US 20070242084A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2007/0242084 A1

BOWMAN et al. (43) Pub. Date: Oct. 18, 2007

(54) STITCHING OF PATHS FOR IMPROVED TEXT-ON-PATH RENDERING OF MAP LABELS

(75) Inventors: **Gordon Gregory BOWMAN**, Kemptville (CA); **Gerhard Dietrich KLASSEN**, Waterloo (CA); **Eric JOHNSON**, Ottawa (CA)

Correspondence Address:
**OGILVY RENAULT LLP**
**1981 MCGILL COLLEGE AVENUE**
**SUITE 1600**
**MONTREAL, QC H3A2Y3 (CA)**

(73) Assignee: **RESEARCH IN MOTION LIMITED**, Waterloo (CA)

(21) Appl. No.: **11/691,257**

(22) Filed: **Mar. 26, 2007**

**Related U.S. Application Data**

(60) Provisional application No. 60/788,434, filed on Mar. 31, 2006. Provisional application No. 60/787,541, filed on Mar. 31, 2006.

**Publication Classification**

(51) **Int. Cl.**
*G09G 5/00* (2006.01)

(52) **U.S. Cl.** .......................................... **345/629**; 345/619

(57) **ABSTRACT**

A wireless communications device obtains sets of map data for rendering portions of a map on a display of the device. The map data includes label data for rendering labels on the map for identifying map features. Once the map data is obtained, the wireless device generates a list of all the labels to be rendered on the map, and for each duplicated label in the list, determines whether the map features associated with the duplicated labels connect on the map. If the labels connect, the device then generates a reconstructed map feature and renders a single instance of the label for the map feature. Accordingly, a map feature, such as a path, can be rendered with a single label even if the map data for the map feature, including its associated labels, was transmitted over-the-air as discrete sets of data.

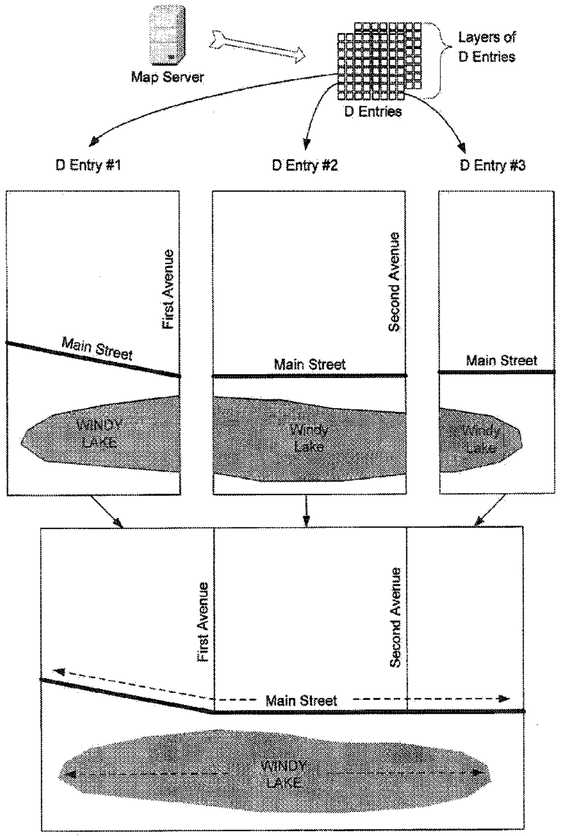
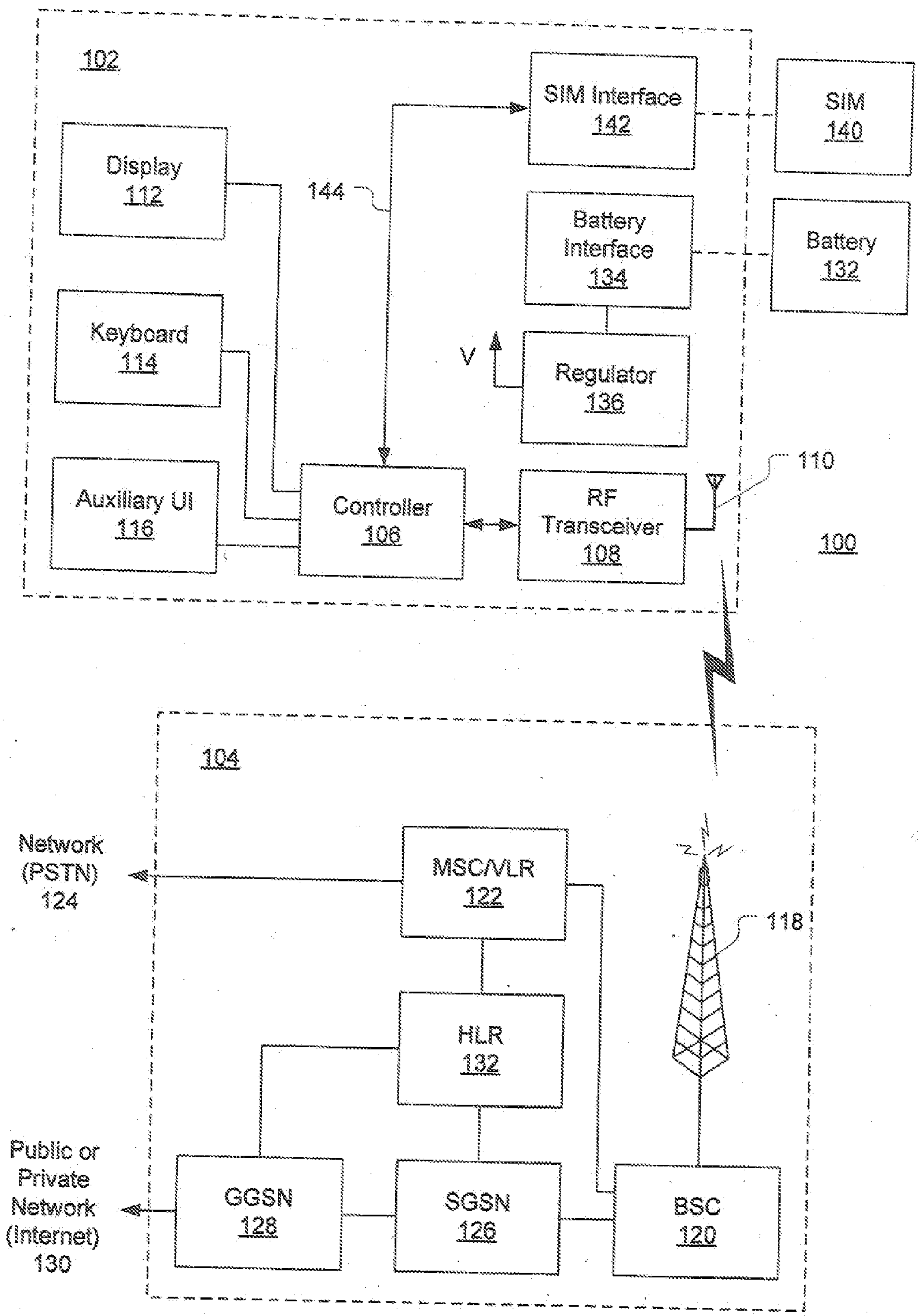


FIG. 1


102
Display
112
Keyboard
114
Auxiliary UI
116
144
Controller
106
SIM Interface
142
SIM
140
Battery
Interface
134
Battery
132
V
Regulator
136
RF
Transceiver
108
110
100
104
Network
(PSTN)
124
MSC/VLR
122
118
HLR
132
Public or
Private
Network
(Internet)
130
GGSN
128
SGSN
126
BSC
120

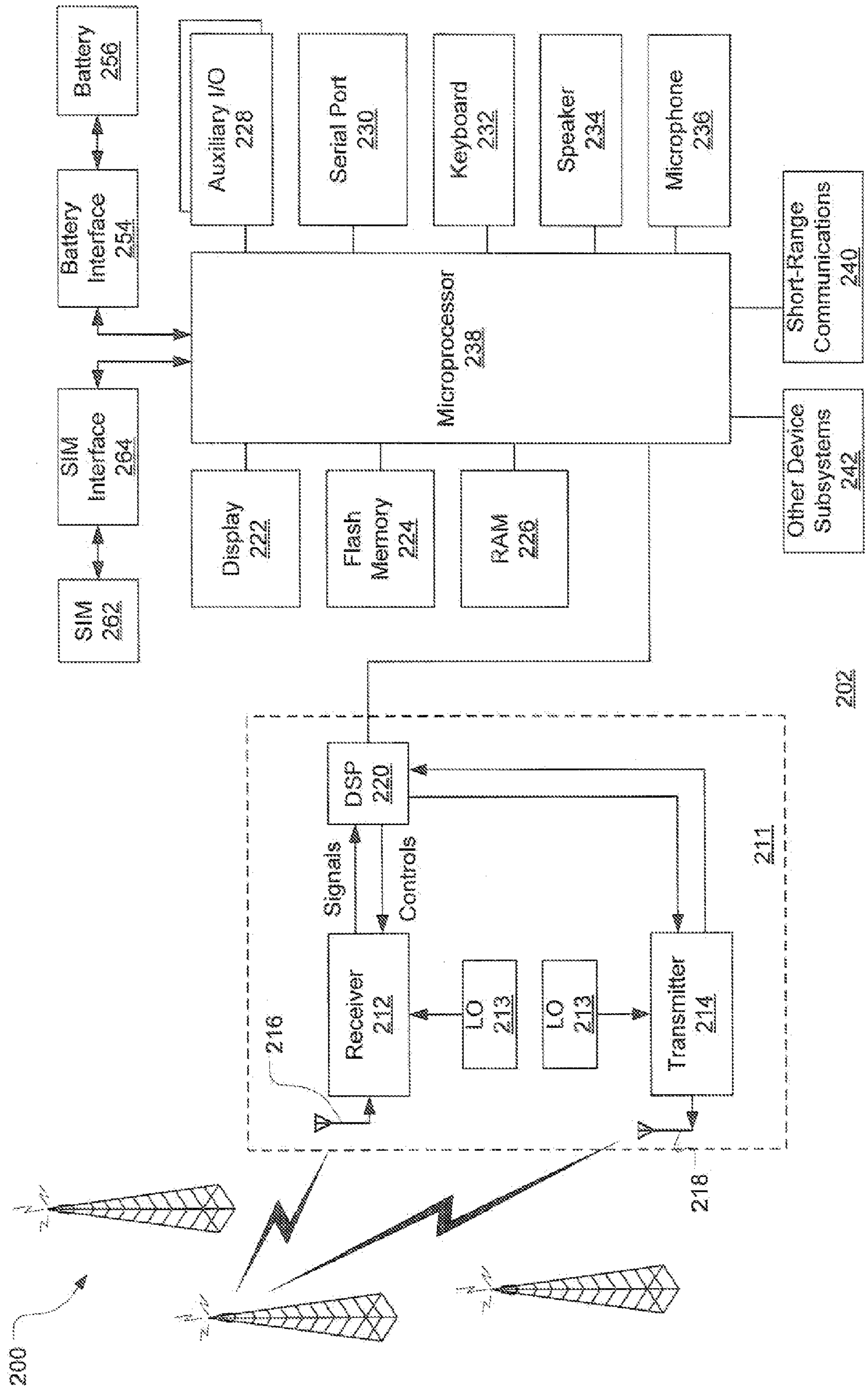

Battery
256
Battery
Interface
254
SIM
Interface
264
SIM
262
Auxiliary I/O
228
Serial Port
230
Keyboard
232
Speaker
234
Microphone
236
Microprocessor
238
Short-Range
Communications
240
Other Device
Subsystems
242
Display
222
Flash
Memory
224
RAM
226
DSP
220
Signals
Controls
Receiver
212
LO
213
LO
213
Transmitter
214
216
218
211
202
200

