US 20140047385A1

(19) United States
(12) Patent Application Publication
Ruble et al.
(10) Pub. No.: US 2014/0047385 A1
(43) Pub. Date: Feb. 13, 2014

(54) **3D DATA ENVIRONMENT DISAMBIGUATION TOOL**

(71) Applicant: **MICROSOFT CORPORATION**, Redmond, WA (US)

(72) Inventors: **B. Scott Ruble**, Bellevue, WA (US); **Kevin Fan**, Seattle, WA (US); **Jai Srinivasan**, Bellevue, WA (US); **Michael Kallay**, Bellevue, WA (US); **Alexandre da Veiga**, Bellevue, WA (US); **Ehab Sobhy**, Redmond, WA (US)

(73) Assignee: **MICROSOFT CORPORATION**, Redmond, WA (US)

(21) Appl. No.: **13/760,032**

(22) Filed: **Feb. 5, 2013**

Related U.S. Application Data

(60) Provisional application No. 61/681,851, filed on Aug. 10, 2012.

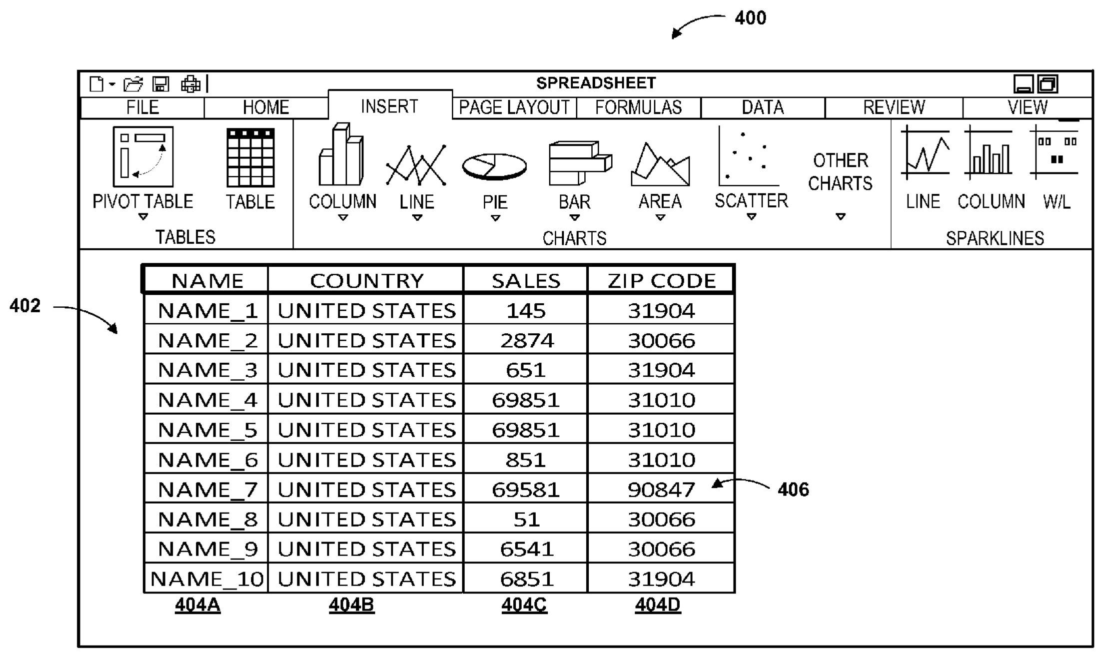
Publication Classification

(51) **Int. Cl.**
G06F 3/01 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 3/01* (2013.01)
USPC **715/810**

(57) **ABSTRACT**

Concepts and technologies are described herein for providing a 3D data environment disambiguation tool. In accordance with some concepts and technologies disclosed herein, a 3D data environment disambiguation tool can be configured to resolve an ambiguity associated with one or more data points to be rendered in a 3D data environment. The 3D data environment disambiguation tool can present various user interfaces to receive inputs to resolve an ambiguity associated with a data point. The ambiguity can be resolved prior to, in conjunction with, or after the selected data is rendered in a 3D data environment, or various combinations thereof.



NAME	COUNTRY	SALES	ZIP CODE
NAME_1	UNITED STATES	145	31904
NAME_2	UNITED STATES	2874	30066
NAME_3	UNITED STATES	651	31904
NAME_4	UNITED STATES	69851	31010
NAME_5	UNITED STATES	69851	31010
NAME_6	UNITED STATES	851	31010
NAME_7	UNITED STATES	69581	90847
NAME_8	UNITED STATES	51	30066
NAME_9	UNITED STATES	6541	30066
NAME_10	UNITED STATES	6851	31904

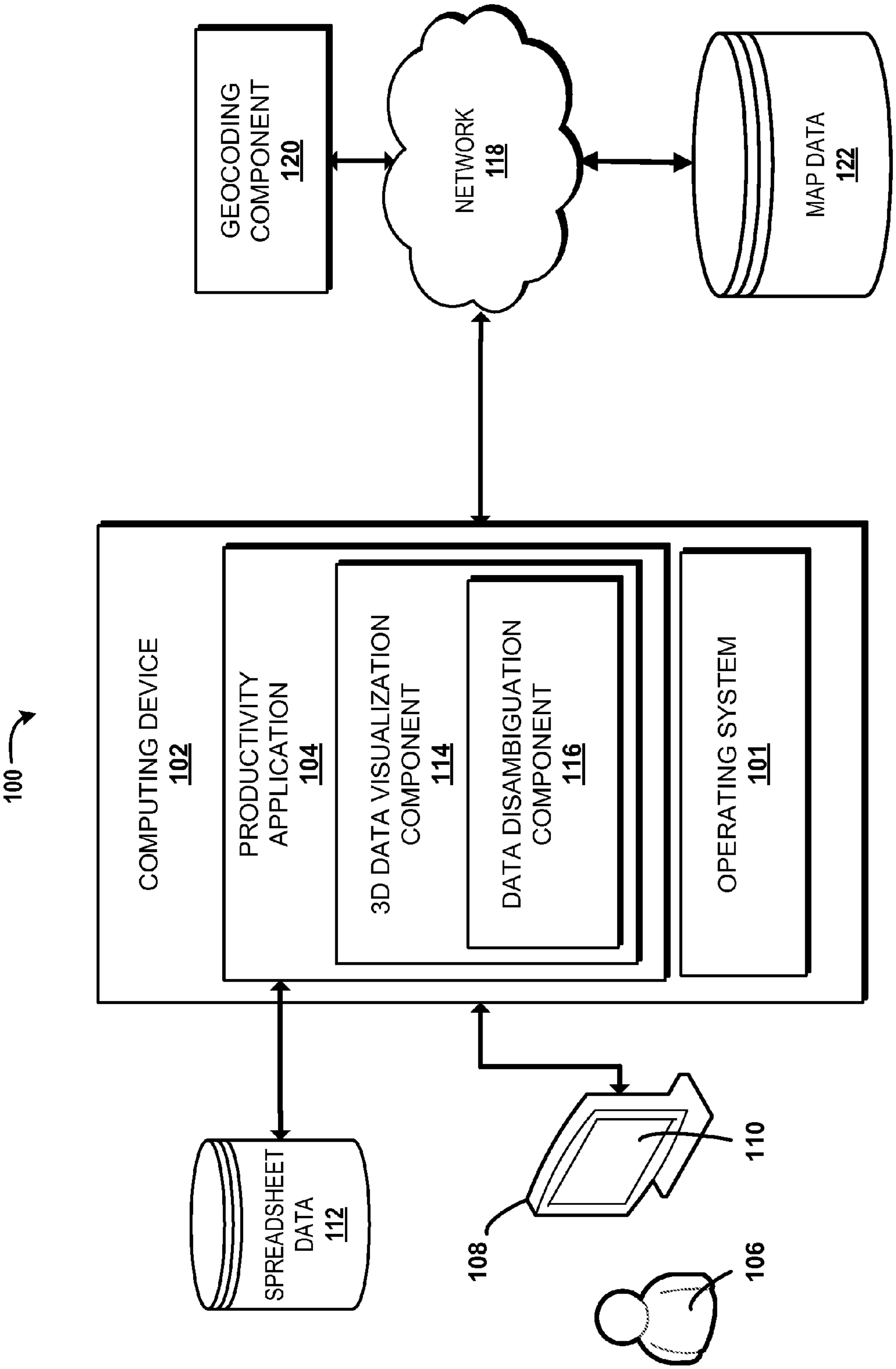

100
COMPUTING DEVICE 102
PRODUCTIVITY APPLICATION 104
3D DATA VISUALIZATION COMPONENT 114
DATA DISAMBIGUATION COMPONENT 116
OPERATING SYSTEM 101
GEOCODING COMPONENT 120
NETWORK 118
MAP DATA 122
SPREADSHEET DATA 112
108
110
106

FIG. 1

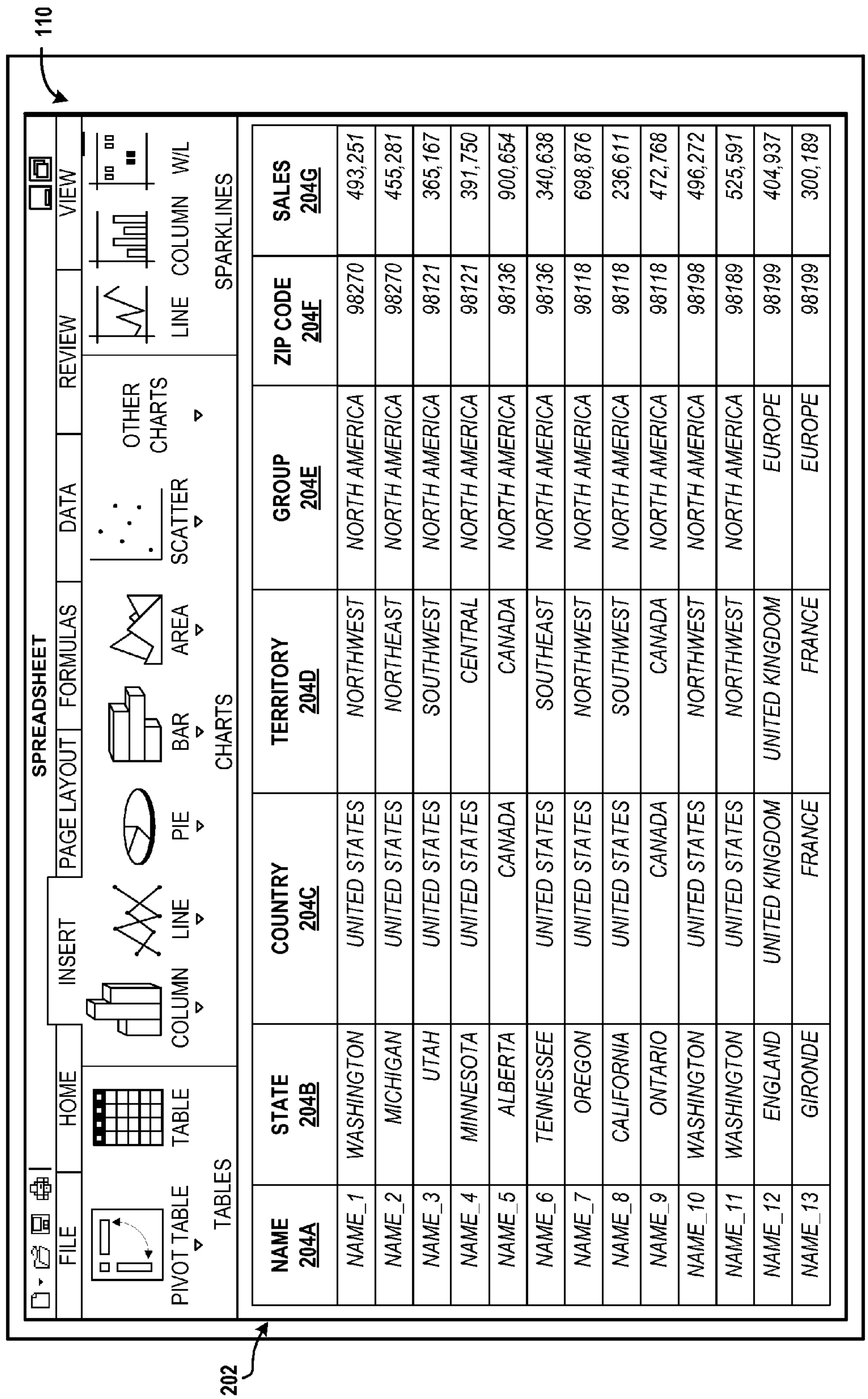

110
202
SPREADSHEET
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW
PIVOT TABLE TABLE
TABLES
COLUMN LINE PIE BAR AREA SCATTER OTHER CHARTS
CHARTS
LINE COLUMN W/L
SPARKLINES
NAME 204A	STATE 204B	COUNTRY 204C	TERRITORY 204D	GROUP 204E	ZIP CODE 204F	SALES 204G
NAME_1	WASHINGTON	UNITED STATES	NORTHWEST	NORTH AMERICA	98270	493,251
NAME_2	MICHIGAN	UNITED STATES	NORTHEAST	NORTH AMERICA	98270	455,281
NAME_3	UTAH	UNITED STATES	SOUTHWEST	NORTH AMERICA	98121	365,167
NAME_4	MINNESOTA	UNITED STATES	CENTRAL	NORTH AMERICA	98121	391,750
NAME_5	ALBERTA	CANADA	CANADA	NORTH AMERICA	98136	900,654
NAME_6	TENNESSEE	UNITED STATES	SOUTHEAST	NORTH AMERICA	98136	340,638
NAME_7	OREGON	UNITED STATES	NORTHWEST	NORTH AMERICA	98118	698,876
NAME_8	CALIFORNIA	UNITED STATES	SOUTHWEST	NORTH AMERICA	98118	236,611
NAME_9	ONTARIO	CANADA	CANADA	NORTH AMERICA	98118	472,768
NAME_10	WASHINGTON	UNITED STATES	NORTHWEST	NORTH AMERICA	98198	496,272
NAME_11	WASHINGTON	UNITED STATES	NORTHWEST	NORTH AMERICA	98189	525,591
NAME_12	ENGLAND	UNITED KINGDOM	UNITED KINGDOM	EUROPE	98199	404,937
NAME_13	GIRONDE	FRANCE	FRANCE	EUROPE	98199	300,189

