



(12) 发明专利

(10) 授权公告号 CN 102612709 B

(45) 授权公告日 2015. 09. 30

(21) 申请号 201080039465. 8

(22) 申请日 2010. 07. 09

(30) 优先权数据

61/213, 746 2009. 07. 09 US

(85) PCT国际申请进入国家阶段日

2012. 03. 06

(86) PCT国际申请的申请数据

PCT/EP2010/059944 2010. 07. 09

(87) PCT国际申请的公布数据

W02011/004026 EN 2011. 01. 13

(73) 专利权人 通腾科技股份有限公司

地址 荷兰阿姆斯特丹

专利权人 通腾发展德国公司

(72) 发明人 海科·席林 埃格尼·高里洛

莫里茨·希尔格

安德列亚斯·普洛福斯

于尔根·韦贝尔 西莫内·特尔图伦

(74) 专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 章蕾

(51) Int. Cl.

G08G 1/096(2006. 01)

(56) 对比文件

CN 1715835 A, 2006. 01. 04,

EP 1566611 A1, 2005. 08. 24,

CN 1991312 A, 2007. 07. 04,

CN 101246021 A, 2008. 08. 20,

US 2009/0005977 A1, 2009. 01. 01,

审查员 周瑜

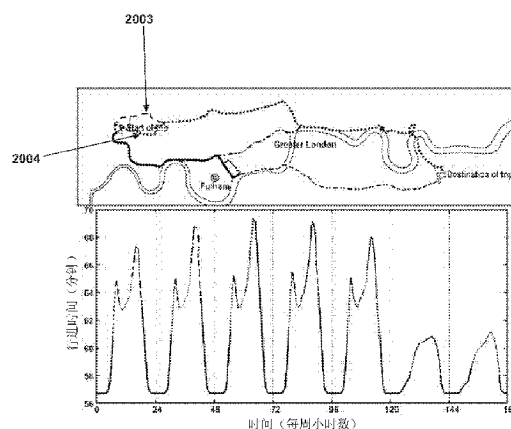
权利要求书2页 说明书58页 附图20页

(54) 发明名称

导航装置及用于依赖于时间的路线计算的方法

(57) 摘要

本发明涉及一种使用包含多个可导航路径的地图数据来确定路线的方法,所述地图数据被划分成多个区。所述方法包含使用至少一个处理设备以:接收所述地图数据上的起点及目的地以及行进时间,使用所述地图数据及识别所述地图数据的区之间的最小成本路径的最小成本数据来确定从所述起点到所述目的地的路线。在不同时间在所述区中的一对区之间存在不同最小成本路径的情况下,所述最小成本数据识别所述对区之间的一个以上最小成本路径,且确定路线包含从包含所述起点及所述目的地的所述对区的所述最小成本路径中识别按所述行进时间具有最低成本的最小成本路径。



1. 一种使用地图数据确定从起点到目的地的路线的成本概况的方法, 所述地图数据被划分成多个区, 所述地图数据包含:

多个可导航段, 其表示所述地图数据的可导航路径的段, 每一可导航段具有相关联的时变成本函数; 及

在所述地图数据的每个区中的用于所述可导航段的最小成本数据, 其识别对于每一个其它区可导航段是否为任一时间周期到所述其它区的最小成本路径的部分,

所述方法包含使用至少一个处理设备来:

搜索从起点到目的地的路线, 所述搜索包含确定连接到节点的一组可导航段中的一个或一个以上可导航段是否被所述最小成本数据识别为包含所述起点及所述目的地的区的最小成本路径的部分, 且在所述组的所述可导航段中的一个或一个以上可导航段被识别为最小成本路径的部分的情况下, 从所述组仅探索被识别为最小成本路径的部分的所述一个或一个以上可导航段; 且

根据所探索的所述可导航段的所述时变成本函数而随着时间的过去确定所述路线的成本概况, 所述成本概况表示在不同行进时间所述起点和所述目的地之间的最佳路线的成本;

其中确定随着时间的过去所述一个或一个以上路线的成本概况包含:

当第一及第二最小成本路径在节点处相交时, 将与所述第一最小成本路径相关联的第一成本概况同与所述第二最小成本路径相关联的第二成本概况组合以确定至少一个组合成本概况, 并进一步从所述节点传播所述至少一个组合成本概况。

2. 根据权利要求 1 所述的方法, 其中所述组合所述第一及第二成本概况的步骤包含叠加所述第一及第二成本概况以及确定最大及最小成本概况中的至少一者以供进一步传播。

3. 根据权利要求 1 所述的方法, 其中所述组合所述第一及第二成本概况的步骤包含确定平均成本概况以供进一步传播。

4. 根据任一前述权利要求所述的方法, 其中与可导航段相关联的所述时变成本函数包含识别在不同时间所述可导航段上的预期速度的速度概况数据。

5. 根据权利要求 1、2 或 3 所述的方法, 其中随着时间的过去的所述一个或一个以上路线的所述成本概况是响应于接收到成本概况请求而确定, 所述成本概况请求包括其中对路线的搜索受限的时段。

6. 根据权利要求 1、2 或 3 所述的方法, 其包含接收对按特定行进时间所述起点与所述目的地之间的路径的请求, 随着时间的过去的所述一个或一个以上路线的所述成本概况是针对包含所述特定行进时间的时间窗而确定, 所述方法进一步包含:

根据所述成本概况确定在所述时间窗内的另一行进时间处所述路线的成本是否小于针对所述特定行进时间的所述成本, 且如果所述成本较小, 那么显示消息, 所述消息向用户告知所述另一行进时间。

7. 根据权利要求 1、2 或 3 所述的方法, 其包含经由用户接口从用户接收对行进时间的选择, 且致使显示器显示根据按所接收到的所述行进时间的所述成本概况而确定的所述路线的图像。

8. 根据权利要求 7 所述的方法, 其包含使所述用户能够在检视所述路线的所述图像的同时改变所述行进时间, 且响应于所述行进时间的改变, 以根据针对所述改变的行进时间

的所述成本概况而确定的新路线的图像更新所述显示。

9. 根据权利要求 8 所述的方法, 其包含致使显示表示所述行进时间的滑动条, 且响应于用户与所述滑动条互动而以所述新路线的所述图像更新所述显示。

10. 一种包含存储器的计算机装置, 所述存储器中存储有地图数据, 所述地图数据划分为多个区, 所述地图数据包含:

多个可导航段, 其表示所述地图数据的可导航路径的段, 每一可导航段具有相关联的时变成本函数; 及

在所述地图数据的每个区中的用于所述可导航段的最小成本数据, 其对于每一个其它区识别可导航段是否为在任一时间周期到所述其它区的最小成本路径的部分, 及

处理设备, 其经布置:

搜索从起点到目的地的路线, 所述搜索包含确定连接到节点的一组可导航段中的一个或一个以上可导航段是否被所述最小成本数据识别为包含所述起点及所述目的地的区的最小成本路径的部分, 且在所述组的所述可导航段中的一个或一个以上可导航段被识别为最小成本路径的部分的情况下, 从所述组仅探索被识别为最小成本路径的部分的所述一个或一个以上可导航段; 及

根据所探索的所述可导航段的所述时变成本函数而确定随着时间的过去的所述一个或一个以上路线的成本概况, 所述成本概况表示在不同行进时间所述起点和所述目的地之间的最佳路线的成本;

其中确定随着时间的过去的所述一个或一个以上路线的成本概况包含:

当第一及第二最小成本路径在节点处相交时, 将与所述第一最小成本路径相关联的第一成本概况同与所述第二最小成本路径相关联的第二成本概况组合以确定至少一个组合成本概况, 并进一步从所述节点传播所述至少一个组合成本概况。

导航装置及用于依赖于时间的路线计算的方法

技术领域

[0001] 本发明涉及使用地图数据确定路线的计算机化方法及相关系统。明确地说（但非排它地），所述方法可用于导航装置中，用以促进路线规划。

背景技术

[0002] 已知路线规划算法（例如，迪科斯彻（Dijkstra）方法、A*方法或穆尔 / 裴帕（Moore/Pape）方法）。然而，这些路线规划算法可能有些慢。在 US 6 636 800 中展示了如何可增加此路线规划的速度实例，所述案的教导特此以引用的方式并入本文中。US6 636 800 教导通过将地图划分成多个区且确定到所述地图的每一区的最短路径（最小成本路径）的树而预处理地图数据。将此经预处理的数据存储在存储器中，且在路线的计算期间使用此经预处理的数据。

[0003] 更具体地说，当起点位于一个区内且目的地位于另一区内时，可将预先确定的最短路径的树用以迅速确定将用作路线的部分的在所述区之间的路径。此方法减少了确定路线所需的处理的等级，因为其减少了当用户请求确定路线时所探索的路径的数目。

[0004] 将理解，“最短”路径未必表示最短距离的路径，而是表示具有最小成本的路径。路径的成本将取决于如指定于成本公式中的所考虑的因素，且可考虑例如最快速路线及燃料使用等因素。

[0005] 当区之间的最短路径保持静态时，此方法为极成功的。然而，交通条件随着时间的过去的改变可改变为区之间的最短路径的路径，使得经预处理的树不再识别区之间的真正最短路径。这可导致针对指定准则来说并非最佳路线的路线被确定。

发明内容

[0006] 根据本发明的第一方面，提供一种使用包含多个可导航路径的地图数据确定路线的方法，所述地图数据被划分成多个区，所述方法包含使用至少一个处理设备以：接收所述地图数据上的起点及目的地以及行进时间，使用所述地图数据及识别所述地图数据的区之间的最小成本路径的最小成本数据确定从所述起点到所述目的地的路线，所述方法的特征在于，在不同时间在所述区中的一对区之间存在不同最小成本路径的情况下，所述最小成本数据识别所述对区之间的一个以上最小成本路径，且确定路线包含从包含所述起点及所述目的地的所述对区的所述最小成本路径中识别在所述进行时间具有最低成本的所述最小成本路径。

[0007] 本发明可在不同时间在区中的一对区之间存在不同最小成本路径的情况下，相对于使用不识别在所述对区之间的一个以上最小成本路径的数据而增加识别按行进时间所述区之间的最小成本路径的可能性。然而，使用最小成本数据使得能够比在使用从头开始确定最小成本路径的方法时更迅速地确定路线。举例来说，交通条件的可预测改变（例如，规则的高峰时间）可变更区之间的最小成本路径。预先计算在不同时间在区之间的最小成本路径允许考虑到此些周期性改变。

[0008] 确定所述路线可包含独立于时间根据所述最小成本数据识别一对区之间的最小成本路径,且针对从行进时间得出的一个或一个以上相关时间而对所述经识别的最小成本路径实行成本分析以确定按所述行进时间的最小成本路径。这可允许最小成本数据具有小得多的大小,因为关于最小成本路径的数据不需要交叉参考所述路径为最小成本路径的时间。例如导航装置等路线规划器可确定由最小成本数据识别的路径子集中的哪一路径为在相关时间时的最小成本路径。此布置确实将小的额外处理负担转移给路线规划器,但是作为回报,大量节省了最小成本数据的大小。

[0009] 然而,在替代布置中,所述最小成本数据针对每一最小成本路径识别所述路径为所述最小成本路径的参考时间,且确定路线包含根据所述最小成本数据选择具有对应于从所述行进时间得出的一个或一个以上相关时间的参考时间的最小成本路径。此布置具有优点,因为将必须探索较少潜在路线以找到最小成本路线。

[0010] 相关时间可为到达由最小成本路径连接的节点的预测时间。节点为地图上的可导航路径发生交叉/分叉的点或可导航路径的终点。因此,经由基于成本的分析而识别或经选择以用于路线中的最小成本路径将不会全部针对单个时间,而是将针对不同时间(视到达沿着路线的节点的时间而定)。举例来说,如果预测出花费一小时来行进过一路线,那么将基于到达沿着所述路线的节点的时间而识别/选择针对所述小时中的时间的最小成本路径。此外,分析/选择最小成本数据及确定路线可联合进行,因为仅可在已确定路线的一部分的情况下选择最小成本数据。

[0011] 行进时间可为从起点的出发时间、到达目的地的时间,或到达沿着路线的特定所在地(例如,节点)的时间。

[0012] 已发现,为了实现所要等级的精确性,最小成本数据的时间分辨率应具有秒的数量级,优选具有十分之一秒的数量级,且最优选具有百分之一秒的数量级。已发现,使用具有较粗略等级的时间样本的最小成本数据导致不可接受的精确性等级。此外,在以十分之一秒或百分之一秒为基数的范围内处理可相对于以 60 秒为基数的处理简化计算。可能需要避免比百分之一秒精细的等级,因为这可能需要较大数目个位来存储数据。应理解,可将百分之一秒的时间分辨率存储为 32 位串,而较高分辨率可能需要 64 位串。

[0013] 路径选择算法可为基于成本公式将成本指派给不同路径的路径选择算法。地图数据可包含识别在不同时间在路径上的预期速度的速度概况数据。可将所述地图数据的多条路径划分成多个可导航段,且速度概况数据可识别每一可导航段的速度概况。可至少部分地根据构成路径的可导航段的速度概况确定所述路径的成本。

[0014] 在一个实施例中,所述方法包含根据速度概况数据确定针对相关时间的速度,其中根据所考虑的路线中的每一者的所确定的速度来计算成本。

[0015] 在替代实施例中,所述方法包含根据正考虑的路线的可导航段的速度概况的多个值(而非单个值)来确定成本概况。换句话说,成本公式接收概况(即,函数)作为输入,且输出表示在不同时间路线的成本的概况。以此方式,可确定随着时间的过去的路线的成本的变化。

