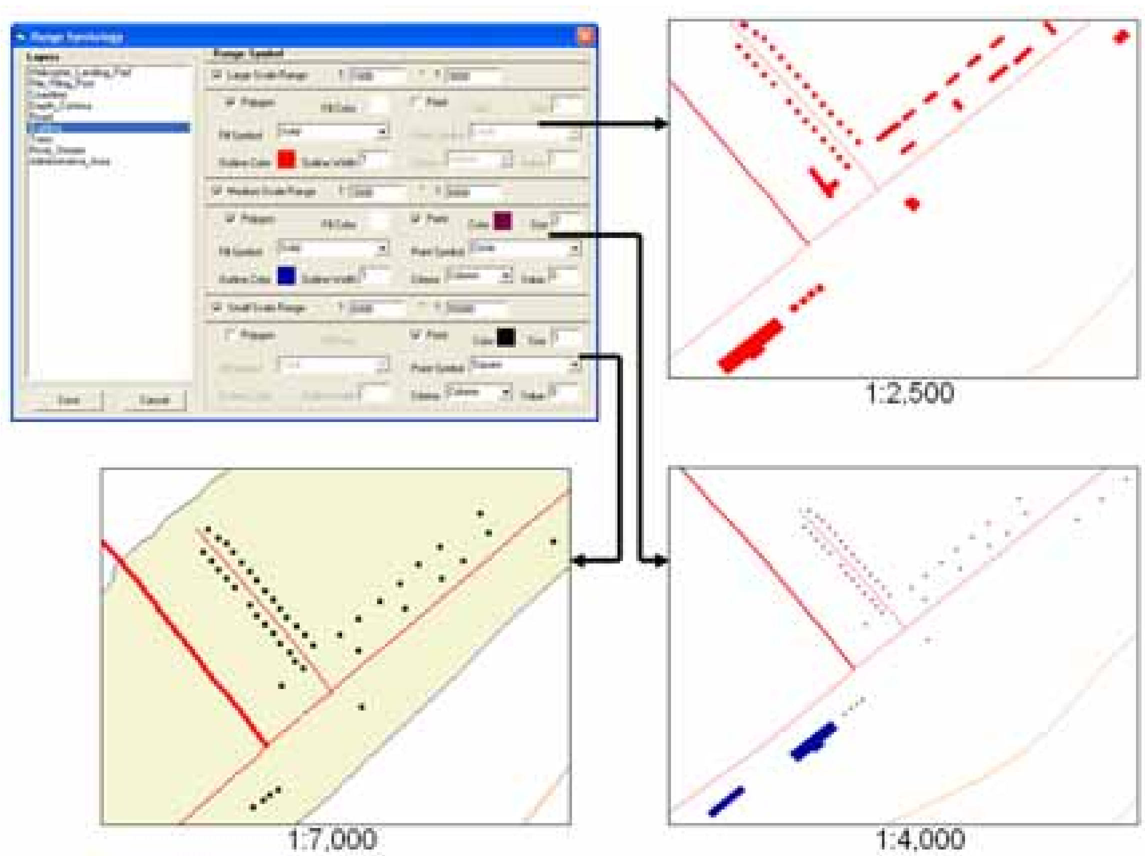

Figure 9. Multiscale Rendering of Buildings along with the Changes of Visual Scale

Figure 10 shows the changes of geometry types and symbols for a river. Between 1:1K and 1:10K (large scale range for rivers), rivers are modeled as polygons. Between 1:10K and 1:24K (medium scale range for rivers), some rivers are mapped as polygons and others as lines based on the width of the river. Also, some parts of a river, where its width is larger than 100 m, are mapped as polygons. If the width of the branches of the river is smaller than 100 m, the branches are mapped as lines using medial-axis. At smaller scale range (smaller than 1:24K), rivers are mapped as lines. Along with the changes of visual scale on screen, geometry types and symbols are changed instantly based on the rules specified in the symbol knowledge base.

