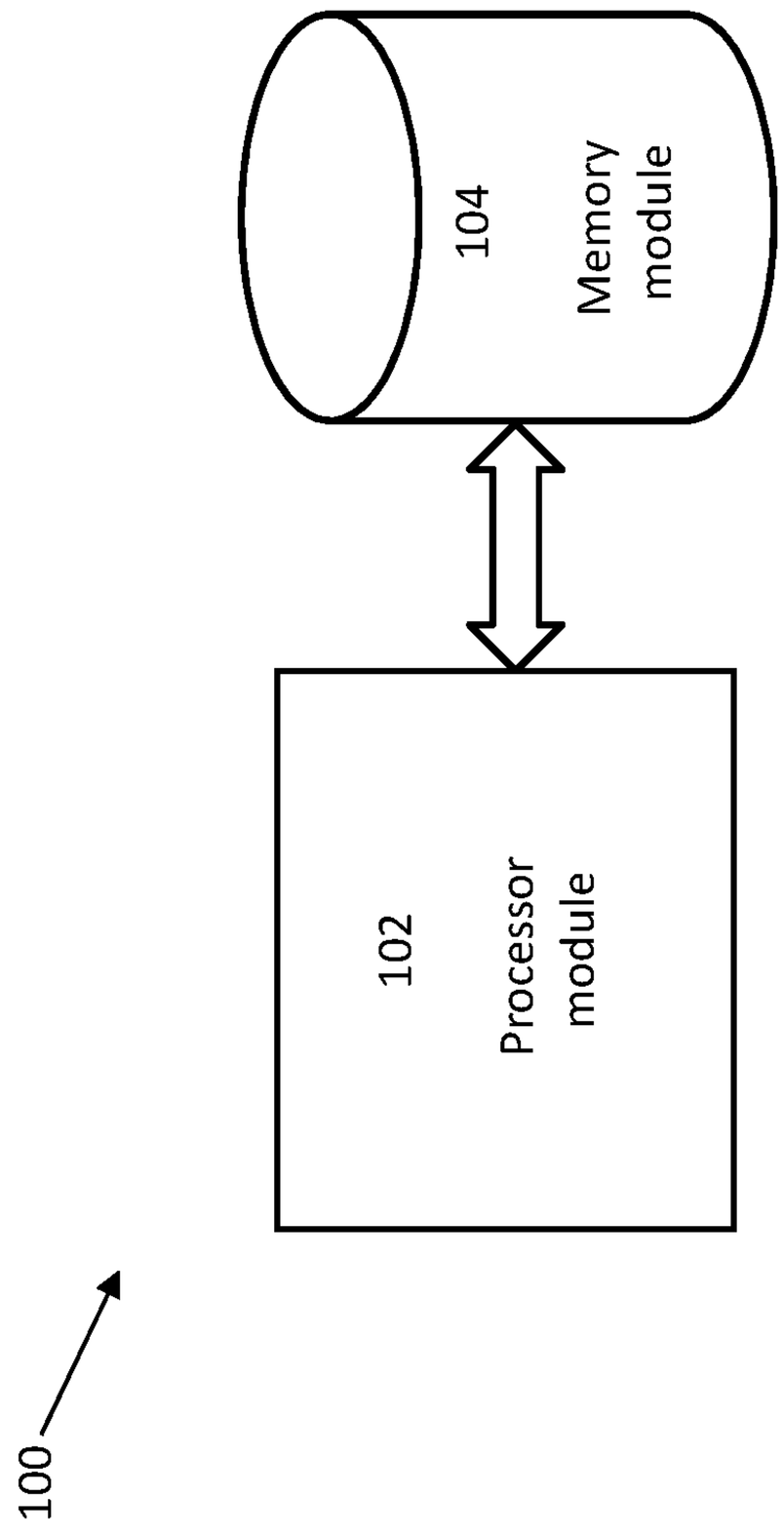



Figure 1

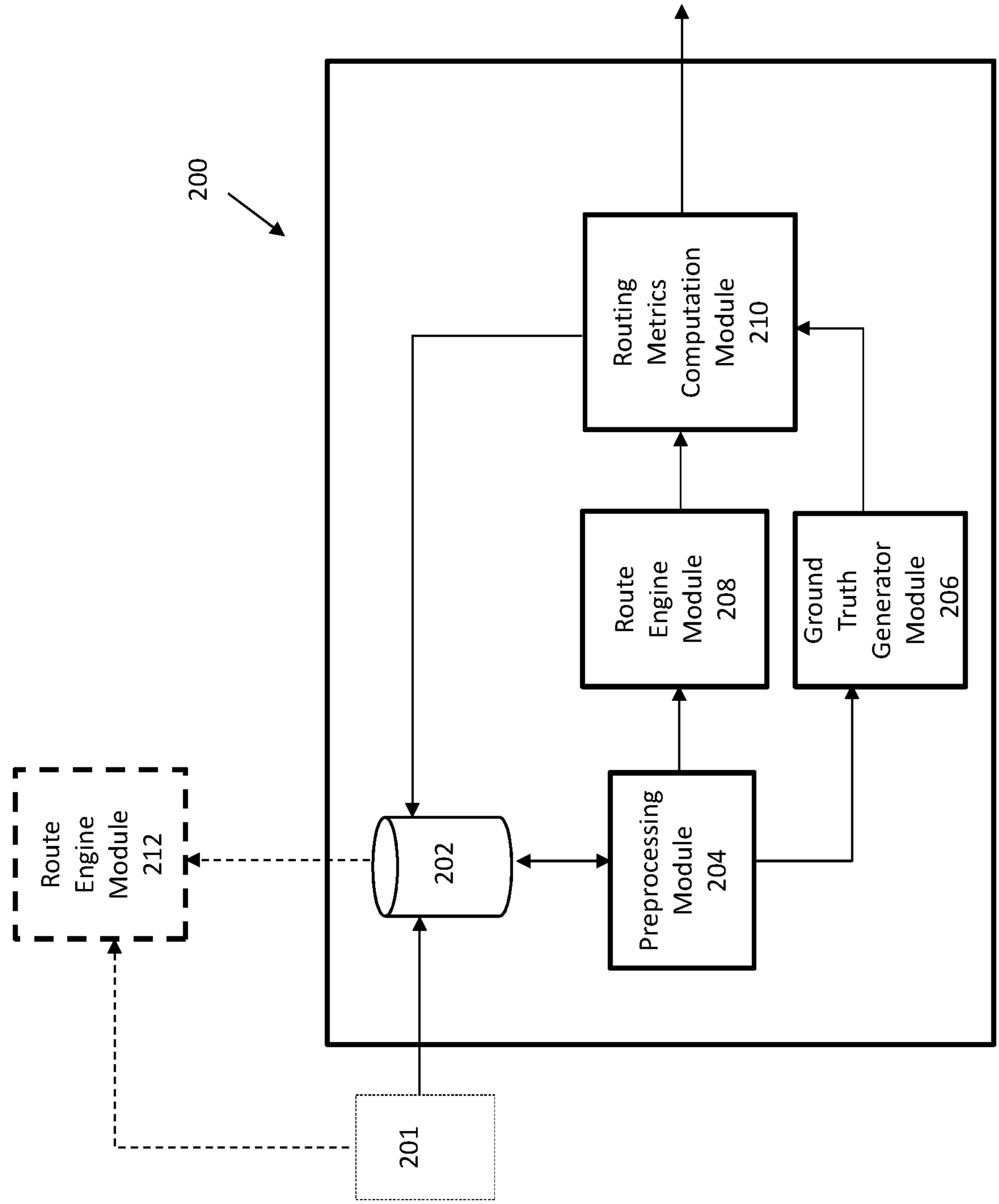

200
Routing Metrics Computation Module 210
Route Engine Module 208
Ground Truth Generator Module 206
Preprocessing Module 204
202
Route Engine Module 212
201

