

METHOD AND APPARATUS FOR DETERMINING INDEX GRIDS OF GEO-FENCE

5 **Technical Field**

The present application relates to the technical field of data processing, and in particular to a method and apparatus for determining index grids of a geo-fence.

Technical Background

10 With the popularity of mobile Internet services and applications such as Online Map, O2O (Online To Offline), more and more user location data is collected and stored. Geo-fence-based services, such as marketing push and security alert, are the most direct, most real-time and most effective explorations and use of user location data, and have broad application prospects.

15 A geo-fence is a geographical area enclosed by a virtual fence, and a device may receive push notifications or warnings when the device enters or leaves the area or moves in the area. Grid indexing technology is the foundation of geo-fence-based services. Gridding is a way of dividing two-dimensional geospace, and each grid covers a certain geographical area. The geo-fence may be approximated as a grid set. A range covered by grids in the set is used to
20 simulate the area contained in the geo-fence, which is used for fast retrieval and positioning of the geo-fence. The grids in the grid set are the index grids of the geo-fence.

In the prior art, a geo-fence is usually represented by a plurality of grids of the same granularity, and these index grids should completely cover the area contained in the geo-fence. In the case of the index grids of a coarser granularity, it is difficult to accurately
25 represent the geo-fence, which may cause geo-fence based services to additionally disturb

users who are not in the geo-fence. To avoid this problem, the precision of the index grids is usually required to exceed a certain threshold, which often means finer-grained index grids and a sharp increase in the number of indexes. Indexes in a large data volume usually need to be stored in a distributed manner on multiple devices. The larger the data volume, the more complicated the management and maintenance of the indexes, and the slower the query speed, the lower the stability and response speed of a geo-fence-based service.

Summary of the Invention

In view of this, the present application provides a method for determining index grids of a geo-fence, including:

determining a circumscribed grid of a geo-fence, wherein the circumscribed grid is a smallest single grid containing the geo-fence;

with the circumscribed grid as a starting point, subdividing grid(s) containing the boundary of the geo-fence into finer-grained grids level by level until index accuracy of a subdivided grid set is not lower than a predetermined precision threshold, wherein the grid set is composed of grids within the geo-fence and the grids containing the boundary of the geo-fence, and the index accuracy is a ratio of the area of the geo-fence to a sum of areas of all the grids in the grid set; and

taking the grids in the grid set as index grids of the geo-fence.

The present application further provides an apparatus for determining index grids of a geo-fence, including:

a circumscribed grid unit configured to determine a circumscribed grid of a geo-fence, wherein the circumscribed grid is a smallest single grid containing the geo-fence;

a grid subdividing unit configured to, with the circumscribed grid as a starting point, subdivide grid(s) containing the boundary of the geo-fence into finer-grained grids level by

level until index accuracy of a subdivided grid set is not lower than a predetermined precision threshold, wherein the grid set is composed of grids within the geo-fence and the grids containing the boundary of the geo-fence, and the index accuracy is a ratio of the area of the geo-fence to a sum of areas of all the grids in the grid set; and

5 an index determining unit configured to take the grids in the grid set as index grids of the geo-fence.

A computer device provided by the present application includes a memory and a processor, wherein the memory stores a computer program executable by the processor; and when the processor runs the computer program, the steps of the above-mentioned method
10 for determining index grids of a geo-fence are executed.

A computer-readable storage medium provided by the present application stores a computer program, and when the computer program is executed by a processor, the steps of the above-mentioned method for determining index grids of a geo-fence are executed.

It can be seen from the above technical solutions that, in embodiments of the present
15 application, with the circumscribed grid of the geo-fence as a starting point, the grid containing the boundary of the geo-fence is subdivided into finer-grained grids level by level until a predetermined precision threshold is met; since the grids within the geo-fence are not subdivided, the grids within the geo-fence may adopt a coarser granularity. Under the premise of fully containing the geo-fence and meeting the precision requirement, the number
20 of index grids is greatly decreased, index storage is less complex, index query speed is increased, and the stability and response speed of a geo-fence-based service are improved.

Brief Description of the Drawings

FIG. 1 is a flowchart of a method for determining index grids of a geo-fence according
25 to some embodiments of the present application;

FIG. 2 is a flowchart of an example of a grid subdivision iteration process according to some embodiments of the present application;

FIG. 3 is a flowchart of a method for determining index grids of a geo-fence according to some application examples of the present application;

5 FIG. 4 is a schematic diagram of a boundary of a Geof-Demo and a circumscribed grid of the boundary according to some application examples of the present application;

FIG. 5 is a schematic diagram of all grids in a grid set at the end of grid subdivision of the Geof-Demo according to some application examples of the present application;

10 FIG. 6 is a schematic diagram of a grid merging mode of the Geof-Demo according to some application examples of the present application;

FIG. 7 is a schematic diagram of a subtree of the item tree hierarchy of the Geof-Demo according to some application examples of the present application;

FIG. 8 is a schematic diagram of index grids of the Geof-Demo according to some application examples of the present application;

15 FIG. 9 is a hardware structural diagram of a device running embodiments of the present application; and

FIG. 10 is a logical structural diagram of an apparatus for determining index grids of a geo-fence according to some embodiments of the present application.