FIG. 2

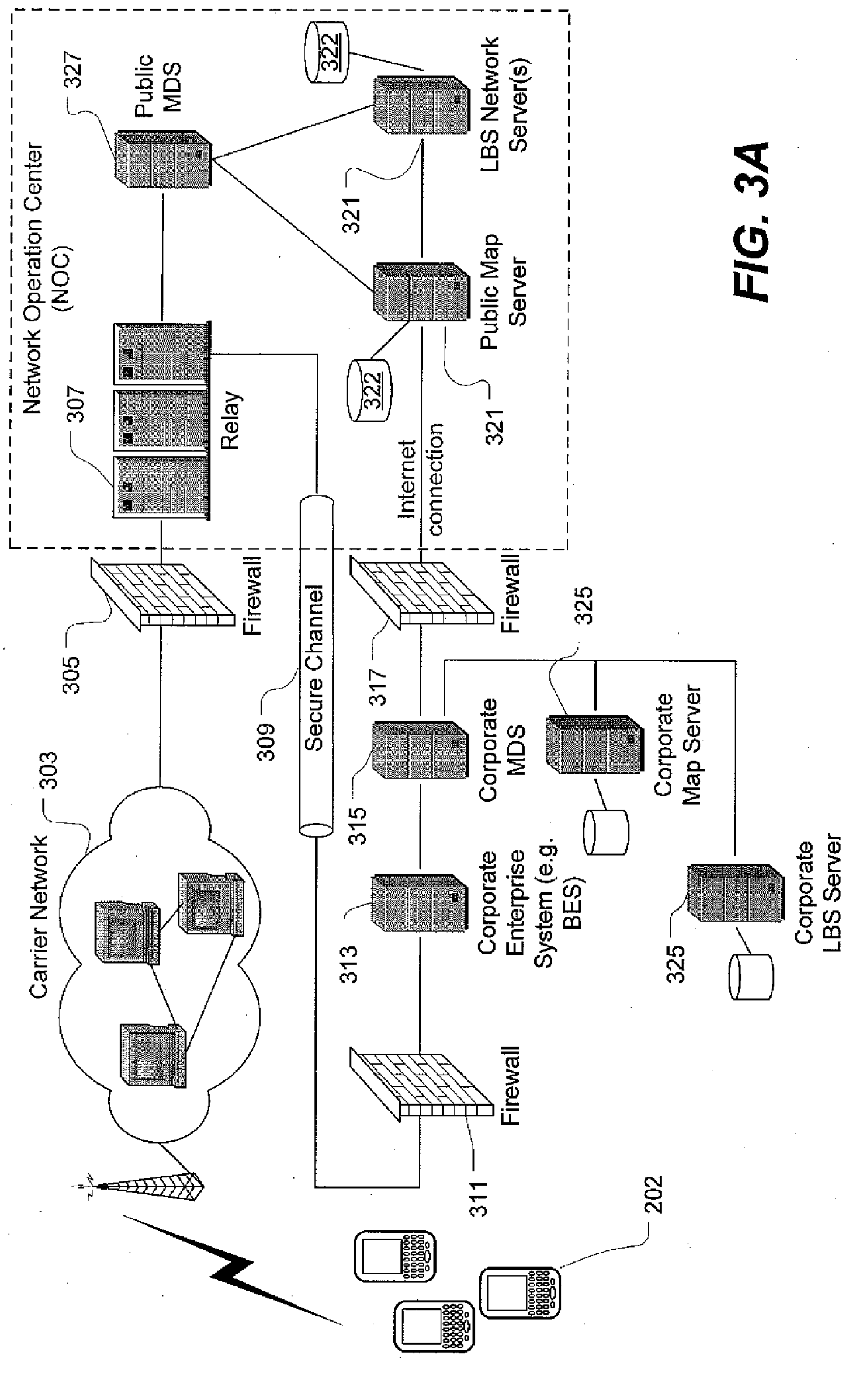
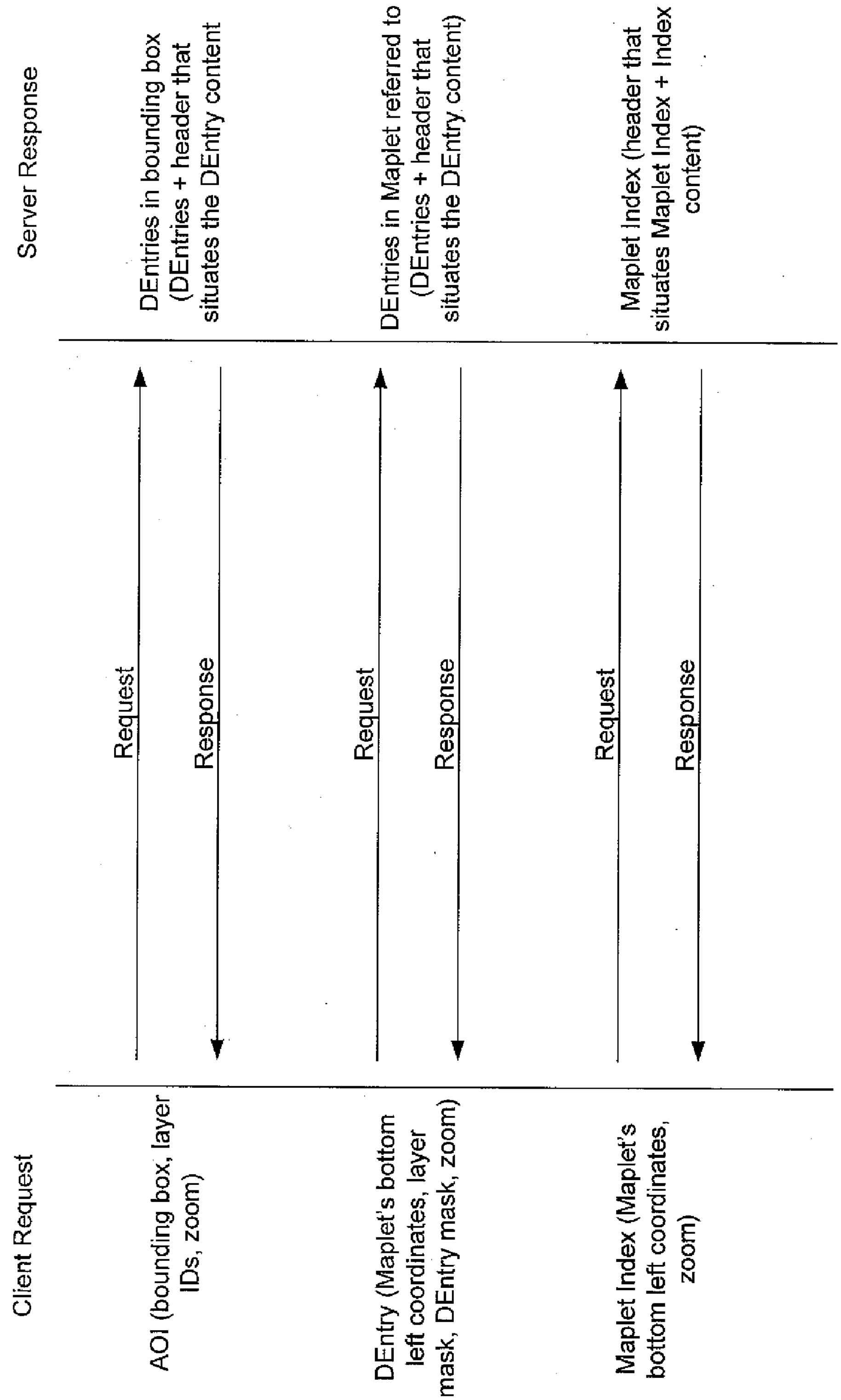

Network Operation Center (NOC)
307
Relay
327
Public MDS
322
LBS Network Server(s)
321
Public Map Server
322
321
Internet connection
Carrier Network
303
305
Firewall
309
Secure Channel
317
Firewall
325
Corporate MDS
315
Corporate Map Server
Corporate Enterprise System (e.g. BES)
313
Corporate LBS Server
325
Firewall
311
202

FIG. 3A



Client Request
Server Response
AOI (bounding box, layer IDs, zoom)
Request
Response
DEntries in bounding box (DEntries + header that situates the DEntry content
DEntry (Maplet's bottom left coordinates, layer mask, DEntry mask, zoom)
Request
Response
DEntries in Maplet referred to (DEntries + header that situates the DEntry content)
Maplet Index (Maplet's bottom left coordinates, zoom)
Request
Response
Maplet Index (header that situates Maplet Index + Index content)


*FIG. 3B*

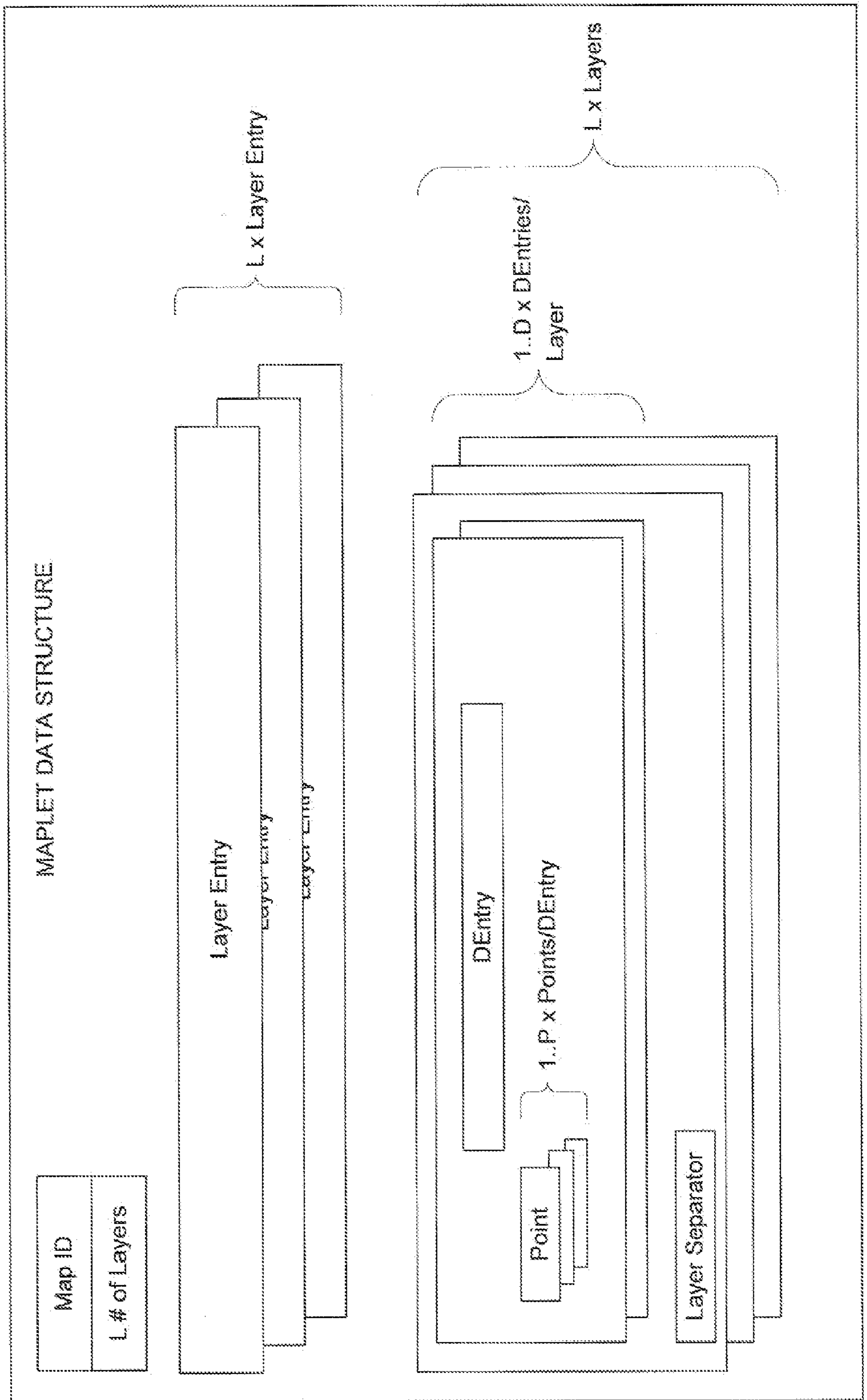
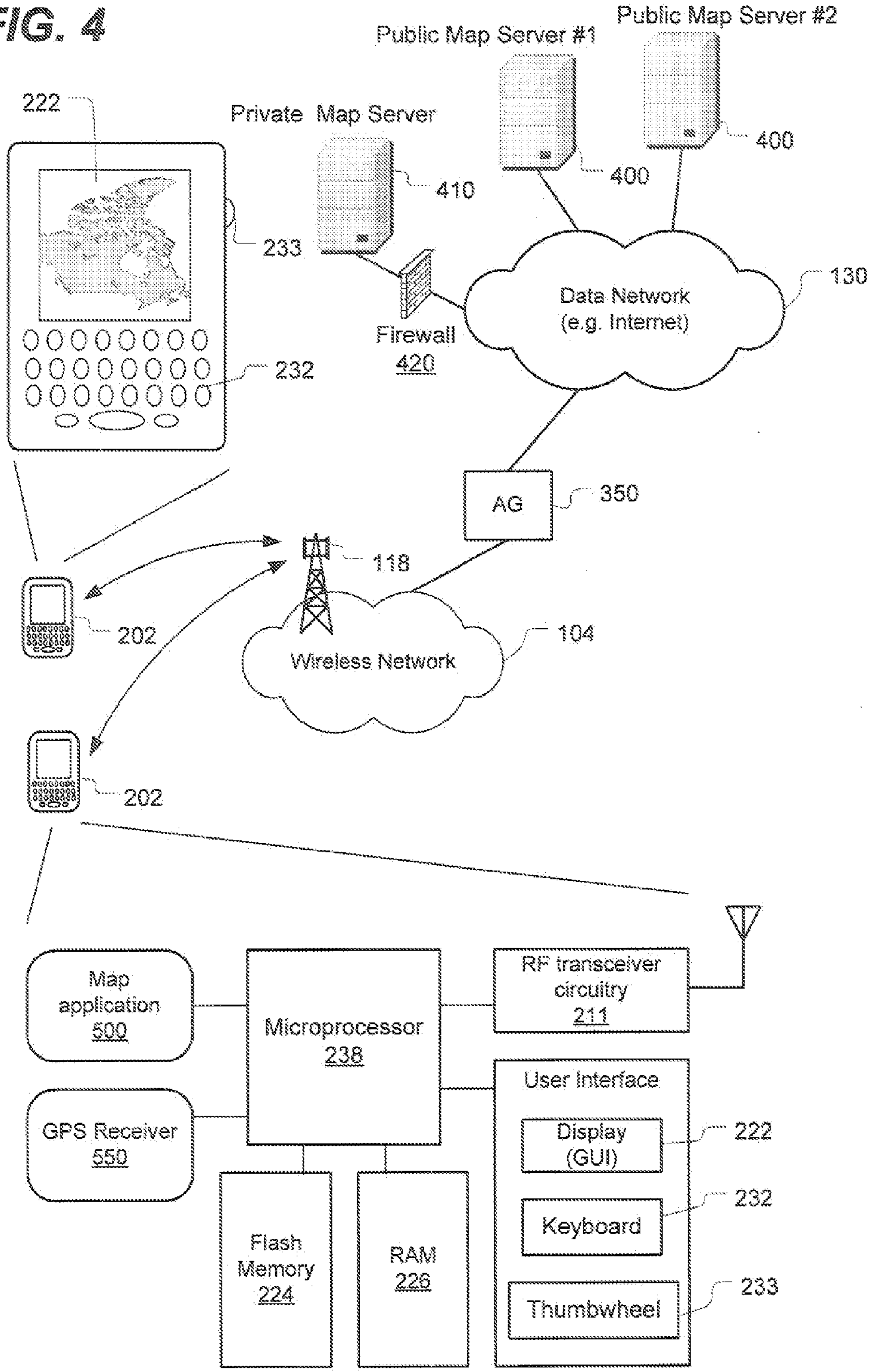