[0016] 所述方法可包含确定针对不同行进时间在共同起点与目的地之间的多条路线。举例来说,所述多条路线可用以比较不同行进时间如何影响路线及/或用以沿着在起点与目的地之间的路线行进的预测时间。举例来说,在高峰时间期间干线公路可能并非区之间的

最小成本路径,但在高峰时间外可变成最小成本路径。这可导致针对不同行进时间的在起点与目的地之间的不同路线及行进的时间。确定多条路线可允许用户基于对所确定的路线的比较而就行进时间作出决策。

[0017] 可将多条路线的确定与成本概况的产生组合以产生所述多条路线的从起点到目的地的成本概况。

[0018] 所述方法可包含接收概况请求以确定针对所定义准则在起点与目的地之间的最佳行进时间,且作为响应,确定多条路线的成本概况(每一路线针对一不同行进时间),且将结果显示给用户。

[0019] 结果的显示可为最佳路线的行进时间的显示。因此,所述方法可包含根据所述成本概况(例如,通过比较成本概况的值而筛选结果)来确定针对所定义的准则所述多条路线中的哪一路线为最佳路线。以此方式,处理设备为用户确定最佳路线,且不需要由用户进行分析。

[0020] 或者,结果的显示可为成本概况的显示(例如,图)。以此方式,用户可基于对所定义准则的评估以及选择最适合他/她的时间来选择路线。举例来说,在起点与目的地之间行进的最快速时间可能为在晚上,但用户可选择次最佳的路线,因为其在更方便的行进时间(例如,白天期间的某时间)。通过显示所述多条路线中的全部路线或至少一条以上路线的结果,用户可就行进时间及此如何影响路线作出明达的决策。

[0021] 在一个实施例中,概况请求包括对路线的搜索受限于其内的时段。举例来说,用户可指定行进时间应在两个时间之间(例如,在上午6点与上午9点之间),或在某一天(例如,在星期一),在某个日期(例如,7月1日),某个星期或某个月。或者,处理设备可自动搜索行进时间落在用户指定的行进时间的窗内(例如,用户行进时间 ± 30 分钟)的路线,且在发现具有不同行进时间的路线比具有用户指定的行进时间的路线具有较低成本时,向用户建议替代行进时间。

[0022] 在一个实施例中,所述方法包含经由用户接口从用户接收行进时间,且致使显示器显示所确定的路线的图像。所述方法可包含使得用户能够在检视展示所确定的路线的地图数据的图像的同时改变行进时间。举例来说,响应于行进时间的改变,可针对改变的指定行进时间确定新路线,且用所述新路线的图像来更新显示。或者,可已(例如,经由如上文所描述的成本概况的确定)预先确定针对其它时间的最小成本路线。可“实时”(例如,在毫秒的数量级)发生显示的更新,例如,小于1000毫秒且优选小于500毫秒。可经由使用用以识别区之间的最小成本路径的经预处理的最小成本数据而获得此些快速重新计算时间。可通过刷新整个屏幕、使旧路线淡出及使新路线淡入及/或其它适当动画来实行显示的更新。将理解,新路线可能与旧路线相同或不同于旧路线,取决于最小成本路径中是否存在改变。

[0023] 在一个实施例中,所述方法可包含致使显示表示行进时间的滑动条,且响应于用户与所述滑动条互动而确定新路线且以所述新路线的图像更新显示。所述方法可包含确定并显示新路线,使得对用户来说在所述用户改变行进时间之时与新路线的显示之时之间存在难以察觉的延迟。举例来说,在用户移动滑动条与显示新路线之间的延迟小于1秒且优选小于0.5秒。

[0024] 根据本发明的第二方面,提供一种上面存储有指令的数据载体,所述指令在由处

理设备执行时致使所述处理设备执行根据本发明的第一方面的方法。

[0025] 根据本发明的第三方面,提供一种数据载体,其上面存储有:包含多个可导航路径的地图数据,所述地图数据被划分成多个区;及识别在不同时间在所述地图数据的所述区之间的最小成本路径的最小成本数据。

[0026] 此数据载体为有利的,因为其可用于根据本发明的第一方面的对路线的确定。

[0027] 根据本发明的第四方面,提供一种包含存储器及处理设备的计算机装置,所述存储器中存储有:包含多个可导航路径的地图数据,所述地图数据被划分成多个区,识别所述地图数据的所述区之间的最小成本路径的最小成本数据;所述处理设备经布置以接收所述地图数据上的起点及目的地及行进时间,且使用所述地图数据及所述最小成本数据来确定从所述起点到所述目的地的路线,所述计算机装置的特征在于,在不同时间在所述区中的一对区之间存在不同最小成本路径的情况下,所述最小成本数据识别所述对区之间的一个以上最小成本路径,且确定路线包含当考虑所述行进时间时根据所述最小成本数据识别包含所述起点及所述目的地的区之间的所述路线的最小成本路径。

[0028] 根据本发明的第五方面,提供一种使用地图数据、最小成本数据及速度概况确定路线的方法,所述地图数据包含表示所述地图数据的可导航路径的段的多个可导航段;所述最小成本数据识别所述地图数据的区之间的最小成本路径,在于不同时间在所述区中的一对区之间存在不同最小成本路径的情况下,所述最小成本数据识别所述对区之间的一个以上最小成本路径;所述可导航段的所述速度概况识别所述可导航段上的速度如何随着时间而变化,所述方法包含使用至少一个处理设备以:搜索所述地图数据上的从起点到目的地的一个或一个以上路线,所述搜索包含确定连接到节点的一组可导航段中的一个或一个以上可导航段是否被所述最小成本数据识别为包含所述起点及所述目的地的区的最小成本路径的部分,且在所述组中的所述可导航段中的一个或一个以上可导航段被识别为最小成本路径的部分的情况下,从所述组仅探索被识别为最小成本路径的部分的所述一个或一个以上可导航段,且根据所述所探索的可导航段的所述速度概况而确定随着时间的过去所述一个或一个以上路线的成本概况。

[0029] 所述成本概况允许用户及/或导航设备确定改变行进时间将如何变更所述路线的成本。以此方式,可基于成本概况选择行进时间。

[0030] 根据本发明的第六方面,提供一种上面存储有指令的数据载体,所述指令在由处理设备执行时致使所述处理设备执行根据本发明的第四方面的方法。

[0031] 根据本发明的第七方面,提供一种包含存储器及处理设备的计算机装置,所述存储器中存储有:地图数据,所述地图数据包含表示所述地图数据的可导航路径的段的多个可导航段;及所述可导航段的速度概况,所述速度概况识别在所述可导航段上的速度如何随着时间而变化,所述处理设备经布置以搜索所述地图数据上的从起点到目的地的一个或一个以上路线,所述搜索包含确定连接到节点的一组可导航段中的一个或一个以上可导航段是否被所述最小成本数据识别为包含所述起点及所述目的地的区的最小成本路径的部分,且在所述组的所述可导航段中的一个或一个以上可导航段被识别为最小成本路径的部分的情况下,从所述组仅探索被识别为最小成本路径的部分的所述一个或一个以上可导航段,且根据所述所探索的可导航段的所述速度概况而确定随着时间的过去所述一个或一个以上路线的成本概况。

[0032] 根据本发明的第八方面,提供一种导航装置,其包含定位系统、显示器、用户接口及根据本发明的第四方面的计算机装置,其中所述处理器经布置以经由所述用户接口上的输入而得出起点及目的地,且将确定的结果显示于所述显示器上。所述处理器可根据所述用户接口上的输入、根据如由内部时钟指示的当前时间而确定行进时间,或可确定具有不同行进时间的多条路线。

[0033] 根据本发明的第九方面,提供一种使用地图数据确定并显示路线的方法,所述地图数据包含表示所述地图数据的可导航路径的段的多个可导航段及所述可导航段的速度概况,所述方法包含:响应于用户接口上的输入而确定沿着所述可导航段的路线;根据相关联的速度概况而确定沿着所述路线的可导航段的预期速度;及显示所述所确定的路线以及对在所述路线的所述可导航段中的至少一些可导航段上的所述预期速度的指示。

[0034] 根据本发明的第十方面,提供一种上面存储有指令的数据载体,所述指令在由处理器执行时致使所述处理器实行本发明的第五方面的方法。

[0035] 根据本发明的第十一方面,提供一种计算机装置,其包含显示器、用户接口、存储器及处理设备,所述存储器中存储有地图数据,所述地图数据包含表示所述地图数据的可导航路径的段的多个可导航段及所述可导航段的速度概况,所述速度概况表示在不同时间沿着所述段的预测行进速度,所述处理设备经布置以响应于所述用户接口上的输入而确定沿着所述可导航段的路线,根据相关联的速度概况而确定沿着所述路线的可导航段的预期速度,且致使所述显示器显示所述所确定的路线以及对在所述路线的所述可导航段中的至少一些可导航段上的预期速度的指示。

[0036] 所述处理设备可致使所述显示器显示行进时间,用户可经由与所述用户接口互动而改变所述行进时间,其中响应于对所述行进时间的改变,所述处理器确定针对新的行进时间的沿着所述路线的可导航段的经更新的预期速度,且更新所述路线的显示以指示所述经更新的预期速度。

[0037] 行进时间可由显示器上的滑动条指示,用户可移动所述滑动条以调整行进时间。

[0038] 预期速度的指示可为可导航段的色彩译码。举例来说,如果预期速度高于第一阈值(例如,在可导航段的速度限制的20%内),那么可将可导航段着色为第一颜色(例如,绿色),且如果预期速度低于第二阈值,那么可将可导航段着色为第二颜色(例如,红色)。第一及第二阈值可为同一值。

[0039] 以此方式,用户可确定拥塞点可能在路线上的何处,且(任选地)变更行进时间对这些拥塞点的影响。

附图说明

[0040] 现在接着是仅举例来说的参看附图对本发明的实施例的详细描述,其中

[0041] 图1为可由导航装置使用的全球定位系统(GPS)的示范性部分的示意性说明;

[0042] 图2为跨越通信信道而通信的服务器与导航装置的示意图;

[0043] 图3a为导航装置的示意图;

[0044] 图3b为由图3a的导航装置使用的架构堆叠的示意性表示;

[0045] 图4为导航装置的透视图;

[0046] 图5展示如由本发明的实施例产生的实例地图的一部分;

- [0047] 图 6 展示图 5 的地图,其上展示了用于路径选择的节点 ;
- [0048] 图 7 展示在处理之后的图 5 的地图 ;
- [0049] 图 8 展示在一些实施例中节点如何被分配到图 7 的地图的实例 ;
- [0050] 图 9 说明地图如何可被分割成多个巢式区的实例 ;
- [0051] 图 9a 说明图 9 的分割的增强 ;
- [0052] 图 10 展示由本发明的实施例利用的位向量的实例集合 ;
- [0053] 图 10a 说明如何在网络的迪科斯彻探索期间利用时间分析 (time profiled) 的信息 ;
- [0054] 图 11 展示本发明的实施例的文件的实例文件格式 ;
- [0055] 图 12 展示可如何编码图 11 的区再映射表及区 ID 列表的实施例 ;
- [0056] 图 13 展示如图 3 中所说明的最粗略巢式区的实例文件格式 ;
- [0057] 图 14 展示除了最粗略巢式区之外的巢式区的实例文件格式 ;
- [0058] 图 15 示范编码方案内的位的聚结 ;
- [0059] 图 16 示范聚结位的效应 ;
- [0060] 图 17 也示范聚结位的效应 ;
- [0061] 图 18 展示使用现有技术路径选择技术来醒目提示所考虑的路线的地图 ;
- [0062] 图 19 展示通过本发明的实施例来醒目提示所考虑的路线的地图 ;
- [0063] 图 20 展示 A*搜索方法 (现有技术) 的实例 ;
- [0064] 图 21 展示由本发明的至少一些实施例使用的对图 20 的修改 ;
- [0065] 图 22 展示概述本方法的步骤的流程图 ;
- [0066] 图 23 展示说明路线的可导航段上的预期速度的导航装置的显示器 ;
- [0067] 图 24 展示说明在起点与目的地之间的一个或一个以上路线的成本概况的导航装置的显示器 ;
- [0068] 图 25 展示说明在起点与目的地之间的一个或一个以上路线的成本概况的导航装置的显示器 ;
- [0069] 图 26 展示说明针对特定行进时间的在起点与目的地之间的路线及用于改变行进时间的控件的导航装置的显示器 ;
- [0070] 图 27 展示在用户已变更行进时间之后对图 26 的显示的更新 ;
- [0071] 图 28 展示在用户已进一步变更行进时间之后对图 27 的显示的更新 ;
- [0072] 图 29 展示导航装置的一系列显示,其中向用户建议行进时间 ;
- [0073] 图 30 展示说明在交叉节点处两个竞争路线的成本概况的曲线图 ;及
- [0074] 图 31 展示从图 30 中所展示的成本概况得出的上界及下界成本概况。

具体实施方式

[0075] 贯穿以下描述,相同参考数字将用以识别相同部分。贯穿对各个实施例的描述,对参考标号处于 19xx 系列中的图 22 的流程图进行了参考。

[0076] 图 1 示意性地展示全球定位系统 (GPS),全球定位系统 (GPS) 为基于卫星-无线电的导航系统,其可用以确定无限数目个用户的连续位置、速度、时间,及 (在一些情况下) 方向信息。过去被称为 NAVSTAR 的 GPS 并入有在极其精确的轨道中绕地球运转的多个卫星。

基于这些精确轨道, GPS 卫星可将其所在地作为 GPS 数据中继到任何数目个接收单元。然而, 将理解, 可使用例如 GLOSNASS、欧洲伽利略 (Galileo) 定位系统、COMPASS 定位系统或 IRNSS (印度区域导航卫星系统) 等全球定位系统。