20 **Detailed Description**

The grid indexes of a geo-fence may be measured based on three indicators: index coverage, index accuracy, and number of index grids. Assuming that the number of set elements in a grid set used to represent a geo-fence is N (N is a natural number), that is, the number of index grids of the geo-fence is N , then the index coverage C of the grid set may
25 be expressed as Equation 1, and the index accuracy P may be expressed as Equation 2:

$$C = \frac{SO}{SF} \quad \text{Equation 1}$$

$$P = \frac{SO}{SI} \quad \text{Equation 2}$$

In Equations 1 and 2, SO is a sum of areas of the portions of all index grids that overlap with the geo-fence, i.e., $SO = \sum_{i=1}^N aoo_i$, where aoo_i is the area of the portion of the
 5 i-th index grid that overlaps with the geo-fence, and $i \in [1, N]$; SF is the area of the geo-fence; SI is the sum of the areas of all the index grids, i.e., $SI = \sum_{i=1}^N aog_i$, where aog_i is the area of the i-th index grid, and $i \in [1, N]$.

The technical problem to be solved in the embodiments of the present application may be expressed as follows: under the condition that the grid set completely covers the geo-fence (i.e., $C=1$), and the index accuracy rate is not lower than a predetermined precision
 10 threshold THR^P (i.e., $P \geq THR^P$), the value of N is reduced.

When a geo-fence is represented as a grid set of index grids, the index grids in the set may be divided into two categories, one category including grids within the geo-fence, that is, grids with coverage completely falling within the geo-fence, the other category including
 15 grids containing the boundary of the geo-fence, that is, grids with coverage both falling within the geo-fence and going beyond the geo-fence. For grids within the geo-fence, regardless of the granularities of those grids, the index accuracy of the grid set is not affected. The index accuracy of a grid set is determined by the granularity of each grid that contains the boundary of the geo-fence. The index accuracy can be improved by increasing the
 20 granularity of the grids containing the boundary of the geo-fence (increasing the granularity means using denser grids or finer grids), and reduce the number N of the index grids by allowing the grids within the geo-fence to use a coarser granularity.

Therefore, embodiments of the present application provide a new method for determining index grids of a geo-fence. According to the method, a smallest single grid containing the geo-fence is considered as a circumscribed grid of the geo-fence, and starting from the circumscribed grid, iteration is performed to subdivide grid(s) containing the boundary of the geo-fence into finer-grained grids level by level and the iteration stops when the index accuracy of the grid set meets a predetermined precision threshold; by only increasing the granularity of the grids containing the boundary of the geo-fence but not further subdividing the grids within the geo-fence, and determining grids of different granularities as index grids, the number of index grids is reduced, thereby solving the problems in the prior art.

Embodiments of the present application may run on any computing and storage device, such as a mobile phone, a tablet computer, a PC (Personal Computer), a notebook computer, and a server; and functions in the embodiments of the present application may also be implemented by logical nodes running on two or more devices.

In some embodiments of the present application, the flow of the method for determining index grids of a geo-fence is as shown in FIG. 1.

In step 110, a circumscribed grid of the geo-fence is determined.

A grid system is a spatial data structure based on the regular hierarchical subdivision of plane subsets. Based on a certain granularity system, a geographic region may be subdivided, level by level, into several mutually adjacent polygon grids from coarse to fine, so as to facilitate the expression of position information and the management of position-related data, and make the organization, analysis and application of data highly efficient.

The grid system usually uses a certain granularity system to divide geographic regions. Each granularity system consists of a series of granularities arranged level by level from coarse to fine, and the granularities used in grid subdivision determine the sizes of the grids.

For a geographical region divided with a defined granularity system, a grid of T (T is a natural number)-level granularity consists of a number of grids of $(T+1)$ -level granularity and the grids of $(T+1)$ -level granularity are equal in area.

For a geo-fence, the smallest single grid containing the geo-fence is a circumscribed grid of the geo-fence. In other words, if in the granularity system adopted, a grid of a certain granularity may cover the entire area of the geo-fence, and after the grid is subdivided into grids of a lower-level granularity, each finer-grained grid may only cover a part of the geo-fence, not the entire area of the geo-fence, and then the grid is the circumscribed grid of the geo-fence.

The specific manner of determining the circumscribed grid of a geo-fence may be implemented by referring to the prior art, and its details are not described herein.

In step 120, with the circumscribed grid as a starting point, grid(s) containing the boundary of the geo-fence are subdivided into finer-grained grids level by level until index accuracy of a subdivided grid set is not lower than a predetermined precision threshold, wherein the grid set is composed of grids within the geo-fence and grids containing the boundary of the geo-fence, and the index accuracy is a ratio of the area of the geo-fence to a sum of areas of all the grids in the grid set.

After the circumscribed grid is determined, iteration is started with the circumscribed grid as an index grid of the geo-fence, and in this case, only one element, i.e., the circumscribed grid, is included in the grid set. The index accuracy of the grid set relative to the geo-fence is calculated. If the calculated index accuracy is lower than a predetermined precision threshold, grids containing the boundary of the geo-fence in the grid set are subdivided into finer-grained grids, and grids within the geo-fence in the grid set are not processed; if a finer-grained grid obtained after the grid subdivision is completely outside the geo-fence, the grid outside the geo-fence is deleted from the grid set. The above process

is repeated until the index accuracy of the grid set relative to the geo-fence is not lower than the predetermined precision threshold.