FIG. 2

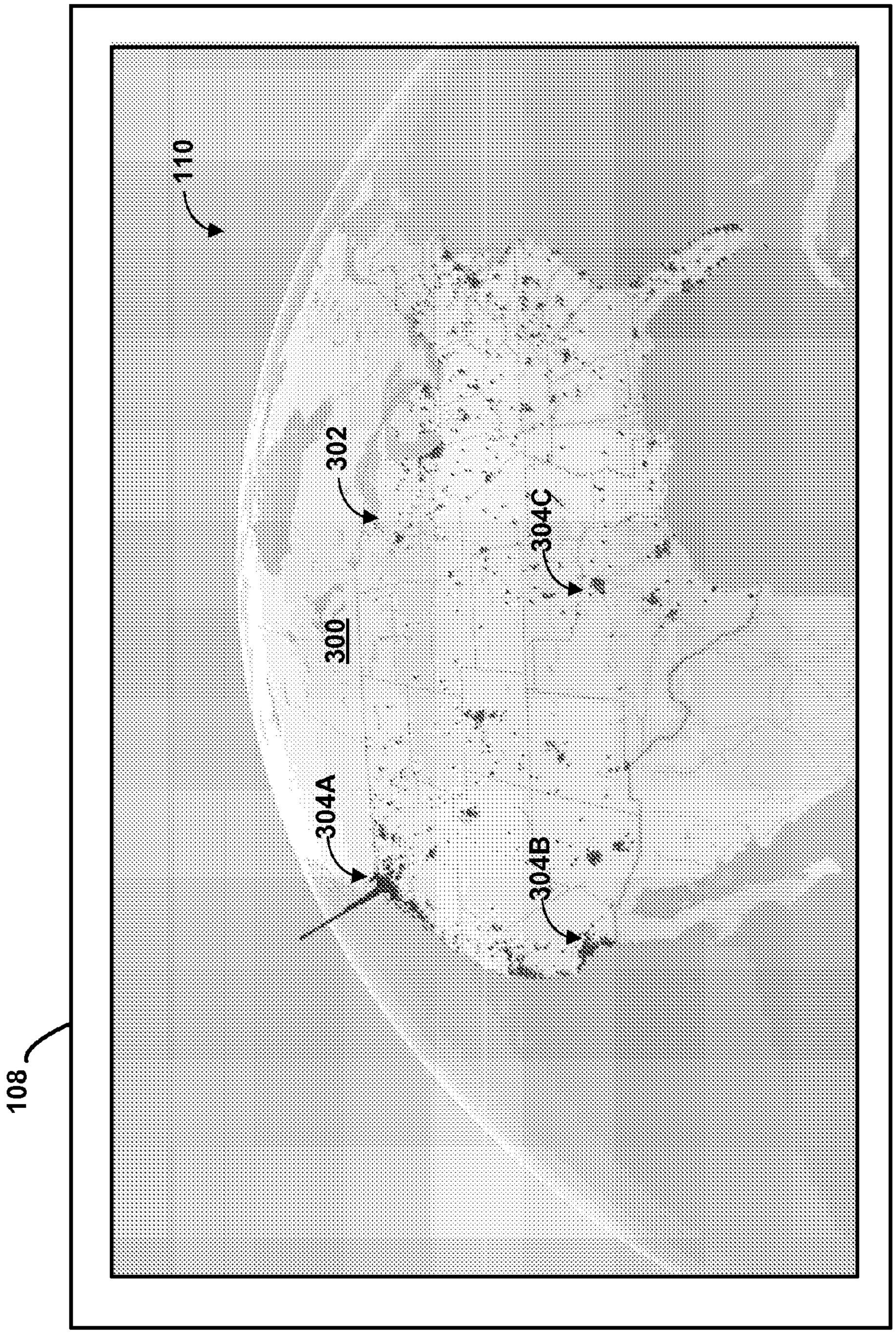


FIG. 3

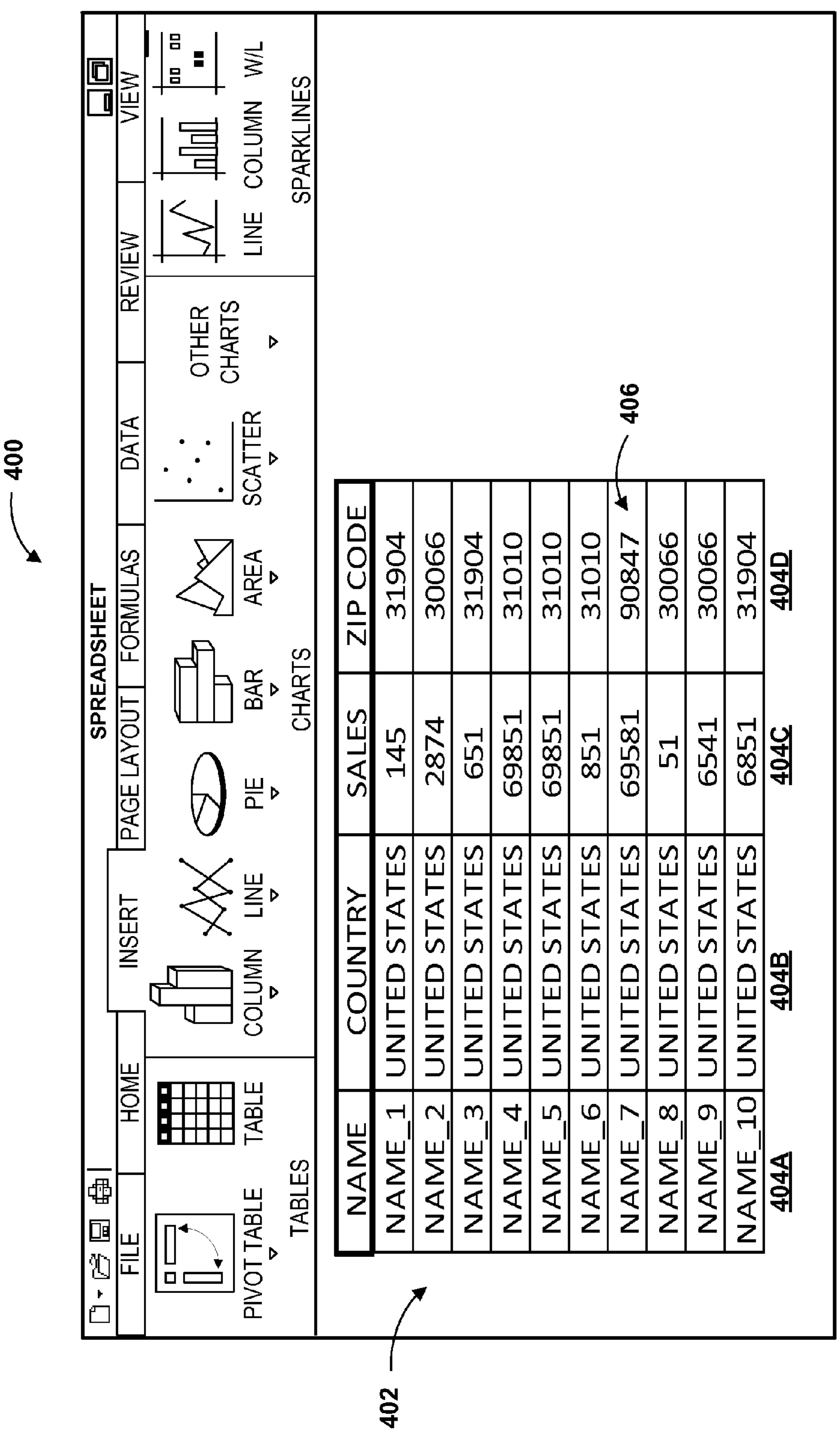

400
SPREADSHEET
FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
PIVOT TABLE
TABLE
TABLES
COLUMN
LINE
PIE
BAR
AREA
SCATTER
OTHER CHARTS
CHARTS
LINE
COLUMN
W/L
SPARKLINES
NAME
COUNTRY
SALES
ZIP CODE
NAME_1 UNITED STATES 145 31904
NAME_2 UNITED STATES 2874 30066
NAME_3 UNITED STATES 651 31904
NAME_4 UNITED STATES 69851 31010
NAME_5 UNITED STATES 69851 31010
NAME_6 UNITED STATES 851 31010
NAME_7 UNITED STATES 69581 90847
NAME_8 UNITED STATES 51 30066
NAME_9 UNITED STATES 6541 30066
NAME_10 UNITED STATES 6851 31904
404A
404B
404C
404D
406
402

FIG. 4

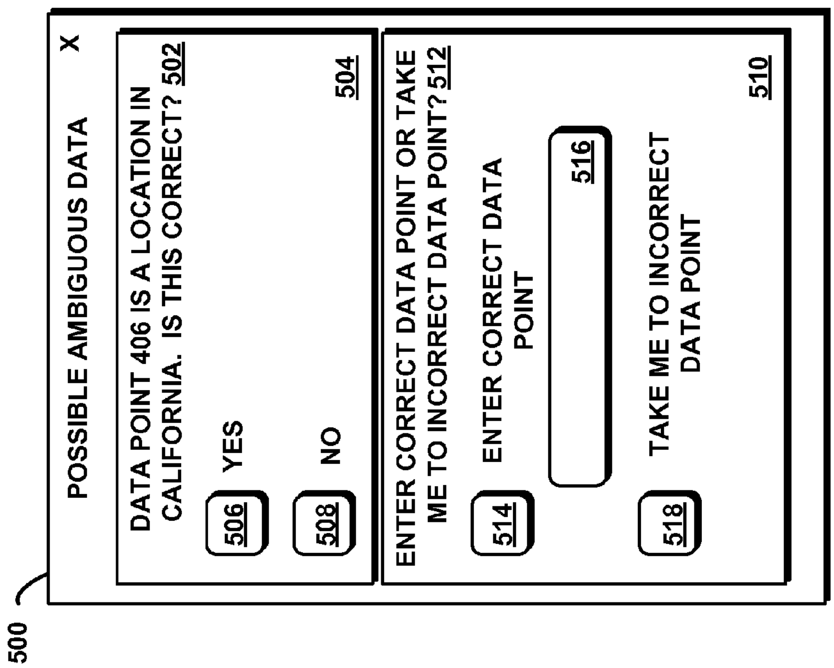

500
POSSIBLE AMBIGUOUS DATA
X
DATA POINT 406 IS A LOCATION IN CALIFORNIA. IS THIS CORRECT? 502
506
YES
508
NO
504
ENTER CORRECT DATA POINT OR TAKE ME TO INCORRECT DATA POINT? 512
514
ENTER CORRECT DATA POINT
516
518
TAKE ME TO INCORRECT DATA POINT
510

FIG. 5A

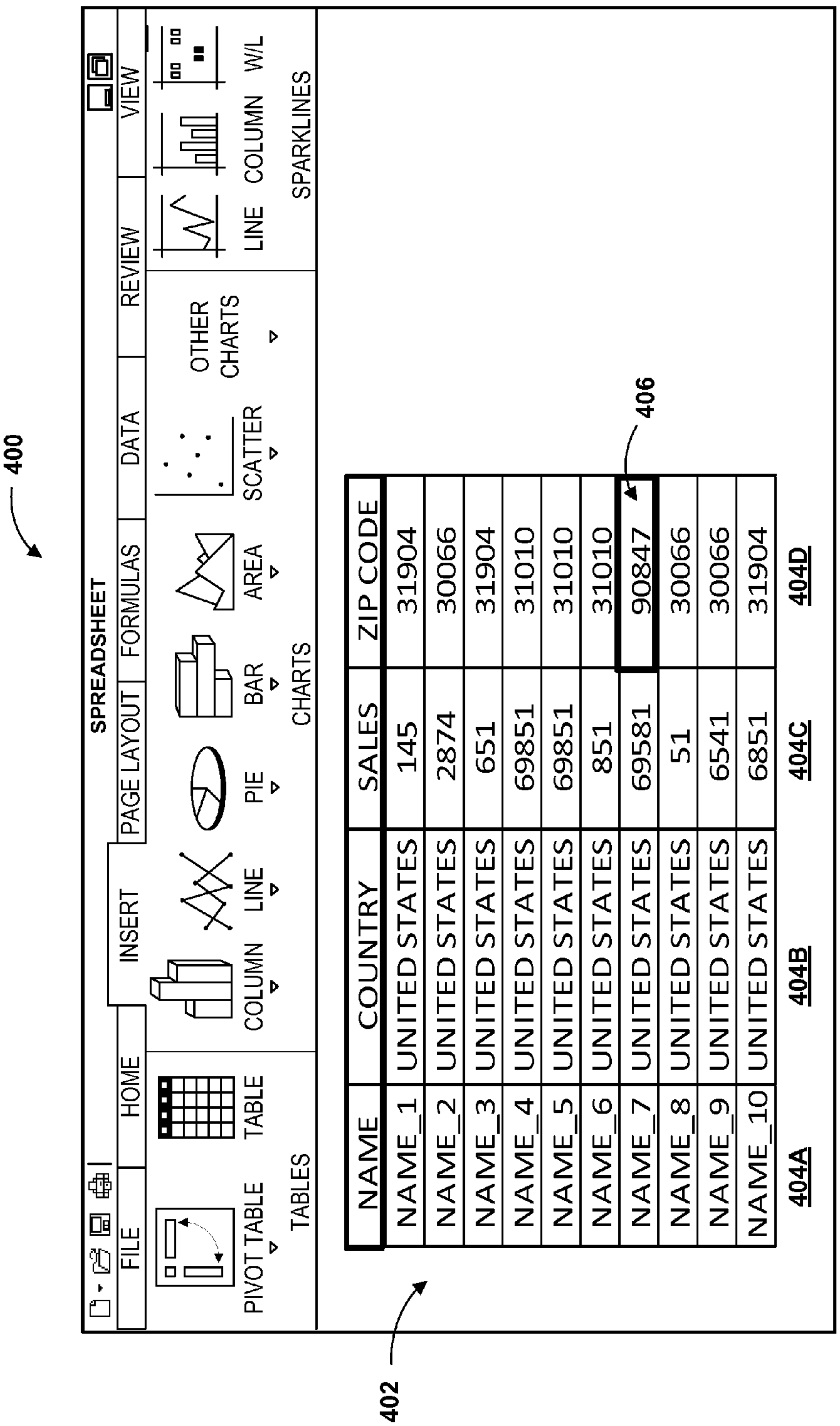

400
SPREADSHEET
FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
PIVOT TABLE
TABLE
TABLES
COLUMN
LINE
PIE
BAR
AREA
SCATTER
OTHER CHARTS
CHARTS
LINE
COLUMN
W/L
SPARKLINES
NAME
COUNTRY
SALES
ZIP CODE
NAME_1
UNITED STATES
145
31904
NAME_2
UNITED STATES
2874
30066
NAME_3
UNITED STATES
651
31904
NAME_4
UNITED STATES
69851
31010
NAME_5
UNITED STATES
69851
31010
NAME_6
UNITED STATES
851
31010
NAME_7
UNITED STATES
69581
90847
NAME_8
UNITED STATES
51
30066
NAME_9
UNITED STATES
6541
30066
NAME_10
UNITED STATES
6851
31904
404A
404B
404C
404D
406
402

FIG. 5B

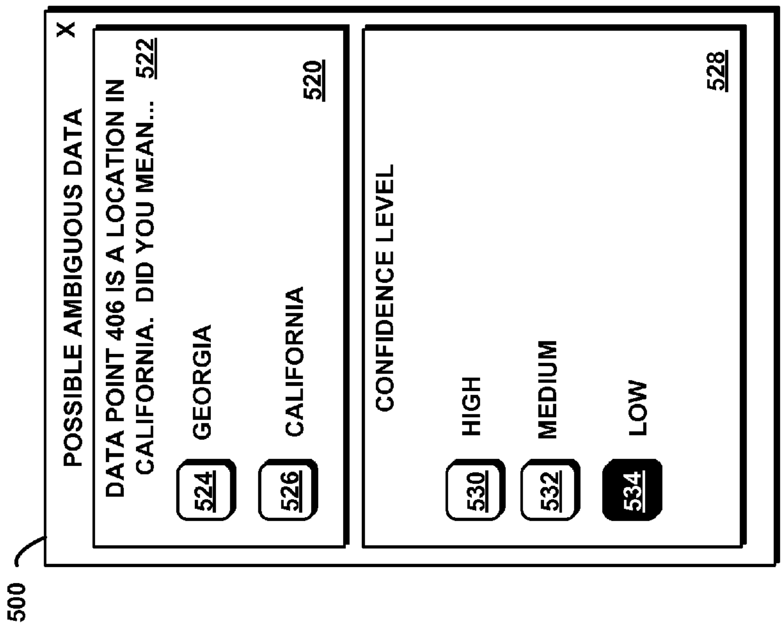

500
POSSIBLE AMBIGUOUS DATA
X
DATA POINT 406 IS A LOCATION IN CALIFORNIA. DID YOU MEAN... 522
524
GEORGIA
526
CALIFORNIA
520
CONFIDENCE LEVEL
530
HIGH
532
MEDIUM
534
LOW
528