Figure 2

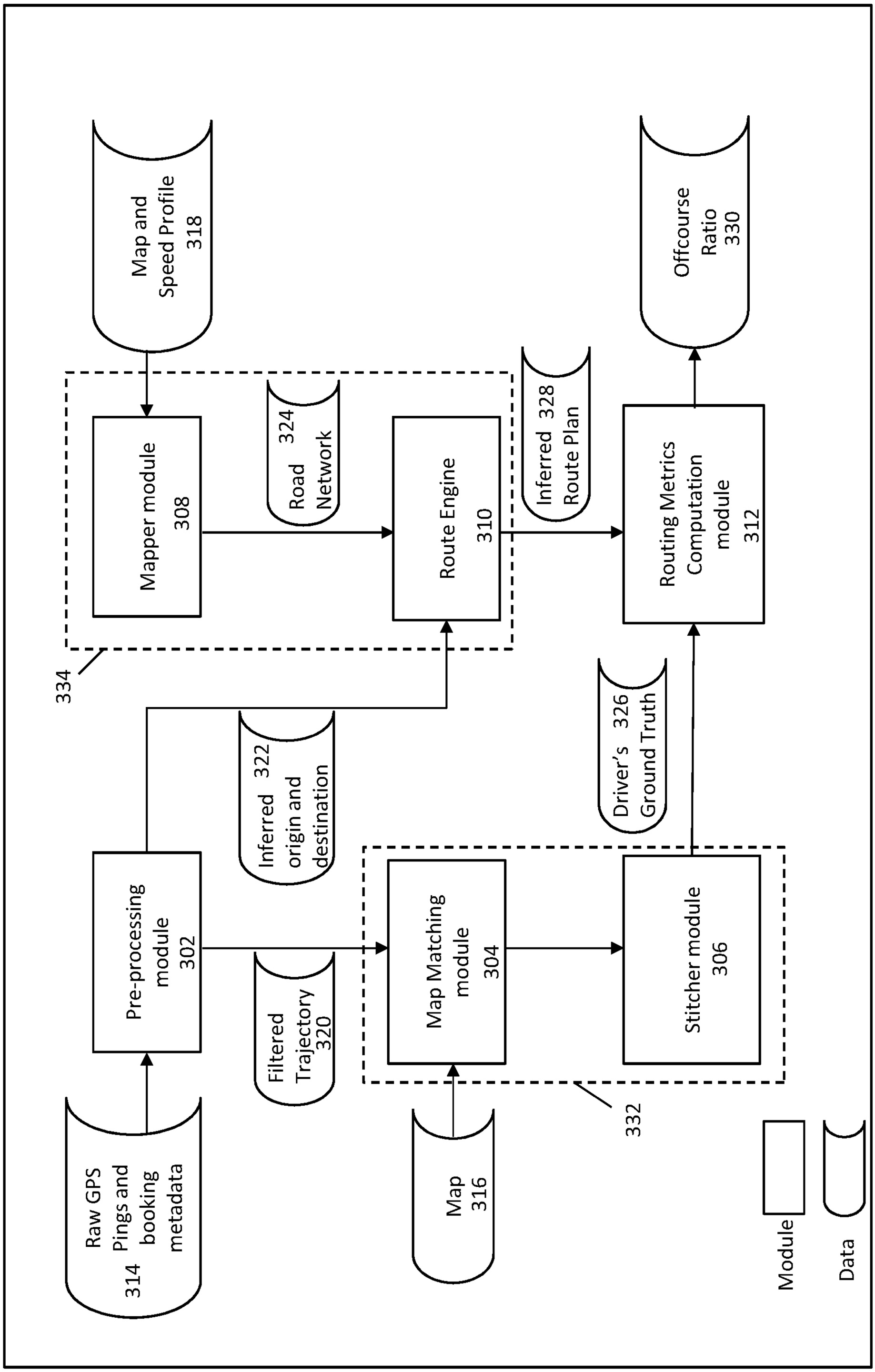

Map and Speed Profile 318
Offcourse Ratio 330
Mapper module 308
Road Network 324
Route Engine 310
Inferred Route Plan 328
Routing Metrics Computation module 312
334
Inferred origin and destination 322
Driver's Ground Truth 326
Pre-processing module 302
Filtered Trajectory 320
Map Matching module 304
Stitcher module 306
332
Raw GPS Pings and booking metadata 314
Map 316
300
Module
Data