Since the circumscribed grid covers the entire area of the geo-fence, and during the iteration, only the grids that are completely outside the geo-fence are deleted from the grid set, all grids in the grid set may always cover the entire area of the geo-fence, and the index coverage of the grid set is 1.

FIG. 2 shows an example of an iteration process flow of the step, including the following specific steps.

In step 201, a grid set is generated, and elements in the grid set are grids within the geo-fence and grids containing the boundary of the geo-fence. In this case, only one element, i.e., the circumscribed grid, is included in the grid set.

In step 202, with all the grids in the grid set as index grids of the geo-fence, the index accuracy of the grid set is calculated according to Equation 2. The calculated index accuracy is compared with the predetermined precision threshold; if the calculated index accuracy is not lower than the predetermined precision threshold, the grid set meets the predetermined precision requirement, and then the process proceeds to step 210; otherwise, step 203 is performed.

In step 203, a first grid in the grid set is determined as the current grid.

In step 204, a determination whether the current grid is completely within the geo-fence is made; if the determination result is yes, the process proceeds to step 208; if the determination result is no, the current grid is a grid containing the boundary of the geo-fence, and step 205 is performed.

In step 205, the current grid is subdivided into lower-level finer-grained grids. The specific number of lower-level finer-grained grids the current grid is subdivided into is determined by the granularity system used in a specific application scenario.

In step 206, in the grid set, the current grid is replaced with the lower-level finer-grained grids obtained after the grid subdivision. That is, the current grid is deleted from the grid set, and a number of lower-level finer-grained grids obtained by subdividing the current grid are added to the grid set. In the lower-level finer-grained grids obtained after the grid subdivision, there may be grids containing the boundary of the geo-fence, may be grids within the geo-fence and may be grids outside the geo-fence.

In step 207, if, in a number of lower-level finer-grained grids obtained after grid subdivision, there are grids outside the geo-fence, and since the grids outside the geo-fence will not be index grids of the geo-fence and should not be used as an element of the grid set, the lower-level grids outside the geo-fence after the grid subdivision are deleted from the grid set. If there are no grids outside the geo-fence in the number of lower-level finer-grained grids obtained after grid subdivision, then the deletion operation is not necessary.

In step 208, a determination whether there is a next grid in the grid set is made. If the determination result is yes, step 209 is performed to continue the current iteration. If there is no next grid in the grid set, the current iteration is completed. All grids containing the boundary of the geo-fence in the grid set are already finer-grained grids, and the process proceeds to step 202 to calculate the index accuracy of the grid set to determine whether the next iteration is necessary.

In step 209, the next grid in the grid set is determined as the current grid, and the process proceeds to step 204.

In step 210, the grid set is output, and the iteration process is completed.

It should be noted that the process in FIG. 2 is only an example of an iteration process flow, and the iteration process may be implemented in other processes according to the specific situation of an actual application scenario, which will not be limited in embodiments of the present application.

Returning back to FIG. 1, in step 130, the grids in the grid set are determined as index grids of the geo-fence.

After step 120, all the grids containing the boundary of the geo-fence in the grid set have the same granularity which is the finest (most dense) granularity (hereinafter referred to as “lower-limit granularity”) of all grid granularities in the grid set. The grids within the geo-fence in the grid set may have different granularities, and the possible range of their granularities is from the lower-level granularity of the granularity of the circumscribed grid to the lower-limit granularity.

Since the index accuracy of the grid set depends on the granularity of the grids containing the boundary of the geo-fence in the set, it is allowed that the grids within the geo-fence have a larger granularity, so as to reduce the number of index grids while meeting the precision requirement. In particular, in the iteration process in the step 120, according to the actual shape of the geo-fence, coarser-grained grids are used as much as possible to cover the area within the geo-fence, which results in a very good effect in reducing the index grids.

The specific processing method where the grids in the grid set are determined as the index grids of the geo-fence typically includes registering an entity corresponding to the geo-fence under an index item of each grid in the grid set. There may be other methods adopted according to the needs of actual application scenarios, which will not be limited herein.

In most cases, for the grid set obtained by the step 120, its index accuracy may exceed a predetermined precision threshold. When the index accuracy of the grid set obtained by the step 120 is higher than the predetermined precision threshold, part of grids in the grid set may be merged into coarser-grained grids containing the boundary of the geo-fence, wherein the merging is performed under the condition that the index accuracy after merging is not lower than the predetermined precision threshold; and then in the step 130, the grids in the

grid set after merging may be determined as the index grids of the geo-fence. This may further reduce the number of index grids of the geo-fence.

It should be noted that the grid merging refers to replacing a number of finer-grained grids belonging to the same coarser-grained grid with the coarser-grained grid instead of
5 taking a union set of the finer-grained grids. The finer-grained grids belonging to the same coarser-grained grid may not all be elements in the grid set. In this case, one or more finer-grained grids in the grid set may also be merged into the coarser-grained grid.