[0077] 当经专门配备以接收 GPS 数据的装置开始扫描无线电频率以寻找 GPS 卫星信号时, 实施 GPS 系统。在从 GPS 卫星接收到无线电信号后, 所述装置即刻经由多种不同常规方法中的一者来确定所述卫星的精确所在地。在多数情况下, 所述装置将继续扫描以寻找信号, 直到其已获得至少三个不同的卫星信号 (注意, 可使用其它三角测量技术通过仅两个信号来确定位置, 虽然这并非常例)。在实施几何三角测量的情况下, 接收器利用三个已知的位置来确定其自己相对于所述卫星的二维位置。可以已知方式进行此确定。另外, 获得第四卫星信号允许接收装置以已知方式通过相同几何计算来计算其三维位置。无限数目个用户可在连续基础上实时更新位置及速度数据。

[0078] 如图 1 中所展示, GPS 系统 100 包含围绕地球 104 运转的多个卫星 102。GPS 接收器 106 从所述多个卫星 102 中的若干卫星接收作为扩频 GPS 卫星数据信号 108 的 GPS 数据。从每一卫星 102 连续地发射扩频数据信号 108, 所发射的扩频数据信号 108 各自包含数据流, 所述数据流包括识别所述数据流所源于的特定卫星 102 的信息。GPS 接收器 106 通常需要来自至少三个卫星 102 的扩频数据信号 108, 以便能够计算二维位置。对第四扩频数据信号的接收使 GPS 接收器 106 能够使用已知技术计算三维位置。

[0079] 因此, GPS 系统允许具有 GPS 接收器 106 的装置的用户将其在地球上的位置确定到几米范围内。为了利用此信息, 常见做法是依赖于允许在其上展示用户的位置的电子地图。例如 TeleAtlas (<http://www.teleatlas.com>) 等提供商示范此些地图。

[0080] 此些电子地图不仅允许使用 GPS 系统 (或通过其它方式) 将用户的位置展示于电子地图上, 而且其允许用户规划行程路线等 (路线选择目的)。为了进行此路线规划, 由导航装置来处理电子地图, 所述导航装置可由通用计算装置提供。

[0081] 导航装置的特定实例包括卫星导航装置 (Sat. Nav), 其方便地称作便携式导航装置 (PND)。然而, 应记住, 本发明的教导不限于 PND, 而是可改为普遍适用于经配置以处理电子地图 (通常) 以便提供路线规划及导航功能性的任何类型的处理装置。因此, 由此可见, 在本申请案的背景下, 导航装置意在包括 (但不限于) 任何类型的路线规划及导航装置, 无论所述装置是体现为 PND、例如汽车等交通工具、便携式计算资源 (例如, 执行路线规划及导航软件的便携式个人计算机 (PC)、移动电话或个人数字助理 (PDA)) 或服务器, 或是体现为跨越网络提供此功能性的其它计算装置。

[0082] 在图 3a 及 4b 中展示所述呈 PND 的形式的导航装置的实例, 且应注意, 导航装置 200 的框图不包括所述导航装置的所有组件, 而是仅为许多实例组件的代表。导航装置 200 位于外壳 (未图示) 内。导航装置 200 包括处理电路, 所述处理电路包含 (例如) 上文所提到的处理器 202, 所述处理器 202 耦合到输入装置 204 及显示装置 (例如, 显示屏 206)。虽然此处以单数形式提到输入装置 204, 但所属领域的技术人员应了解, 输入装置 204 表示任何数目个输入装置, 包括键盘装置、话音输入装置、触摸面板及 / 或用以输入信息的任何其它已知输入装置。同样, 显示屏 206 可包括例如液晶显示器 (LCD) 等任何类型的显示屏。

[0083] 在导航装置 200 中, 处理器 202 经由连接 210 而操作性地连接到输入装置 204 且能够经由连接 210 从输入装置 204 接收输入信息, 且经由相应的输出连接 212 而操作性地

连接到显示屏 206 及输出装置 208 中的至少一者,以将信息输出到所述至少一者。导航装置 200 可包括输出装置 208,例如,可听输出装置(例如,扬声器)。在输出装置 208 可为导航装置 200 的用户产生可听信息的同时,应同等地理解,输入装置 204 还可包括用于接收输入话音命令的麦克风及软件。另外,导航装置 200 还可包括任何额外输入装置 204 及/或任何额外输出装置,例如,音频输入/输出装置。

[0084] 处理器 202 经由连接 216 而操作性地连接到存储器 214,且进一步适于经由连接 220 从输入/输出(I/O)端口 218 接收信息/将信息发送到输入/输出(I/O)端口 218,其中 I/O 端口 218 可连接到在导航装置 200 外部的 I/O 装置 222。外部 I/O 装置 222 可包括(但不限于)外部收听装置,例如,听筒(earpiece)。到 I/O 装置 222 的连接可另外为到任何其它外部装置(例如汽车立体声单元)的有线或无线连接,以用于(例如)免提操作及/或用于话音激活的操作、用于连接到听筒或头戴式耳机及/或用于连接到(例如)移动电话,其中移动电话连接可用以建立在导航装置 200 与(例如)因特网或任何其它网络之间的数据连接,及/或用以建立经由(例如)因特网或某一其它网络到服务器的连接。

[0085] 导航装置 200 的存储器 214 包含非易失性存储器的一部分(例如以存储程序代码)及易失性存储器的一部分(例如以在所述程序代码被执行时存储数据)。所述导航装置还包含端口 228,所述端口 228 经由连接 230 而与处理器 202 通信,以允许可装卸存储卡(通常被称作卡)被添加到装置 200。在所描述的实施例中,所述端口经布置以允许添加 SD(安全数字)卡。在其它实施例中,所述端口可允许连接其它格式的存储器(例如,压缩快闪(CF)卡、Memory Stick™、xD 存储卡、USB(通用串行总线)快闪驱动器、MMC(多媒体)卡、智能媒体卡、微驱动器等)。

[0086] 图 2 进一步说明经由连接 226 的在处理器 202 与天线/接收器 224 之间的操作性连接,其中天线/接收器 224 可为(例如)GPS 天线/接收器,且因而将充当图 1 的 GPS 接收器 106。应理解,为了说明而示意性地组合由参考数字 224 表示的天线及接收器,但天线及接收器可为单独定位的组件,且天线可为(例如)GPS 片状天线或螺旋天线。

[0087] 另外,图 3 及 4 的便携式或手持型导航装置 200 可以已知方式连接或“对接”到例如自行车、摩托车、汽车或船等交通工具。可接着将此导航装置 200 从对接所在地移除以用于便携式或手持型导航用途。实际上,在其它实施例中,装置 200 可布置为手持型的以允许用户的导航。

[0088] 参看图 3a,导航装置 200 可为包括集成输入及显示装置 206 及图 3a 的其它组件(包括但不限于,内部 GPS 接收器 224、处理器 202、电力供应器(未图示)、存储器系统 214 等)的单元。

[0089] 导航装置 200 可位于臂状物 252 上,可使用吸盘 254 将臂状物 252 本身紧固到交通工具仪表板/窗/等等。此臂状物 252 为导航装置 200 可对接到的对接台的一个实例。可通过将导航装置 200 搭扣连接到臂状物 252 而将导航装置 200 对接或以其它方式连接到对接台的臂状物 252。导航装置 200 可接着可在臂状物 252 上旋转。为了释放导航装置 200 与对接台之间的连接,可(例如)按压导航装置 200 上的按钮(未图示)。用于将导航装置 200 耦合到对接台及将导航装置 200 从对接台去耦的其它同等合适的布置是所属领域的技术人员众所周知的。

[0090] 转到图 3b,处理器 202 与存储器 214 合作以支持 BIOS(基本输入/输出系统)282,

其充当导航装置 200 的功能硬件组件 280 与由所述装置执行的软件之间的接口。处理器 202 接着从存储器 214 加载操作系统 284, 所述操作系统 284 提供应用程序软件 286 (其实实施所描述的路线规划及导航功能性中的一些或全部) 可运行的环境。应用程序软件 286 提供操作环境, 所述操作环境包括支持导航装置的核心功能 (例如, 地图检视、路线规划、导航功能及与此相关联的任何其它功能) 的图形用户接口 (GUI)。就此来说, 应用程序软件 286 的部分包含视图产生模块 288。

[0091] 在正描述的实施例中, 导航装置的处理器 202 经编程以接收由天线 224 接收到的 GPS 数据, 且不时地将所述 GPS 数据连同表示 GPS 数据被接收的时间的时戳一起存储在存储器 214 内以积累导航装置的一系列位置。经如此存储的每一数据记录可被认为是 GPS 定位 ; 即, 其为导航装置的所在地的定位且包含纬度、经度、时戳及精确性报告。此系列位置可被认为是定位数据。

[0092] 在一个实施例中, 大体上周期性地 (例如, 每 5 秒) 存储数据。所属领域的技术人员将了解, 其它周期将为可能的, 且在数据分辨率与存储器容量之间存在平衡 ; 即, 随着通过取得更多样本而使数据的分辨率增加, 需要更多存储器来保持数据。然而, 在其它实施例中, 分辨率可为大体上每 : 1 秒、10 秒、15 秒、20 秒、30 秒、45 秒、1 分钟、2.5 分钟 (或实际上, 这些周期之间的任何周期)。因此, 在装置的存储器内, 积累了装置 200 在各时间点处的行踪的记录。

[0093] 在一些实施例中, 可发现, 所捕捉的数据的质量随着周期增加而减小, 且同时降级的程度将至少部分地视导航装置 200 移动的速度而定 (约略 15 秒的周期可提供合适上限)。

[0094] 另外, 处理器 202 布置成不时地将装置 200 的行踪的记录 (即, GPS 数据及时戳) 上载到服务器。在导航装置 200 具有将其连接到服务器的永久的或至少通常存在的通信信道的一些实施例中, 周期性地发生数据的上载 (其可为 (例如) 每 24 小时一次)。所属领域的技术人员将了解, 其它周期为可能的, 且可大体上为以下周期中的任一者 : 15 分钟、30 分钟、每小时、每 2 小时、每 5 小时、每 12 小时、每 2 天、每周或在这些周期之间的任何时间。实际上, 在此些实施例中, 处理器 202 可经布置以大体上实时地上载行踪的记录, 但这可能不可避免地意味着事实上数据被不时地发射而在发射之间具有相对较短的周期, 且因而数据可被较为正确地认为是伪实时的。在此些伪实时实施例中, 导航装置可经布置以在存储器 214 内及 / 或插入于端口 228 中的卡上缓冲 GPS 定位, 且当已存储预定数目时发射这些 GPS 定位。此预定数目可为大约 20、36、100、200, 或这些数目之间的任何数目。所属领域的技术人员将了解, 所述预定数目部分地由存储器 214 / 端口 228 内的卡的大小控管。

[0095] 在不具有通常存在的通信信道的其它实施例中, 处理器可经布置以在通信信道被创建时将记录上载到服务器。这可 (例如) 在导航装置 200 连接到用户的计算机时发生。并且, 在此些实施例中, 导航装置可经布置以在存储器 214 内或插入于端口 228 中的卡上缓冲 GPS 定位。如果存储器 214 或插入于端口 228 中的卡变得充满 GPS 定位, 那么导航装置可经布置以删除最旧的 GPS 定位, 且因而, 其可被认为是先入先出 (FIFO) 缓冲器。

[0096] 在正描述的实施例中, 行踪的记录包含一个或一个以上迹线, 其中每一迹线表示导航装置 200 在 24 小时周期内的移动。每一 24 小时周期经布置成与日历日一致, 但在其它实施例中, 无需为此情况。

[0097] 服务器 150 经布置以接收装置的行踪的记录,且将此存储在大容量数据存储装置内以供处理。因此,随着时间过去,大容量数据存储装置累积已上载了数据的导航装置 200 的行踪的多个记录。一般来说,服务器将收集多个导航装置 200 的行踪。

[0098] 服务器可接着经布置以根据所述装置或每一装置的所收集的行踪而产生速度数据。此速度数据可包含时变速度概况,其展示沿着地图的可导航段的平均速度如何随着时间变化。

[0099] 图 7 展示电子地图的实例,在所描述的实施例中,跨越网络(在此情况下为因特网)在导航装置上检视所述电子地图。在本文中描述的实施例中的任一者中的第一步骤为产生此电子地图 1900。可在 <http://routes.tomtom.com/?Lid=1#/map/?center=55.01%2C-3.445&zoom=4&map=basic> 检视所述地图。

[0100] 为了实现图 7 的将被用于路线选择目的的电子地图,所述电子地图具有一系列节点,所述电子地图的用户通常看不到所述节点。因而,节点等等可通常被称作地图数据。然而,为了便于理解,在图 6 中展示节点 300 中的一些节点。这些节点被设在可导航段(在此情况下为道路段)的交叉点、末端区等等处,且被用于路线选择的目的。当用户要求其导航装置规划在两个点或两个以上点之间的路线时,所述导航装置规划地图的相关节点之间的路线。因而,地图数据可用以根据由导航装置的用户设定的至少一个准则来产生路线。

[0101] 所属领域的技术人员将了解,电子地图还包含提供道路段的形状、建筑物的入口的所在地等等的其它节点。

[0102] 因此,每一节点 300 具有从其发出的、用户可沿其行进的至少单个道路段,且通常为多个道路段。举例来说,节点 302 具有四个此种道路段 304a、b、c、d。在其它实施例中,可导航段可表示并非道路的路径的段,例如,人行道、自行车道、沟渠、铁路、道路段、轨道等。然而,为方便起见,参考道路段。因此,如图 7 中所展示的地图向其用户展示多个道路段,所述多个道路段中的每一者表示在由所述地图覆盖的区域中的可导航路线的一部分。