Figure 3

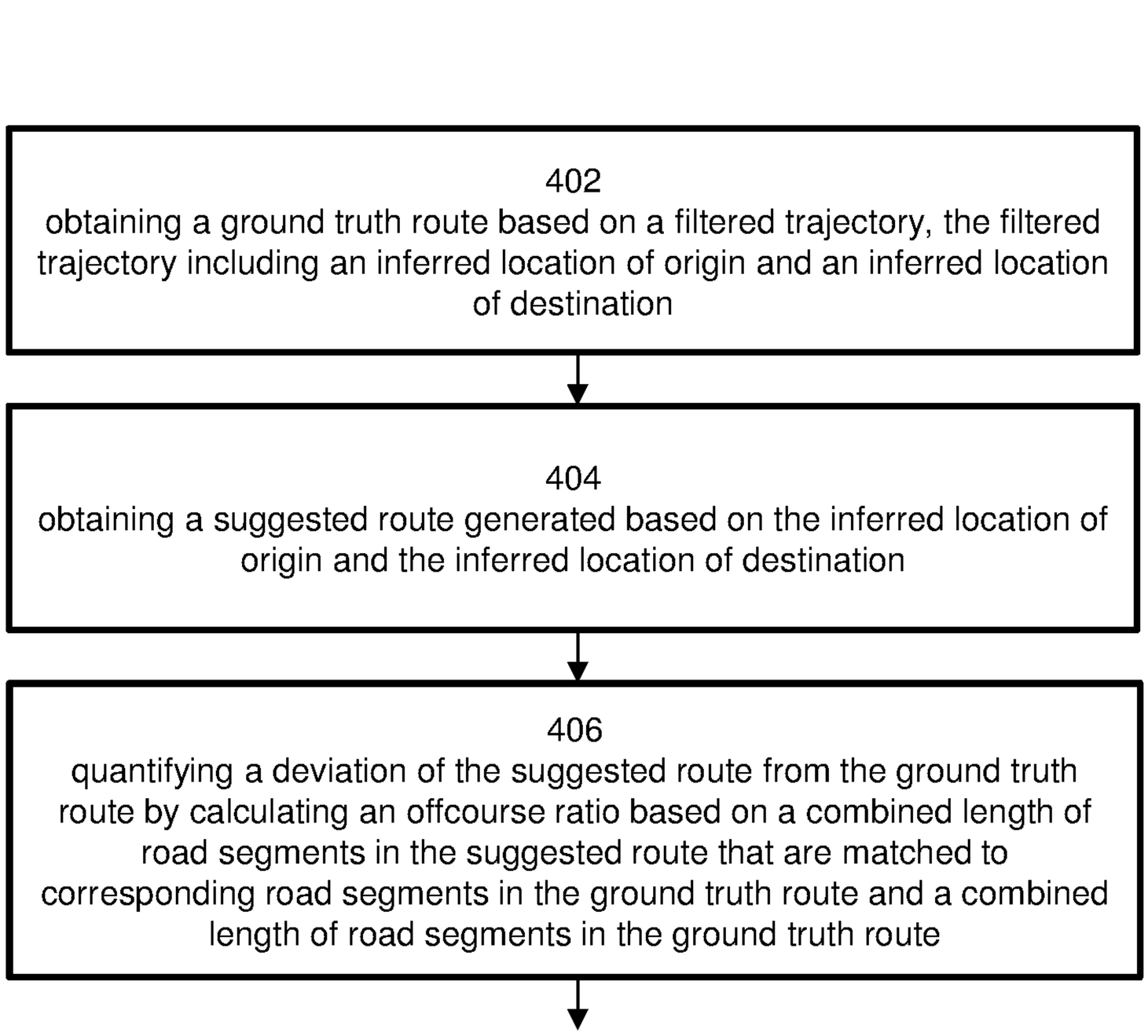

400
402
obtaining a ground truth route based on a filtered trajectory, the filtered trajectory including an inferred location of origin and an inferred location of destination
404
obtaining a suggested route generated based on the inferred location of origin and the inferred location of destination
406
quantifying a deviation of the suggested route from the ground truth route by calculating an offcourse ratio based on a combined length of road segments in the suggested route that are matched to corresponding road segments in the ground truth route and a combined length of road segments in the ground truth route

**Figure 4**

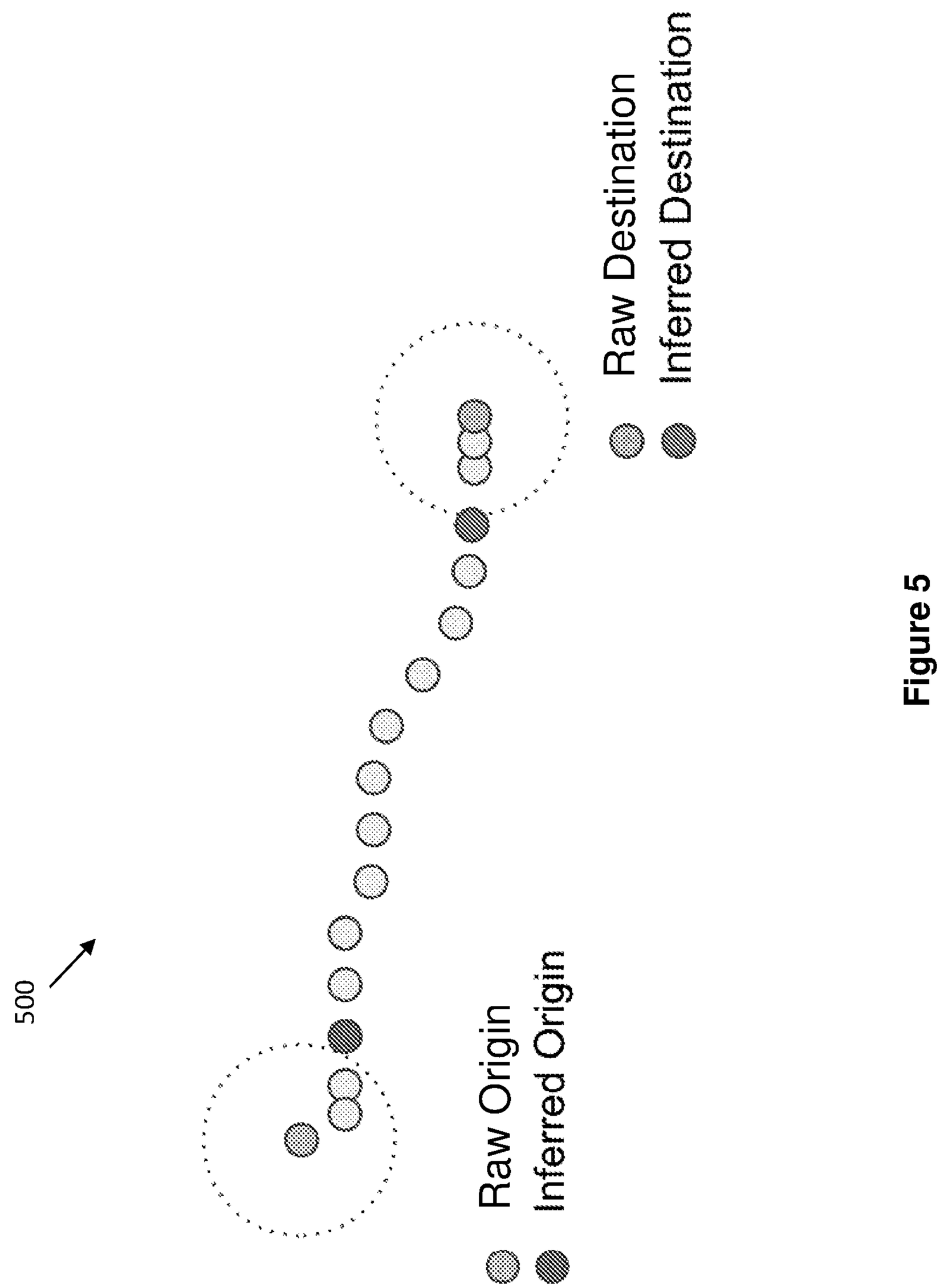

500
Raw Origin
Inferred Origin
Raw Destination
Inferred Destination

Figure 5

600

**Algorithm 1:** SequentialOverlap

**Input:** $\hat{P}$, $P$

**Output:** Length of Overlap

```
1 i, j ← 0 // match P̂(i) with P(j) starting with the first element
2 match ← 0 // length of match
3 while i < size(P̂) and j < size(P) do
      // find the first appearance of P(j) in P̂ with index no smaller than i
4     m ← FirstMatch(P̂, i, P(j))
5     if m >= i then
6         match ← match + len(P(j))
7         i ← m + 1
8     j ← j + 1
9 return match
```