For example, two or more grids subdivided from the same grid and having the lower-limit granularity may be found in the grid set, the found grids are merged into a coarser-grained grid, and then the index accuracy of the merged grid set is calculated. If the
10 calculated index accuracy is lower than the predetermined precision threshold, the merging stops, and the grid set in this case is the merged grid set. If the calculated index accuracy is not lower than the predetermined precision threshold, all the grids before merging are replaced with the merged coarser-grained grid in the grid set, and then the above process is
15 repeated to find the next set of two or more grids subdivided from the same grid, and whether to merge is determined until the merging stops. The merging mode in this example may be able to further reduce the number of index grids, but usually does not have the effect of minimizing the number of grid indexes.

In one implementation, a knapsack problem may be used to solve the merging mode that
20 minimizes the number of grid indexes. The knapsack problem may be described as: given a group of items, each item has its own weight and price, and within a defined total weight, how to choose to maximize the total price of the items. In some embodiments of the present application, a knapsack problem may be generated in the case where a merged grid containing the boundary of the geo-fence is considered as an item, the increase in the area
25 outside the geo-fence when the item is considered as the index grid in comparison with the

area outside the geo-fence before merging is used as the weight of the item, the reduction in the number of grids when the item is used as the index grid in comparison with the number of grids before merging is considered as the price of the item, and the area outside the geo-fence corresponding to the exceedance of the index accuracy of the grid set relative to the predetermined precision threshold is considered as a limited total weight; and the optimal solution to this knapsack problem is considered as a result of grid merging. The result of the grid merging is updated into the grid set to obtain the merged grid set. For example, grids in the result of the grid merging may be added to the grid set, and all the original grids covering the same geographic area with the grids in the result of the grid merging are deleted.

After the knapsack problem is generated, its optimal solution may be obtained by referring to various solutions to the knapsack problem in the prior art, and the embodiments of the present application do not limit it.

A specific implementation of solving the knapsack problem is given below. Given that Item a is a merged grid; a grid merged by Item 1 and other items may also be considered as an item, denoted as Item 2; then, Item 2 is a parent item of Item 1, and Item 1 is a child item of Item 2. Since Item 1 and Item 2 contain the same geographic area, and when Item 1 functions as an element of a grid set, Item 2 is certainly not an element of the grid set, and vice versa. That is to say, items having a parent-child relationship cannot exist in one knapsack at the same time, and items having a parent-child relationship include items having direct and indirect parent-child relationships. For example, if Item 3 is the parent item of Item 2, item 1 and item 3 cannot exist in one knapsack at the same time.

When solving the knapsack problem in the present implementation, the tree hierarchy of items may be constructed according to the parent-child relationship between the items, and under a constraint condition that items with the parent-child relationship cannot exist in one knapsack at the same time, all knapsack states of the knapsack problem are then enumerated;

each knapsack state is to be solved, and the optimal one in all the answers is the optimal solution to the knapsack problem.

Specifically, with the lower-limit granularity (i.e., the granularity of the densest grids in the grid set before merging) as the starting point of the current granularity, grids of the current granularity in the grid set may be merged into coarser-grained grids level by level; and with each merged grid as an item, the weight and price of each item are determined until all grids are merged into the circumscribed grid. A tree hierarchy of items may be generated according to the parent-child relationship between the items; leaf nodes (i.e., nodes without subsequent nodes) of the tree hierarchy are grids formed by the grids of the lower-limit granularity, and a root node (i.e., node without a precursor node) is the circumscribed grid.

After the tree hierarchy of the items is constructed, the knapsack state of each item that functions as a leaf node in the tree hierarchy is determined. With child items functioning as leaf nodes as starting points, an intersection of the Cartesian product of knapsack states of all the child items of the parent item and the parent item itself as the knapsack state of the parent item, iteration is performed until the knapsack state of the root node in the tree hierarchy corresponding to the circumscribed grid is finally obtained. In other words, starting from child items functioning as leaf nodes, the knapsack state of their parent item is calculated upward level by level in the tree hierarchy according to Equation 3 until the knapsack state of the root node is calculated.

$$State^F = Node^F + \prod_{i=1}^k State_i^G \quad \text{Equation 3}$$

In Equation 3, $State^F$ is the knapsack state of the parent item, $Node^F$ is the parent item, $State_i^G$ is the knapsack state of the i -th child item, and k is the number of child items of the parent item.

After the parent item functioning as the root node (i.e., the knapsack state of the circumscribed grid) is obtained through iteration, all the knapsack states of the circumscribed grid are all the knapsack states of the knapsack problem. Each knapsack state may be regarded as a basic knapsack problem. According to the solution to the basic knapsack
5 problem in the prior art, each knapsack state may be solved, and the optimal one of all the solutions to the knapsack states is the optimal solution to the knapsack problem. By taking the optimal solution to the knapsack problem in the present implementation as the merging result of grids in the grid set, the minimum number of index grids of the geo-fence may be obtained while the predetermined precision threshold is met.

10 It can be seen that, in some embodiments of the present application, a smallest single grid containing the geo-fence is considered as the circumscribed grid of the geo-fence, and with the circumscribed grid of the geo-fence as a starting point, grid(s) containing the boundary of the geo-fence are subdivided into finer-grained grids level by level until a predetermined precision threshold is met; by only increasing the granularity of the grids
15 containing the boundary of the geo-fence but not further subdividing the grids within the geo-fence and determining grids of different granularities as index grids, the number of index grids is greatly decreased, index storage is less complex, and index query speed is increased.