[0103] 已知路线规划方法(例如,迪科斯彻(Dijkstra)方法、A*方法或穆尔/裴帕(Moore/Pape)方法)。然而,这些路线规划方法在计算时间方面可为极其慢的。在 US 6636 800 中展示了如何可增加此路线规划的速度实例,所述案的教导特此以引用的方式并入本文中。

[0104] 本文中所描述的实施例在预处理步骤中产生含有最小成本数据的所谓的侧文件(side file),所述最小成本数据由导航装置用以在处理电子地图时加速路线的产生。可将此信息以可被称为位向量的形式保持为二进制数据;即,0 与 1 的串。因而,所述侧文件还可被认为是地图数据,但可与电子地图一起或不与电子地图一起供应侧文件(例如,如图 7 中所展示)。因此,本发明的一些实施例可将地图数据提供为可与侧文件分离的地图,而本发明的其它实施例可组合地图与侧文件数据。

[0105] 然而,所属领域的技术人员将了解,如果使用侧文件,那么所述侧文件应与电子地图(针对其而产生所述侧文件)一起使用。如果并未执行此,那么可想象,将获得不正确路线(例如,未使其用户设定的成本公式最小化的路线)。

[0106] 并且,本发明的不同实施例可指定用于产生最小成本数据的不同参数(在此实施例中,为一系列位向量)。因而,如果使用所产生的地图数据的后续路线规划使用了与用以创建最小成本数据的所述参数不同的参数,那么所述最小成本数据不太可能对所述路线规

划有用。

[0107] 举例来说,一些实施例可产生用于乘汽车行进穿过地图的最小成本数据。如果后续路线规划系用以产生用于步行的路线,那么汽车特定最小成本数据不太可能为有用的。在另一实例中,一些实施例可产生在假定用户乐于沿着高速公路、收费道路等等行进的情况下的最小成本数据。在后续路线规划中,如果用户请求利用高速公路、收费道路等等的路线,那么所述最小成本数据不太可能为有用的。

[0108] 本发明的一些实施例可产生最小成本数据的多个集合,其各自具有预定准则的不同集合。举例来说,本发明的实施例可约略产生以下各项中的任一者:2、3、4、5、6、10、15,或更多集合的最小成本数据。

[0109] 因此,在一些实施例中且为了产生侧文件,在地图预处理步骤中,将节点划分到多个区中,且因而,任何地图均被划分成已知数目个区—例如,N个,且确定所述区之间的最小成本路径。此预处理产生可与地图一起利用的数据以便增加路线规划方法的速度。

[0110] 通常,在使用地图数据之前执行预处理,而不管所述地图数据将用于网站上还是用于例如 PND 等装置上。因而,预处理步骤常常被称作服务器侧进程。

[0111] 虽然任何通用计算装置均将适于执行预处理,但所属领域的技术人员将了解,装置的性能越高,预处理将执行得越快。通常,将 X86 架构计算装置用于预处理。此种 X86 架构装置将通常运行例如 Microsoft™ Windows™、UNIX、LINUX、OSX™等等的操作系统(OS)。然而,其它实施例可使用例如 RISC 架构等其它计算平台。

[0112] 并且,如别处所论述,可并行执行预处理,且因而,可在多个计算装置上或至少在多个处理器核心(其可为虚拟或真实处理器核心)上执行预处理。

[0113] 作为下一预处理步骤,处理一区内的每一道路段(例如,304a-d)以确定其是否为到所述地图内的所述数目(N)个区中的每一区的最小成本路径的部分,且产生位向量(最小成本评估)。因此,对于一区的每一道路段,位向量针对地图的每一区包含一位。即,位向量包含 N-1 个位(除在考虑中的区外,每一区 1 个),其经设定成 0 或 1,取决于所述道路段是否形成到由所述位表示的区的最小成本路径的部分。一些实施例可添加额外位以提供进一步信息,例如,标头、可见度区域、相邻区域等等。

[0114] 所属领域的技术人员将了解,在此意义上,可对照若干不同成本准则确定最小成本路径。举例来说,可对照以下准则中的任一者判断最小成本:最短距离;最短行进时间;花费最小(在环境影响方面);使用最少汽油;产生最少 CO₂,等等。在本实施例中,对照最短行进时间来判断最小成本。

[0115] 因此,所属领域的技术人员将了解,对于覆盖大量区域的地图来说,N很可能为较大的。因此,预处理可花费大量时间。

[0116] 因此,在(例如)覆盖西欧及中欧的实例地图中,通常可能存在 40,000,000 个节点,其具有 100,000,000 个道路段。假定在地图内存在 100 个区,那么 N 等于 100,且存在 100,000,000×100(N)个;即,需要 10×10⁹个位来存储所述地图的每一道路段的 99(即,N-1)个位。在使用下文所描述的本发明的实施例的情况下,此地图可能使用约略 500Mb 的存储装置。

[0117] W02009/053410 论述指派速度概况的方法,所述速度概况提供随着时间而变的沿着地图的可导航段的速度。此申请案的内容特此以引用的方式并入本文。因而且出于路线

选择的目的是,可导航段是否构成到另一区的最快速路线的部分将根据时间而变化。即,随着交通密度增加/减小(例如,在高峰时间等),使用可导航段分别将变得较不/较合意。

[0118] 一些实施例可针对每一道路段存储多个位向量。所属领域的技术人员将了解,记录速度的分辨率越高,所需位向量的数目越高。然而,其它实施例可利用时变函数来对最快速路线作出评估(非常像 W02009/053410 中所描述)。

[0119] 在跨越电子地图进行路线规划时,相信,有必要能够使用约略百分之一秒精确度的时间粒度进行路线规划;即,能够将从节点的出发时间指定到 0.01 秒的精确度以便正确地确定最低成本路线。

[0120] 因此,在考虑此精确性的等级时,将了解,上文所考虑的具有 40,000,000 个节点及 100,000,000 个道路段的地图具有穿过其中的 100,000,000²个可能的路线。当进一步考虑时间维度时,路线的数目进一步增加:7(即,每周天数)×24(即,每日小时数)×3600(即,每小时秒数)×100(即,每秒的百分之一秒数)×100,000,000²=604.800.000.000.000.000.000(千的七乘方)个可能的路线。在当前处理等级且在使用在每一时间间隔处探索每一段的单纯方法的情况下,将花费 19.178.082.191.780.821 年。申请人当前销售的西欧及中欧的地图具有 120,000,000 个道路段,藉此进一步增加此时间。

[0121] 因此,将了解,为了在预处理阶段中存储并处理数据,需要大量数据。

[0122] 因而,本发明的实施例使用技术以便显著减少预处理的量。现在更详细地描述这些技术,且本发明的一些实施例可能利用所有所述技术,而其它实施例可能仅利用所述技术中的一些技术。

[0123] 本发明的不同实施例可使用下文所概述的特征的不同组合。因而,一些实施例可利用所描述的全部技术以减少信息的量。在另一极端,其它实施例可能不利用所述技术当中任一者来减少数据。

[0124] 本发明的一些实施例不计算不符合预定准则的道路段的位向量-如步骤 1902 所展示。举例来说,如果不可能以预定运输类别沿着道路段行进,那么可能不计算所述道路段的位向量。

[0125] 因此,在将运输模式的准则设定成汽车的实例中,可能不计算以下道路段的位向量:

[0126] • 铁路;

[0127] • 存在于地图数据中的未对应于道路段的段;

[0128] • 错误(非法)方向上的单行道路;

[0129] • 不可由某运输形式通行的道路段(例如,在考虑汽车的驾驶性时的行人徒步区、人行道);

[0130] • 非决策点处的道路段(即,不能从道路段避开);

[0131] 网络的减小

[0132] 一种技术为减小经考虑以评估道路段是否形成最小成本路径的部分的网络的大小;减少道路段的数目会减小进行计算所花费的时间量,如步骤 1904 处所展示。

[0133] 为了说明此技术,还在图 6 中展示如图 7 中所展示的道路网络,但从图 4 删除了对于最小成本评估来说没有用的道路段。对网络的此减小旨在提供最小子网络,应对所述最小子网络进行最小成本评估。

[0134] 在数学术语中,被留下的且如图 7 中说明的网络可被描述为图 7 的道路网络的最大强组成部分的不定向表示的最大的双向连接的组成部分。通过将标准技术应用于两种强连接性(还称为双向连接性)同时适应于道路交通规则(例如,转弯限制及本地准入(local access))来确定核心网络。所属领域的技术人员将了解,强连接性指示在两个节点之间在两个方向(即,从 $A \rightarrow B$ 以及 $B \rightarrow A$)上行进为可能的。

[0135] 因此,参看图 5 与 7 两者,可见,已删除例如死胡同 250、冗余环路 252 等等的无出口的路。然而,将看见,在已被移除的道路段的起点处剩余有节点(例如,图 7 中的 400)。这是由于下文中所论述的原因。

[0136] 在减小到核心网络之后(例如,如图 7 中所展示),分割核心网络以提供上文所描述的区。在所描述的实施例中,根据多路弧分离器(multiway arc separator)来分割网络。所属领域的技术人员将了解,在此网络中,如果由区边界等分的道路段(即,弧)被移除,那么留在任一区中的道路段将不连接到任何其它区。

[0137] 分割网络 – 步骤 1906

[0138] 然而,在执行分割之前,进行以下步骤:

[0139] 1. 独立地确定并分割岛屿。岛屿趋向于与网络的其它部分物理上分离,且藉此通过例如轮渡链等等的方式而链接。这些以与典型道路段不同的方式表现(即,平均速度显著较低,等等),且如果此些链接被包括于分割中,那么所述方法不会有像预期一样好的性能。

[0140] 2. 收缩由网络中的一个以上节点表示的十字路口、环形道及其它接点,使得属于同一十字路口等等的节点不会落到不同区中。

[0141] 3. 收缩所有可导航段共享相同特性集合(例如,轮渡段、禁止驾驶、禁止通行等等)的简单路径(即,不具有转弯可能性的两个节点之间的连接)以减小传递到分割系统的网络的输入大小。举例来说,查看图 6,可将节点 308 折叠到道路 310 上。

[0142] 本发明的正描述的实施例使用分治(divide and conquer)方法,在 <http://labri.fr/perso/pelegrin/scotch> 处概述所述分治方法。本文中所概述的方法划分地图以包含若干个区,所述区并不基于地图的区域,而是基于区内所含有的节点的数目。具有经如此组织的节点具有以下优点:其可有助于使使用所述区的路线选择更有效率,因为其有助于递送具有类似本地搜索直径的区。

[0143] 每一节点具有与其相关联的包括区 ID 的身份编号,且因而,同一区中的节点具有相同区 ID 号。节点属于在 L 个层级式等级处的区。按照惯例,等级 0 为最粗略等级(用于长距离路线选择)。等级 L-1 为详细等级。

[0144] 方便地对节点的最大数目设定上限且允许低于此上限的某种灵活性,而不是设定任一区内含有的节点的绝对数目。举例来说,所述方法可指定每区最大节点数目为 100 个,这可导致每区通常有 60 个到 100 个节点的节点散布。上限被认为比固定数目个节点有利,因为固定数目可导致具有强制形状的区,这又导致在使用所述区时的次最佳的路线选择。

[0145] 多个等级

[0146] 在所描述的实施例中,执行分割以提供多个等级的巢式区,在图 9 中在概念上示范所述区。从图 9 将清楚,存在层级本质,其中将最低等级(0)再分以提供等级 1,又将等级 1 再分以提供等级 2。

[0147] 每一区中的节点的数目在等级之间变化,且与较低等级相比,较粗略等级(其趋向于覆盖较大地理区域)具有更多节点。因而,通常出于长程路线选择目的而使用正描述的实施例中的较粗略等级(例如,等级 0),而较精细等级(例如,等级 2)用于短程的路线选择。

[0148] 每一道路段的位向量的一部分包含针对所述等级中的每一者的位。位的数目为每一等级中的区的数目。

[0149] 为了提供要编码的数据量的概念(在典型实施例中,等级的数目将通常为 $L = 3$):

[0150] • 全局等级 = 0,将通常具有 100 个区。

[0151] • 中间等级 = 1,将通常针对等级 0 的每一区具有 10 个区(即, 100×10)。

[0152] • 最详细等级 = 2,将通常针对等级 1 的每一区具有 5 个区(即, $100 \times 10 \times 5$)。

[0153] 使用层级式等级为有利的,因为其减小(可能显著减小)所需的处理量。在正描述的本实施例中,存在 $100 \times 10 \times 5$ 个区(即, 5000 个区)。因此,在具有相同数目个区的平坦结构中,将要求本文中所概述的方法涉及 5000 个区。然而,在本文中所描述的实施例中,可通过参考处于适当等级中的节点来减少此数目。

[0154] 一旦已知地图的各个因素,等级的数目及每等级的区的数目将通常被调谐。举例来说,多少存储装置可被分配给地图,以及路线选择的应有执行速度。所属领域的技术人员将了解,存在折衷,因为随着预处理的量增加,那么地图大小变得较大以保持数据,但是可使用所述地图来计算较快速路线。

[0155] 在一个实施例中,存在 3 个等级的区。然而,在其它实施例中,可能存在任何数目个等级。举例来说,可能约略存在以下数目个等级中的任一者:1、2、4、5、6、7、8、9 或 10。