**Figure 6**

SUBSTITUTE SHEET (RULE 26)

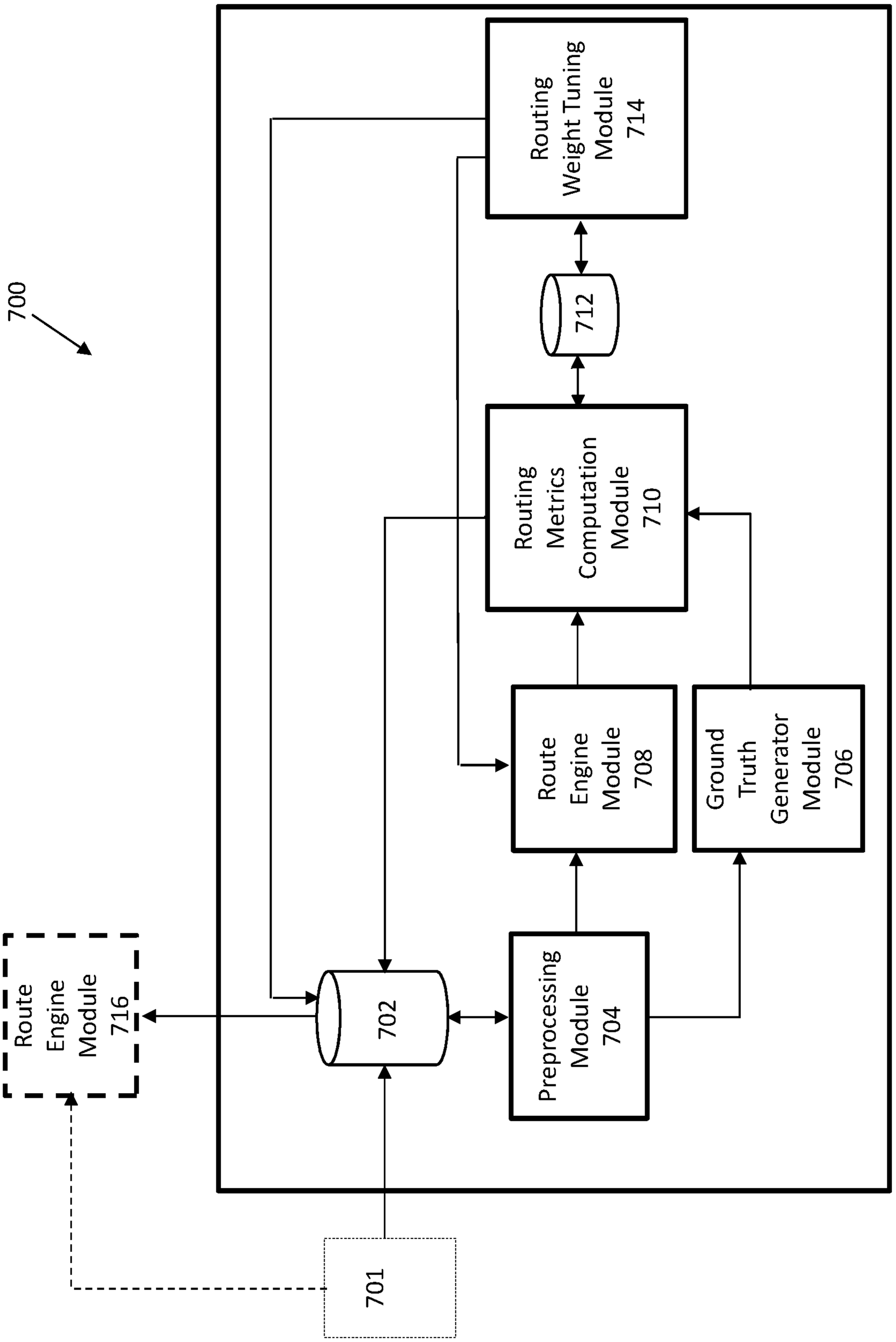

700
Routing Weight Tuning Module 714
712
Routing Metrics Computation Module 710
Route Engine Module 708
Ground Truth Generator Module 706
Preprocessing Module 704
702
Route Engine Module 716
701