Further, for the grid set of which the index accuracy exceeds the predetermined precision threshold, a knapsack problem model is used to merging part of grids in the grid set into a
20 coarser-grained grid, the optimal solution is obtained by constructing a tree hierarchy of the knapsack problem, thereby minimizing the number of indexes while meeting the index accuracy requirement.

In an application example of the present application, the process flow shown in FIG. 3 is used to obtain index grids of the geo-fence, and it is required that these index grids 100%
25 cover the area of the geo-fence, and the index accuracy is not lower than a predetermined

precision threshold. The following description refers to an example of a geo-fence called Geof-Demo.

In step 310, a circumscribed grid of the geo-fence is determined. The Geof-Demo geo-fence and the determined circumscribed grid are shown in FIG. 4.

5 In step 320, a grid set is formed by using the circumscribed grid as an element, and grid(s) containing the boundary of the geo-fence are subdivided into finer-grained grids level by level until index accuracy of a subdivided grid set is not lower than a predetermined precision threshold.

Given that a granularity system adopted by the Geof-Demo is as follows: in the case of
10 grid subdivision by one granularity, the original grid is subdivided into four equal grids. Starting from a circumscribed grid of the Geof-Demo, the distribution of grids after the first four subdivisions is as shown in FIG. 5; given that the index accuracy of the grid set after the four subdivisions exceeds a predetermined precision threshold, then the subdivision stops. In this case, all the elements in the grid set are grids with the gray background in FIG.
15 5, where grids containing the boundary of the Geof-Demo have the same granularity which is the finest granularity (i.e., the lower-limit granularity) of all the grids in the grid set, while grids within the boundary of the Geof-Demo have different granularities. Grids with a white background in FIG. 5 are grids outside the geo-fence and will be deleted from the grid set during the subdivision process.

20 In step 330, a determination whether the index accuracy of the grid set is equal to the predetermined precision threshold is made. If the determination result is yes, the process proceeds to step 360. If the index accuracy is greater than the predetermined precision threshold, step 340 is performed.

In step 340, a knapsack problem is generated in the case where a merged grid containing
25 the boundary of the geo-fence is considered as an item, the increase in the area outside the

geo-fence when the item is considered as the index grid in comparison with the area outside the geo-fence before merging is used as the weight of the item, the reduction in the number of grids when the item is used as the index grid in comparison with the number of grids before merging is considered as the price of the item, and the area outside the geo-fence corresponding to the exceedance of the index accuracy of the grid set relative to the predetermined precision threshold is considered as a limited total weight.

With the grids, in the grid set, containing the boundary of the geo-fence and having the lower-limit granularity as starting points, grids are merged into coarser-grained grids level by level, and with each merged grid as an item, the weights and prices of items are determined. The parent-child relationship of the items is determined according to the merged hierarchy, and the tree hierarchy of a knapsack problem is constructed.

Taking the grids d1, d2, d3, e1, f1, f2 and f3 with the lower-limit granularity shown in the upper right corner of FIG. 6 as an example, d1, d2 and d3 may be merged into a coarser-grained grid which is Item d; e1 may be merged into a coarser-grained grid which is Item e; f1, f2, and f3 may be merged into a coarser-grained grid which is Item f. With Item d as an index grid, the increase in the area outside the geo-fence is one lower-limit granularity grid in comparison with the area outside the geo-fence before merging of the grid set, and the reduction in the number of index grids is 2, then it is concluded that, for Item d, weight $w = 1$, price $p = 2$; by that analogy, it is concluded that, for Item e, weight $w = 3$, price $p = 0$, and for Item f, weight $w = 1$, price $p = 2$.

The grid merging further proceeds to a coarser granularity, and d may be merged with other grids in the grid set into a coarser-grained grid which is Item b (indicated by a dashed box in FIG. 6), and Item b is the parent item of Item d; e and f may be merged with other grids in the grid set into a coarser-grained grid which is Item c (indicated by a dashed box in FIG. 6), and Item c is the parent item of Item e and Item f. With Item b as an index grid, the

increase in the area outside the geo-fence is one lower-limit granularity grid in comparison with the area outside the geo-fence before merging of the grid set, and the reduction in the number of index grids is 5, then it is concluded that, for Item b, weight $w = 1$, price $p = 5$; by that analogy, it is concluded that, for Item c, weight $w = 8$, price $p = 4$. Item b and Item c
5 may also be merged with other grids in the grid set into a coarser-grained grid which is Item a, it is concluded that, for Item a, weight $w=9$, price $p=12$. By that analogy, the merging stops until the circumscribed grid of Geof-Demo is obtained.

According to the parent-child relationship between the items at the time of grid merging, a tree hierarchy from items (such as Item d) merged from grids of the lower-limit granularity
10 to the circumscribed grid may be obtained. Taking the above items a to f as an example, the tree hierarchy shown in FIG. 7 may be obtained according to the parent-child relationship between the items at the time of merging. It should be noted that the tree hierarchy shown in FIG. 7 is only a subtree of the tree hierarchy of Geof-Demo.