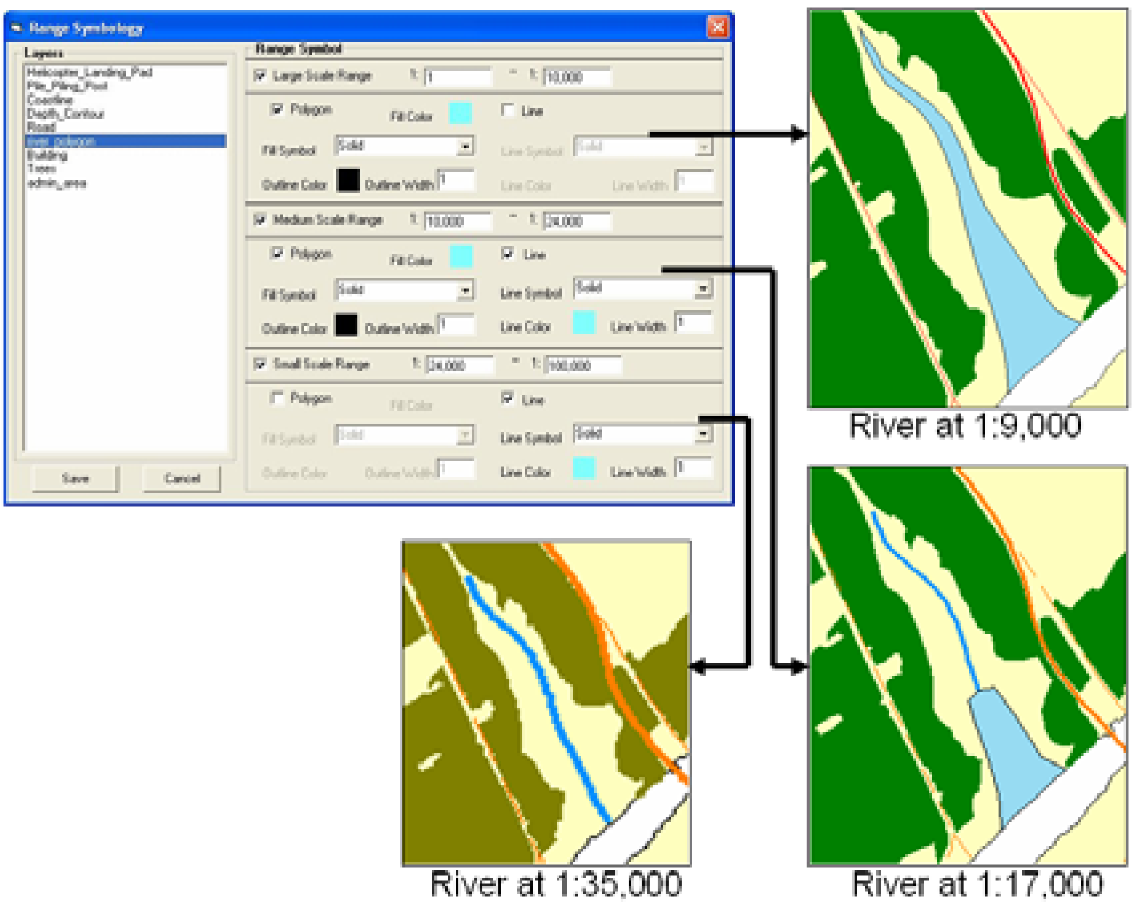

Figure 10. Multiscale Rendering of a River along with the Changes of Visual Scale

6. Conclusions

This paper has discussed a multiscale rendering method with a symbol knowledge base. Some features such as buildings or rivers can be represented as polygons in very large visual scale. Along with the changes of visual scale, the geometry types of a feature can be changed: a polygon to a point or to a line. In some cases, such as a river, a feature can be represented with two different types of geometry through partial geometry change.

In this paper, two methods have been suggested for geometry conversion: polygon to point conversion and a partial geometry change from polygon to line. The algorithms were implemented in the multiscale rendering tool that consists of the symbol selector, the rendering tools, and a symbol knowledge base. A symbol knowledge base stores geometry and symbol rules for each visual scale range. The symbol selector retrieves feature symbols from the symbol knowledge base. The rendering tool applies specified geometries and symbols to the features on screen based on the visual scale.

The implemented tools were tested on buildings and a river. In the medium scale range, some buildings were mapped as polygons and others as points based on their sizes. Also, a partial geometry change was applied for a river based on its width. Some parts of the river, where it was wider than 100 m, were mapped as a polygon. With these applications, the scale dependent rendering tool in this paper effectively visualized feature geometries and symbols based on user interaction on the visual scale. Also, the proposed symbol knowledge base can be accumulated and adapted for various mapping applications.