Figure 7

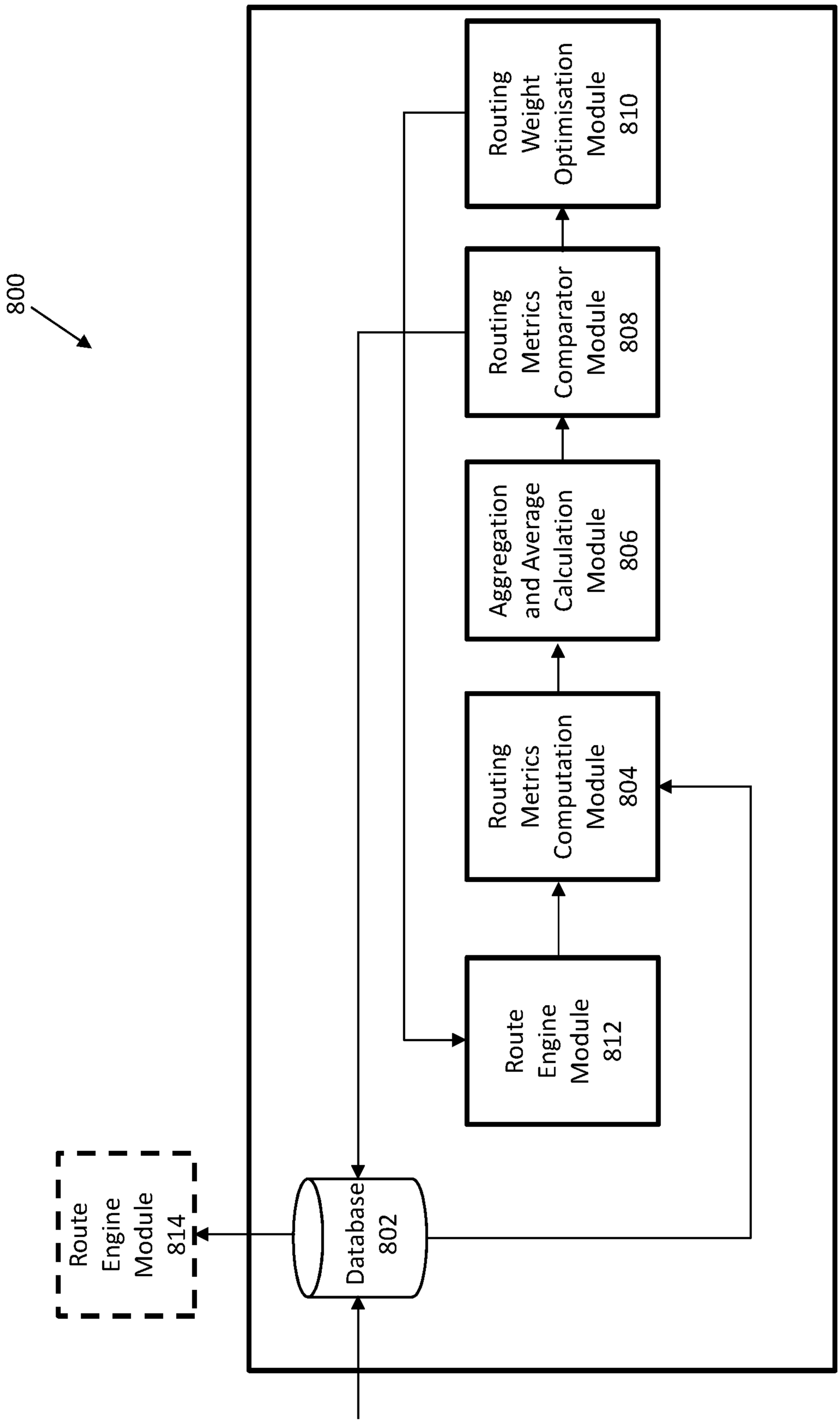

800
Routing Weight Optimisation Module 810
Routing Metrics Comparator Module 808
Aggregation and Average Calculation Module 806
Routing Metrics Computation Module 804
Route Engine Module 812
Database 802
Route Engine Module 814

Figure 8

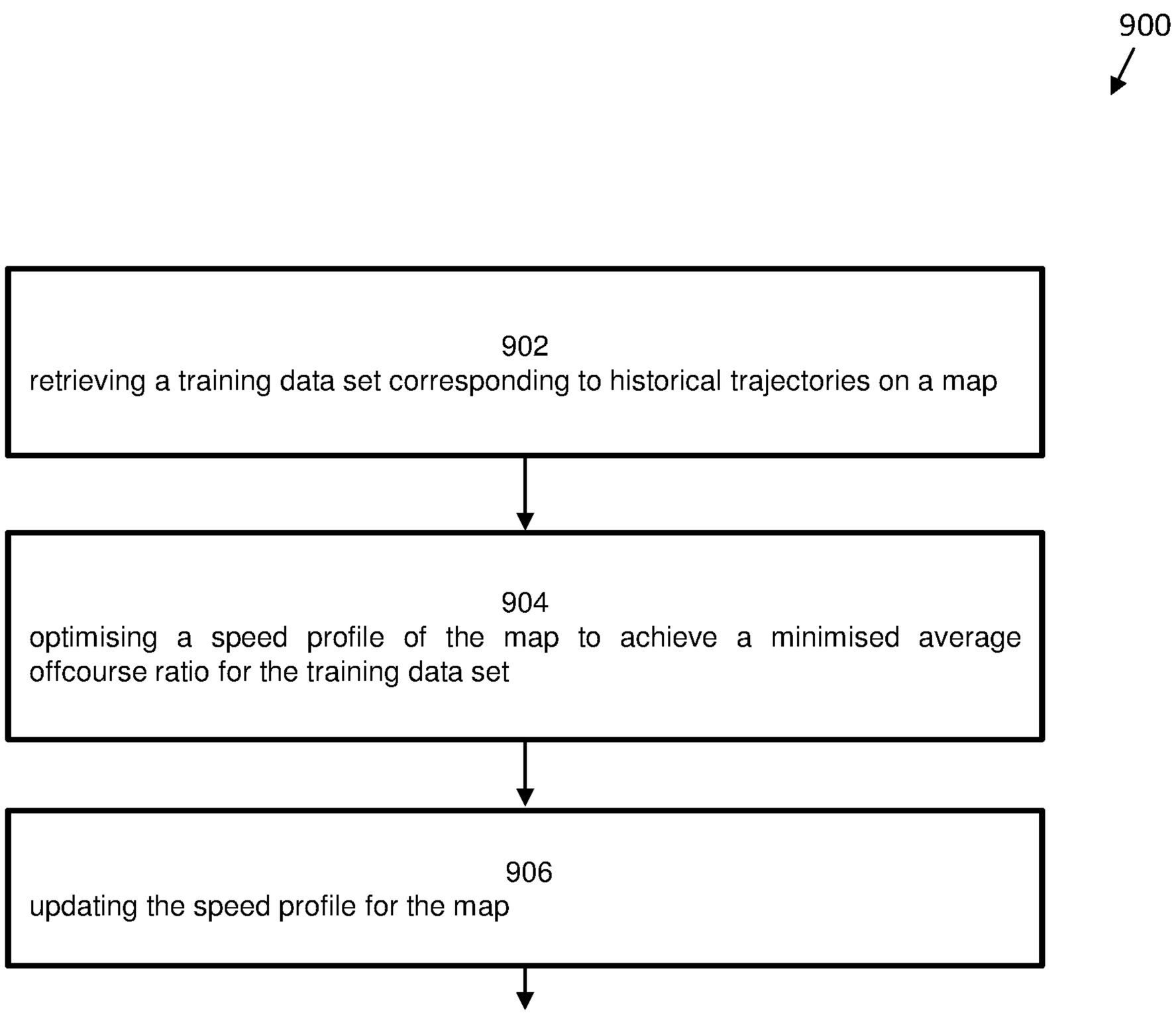

900
902
retrieving a training data set corresponding to historical trajectories on a map
904
optimising a speed profile of the map to achieve a minimised average offcourse ratio for the training data set
906
updating the speed profile for the map