In step 350, according to a tree hierarchy of items, under a constraint condition that items
15 containing the same geographical area (i.e., items having direct or indirect parent-child relationship) cannot exist in one knapsack at the same time, all the knapsack states of the knapsack problem are enumerated, each knapsack state is solved, and the optimal one in all solutions is determined as the result of grid merging.

Starting with the states of leaf nodes (i.e., items merged from the grids of the lower-limit
20 granularity) of the tree hierarchy, iteration is performed according to Equation 3 until the states of the circumscribed grid are obtained, that is, all the knapsack states of the knapsack problem.

Taking the subtree shown in FIG. 7 as an example, the states of leaf nodes, i.e., Item d, Item e, and Item f, are first determined: state D of Item d = {d}, state E of Item d = {e}, and
25 state F of Item f = {f}. According to Equation 3, the state of Item b is calculated as $B = b +$

$D = \{\{b\}, \{d\}\}$, and the state of Item c is calculated as $C = c + E \diamond F = \{c, \{e, f\}\}$. The state of Item a is then calculated as $A = a + B \diamond C = \{a, \{b, c\}, \{b, e, f\}, \{d, c\}, \{d, e, f\}\}$, and by that analogy, the states of the circumscribed grid may be obtained.

The state of each circumscribed grid is a basic knapsack problem which may be solved according to the prior art. The optimal one of all the solutions to the states is considered as the result of the grid merging. The grid set is then updated according to the result of the grid merging.

In step 360, the grids in the grid set are used as the index grids of the geo-fence. The grids in the final grid set of Geof-Demo are shown in FIG. 8.

After applying the method in the present application example to practice, significant effects have been achieved in comparison to the prior art using single-grained index grids. With a geographical area including fences such as office buildings, residential areas, shopping malls, airports, hospitals, etc. as an object, the comparison results are shown in Table 1:

Index	Single-grained index grid	The present application example
Number of index grids	5.3 billion	0.15 billion
Index coverage	100%	100%
Index accuracy	90.5%	95%

Table 1

It can be seen that in the case of a higher precision, the number of indexes in the present application example is reduced by more than 30 times than the single-grained index grids. The higher index accuracy makes the positioning of a geo-fence-based service more accurate, thereby improving service performance and reducing the rate of interruption; a

fewer number of indexes make the index data occupy less storage space, thus improving the environment on which the service construction depends and improving the robustness and flexibility of the service.

Corresponding to the foregoing process implementation, embodiments of the present application further provide an apparatus for determining index grids of a geo-fence. The apparatus may be implemented by software, or hardware, or a combination of hardware and software. Taking the software implementation as an example, as a logical apparatus, it is formed in a way that a CPU (Central Process Unit) of a device in which the apparatus is located reads corresponding computer program instructions into a memory for operation. At the hardware level, in addition to the CPU, memory, and storage shown in FIG. 9, the device in which the apparatus for determining index grids of a geo-fence is located typically includes other hardware such as a chip for transmitting and receiving wireless signals, and/or other hardware such as a board card for implementing a network communication function.

FIG. 10 shows an apparatus for determining index grids of a geo-fence according to some embodiments of the present application, including a circumscribed grid unit, a grid subdividing unit, and an index determining unit, wherein the circumscribed grid unit is configured to determine a circumscribed grid of the geo-fence, the circumscribed grid being a smallest single grid containing the geo-fence; the grid subdividing unit is configured to, with the circumscribed grid as a starting point, subdivide grid(s) containing the boundary of the geo-fence into finer-grained grids level by level until index accuracy of a subdivided grid set is not lower than a predetermined precision threshold, wherein the grid set is composed of grids within the geo-fence and the grids containing the boundary of the geo-fence, and the index accuracy is a ratio of the area of the geo-fence to a sum of areas of all the grids in the grid set; and the index determining unit is configured to take the grids in the grid set as the index grids of the geo-fence.

In an implementation, the apparatus further includes a grid merging unit configured to merge part of grids in the grid set into coarser-grained grids containing the boundary of the geo-fence when the index accuracy of the grid set is higher than the predetermined precision threshold, so that the index accuracy after merging is not lower than the predetermined precision threshold; and the index determining unit is specifically configured to take the grids in the grid set after merging as index grids of the geo-fence.

In the above implementation, the grid merging unit merges part of grids in the grid set into coarser-grained grids containing the boundary of the geo-fence so that the index accuracy after merging is not lower than the predetermined precision threshold, which includes: generating a knapsack problem in the case where a merged grid containing the boundary of the geo-fence is considered as an item, the increase in the area outside the geo-fence when the item is considered as the index grid in comparison with the area outside the geo-fence before merging is used as the weight of the item, the reduction in the number of grids when the item is used as the index grid in comparison with the number of grids before merging is considered as the price of the item, and the area outside the geo-fence corresponding to the exceedance of the index accuracy of the grid set relative to the predetermined precision threshold is considered as a limited total weight; and taking an optimal solution to the knapsack problem as a result of grid merging. The index determining unit is specifically configured to update the grid set according to the result of the grid merging and taking grids in the updated grid set as index grids of the geo-fence.