FIG. 5C

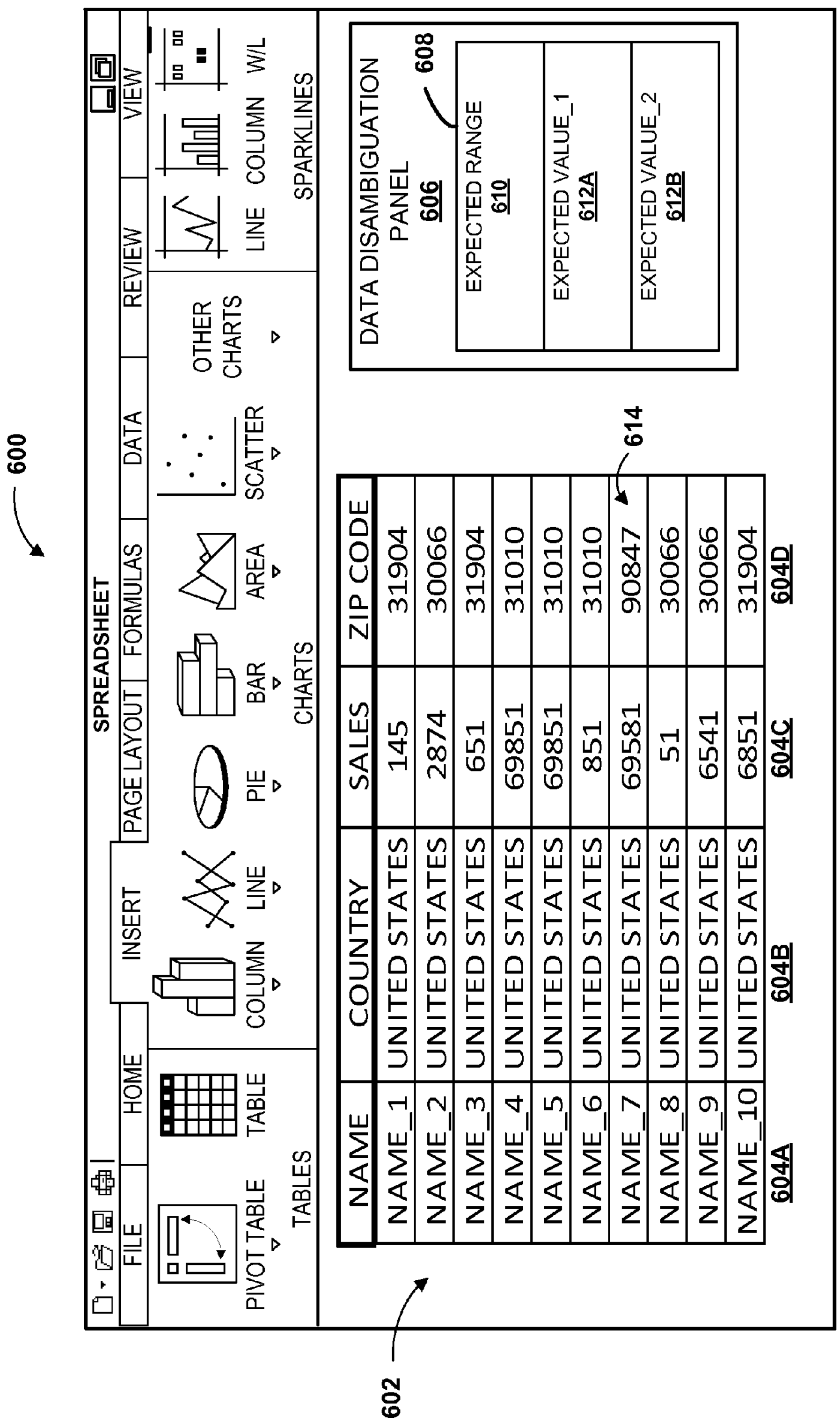

600
602
SPREADSHEET
FILE
HOME
INSERT
PAGE LAYOUT
FORMULAS
DATA
REVIEW
VIEW
PIVOT TABLE
TABLE
TABLES
COLUMN
LINE
PIE
BAR
AREA
SCATTER
OTHER CHARTS
CHARTS
LINE
COLUMN
W/L
SPARKLINES
NAME
COUNTRY
SALES
ZIP CODE
NAME_1 UNITED STATES 145 31904
NAME_2 UNITED STATES 2874 30066
NAME_3 UNITED STATES 651 31904
NAME_4 UNITED STATES 69851 31010
NAME_5 UNITED STATES 69851 31010
NAME_6 UNITED STATES 851 31010
NAME_7 UNITED STATES 69581 90847
NAME_8 UNITED STATES 51 30066
NAME_9 UNITED STATES 6541 30066
NAME_10 UNITED STATES 6851 31904
604A
604B
604C
604D
614
DATA DISAMBIGUATION PANEL
606
608
EXPECTED RANGE
610
EXPECTED VALUE_1
612A
EXPECTED VALUE_2
612B

FIG. 6

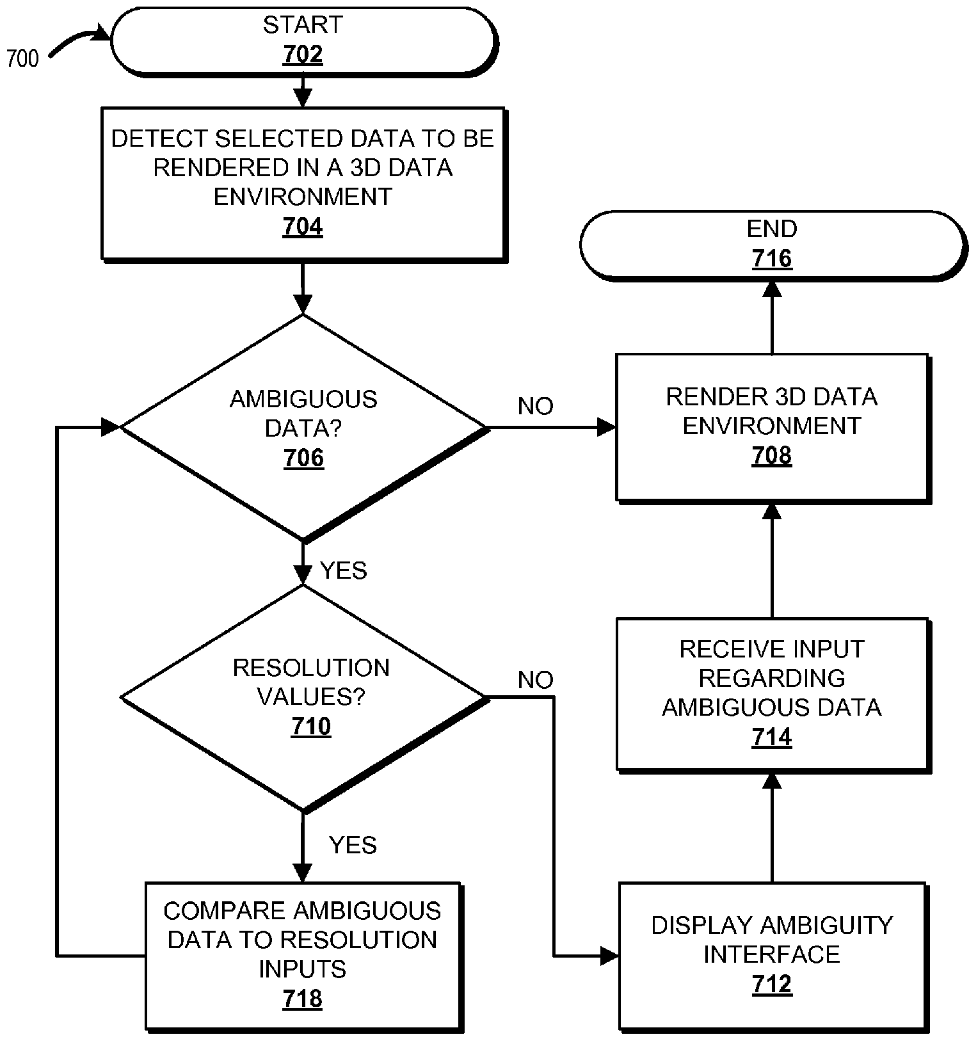

700
START
702
DETECT SELECTED DATA TO BE RENDERED IN A 3D DATA ENVIRONMENT
704
AMBIGUOUS DATA?
706
NO
RENDER 3D DATA ENVIRONMENT
708
END
716
YES
RESOLUTION VALUES?
710
NO
YES
COMPARE AMBIGUOUS DATA TO RESOLUTION INPUTS
718
DISPLAY AMBIGUITY INTERFACE
712
RECEIVE INPUT REGARDING AMBIGUOUS DATA
714

FIG. 7

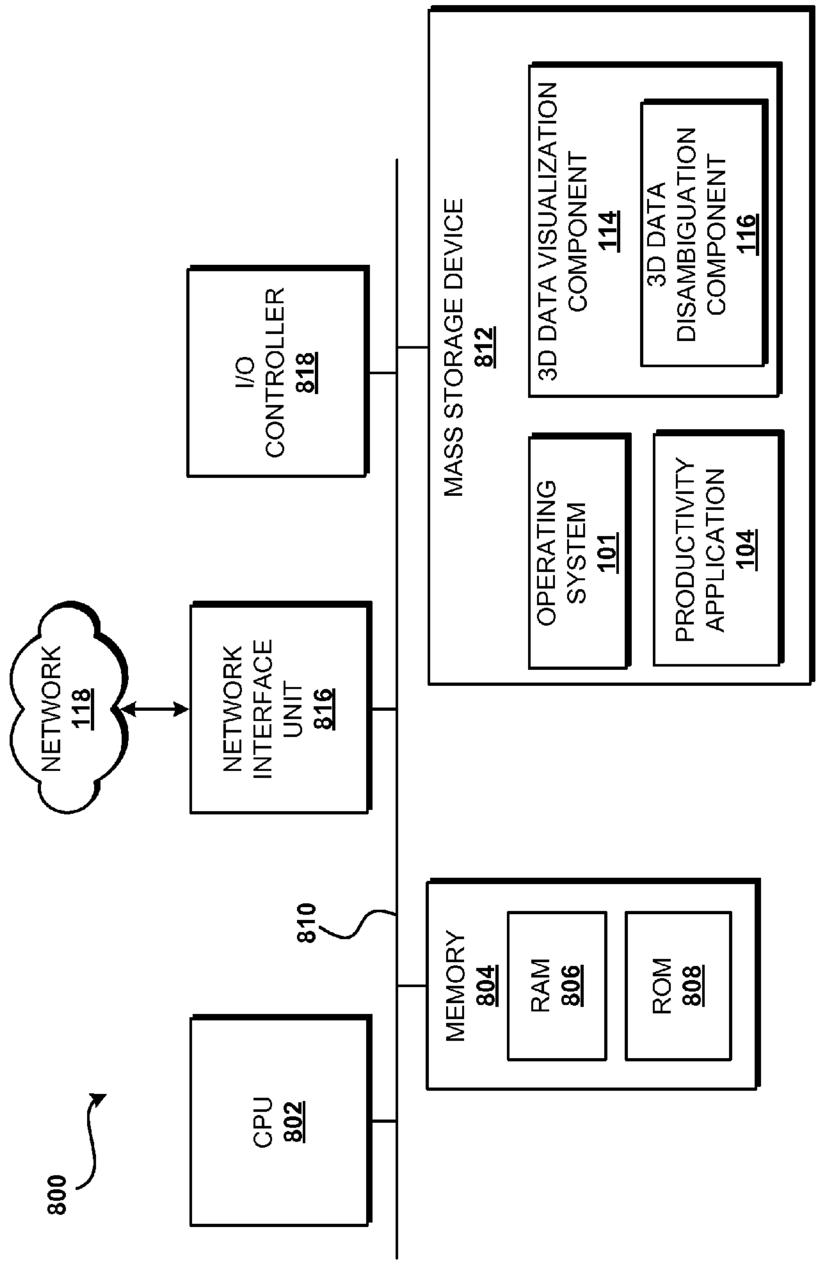

800
CPU
802
810
MEMORY
804
RAM
806
ROM
808
NETWORK
118
NETWORK
INTERFACE
UNIT
816
I/O
CONTROLLER
818
MASS STORAGE DEVICE
812
OPERATING
SYSTEM
101
PRODUCTIVITY
APPLICATION
104
3D DATA VISUALIZATION
COMPONENT
114
3D DATA
DISAMBIGUATION
COMPONENT
116

FIG. 8

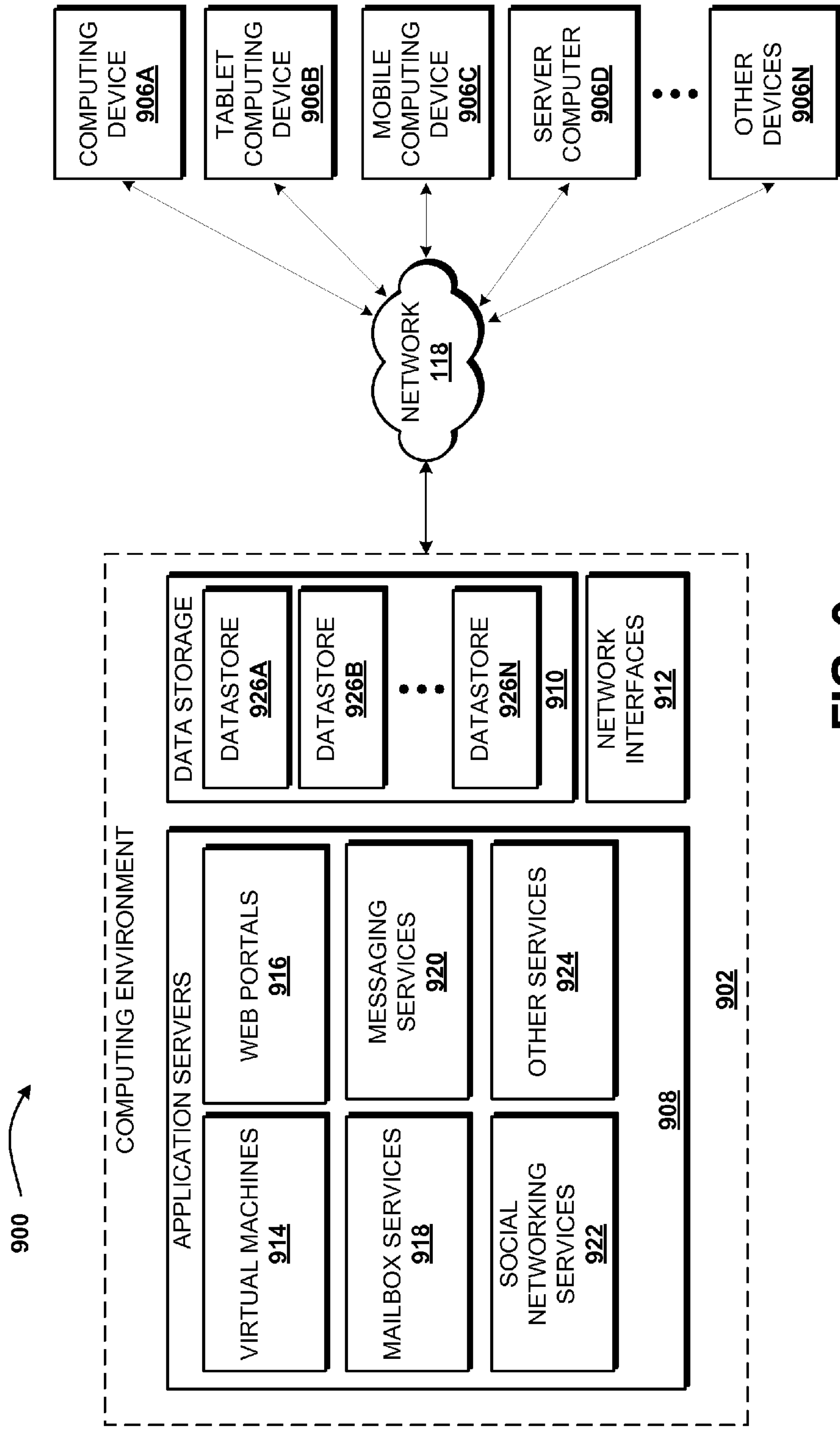

900
COMPUTING ENVIRONMENT
APPLICATION SERVERS
VIRTUAL MACHINES 914
WEB PORTALS 916
MAILBOX SERVICES 918
MESSAGING SERVICES 920
SOCIAL NETWORKING SERVICES 922
OTHER SERVICES 924
908
902
DATA STORAGE
DATASTORE 926A
DATASTORE 926B
DATASTORE 926N
910
NETWORK INTERFACES 912
NETWORK 118
COMPUTING DEVICE 906A
TABLET COMPUTING DEVICE 906B
MOBILE COMPUTING DEVICE 906C
SERVER COMPUTER 906D
OTHER DEVICES 906N