**Figure 9**

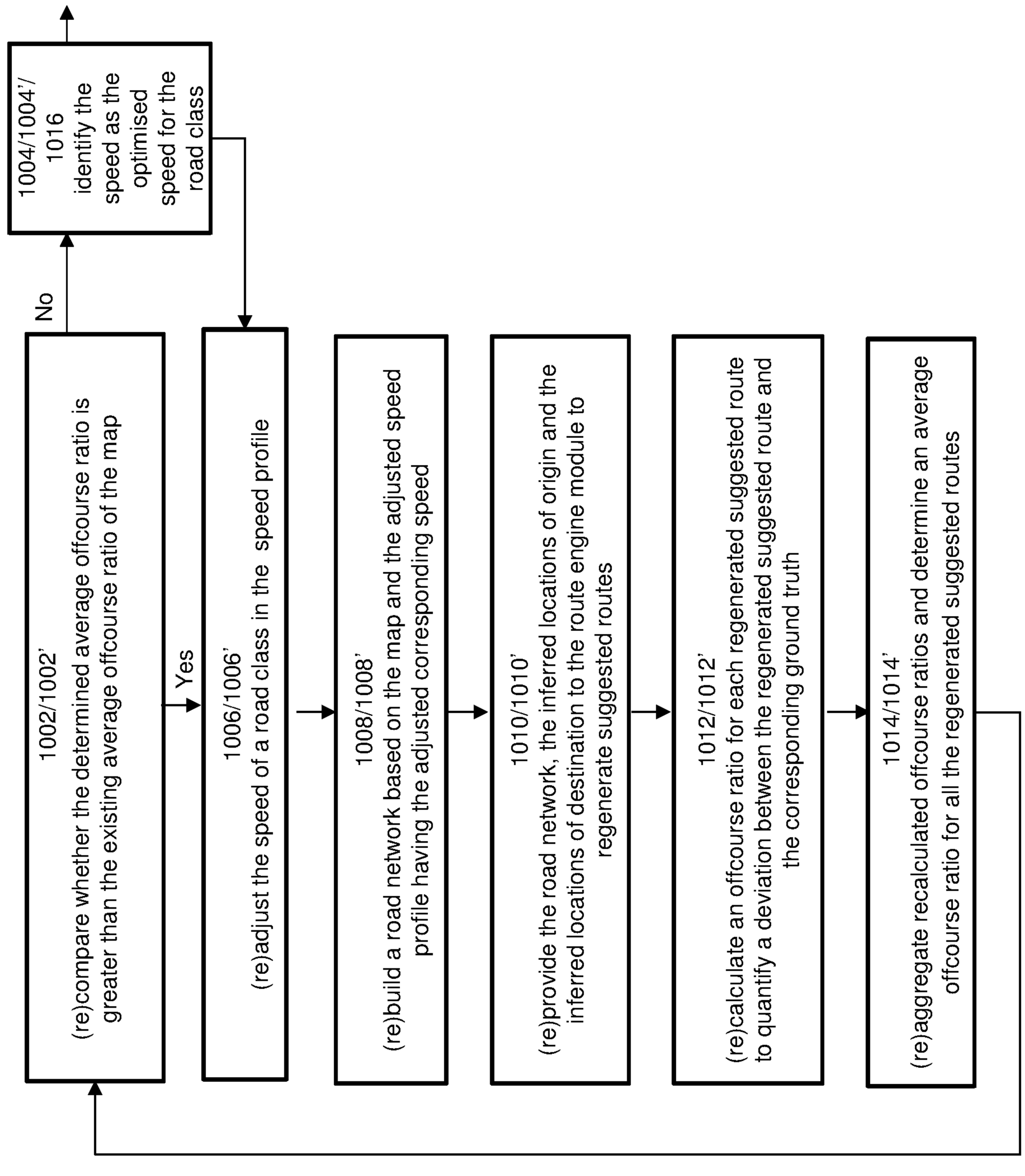

1000
1002/1002'
(re)compare whether the determined average offcourse ratio is greater than the existing average offcourse ratio of the map
No
Yes
1004/1004'/1016
identify the speed as the optimised speed for the road class
1006/1006'
(re)adjust the speed of a road class in the speed profile
1008/1008'
(re)build a road network based on the map and the adjusted corresponding speed profile having the adjusted corresponding speed
1010/1010'
(re)provide the road network, the inferred locations of origin and the inferred locations of destination to the route engine module to regenerate suggested routes
1012/1012'
(re)calculate an offcourse ratio for each regenerated suggested route to quantify a deviation between the regenerated suggested route and the corresponding ground truth
1014/1014'
(re)aggregate recalculated offcourse ratios and determine an average offcourse ratio for all the regenerated suggested routes

**Figure 10**

1100

**Algorithm 2:** Two-step Tuning Framework

**Input:** training data $D$, map, speed profile
**Output:** optimised speed profile

```
// iterate each (road class r, speed s) pair in speed profile
for (r, s) in speed profile do
    // first iteration to find initial value of s
    s* ← s_MIN
    ratio ← CalcRatioAndCompare(D, speed profile, map)
    // first step speed tuning with large step
    while s* < s_MAX do
        UpdateSpeedProfile(r, s*)
        ratio, s ← CalcRatioAndCompare(D, speed profile, map)
        s* ← s* + step_1
    // second step speed tuning with small step
    s* ← s - Δ
    while s* < s + Δ do
        UpdateSpeedProfile(r, s*)
        ratio, s ← CalcRatioAndCompare(D, speed profile, map)
        s* ← s* + step_2
    UpdateSpeedProfile(r, s)
```

**Figure 11**

SUBSTITUTE SHEET (RULE 26)