In one example, the grid merging unit determines the optimal solution to the knapsack problem as the result of the grid merging, which includes: constructing a tree hierarchy of items of the knapsack problem, wherein a parent item in the tree hierarchy is merged from its child items; enumerating all knapsack states of the knapsack problem under a constraint condition that items with the parent-child relationship cannot exist in one knapsack at the

same time; and solving an answer to each knapsack state, and taking the optimal one in all the answers as the optimal solution to the knapsack problem.

In the above example, the grid merging unit constructs a tree hierarchy of the items of the knapsack problem, which includes: with the lower-limit granularity as a starting point of the current granularity, merging grids of the current granularity in the grid set into coarser-grained grids level by level; with each merged grid as an item, determining the weight and price of each item until all grids are merged into the circumscribed grid; and generating a tree hierarchy of items according to the parent-child relationship between the items, wherein the lower-limit granularity is the granularity of the densest grids in the grid set before merging.

In the above example, under a constraint condition that items with the parent-child relationship cannot exist in one knapsack at the same time, the grid merging unit enumerates all knapsack states of the knapsack problem, which includes: determining the knapsack state of each item functioning as a leaf node of the tree hierarchy; and with child items functioning as leaf nodes as a starting point and an intersection of the Cartesian product of knapsack states of all the child items of the parent item and the parent item itself as the knapsack state of the parent item, performing iteration until the knapsack states of the circumscribed grid corresponding to the root node in the tree hierarchy are obtained, wherein the knapsack states of the circumscribed grid are all the knapsack states of the knapsack problem.

Embodiments of the present application provide a computer device including a memory and a processor. The memory herein stores a computer program executable by the processor; and when running the stored computer program, the processor executes the steps of the method for determining index grids of a geo-fence in the embodiments of the present application. For a detailed description of the steps of the method for determining index grids of a geo-fence, reference may be made to the above content, and it will not be repeated.

Embodiments of the present application provide a computer-readable storage medium which stores computer programs, and when the computer programs are run by a processor, the steps of the method for determining index grids of a geo-fence in the embodiments of the present application are executed. For a detailed description of the steps of the method for determining index grids of a geo-fence, reference may be made to the above content, and it will not be repeated.

The embodiments described above are merely preferred embodiments of the present application and not intended to limit the present application. Any modifications, equivalent substitutions, improvements and the like, made within the spirit and principle of the present application shall be covered in the scope of the present application.

In a typical configuration, a computing device includes one or more processors (CPUs), input/output interfaces, network interfaces, and memories.

The memory may include a non-permanent memory, a random access memory (RAM), and/or a nonvolatile memory, such as a read-only memory (ROM) or a flash RAM, in computer-readable media. The memory is an example of computer-readable media.

Computer-readable media include permanent and non-permanent, removable and non-removable media and may store information by any method or technology. The information may be a computer-readable instruction, a data structure, a module of a program or other data. Examples of computer storage media include, but are not limited to, a phase change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), other types of random access memory (RAM), a read only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or other memory technologies, a read-only optical disk read-only memory (CD-ROM), a digital versatile disc (DVD) or other optical storages, a magnetic tape cassette, a magnetic disk storage or other magnetic storage devices or any other non-transmission

medium for storing information that may be accessed by computing devices. As defined herein, the computer-readable media do not include transitory media, such as modulated data signals and carriers.

It is also to be understood that the term “include,” “comprise,” or any of other variants thereof is intended to cover non-exclusive inclusions such that a process, method, article, or device that includes a series of elements not only includes those elements but also includes other elements that are listed explicitly, or also includes inherent elements of this process, method, article, or device. In the absence of more restrictions, an element defined by the sentence "including a/an..." does not preclude other identical elements existing in the process, method, article or device that includes the element.

Those skilled in the art should understand that the embodiments of the present application may be provided as a method, a system, or a computer program product. Accordingly, this application may take the form of an entire hardware embodiment, an entire software embodiment, or an embodiment combining software and hardware. Moreover, the present application may take the form of a computer program product implemented on one or more computer storage media (including, but not limited to, disk memories, CD-ROMs and optical memories) comprising computer program codes.

CLAIMS

1. A method for determining index grids of a geo-fence, comprising:

determining a circumscribed grid of a fence, the circumscribed grid being a smallest single grid containing the fence;

with the circumscribed grid as a starting point, subdividing grid(s) containing a boundary of the fence into finer-grained grids level by level until an index accuracy of a subdivided grid set is not lower than a predetermined precision threshold, the grid set being composed of grids within the fence and grids containing the boundary of the fence, the index accuracy being a ratio of an area of the fence to a sum of areas of all the grids in the grid set;

taking the grids in the grid set as index grids of the fence; if the index accuracy of the grid set is higher than the predetermined precision threshold, merging part of the grids in the grid set into coarser-grained grids containing the boundary of the fence so that the index accuracy after merging is not lower than the predetermined precision threshold, wherein the merging comprises: generating a knapsack problem in which a merged grid containing the boundary of the fence is considered as an item, an increase in an area outside the fence when the item is considered as an index grid in comparison with the area outside the fence before merging is considered as a weight of the item, a reduction in a number of grids when the item is considered as the index grid in comparison with a number of grids before merging is considered as a price of the item, and an area outside the fence corresponding to an exceedance of the index accuracy of the grid set relative to the predetermined precision threshold before merging is considered as a limited total weight; and taking an optimal solution to the knapsack problem as a result of grid merging; and

updating the grid set according to the result of the grid merging and taking grids in the updated grid set as index grids of the fence.