References

Bertin, J., 1983, *Semiology of Graphics*, University of Wisconsin Press, Madison, Wisconsin.

Butterfield, B. and W.A. Mackaness, 1991, “Visualization”, In D.J. Maguire, M.F. Goodchild, and D.W. Rhind (Eds), *Geographic Information Systems: Principles and Practice*, Longman, London, pp. 427-443.

Chin. F., J. Snoeyink, and C.A., Wang, 1995, “Finding the Medial Axis of a Simple Polygon in Linear Time”, *In Proceedings of the International Symposium on Algorithms and Computation*, pp. 382-391.

Dent, B.D., 1990, *Principles of Thematic Map Design*, Addison-Wesley, Reading, Mass.

Elzakker van, C., 2001, “Users of Maps on the Web”, In M.J. Kraak and A. Brown (Eds), *Web Cartography: Developments and Prospects*, Taylor and Francis, London, pp. 37-52.

Eppstein, D. and J. Erickson, 1999, “Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions”, *In Proceedings of the 14th Annual ACM Symposium on Computational Geometry*, Minneapolis, Minnesota, pp. 58-67.

ESRI, 1996, *Automation of Map Generalization*, Environmental Systems Research Institute, Inc., Redlands, CA.

Feringa, W., 2001, “File Formats and Plugins”, In M.J. Kraak and A. Brown (Eds), *Web Cartography: Developments and Prospects*, Taylor and Francis, London, pp. 177-193.

Goodchild, M.F., 1999, “Cartographic Futures on a Digital Earth”, *In Proceedings of the 19th ICA/ACI Conference*, Ottawa, pp 4-12.

Gower, R., J. Pepper, and T. Edwards, 1997, “Landplan – Automated Generalization Comes of Age”, *In Proceedings of the 16th ICA/ACI Conference*, Stockholm, pp. 126-133.

Haunert and Sester, 2004, *Using the Straight Skeleton for Generalization in a Multiple Representation Environment*, ICA Workshop on Generalization and Multiple Representation, Leicester.

Jones C.B., A.I. Abdelmoty, M.E. Lonergan, P. van der Poorten, S. Zhou, 2000, “Multi-Scale Spatial Database Design for Online Generalization”, *In Proceedings of the 9th International Symposium on Spatial Data Handling*, Beijing, pp. 7b.34-7b.44.

Kraak, M.J. and A. Brown, 2001, *Web Cartography: Developments and Prospects*, Taylor and Francis, London.

Longley, P., M. Goodchild, D.M. Maguire, and D. Rhind, 1999, *Geographical Information Systems: Principles, Techniques, Management, and Applications*, Wiley and Sons, New York.

McMaster, R. and S. Shea, 1992, *Generalization in Digital Cartography*, Association of American Geographers, Washington, USA.

Muller, J.C., 1991, “Generalization of Spatial Databases”, In D.J. Maguire, M.F. Goodchild, and D.W. Rhind (Eds), *Geographic Information Systems: Principles and Practice*, Longman, London, pp. 457-475.

Peterson, M.P., 1994, “Cognitive Issues in Cartographic Visualization”, In A. MacEachern and D. Taylor (Eds), *Visualization in Modern Cartography*, Pergamon, New York, pp. 27-43.

Peterson, M.P., 2001, “Trends in Internet Map Use”, *In Proceedings of the 20th ICA/ACI Conference*, Beijing, pp. 2306-2312.

Prasad, L and R. Rao, 1998, *Morphological Analysis of Shapes*, Web Documents, http://computervision.lanl.gov/pub_001/pub_001.pdf, Last Visit: June 27, 2006.

Weibel, R. and G. Dutton, 1999, “Generalizing Spatial Data and Dealing with Multiple Representations”, In P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind (Eds), *Geographical Information Systems: Principles and Technical Issues*, New York, 2nd edition, pp. 125-155.

Jinmu Choi

Assistant Professor

Department of Geosciences

Mississippi State University

P.O. Box 5448, Mississippi State, MS 39762

e-mail: jc778@msstate.edu

Phone: 1-662-325-5803

Fax: 1-662-325-9423