[0156] 因此,在使用上文的实例(通常具有 40,000,000 个节点及 100,000,000 个道路段的西欧及中欧的地图)的情况下,使用最单纯编码将针对每一等级的每个节点的区 ID 使用固定大小编码,且针对每一等级的每一道路段使用固定大小位向量(arp 旗标)。因此,可易于将此基本编码的大小计算如下:

[0157] • 等级 0 处的每一节点区 ID 将使用 $\log_2(100)$ 位 = 7 个位

[0158] • 等级 1 处的每一节点区 ID 将使用 $\log_2(10)$ 位 = 4 个位

[0159] • 等级 2 处的每一节点区 ID 将使用 $\log_2(5)$ 位 = 3 个位

[0160] • 等级 0 处的每一位向量(arp 旗标)将使用 100 个位(100 个区减去针对当前区的 1)

[0161] • 等级 1 处的每一位向量(arp 旗标)将使用 10 个位(10 个区减去针对当前区的 1)

[0162] • 等级 2 处的每一位向量(arp 旗标)将使用 5 个位(5 个区减去针对当前区的 1)

[0163] 图 9 展示最粗略的区等级 600,其在所述图中提供 6 个区 1 到 6。将这些区中的每一者再分成更多的区(在此实施例中,九个子区),如由粗略区等级 600 内的虚线道路段表示。在所描述的实施例中,使用另一等级的再分,其中还再分由虚线道路段提供的区中的每一者,因而提供三个等级的分割,但为了便于参考未在图 9 中展示这些等级。其它实施例当然可使用较多等级(例如,4、5、6、7 个或更多等级)或较少等级(例如,1 或 2 个等级)。

[0164] 因此,本发明的实施例针对较精细等级区引入所谓的可见度区域,步骤 1908。k 区

的可见度区域为一组 $(k-1)$ 个区,其中可在旗标集合中的所述区自己的位上区别所述区。自然地,可见度区域总是包括给定 k 区所属于的 $(k-1)$ 区。另外,可见度区域可能含有一些相邻的 $(k-1)$ 区。将在预处理期间计算可见度区域,且将可见度区域与位向量相关联地存储。

[0165] 还可从相反侧看出 k 区与其 $(k-1)$ 等级的可见度区域之间的关系:每个 $(k-1)$ 区知晓其由位于邻域的 k 等级区组成的外围(outskirt)。以此方式,可看出道路段的旗标集合不再贯穿整个地图具有固定长度;实情为,每个 1 区中的位向量具有其特定长度 $A+B+C+N$ 外围 $0+N$ 外围 1。其中 A 指代区的最粗略等级, B 指代中间粒度区,且 C 指代最精细粒度的区(在存在三个等级的实施例中)。

[0166] 后续预处理在最短路径计算之前计算可见度区域以产生位向量。可将找到可见度区域的方法描述如下:

[0167] 对于每一 k 区来说,开始区相邻图中的广度优先搜索;

[0168] 接着对于每一到访过的 k 区(其位置足够接近开始区),将其含有的 $(k-1)$ 区添加到可见度区域。

[0169] “接近”关系应将地理量度(例如,区中值点之间的距离)与图理论距离两者考虑在内。(因此,如果一区在地理上为远的但通过快速连接而链接到当前区,那么所述区可为“接近”的。同样,如果难以在区之间行进,那么一区可被认为“遥远的”(即使其在地理上接近))。

[0170] 确切的阈值可经受实验调谐,因为其视地图特定的特性(如都市区域的平均直径)而定,其最易受上文所描述的负面效应(例如,增加的预处理等等)影响。具有格外长行进时间的可导航段(如轮渡或禁止驾驶的道路段)在相邻图遍历(adjacency graphtraversal)期间被隐藏,这样可见度区域总是限于单一岛屿或属于同一区的小群岛。

[0171] 因此,在广度优先搜索期间到访过的区构成 R 的邻域。完全覆盖所述邻域的下一较粗略等级的区的包括最小(inclusion-minimal)集合被称为 R 的可见度区域。反向关系被称为附近(vicinity):区 Q (等级 L 处)的附近列表是由在等级 $L+1$ 处的在其可见度区域中有 Q 的所有区组成。

[0172] 为了提供特定实例,且参看图 9,可见度区域的实例如下:

[0173] 如果决定每一 1 区与其可见度区域的边缘之间的最小距离应为至少 1,那么一些选定 1 区的可见度区域将由以下 0 区组成(自己的 0 区可被省略,因为其总是存在):

[0174] • 15:

[0175] (即,等级 1 区第 15 号不具有可见度区域,因为其处于其等级 0 区的中心)。

[0176] • 28:5

[0177] (即,等级 1 区第 28 号具有为区(等级 0)第 5 号的单个可见度区)。

[0178] • 37:2,5,6

[0179] (即,等级 1 区第 37 号具有三个可见度区,其为等级 0 区 2、5 及 6)

[0180] 除了考虑等级 1 区的等级 0 邻居,还确定每一等级 0 区的等级 1 邻居。因此,作为实例,

[0181] • 1:21,24,27,28,41,42,43,51

[0182] (即,对于等级 0 区第 1 号,等级 1 区为:21、24、27、41、42、43、51)。

[0183] 因此,在此实例中,可看出,区 28 被列出,尽管其不在区 2 的最左侧列中。这(例如)可能因为区 28 具有到区 1 的快速链接,且因而当在时间而非距离方面进行考虑时为接近的。所属领域的技术人员将了解,可对照例如环保性(greenness)(最少 CO₂或其类似者)等其它量度来判断接近性。

[0184] 邻居列表的编码

[0185] 与图 9 相比,图 9a 展示可使用的等级的更多细节。等级 0 为最粗略等级且可被认为是等级 k-1。为简易起见,在图 9a 中,仅展示区 1 到区 4(即,602;604;606;608)。这些 k-1 区中的每一者进一步被划分成 9 个区(1 到 9),其可被称作 k 等级区或等级 1 区。另外,这些等级 1 区中的每一者被划分成 4 个 k+1 等级区(即,等级 2 区)。

[0186] 位向量的产生

[0187] 一旦网络已被分割成三个等级的区,那么其经处理以确定到所述区中的每一者的最小成本路径,且为区内的每一道路段创建位向量,步骤 1910。因此,如上文所论述,关于成本公式来分析任一区内的每一道路段,以确定其是否为到每个其它区的最低成本路径的部分。产生一区中的每一道路段的位向量,如图 7 中所展示,为便于理解,图 7 展示简化的位向量。

[0188] 还应了解,确定最小成本路径,其不仅针对单个时间段,而是针对多个时间段来计算(如下文中所描述)。

[0189] 针对图 7 中所展示的网络(即,减小的网络)来执行此计算。网络的已被移除的部分被有效地折叠到其所源于的节点上。举例来说,图 5 上所展示区 250 被折叠到图 7 中的节点 400 上。

[0190] 将看出,每一位向量包含三列:含有在考虑中的道路段的起始处的节点的身份编号的最左侧列 700;含有在考虑中的道路段的末端处的节点的身份编号的第二列 702;及含有所述道路段的位向量的第三列 704。因此,将看出,每一道路段由两个节点识别,在道路段的每一末端处存在一个节点。

[0191] 可使用任何合适的路线选择方法来确定道路段是否为最低成本路线的部分。在此特定实施例中,使用众所周知的迪科斯彻方法来探索整个网络。

[0192] 然而,可通过使用各种策略来减少处理时间的量。所属领域的技术人员应了解,上文所描述的用于减小网络的技术还将减小处理时间的量。本发明的实施例可使用以下各项中的任一者或一者以上:

[0193] •同时计算对应于一区的所有位向量条目。在反向图中,针对每一边界节点得出最短路径树。此方法为有利的,因为可并行处理每一区,藉此减少处理时间。

[0194] •减少对类似最短路径子树的重新计算。

[0195] •不重新计算已经产生的位向量。

[0196] 因此,总之,一个实施例可执行以下操作以便产生位向量:

[0197] 准备步骤:

[0198] 1. 如所属领域的技术人员将了解,经由电子地图的搜索可由有向非循环图(DAG)表示。此图及伴随数据结构(转弯成本及长扩展表)为反向的。

[0199] 2. 收缩关于最精细等级的简单道路段(即,将端节点一者折叠到另一者上,如别

处所论述)。如果道路段由次数 = 2 的一个或一个以上节点组成(所有节点均位于给定等级的同一区中),且可导航段具有相同属性:例如“是轮渡”、“禁止通行”及“禁止驾驶”,那么所述道路段被称为简单的。

[0200] 3. 对于每一区,视头(即,来自节点)及/或尾(即,去往节点)节点是否属于所述区而定,将道路网络的道路段分类成三组:出端口、入端口及内部道路段。即,如果头与尾两者在同一区内,那么道路段被称为内部道路段;如果头在区内但尾不在区内,那么道路段被称为出端口道路段;且如果尾在区中但头不在区中,那么道路段被称为入端口道路段。

[0201] 4. 对离开禁止通行及禁止驾驶区域的所有道路段设立特殊转弯限制。

[0202] 预处理例程以从下到上方式进行,以最精细分割等级-即,在正描述的实施例中以等级 2 开始。在每一等级的区处,针对每一区 R 重复以下步骤:

[0203] 1. 确定 R 的探索区域。在最高等级(例如,正描述的实施例中的区 0),其为整个图,在较精细等级,其为由 R 的可见度区域限制的子图(如上文所描述)。

[0204] 因此,在中间等级(即,等级 1)处,仅针对等级 0 区的含有所述等级 1 区的可见度区域执行以下步骤。在等级 0(应记住,其用于长程路线选择)处,考虑整个图的道路段。

[0205] 收集可见度区域的入端口道路段,即,从区域外部的节点通向所述区域内的道路段。接着收集边境道路段,即,在入端口道路段之后的道路段。如果头(L1) = 尾(L2)且不禁止从 L1 到 L2 中的转弯,那么将道路段 L2 称为在 L1 之后(及 L1 在 L2 之前)。为了找到边境道路段,将复杂十字路口收缩成单一节点。具有“禁止驾驶”属性的轮渡道路段及道路段不被视为边境道路段。

[0206] 以此方式收集入端口道路段可减少(可能显著减少)在下文中所描述的探索步骤中所需的处理量。探索步骤可减少对图的探索而考虑那些到给定区的包括入端口路线的路线。

[0207] 2. 对于 R 的每一出端口道路段来说,找出根道路段及探索步骤的数目。如果出端口道路段为其位于 R 内的前任者(predecessor)中的至少一者的唯一后继者,那么此出端口道路段为仅有的根道路段,且执行单个探索步骤。否则,如果出端口道路段为双向的(即,可在两个方向上行驶),那么将所述道路段自身及其相反(入端口)道路段取作单个探索步骤的根道路段。否则,如果出端口道路段为单向的,那么将每一前置的内部道路段(且如果存在,其相反道路段)取作用于单独探索步骤的根道路段。最后,不管出端口道路段的种类是什么,对于穿过出端口道路段的每一交通扩展路径,将所述扩展的开始道路段(如果其位于 R 内)取作用于单独探索步骤的根道路段。

[0208] 在较精细等级(即,除了等级 0 之外的所有等级)上,特殊对待出端口轮渡道路段。如上文所提到,在确定区的邻域时,忽略轮渡道路段。如果轮渡道路段的头部区不属于 R 的可见度区域,那么将执行单个探索步骤,其中轮渡道路段为唯一根道路段,且探索区域被约束到头部区自身。

[0209] 3. 执行经调度的探索步骤(下文中所描述)。

[0210] 4. 追溯从探索区域内的道路段到根道路段的最短路径。在除了最高等级(即,等级 0)之外的所有等级处,相应跟踪起始于的道路段的排序会影响结果。假设 R 为等级 L 区(其中 $L > 0$),那么首先处理等级 (L-1) 区的出端口道路段,接着处理等级 L 处的剩余出端口道路段,并依此处理到最精细等级;相对于最精细等级来说为内部(且尚未被移除)的道

路段被最后处理。每当遇到已在较早跟踪中到访过的道路段时,都没有必要再次沿着所述路径前进。在跟踪的同时,为每一到访过的道路段设定适当的目标位向量。在除了最高等级(即,等级0)之外的所有等级处,进行额外动作:

[0211] 在路径已初次横穿某等级L区边界之后,用运输道路段旗标来标记所述边界道路段及去往根道路段的所有其它道路段;

[0212] 用U形转弯旗标来标记最短路径进行U形转弯的节点。

[0213] 最后,在除了最精细等级外的所有等级处,填充相关矩阵条目。

[0214] 在处理所述级的所有区之后,针对下一等级而简化所述图。首先,移除未标记为运输道路段的所有道路段。接着,收缩积累的新的简单路径;保留被标记为U形转弯节点的节点。

[0215] 最后,根据相关矩阵将位向量传播到在处理此等级之前移除的道路段。

[0216] 相关矩阵

[0217] 本发明的一些实施例使用相关矩阵,其存储对道路段设定位向量所需的目标区之间的关系。在除了最精细等级之外的每一等级L处,产生且使用新的相关矩阵。在等级L处,矩阵的行按等级(L+1)区编索引,列按等级L区编索引,且每一矩阵条目为零或零个以上等级(L+1)区的集合。在较低等级处,大多数矩阵条目等于空集,即,矩阵为稀疏的。

[0218] 相关矩阵的目的是为了帮助对已在较早阶段删除的道路段设定位向量。这些道路段未包含于由等级L探索步骤导致的有向非循环图中,但如果其尚未被删除,那么其将包含于有向非循环图中。更精确地说,相关矩阵用以对在S的探索区域中的道路段(在 $>L$ (其中L不为最精细等级)的某等级的计算期间,其已被删除)上的某等级L区S设定位向量。对于包含于S的探索区域中的等级(L+1)区R,矩阵元素 $M[R,S]$ 将最终为在R的可见度区域的边界处的等级(L+1)区的集合,使得从S到R的所有最短路径(在反向图中)均穿过所述区中的一者。