2. The method according to claim 1, wherein the taking an optimal solution to the knapsack problem as a result of grid merging comprises: constructing a tree hierarchy of items of the knapsack problem, a parent item in the tree hierarchy being merged from its child items; enumerating all knapsack states of the knapsack problem under a constraint condition that items with a parent-child relationship cannot exist in one knapsack at the same time; and solving an answer to each knapsack state, and taking an optimal one in all the answers as the optimal solution to the knapsack problem.

3. The method according to claim 2, wherein the constructing a tree hierarchy of items of the knapsack problem comprises: with a lower-limit granularity as a starting point of a current granularity, merging grids of the current granularity in the grid set into coarser-grained grids level by level; with each merged grid as an item, determining the weight and the price of the item until all grids are merged into the circumscribed grid; and generating the tree hierarchy of items according to the parent-child relationship between the items, wherein the lower-limit granularity is a granularity of densest grids in the grid set before merging.

4. The method according to claim 2, wherein the enumerating all knapsack states of the knapsack problem under a constraint condition that items with the parent-child relationship cannot exist in one knapsack at the same time comprises: determining the knapsack state of each item functioning as a leaf node of the tree hierarchy; and with child items functioning as leaf nodes as a starting point and an intersection of a Cartesian product of knapsack states of all the child items of the parent item and the parent item itself as the knapsack state of the parent item, performing iteration until the knapsack states of the circumscribed grid corresponding to a root node in the tree hierarchy are obtained, wherein the knapsack states of the circumscribed grid are all the knapsack states of the knapsack problem.

5. An apparatus for determining index grids of a geo-fence, comprising:

a circumscribed grid unit configured to determine a circumscribed grid of a fence, the circumscribed grid being a smallest single grid containing the fence;

a grid subdividing unit configured to, with the circumscribed grid as a starting point, subdivide grid(s) containing a boundary of the fence into finer-grained grids level by level until an index accuracy of a subdivided grid set is not lower than a predetermined precision threshold, the grid set being composed of grids within the fence and grids containing the boundary of the fence, the index accuracy being a ratio of an area of the fence to a sum of areas of all the grids in the grid set; and

an index determining unit configured to take the grids in the grid set as index grids of the fence; and

a grid merging unit configured to merge part of grids in the grid set into coarser-grained grids containing the boundary of the fence when the index accuracy of the grid set is higher

than the predetermined precision threshold, so that the index accuracy after merging is not lower than the predetermined precision threshold; wherein the merging comprises: generating a knapsack problem in which a merged grid containing the boundary of the fence is considered as an item, an increase in an area outside the fence when the item is considered as an index grid in comparison with the area outside the fence before merging is considered as a weight of the item, a reduction in a number of grids when the item is considered as the index grid in comparison with a number of grids before merging is considered as a price of the item, and an area outside the fence corresponding to an exceedance of the index accuracy of the grid set relative to the predetermined precision threshold is before merging considered as a limited total weight; and taking an optimal solution to the knapsack problem as a result of grid merging,

wherein the index determining unit is further configured to update the grid set according to the result of the grid merging and take grids in the updated grid set as index grids of the fence.

6. The apparatus according to claim 5, wherein the grid merging unit taking an optimal solution to the knapsack problem as a result of grid merging comprises: constructing a tree hierarchy of items of the knapsack problem, a parent item in the tree hierarchy being merged from its child items; enumerating all knapsack states of the knapsack problem under a constraint condition that items with a parent-child relationship cannot exist in one knapsack at the same time; and solving an answer to each knapsack state, and taking an optimal one in all the answers as the optimal solution to the knapsack problem.

7. The apparatus according to claim 6, wherein the grid merging unit constructing a tree hierarchy of items of the knapsack problem comprises: with a lower-limit granularity as a starting point of a current granularity, merging grids of the current granularity in the grid set into coarser-grained grids level by level; with each merged grid as an item, determining the weight and the price of the item until all grids are merged into the circumscribed grid; and generating the tree hierarchy of items according to the parent-child relationship between the items, wherein the lower-limit granularity is a granularity of the densest grids in the grid set before merging.

8. The apparatus according to claim 6, wherein the grid merging unit enumerating all knapsack states of the knapsack problem under a constraint condition that items with the parent-child relationship cannot exist in one knapsack at the same time comprises: determining the knapsack state of each item functioning as a leaf node of the tree hierarchy; and with child items functioning as leaf nodes as a starting point and an intersection of a Cartesian product of knapsack states of all the child items of the parent item and the parent item itself as the knapsack state of the parent item, performing iteration until the knapsack states of the circumscribed grid corresponding to a root node in the tree hierarchy are obtained, wherein the knapsack states of the circumscribed grid are all the knapsack states of the knapsack problem.

9. A computer device, comprising: a memory and a processor, wherein the memory stores a computer program executable by the processor; and when the processor runs the computer program, the steps of any of claims 1-4 are executed.

10. A computer-readable storage medium, having stored there on a computer program, wherein when the computer program is run by a processor, the steps of any of claims 1-4 are executed.