FIG. 9

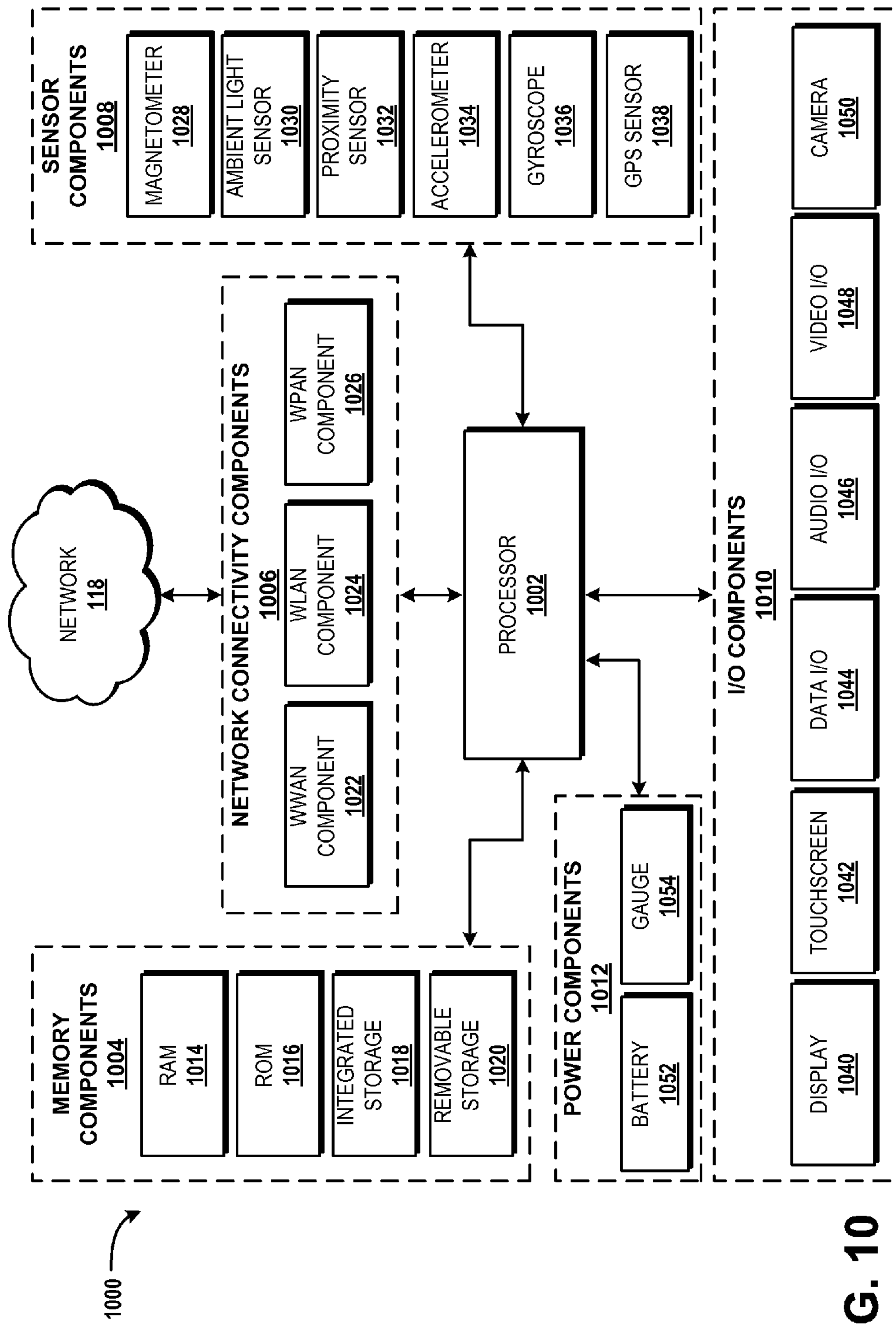

1000
MEMORY COMPONENTS 1004
RAM 1014
ROM 1016
INTEGRATED STORAGE 1018
REMOVABLE STORAGE 1020
NETWORK 118
NETWORK CONNECTIVITY COMPONENTS 1006
WWAN COMPONENT 1022
WLAN COMPONENT 1024
WPAN COMPONENT 1026
SENSOR COMPONENTS 1008
MAGNETOMETER 1028
AMBIENT LIGHT SENSOR 1030
PROXIMITY SENSOR 1032
ACCELEROMETER 1034
GYROSCOPE 1036
GPS SENSOR 1038
PROCESSOR 1002
POWER COMPONENTS 1012
BATTERY 1052
GAUGE 1054
I/O COMPONENTS 1010
DISPLAY 1040
TOUCHSCREEN 1042
DATA I/O 1044
AUDIO I/O 1046
VIDEO I/O 1048
CAMERA 1050

FIG. 10

3D DATA ENVIRONMENT DISAMBIGUATION TOOL

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is related to and claims the benefit of U.S. Provisional Patent Application No. 61/681,851, filed on Aug. 10, 2012, entitled "3D Visualization of Data in Geographical and Temporal Contexts," the entirety of which is hereby incorporated by reference.

BACKGROUND

[0002] A spreadsheet application, reporting application, or other data presentation application may present data in a format for users to gain insight into the data. Some spreadsheet applications present data in cells, which may be organized in a column/row format. A user can input data into one or more cells, or the data can be automatically input into one or more of the cells from one or more databases, data stores, or other sources of data. Additionally, spreadsheet data cells can be populated with data calculated from other spreadsheet data cells. In this manner, the user can interact with data in one convenient location such as one or more spreadsheets rather than at each data source. In some instances, a user may want to interact with the data in a particular format, such as numbers in a cell, which may not be provided by some spreadsheet applications. When using spreadsheet applications, a user may not be cognizant of mistakes, errors or omissions in the data. If there are mistakes, errors or omissions in the data, a user or other entity may make an incorrect determination about one or more values or relationships contained in the data.

[0003] It is with respect to these considerations and others that the disclosure made herein is presented.

SUMMARY

[0004] Concepts and technologies are described herein for providing a 3D data environment disambiguation tool. In accordance with some concepts and technologies disclosed herein, the 3D data environment disambiguation tool provides feedback regarding possible ambiguities associated with one or more data points. In some configurations, the 3D data environment disambiguation tool analyzes geographic data. In further configurations, the 3D data environment disambiguation tool determines a confidence level associated with geographic data. In some configurations, if the confidence level does not satisfy a specified threshold or value, the 3D data environment disambiguation tool can obtain input to correct the possibly incorrect geographic information, and/or submit an input that the ambiguous geographic information is correct. In some configurations, the 3D data environment disambiguation tool can also display alternatives to the ambiguous geographic information.

[0005] According to one aspect, a computer system executes a data disambiguation component. The data disambiguation component can be configured to analyze various portions of data selected to be rendered in a 3D data environment and determine if one or more of the data points in the data are ambiguous. According to various embodiments, the data disambiguation component can generate and/or present one or more user interfaces to resolve an ambiguity. In one example, the data disambiguation component can present an ambiguity interface for receiving various inputs to resolve an ambiguity. In some configurations, a user or other entity can enter an input that the data point is not ambiguous. In other configurations, a user or other entity can input correct data to resolve the ambiguity. In another example, the data disambiguation component can present an ambiguity alternative interface to provide one or more alternatives to the ambiguous data. In a still further example, the data disambiguation component can present a confidence interface for indicating a confidence level of the ambiguous data.

[0006] It should be appreciated that the above-described subject matter may also be implemented as a computer-controlled apparatus, a computer process, a computing system, or as an article of manufacture such as a computer-readable storage medium. These and various other features will be apparent from a reading of the following Detailed Description and a review of the associated drawings.

[0007] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended that this Summary be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. **1** is a system diagram of an illustrative operating environment that may be used to implement various embodiments disclosed herein.

[0009] FIG. **2** is a user interface ("UI") diagram showing spreadsheet data that is selected to be rendered in a 3D data environment, in accordance with some embodiments.

[0010] FIG. **3** is a UI diagram showing the rendering of the data selected in FIG. **2** in a 3D data environment, in accordance with some embodiments.

[0011] FIG. **4** is UI diagram illustrating an example data set that can be rendered in a 3D data environment including a data point that can be disambiguated, in accordance with some embodiments.

[0012] FIG. **5**A is a UI diagram illustrating a screen display providing feedback to resolve ambiguous data, in accordance with some embodiments.

[0013] FIG. **5**B is a UI diagram showing spreadsheet data in which an ambiguous data point is highlighted, in accordance with some embodiments.

[0014] FIG. **5**C is a UI diagram showing an alternative ambiguity interface for resolving an ambiguity associated with a data point, in accordance with some embodiments.

[0015] FIG. **6** is a UI diagram showing a spreadsheet including a data disambiguation panel for receiving values to resolve an ambiguity associated with a data point, in accordance with some embodiments.

[0016] FIG. **7** is a flow diagram showing aspects of a method for providing a 3D data environment disambiguation tool, in accordance with some embodiments.

[0017] FIG. **8** illustrates a computer architecture for a device capable of executing the software components presented herein, in accordance with some embodiments.

[0018] FIG. **9** is a diagram illustrating a distributed computing environment capable of implementing aspects of the embodiments presented herein, in accordance with some embodiments.

[0019] FIG. **10** is a computer architecture diagram illustrating a computing device architecture capable of implementing aspects of the embodiments presented herein.

DETAILED DESCRIPTION

[0020] The following detailed description is directed to a 3D data environment disambiguation tool. A 3D data environment disambiguation tool can be executed by a computer system. In some embodiments, the 3D data environment disambiguation tool can be included as a part of an application that provides 3D visualizations of data. The 3D data environment disambiguation tool may provide feedback that can be used to determine if one or more data within data selected for 3D visualization is ambiguous when viewed against remaining portions of the selected data. As used herein, "ambiguous data" can be used to refer to data that, when analyzed against other data, does not correlate in some predetermined fashion.

[0021] The 3D data environment disambiguation tool may analyze the selected data and determine one or more confidence levels based on possible ambiguities determined when analyzing the selected data. In some configurations, the feedback presented to the user can be based on the one or more confidence levels. In other configurations, the 3D data environment data disambiguation tool can provide a user with one or more alternative suggestions that may have higher confidence levels than the ambiguous data. As used herein, a "3D data environment" can be used to refer to a visualization or simulation of data in a space having three dimensions. In some examples, the three dimensions are represented by a spatial coordinate system, such as a 3-dimensional Euclidean space having three directional axes (e.g. X, Y, and Z).

[0022] While the subject matter described herein is presented in the general context of program modules that execute in conjunction with the execution of an operating system and application programs on a computer system, those skilled in the art will recognize that other implementations may be performed in combination with other types of program modules. Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the subject matter described herein may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like.

[0023] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments or examples. Referring now to the drawings, in which like numerals represent like elements throughout the several figures, aspects of a computing system, computer-readable storage medium, and computer-implemented methodologies for a 3D data environment disambiguation tool and other aspects will be presented.

[0024] Referring now to FIG. **1**, aspects of an operating environment **100** for implementing various embodiments presented herein will be described. The operating environment **100** shown in FIG. **1** includes a computing device **102** operating on or in communication with a network **118**. It should be noted that the concepts and technologies described herein are not limited to computing devices, such as the computing device **102**, in communication with a network, such as network **118**. In some embodiments, the computing device **102** can include a desktop computer, a laptop computer, a notebook computer, an ultra-portable computer, a netbook computer, or a computing device such as a mobile telephone, a tablet device, a slate device, a portable video game device, or the like. Illustrative architectures for the computing device **102** are illustrated and described herein below with reference to FIGS. **8-10**. It should be understood that the concepts and technologies disclosed herein are not limited to an operating environment connected to a network or any external computing system, as various embodiments of the concepts and technologies disclosed herein can be implemented locally on the computing device **102**.

[0025] An operating system **101** is executing on the computing device **102**. The operating system **101** is an executable program for controlling functions on the computing device **102**. The computing device **102** also can execute a productivity application **104**. The productivity application **104**, in some examples, is used by a user **106** to collect, store, manipulate and analyze data stored in spreadsheet data **112**. It should be appreciated that the spreadsheet data **112** is represented as being stored on a single data store for purposes of illustration. It should be appreciated that the spreadsheet data **112** may be stored in one or more data stores accessible to the computing device **102**. Although the concepts and technologies disclosed herein are not limited to any type of data stored in the spreadsheet data **112**, in some examples, data stored in the spreadsheet data **112** may be data associated with various geographic locations, conditions, events, workflow processes, business environments, and the like, with which the user **106** may use in the productivity application **104**.

[0026] In some embodiments, the productivity application **104** may include, but is not limited to, one or more productivity application programs that are part of the MICROSOFT OFFICE family of products from Microsoft Corporation in Redmond, Wash. The examples of the application programs can include a member of, but are not limited, MICROSOFT WORD, MICROSOFT EXCEL, MICROSOFT POWERPOINT, MICROSOFT ACCESS, MICROSOFT VISIO, or MICROSOFT OUTLOOK families of application programs. In other embodiments, the productivity application **104** may include, but is not limited to, one or more productivity application programs that are part of the GOOGLE DOCS family of products from Google, Inc. in Mountain View, Calif. In the described embodiments, the productivity application **104** is illustrated as including the MICROSOFT EXCEL application program. The MICROSOFT EXCEL application program is a spreadsheet application featuring various functionalities including, but not limited to, calculation, graphing tools, data pivot tables, and a macro programming language called VISUAL BASIC for APPLICATIONS. It should be understood that examples provided herein using MICROSOFT EXCEL are illustrative, and should not be construed as limiting in any way.

[0027] In addition to accessing data from the spreadsheet data **112**, the productivity application **104** may also be configured to access data from other sources. In one example, the user **106** may wish to augment data stored in the spreadsheet data **112** with additional geographic information. In such an example, the productivity application **104** may be configured to access map data **122**. The productivity application **104** may be configured to access the map data **122** at a local storage device associated with the computing system **102** and/or may access the map data **122** via the network **118**. The map data **122** may include, among other information, geographic location information, digital representations of maps, and/or other information. The 3D data visualization component **114** can be configured to integrate the map data **122** into data stored in the spreadsheet data **112**.

[0028] The operating environment **100** also can include a geocoding component **120**. The geocoding component **120** may be a component of the computing device **102** or a separate component accessible to the computing device **102**. The geocoding component **120** can be accessed by the productivity application **104** to map or correlate data stored in the spreadsheet data **112** to location data included in or represented by the map data **122**. It should be appreciated that the geocoding component **120** is illustrated as a separate component for purposes of illustration only. In some examples, the geocoding component **120** may be part of one or more other components or programs, including, but not limited to, the productivity application **104**. The geocoding component **120** and/or the map data **122** may be provided using various data sources, including, but not limited to, the BING mapping services provided by Microsoft Corporation in Redmond, Wash. Because additional and/or alternative mapping services are possible and are contemplated, it should be understood that this example is illustrative, and should not be construed as being limiting in any way.