MAPLET DATA STRUCTURE
Map ID
L # of Layers
Layer Entry
Layer Entry
Layer Entry
L x Layer Entry
1..D x DEntries/
Layer
L x Layers
DEntry
1..P x Points/DEntry
Point
Layer Separator

FIG. 3C

FIG. 4



222
233
232
Private Map Server
410
Public Map Server #1
400
Public Map Server #2
400
Firewall
420
Data Network
(e.g. Internet)
130
AG
350
118
202
104
Wireless Network
202
Map application
500
GPS Receiver
550
Microprocessor
238
RF transceiver circuitry
211
User Interface
Display (GUI)
222
Keyboard
232
Thumbwheel
233
Flash Memory
224
RAM
226

FIG. 5

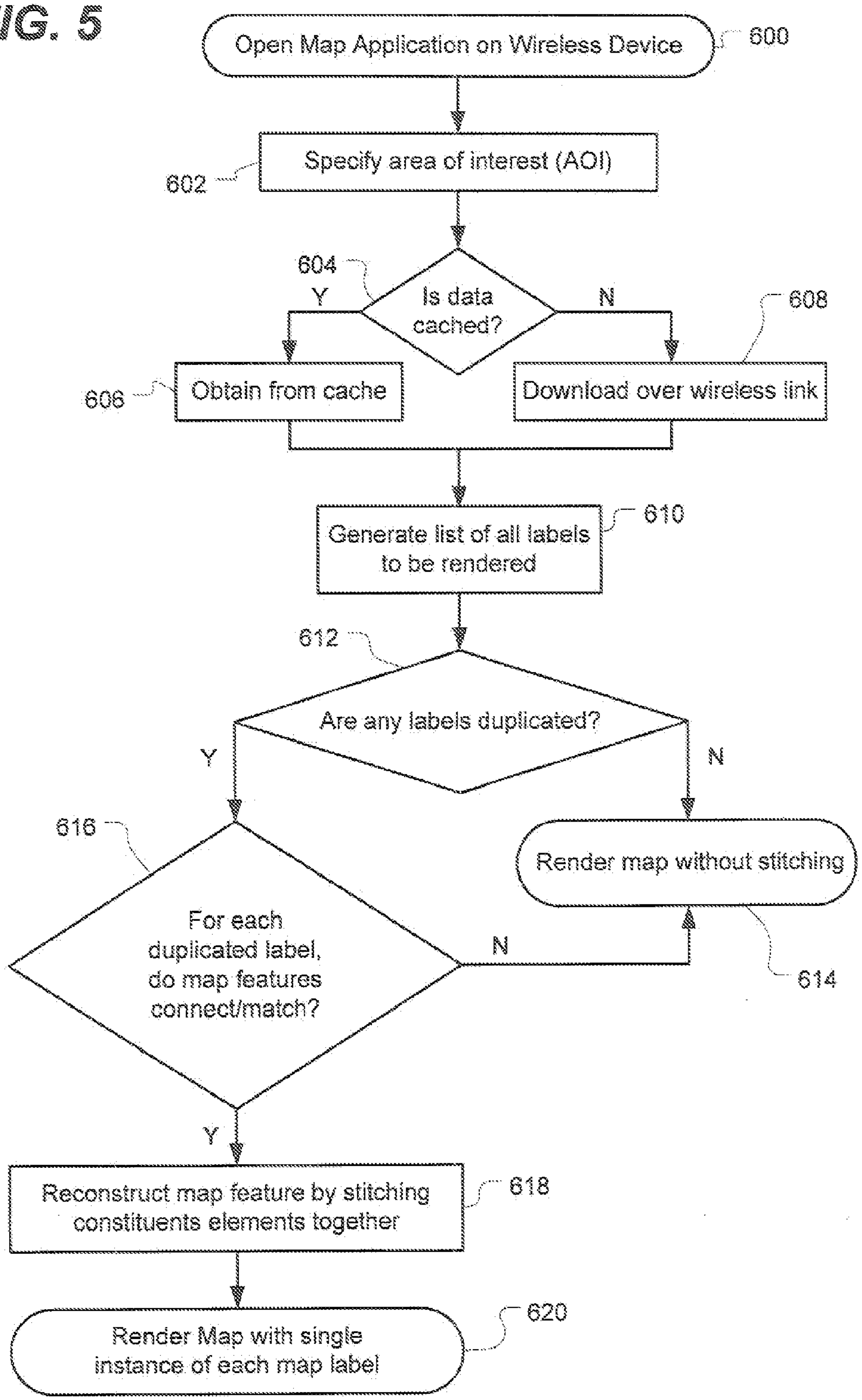

Open Map Application on Wireless Device
600
Specify area of interest (AOI)
602
604
Is data cached?
Y
N
608
Obtain from cache
606
Download over wireless link
Generate list of all labels to be rendered
610
612
Are any labels duplicated?
Y
N
616
Render map without stitching
For each duplicated label, do map features connect/match?
N
614
Y
Reconstruct map feature by stitching constituents elements together
618
Render Map with single instance of each map label
620

FIG. 6

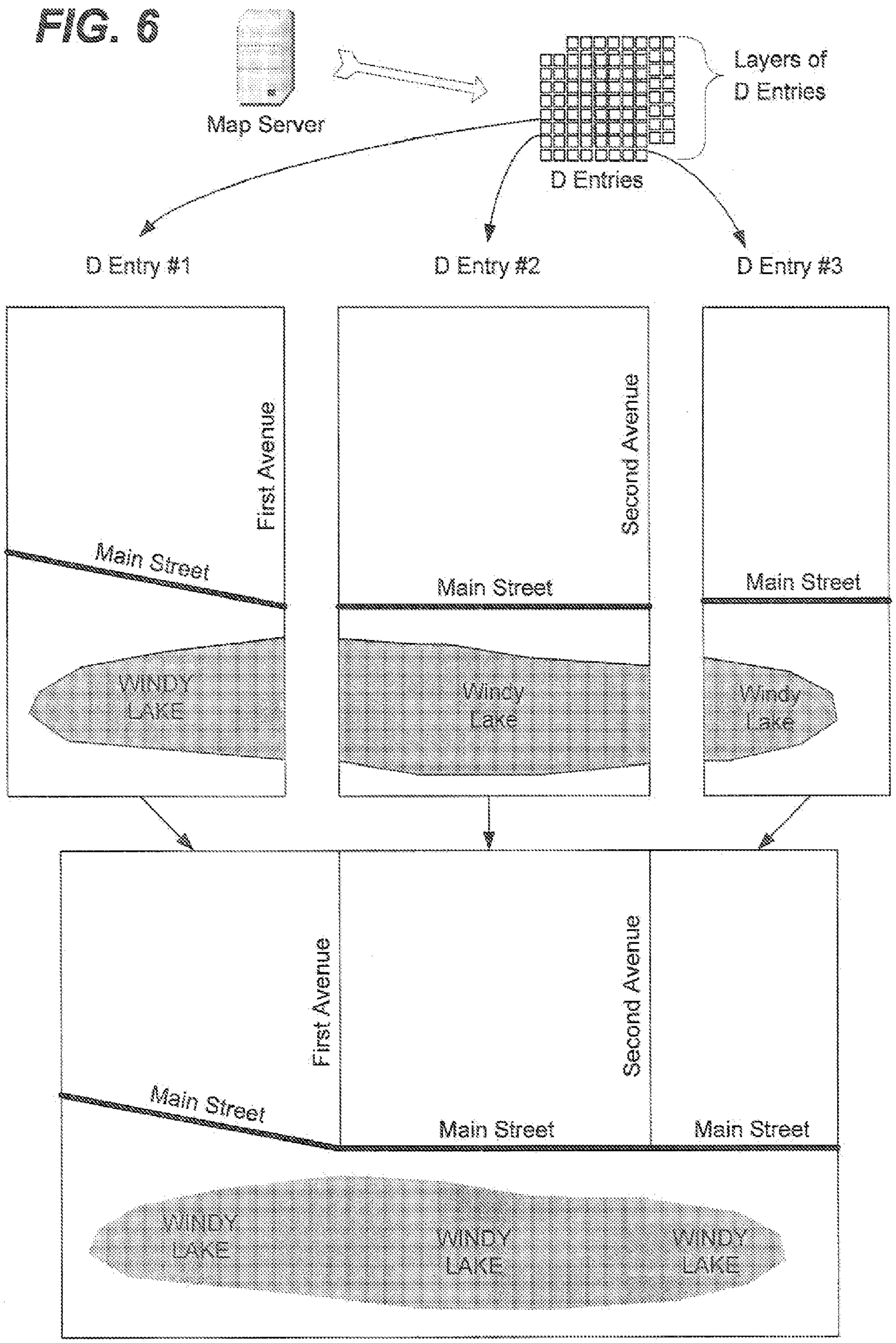

Map Server
Layers of D Entries
D Entries
D Entry #1
D Entry #2
D Entry #3
First Avenue
Main Street
WINDY LAKE
Second Avenue
Main Street
Windy Lake
Main Street
Windy Lake
First Avenue
Main Street
Second Avenue
Main Street
Main Street
WINDY LAKE
WINDY LAKE
WINDY LAKE

FIG. 7

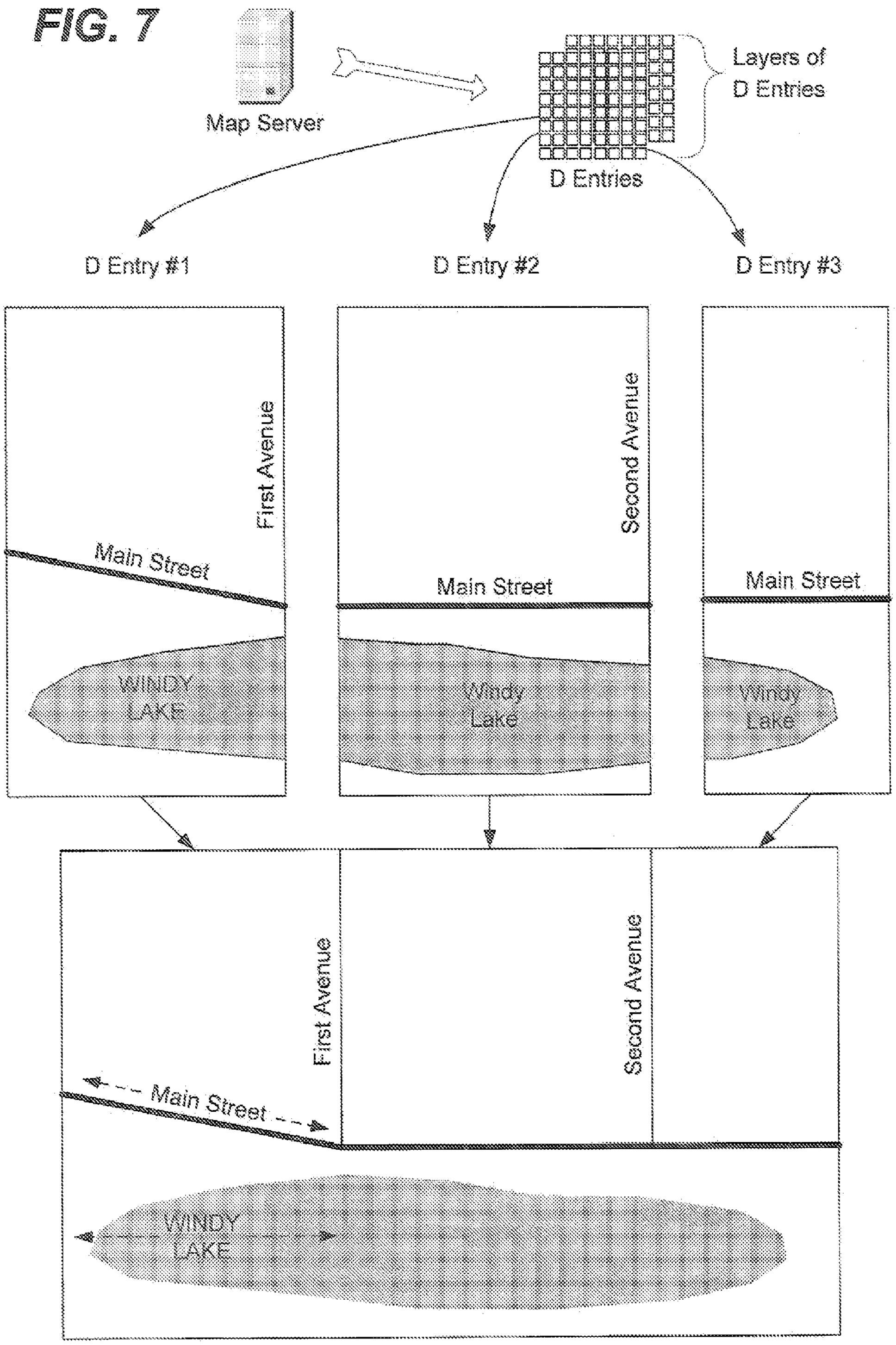

Map Server
Layers of D Entries
D Entries
D Entry #1
D Entry #2
D Entry #3
First Avenue
Main Street
WINDY LAKE
Second Avenue
Main Street
Windy Lake
Main Street
Windy Lake
First Avenue
Main Street
Second Avenue
WINDY LAKE