[0219] 当创建矩阵时,将所有条目初始化成空集。接着,对于所有等级L区S,执行以下两个动作。

[0220] 1. 矩阵建立:对于根道路段1在S及所得有向非循环图D中或上的每一探索步骤,针对包含于S的探索区域中的每一等级(L+1)区R,及针对R的每一入端口道路段 $1'$,将矩阵元素 $M[R,S]$ 更新如下:

[0221] 用A表示R的探索区域(如较早针对等级L+1而计算)。在D中从 $1'$ 追溯路径到1且检查其是否离开A:如果此路径上的道路段从等级(L+1)区T转到等级(L+1)区 T' (其中T仍包含于A中,但 T' 未包含于A中),那么将T添加到集合 $M[R,S]$,且处理下一 $1'$ 。

[0222] 2. 读取矩阵:对于S的探索区域中的已在等级L计算开始之前被删除的每一道路段,令R为所述道路段结束于的等级(L+1)区(再一次,相对于反向图来说)。现在将在所述道路段上的区S的位向量位设定成区T的位向量位的逻辑“或”,其中T包括 $M[R,S]$ 的所有元素。

[0223] 注意,将已在较早等级处直接根据某有向非循环图或按照涉及某较低等级处的相关矩阵的类似程序设定每一T的位向量位。

[0224] 探索

[0225] 探索步骤由建立以给定道路段(对)为根的最小成本路径的有向非循环图组成。

此是通过使用用于最小成本计算的众所周知的迪科斯彻方法的变化形式来实现。可自由选择要最小化的目标函数,例如,行进时间或估计燃料消耗,或别处所论述的其它因素中的任一因素。

[0226] 一个实施例使用行进时间、路径长度及抑制不当转弯及机动的其它惩罚项的加权总和。

[0227] 对经典(迪科斯彻)方法作出以下修改:

[0228] 1. 其对道路段图起作用,即,正到访、放松等等的项目为道路段,而非节点。此有助于允许所述方法考虑到转弯限制及/或交通扩展。

[0229] 2. 标签中的目标函数为在根道路段的针对到达时隙的固定集合的两个一组(行进时间、成本)的向量。选择时隙以使得覆盖所有相关交通模式(自由流动、工作日早晨高峰时间、傍晚高峰时间,等等)。在下文关于评估多个时间段时的成本的论述中进一步详述此内容。

[0230] 3. 可针对给定道路段存储一个以上标签。如果两个标签上的待决(未完成的)交通扩展的集合不相等,那么所述标签自身被称为独立的且两者均在随后道路段上不断传播。否则,如果不同到达时隙的成本函数值之间的关系不交替(即,一个标签明显比另一标签好),那么丢弃较差标签。否则,通过合并每一时隙的较好值而产生新标签,将其代替原始标签来传播。经合并标签的前任者集合于是为原始标签的前任者集合的联合。

[0231] 4. 在每一双向道路段上产生特殊U形转弯标签。所述标签编码在两个方向上开始真实(非反向)路线的可能性。不传播U形转弯标签,且不可将U形转弯标签与一般标签合并。然而,U形转弯标签影响设定位向量时的回溯阶段:仅在开始标签不比同一段道路上的U形转弯标签差时,最短路径才被加上旗标。

[0232] 在较精细等级处(其中将探索区域限于区的集合),永久地监看如上文所界定的边境道路段。一旦搜索前端到达所有边境道路段,就根据每时隙的最大(=最差)成本函数值建立监看梳(watch comb)。接着,在探索区域外部的道路段上的标签在弹离堆积时,仅在所述标签在至少一个时隙中的成本函数值低于当前监看梳的情况下被传播。如果探索区域伸展越过若干岛屿,那么针对每一岛屿维持单独的监看梳。

[0233] 时变函数

[0234] 本发明的一些实施例可计算展示在多个时间段而不是在单个时间处跨越网络的最小成本路径的位向量。所属领域的技术人员将了解,穿过道路网络的最低成本路线可归因于交通密度等等的影响而随着时间变化。因此,对于任一节点,可能存在两个或两个以上最小成本路径,每一最小成本路径是针对不同时间。在此实施例中,不以关于何时最小成本路径可适用的时间基准来译码位向量。位向量仅被设定成将可导航段识别为是最小成本路径的部分或不是最小成本路径的部分。因此,当使用最小成本数据进行路线选择时,路线选择算法将必须考虑来自节点的所有可能的最小成本路径。现在借助于图10a简要描述此过程。

[0235] 在网络的标准迪科斯彻探索中,当探索网络时,所述方法使用为了到达网络中的所述点迄今所发生的总成本加上尚待发生的预期成本。

[0236] 因此,一些实施例利用函数而非离散值来作出在每一节点处的成本评估。因此,在图10a中,图751的正被探索的每一节点(例如,750)具有与之相关联的成本函数,其识别

正探索的参数（例如，时间、花费的燃料或其类似者）如何随着时间而变化。

[0237] 成本函数可与道路段而非节点相关联。

[0238] 所述方法接着处理花费的成本以通过将当前节点处的函数与迄今已累积的成本求和来添加估计成本以产生新函数。图 10a 中所展示的实例在 752 处展示已在节点 750 处通过搜索方法产生的成本函数，且展示行进时间（y 轴）如何随着出发时间（x 轴）而变化。将看出，成本函数归因于早晨及傍晚高峰时间而在点 754 及 756 处增加。

[0239] 在一个特定实施例中，以 5 分钟间隔存储成本函数（例如，道路段上的平均速度）；即，其为具有 5 分钟的时间周期的经量化函数而非连续变化的函数。

[0240] 如果道路段在任何时间段均为最低成本路线的部分，那么设定所述道路段的位向量。

[0241] 将核心数据投射到整个网络

[0242] 上文描述了如何减少地图中所含有的网络以便减少必须由分割方法考虑的道路段及节点的数目。然而，还应进一步考虑在减少步骤中被删除的节点，以便路线选择方法仍可产生去往或来自被删除的道路段及节点的路线。

[0243] 因而，将被删除的节点及道路段指派到与在核心网络中其所连接到的节点相同的区。

[0244] 压缩

[0245] 如所论述，所产生的位向量的大小为非常大的，且因此需要压缩信息。本发明的实施例可以不同方式执行此压缩。然而，一个实施例利用各种技术来压缩、聚结位向量及 / 或使位向量相关，接着是对所述位向量的后续霍夫曼编码。

[0246] 因此，一些实施例可试图并确保存在位向量的不均匀分布，因为这可有助于确保霍夫曼编码比在其它情况下更有效率。

[0247] 举例来说，如果位向量分布如下：

[0248]

00000....	(49%的时间)
11111....	(49%的时间)
?????...	(2%的时间)

[0249] 那么可能需要在霍夫曼编码之前操控位向量以具有更不均匀的分布，例如：

[0250]

00000....	(79%的时间)
11111....	(19%的时间)
?????...	(2%的时间)

[0251] 减少所产生的位向量

[0252] 为了减少需要产生的位向量的量，本发明的实施例可使用以下策略中的任一者或一者以上：

[0253] •并非针对所有节点使用区 ID，而使仅针对可导航节点产生区 ID（例如，忽略对应于铁路的节点）。

[0254] •并非所有道路段均需要位向量，且位向量可用于决策节点周围的决策道路段。可通过查看道路段数据来确定决策节点及决策道路段（如本文献中所描述）。

[0255] • 尽管存在许多可能的位向量,但一些位向量远比其它位向量频繁,所以特殊编码可用于最频繁的位向量(例如,000...000 及 111...111)。

[0256] • 并非 000...000 或 111...111 的位向量的大多数位仍常常设定成 1,或设定成 0。所以部分块的霍夫曼码应相当高效地编码所述位向量。

[0257] • 在彼此接近的节点中,位向量常常相同,所以增量(delta)编码可高效地编码所述位向量。

[0258] • 可通过按源区将目的地区 ID 重新排序来使不同区具有更类似的位向量模式(想法描述于本文献中)。

[0259] • 或者节点的道路段周围的所有位向量应总是给出 111...111。这适当地可用以更高效地编码位向量。

[0260] 下文中进一步详细论述这些情况中的一些情况。

[0261] 这里值得注意的是,本文中所描述的技术旨在减小位向量的大小。然而,应注意,出于路线选择的目的地使用地图数据的装置需要对数据的随机存取。通常,对数据的高效编码需要可变大小,然而,这将阻止对数据的随机存取。

[0262] 因而,本发明的实施例可使用一种折衷,其中数据被编码为一系列加索引的页,且接着利用所述页内的可变编码。在此些实施例中,可实现对每一页的随机存取(经由索引)。一旦一页已被存取,那么实施例可随后解码整个页。这提供效率与地图大小之间的折衷-增加每页的节点的数目会减小地图大小,但使数据存取变慢。本发明的一个特定实施例使用每页 16 个节点。将了解,任何一个节点还可具有离开所述节点的不同数目或道路段。因而,即使可能存在相同数目个节点,每页仍可存在可变量的道路段。另外,对于存储在每一页上的位向量中的每一者,还可发生不同压缩。

[0263] 此结构可导致如图 11 中所展示的地图格式。在此图中,对 Arp 旗标的参考与位向量同义。将数字“n”存储在标头内,且可针对不同地图变更所述数字“n”,以便优化所述地图的性能。

[0264] 调谐“n”为在地图大小与在解码地图数据时的存取速度之间的折衷:

[0265] • 大的 n 值将把许多节点分组在一起,这对于地图压缩有益,但对于对数据的随机存取的速度不利。

[0266] • 小的 n 值将把少数节点分组在一起,这对于对数据的随机存取的速度有益,但对于地图压缩不利。

[0267] • 可将“n”设定成(例如)4,即,16 个始发节点(始发节点在道路段的开始端处-即,图 10 的列 700)的页,但应记住,每一始发节点具有若干个到达道路段(to-roadsegment),所以在假定平均有 3 个到达道路段的情况下,每一者意味着每一页的存储量相当于约 48 个道路段。

[0268] 在此格式中,取决于经编码的区的等级,数据具有不同格式。图 12 展示等级 0(即,最粗略区)的格式的实例,且图 12 展示其它等级的区的实例格式。

[0269] 位向量及相关信息存储在数据结构中,在图 11 中展示所述数据结构的格式,所述格式包含以下各项:标头 800;用于稍后描述的霍夫曼编码中的霍夫曼树 802;每一层级中的区计数 804(在每等级具有恒定数目个区的情况下为空的);区邻居 806(在无区邻居的情况下为空的);区 ID 重新映射表 808;位向量页索引(2ⁿ个节点)810;及区 ID 与位向量

812. 可将保持位向量的数据结构保持于单个文件内,或可将其保持于多个文件内。

[0270] 在一些实施例中,地图标头 800 经布置成含有指示以下各项的进一步信息:

[0271] • 等级的最大数目

[0272] • 最短邻居列表的长度。

[0273] • 最长邻居列表的长度。

[0274] • 含有所有邻居列表的区段的字节偏移。

[0275] 保持信息的文件或每一文件可具有用以编码邻居列表的区段。首先编码所有列表的大小。相对于最短列表长度、在由 BitsRequired(所需位)(最长列表长度 - 最短列表长度(longestListLength-shortestListLength))确定的固定数目个位上编码每一列表长度。注意,如果所有列表具有相同长度,那么不需要位来编码所述长度。

[0276] 接着编码所有列表的内容。每一列表由邻居区 ID 的若干元组构成:等级 0 处的对、等级 2 处的 3 元素元组,等等。

[0277] 注意,不编码始发区元组(ASCII 文件中的在“:”之前的部分)。其为隐含的,因为所有区的列表以递升次序存储。举例来说,如果地图具有 $100 \times 10 \times 5$ 的 3 个等级(等级 0 处的 100 个区、等级 1 处的 10 个区、等级 2 处的 5 个区),那么:

[0278] • 在等级 0 处,存储始发区 1、2、3、...、100 的列表(以此次序的 100 个列表)。这些列表中的每一者含有若干对。

[0279] • 在等级 1 处,存储始发区 1.1、1.2、1.3、...、1.10、2.1、2.2、...、2.10、3.1、...、100.9、100.10 的列表(以此次序的 1000 个列表)。这些列表中的每一者含有 3 元素元组。

[0280] • 在等级 2 处:不存储任何东西,因为其为最后等级,所以不存在邻居。

[0281] 将元组中的每一组成部分存储为 n 位。根据处于对应等级的区的数目确定每一等级的位的数目。所以,有可能随机存取列表。在 3 个等级 $100 \times 10 \times 5$ 的情况下,编码元组 a. b. c 将针对 a 使用 7 个位(因为在等级 0 处存在 100 个区),针对 b 使用 4 个位(因为在等级 1 处存在 10 个区),且针对 c 使用 3 个位(因为在等级 2 处存在 5 个区)。

[0282] 实例:假定 $100 \times 10 \times 5$ 个区有如下分割:粗略等级 0 处的 100 个区、中间等级 1 处的 10 个区,及详细等级 2 处的 5 个区。

[0283] 在等级 0 处的文件中,含有邻居列表的区段将含有:

[0284] • 指示处于等级 0 的 100 个区的列表的长度的 100 个数目。根据 BitsRequired(所需位)(longestListLength-shortestListLength(最长列表长度 - 最短列表长度))计算位的数目。每一数目是相对于所述级处的最短列表(最短列表存储在标头中)。