[0029] The data stored in the spreadsheet data **112** can be rendered in a 3D data environment by the 3D data visualization component **114**. It should be understood that the 3D data visualization component **114** can render data obtained from other sources, as described more fully below. As discussed briefly above, the user **106** or another entity may request rendering of portions of the data stored in the spreadsheet data **112** to perform various functions or tasks within a 3D data environment. For example, the user **106** may request a rendering of the data stored in the spreadsheet data **112** for navigating through the data within the 3D data environment. In another example, the user **106** may request rendering of the data stored in the spreadsheet data **112** for creating or recording a "tour." As used herein, a "tour" can refer to a created or recorded movement, path, and/or collection of scenes within a 3D data environment corresponding to the spreadsheet data **112**. The tours can be saved and/or shared to allow other users to view or watch the tour. Thus, a tour or navigation of data or information can include manipulating the 3D data environment and/or simulating movement through or around the 3D data environment. In some configurations, manipulating the 3D data environment includes moving or rotating the 3D data environment about various geometric axes.

[0030] In some configurations, the productivity application **104** may present feedback, in various forms, to indicate that one or more data within the selected data may be ambiguous. For example, data such as the spreadsheet data **112** may include data points including or specifying various United States ZIP codes. One of the data pints may include or specify a ZIP code corresponding to a location in California, while the remaining data points may include or specify ZIP codes corresponding to locations in Georgia. In this and/or similar situations, the productivity application **104** may be configured to utilize a data disambiguation component **116** to analyze the ZIP codes for possible ambiguities. In this example, the data disambiguation component **116** may determine that the ZIP code corresponding to the location in California is ambiguous because it does not correlate to the other ZIP codes corresponding to locations in Georgia.

[0031] FIGS. **2** and **3** provide an example of a manner in which data including geographic information may be displayed in a 3D data environment. FIG. **2** is a UI diagram showing selection of spreadsheet data to be rendered in a 3D data environment. It should be appreciated that the example provided below using a spreadsheet application is for purposes of illustration only and does not limit the disclosure to a spreadsheet application. In particular, other applications or programs that allow a user to interact with data from various sources may also be used. Illustrated in FIG. **2** is a display **110** that includes a representation of a spreadsheet **202**. The spreadsheet **202** has columns **204**A-G (hereinafter collectively and/or generically referred to as "columns **204**"). The columns **204** may be populated by data stored as or in the spreadsheet data **112** illustrated in FIG. **1**. It should be appreciated that the columns **204** may also be populated from other types of data stored in the spreadsheet data **112** or other sources. For example, the data may include map information obtained from the map data **122** and/or other sources. Because the illustrated spreadsheet **202** is illustrative, the illustrated embodiment should not be construed as being limiting in any way.

[0032] In the spreadsheet **202**, a column **204**F has been populated with ZIP codes retrieved from the map data **122** using the geocoding component **120**. There may be several ways in which the columns **204** or other data contained in the spreadsheet **202** can be populated with data. For example, and not by way of limitation, the user **106** can manually enter the data. In another example, and not by way of limitation, the data may be automatically populated within the spreadsheet **202** with data obtained from other sources such as, for example, the geocoding component **120**, the map data **122**, databases, data stores, internet data sources, and/or other sources. Additionally, the data within the spreadsheet **202** may be based on other data and therefore need not originate from an external source. For example, the data within the spreadsheet **202** may be the result of one or more arithmetic operations on data in one or more of the columns **204**. It should be understood that these embodiment are illustrative and should not be construed as being limiting in any way.

[0033] According to various embodiments of the concepts and technologies disclosed herein, data in the spreadsheet **202** can be selected and rendered in a 3D data environment. For example, a request, command, or input may be received to render data included in the spreadsheet **202** in a 3D data environment. In one contemplated example, one or more of the columns **204** of data within the spreadsheet **202**, and/or particular records included in the spreadsheet **202**, can be selected for rendering by the 3D data visualization component **114**. It should be understood that data in the spreadsheet **202** also can be selected by rows. The 3D visualization component **114** can identify the selected data and render the selected data in a 3D data environment. Rendering of the data in the spreadsheet **202** is illustrated and described in additional detail below.

[0034] FIG. **3** is a UI diagram illustrating an example screen display of data from spreadsheet **202** rendered in a 3D data environment. In FIG. **3**, the data included in a spreadsheet, for example the spreadsheet **202** described above, has been selected. In one contemplated example, a user **106** may select data in the spreadsheet **202**, and may request, command, or direct that the 3D data visualization component **114** render the selected data. It should be understood that the illustrated rendering is illustrative and should not be construed as being limiting in any way.

[0035] The screen display shown in FIG. **3** includes a map **300** showing the rendered data. The map **300** is illustrated as being included in the display **110**, which can be presented on the monitor **108**. The map **300** is illustrated as having multiple data points **302**, which can be spread across and/or throughout the map **300** (in this example, a map of the United States). As shown, the data points **302** can include clusters of data points **304**A, **304**B and **304**C (hereinafter collectively and/or generically referred to as "clusters **304**"). The clusters **304** can include groups and/or sets of the data points **302**. Because data can be represented by other types of indicators and/or can be presented without including clusters **304**, it should be understood that the illustrated map **300** is illustrative and should not be construed as being limiting in any way.

[0036] It can be appreciated from the map **300** that if a placement, associated value, or other aspect of one or more of the data points **302** shown on the map **300** are incorrect, that the user **106** or other viewer may incorrectly interpret various relationships of the data points **302**. As used herein, "incorrect" means that a data point does not comply with an expected value. For example, if the data underlying the cluster **304**C includes incorrect geographical information, whereby the cluster **304**C should actually be represented as being physically and/or geographically located closer to the cluster **304**A, the user **106** may incorrectly interpret the spatial relationship between the cluster **304**A and **304**C. In another example, the data underlying the cluster **304** may have missing values, misspelling, or other inaccuracies. The data disambiguation component **116** may be configured to check for ambiguities in the data points **302**, to provide feedback regarding possible ambiguities, and to obtain input to accept the ambiguous data as being correct data. In some embodiments, these and other functions of the data disambiguation component **116** may be used to reduce a probability that the data points **302** in the map **300** are represented inaccurately in the display **110**.

[0037] FIG. **4** is a UI diagram illustrating an example data set than can be disambiguated by the data disambiguation component **116**. Illustrated in FIG. **4** is a spreadsheet **400** including data **402**. The data **402** includes columns **404**A-**404**D of data (hereinafter collectively and/or generically referred to as "columns **404**"). The productivity application **104** may receive an input from the user **106** (or other entity) that the data **402** is to be rendered in a 3D data environment. The productivity application **104** can be configured to invoke the data disambiguation component **116** to analyze the data **402** for possible ambiguities at any time. It should be appreciated that, although the concepts and technologies described herein refer to geographical ambiguities, other types of ambiguities may be analyzed by the data disambiguation component **116** and are within the scope of the present disclosure.

[0038] If the data disambiguation component **116** is invoked, from the productivity application **104** or another entity, the data disambiguation component **116** can analyze the data **402** for possible ambiguities. In the context of the present example, the data disambiguation component **116** can analyze the data **402** for geographic ambiguities. In the illustrated example, the data disambiguation component **116** may analyze the data **402** and determine that data point **406**, corresponding to a ZIP code in column **404**D, is ambiguous. In one configuration, the data disambiguation component **116** may determine that the data point **406** is ambiguous because the other ZIP codes included in column **404**D correspond to locations in Georgia, while the data point **406** corresponds to a ZIP code for a location in California. Thus, it can be appreciated that the data disambiguation component **116** can be configured to analyze data to identify ambiguities based upon similarities with and/or differences from other data. These and other aspects of the data disambiguation component **116** are illustrated and described in additional detail below, particularly with reference to FIGS. **5**-**7**.

[0039] FIG. **5**A is a UI diagram illustrating a screen display providing feedback to resolve ambiguous data. Illustrated in FIG. **5**A is an ambiguity interface **500**. In some configurations, the ambiguity interface **500** is displayed if the data disambiguation component **116** detects one or more ambiguous data points. Thus, for example, the ambiguity interface **500** may be displayed by the disambiguation component **116** in response to detecting the data point **406** shown in FIG. **4**. It should be understood that this embodiment is illustrative, and should not be construed as being limiting in any way.

[0040] The ambiguity interface **500** can include an ambiguity presentation interface **502**. In one configuration, the ambiguity presentation interface **502** displays an ambiguous data point. In the example described above with reference to FIG. **4**, the ambiguity presentation interface **502** can display information **504** indicating that the data point **406** corresponds to a location in California. The information **502** also can include a request to confirm that the data point **406** is correct. Additionally, the ambiguity presentation interface **502** includes a selection box **506** that, when selected, indicates that the data point **406** is correct, and a selection box **508** that, when selected, indicates that the data point **406** is incorrect.

[0041] If the data disambiguation component **116** determines that the selection box **506** has been selected, thereby resolving the ambiguity by determining that the data point **406** is correct, the data disambiguation component **116** can determine if other ambiguous data points to be resolved exist. If there are no further ambiguous data points to be resolved, the data disambiguation component **116** can determine and/or inform the 3D data visualization component **114** that all ambiguities have been resolved. The 3D data visualization component **114** can render a 3D data environment. In some configurations, the 3D data visualization component **114** may render a 3D data environment prior to, in conjunction with, or after the resolution of one or more ambiguous data points.

[0042] If the data disambiguation component **116** determines that the selection box **508** has been selected, thereby confirming that the data point **406** is incorrect, an ambiguity resolution interface **510** can be displayed. In the illustrated embodiment, the ambiguity resolution interface **510** is displayed in the ambiguity interface **500**, though this is not necessarily the case. The ambiguity resolution interface **510** can include a caption **512** requesting correction input to correct the ambiguity of the data point **406** and/or an option to display the data point **406** in the spreadsheet **400**. If the data disambiguation component **116** determines that the selection box **514** has been selected, an input box **516** can be displayed. A correction input can be entered in the input box **516**.

[0043] If the data disambiguation component **116** determines that the selection box **518** has been selected, the data disambiguation component **116** can interpret the input as an ambiguous data display input and/or display an interface highlighting the incorrect data point, as illustrated in FIG. **5**B.

[0044] FIG. **5**B is a UI diagram illustrating highlighting of an incorrect data point, as described above with reference to FIG. **5**A. In particular, FIG. **5**B illustrates a screen display showing highlighting of the data point **406** in the spreadsheet **400** shown in FIG. **4**. In some embodiments, the screen display shown in FIG. **5**B can be presented in response to detecting selection of the selection box **518** shown in FIG. **5**A. Because the screen display shown in FIG. **5**B can be presented at additional and/or alternative times, it should be understood that this embodiment is illustrative, and should not be construed as being limiting in any way. In FIG. **5**B, the data point **406** is shown highlighted. The user **106** or other entity may perform various tasks with respect to the highlighted data point **406**. For example, the user **106** or other entity may correct the data point **406**, correct other data points, or continue with the use of the spreadsheet **400** in the configuration illustrated in FIG. **5**B. It should be appreciated that the type of data that may be determined to be ambiguous may include data other than numerical data. For example, and not by way of limitation, geographical data can include names of locations, mailing addresses, etc., all of which are considered to be within the scope of the present disclosure.

[0045] FIG. **5**C is a UI line diagram showing an alternative configuration of the ambiguity interface **500**. In some configurations, the data disambiguation component **116** may determine that the data point **406** is ambiguous and may determine one or more possible alternatives to the data point **406**. For example, as shown in FIG. **5**C, the ambiguity interface **500** can include an ambiguity alternative interface **520** for presenting the one or more possible alternatives to the data point **406**. The ambiguity alternative interface **520** can include a notification **522** that the data point **406** is ambiguous because it corresponds to a location in California rather than a location in Georgia. The ambiguity alternative interface **520** also can allow selection of one or more of the possible alternatives to the data point **406**. As shown in FIG. **5**C, the ambiguity alternative interface **520** includes a selection box **524** corresponding to Georgia and a selection box **525** corresponding to California. It should be understood that this example is illustrative, and should not be construed as being limiting in any way.

[0046] The data disambiguation component **116** can determine that the data point **406** actually corresponds to a location in Georgia by detecting selection of selection box **524**. The data disambiguation component **116** can determine that the data point **406** actually corresponds to a location in California by detecting selection of the selection box **526**. Although not limited to any manner in which the alternatives presented by the selection box **524** or the selection box **526** are determined, in one example, the data disambiguation component **116** may be configured to analyze the column **404**D of the spreadsheet **400** to determine possible patterns in the data and/or other aspects of the data to determine possible alternatives, such as the alternatives presented by the selection box **524** or the selection box **526**.

[0047] In some configurations, the data disambiguation component **116** may also determine a confidence level associated with ambiguities detected in the data. As used herein, a "confidence level" can refer to a value representing a degree of correlation between data and an ambiguous data point included in and/or associated with the data. For example, a low confidence level may indicate a data point that has a low level of correlation or similarity (or even no correlation or similarity) to other data. In another example, a high confidence level may indicate a data point that has a high level of correlation and/or similarity to other data. In the example illustrated in FIG. **5**C, the data point **406** corresponds to a location in California, wherein the remaining data in the column **404**D corresponds to locations in Georgia. In this example, the data disambiguation component **116** may determine that the data point **406** has a low confidence level, indicating that the location in California has a low level of correlation or similarity with the other data points in Georgia. As described above, there may be various reasons why a data point is determined to have a low confidence level. For example, the data disambiguation component **116** may determine that the data point **406** has a low confidence level because of the relatively large variance between the location represented by the data point **406** and the other data points in the column **404**D.

[0048] According to various embodiments, the confidence level determined by the data disambiguation component **116** can be presented or displayed. In the ambiguity interface **500** illustrated in FIG. **5**C, a confidence interface **528** is displayed. Displayed in the confidence interface **528** are a high confidence level indicator **530**, a medium confidence level indicator **532**, and a low confidence level indicator **534**. As described above, the data disambiguation component **116** can be configured to determine that the data point **406** has a low confidence level. Thus, as illustrated, the low confidence level **534** can be highlighted. It should be understood that this example is illustrative, and should not be construed as being limiting in any way. It should also be understood that the confidence levels described herein can be determined for a degree or level to which the data disambiguation component **116** believes a data point is ambiguous. In such an embodiment, a high confidence level can indicate a high probability that a possibly ambiguous data point is ambiguous, while a low confidence level can indicate a low probability that a possibly ambiguous data point is ambiguous. Because the confidence levels disclosed herein can be developed for various aspects of identifying ambiguities, it should be understood that these examples of confidence levels and/or what these confidence levels mean are illustrative and should not be construed as being limiting in any way.

[0049] The data disambiguation component **116** may receive inputs from various sources to determine ambiguities in data. In the configuration illustrated in FIGS. **5**A-**5**C, the data disambiguation component **116** analyzed the data **402** from the spreadsheet **400**. The data disambiguation component **116** determined that the data point **406** was ambiguous based on the other data in column **404**D. In an alternative embodiment, the disambiguation component **116** may use other sources of data to determine whether or not one or more data points are ambiguous.

[0050] FIG. **6** is a UI diagram illustrating a screen display showing obtaining values for use in determining if a data point is ambiguous. Illustrated in FIG. **6** is the spreadsheet **600** including data **602**. The data **602** is organized in columns **604**A-**604**D. Also illustrated is a data disambiguation panel **606**. The data disambiguation panel **606** can be configured to receive input from one or more sources, including the user **106**, to resolve possible ambiguities in the data **602** prior to, in conjunction with, or after rendering the data **602** in a 3D data environment.