FIG. 8

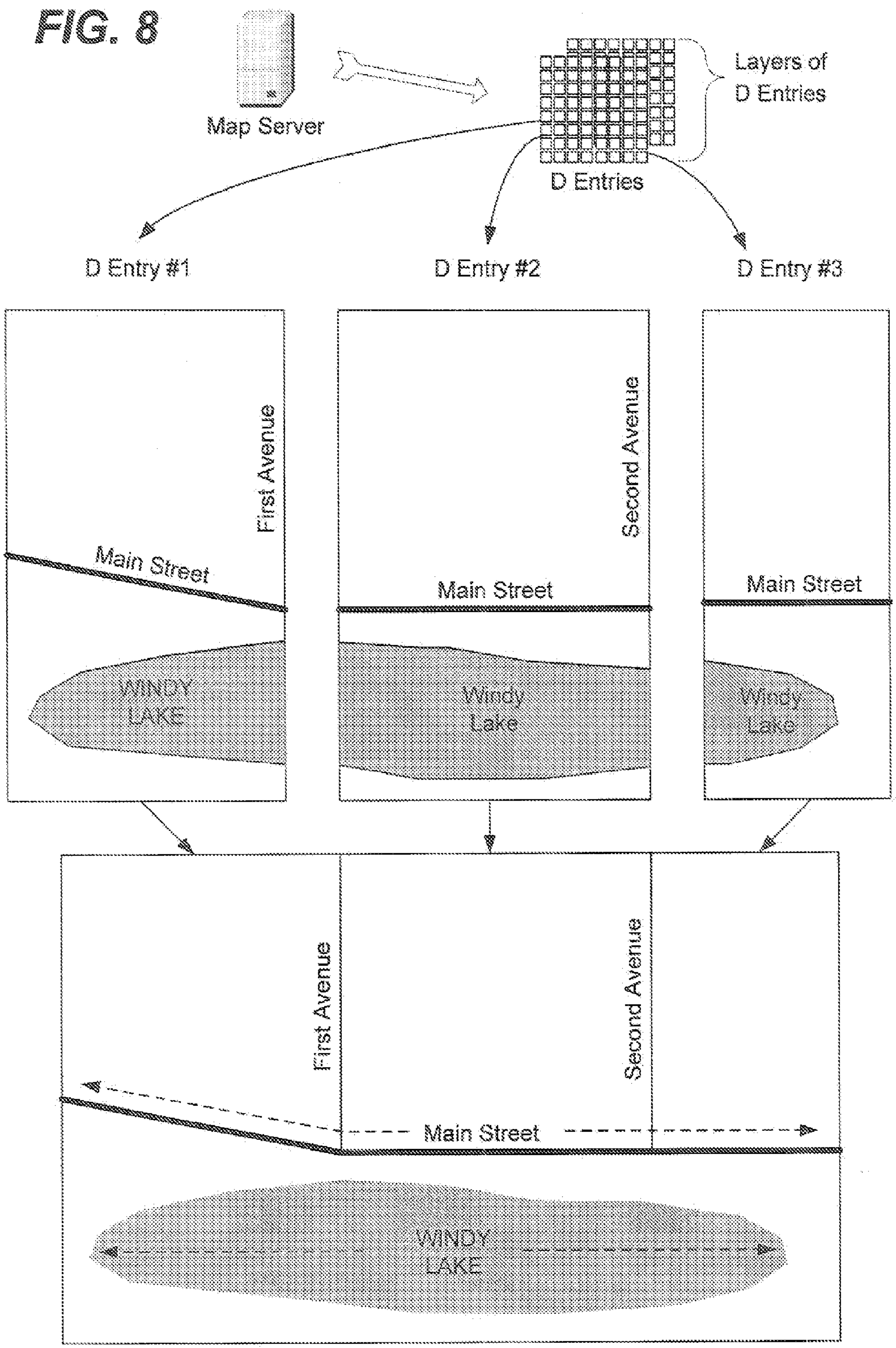
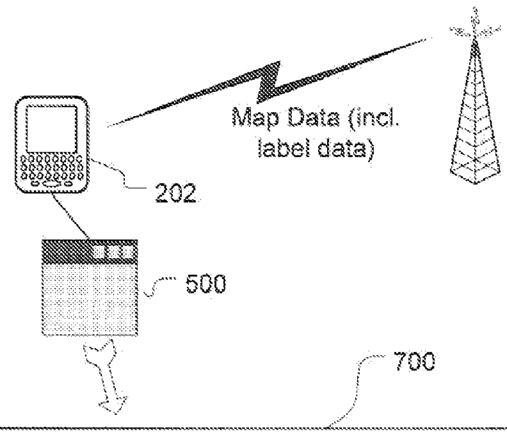

Map Server
Layers of D Entries
D Entries
D Entry #1
D Entry #2
D Entry #3
First Avenue
Main Street
WINDY LAKE
Second Avenue
Main Street
Windy Lake
Main Street
Windy Lake
First Avenue
Second Avenue
Main Street
WINDY LAKE

FIG. 9A




| Label List | Link | Vector | Flag |
|---|---|---|---|
| First Avenue | None | (0,-1) | No |
| Main Street | Next | (1,-1) | No |
| Main Street | Previous | (1,0) | No |
| Main Street | Previous | (-1,0) | Yes |
| Second Avenue | None | (0,1) | No |
| Windy Lake | Next | (1,0) | No |
| Windy Lake | Previous | (1,0) | No |
| Windy Lake | Previous | (1,0) | No |

FIG. 9B

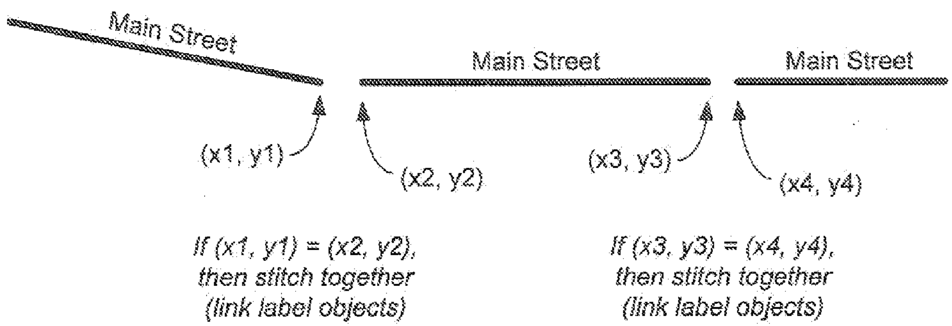



FIG. 9C

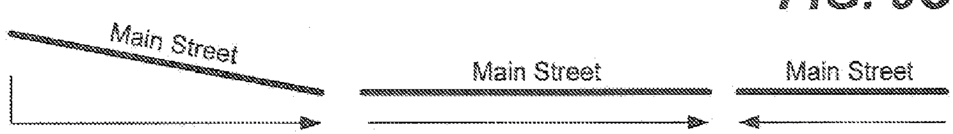


FIG. 10

Without stitching:

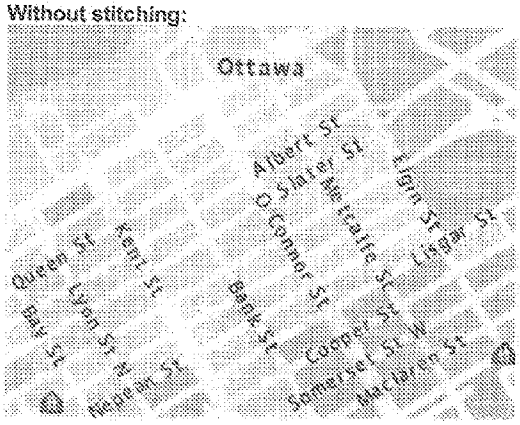
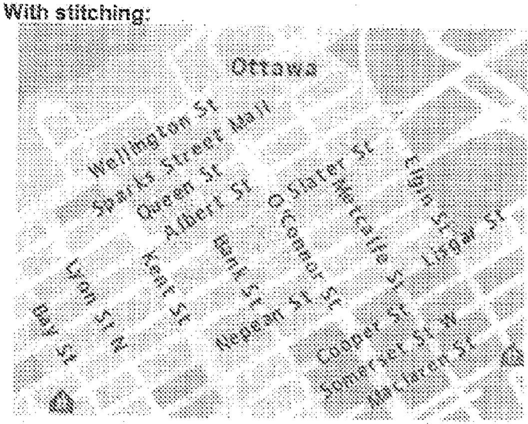



FIG. 11

With stitching:



# STITCHING OF PATHS FOR IMPROVED TEXT-ON-PATH RENDERING OF MAP LABELS

## CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority from U.S. Provisional Patent Application No. 60/788,434 entitled "Methods and Apparatus for Dynamically Labelling Map Objects in Visually Displayed Maps of Mobile Communication Devices" filed on Mar. 31, 2006 and from U.S. Provisional Patent Application No. 60/787,541 entitled "Method and System for Distribution of Map Content to Mobile Communication Devices" filed on Mar. 31, 2006.

## TECHNICAL FIELD

[0002] The present disclosure relates generally to wireless communications devices and, in particular, to techniques for generating map content on wireless communications devices.

## BACKGROUND

[0003] Wireless communications devices such as the BlackBerry™ by Research in Motion Limited enable users to download map content from web-based data sources such as BlackBerry Maps™, Google Maps™ or Mapquest™. Downloaded map content is displayed on a small LCD display screen of the wireless communications device for viewing by the user. The user can pan up and down and side to side as well as zoom in or out. Due to the small display on the device and due to the limited over-the-air (OTA) bandwidth, there is a need to optimize the delivery and handling of the map data.

[0004] With the increasing availability of wireless communications devices having onboard Global Positioning System (GPS) receivers for providing location-based services (LBS), the efficient delivery and handling of map data is increasingly important.

[0005] Map data, including label data for labelling map features, is communicated from map servers to wireless communications devices in discrete portions which are assembled client-side to provide the map content requested by the user. However, when reconstructing a map from discrete portions of data, however, redundant labelling can occur if labels associated with each portion of data are rendered for the same feature. Furthermore, even if redundant labels are suppressed, the label associated with a reconstructed map feature cannot be placed aesthetically in the center of the reconstructed feature but rather is placed in the center of one of the constituent elements of the feature.

[0006] Accordingly, a technique for efficiently and aesthetically labelling maps reconstructed from discrete portions of downloaded data remains highly desirable.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Further features and advantages of the present technology will become apparent from the following detailed description, taken in combination with the appended drawings, in which:

[0008] FIG. **1** is a block diagram schematically illustrating pertinent components of a wireless communications device and of a wireless communications network;

[0009] FIG. **2** is a more detailed block diagram of a wireless communications device;

[0010] FIG. **3**A is a system diagram of network components which provide mapping functionality in the wireless communications devices of FIG. **1** and FIG. **2**;

[0011] FIG. **3**B illustrates a message exchange between a wireless communications device and a map server for downloading map content to the wireless communications device based on the system of FIG. **3**A;

[0012] FIG. **3**C is a diagram showing a preferred Maplet data structure;

[0013] FIG. **4** is a schematic depiction of a wireless network having an applications gateway for optimizing the downloading of map data from map servers to wireless communications devices;

[0014] FIG. **5** is a flowchart presenting steps of a method of displaying a map on a wireless device by stitching together constituent path segments to generate a reconstructed path having a single label associated with the reconstructed path;

[0015] FIG. **6** schematically depicts the potential problem of redundant labelling that may be encountered when map features are rendered from discrete sets of map data;

[0016] FIG. **7** schematically depicts the potential problems of having both poorly placed labels and highly constrained labels that may be encountered when map features are rendered from discrete sets of map data by blindly squelching duplicated labels;

[0017] FIG. **8** schematically depicts a process of stitching together path segments (and constituent elements of other map features) to create reconstructed paths (and map features);

[0018] FIG. **9**A schematically depicts constructing a label list with link, vector, and flag information for efficiently stitching together the map features of FIG. **6**;

[0019] FIG. **9**B schematically depicts a process of determining whether an endpoint of one path associated with a duplicated label matches an endpoint of another path having the same duplicated label;

[0020] FIG. **9**C schematically depicts a process of determining vector directionality for vector map data;

[0021] FIG. **10** is a screenshot of a street map of downtown Ottawa, Canada without stitching (reconstruction) of paths; and

[0022] FIG. **11** is a screenshot of a street map of downtown Ottawa, Canada with stitching (reconstruction) of paths.

[0023] It will be noted that throughout the appended drawings, like features are identified by like reference numerals.

## DETAILED DESCRIPTION

[0024] The present technology provides, in general, a method of rendering a map on a display of a wireless communications device where the map data, including label data, is downloaded over-the-air in small, discrete portions or sets of data, each with its own label data for labelling features of the map. The wireless device generates a list of all the labels to be rendered on the map, and for each duplicated label in the list, determines whether the map features associated with the duplicated labels connect or coincide on the map. If the map features associated with the duplicated labels do coincide, a map application running on the device then generates a reconstructed map feature and renders a single (preferably centrally-positioned) instance of the label for the map feature. Accordingly, a map feature, such as a path, can be rendered with a single label even if the map data for the map feature, including its associated labels, was transmitted over-the-air as discrete sets of data containing redundant labels.

**[0025]** Thus, an aspect of the present technology is a method of displaying a map on a wireless communications device. The method includes steps of obtaining sets of map data for rendering portions of the map on a display of the device, the map data including label data for rendering labels on the map for identifying one or more map features and generating a list of all the labels to be rendered on the map. For each duplicated label in the list, the method entails determining whether the map features associated with the duplicated labels connect on the map. Finally, a reconstructed map feature is generated and a single instance of the label is rendered for the map feature.