[0285] • 接着为 100 个列表的内容(100 对)。在 7 个位上编码每一对的第一元素(因为在等级 0 处存在 100 个区),且在 4 个位上编码每一对的第二元素(因为在等级 1 处存在 10 个区)。

[0286] 在等级 1 处的文件中,含有邻居列表的区段将含有:

[0287] • • $100 \times 10 = 1000$ 个数目,其指示等级 1 处的 1000 个区的列表的长度。

[0288] • • 接着为 1000 个列表的内容(1000 个 3 元素元组)。在 7 个位上编码每一元组的第一元素(因为在等级 0 处存在 100 个区),在 4 个位上编码每一元组的第二元素(因为在每一等级 0 区中存在 10 个等级 1 区),且在 3 个位上编码每一元组的第三元素(因为在每一等级 1 区中存在 5 个等级 2 区)。

- [0289] 在等级 2 处的文件中,不需要存储任何内容,因为其为最后等级。
- [0290] 对于编码器输入文件,可采用任一形式的列表。
- [0291] 因此,一旦已执行分割为三个等级,就将每一节点指派给每等级一个区;即三个区。
- [0292] 标头 800
- [0293] 通常,用于本发明的实施例中的标头为较小的,且因而,不需要为了减小其大小而使所述大小优化。通常,为了方便起见,全部内容均为字节对准或字对准的:
- [0294] • (4 个字节) 编码版本(每次地图格式改变时会增加)
- [0295] • (4 个字节) 地图旗标(用以开启或关闭特征,最初为 0,但以后在需要添加任选特征的情况下可被使用)
- [0296] • (4 个字节) 地图中的节点的总数目
- [0297] • (4 个字节) 地图中的道路段的总数目
- [0298] • (4 个字节) 区段霍夫曼树的字节偏移
- [0299] • (4 个字节) 区段区二进制大型对象(blob)的字节偏移
- [0300] • (4 个字节) 区段区页信息的字节偏移
- [0301] • (4 个字节) 区段位向量页信息的字节偏移
- [0302] • (4 个字节) 区段可变大小记录的字节偏移
- [0303] • (4 个字节) 最大位向量(arp 旗标)页(以位计)(可由路线规划方法用以在启动时预分配位流解码器的较坏情况)
- [0304] • (4 个字节) 平均位向量(arp 旗标)页大小(以位计)(用以内插位向量页位置)
- [0305] • (4 个字节) 最小位向量(arp 旗标)页增量(用以使得所有增量 ≥ 0 ,从而避免存储位正负号)
- [0306] • (2 个字节) 位向量(arp 旗标)历史的最大大小(可由路线规划方法用以在启动时预分配历史缓冲器)
- [0307] • (2 个字节) 每页的道路段的最大数目(当前未使用)
- [0308] • (1 个字节) 此文件的阿波罗(Apollo)等级
- [0309] • (1 个字节) 每位向量(arp 旗标)的位
- [0310] • (1 个字节) 每位向量(arp 旗标)页增量的位(位向量(arp 旗标)页的固定大小记录中的字段)
- [0311] • (1 个字节) 每二进制大型对象索引的位(区页信息的固定大小记录中的字段)
- [0312] • (1 个字节) 每区计数的位(区页信息的固定大小记录中的字段)
- [0313] • (1 个字节) 每非平凡(trivial)位向量(arp 旗标)块的位
- [0314] • (1 个字节) 区节点页大小的 $\log_2()$
- [0315] • (1 个字节) 位向量(arp 旗标)页大小的 $\log_2()$
- [0316] • (1 个字节) 用以编码本地区 ID 的霍夫曼树的数目
- [0317] • (1 个字节) 用以编码位向量(arp 旗标)历史代码的霍夫曼树的数目
- [0318] 霍夫曼树 802
- [0319] • 用来编码在每一节点周围的道路段的数目的霍夫曼树:极小的,仅大约 10 个码,

仅存在于等级 0 处的文件中)

[0320] • 用来存储非平凡位向量 (arp 旗标) 的块的霍夫曼树: 最大霍夫曼树, 对于压缩来说, 越大越好, 但在路线规划方法中所需的存储器越多 (在地图压缩与路线规划方法中的存储器使用之间的折衷)。

[0321] • 当历史大小为 0 时位向量 (arp 旗标) 增量码的霍夫曼树: 极小的, 仅 3 个码

[0322] • 当历史大小为 1 时位向量 (arp 旗标) 增量码的霍夫曼树: 极小的, 仅 4 个码

[0323] • 当历史大小 $\geq n$ 时位向量 (arp 旗标) 增量码的霍夫曼树: 极小的 (存储在标头中的霍夫曼树的数目)

[0324] • 当在区页中存在 3 个区时区 ID 的霍夫曼树: 极小的, 仅 3 个码

[0325] • 当在区页中存在 4 个区时区 ID 的霍夫曼树: 极小的, 仅 4 个码

[0326] • 当在区页中存在 $\geq n$ 个区时区 ID 的霍夫曼树: 极小的 (存储在标头中的霍夫曼树的数目)。

[0327] 区重新映射表 804 及区 ID 列表 806

[0328] 虽然区 ID 806 比图 11 的文件格式的其它部分小, 但还可压缩区 ID 806, 如图 14 中示范。其中, 可使用地理相关以便减少所使用的数据量。

[0329] 每一区页在所述区页中存储相异区的列表。在大多数情况下预期此列表为较小的 (事实上, 许多页很可能仅含有 1 个至少在等级 0 的区)。区的列表具有可变大小。应可随机地存取页内的数据 (即, 随机存取) 且因而, 使用固定表大小以允许此情况。

[0330] 将相异区的每一列表按每一区中的节点的频率排序: 列表的第一元素对应于所述页中具有最大数目个节点的区, 列表的最后元素为所述页中具有最小数目个节点的区。

[0331] 对于区页中的每一节点, 可使用本地区 ID (对页来说为本地的) 来编码其区 ID。此本地 ID 为页中的区的索引 (其为较小的整数, 常常为 0, 因为 0 对应于区页中的最普遍的区)。

[0332] 将节点的区 ID 存储如下:

[0333] • 含有区 ID 的区阵列存储页中的区的所有可能的重叠列表。区的列表为所述阵列中的连续区 ID。列表可 (且确实) 重叠。所述阵列不存储每一列表的开始及结束 (此由区页信息表进行)。

[0334] • 区页信息表为固定大小的记录表 (因此可随机存取), 且每一记录含有在列表的开始的阵列中的索引 + 所述列表中的项目的数目。

[0335] • 每一节点含有本地节点 ID (相对于其页来说为本地的)。

[0336] 在下文中进一步界定这些概念中的每一者。

[0337] 区阵列

[0338] 区阵列编码页的所有可能的区列表。区阵列为区 ID 的简单阵列, 其中区 ID 的列表可重叠。区阵列的大小应为较小的, 因为列表重叠。在区段区页信息中进一步描述所述区阵列。

[0339] 区页信息

[0340] 使用区页信息表的固定大小记录中的 2 个字段指定区页表中的区 ID 的列表:

[0341] • 区计数 (此页中的区的数目, 预期为较小的)。

[0342] • 向区列表的阵列中的偏移 (区的列表开始之处)。

[0343] 在一个实施例中,此在图 12 中描述。

[0344] 偏移字段指向区阵列中:(例如)在假定在每一等级中总是存在小于 256 个区的情况下(其为合理假定),每区 ID 具有 1 字节的固定大小记录是足够的(但如果每等级 256 个位被认为太有限制性,那么使得每个区 ID 大于 8 个位为易于实现的)。区页表记录中的区计数指示在指定偏移处在阵列中有多少区要被考虑。

[0345] 如果若干区具有相同列表,那么其可指向相同所在地,其应为紧凑的,因为可预期许多页共享相同列表或共享所述相同列表的部分。

[0346] 参考图 12 来更详细解释此,图 12 展示具有 2^{nr} 个节点((例如) $nr = 9$ 以将 512 个节点分组在一起)的页的实施例。

[0347] 注意区 ID 的阵列 900 紧凑的程度,因为若干页可指向阵列中的同一所在地或重叠的所在地(图中标记的区二进制大型对象)。事实上,增加页的数目可能不增加阵列的大小,因为每一页接着在较少的区上重叠,所以组合的可能性减小。所以此阵列应允许创建许多页,而无需太多地图空间,也无需所产生的地图数据被加载于其中的装置上的过多存储器。

[0348] 所述方法旨在使含有区 ID 的列表的阵列 900 尽可能地小。所述方法旨在在阵列中尽可能经常地重新使用相同的区 ID 列表。所述方法可自由地将列表的最后 2 个元素重新排序,因为其不会影响霍夫曼码的大小。

[0349] 举例来说,当列表含有 2 个区 10 及 34 时,存储列表 10、34 或 34、10(与频率无关)为等效的,因为当仅存在 2 个节点时,霍夫曼树在所有情况下仅使用 1 个位。换句话说,对于最后 2 个区来说,放松了根据频率排序的要求。类似地,还可将 3 个元素 10、34、40 的列表存储为 10、40、34,因为仅第一个码 10(最频繁的)将使用 1 个位,且其它码 40 及 34 均使用 2 个位(与次序无关)。

[0350] 因此,查看图 12,可看出,阵列 900 存储两个值:长度及从区数据的开始的偏移。因此,以第一行(3 : 0)为例,此指代从文件的开始偏移了 0 的三条数据(即,10、34、40)。作为另一实例,阵列条目(1 : 2)指代单个区(即,长度为 1),其中从文件的开始的偏移为二;(即,区 40)。

[0351] 在替代实施例中,根据以下方法来编码区页信息:

[0352] 此区段编码每一区中的子区的数目。子区的数目可在每等级中可变。然而,每一等级的子区的数目常常为恒定的。相对于每一等级处的区的最小数目且使用 $\log_2(\max_number_of_regions - \min_number_of_region)$ (区的最大数目 - 区的最小数目)个位来编码子区的数目。所以如果区的数目为恒定的,那么将 0 个位用以编码区计数,且此区段为空的。区的最小数目及最大数目存储在侧文件的标头中。

[0353] 区邻居区段(关于图 6 及 6a 来论述邻域)的编码

[0354] 此区段针对在给定等级 L 处的每一区层级而编码在更详细等级 L+1 处的区邻居的列表。举例来说,等级 L = 1 处的区 3.9 可具有以下在等级 L = 2 处的邻居的列表: 3. 5. 43. 6. 33. 6. 44. 7. 14. 7. 3。如别处所论述,可使用邻居的列表来增加用以产生所述侧文件或每一侧文件的预处理的速度。

[0355] 此区段被分裂成 2 个子区段:

[0356] • 用以编码所有邻居列表(与在给定等级处的区数目一样多)的长度的子区

段。相对于最短列表来编码所述长度,且接着,将位的数目计算为 $\log_2(\text{length_longest_list} - \text{length_shortest_list})$ (最长列表长度 - 最短列表长度)。如果所有列表具有相同长度,那么使用 0 个位来编码长度 (且因此,区段就为空的)。

[0357] • 用以编码所有邻居列表 (与在给定等级处的区数目一样多) 的子区段。

[0358] 区 ID 重新映射表的编码

[0359] 仅在等级 0 的侧文件中编码此区段。所述编码编码二维表,所述二维表被用以重新排序且聚结等级 0 处的每一区中的位 (以便高效地编码位向量,在本文献中在下文进一步描述聚结及位重新排序)。进行对位向量中的位的重新排序及聚结以优化对位向量的霍夫曼编码。此表由路线规划方法用以在知晓以下各项时找到位向量中的要解码的位位置:

[0360] • 当前节点的始发区 ID

[0361] • 原始的到达位 (to-bit) 索引 (即,在解聚结位向量位之后的位索引)

[0362] 二维表的 2 个索引为:

[0363] • 始发区 ID

[0364] • 目的地位索引 (目的地的区)

[0365] 此区段由 2 个子区段构成:

[0366] • 用以编码等级 0 处的每一区的经聚结位的数目的区段。用以编码每一数目的位数目为 $\log_2(\text{max_number_of_coalesed_bits})$ (经聚结位的最大数目)

[0367] • 用以编码位重新映射表的区段。因为在进行路线选择时,目的地位元不改变 (在进行路线选择时目的地保持相同),但始发区 ID 改变 (视在进行路线选择时所探索的节点而定),矩阵按目的地位的行进行存储。在每一行中,存储每一始发区的位重新排序数目。路线规划方法在进行路线选择时应通常仅将对应于给定路线选择目的地的 1 行加载于存储器中。路线规划方法不需要将整个二维矩阵存储在存储器中。路线规划方法可随机存取所述行。将用以编码每一重新映射条目的位数目计算为 $\log_2$ (经聚结位的最大数目)。

[0368] 图 13 针对等级 0 区 (见图 9) 扩展了图 11 中所展示的文件的位置向量区段 812。可看出,每一页包含道路段计数、区 ID 及位向量。

[0369] 图 14 针对除了等级 0 之外的等级扩展了图 11 中所展示的文件的位置向量区段 812。可看出,对于其它区来说,存储位向量,且不存储道路段计数或区 ID (针对等级 0 存储道路段计数或区 ID)。

[0370] 鉴于由每一等级的区覆盖的区域的差别,在每等级中每页的节点的数目可变化。举例来说,由于等级 0 内的区覆盖较大的区域,因此对于等级 0,每区每页可能存在较多节点。举例来说,对于等级 0,一些实施例每区每页可存储 512 个节点。因而,更详细等级可具有较小数目个节点 - 例如,256 个节点、128 个节点、64 个节点、32 个节点、16 个节点或其类似者。实际上,某实施例可利用每页 1024 个节点、2048 个节点或 4096 个节点。