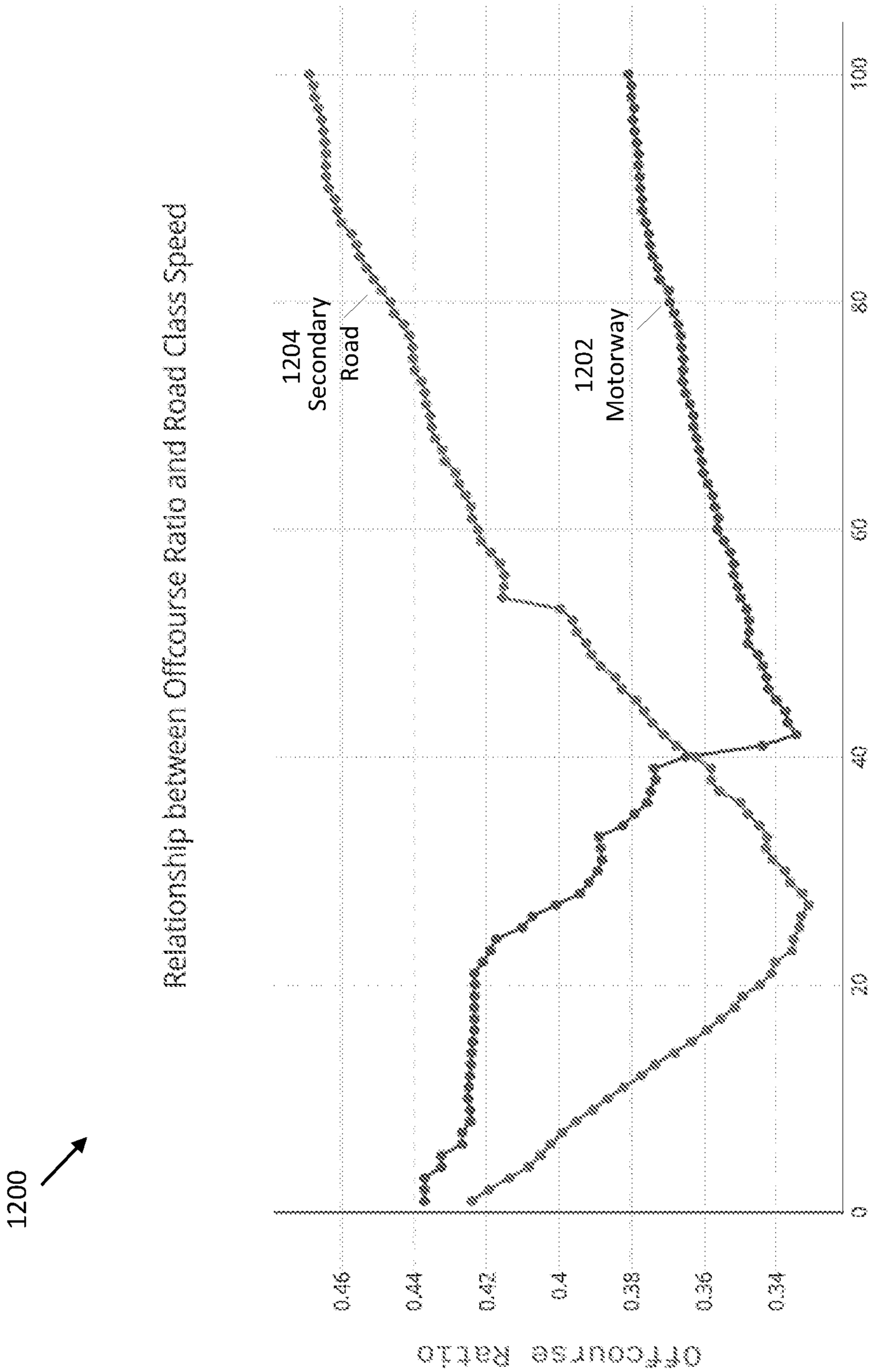

1200
Relationship between Offcourse Ratio and Road Class Speed
Offcourse Ratio
0.46
0.44
0.42
0.4
0.38
0.36
0.34
0
20
40
60
80
100
Road Class Speed (km/h)
1202
Motorway
1204
Secondary
Road

Figure 12

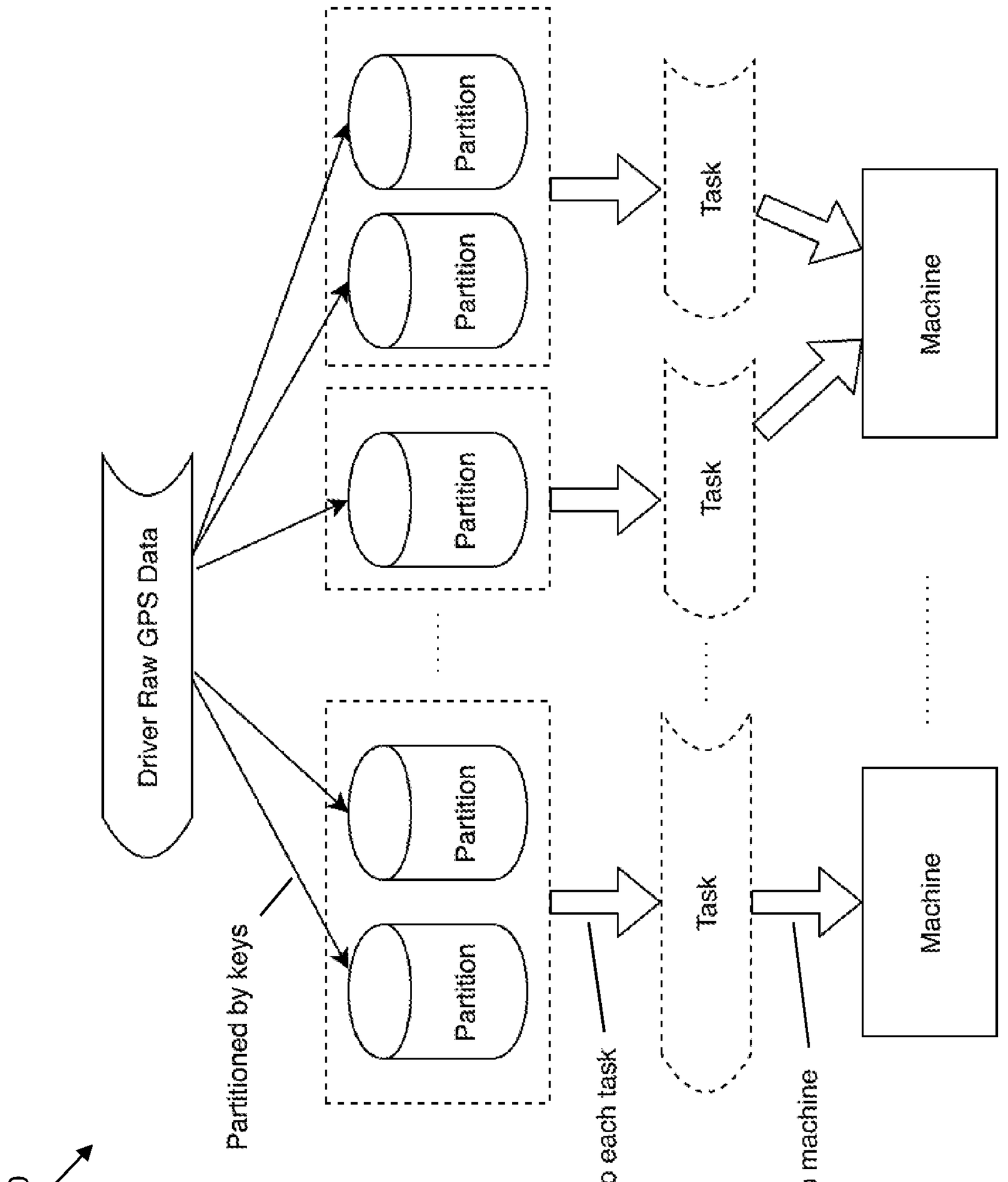

1300
Driver Raw GPS Data
Partitioned by keys
Partition
Partition
Partition
Partition
Partition
Aggregated to each task
Task
Task
Task
Allocated to machine
Machine
Machine