**[0026]** Another aspect of the present technology is a computer program product that includes code adapted to perform the steps of the foregoing method when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**[0027]** Yet another aspect of the present technology is a wireless communications device for enabling a user of the device to display a map on the device. The wireless device includes an input device for enabling the user to cause the device to obtain map data for rendering the map to be displayed on a display of the device, the map data including label data for rendering labels on the map for identifying one or more map features. The wireless communications device further includes a memory for storing code to instruct a processor to generate a list of all the labels to be rendered on the map, determine, for each duplicated label in the list, whether the map features associated with the duplicated labels connect on the map, generate a reconstructed map feature, and render on the display of the device a single instance of the label for the map feature.

**[0028]** The details and particulars of these aspects of the technology will now be described below, by way of example, with reference to the attached drawings.

**[0029]** FIG. **1** is a block diagram of a communication system **100** which includes a wireless communications device **102** (also referred to as a mobile communications device) which communications through a wireless communication network **104**. For the purposes of the present specification, the expression "wireless communications device" encompasses not only a wireless handheld, cell phone or wireless-enabled laptop but also any mobile communications device or portable communications device such as a satellite phone, wireless-enabled PDA or wireless-enabled MP3 player. In other words, for the purposes of this specification, "wireless" shall be understood as encompassing not only standard cellular or microwave RF technologies, but also any other communications technique that conveys data over the air using an electromagnetic signal.

**[0030]** The wireless communications device **102** preferably includes a visual display **112**, e.g. an LCD screen, a keyboard **114** (or keypad), and optionally one or more auxiliary user interfaces (UI) **116**, each of which is coupled to a controller **106**. The controller **106** is also coupled to radio frequency (RF) transceiver circuitry **108** and an antenna **110**. Typically, controller **106** is embodied as a central processing unit (CPU) which runs operating system software in a memory device (described later with reference to FIG. **2**). Controller **106** normally controls the overall operation of the wireless communications device **102**, whereas signal processing operations associated with communications functions are typically performed in the RF transceiver circuitry **108**. Controller **106** interfaces with the display screen **112** to display received information, stored information, user inputs, and the like. Keyboard/keypad **114**, which may be a telephone-type keypad or a full QWERTY keyboard, is normally provided for entering commands and data.

**[0031]** The wireless communications device **102** sends communication signals to and receives communication signals from network **104** over a wireless link via antenna **110**. RF transceiver circuitry **108** performs functions similar to those of station **118** and Base Station Controller (BSC) **120**, including, for example, modulation and demodulation, encoding and decoding, and encryption and decryption. It will be apparent to those skilled in the art that the RF transceiver circuitry **108** will be adapted to the particular wireless network or networks in which the wireless communications device is intended to operate.

**[0032]** The wireless communications device **102** includes a battery interface **134** for receiving one or more rechargeable batteries **132**. Battery **132** provides electrical power to electrical circuitry in the device **102**, and battery interface **134** provides for a mechanical and electrical connection for battery **132**. Battery interface **134** is couple to a regulator **136** which regulates power to the device. When the wireless device **102** is fully operationally, an RF transmitter of RF transceiver circuitry **108** is typically keyed or turned on only when it is sending to network, and is otherwise turned off to conserve resources. Similarly, an RF receiver of RF transceiver circuitry **108** is typically periodically turned off to conserve power until it is needed to receive signals or information (if at all) during designated time periods.

**[0033]** Wireless communications device **102** operates using a Subscriber Identity Module (SIM) **140** which is connected to or inserted in the wireless communications device **102** at a SIM interface **142**. SIM **140** is one type of a conventional "smart card" used to identify an end user (or subscriber) of wireless device **102** and to personalize the device, among other things. Without SIM **140**, the wireless communications device **102** is not fully operational for communication through wireless network **104**. By inserting the SIM card **140** into the wireless communications device **102**, an end user can have access to any and all of his subscribed services. SIM **140** generally includes a processor and memory for storing information. Since SIM **140** is coupled to SIM interface **142**, it is coupled to controller **106** through communication lines **144**. In order to identify the subscriber, SIM **140** contains some user parameters such as an International Mobile Subscriber Identity (IMSI). An advantage of using SIM **140** is that end users are not necessarily bound by any single physical wireless device.

SIM **140** may store additional user information for the wireless device as well, including datebook (calendar) information and recent call information.

[0034] The wireless communications device **102** may consist of a single unit, such as a data communication device, a cellular telephone, a Global Positioning System (GPS) unit, a multiple-function communication device with data and voice communication capabilities, a wireless-enabled personal digital assistant (PDA), or a wireless-enabled laptop computer. Alternatively, the wireless communications device **102** may be a multiple-module unit comprising a plurality of separate components, including but in no way limited to a computer or other device connected to a wireless modem. In particular, for example, in the block diagram of FIG. **1**, RF circuitry **108** and antenna **110** may be implemented as a radio modem unit that may be inserted into a port on a laptop computer. In this case, the laptop computer would include display **112**, keyboard **114**, one or more auxiliary UIs **116**, and controller **106** embodied as the computer's CPU.

[0035] The wireless communications device **102** communicates in and through a wireless communication network **104**. The wireless communication network may be a cellular telecommunications network. In the example presented in FIG. **1**, wireless network **104** is configured in accordance with Global Systems for Mobile communications (GSM) and General Packet Radio Service (GPRS) technologies. Although wireless communication network **104** is described herein as a CSM/GPRS-type network, any suitable network technologies may be utilized such as Code Division Multiple Access (CDMA), Wideband CDMA (WCDMA), whether 2G, 3G, or Universal Mobile Telecommunication System (UMTS) based technologies. In this example, the GSM/GPRS wireless network **104** includes a base station controller (BSC) **120** with an associated tower station **118**, a Mobile Switching Center (MSC) **122**, a Home Location Register (HER) **132**, a Serving General Packet Radio Service (GPRS) Support Node (SGSN) **126**, and a Gateway GPRS Support Node (GGSN) **128**. MSC **122** is coupled to BSC **120** and to a landline network, such as a Public Switched Telephone Network (PSTN) **124**. SGSN **126** is coupled to BSC **120** and to GGSN **128**, which is, in turn, coupled to a public or private data network **130** (such as the Internet). HELR **132** is coupled to MSC **122**, SGSN **126** and GGSN **128**.

[0036] Tower station **118** is a fixed transceiver station. Tower station **118** and BSC **120** may be referred to as transceiver equipment. The transceiver equipment provides wireless network coverage for a particular coverage area commonly referred to as a "cell". The transceiver equipment transmits communication signals to and receives communication signals from wireless communications devices **102** within its cell via station **118**. The transceiver equipment normally performs such functions as modulation and possibly encoding and/or encryption of signals to be transmitted to the wireless communications device in accordance with particular, usually predetermined, communication protocols and parameters. The transceiver equipment similar demodulates and possibly decodes and decrypts, if necessary, any communication signals received from the wireless communications device **102** transmitting within its cell. Communication protocols and parameters may vary between different networks. For example, one network may employ a different modulation scheme and operate at different frequencies than other networks.

[0037] The wireless link shown in communication system **100** of FIG. **1** represents one or more different channels, typically different radio frequency (RF) channels, and associated protocols used between wireless network **104** and wireless communications device **102**. An RF channel is a limited resource that must be conserved, typically due limits in overall bandwidth and a limited battery power of the wireless device **102**. Those skilled in the art will appreciate that a wireless network in actual practice may include hundreds of cells, each served by a station **118**, depending upon desired overall expanse of network coverage. All pertinent components may be connected by multiple switches and routers (not shown), controlled by multiple network controllers.

[0038] For all wireless communications devices **102** registered with a network operator, permanent data (such as the user profile associated with each device) as well as temporary data (such as the current location of the device) are stored in the HLR **132**. In case of a voice call to the wireless device **102**, the HLR **132** is queried to determine the current location of the device **102**. A Visitor Location Register (VLR) of MSC **122** is responsible for a group of location areas and stores the data of those wireless devices that are currently in its area of responsibility. This includes parts of the permanent data that have been transmitted from HLR **132** to the VLR for faster access. However, the VLR of MSC **122** may also assign and store local data, such as temporary identifications. Optionally, the VLR of MSC **122** can be enhanced for more efficient co-ordination of GPRS and non-GPRS services and functionality (e.g. paging for circuit-switched calls which can be performed more efficiently via SGSN **126**, and combined GPRS and non-GPRS location updates).

[0039] Serving GPRS Support Node (SGSN) **126** is at the same hierarchical level as MSC **122** and keeps track of the individual locations of wireless devices **102**. SGSN **126** also performs security functions and access control. Gateway GPRS Support Node (GGSN) **128** provides internetworking with external packet-switched networks and is connected with SGSNs (such as SCSN **126**) via an IP-based GPRS backbone network. SGSN **126** performs authentication and cipher setting procedures based on the same algorithms, keys, and criteria as in existing GSM. In conventional operation, cell selection may be performed autonomously by wireless device **102** or by the transceiver equipment instructing the wireless device to select a particular cell. The wireless device **102** informs wireless network **104** when it reselects another cell or group of cells, known as a routing area.

[0040] In order to access GPRS services, the wireless device **102** first makes its presence known to wireless network **104** by performing what is known as a GPRS "attach". This operation establishes a logical link between the wireless device **102** and SGSN **126** and makes the wireless device **102** available to receive, for example, pages via SGSN, notifications of incoming GPRS data, or SMS messages over GPRS. In order to send and receive GPRS data, the wireless device **102** assists in activating the packet data address that it wants to use. This operation makes the wireless device **102** known to GCSN **128**; internetworking with external data networks can thereafter commence. User data may be transferred transparently between the wireless device **102** and the external data networks using, for example, encapsulation and tunnelling. Data packets are equipped with GPRS-specific protocol information and transferred between wireless device **102** and GGSN **128**.

[**0041**] Those skilled in the art will appreciate that a wireless network may be connected to other systems, possibly including other networks, not explicitly shown in FIG. **1**. A network will normally be transmitting at very least some sort of paging and system information on an ongoing basis, even if there is no actual packet data exchanged. Although the network consists of many parts, these parts all work together to result in certain behaviours at the wireless link.

[**0042**] FIG. **2** is a detailed block diagram of a preferred wireless communications device **202**. The wireless device **202** is preferably a two-way communication device having at least voice and advanced data communication capabilities, including the capability to communicate with other computer systems. Depending on the functionality provided by the wireless device **202**, it may be referred to as a data messaging device, a two-way pager, a cellular telephone with data message capabilities, a wireless Internet appliance, or a data communications device (with or without telephony capabilities). The wireless device **202** may communicate with any one of a plurality of fixed transceiver stations **200** within its geographic coverage area.

[**0043**] The wireless communications device **202** will normally incorporate a communication subsystem **211**, which includes a receiver **212**, a transmitter **214**, and associated components, such as one or more (preferably embedded or internal) antenna elements **216** and **218**, local oscillators (LO's) **213**, and a processing module such as a digital signal processor (DSP) **220**. Communication subsystem **211** is analogous to RF transceiver circuitry **108** and antenna **110** shown in FIG. **1**. As will be apparent to those skilled in the field of communications, the particular design of communication subsystem **211** depends on the communication network in which the wireless device **202** is intended to operate.

[**0044**] The wireless device **202** may send and receive communication signals over the network after required network registration or activation procedures have been completed. Signals received by antenna **216** through the network are input to receiver **212**, which may perform common receiver functions as signal amplification, frequency down conversion, filtering, channel selection, and the like, and, as shown in the example of FIG. **2**, analog-to-digital (A/D) conversion. A/D conversion of a received signal allows more complex communication functions such as demodulation and decoding to performed in the DSP **220**. In a similar manner, signals to be transmitted are processed, including modulation and encoding, for example, by DSP **220**. These DSP-processed signals are input to transmitter **214** for digital-to-analog (D/A) conversion, frequency up conversion, filtering, amplification and transmission over communication network via antenna **218**. DSP **220** not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in receiver **212** and transmitter **214** may be adaptively controlled through automatic gain control algorithms implemented in the DSP **220**.

[**0045**] Network access is associated with a subscriber or user of the wireless device **202**, and therefore the wireless device requires a Subscriber Identity Module or SIM card **262** to be inserted in a SIM interface **264** in order to operate in the network. SIM **262** includes those features described in relation to FIG. **1**. Wireless device **202** is a battery-powered device so it also includes a battery interface **254** for receiving one or more rechargeable batteries **256**. Such a battery **256** provides electrical power to most if not all electrical circuitry in the device **102**, and battery interface provides for a mechanical and electrical connection for it. The battery interface **254** is coupled to a regulator (not shown) which provides a regulated voltage V to all of the circuitry.

[**0046**] Wireless communications device **202** includes a microprocessor **238** (which is one implementation of controller **106** of FIG. **1**) which controls overall operation of wireless device **202**. Communication functions, including at least data and voice communications, are performed through communication subsystem **211**. Microprocessor **238** also interacts with additional device subsystems such as a display **222**, a flash memory **224**, a random access memory (RAM) **226**, auxiliary input/output (I/O) subsystems **228**, a serial port **230**, a keyboard **232**, a speaker **234**, a microphone **236**, a short-range communications subsystem **240**, and any other device subsystems generally designated at **242**. Some of the subsystems shown in FIG. **2** perform communication-related functions, whereas other subsystems may provide "resident" or on-board functions. Notably, some subsystems, such as keyboard **232** and display **222**, for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list. Operating system software used by the microprocessor **238** is preferably stored in a persistent (non-volatile) store such as flash memory **224**, which may alternatively be a read-only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device applications, or parts thereof, may be temporarily loaded into a volatile store such as RAM **226**.