[0051] The data disambiguation panel **606** can include value boxes **608** that can be used by the data disambiguation component **116** to obtain data for resolving ambiguities. In some configurations, it may be useful for the data disambiguation component **116** to have as an input one or more expected ranges of data in data **602**. If a data point is within the expected range entered in expected range box **610**, the data disambiguation component **116** may be configured to determine that the data point is not ambiguous. If a data point is not within the expected range entered into the expected range box **610**, the data disambiguation component **116** may be configured to determine that the data point is ambiguous or may be configured to make additional determinations. In one configuration, the range entered into the expected range box **610** may correlate to a geographical area, such as a city, state or country. In another configuration, the range entered into the expected range box **610** may be a range of ZIP codes.

[0052] Another example of values that may be used by the data disambiguation component **116** is expected values. In this configuration, the data disambiguation component **116** may be provided with one or more values that are expected or anticipated. Illustrated in FIG. **6** are an expected value__1 box **612**A and an expected value__2 box **612**B. In one configuration, if a data point determined to be ambiguous is the expected value entered in the expected value__1 box **612**A or the expected value__2 box **612**B, the data disambiguation component **116** may be configured to determine that the data point is not ambiguous. If the data point is not the expected value entered in the expected value__1 box **612**A or the expected value__2 box **612**B, the data disambiguation component **116** may be configured to determine that the data point is ambiguous.

[0053] In some configurations, the data disambiguation component **116** may use one or more of the values entered in the value box **608** of the data disambiguation panel **606** to determine ambiguities in the data. In one example, the data disambiguation component **116** may be configured to receive as an input the range entered in expected range box **610**, the value entered in expected value__1 box **612**A, and the value entered in expected value__2 box **612**B. The data disambiguation component **116** may be configured to compare a data point to one value, and if a determination is made that the data point is ambiguous, move to a next value to determine ambiguity.

[0054] For example, the data disambiguation panel **606** may receive as input in the expected range box **610** a range of 30000-32000, corresponding to an expected range of ZIP codes in the column **604**D. Continuing with the present example, the data disambiguation panel **606** may receive as an input in the expected value__1 box **612**A a value of 31904 and in the expected value__2 box **612**B a value of 90847. The data disambiguation component **116** may determine possible ambiguities in the column **604**D prior to, in conjunction with, or after rendering the data **602** in a 3D data environment. In one configuration, the data disambiguation component **116** may determine that the data point **614** is ambiguous. In other configurations, as described above, the data disambiguation component **116** may determine that the data point **614** has a low confidence level based on the value when compared to the other values in the column **604**D.

[0055] The data disambiguation component **116** may be configured to use the values in the value box **608** to resolve the possible ambiguity. Thus, the data disambiguation component **116** may compare the data point **614** value ("90847") against the range entered in the expected range box **610**. In the present example, the value "90847" is outside of the range of 30000-32000, and thus, the data point **614** may be determined to be ambiguous. The data ambiguity component **116** may compare the data point **614** to the value entered into the expected value__1 box **612**A ("31904"). In this example, because the values are not the same, the data ambiguity component **116** may maintain the determination that the data point **614** is ambiguous. The data ambiguity component **116** may compare the data point **614** to the value entered into the expected value__1 box **612**A ("90847"). The data disambiguation component **116** may determine that the values match and that the data point **614** is not ambiguous. It should be appreciated that the present disclosure is not limited to any particular order of determination nor is it limited to any particular type of values that may be entered in the data disambiguation panel **606**.

[0056] The data disambiguation component **116** may also be configured to compare data values, determine if any values deviate a certain amount or degree from an expected amount, thus indicating an ambiguity, and resolve the ambiguity without user input. For example, data point **614** can have a numerical value of "90847". The remaining numerical value of the data points in the column **604**D (representing zip codes) generally are in the lower 30000 range. Thus, the data disambiguation component **116** may be configured to determine that the numerical value of 90847 lies outside of expected norms, such as 30000 plus or minus a certain amount. In that configuration, the data disambiguation component **116** may look to other data, such as data contained in the columns **604**A, **604**B and/or **604**C, to resolve the ambiguity. In some configurations, if the data is insufficient to resolve the ambiguity, the data disambiguation component **116** may be configured to request additional information.

[0057] In another configuration, the data disambiguation component **116** may be configured to anticipate possible ambiguities. The data disambiguation component **116** may be configured to anticipate that some location data may have a higher probability of ambiguity than location data. In one example, the data in the column **604**B may be a city instead of a country. The data disambiguation component **116** may be configured to recognized that there may be two cities with the same name, indicating an inherent ambiguity within the data itself. In that example, the data disambiguation component **116** may be configured to ask for additional information to avoid an ambiguity, such as a state, zip code or address. In that configuration, an entire set of data points can be disambiguated instead of individually.

[0058] In a further configuration, the data disambiguation component **116** may be configured to store resolved ambiguities for later use. For example, the data disambiguation component **116** may have resolved in a previous instance that the data point **614** is ambiguous. In another example, the data disambiguation component **116** may compare a possibly ambiguous data point to a previously disambiguated data point. In a different instance, the data disambiguation component **116** may determine that a similar data point is ambiguous and use the information stored for the disambiguation of the data point **614** to disambiguate the similar, ambiguous data point. In one example, the data disambiguation component **116** may use the disambiguation information to disambiguate similar occurrences of the ambiguous data without requiring additional input. In other configurations, the data disambiguation component **116** may present the disambiguation information in a user interface to assist a user in providing an input to disambiguate a presently ambiguous data.

[0059] FIG. **7** is a flow diagram showing aspects of a method **700** for providing 3D data environment disambiguation tool within a 3D data environment are illustrated, according to an illustrative embodiment. It should be understood that the operations of the method **700** are not necessarily presented in any particular order and that performance of some or all of the operations of the method **700** in an alternative order(s) is possible and is contemplated. The operations of the method **700** have been presented in the demonstrated order for ease of description and illustration. Operations may be added, omitted, and/or performed simultaneously, without departing from the scope of the appended claims.

[0060] It also should be understood that the illustrated method **700** can be ended at any time and need not be performed in its entirety. Some or all operations of the method **700**, and/or substantially equivalent operations, can be performed by execution of computer-readable instructions included on a computer-storage media, as defined herein. The term "computer-readable instructions," and variants thereof, as used in the description and claims, is used expansively herein to include routines, applications, application modules, program modules, programs, components, data structures, algorithms, and the like. Computer-readable instructions can be implemented on various system configurations, including single-processor or multiprocessor systems, minicomputers, mainframe computers, personal computers, hand-held computing devices, microprocessor-based, programmable consumer electronics, combinations thereof, and the like.

[0061] Thus, it should be appreciated that the logical operations described herein are implemented (1) as a sequence of computer implemented acts or program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance and other requirements of the computing system. Accordingly, the logical operations described herein are referred to variously as states, operations, structural devices, acts, or modules. These operations, structural devices, acts, and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof.

[0062] The operations of the method **700** are described herein below as being implemented, at least in part, by the productivity application **104**, the 3D data visualization component **114**, and/or the data disambiguation component **116**, or combinations thereof. One or more of the operations of the method **700** may alternatively or additionally be implemented, at least in part, by the similar components in either the computing device **102** or other device such as, for example, a server computer.

[0063] The method **700** begins at operation **702** and proceeds to operation **704**, wherein the computing device **102** detects selected data to be rendered in a 3D data environment. In some embodiments, the data to be rendered can be selected by the user **106** or other entity. The selection of the data can be received by the productivity application **104** executing on the computing device **102**. Various methods can be used to select the data to be rendered in the 3D data environment. For example, the data can be selected using a keyboard, a mouse or a monitor if the monitor is configured to be a touchscreen capable of receiving tactile inputs. It should be appreciated that the concepts and technologies disclosed herein are not limited to any particular data selection input method. Additionally, it should be appreciated that the concepts and technologies disclosed herein are not limited to any particular type of selected data. For example, data can selected from the spreadsheet data **112**, the map data **122**, other sources of data (not shown), or any combination thereof.

[0064] From operation **704**, the method **700** proceeds to operation **706**, wherein the data disambiguation component **116** determines if one or more of the selected data is ambiguous. The determination at operation **706** may be made according to various configurations. As discussed with reference to the examples provided in FIGS. **4** and **5**A-**5**C, a data point or value may be determined to be ambiguous based upon that point or value not corresponding to other points or values included in the data. If a determination is made that the selected data is not ambiguous, the method proceeds to operation **708**, wherein the selected data is rendered in a 3D data environment. From operation **708**, the method **700** proceeds to operation **716**, wherein the method **700** ends.

[0065] If the data disambiguation component **116** determines, in operation **706**, that one or more of the selected data is ambiguous, the method **700** proceeds to operation **710**. In operation **710**, the data disambiguation component **116** determines if there are resolution values. As explained above, a resolution value is a value or range of values that can be used to resolve ambiguities, such as the values entered in the data disambiguation panel **606** in FIG. **6**. If the data disambiguation component **116** determines that there are no resolution values, the method **700** proceeds to operation **712**, wherein the ambiguity interface **500** is displayed. The data disambiguation component **116** can use input obtained via the ambiguity interface **500** to resolve possible ambiguities.

[0066] From operation **712**, the method **700** proceeds to operation **714**, wherein the data disambiguation component **116** receives one or more inputs via the ambiguity interface **500** regarding the ambiguous data. The data disambiguation component **116** can obtain one or more inputs using the ambiguity interface **500** to resolve ambiguities. From operation **714**, the method **700** proceeds to operation **708**, wherein the selected data is rendered in a 3D data environment. From operation **708**, the method **700** proceeds to operation **716**, wherein the method **700** ends.

[0067] If the data disambiguation component **116** determines, in operation **710**, that there are resolution values, the method **700** proceeds to operation **718**. In operation **718**, the data disambiguation component **116** compares the ambiguous data to the resolution values. In operation **718**, the data disambiguation component **116** can compare the ambiguous data to a first value, such as a range entered into the expected range box **610**.

[0068] From operation **718**, the method **700** returns to operation **706**, wherein the data disambiguation component **116** again determines if the results of the check indicate that the selected data is ambiguous. If the selected data is determined to not be ambiguous, the method **700** proceeds to operation **708** in the manner described above. If the selected data is determined to be ambiguous based on the first comparison, the method **700** proceeds to operation **710** in the manner described above. The method **700** can continue until the data ambiguity component **116** has compared the ambiguous data to the resolution inputs, such as values entered into the expected value_1 box **612**A and/or a value entered into the expected value_2 box **612**B. Once the ambiguous data have been compared to the resolution values by the data disambiguation component **116**, the method **700** proceeds to operation **712**, as described above.

[0069] FIG. **8** illustrates an illustrative computer architecture **800** for a device capable of executing the software components described herein for providing a 3D environment data disambiguation tool. Thus, the computer architecture **800** illustrated in FIG. **8** illustrates an architecture for a server computer, mobile phone, a PDA, a smart phone, a desktop computer, a netbook computer, a tablet computer, and/or a laptop computer. The computer architecture **800** may be utilized to execute any aspects of the software components presented herein.

[0070] The computer architecture **800** illustrated in FIG. **8** includes a central processing unit ("CPU") **802**, a system memory **804**, including a random access memory **806** ("RAM") and a read-only memory ("ROM") **808**, and a system bus **810** that couples the memory **804** to the CPU **802**. A basic input/output system containing the basic routines that help to transfer information between elements within the computer architecture **800**, such as during startup, is stored in the ROM **808**. The computer architecture **800** further includes a mass storage device **812** for storing the operating system **101** from FIG. **1** and one or more application programs including, but not limited to, the productivity application **104**, the 3D data visualization component **114** and the data disambiguation component **116**.

[0071] The mass storage device **812** is connected to the CPU **802** through a mass storage controller (not shown) connected to the bus **810**. The mass storage device **812** and its associated computer-readable media provide non-volatile storage for the computer architecture **800**. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, it should be appreciated by those skilled in the art that computer-readable media can be any available computer storage media or communication media that can be accessed by the computer architecture **800**.

[0072] Communication media includes computer readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics changed or set in a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer-readable media.

[0073] By way of example, and not limitation, computer storage media may include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. For example, computer media includes, but is not limited to, RAM, ROM, EPROM, EEPROM, flash memory or other solid state memory technology, CD-ROM, digital versatile disks ("DVD"), HD-DVD, BLU-RAY, or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store the desired information and which can be accessed by the computer architecture **900**. For purposes of the claims, the phrase "computer storage medium," and variations thereof, does not include waves or signals per se and/or communication media.

[0074] According to various embodiments, the computer architecture **800** may operate in a networked environment using logical connections to remote computers through a network such as the network **118**. The computer architecture **800** may connect to the network **118** through a network interface unit **816** connected to the bus **810**. It should be appreciated that the network interface unit **816** also may be utilized to connect to other types of networks and remote computer systems. The computer architecture **800** also may include an input/output controller **818** for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus. Similarly, the input/output controller **818** may provide output to a display screen, a printer, or other type of output device.

[0075] It should be appreciated that the software components described herein may, when loaded into the CPU **802** and executed, transform the CPU **802** and the overall computer architecture **800** from a general-purpose computing system into a special-purpose computing system customized to facilitate the functionality presented herein. The CPU **802** may be constructed from any number of transistors or other discrete circuit elements, which may individually or collectively assume any number of states. More specifically, the CPU **802** may operate as a finite-state machine, in response to executable instructions contained within the software modules disclosed herein. These computer-executable instructions may transform the CPU **802** by specifying how the CPU **802** transitions between states, thereby transforming the transistors or other discrete hardware elements constituting the CPU **802**.

[0076] Encoding the software modules presented herein also may transform the physical structure of the computer-readable media presented herein. The specific transformation of physical structure may depend on various factors, in different implementations of this description. Examples of such factors may include, but are not limited to, the technology used to implement the computer-readable media, whether the computer-readable media is characterized as primary or secondary storage, and the like. For example, if the computer-readable media is implemented as semiconductor-based memory, the software disclosed herein may be encoded on the computer-readable media by transforming the physical state of the semiconductor memory. For example, the software may transform the state of transistors, capacitors, or other discrete circuit elements constituting the semiconductor memory. The software also may transform the physical state of such components in order to store data thereupon.

[0077] As another example, the computer-readable media disclosed herein may be implemented using magnetic or optical technology. In such implementations, the software presented herein may transform the physical state of magnetic or optical media, when the software is encoded therein. These transformations may include altering the magnetic characteristics of particular locations within given magnetic media. These transformations also may include altering the physical features or characteristics of particular locations within given optical media, to change the optical characteristics of those locations. Other transformations of physical media are possible without departing from the scope and spirit of the present description, with the foregoing examples provided only to facilitate this description.

[0078] In light of the above, it should be appreciated that many types of physical transformations take place in the computer architecture **800** in order to store and execute the software components presented herein. It also should be appreciated that the computer architecture **800** may include other types of computing devices, including hand-held computers, embedded computer systems, personal digital assistants, and other types of computing devices known to those skilled in the art. It is also contemplated that the computer architecture **800** may not include all of the components shown in FIG. **8**, may include other components that are not explicitly shown in FIG. **8**, or may utilize an architecture completely different than that shown in FIG. **8**.

[0079] FIG. **9** illustrates an illustrative distributed computing environment **900** capable of executing the software components described herein for providing a 3D environment data disambiguation tool. Thus, the distributed computing environment **900** illustrated in FIG. **9** can be used to provide the functionality described herein. The distributed computing environment **900** thus may be utilized to execute any aspects of the software components presented herein.

