Chapter 1
Introduction

This book is intended to provide the reader with a thorough understanding of the purpose and function of tile-based mapping systems. In addition, it is meant to be a technical guide to the development of tile-based mapping systems. Complex issues like tile rendering, storage, and indexing are covered along with map projections, network communication, and client/server applications. Computer code as well as numerous mathematical formulae are included to provide the reader with usable forms of the algorithms presented in this book.

1.1 Background of Web-Based Mapping Applications

The first Web-based mapping applications were introduced in the mid to late 1990's. They included Yahoo! Maps, MapQuest, and Microsoft's TerraServer. These providers offered mapping applications through a Web browser. Their map navigation systems were rudimentary. Some allowed simple map movements by requiring users to click on navigation arrow buttons surrounding the map view. When users clicked on an arrow, the map moved a predetermined amount in the direction clicked. There were also buttons for zooming in and out. Others allowed users to drag and draw boxes on the map to relocate the map view.

All of these systems had several disadvantages, including slow rendering and downloading of map views because the map view was often represented by a single large image file. Each time the map was moved to the left or right; the entire image would be re-rendered and re-sent to the client even though only a portion of the image was new. However, the interfaces were relatively simple and had several advantages to developers. Basic interfaces were well suited to early Web browsers. The map interface could be written entirely in HTML or with very minimal JavaScript. Second, since all navigations were fixed, map servers could cache rendered maps. Other map viewers adopted a map view and navigation style more similar to desktop GIS systems. These systems were more complicated and used browser plugin technology and development platforms like Java or Flash.

J.T. Sample and E. Ioup, *Tile-Based Geospatial Information Systems: Principles and Practices*, DOI 10.1007/978-1-4419-7631-4_1,

Google Maps was introduced in 2005 and dramatically changed the way people viewed maps. Instead of clunky and slow map navigation methods, Google Maps provided what has come to be known as a ”Slippy Map” type interface. That interface allowed users to quickly move and zoom the map and yet was written entirely in HTML and JavaScript. Soon many more Web mapping applications appeared with a similar style map interface. Eventually slippy map type interfaces appeared in many places including portable computing devices and cell phones.

A key enabling technology behind this new generation of mapping applications was the concept of tile-based mapping. Mapping applications were made responsive by using background maps that had been broken into smaller tiled images. Those tiles were stored, already rendered, on a central server. Because they were already rendered, they could be sent to clients quickly. The tiles were discretely addressed so they could be cached by Internet caching services and by clients' own browsers. The map images were broken into smaller pieces, so when users navigated the map view, only the new parts of the map had to be resent from the server.

1.2 Properties of tile-based mapping systems

Tile-based mapping systems have several core properties which distinguish them from other types of mapping systems. We have defined what we believe to be those core properties, and they are as follows:

1. Map views are based on multiple discrete zoom levels, each corresponding to a fixed map scale.
2. Multiple image tiles are used to virtualize a single map view.
3. Image tiles are accessible using a discrete addressing scheme.
4. Tiled images stored on a server system are sent to the client with minimal processing; as much processing is done ahead of time as is possible.

The following are important but optional properties of tile-based mapping systems.

1. Tile addressing follows a single global projection.
2. Tiles are primarily distributed using a client/server system architecture.
3. Tiles are organized into relatively few, fixed layers.

1.3 Book Organization

This book is organized to take the reader through the logical development of a complete tile-based mapping system with small detours into important topics along the way. Chapter 2 introduces logical tile addressing schemes that any tile system must implement. It discusses some common schemes used by popular Web mapping systems and defines the common tile scheme that will be used throughout this book.

Chapter 3 gives an overview of the challenges and the techniques used to overcome these challenges to develop client software for tile-based mapping systems. An example client application is shown with source code. Chapter 4 provides extensive background into techniques needed to process source data images into tiled images. Chapters 5 and 6 provide a detailed look at techniques for creating sets of tiled images. Chapters 7 and 8 explain how to efficiently store, index, and retrieve tiled images. Several techniques are detailed, implemented, and benchmarked. Chapter 9 shows the reader how to create a Web based server for tiled images. Chapter 10 introduces and explains map projections within the context of tile based mapping. Chapter 11 explains how vector mapping data can be used in a tile-based environment. Finally, Chapters 12 and 13 are detailed case studies of real-world usage of the techniques presented in this book.

Most chapters include computer code listings in Java and Python. Java and Python are two of the most commonly used programming languages for geospatial programming. Short code segments are interspersed with the chapter text, while longer code segments are placed at the end of each chapter. The code sections are intended to provide readers with example implementations of the algorithms explained in the book.