[**0047**] Microprocessor **238**, in addition to its operating system functions, enables execution of software applications on the wireless device **202**. A predetermined set of applications which control basic device operations, including at least data and voice communication applications, will normally be installed on the device **202** during its manufacture. For example, the device may be pre-loaded with a personal information manager (PIM) having the ability to organize and manage data items relating to the user's profile, such as e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores are available on the device **202** and SIM **256** to facilitate storage of PIM data items and other information.

[**0048**] The PIM application preferably has the ability to send and receive data items via the wireless network. PIM data items may be seamlessly integrated, synchronized, and updated via the wireless network, with the wireless device user's corresponding data items stored and/or associated with a host computer system thereby creating a mirrored host computer on the wireless device **202** with respect to such items. This is especially advantageous where the host computer system is the wireless device user's office computer system. Additional applications may also be loaded into the memory store(s) of the wireless communications device **202** through the wireless network, the auxiliary I/O subsystem **228**, the serial port **230**, short-range communications subsystem **240**, or any other suitable subsystem **242**, and installed by a user in RAM **226** or preferably a non-volatile store (not shown) for execution by the microprocessor **238**. Such flexibility in application installation increases the functionality of the wireless device **202** and may provide enhanced onboard functions, communication-related functions or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the wireless device **202**.

**[0049]** In a data communication mode, a received signal such as a text message, an e-mail message, or a web page download will be processed by communication subsystem **211** and input to microprocessor **238**. Microprocessor **238** will preferably further process the signal for output to display **222** or alternatively to auxiliary I/O device **228**. A user of the wireless device **202** may also compose data items, such as email messages, for example, using keyboard **232** in conjunction with display **222** and possibly auxiliary I/O device **228**. Keyboard **232** is preferably a complete alphanumeric keyboard and/or telephone-type keypad. These composed items may be transmitted over a communication network through communication subsystem **211**.

**[0050]** For voice communications, the overall operation of the wireless communications device **202** is substantially similar, except that the received signals would be output to speaker **234** and signals for transmission would be generated by microphone **236**. Alternative voice or audio I/O subsystems, such as a voice message recording subsystem, may also be implemented on the wireless device **202**. Although voice or audio signal output is preferably accomplished primarily through speaker **234**, display **222** may also be used to provide an indication of the identity of the calling party, duration on a voice call, or other voice call related information, as some examples.

**[0051]** Serial port **230** in FIG. **2** is normally implemented in a personal digital assistant (PDA)-type communication device for which synchronization with a user's desktop computer is a desirable, albeit optional, component. Serial port **230** enables a user to set preferences through an external device or software application and extends the capabilities of wireless device **202** by providing for information or software downloads to the wireless device **202** other than through the wireless network. The alternate download path may, for example, be used to load an encryption key onto the wireless device **202** through a direct and thus reliable and trusted connection to thereby provide secure device communications.

**[0052]** Short-range communications subsystem **240** of FIG. **2** is an additional optional component which provides for communication between mobile station **202** and different systems or devices, which need not necessarily be similar devices. For example, subsystem **240** may include an infrared device and associated circuits and components, or a Bluetooth™ communication module to provide for communication with similarly-enabled systems and devices. Bluetooth™ is a trademark of Bluetooth SIG, Inc.

**[0053]** FIG. **3**A is a system diagram of network components which provide mapping functionality in the wireless communication devices of FIGS. **1** and **2**. To achieve this, a mapping application is also provided in memory of the wireless communications device for rendering visual maps in its display. Wireless communications devices **202** are connected over a mobile carrier network **303** for communication through a firewall **305** to a relay **307**. A request for map data from any one of the wireless communications devices **202** is received at relay **307** and passed via a secure channel **309** through firewall **311** to a corporate enterprise server **313** and corporate mobile data system (MDS) server **315**. The request is then passed via firewall **317** to a public map server and/or to a public location-based service (LBS) server **321** which provides location-based services (LBS) to handle the request. The network may include a plurality of such map servers and/or LBS servers where requests are distributed and processed through a load distributing server. The map/LBS data may be stored on this network server **321** in a network database **322**, or may be stored on a separate map server and/or LBS server (not shown). Private corporate data stored on corporate map/LBS server **325** may be added to the public data via corporate MDS server **315** on the secure return path to the wireless device **202**. Alternatively, where no corporate servers are provided, the request from the wireless device **202** may be passed via relay **307** to a public MDS server **327**, which sends the request to the public map/LBS server **321** providing map data or other local-based service in response to the request. For greater clarity, it should be understood that the wireless devices can obtain map data from a "pure" map server offering no location-based services, from an LBS server offering location-based services in addition to map content, or from a combination of servers offering map content and LBS.

**[0054]** A Maplet data structure is provided that contains all of the graphic and labelled content associated with a geographic area (e.g. map features such as restaurants (point features), streets (line features) or lakes (polygon features)). Maplets are structured in Layers of Data Entries ("DEntries") identified by a "Layer ID" to enable data from different sources to be deployed to the device and meshed for proper rendering. Each DEntry is representative of one or more artefact or label (or a combination of both) and includes coordinate information (also referred to as a "bounding box" or "bounding area") to identify the area covered by the DEntry and a plurality of data points that together represent the artefact, feature or label. For example, a DEntry may be used to represent a street on a city map (or a plurality of streets), wherein the carious points within the DEntry are separated into different parts representing various portions of the artefact or map feature (e.g. portions of the street). A wireless device may issue a request for the map server to download only those DEntries that are included within a specified area or hounding box representing an area of interest that can be represented by, for example, a pair of bottom left, top right coordinates.

**[0055]** As depicted in FIG. **3**B, the wireless communications device issues one or more AOI (Area of Interest) requests, DEntry or data requests and Maplet Index requests to the map server for selective downloading of map data based on user context. Thus, rather than transmitting the entire map data for an area in reply to each request from the device (which burdens the wireless link), local caching may be used in conjunction with context filtering of map data on the server. For example, if a user's wireless device is GPS enabled and the user is traveling in an automobile at 120 km/h along a freeway then context filtering can by employed to prevent downloading of map data relating to passing side streets. Or, if the user is traveling in an airplane at 30,000 feet, then context filtering can be employed to prevent downloading of map data for any streets whatsoever. Also, a user's context can be defined, for example, in terms of occupation, e.g. a user whose occupation is a transport truck driver can employ context filtering to prevent downloading of map data for side streets on which the user's truck is incapable of traveling, or a user whose occupation is to replenish supplied of soft drink dispensing machines can employ context filtering to download public map data showing the user's geographical area of responsibility with irrelevant features such as lakes and parks filtered out and private map data containing the location of soft drink dispensing machines superimposed on the public map data.

**[0056]** The Maplet Index request results in a Maplet Index (i.e. only a portion of the Maplet that provides a table of contents of the map data available within the Maplet rather than the entire Maplet) being downloaded from the map server to the device, thereby conserving OTA (Over-the-Air) bandwidth and device memory caching requirements. The Maplet Index conforms to the same data structure as a Maplet, but omits the data points. Consequently, the Maplet Index is small (e.g. 300-400 bytes) relative to the size of a fully populated Maplet or a conventional bit map, and includes DEntry bounding boxes and attributes (size, complexity, etc.) for all artefacts within the Maplet. As the field of view changes (e.g. for a location-aware device that displays a map while moving), the device (client) software assesses whether or not it needs to download additional data from the server. Thus, if the size attribute or complexity attribute of an artefact that has started to move into the field of view of the device (but is not yet being displayed) is not relevant to the viewer's current context, then the device can choose not to display that portion of the artifact. On the other hand, if the portion of the artefact is appropriate for display, then the device accesses its cache to determine whether the DEntries associated with that portion of the artefact have already been downloaded, in which case the cached content is displayed. Otherwise, the device issues a request for the map server to download all the of the DEntries associated with the artifact portion.

**[0057]** By organizing the Maplet data structure in Layers, it is possible to seamlessly combine and display information obtained from public and private databases. For example, it is possible for the device to display an office building at a certain address on a street (e.g. a 1st z-order attribute from public database), adjacent a river (e.g. a 2nd z-order attribute from public database), with a superimposed floor plane of the building to show individual offices (e.g. 11th z-order attribute from a private database, accessible through a firewall).

**[0058]** Referring back to FIG. **3**A, within the network having map server(s) and/or LBS server(s) **321** and database(s) **322** accessible to it, all of the map data for the entire world is divided and stored as a grid according to various levels of resolution (zoom), as set forth below in Table A. Thus, a single A-level Maplet represents a 0.05×0.05 degree grid area; a single B-level Maplet represents a 0.5×0.5 degree grid area; a single C-level Maplet represents a 5×5 degree grid area; a single D-level Maplet represents a 50×50 degree grid area; and a single E level Maplet represents the entire world in a single Maplet. It is understood that Table A is only an example of a particular Maplet grid division; different grid divisions having finer or coarser granularity may, of courser, be substituted. A Maplet includes a set of layers, with each layer containing a set of DEntries, and each DEntry containing a set of data points.

TABLE A

| Level | Grid (degrees) | # of Maplets to cover the World | # of Maplets to cover North America | # of Maplets to cover Europe |
|---|---|---|---|---|
| A | 0.05 × 0.05 | 25,920,000 | 356,000 | 100,000 |
| B | 0.5 × 0.5 | 259,200 | 6,500 | 1000 |
| C | 5 × 5 | 2,592 | 96 | 10 |
| D | 50 × 50 | 32 | 5 | 5 |
| E | World | 1 | 1 | 1 |

**[0059]** As mentioned above, three specific types of requests may be generated by a wireless communications device (i.e. the client)—AOI requests, DEntry requests and Maplet Index requests. The requests may be generated separately or in various combinations, as discussed in greater detail below. An AOI (area of interest) request calls for all DEntries in a given area (bounding box) for a predetermined or selected set of z-order Layers. The AOI request is usually generated when the device moves to a new area so as to fetch DEntries for display before the device client knows what is available in the Maplet. The Maplet Index has the exact same structure as a Maplet but does not contain complete DEntries (i.e. the data Points actually representing artifacts and labels are omitted). Thus, a Maplet Index defines what Layers and DEntries are available for a given Maplet. A data or DEntry request is a mechanism to bundle together all of the required Dentries for a given Maplet.

**[0060]** Typically, AOI and Maplet Index requests are paired together in the same message, although they need not be, while DEntry requests are generated most often. For example, when a wireless device moves into an area for which no information has been stored on the device client, the Maplet Index request returns a Maplet Index that indicates what data the client can specifically request from the server **321**, while the AOI request returns any DEntries within the area of interest for the specified Layers (if they exist). In the example requests shown on FIG. **3**B, the desired Maplet is identified within a DEntry request by specifying the bottom-left Maplet coordinate. In addition, the DEntry request may include a layer mask so that unwanted Layers are not downloaded, a DEntry mask so that unwanted data Points are not downloaded, and zoom values to specify a zoom level for the requested DEntry. Once the device client has received the requested Maplet Index, the client typically then issues multiple DEntry requests to ask for specific DEntries (since the client knows all of the specific DEntries that are available based on the Maplet Index).

**[0061]** In this particular implementation, a collection of 20×20 A-level Maplets (representing a 1×1 degree square) is compiled into a Maplet Block File (.mbl). An .mbl file contains a header which specifies the offset and length of each Maplet in the .mbl file. The same 20×20 collection of Maplet index data is compiled into a Maplet Index file

(.mbx). The .mbl and .mbx file structures are set forth in Tables B and C, respectively.

TABLE B

| Address Offset | Offset | Length |
|---|---|---|
| 0x000 | Maplet #0 Offset (4 bytes) | Maplet #0 Length (4 bytes) |
| 0x008 | Maplet #1 Offset | Maplet #1 Length |
| 0x010 | Maplet #2 Offset | Maplet #2 Length |
| . . . | . . . | . . . |
| 0xC78 | Maplet #399 Offset | Maplet #399 Length |
| 0xC80 | Beginning of Maplet #0 | |
| 0xC80 + Size of Maplet #0 | Beginning of Maplet #1 | |
| 0xC80 + Size of Maplet #0 + #1 | Beginning of Maplet #2 | |
| . . . | . . . | |
| 0xC80 + Σ of Size of Maplets (#0:#398) | Beginning of Maplet #399 | |

**[0062]** In Table B, the offset of Maplet #0 is 0x0000_0000 since, in this particular example, the data structure is based on the assumption that the base address for the actual Maplet data is 0x0000_0C80. Therefore the absolute address for Maplet #0 data is: Maplet #0 Address=Base Address (0x0000_0C80)+Maplet #0 Offset (0x0000_0000), and additional Maplet addresses are calculated as: Maplet #(n+1) Offset=Maplet #(n) Offset+Maplet #(n) Length. If a Maplet has no data or does not exist, the length parameter is set to zero (0x0000_0000).

TABLE C

| Address Offset | Offset (4 bytes) | Length (4 bytes) |
|---|---|---|
| 0x000 | Maplet Index #0 Offset | Maplet Index #0 Length |
| 0x008 | Maplet Index #1 Offset | Maplet Index #1 Length |
| 0x010 | Maplet Index #2 Offset | Maplet Index #2 Length |
| . . . | . . . | . . . |
| 0xC78 | Maplet Index #399 Offset | Maplet Index #399 Length |
| 0xC80 | Beginning of Maplet Index #0 | |
| 0xC80 + Size of Maplet Index #0 | Beginning of Maplet Index #1 | |
| 0xC80 + Size of Maplet Index #0 + #1 | Beginning of Maplet Index #2 | |
| . . . | . . . | |
| 0xC80 + Σ of Size of Maplet Indices (#0:#399) | Beginning of Maplet Index #399 | |

**[0063]** In Table C, the offset of Maplet Index #0 is 0x0000_0000 since, according to an exemplary embodiment the data structure is based on the assumption that the base address for the actual Maplet index data is 0x0000_0C80. Therefore, the absolute address for Maplet Index #0 data is: Maplet Index #0 Address=Base Address (0x0000_0C80)+Maplet Index #0 Offset (0x0000_0000), and additional Maplet index addresses are calculated as: Maplet Index #(n+1) Offset=Maplet Index #(n) Offset+Maplet Index #(n) Length. If a Maplet Index has no data or does not exist, the length parameter is set to zero (0x0000_0000).