[0080] According to various implementations, the distributed computing environment **900** includes a computing environment **902** operating on, in communication with, or as part of the network **118**. The network **118** also can include various access networks. One or more client devices **906**A-**906**N (hereinafter referred to collectively and/or generically as "clients **906**") can communicate with the computing environment **902** via the network **118** and/or other connections (not illustrated in FIG. **9**). In the illustrated embodiment, the clients **906** include a computing device **906**A such as a laptop computer, a desktop computer, or other computing device; a slate or tablet computing device ("tablet computing device") **906**B; a mobile computing device **906**C such as a mobile telephone, a smart phone, or other mobile computing device; a server computer **906**D; and/or other devices **906**N. It should be understood that any number of clients **906** can communicate with the computing environment **902**. It should be understood that the illustrated clients **906** and computing architectures illustrated and described herein are illustrative, and should not be construed as being limited in any way.

[0081] In the illustrated embodiment, the computing environment **902** includes application servers **908**, data storage **910**, and one or more network interfaces **912**. According to various implementations, the functionality of the application servers **908** can be provided by one or more server computers that are executing as part of, or in communication with, the network **118**. The application servers **908** can host various services, virtual machines, portals, and/or other resources. In the illustrated embodiment, the application servers **908** host one or more virtual machines **914** for hosting applications or other functionality. According to various implementations, the virtual machines **914** host one or more applications and/or software modules for providing the functionality described herein for searching for providing a 3D environment data disambiguation tool. It should be understood that this embodiment is illustrative, and should not be construed as being limiting in any way. The application servers **908** also host or provide access to one or more Web portals, link pages, Web sites, and/or other information ("Web portals") **916**.

[0082] According to various implementations, the application servers **908** also include one or more mailbox services **918** and one or more messaging services **920**. The mailbox services **918** can include electronic mail ("email") services. The mailbox services **918** also can include various personal information management ("PIM") services including, but not limited to, calendar services, contact management services, collaboration services, and/or other services. The messaging services **920** can include, but are not limited to, instant messaging services, chat services, forum services, and/or other communication services.

[0083] The application servers **908** also can include one or more social networking services **922**. The social networking services **922** can include various social networking services including, but not limited to, services for sharing or posting status updates, instant messages, links, photos, videos, and/or other information; services for commenting or displaying interest in articles, products, blogs, or other resources; and/or other services. In some embodiments, the social networking services **922** are provided by or include the FACEBOOK social networking service, the LINKEDIN professional networking service, the MYSPACE social networking service, the FOURSQUARE geographic networking service, the YAMMER office colleague networking service, and the like. In other embodiments, the social networking services **922** are provided by other services, sites, and/or providers that may or may not explicitly be known as social networking providers. For example, some web sites allow users to interact with one another via email, chat services, and/or other means during various activities and/or contexts such as reading published articles, commenting on goods or services, publishing, collaboration, gaming, and the like. Examples of such services include, but are not limited to, the WINDOWS LIVE service and the XBOX LIVE service from Microsoft Corporation in Redmond, Wash. Other services are possible and are contemplated.

[0084] The social networking services **922** also can include commenting, blogging, and/or microblogging services. Examples of such services include, but are not limited to, the YELP commenting service, the KUDZU review service, the OFFICETALK enterprise microblogging service, the TWITTER messaging service, the GOOGLE BUZZ service, and/or other services. It should be appreciated that the above lists of services are not exhaustive and that numerous additional and/or alternative social networking services **922** are not mentioned herein for the sake of brevity. As such, the above embodiments are illustrative, and should not be construed as being limited in any way.

[0085] As shown in FIG. **9**, the application servers **908** also can host other services, applications, portals, and/or other resources ("other resources") **924**. The other resources **924** can include, but are not limited to, the productivity application **104**, the 3D data visualization component **114** and/or the data disambiguation component **116**. It thus can be appreciated that the computing environment **902** can provide integration of the concepts and technologies disclosed herein for a 3D environment data disambiguation tool with various mailbox, messaging, social networking, and/or other services or resources. For example, the concepts and technologies disclosed herein can integrated social networking information into various searches and provide those results to a user in conjunction with internal knowledge.

[0086] As mentioned above, the computing environment **902** can include the data storage **910**. According to various implementations, the functionality of the data storage **910** is provided by one or more data stores operating on, or in communication with, the network **118**. The functionality of the data storage **910** also can be provided by one or more server computers configured to host data for the computing environment **902**. The data storage **910** can include, host, or provide one or more real or virtual datastores **926**A-**926**N (hereinafter referred to collectively and/or generically as "datastores **926**"). The datastores **926** are configured to host data used or created by the application servers **908** and/or other data. Although not illustrated in FIG. **9**, the datastores **926** also can host or store data stores **224**A-**224**N in data store **224** shown in FIG. **2**.

[0087] The computing environment **902** can communicate with, or be accessed by, the network interfaces **912**. The network interfaces **912** can include various types of network hardware and software for supporting communications between two or more computing devices including, but not limited to, the clients **906** and the application servers **908**. It should be appreciated that the network interfaces **912** also may be utilized to connect to other types of networks and/or computer systems.

[0088] It should be understood that the distributed computing environment **900** described herein can provide any aspects of the software elements described herein with any number of virtual computing resources and/or other distributed computing functionality that can be configured to execute any aspects of the software components disclosed herein. According to various implementations of the concepts and technologies disclosed herein, the distributed computing environment **900** provides the software functionality described herein as a service to the clients **906**. It should be understood that the clients **906** can include real or virtual machines including, but not limited to, server computers, web servers, personal computers, mobile computing devices, smart phones, and/or other devices. As such, various embodiments of the concepts and technologies disclosed herein enable any device configured to access the distributed computing environment **900** to utilize the functionality described herein for providing a 3D environment data disambiguation tool.

[0089] Turning now to FIG. **10**, an illustrative computing device architecture **1000** for a computing device that is capable of executing various software components described herein for providing a 3D data environment disambiguation tool. The computing device architecture **1000** is applicable to computing devices that facilitate mobile computing due, in part, to form factor, wireless connectivity, and/or battery-powered operation. In some embodiments, the computing devices include, but are not limited to, mobile telephones, tablet devices, slate devices, portable video game devices, and the like. Moreover, the computing device architecture **1000** is applicable to any of the clients **906** shown in FIG. **9**. Furthermore, aspects of the computing device architecture **1000** may be applicable to traditional desktop computers, portable computers (e.g., laptops, notebooks, ultra-portables, and netbooks), server computers, and other computer systems, such as described herein with reference to FIG. **1**. For example, the single touch and multi-touch aspects disclosed herein below may be applied to desktop computers that utilize a touchscreen or some other touch-enabled device, such as a touch-enabled track pad or touch-enabled mouse.

[0090] The computing device architecture **1000** illustrated in FIG. **10** includes a processor **1002**, memory components **1004**, network connectivity components **1006**, sensor components **1008**, input/output ("I/O") components **1010**, and power components **1012**. In the illustrated embodiment, the processor **1002** is in communication with the memory components **1004**, the network connectivity components **1006**, the sensor components **1008**, the I/O components **1010**, and the power components **1012**. Although no connections are shown between the individuals components illustrated in FIG. **10**, the components can interact to carry out device functions. In some embodiments, the components are arranged so as to communicate via one or more busses (not shown).

[0091] The processor **1002** includes a central processing unit ("CPU") configured to process data, execute computer-executable instructions of one or more application programs, and communicate with other components of the computing device architecture **1000** in order to perform various functionality described herein. The processor **1002** may be utilized to execute aspects of the software components presented herein and, particularly, those that utilize, at least in part, a touch-enabled input.

[0092] In some embodiments, the processor **1002** includes a graphics processing unit ("GPU") configured to accelerate operations performed by the CPU, including, but not limited to, operations performed by executing general-purpose scientific and engineering computing applications, as well as graphics-intensive computing applications such as high resolution video (e.g., 720P, 1080P, and greater), video games, three-dimensional ("3D") modeling applications, and the like. In some embodiments, the processor **1002** is configured to communicate with a discrete GPU (not shown). In any case, the CPU and GPU may be configured in accordance with a co-processing CPU/GPU computing model, wherein the sequential part of an application executes on the CPU and the computationally-intensive part is accelerated by the GPU.

[0093] In some embodiments, the processor **1002** is, or is included in, a system-on-chip ("SoC") along with one or more of the other components described herein below. For example, the SoC may include the processor **1002**, a GPU, one or more of the network connectivity components **1006**, and one or more of the sensor components **1008**. In some embodiments, the processor **1002** is fabricated, in part, utilizing a package-on-package ("PoP") integrated circuit packaging technique. Moreover, the processor **1002** may be a single core or multi-core processor.

[0094] The processor **1002** may be created in accordance with an ARM architecture, available for license from ARM HOLDINGS of Cambridge, United Kingdom. Alternatively, the processor **1002** may be created in accordance with an x86 architecture, such as is available from INTEL CORPORATION of Mountain View, Calif. and others. In some embodiments, the processor **1002** is a SNAPDRAGON SoC, available from QUALCOMM of San Diego, Calif., a TEGRA SoC, available from NVIDIA of Santa Clara, Calif., a HUMMINGBIRD SoC, available from SAMSUNG of Seoul, South Korea, an Open Multimedia Application Platform ("OMAP") SoC, available from TEXAS INSTRUMENTS of Dallas, Tex., a customized version of any of the above SoCs, or a proprietary SoC.

[0095] The memory components **1004** include a random access memory ("RAM") **1014**, a read-only memory ("ROM") **1016**, an integrated storage memory ("integrated storage") **1018**, and a removable storage memory ("removable storage") **1020**. In some embodiments, the RAM **1014** or a portion thereof, the ROM **1016** or a portion thereof, and/or some combination the RAM **1014** and the ROM **1016** is integrated in the processor **1002**. In some embodiments, the ROM **1016** is configured to store a firmware, an operating system or a portion thereof (e.g., operating system kernel), and/or a bootloader to load an operating system kernel from the integrated storage **1018** or the removable storage **1020**.

[0096] The integrated storage **1018** can include a solid-state memory, a hard disk, or a combination of solid-state memory and a hard disk. The integrated storage **1018** may be soldered or otherwise connected to a logic board upon which the processor **1002** and other components described herein also may be connected. As such, the integrated storage **1018** is integrated in the computing device. The integrated storage **1018** is configured to store an operating system or portions thereof, application programs, data, and other software components described herein.

[0097] The removable storage **1020** can include a solid-state memory, a hard disk, or a combination of solid-state memory and a hard disk. In some embodiments, the removable storage **1020** is provided in lieu of the integrated storage **1018**. In other embodiments, the removable storage **1020** is provided as additional optional storage. In some embodiments, the removable storage **1020** is logically combined with the integrated storage **1018** such that the total available storage is made available and shown to a user as a total combined capacity of the integrated storage **1018** and the removable storage **1020**.

[0098] The removable storage **1020** is configured to be inserted into a removable storage memory slot (not shown) or other mechanism by which the removable storage **1020** is inserted and secured to facilitate a connection over which the removable storage **1020** can communicate with other components of the computing device, such as the processor **1002**. The removable storage **1020** may be embodied in various memory card formats including, but not limited to, PC card, CompactFlash card, memory stick, secure digital ("SD"), miniSD, microSD, universal integrated circuit card ("UICC") (e.g., a subscriber identity module ("SIM") or universal SIM ("USIM")), a proprietary format, or the like.

[0099] It can be understood that one or more of the memory components **1004** can store an operating system. According to various embodiments, the operating system includes, but is not limited to, SYMBIAN OS from SYMBIAN LIMITED, WINDOWS MOBILE OS from Microsoft Corporation of Redmond, Wash., WINDOWS PHONE OS from Microsoft Corporation, WINDOWS from Microsoft Corporation, PALM WEBOS from Hewlett-Packard Company of Palo Alto, Calif., BLACKBERRY OS from Research In Motion Limited of Waterloo, Ontario, Canada, IOS from Apple Inc. of Cupertino, Calif., and ANDROID OS from Google Inc. of Mountain View, Calif. Other operating systems are contemplated.

[0100] The network connectivity components **1006** include a wireless wide area network component ("WWAN component") **1022**, a wireless local area network component ("WLAN component") **1024**, and a wireless personal area network component ("WPAN component") **1026**. The network connectivity components **1006** facilitate communications to and from the network **118**, which may be a WWAN, a WLAN, or a WPAN. Although a single network **118** is illustrated, the network connectivity components **1006** may facilitate simultaneous communication with multiple networks. For example, the network connectivity components **1006** may facilitate simultaneous communications with multiple networks via one or more of a WWAN, a WLAN, or a WPAN.

[0101] The network **118** may be a WWAN, such as a mobile telecommunications network utilizing one or more mobile telecommunications technologies to provide voice and/or data services to a computing device utilizing the computing device architecture **1000** via the WWAN component **1022**. The mobile telecommunications technologies can include, but are not limited to, Global System for Mobile communications ("GSM"), Code Division Multiple Access ("CDMA") ONE, CDMA2000, Universal Mobile Telecommunications System ("UMTS"), Long Term Evolution ("LTE"), and Worldwide Interoperability for Microwave Access ("WiMAX"). Moreover, the network **118** may utilize various channel access methods (which may or may not be used by the aforementioned standards) including, but not limited to, Time Division Multiple Access ("TDMA"), Frequency Division Multiple Access ("FDMA"), CDMA, wideband CDMA ("W-CDMA"), Orthogonal Frequency Division Multiplexing ("OFDM"), Space Division Multiple Access ("SDMA"), and the like. Data communications may be provided using General Packet Radio Service ("GPRS"), Enhanced Data rates for Global Evolution ("EDGE"), the High-Speed Packet Access ("HSPA") protocol family including High-Speed Downlink Packet Access ("HSDPA"), Enhanced Uplink ("EUL") or otherwise termed High-Speed Uplink Packet Access ("HSUPA"), Evolved HSPA ("HSPA+"), LTE, and various other current and future wireless data access standards. The network **118** may be configured to provide voice and/or data communications with any combination of the above technologies. The network **118** may be configured to or adapted to provide voice and/or data communications in accordance with future generation technologies.

[0102] In some embodiments, the WWAN component **1022** is configured to provide dual-multi-mode connectivity to the network **118**. For example, the WWAN component **1022** may be configured to provide connectivity to the network **118**, wherein the network **118** provides service via GSM and UMTS technologies, or via some other combination of technologies. Alternatively, multiple WWAN components **1022** may be utilized to perform such functionality, and/or provide additional functionality to support other non-compatible technologies (i.e., incapable of being supported by a single WWAN component). The WWAN component **1022** may facilitate similar connectivity to multiple networks (e.g., a UMTS network and an LTE network).

[0103] The network **118** may be a WLAN operating in accordance with one or more Institute of Electrical and Electronic Engineers ("IEEE") 802.11 standards, such as IEEE 802.11a, 802.11b, 802.11g, 802.11n, and/or future 802.11 standard (referred to herein collectively as WI-FI). Draft 802.11 standards are also contemplated. In some embodiments, the WLAN is implemented utilizing one or more wireless WI-FI access points. In some embodiments, one or more of the wireless WI-FI access points are another computing device with connectivity to a WWAN that are functioning as a WI-FI hotspot. The WLAN component **1024** is configured to connect to the network **118** via the WI-FI access points. Such connections may be secured via various encryption technologies including, but not limited, WI-FI Protected Access ("WPA"), WPA2, Wired Equivalent Privacy ("WEP"), and the like.

[0104] The network **118** may be a WPAN operating in accordance with Infrared Data Association ("IrDA"), BLUETOOTH, wireless Universal Serial Bus ("USB"), Z-Wave, ZIGBEE, or some other short-range wireless technology. In some embodiments, the WPAN component **1026** is configured to facilitate communications with other devices, such as peripherals, computers, or other computing devices via the WPAN.

[0105] The sensor components **1008** include a magnetometer **1028**, an ambient light sensor **1030**, a proximity sensor **1032**, an accelerometer **1034**, a gyroscope **1036**, and a Global Positioning System sensor ("GPS sensor") **1038**. It is contemplated that other sensors, such as, but not limited to, temperature sensors or shock detection sensors, also may be incorporated in the computing device architecture **1000**.