Figure 13

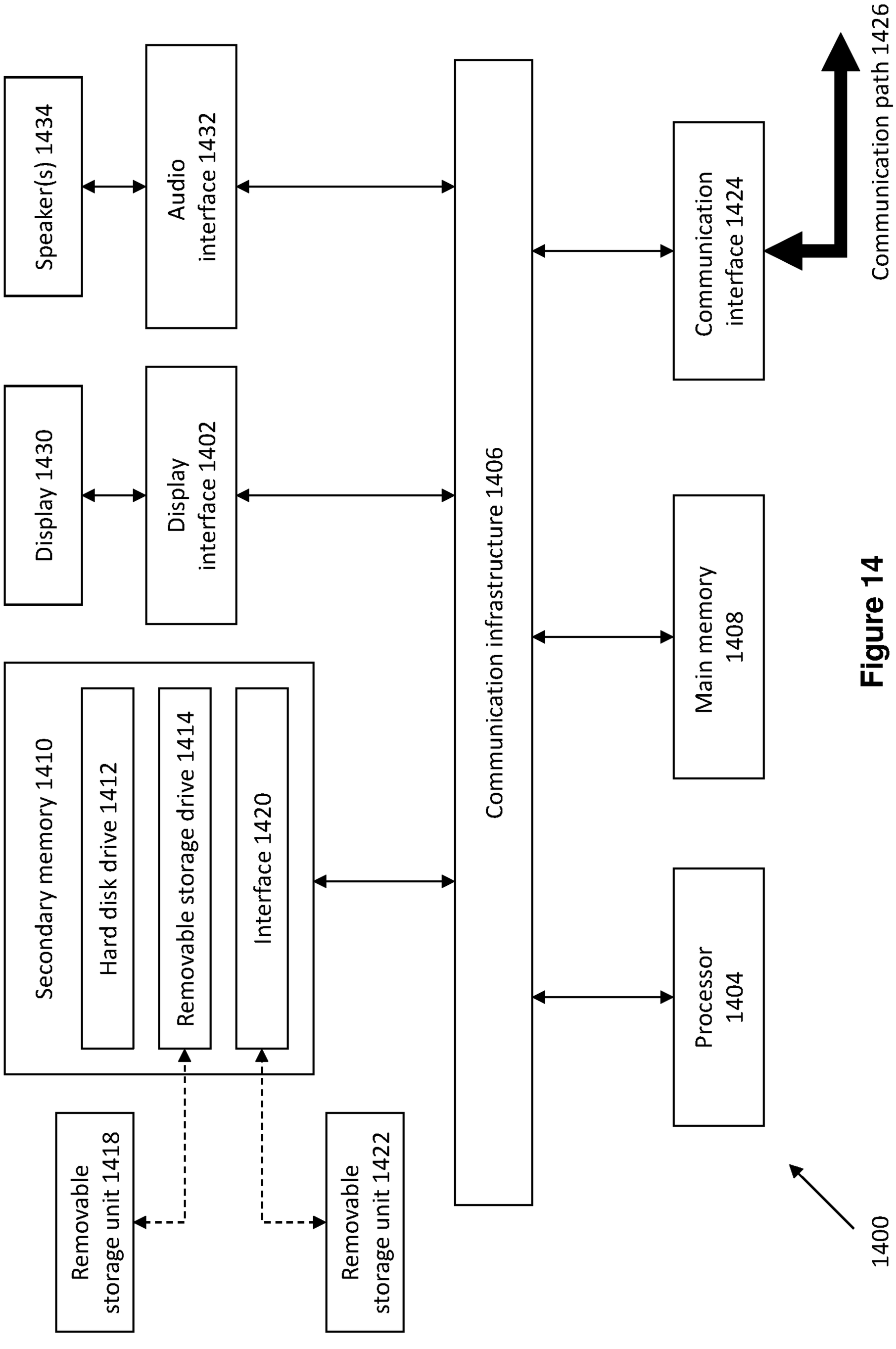

Speaker(s) 1434
Audio interface 1432
Display 1430
Display interface 1402
Secondary memory 1410
Hard disk drive 1412
Removable storage drive 1414
Interface 1420
Removable storage unit 1418
Removable storage unit 1422
Communication infrastructure 1406
Communication interface 1424
Communication path 1426
Main memory 1408
Processor 1404
1400

**Figure 14**

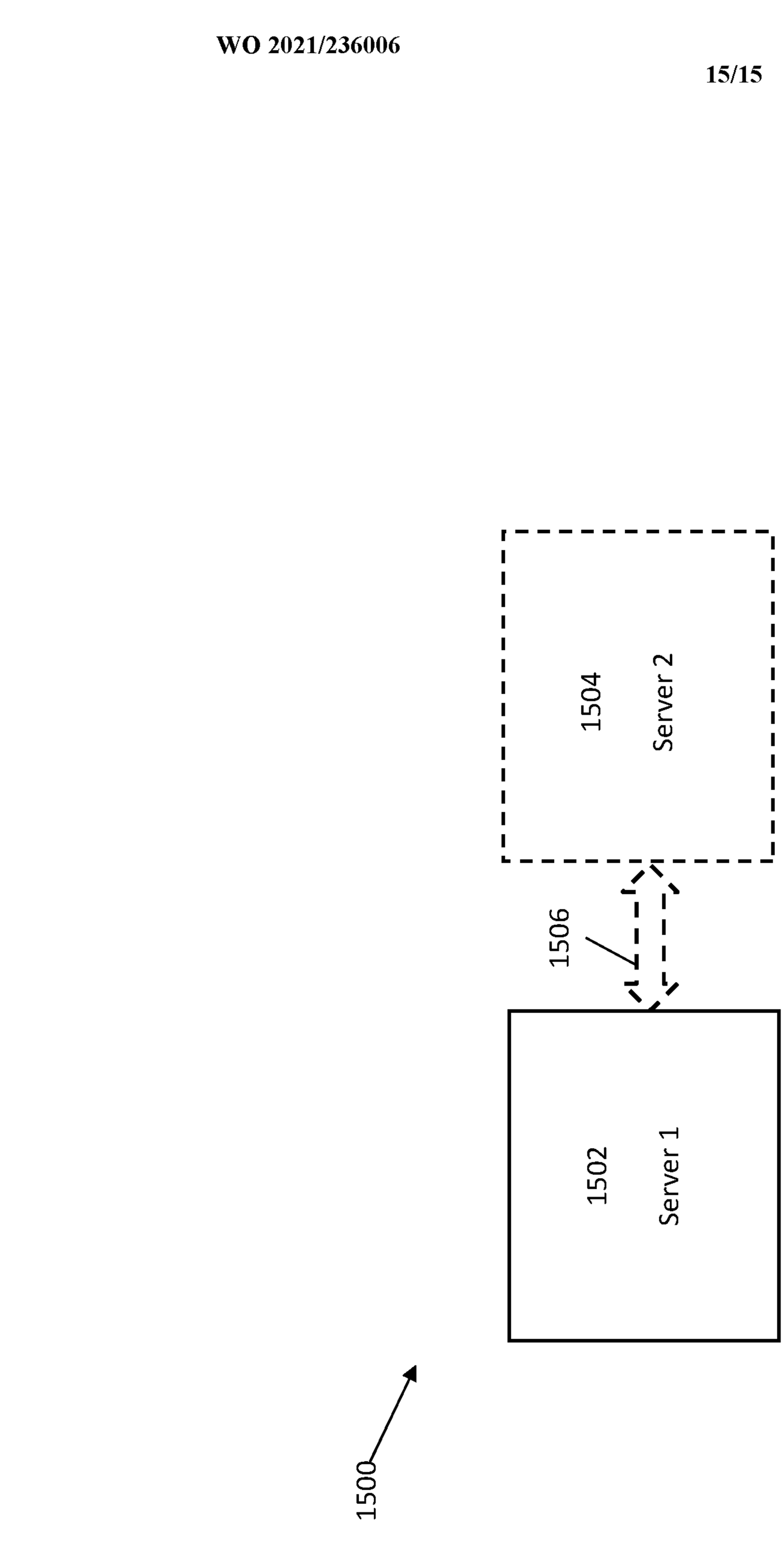



Figure 15