**[0064]** FIG. 3C and Table D (below), in combination, illustrate, by way of example only, a basic Maplet data structure. Generally, as noted above, the Maplet data structure can be said to include a Maplet Index (i.e. an index of the DEntries, each of which is representative of either an artifact or a label or both) together with data Points for each DEntry that actually form such artifacts and labels. In this example, each Maplet includes a Map ID (e.g. 0xA1B1C1D1), the # of Layers in the Maplet, and a Layer Entry for each Layer. The Map ID identifies the data as a valid Maplet, and according to one alternative, may also be used to identify a version number for the data. The # of Layers is an integer which indicates the number of Layers (and therefore Layer Entries) in the Maplet. Each Layer Entry defines rendering attributes and is followed by a list of DEntries for each Layer. The above forms a Maplet Index. For a complete Maplet, each DEntry contains a set of data Points (referred to herein as oPoints) or Labels). It will be noted that Layers can have multiple DEntries and the complete list of DEntries and Points are grouped by Layer and separated by a Layer Separator (e.g. hex value 0xEEEEEEEE). In this example, each Layer Entry is 20 bytes long, and a DEntry is 12 bytes long. However, the number of Layers, number of DEntries per Layer and the number of Points per DEntry depends on the map data and is generally variable.

**[0065]** Table D provides a high "byte-level" description of a Maplet for this example.

TABLE D

| Data | Quantity | Total # of Bytes |
|---|---|---|
| Map ID | 1 | 4 bytes |
| # of Layers | 1 | 4 bytes |
| Layer Entries | # of Layers | 20 bytes × (# of Layers) |
| DEntry of a Layer x (# of DEntries in a Layer) | # of Layers | 12 bytes × (Σ of the # of DEntries in each Layer) + |
| Points for DEntry of a Layer | | 4 bytes × (Σ of the # of Points in each DEntry in each Layer) + |
| Layer Separator | | 4 bytes × (# of Layers) |

**[0066]** By way of a further example, the wireless network **200** depicted in FIG. **4** can include an applications gateway (AG) **350** for optimizing data flow for onboard applications such as a mapping application **500** stored in memory (e.g. stored in a flash memory **224**) and executable by the microprocessor **238** of the wireless device **202**.

**[0067]** As shown in FIG. **4**, the wireless network **200** hosts a plurality of handheld wireless communications devices **202** (such as the BlackBerry™ by Research in Motion Limited) having voice and data capabilities (for both e-mail and web browsing) as well as a full QWERTY keyboard. These wireless communications devices **202** can access Web-based map data on public map servers **400** hosted on the Internet or other data network **130** via the applications gateway (AG) **350** which mediates and optimizes data flow between the wireless network **200** and the data network by performing various mappings, compressions and optimizations on the data.

**[0068]** The map server extracts generic map content from a Geographical Information Systems (GIS) map database (e.g. Navtech®, TelAtlas®, etc.) at a specified level of resolution (zoom level). Custom graphics associated with the query, such as highlighted route, pushpin for current position or street address, etc. are post-processed and merged by the server with the generic map content. Relevant screen graphics are then labelled, and the merged map graphic is compressed and delivered to the device for display.

[0069] In operation, a user of the wireless communications device **202** uses an input device such as keyboard **232** and/or thumbwheel **233** to cause the microprocessor **238** to open the map application **500** stored in the memory **224**. Using the keyboard **232** and thumbwheel **233**, the user specifies a map location on the map application **500**. In response to this request/command, the microprocessor **238** instructs the RF transceiver circuitry **211** to transmit the request over the air through the wireless network **104**. The request is processed by the AG **350** and forwarded into the data network (Internet) using standard packet-forwarding protocols to one or more of the public and/or private map servers **400**, **410**. Accessing a private map server **410** behind a corporate firewall **420** was described above with reference to FIG. **3**A. Map data downloaded from these one or more map servers **400**, **410** is then forwarded in data packets through the data network and mapped/optimized by the AS **350** for wireless transmission through the wireless network **104** to the wireless communications device **202** that originally sent the request.

[0070] The downloaded map data can be cached locally in RAM **226**, and displayed on the display **222** or graphical user interface (GUI) of the device after the map application **500** reconstructs or "stitches together" portions of features or constituent path segments to generate a reconstructed map feature or path, as will elaborated below, so that a single instance of the label can be centrally rendered for the reconstructed feature or path (provided it does not collide with another label of higher priority). If a further request is made by the user (or if the user wants a change in the field of view by zooming or panning), the device will check whether the data required can be obtained from the local cache (RAM **226**). If not, the device issues a new request to the one or more map servers **400**, **410** in the same manner as described above.

[0071] As described earlier, map data can optionally be downloaded first as a Maplet Index enabling the user to then choose which DEntries listed in the Index to download in full. Furthermore, as described earlier, the map application can include user-configurable context filtering that enables the user to filter out unwanted map features or artifacts by not downloading specific DEntries corresponding to those unwanted map features or artifacts.

[0072] As a variant, the wireless communications device can optionally include a Global Positioning System (GPS) receiver ("GPS chip") **550** for providing location-based services (LBS) to the user in addition to map content. Embedding a GPS chip **550** capable of receiving and processing signals from UPS satellites enable the GPS chip to generate latitude and longitude coordinates, thus making the device "location aware". To obtain local-based services, the map application within the wireless communications device sends a request to the map server for information relating to a city, restaurant, street address, route, etc. If the device is "location aware", the request would include the current location of the device.

[0073] In lieu of, or in addition to, GPS coordinates, the location of the device can be determined using triangulation of signals from in-range base towers, such as used for Wireless E911. Wireless Enhanced 911 services enable a cell phone or other wireless device to be located geographically using radiolocation techniques such as (i) angle of arrival (AOA) which entails locating the caller at the point where signals from two towers intersect; (ii) time difference of arrival (TDOA), which uses multilateration like GPS, except that the networks determine the time difference and therefore the distance from each tower; and (iii) location signature, which uses "fingerprinting" to store and recall patterns (such as multipath) which mobile phone signals exhibit at different locations in each cell.

[0074] Operation of the systems described above will now be described with reference to the method steps depicted in the flowchart of FIG. **5**. As depicted in FIG. **5**, this method of displaying a map on a wireless communications device includes initial steps of opening the map application on the device (step **600**) and specifying an area of interest (AOI) using the map application (step **602**), e.g. specifying a street address, coordinates of latitude or longitude, or clicking on a location on a world map, etc. In response to the specifying of an AOI, map data is then obtained (step **604**) for rendering the map to be displayed on the wireless communications device. For the purposes of this specification, "obtaining map data" means receiving or downloading the map data over the air, i.e. over a wireless link, retrieving the map data from a local cache, or downloading the map data over a wired connection, or any combination thereof. In other words, as depicted in FIG. **5**, obtaining map data includes steps of determining whether the data is already cached locally (step **604**). If the data is locally cached, the map data is retrieved from the cache (step **606**). Otherwise, if not all of the map data is cached, then the map data is downloaded over the air (step **608**).

[0075] As depicted in FIG. **5**, once the map data is obtained, the device generates a list of all labels to be rendered (step **610**). Once this label list is created (and preferably sorted into alphabetical order for more efficient processing), the device checks to see whether any of the labels are duplicated or redundant (step **612**). If no duplicated (redundant) labels are found in the label list, the map is then rendered (step **614**) without performing any stitching or reconstruction of paths or features. Rendering of the map can include a step (not shown) of verifying that the labels do not interfere with other labels. For example, this step of verifying that the labels do not interfere with other labels may involve a step of generating a collision-avoidance array representative of the map to be rendered for provisionally testing potential map positions prior to actually rendering the map. This "virtual rendering" enables the map application to ascertain that labels do not collide or overlap with other labels for which a map position has been previously assigned.

[0076] As further depicted in the flowchart of FIG. **5**, if duplicated or redundant labels are found in the label list at step **612**, then, for each duplicated label, then the device determines (step **616**) whether the paths (or other map features) match or connect, e.g. by looking at endpoints of path segments, as will be elaborated below with reference to FIG. **7**B. If the paths or features do not connect or otherwise fit together, then the map is rendered without stitching the segments together, i.e. labels are rendered for each of the segments or constituent elements because these segments or constituent are disjointed or disconnected and therefore require separate labelling. On the other hand, if, for any duplicated labels, the path segments or constituent elements of the map features connect or match, then the path segments or constituent elements of the map features are reconstructed ("stitched together") to form a reconstructed path or reconstructed map feature (step **618**). Once the reconstructed path or reconstructed map feature is generated, the map can be rendered with a single instance of the map label, preferably centrally positioned along the entire length of the reconstructed path or centrally disposed vis-à-vis the reconstructed map feature (step **620**). Rendering the map with the single instance of the map label should involve checking whether the label interferes or overlaps with any other label for which a label position has been designated. As noted above, this can be accomplished by generating a collision-avoidance array representing the map to be rendered and then populating the array with label positions from highest priority to lowest, checking that each successive label of decreasing importance does not collide in the array with any previously assigned label positions.

**[0077]** For the purposes of this specification, "label" includes not only all conventional forms of labels, such as city names, street names, etc, but also any symbols or icons, such as highway number icons, or symbols or icons used to denote airports, tourist information kiosks, campgrounds, ferry crossings, etc. on large scale (regional) maps or restaurants, hotels, bus stations, etc. on city maps.

**[0078]** For the purposes of this specification, "map feature" means a path, road, street, highway or other route and also includes features such as a body of water (river, lake, bay, strait, sea, ocean), an island, a park or other geographical feature that can be rendered from two or more separate sets of map data (i.e. vectors) for which individual labels are provided (and which are thus potentially duplicated upon rendering).

**[0079]** FIG. **6** schematically depicts the process of reconstructing ("stitching") paths and/or map features in order to efficiently generate aesthetically-labelled maps for being displayed on wireless communications devices. By way of overview, map data (which includes label data) is obtained from a map server in the form of Data Entries ("D Entries"). Different layers of these D Entries are used to render features of the same type or class. Thus, for example, one layer of D Entries may be for lakes, rivers and bodies of water, while another layer of D Entries may be for highways, roads and streets. This layered implementation enables context-filtering of desired or pertinent map data so that only desired or pertinent features are rendered onscreen.

**[0080]** In the example depicted in FIG. **6**, the map is rendered from three separate D Entries (or three separate groups of P Entries from different layers). For the sake of illustration, the three D Entries (D Entry #**1**, D Entry #**2**, and D Entry #**3**) are rendered together to constitute the (composite) map. As each D Entry has its own (independent) label for "Main Street" as well as its own label for "Windy Lake", simply rendering the map data as a composite map would unacceptably result in duplication of the labels, as shown in FIG. **6**.

**[0081]** Even if any duplicated labels are suppressed, the resulting map, as depicted in FIG. **7**, would not be aesthetically pleasing because only one of the three path labels would appear along its respective path segment (e.g. the Main Street label would appear, say, only along the first path segment), which is not necessarily centered. Similarly, only a single instance of the map feature label (e.g. Windy Lake) would appear on only one of the elements of the feature (e.g. the Windy Lake label would appear, say, on only the first constituent portion of the lake). A corollary problem is that the label can only be displaced over a limited range corresponding to the segment or constituent element (if repositioning is mandated by a collision with another label). Since the label can only be repositioned over a limited range, the resulting labelled map is aesthetically compromised.

**[0082]** These problems can be overcome by stitching or reconstructing paths (or other map features) to create reconstructed paths (or reconstructed features), as depicted in FIG. **8**. Since the path segments have duplicated labels and connecting endpoints, the path segments are stitched/reconstructed to form a single reconstructed path.

**[0083]** In one implementation, this reconstruction (stitching) can be accomplished in the manner described in FIGS. **9**A-**9**C. Similarly, other map features (such as the lake in FIG. **8**) can be reconstructed, or stitched together, to form a single reconstructed feature.

**[0084]** For the reconstructed path, only a single instance of the label (e.g. "Main Street") is rendered, preferably in a central position vis-à-vis the path (i.e. the most aesthetic place for the label). Since the path has been stitched together to form a single reconstructed path, the label can be displaced anywhere along the reconstructed path. Therefore, as shown by the dashed-line arrows in FIG. **8**, the label can be displaced over a much greater range than was previously possible when the label was confined to being rendered somewhere along the limited range of its original path segment. In other words, not only can the label be centered vis-à-vis the "true" (from the viewer's perspective) center of the reconstructed path or feature, but the label can also be displaced substantially to avoid collisions with other labels.

**[0085]** Likewise, for the reconstructed map feature (in this example, the lake), only a single instance of the label (e.g. "Windy Lake") is rendered, preferably in, a central, prominent location vis-à-vis the feature, provided it does not collide or interfere with another pre-existing or higher-priority label. Furthermore, because of the reconstruction of the feature, the feature is no longer composed of constituent parts for the purposes of labelling. Accordingly, the label can be displaced over the entire range of the feature, not just over the constituent part with which the label was originally associated. This provides much more leeway in finding a suitable position for a label on the map, i.e. a label position that does not collide or interfere with any other label. In order words, this stitching technique enables labels to be rendered in preferred positions (e.g. centrally, prominently, aesthetically, etc.) while providing maximal leeway for displacing the label in the event that it collides with another (pre-existing or higher-priority) label.