[0106] The magnetometer **1028** is configured to measure the strength and direction of a magnetic field. In some embodiments the magnetometer **1028** provides measurements to a compass application program stored within one of the memory components **1004** in order to provide a user with accurate directions in a frame of reference including the cardinal directions, north, south, east, and west. Similar measurements may be provided to a navigation application program that includes a compass component. Other uses of measurements obtained by the magnetometer **1028** are contemplated.

[0107] The ambient light sensor **1030** is configured to measure ambient light. In some embodiments, the ambient light sensor **1030** provides measurements to an application program stored within one the memory components **1004** in order to automatically adjust the brightness of a display (described below) to compensate for low-light and high-light environments. Other uses of measurements obtained by the ambient light sensor **1030** are contemplated.

[0108] The proximity sensor **1032** is configured to detect the presence of an object or thing in proximity to the computing device without direct contact. In some embodiments, the proximity sensor **1032** detects the presence of a user's body (e.g., the user's face) and provides this information to an application program stored within one of the memory components **1004** that utilizes the proximity information to enable or disable some functionality of the computing device. For example, a telephone application program may automatically disable a touchscreen (described below) in response to receiving the proximity information so that the user's face does not inadvertently end a call or enable/disable other functionality within the telephone application program during the call. Other uses of proximity as detected by the proximity sensor **1032** are contemplated.

[0109] The accelerometer **1034** is configured to measure proper acceleration. In some embodiments, output from the accelerometer **1034** is used by an application program as an input mechanism to control some functionality of the application program. For example, the application program may be a video game in which a character, a portion thereof, or an object is moved or otherwise manipulated in response to input received via the accelerometer **1034**. In some embodiments, output from the accelerometer **1034** is provided to an application program for use in switching between landscape and portrait modes, calculating coordinate acceleration, or detecting a fall. Other uses of the accelerometer **1034** are contemplated.

[0110] The gyroscope **1036** is configured to measure and maintain orientation. In some embodiments, output from the gyroscope **1036** is used by an application program as an input mechanism to control some functionality of the application program. For example, the gyroscope **1036** can be used for accurate recognition of movement within a 3D data environment of a video game application or some other application. In some embodiments, an application program utilizes output from the gyroscope **1036** and the accelerometer **1034** to enhance control of some functionality of the application program. Other uses of the gyroscope **1036** are contemplated.

[0111] The GPS sensor **1038** is configured to receive signals from GPS satellites for use in calculating a location. The location calculated by the GPS sensor **1038** may be used by any application program that requires or benefits from location information. For example, the location calculated by the GPS sensor **1038** may be used with a navigation application program to provide directions from the location to a destination or directions from the destination to the location. Moreover, the GPS sensor **1038** may be used to provide location information to an external location-based service, such as E911 service. The GPS sensor **1038** may obtain location information generated via WI-FI, WIMAX, and/or cellular triangulation techniques utilizing one or more of the network connectivity components **1006** to aid the GPS sensor **1038** in obtaining a location fix. The GPS sensor **1038** may also be used in Assisted GPS ("A-GPS") systems.

[0112] The I/O components **1010** include a display **1040**, a touchscreen **1042**, a data I/O interface component ("data I/O") **1044**, an audio I/O interface component ("audio I/O") **1046**, a video I/O interface component ("video I/O") **1048**, and a camera **1050**. In some embodiments, the display **1040** and the touchscreen **1042** are combined. In some embodiments two or more of the data I/O component **1044**, the audio I/O interface component **1046**, and the video I/O component **1048** are combined. The I/O components **1010** may include discrete processors configured to support the various interface described below, or may include processing functionality built-in to the processor **1002**.

[0113] The display **1040** is an output device configured to present information in a visual form. In particular, the display **1040** may present graphical user interface ("GUI") elements, text, images, video, notifications, virtual buttons, virtual keyboards, messaging data, Internet content, device status, time, date, calendar data, preferences, map information, location information, and any other information that is capable of being presented in a visual form. In some embodiments, the display **1040** is a liquid crystal display ("LCD") utilizing any active or passive matrix technology and any backlighting technology (if used). In some embodiments, the display **1040** is an organic light emitting diode ("OLED") display. Other display types are contemplated.

[0114] The touchscreen **1042** is an input device configured to detect the presence and location of a touch. The touchscreen **1042** may be a resistive touchscreen, a capacitive touchscreen, a surface acoustic wave touchscreen, an infrared touchscreen, an optical imaging touchscreen, a dispersive signal touchscreen, an acoustic pulse recognition touchscreen, or may utilize any other touchscreen technology. In some embodiments, the touchscreen **1042** is incorporated on top of the display **1040** as a transparent layer to enable a user to use one or more touches to interact with objects or other information presented on the display **1040**. In other embodiments, the touchscreen **1042** is a touch pad incorporated on a surface of the computing device that does not include the display **1040**. For example, the computing device may have a touchscreen incorporated on top of the display **1040** and a touch pad on a surface opposite the display **1040**.

[0115] In some embodiments, the touchscreen **1042** is a single-touch touchscreen. In other embodiments, the touchscreen **1042** is a multi-touch touchscreen. In some embodiments, the touchscreen **1042** is configured to detect discrete touches, single touch gestures, and/or multi-touch gestures. These are collectively referred to herein as gestures for convenience. Several gestures will now be described. It should be understood that these gestures are illustrative and are not intended to limit the scope of the appended claims. Moreover, the described gestures, additional gestures, and/or alternative gestures may be implemented in software for use with the touchscreen **1042**. As such, a developer may create gestures that are specific to a particular application program.

[0116] In some embodiments, the touchscreen **1042** supports a tap gesture in which a user taps the touchscreen **1042** once on an item presented on the display **1040**. The tap gesture may be used for various reasons including, but not limited to, opening or launching whatever the user taps. In some embodiments, the touchscreen **1042** supports a double tap gesture in which a user taps the touchscreen **1042** twice on an item presented on the display **1040**. The double tap gesture may be used for various reasons including, but not limited to, zooming in or zooming out in stages. In some embodiments, the touchscreen **1042** supports a tap and hold gesture in which a user taps the touchscreen **1042** and maintains contact for at least a pre-defined time. The tap and hold gesture may be used for various reasons including, but not limited to, opening a context-specific menu.

[0117] In some embodiments, the touchscreen **1042** supports a pan gesture in which a user places a finger on the touchscreen **1042** and maintains contact with the touchscreen **1042** while moving the finger on the touchscreen **1042**. The pan gesture may be used for various reasons including, but not limited to, moving through screens, images, or menus at a controlled rate. Multiple finger pan gestures are also contemplated. In some embodiments, the touchscreen **1042** supports a flick gesture in which a user swipes a finger in the direction the user wants the screen to move. The flick gesture may be used for various reasons including, but not limited to, scrolling horizontally or vertically through menus or pages. In some embodiments, the touchscreen **1042** supports a pinch and stretch gesture in which a user makes a pinching motion with two fingers (e.g., thumb and forefinger) on the touchscreen **1042** or moves the two fingers apart. The pinch and stretch gesture may be used for various reasons including, but not limited to, zooming gradually in or out of a website, map, or picture.

[0118] Although the above gestures have been described with reference to the use one or more fingers for performing the gestures, other appendages such as toes or objects such as styluses may be used to interact with the touchscreen **1042**. As such, the above gestures should be understood as being illustrative and should not be construed as being limiting in any way.

[0119] The data I/O interface component **1044** is configured to facilitate input of data to the computing device and output of data from the computing device. In some embodiments, the data I/O interface component **1044** includes a connector configured to provide wired connectivity between the computing device and a computer system, for example, for synchronization operation purposes. The connector may be a proprietary connector or a standardized connector such as USB, micro-USB, mini-USB, or the like. In some embodiments, the connector is a dock connector for docking the computing device with another device such as a docking station, audio device (e.g., a digital music player), or video device.

[0120] The audio I/O interface component **1046** is configured to provide audio input and/or output capabilities to the computing device. In some embodiments, the audio I/O interface component **1044** includes a microphone configured to collect audio signals. In some embodiments, the audio I/O interface component **1044** includes a headphone jack configured to provide connectivity for headphones or other external speakers. In some embodiments, the audio I/O interface component **1046** includes a speaker for the output of audio signals. In some embodiments, the audio I/O interface component **1044** includes an optical audio cable out.

[0121] The video I/O interface component **1048** is configured to provide video input and/or output capabilities to the computing device. In some embodiments, the video I/O interface component **1048** includes a video connector configured to receive video as input from another device (e.g., a video media player such as a DVD or BLURAY player) or send video as output to another device (e.g., a monitor, a television, or some other external display). In some embodiments, the video I/O interface component **1048** includes a High-Definition Multimedia Interface ("HDMI"), mini-HDMI, micro-HDMI, DisplayPort, or proprietary connector to input/output video content. In some embodiments, the video I/O interface component **1048** or portions thereof is combined with the audio I/O interface component **1046** or portions thereof.

[0122] The camera **1050** can be configured to capture still images and/or video. The camera **1050** may utilize a charge coupled device ("CCD") or a complementary metal oxide semiconductor ("CMOS") image sensor to capture images. In some embodiments, the camera **1050** includes a flash to aid in taking pictures in low-light environments. Settings for the camera **1050** may be implemented as hardware or software buttons.

[0123] Although not illustrated, one or more hardware buttons may also be included in the computing device architecture **1000**. The hardware buttons may be used for controlling some operational aspect of the computing device. The hardware buttons may be dedicated buttons or multi-use buttons. The hardware buttons may be mechanical or sensor-based.

[0124] The illustrated power components **1012** include one or more batteries **1052**, which can be connected to a battery gauge **1054**. The batteries **1052** may be rechargeable or disposable. Rechargeable battery types include, but are not limited to, lithium polymer, lithium ion, nickel cadmium, and nickel metal hydride. Each of the batteries **1052** may be made of one or more cells.

[0125] The battery gauge **1054** can be configured to measure battery parameters such as current, voltage, and temperature. In some embodiments, the battery gauge **1054** is configured to measure the effect of a battery's discharge rate, temperature, age and other factors to predict remaining life within a certain percentage of error. In some embodiments, the battery gauge **1054** provides measurements to an application program that is configured to utilize the measurements to present useful power management data to a user. Power management data may include one or more of a percentage of battery used, a percentage of battery remaining, a battery condition, a remaining time, a remaining capacity (e.g., in watt hours), a current draw, and a voltage.

[0126] The power components **1012** may also include a power connector, which may be combined with one or more of the aforementioned I/O components **1010**. The power components **1012** may interface with an external power system or charging equipment via a power I/O component (not illustrated).

[0127] Based on the foregoing, it should be appreciated that concepts and technologies for providing a 3D data environment disambiguation tool have been disclosed herein. Although the subject matter presented herein has been described in language specific to computer structural features, methodological and transformative acts, specific computing machinery, and computer readable media, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features, acts, or media described herein. Rather, the specific features, acts and mediums are disclosed as example forms of implementing the claims.

[0128] The subject matter described above is provided by way of illustration only and should not be construed as limiting. Various modifications and changes may be made to the subject matter described herein without following the example embodiments and applications illustrated and described, and without departing from the true spirit and scope of the present invention, which is set forth in the following claims.

What is claimed is:

1. A computer, comprising:

a processor; and

a computer-readable storage medium in communication with the processor, the computer-readable storage medium comprising computer-executable instructions stored thereupon that, when executed by the processor, cause the processor to

detect a selection of selected data to be rendered in a 3D data environment,

determine if a data point in the selected data is ambiguous,

if the data point in the selected data is determined not to be ambiguous, render the selected data in the 3D data environment, and if the data point in the selected data is determined to be ambiguous,

display an ambiguity interface,

receive an input to resolve the ambiguity, and

render the selected data in the 3D data environment using the input to resolve the ambiguity.

2. The computer of claim **1**, wherein the computer-readable storage medium further comprises computer-executable instructions that, when executed by the processor, cause the processor to determine a confidence level if the data point in the selected data is determined to be ambiguous.

3. The computer of claim **2**, wherein the ambiguity interface further comprises a confidence interface for displaying the confidence level.

4. The computer of claim **1**, wherein the ambiguity interface comprises an ambiguity presentation interface to receive the input to resolve the ambiguity.

5. The computer of claim **4**, wherein the ambiguity presentation interface comprises a plurality of selection boxes to receive the input to resolve the ambiguity.

6. The computer of claim **1**, wherein the ambiguity interface comprises an ambiguity resolution interface for receiving a correction input to correct the data point determined to be ambiguous.

7. The computer of claim **6**, wherein the ambiguity interface further comprises an ambiguous data display input for displaying an interface highlighting the data point determined to be ambiguous.

8. The computer of claim **1**, wherein the ambiguity interface comprises an ambiguity alternative interface displaying one or more alternates to the data point determined to be ambiguous.

9. The computer of claim **1**, wherein the computer-readable storage medium further comprises computer-executable instructions that, when executed by the processor, cause the processor to display an expected range box for receiving at least one range to resolve the ambiguity or an expected value box for receiving a value to resolve the ambiguity.

10. A method for resolving data disambiguation, the method comprising:

detecting, at a computer executing a data disambiguation component, a selection of selected data to be rendered in a 3D data environment,

determining, at the computer, if a data point in the selected data is ambiguous,

if the data point in the selected data is determined to be ambiguous, rendering, by the computer the selected data in the 3D data environment,

if the data point in the selected data is determined not to be ambiguous, determining, by the computer if the data point is within an expected range or is an expected value,

if the data point is within the expected range or is the expected value, rendering the selected data in the 3D data environment, and

if the data point is not within the expected range or is not the expected value,

displaying an ambiguity interface,

receiving an input to resolve the ambiguity, and

rendering the selected data in the 3D data environment using the input to resolve the ambiguity.

11. The method of claim **10**, further comprising determining a confidence level if the data point in the selected data is determined to be ambiguous.

12. The method of claim **11**, further comprising displaying a confidence interface for displaying the confidence level.

13. The method of claim **10**, wherein displaying the ambiguity interface comprises displaying an ambiguity presentation interface to receive the input to resolve the ambiguity.

14. The method of claim **10**, wherein determining, at the computer, if a data point in the selected data is ambiguous comprises comparing the data point to a previously disambiguated data point.

15. The method of claim **10**, wherein displaying the ambiguity interface comprises displaying an ambiguity resolution interface for receiving a correction input to correct the data point determined to be ambiguous.

16. The method of claim **10**, wherein displaying the ambiguity interface comprises displaying an ambiguous data display input comprising an interface to highlight the data point determined to be ambiguous.

17. The method of claim **10**, wherein displaying the ambiguity interface comprises displaying an ambiguity alternative interface comprising one or more alternatives to the data point determined to be ambiguous.

18. A computer-readable storage medium having computer-executable instructions stored thereon that, when executed by the processor, cause the processor to:

detect a selection of selected data to be rendered in a 3D data environment,

determine if a data point in the selected data is ambiguous,

if the data point in the selected data is determined not to be ambiguous, render the selected data in the 3D data environment,

if the data point in the selected data is determined to be ambiguous, determine if the data point is within an expected range or is an expected value,

if the data point is within the expected range or is the expected value, render the selected data in the 3D data environment, and

if the data point is not within the expected range or is not the expected value, display an ambiguity interface comprising an ambiguity presentation interface for receiving an input to resolve the ambiguity or an ambiguity resolution interface for receiving a correction input to correct the data point.

19. The computer-readable storage medium of claim **18**, wherein the computer-readable storage medium further comprises computer-executable instructions that, when executed by the processor, cause the processor to display a confidence interface comprising a confidence level of the data point determined to be ambiguous.

20. The computer-readable storage medium of claim **18**, wherein the computer-readable storage medium further comprises computer-executable instructions that, when executed by the processor, cause the processor to display an ambiguity alternative interface comprising one or more alternates to the data point determined to be ambiguous.

* * * * *