**[0086]** FIG. **9**A schematically depicts the generation of a label list **700**, in accordance with one implementation of present technology, for determining whether any labels are duplicated in a given set of D Entries that are to be used to render the map. The label list **700**, in this implementation, is generated by the map application **500** using label data received wirelessly by the wireless communications device **202**.

[0087] In the example shown in FIG. 9A, the label list 700 includes the list of label names itself (i.e. a field for storing the name of each label instance), a link field (indicating how, if at all, the label can be linked to any duplicate labels), a vector field (indicating directionality of map data stored in vector format) and a flag field (indicating whether the data vector needs to be reversed to concord with the directionality of a vector of the same label). Although each of these four fields of the label list is described in greater detail below, it should be understood that the details of this label list are presented solely for the purposes of illustration. Persons of ordinary skill in the art will appreciate that other implementations of label lists or equivalent algorithms can be used to determine label redundancy and whether endpoints of the paths or other non-path features of any redundant labels match.

[0088] In the example presented in FIG. 9A, the label list 700 includes path labels "First Avenue", "Main Street", and "Second Avenue" as well as any non-path feature labels, i.e. "Windy Lake". As shown, multiple instances of each label appear in the label list 700, representing each instance that one of the D Entries used to render the map carries that particular label. Thus, in this example, the label "Main Street" is listed three times in the label list because each of the D Entries used to create the map contains its own instance of the label "Main Street". Likewise, since each of the three D Entries contains the non-path feature label "Windy Lake", this label is listed three times in the label list. Preferably, the label list is sorted alphabetically to streamline the algorithm that searches for redundancies and performs the linking.

[0089] In the example depicted in FIG. 9A, the label list 700 includes a link field or link parameter that indicates for each label entry (each listed instance of each label) what its relationship is with a previous or subsequent label of the same name. Linking of labels can be accomplished using a standard linked-list construct for objects, which is well known in the art. If a label appears only once in the label, i.e. has merely a single instance, then it cannot be linked to another label, and therefore its link parameter, or link field, is simply indicated as "None". Thus, returning to the specific example presented in FIG. 9A, the path label "First Avenue" appears only once, and therefore its link parameter is "None". The same holds for Second Avenue, which appears only once. Its link parameter is thus also designated as "None."

[0090] Unlike First Avenue and Second, the path label "Main Street" appears three times, and thus its link parameters need to determined. Determining link parameters (or link status) can be accomplished by comparing endpoints of each link segment, as depicted in FIG. 9B. As shown in FIG. 95, the endpoints (x1,y1) of the first path segment of Main Street are compared with the endpoints (x2,y2) of the second segment of Main Street. If the endpoints (x,y coordinates) are equal (or at least match within a predetermined tolerance), then the segments are eligible to be stitched together. Accordingly, the link parameter/status is updated to reflect the concordance of the endpoints of the path segments. In this example, the first instance of the Main Street label shows that the link parameter is "Next" (meaning that the segment with this label connects to the segment associated with the next label in the list).

[0091] As depicted in FIG. 9B, the endpoints (x3,y3) of the second segment of path label "Main Street" are compared with the endpoints (x4,y4) of the third segment of "Main Street" to determine whether these endpoint coordinates coincide. If the endpoints coincide (or match within the tolerance), as they do in this example, the link parameter is updated to indicate that the third label is linked to the "previous" label, i.e. the second label. Using this linked-list construct, the relationships between the first Main Street label and the second Main Street label and between the second Main Street label and the third Main Street label are defined. In this example, the first Main Street label is linked to the next label, i.e. the second Main Street label (and, conversely, the second Main Street label is linked to the previous Main Street label). The third Main Street label is linked to the previous Main Street label as well (i.e. to the second Main Street label) As a result, all three labels are linked together, meaning that the three path segments can be stitched together.

[0092] Likewise, the label list also accounts for linkage relationships between non-path map features such as the lake shown in FIGS. 6-8. Its label "Windy Lake" appears in each D Entry and thus three instances of this label appear in the label list. Again, by comparing endpoints or perimeter points, the constituent parts of the lake can be compared to see whether they match or align. If so, the link fields for each Windy Lake entry can be updated as was done for Main Street.

[0093] As further depicted in FIG. 9B, the label list 700 can include a vector field and a flag field. The map data is stored in vector format, allowing the vectors (data) to be packaged in small "chunks" to facilitate OTA transmission and rendering. If the bounding box of each "chunk" of data, or D Entry, is small then it is quicker to intersect its bounding box with the screen's bounding box to determine if it needs to be rendered (and requested/transmitted if it not already cached client-side). Chunking up the data into small packages or D Entries, however, means that paths and other map features (each having their own labels) will likely extend into more than one D Entry. The foregoing technique effectively reconstructs the paths (or features) by checking whether the endpoints of path segments (or whether the constituent parts of features) coincide or fit together. A further problem that arising with the D Entries being in vector format is that the directionality of one segment of a path (or feature) could be opposite to that of another segment even if their respective endpoints coincide. Prior to stitching these segments (or constituent parts) together, then, it is preferable to assess the directionality of the vectors defining the segments (or constituent parts). This can be accomplished using a unit vector notation such as a presented, by way of example only, in FIG. 9A. In this example, the First Avenue label is rendered using a unit vector that runs vertically download (in the y-direction) without any horizontal component (x=0). Thus, the vector is denoted as (0,−1). The magnitude of the vector is not relevant, only direction. Comparison of vector directionality is depicted, again by way of example only, in FIG. 9C which shows the three segments of the path Main Street and their associated vectors. Assuming that the first segment of Main Street has unit vector (1, −1), that the second segment has unit vector (1,0) and that the third segment has unit vector (−1,0), then it is observed that the second and third segments have opposite directions, as shown in FIG. 9C. Accordingly, this inconsistency in the directionality of two contiguous segments that are eligible to be conjoined or stitched together for the purposes of feature reconstruction is flagged in label list **700**. The flag indicates that the third segment of Main Street needs to have the direction of its vector reversed. Once all vectors have been aligned by reversing any inconsistent segments of the path, the three segments can be "stitched" or "spliced" together to generate the contiguous, reconstructed path.

[**0094**] Optionally, when the reconstructed path is generated, the length of each constituent path segment and/or the total length of the reconstructed path are stored. These values can be used to determine an initial starting point for centering each label vis-à-vis the midpoint of the reconstructed path. Knowledge of these values also facilitates repositioning of the label when a potential label collision is detected or foreseen. These values can be stored as further fields of the label list **700**. Labels can thus be rendered, or virtually rendered, with reference to the center of the reconstructed path, which is the preferred technique. Alternatively, labels can be rendered (or repositioned in the virtual rendering process) by virtually rendering a label along a center of a middle segment (or the segment closest to the middle of the onscreen bounding box) and then, if all of the label does not fit along that segment, checking whether the segment is spliced to a further segment (i.e. checking whether a reconstructed path exists for that label).

[**0095**] Similarly, when reconstructing non-path features (i.e. features that are not lines but rather polygons), other dimensions such as, for example, the average horizontal width of the polygon feature, can be stored for each of the constituent elements of the non-path map feature and for the reconstructed map feature, also for the purposes of facilitating centered labelling of the reconstructed feature.

[**0096**] The label list **700** depicted in FIG. **9**A can be implemented using a labelPath object created for the path, containing the path data, the label and any other information used to render the label on the map. Every labelPath object would then be placed into a list sorted alphabetically according to the label associated with each path. When each labelPath object is added to the list, a check is performed to see if there are any other labelPath objects whose labels are identical to the label of the path being added. If other labelPath objects are found with identical labels, the endpoints are compared and, if any match, then the labelPath objects are associated with each other using a standard linked-list construct, as alluded to above.

[**0097**] FIG. **10** is a screenshot of a street map of downtown Ottawa, Canada without stitching (reconstruction) of paths. As shown, many of the street labels are not centered onscreen. Some egregious examples are "Queen St", "Albert St" and "Nepean St" which are far from being properly centered onscreen. In contrast, FIG. **11** is a screenshot of a street map of downtown Ottawa, Canada with stitching (reconstruction) of paths. The labels on this map are centered vis-à-vis their respective streets. Note, in particular, how the path labels "Queen St", "Albert St" and "Nepean St" are centrally located by first stitching the path segments constituting the entire path. Some repositioning (off-centering) may, of course, prove to be inevitable as a consequence of having to avoid collisions with other labels. A collision-avoidance algorithm will attempt to place the label centrally (at a midpoint of the path), if possible, but will then reposition the label, if necessary, to avoid collisions with other labels onscreen.

[**0098**] The foregoing method steps can be implemented as coded instructions in a computer program product. In other words, the computer program product is a computer-readable medium upon which software code is recorded to perform the foregoing steps when the computer program product is loaded into memory and executed on the microprocessor of the wireless communications device.

[**0099**] This new technology has been described in terms of specific implementations and configurations which are intended to be exemplary only. The scope of the exclusive right sought by the Applicant is therefore intended to be limited solely by the appended claims.

**1**. A method of displaying a map on a wireless communications device, the method comprising steps of:

obtaining sets of map data for rendering portions of the map on a display of the device, the map data including label data for rendering labels on the map for identifying one or more map features;

generating a list of all the labels to be rendered on the map;

for each duplicated label in the list, determining whether the map features associated with the duplicated labels connect on the map;

generating a reconstructed map feature; and

rendering a single instance of the label for the map feature.

**2**. The method as claimed in claim 1 wherein the step of rendering the single instance of the label comprises a step of rendering the label such that the label is aligned with a middle of the reconstructed map feature.

**3**. The method as claimed in claim 1 wherein the reconstructed map feature is a reconstructed path that is reconstructed from constituent path segments derived from separate sets of map data.

**4**. The method as claimed in claim 3 wherein the step of determining whether the map features associated with the duplicated labels connect on the map comprises a step of determining whether a first endpoint of a first path associated with the duplicated label matches a second endpoint of a second path also associated with the duplicated label.

**5**. The method as claimed in claim 4 wherein the step of rendering the single instance of the label comprises a step of rendering the label midway along the reconstructed path.

**6**. The method as claimed in claim 1 wherein the step of generating the reconstructed map feature comprises steps of:

determining a direction of a data vector for each constituent set of map data that, when rendered, together constitutes the reconstructed map feature; and

reversing the direction of one or more data vectors so that all data vectors in the reconstructed map feature have a common direction.

**7**. The method as claimed in claim 3 wherein the step of generating the reconstructed map feature comprises steps of:

determining a direction of a data vector for each constituent set of map data that, when rendered, together constitutes the reconstructed path; and reversing the direction of one or more data vectors so that all data vectors corresponding to each constituent path segment in the reconstructed path have a common direction.

**8**. The method as claimed in claim 3 wherein the step of generating the reconstructed path comprises a step of storing a length of each constituent path segment and a total length of the reconstructed path.

**9**. The method as claimed in claim 7 wherein the step of generating the reconstructed path comprises a step of storing a length of each constituent path segment and a total length of the reconstructed path.

**10**. The method as claimed in claim 1 wherein the step of generating the list comprises a step of sorting the labels into alphabetical order.

**11**. A computer program product comprising code adapted to perform the steps of claim 1 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**12**. The computer program product comprising code adapted to perform the steps of claim 2 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**13**. The computer program product comprising code adapted to perform the steps of claim 3 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**14**. The computer program product comprising code adapted to perform the steps of claim 4 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**15**. The computer program product comprising code adapted to perform the steps of claim 5 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**16**. The computer program product comprising code adapted to perform the steps of claim 6 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**17**. The computer program product comprising code adapted to perform the steps of claim 7 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**18**. The computer program product comprising code adapted to perform the steps of claim 8 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**19**. The computer program product comprising code adapted to perform the steps of claim 9 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**20**. The computer program product comprising code adapted to perform the steps of claim 10 when the computer program product is loaded into memory and executed on a processor of a wireless communications device.

**21**. A wireless communications device for enabling a user of the device to display a map on the device, the wireless device comprising:

an input device for enabling the user to cause the device to obtain map data for rendering the map to be displayed on a display of the device, the map data including label data for rendering labels on the map for identifying one or more map features; and

a memory for storing code to instruct a processor to:

generate a list of all the labels to be rendered on the map;

determine, for each duplicated label in the list, whether the map features associated with the duplicated labels connect on the map;

generate a reconstructed map feature; and

render on the display of the device a single instance of the label for the map feature.

**22**. The wireless communications device as claimed in claim 21 wherein the processor renders the label such that the label is aligned with a middle of the reconstructed map feature.

**23**. The wireless communications device as claimed in claim 21 wherein the reconstructed map feature is a reconstructed path that is reconstructed from constituent path segments derived from separate sets of map data.

**24**. The wireless communications device as claimed in claim 23 wherein the processor determines whether a first endpoint of a first path associated with the duplicated label matches a second endpoint of a second path also associated with the duplicated label.

**25**. The wireless communications device as claimed in claim 24 wherein processor renders the single instance of the label midway along the reconstructed path.

**26**. The wireless communications device as claimed in claim 21 wherein the processor determines a direction of a data vector for each constituent set of map data that, when rendered, together constitutes the reconstructed map feature and then reverses the direction of one or more data vectors so that all data vectors in the reconstructed map feature have a common direction.

**27**. The wireless communications device as claimed in claim 23 wherein the processor determines a direction of a data vector for each constituent set of map data that, when rendered, together constitutes the reconstructed path and then reverses the direction of one or more data vectors so that all data vectors corresponding to each constituent path segment in the reconstructed path have a common direction.

**28**. The wireless communications device as claimed in claim 23 wherein the processor causes the memory to store a length of each constituent path segment and a total length of the reconstructed path.

**29**. The wireless communications device as claimed in claim 27 wherein the processor causes the memory to store a length of each constituent path segment and a total length of the reconstructed path.

* * * * *