[0371] 位向量 810、812 的编码

[0372] 表 810 含有固定大小的记录。始发节点 ID 被一起分组于具有 2^n 个的页中。

[0373] 将数据分组于具有多个连续节点的页中是方便的,因为预期位向量对于相同邻域中的若干道路段具有类似模式。通过使用页,有可能以增量编码若干道路段且实现良好压缩。类似地,有可能以增量将节点的区 ID 编码于页中。另一方面,这意味着存取一个道路段的数据需要将若干道路段的数据解封装 (无直接随机存取)。可将必须将若干节点或道

路段解封装以存取一个道路段或节点的数据认为是可接受的,因为:

[0374] • 数据可被高速缓存,所以在存取一个道路段时读取的额外数据常常并非无用的。可能为以下情况:此额外数据不久之后将为有用的(这类似于磁盘高速缓存的预先读取(read-ahead)参数)。

[0375] • 当与迪科斯彻 A*路线选择比较时,使用所述位向量的路线选择将扩展搜索的大小减小了一数量级。通过按页分组数据,仅地图的小部分仍需要解码(沿着实际路径的页)。

[0376] • 归功于对区 ID 及位向量的增量压缩,其应显著减小经编码的数据大小。

[0377] • 页减小索引的大小,因为如所描述,数据将存储在侧文件中。

[0378] 表 810 内的每一记录含有一“增量”字段,其用以找出每一页的可变大小的开始的位置(相对于经内插位置的增量)。针对每一增量的位的数目存储在标头中。

[0379] 为了存取区 ID 及位向量 812,解码器可通过进行以下线性内插而计算开始页的估计偏移:

[0380] $\text{interpolated_offset}(\text{经内插_偏移}) = \text{from_node_id} * \text{avg_page_size}(\text{始发节点 id} * \text{平均页大小})$

[0381] 其中 avg_page_size 为存储在标头中的以位计的平均页大小(可能在固定点中以改进精确性)。可接着将数据的偏移计算如下:

[0382] $\text{偏移} = \text{interpolated_offset} + \text{min_delta} + \text{delta}(\text{经内插_偏移} + \text{最小_增量} + \text{增量})$

[0383] 其中 min_delta 为所有页的所有增量字段的最小值(存储在标头中),且增量为存储在页中的无正负号字段。min_delta 值确保所有增量字段为正值(不存储位正负号)。

[0384] 经由先前所描述的位向量页信息的“增量”字段来存取可变大小的记录。

[0385] 每一记录含有 2^n 个节点的数据(始发节点的区 ID 及其在所有等级处的所附接道路段的位向量)。同一编索引方案因此将被用于所有等级。

[0386] 可变大小记录存储:

[0387] • 页码 - 针对整个页指示所述页内的节点是否为同一区的部分的码;

[0388] • 在位向量页中的每一节点周围的道路段的数目(仅在等级 0 处存储,因为其对于所有等级将为相同的);

[0389] • 页中的始发节点的区 ID,每等级有一个 ID(针对所有等级的信息存储在等级 0 处的文件中 {ap_0_*.dat})

[0390] • 仅等级 i 处的在页中的节点周围(仅在具有 > 2 个所附接道路段的节点周围)的道路段的位向量。此为数据的最大部分。

[0391] • 在每一节点周围的道路段的数目的编码

[0392] 对于位向量页中的 2^n 个节点中的每一者来说,霍夫曼码编码在所述节点周围的道路段的数目。此信息并非具体到所有等级,且其仅存储在等级 0 处的文件中 (ap_0_*.dat)。

[0393] 知晓在节点周围的道路段的数目是用以解码位向量(见 1000,图 13)。此信息与已经存在于其它文件中的信息重复,但其使得更容易且快速地解码页中的位向量,而无须在别处查找所述信息;藉此大小方面的小增加提供性能的增加。

[0394] 可变大小记录中的区 ID 的编码

[0395] 在编码每一节点周围的道路段的数目 1000(见编码布局)之后,即刻在可变大小记录中编码节点的区 ID 1002。为了执行使用在预处理中产生的位向量的路线选择,通常需要存取给定节点的所有等级处的区 ID 1002,所有等级的区 ID 被彼此靠近地存储在同一文件中,而不是被分裂于每等级的不同文件中。

[0396] 具有 2^n 个节点(例如, $n = 4$) 且具有 3 个阿波罗等级的位向量页因此将把节点 ID 存储如下:

[0397] 节点 #0:本地_区_id_等级_0 本地_区_id_等级_1 本地_区_id_等级_2

[0398] 节点 #1:本地_区_id_等级_0 本地_区_id_等级_1 本地_区_id_等级_2

[0399] 节点 #2:本地_区_id_等级_0 本地_区_id_等级_1 本地_区_id_等级_2

[0400] 节点 #15:本地_区_id_等级_0 本地_区_id_等级_1 本地_区_id_等级_2

[0401] 另外:

[0402] • 在具有 1 或 2 个所附接道路段的节点周围,不编码任何东西(即,对于存储 0 作为所附接道路段的数目的节点来说)。

[0403] • 当在给定等级设定了页码中的位时,则知晓所有节点均处于所述级处的相同区 ID 中,且接着仅将所述级处的区 ID 编码一次(对于具有 ≥ 3 个所附接道路段的第一节点)。用来编码区 ID 的位的数目为 $\log_2(\text{regionCount}-1)$ (区计数-1)。

[0404] • 除了页中的区 ID 经编码的第一节点之外,在编码每一区 ID 之前还编码一位。此位指示区 ID 是否与处于同一等级的先前经编码的节点 ID 相同。当此位被设定时,无需编码所述区 ID,因为其与所述级处的先前所编码者相同。当此位为 0 时,以 $\log_2(\text{regionCount}-1)$ 个位编码区 ID。由于许多连续节点在同一区中,因此常常仅需要 1 个位来编码区 ID。

[0405] 对本地区索引的霍夫曼编码为高效的,因为:

[0406] • 在每一区页中按频率将区分类(所以本地索引 0 比本地索引 1 更频繁,...)

[0407] • 对于页中的每一数目个区存在相异的专门的霍夫曼树(对于页中的 3 个区存在 1 个霍夫曼,对于页中的 4 个区存在 1 个霍夫曼树,等等)。霍夫曼树为较小的,且因而有可能在不使用大量存储器的情況下存储若干个霍夫曼树。

[0408] • 无论如何,至少在等级 0 处,具有 3 个区或 3 个区以上应为相当罕见的(但在其它等级处并不罕见)。

[0409] 可变大小记录中的位向量的编码

[0410] 每一可变大小记录含有在页中的页周围的所有道路段的位向量。仅在具有 3 个或 3 个以上所附接道路段(道路段)的节点周围编码位向量。对于具有 1 个或 2 个所附接道路段的节点来说,路线选择方法可将位向量值 111...111 隐含地赋予所述节点。

[0411] 不编码到达节点。

[0412] 参看图 10,应注意,可由两个节点指定一道路段;在所述道路段的每一末端处有一个节点。因此,当考虑方向时,在方向的开始处的节点可被称作 from_node(始发节点),且在末端处的节点可被称作 to_node(去往节点)。

[0413] 在编码内使用关于位向量的各种性质以使得编码有效率:

[0414] • 所述位向量中的许多位向量为 000...000 或 111...111。

[0415] • 对于位向量的其它值(即,非 000...000 或 111...111),很可能存在重复且相同

值将很可能被重复。

[0416] • 并且,给定节点周围的所有位向量的“或”应为 111...111。

[0417] 将位向量编码为第一霍夫曼码及任选的其它霍夫曼码。第一霍夫曼码指示位向量是否为:

[0418] • 针对平凡位向量 000...000 的码 0

[0419] • 针对平凡位向量 111...111 的码 1

[0420] • 码 2,或用以指示在所述页中尚未遇到非平凡位向量。在此情况下且仅在此情况下,后接有其它霍夫曼码以编码新遇到的位向量。

[0421] • 当位向量与在当前页中先前遇到的位向量相同时(忽略平凡位向量 000...000 及 111...111),码 ≥ 2 。此编码因此使用所述页中先前遇到的码的历史。所述码因而实际上按所有先前遇到的码的历史给出索引。

[0422] 不同于此霍夫曼码,仅在于历史中未找到非平凡位向量的情况下,需要编码更多信息(码 = 2)。在此情况下,紧接在霍夫曼码 = 2 之后,编码:

[0423] • 否定位

[0424] • 用以通过 N 个位的块编码 n 个区的位向量的若干霍夫曼码(N 及 n 给出于地图标题中)。举例来说,使用 11 个位的块编码 100 个区(99 位位向量)需要编码 9 个霍夫曼码($9 \times 11 = 99$)。

[0425] 由于大多数位向量主要含有 0 或含有 1,因此否定位指示所述位向量是否被存储为否定的。这使得能够在霍夫曼树中存储决大多数含有 0 的码(因此,改进块的霍夫曼编码)。仅在块的大小小于区的数目的情况下存在否定位,此为实际上在等级 0 处的情况,但在等级 1 或 2 处,可能在仅 1 个块中编码整个位向量,所以否定位为不需要的。

[0426] 如果存在 100 个区; $N = 100$ (因此,99 位的位向量),那么第一块编码针对目的地区 1 到 11 的位,第二块编码区 12 到 22,等等。在第一块中,LSB(0x1)对应于目的地区 1,下一位(0x2)对应于区 2,下一位(0x4)对应于区 3,等等。

[0427] 对于使用历史的位向量来说,历史阵列的深度为页中的先前遇到的相异位向量的数目(不考虑平凡位向量 000...000 及 111...111)。不管历史向量含有 0 个元素、1 个元素、2 个元素、3 个元素等等,使用不同霍夫曼树。使所述霍夫曼树倍增为可接受的,因为所有霍夫曼树均较小,且因而不需要相当大的存储装置:

[0428] • 当历史具有 0 个元素时,霍夫曼树具有 3 个码:针对 000...000 的 0、针对 111...111 的 1、针对新位向量的 2。

[0429] • 当历史具有 1 个元素时,霍夫曼树具有 4 个码:针对 000...000 的 0、针对 111...111 的 1、针对新位向量的 2、针对与历史中的元素 #0 相同的位向量的 3。

[0430] • 当历史具有 2 个元素时,霍夫曼树具有 5 个码:针对 000...000 的 0、针对 111...111 的 1、针对新位向量的 2、针对与历史中的元素 #0 相同的位向量的 3、针对与历史中的元素 #1 相同的位向量的 4。

[0431] • 等等。

[0432] 预期位向量页的大小较小(例如,2ⁿ个始发节点),所以预期霍夫曼树的数目较小。

[0433] 然而,有可能将霍夫曼树的大小限于最大值:例如,每当历史含有 H 以上个元素

时,就将使用单个霍夫曼树(值H存储在地图标头中)。

[0434] 此编码仅编码每一页中的相异位向量+一些码。

[0435] 在较大的索引页的情况下,所述编码在大小上较有效率,但代价是为了使用位向量来进行路线选择目的而减慢了解码(页中有更多位向量要解码)。

[0436] 统计数据

[0437] 此处为在按 254 个区(1 等级)编码比荷卢经济联盟(Benelux)时的文件格式的详细统计数据。使用以下输入参数:

[0438] 每位向量块的位数目:11

[0439] 每位向量页的节点数目: $2^4 = 16$

[0440] 每区页的节点数目: $2^9 = 512$

[0441] 提供统计数据以在地图大小方面给出地图形式的概念,且基于一些实际数据说明地图格式的描述:

[0442]

---全局统计计数器:	
节点计数	1598632
线路计数	1598632 (100.000%)
跳过具有 1 条线路的节点周围的线路	220180 (13.773%)
跳过具有 2 条线路的节点周围的线路	727138 (45.485%)
---等级=[0]处的统计数据:	
地图计数器	87437736 (100.000%)
经编码平凡 arp 旗标 000...000	1310914 (31.651%)
经编码平凡 arp 旗标 111...111	913348 (22.052%)
历史中的经编码 arp 旗标	362432 (8.751%)
不在历史中的经编码 arp 旗标	607717 (14.673%)
否定块	235171 (5.678%)
地图大小(以位计)	87437736 (100.000%)
全局标头	496 (0.001%)
霍夫曼树	28808 (0.033%)
区二进制大型对象	52664 (0.060%)
区页信息	56216 (0.064%)
arp 旗标页信息	2497880 (2.857%)
可变大小记录	84801672 (96.985%)
节点周围的线路计数	2847844 (3.257%)
节点区 ID	2112451 (2.416%)
arp 旗标	79841370 (91.312%)
码平凡 000...000	1689322 (1.932%)
码平凡 111...111	1826696 (2.089%)
在历史中找到的码	1668053 (1.908%)
未在历史中找到	74657299 (85.383%)

[0443]

未在历史中找到的码	1463183 (1.673%)
否定位.....	607717 (0.695%)
块	72586399 (83.015%)

[0444] 所有大小以位计。总地图大小为 87, 437, 736 个位 (10, 929, 717 个字节)。

[0445] 缩进反映了层级。可变大小记录信息为目前为止最大的一条信息 (地图大小的 96. 975%)。在可变大小记录中, 子项目 (缩进的) 给出更多细节。位向量为目前为止可变大小记录中要存储的最大的一条信息 (91. 312%)。且在位向量中, 存储在历史中尚未遇到的非平凡位向量构成了地图的最大部分 (83. 015%)。

[0446] 关于霍夫曼树的统计数据

[0447] 此节给出在按 255 个区 (即, 针对上文所展示的地图数据) 编码比荷卢经济联盟时的霍夫曼树的统计数据。

[0448] 在每一节点周围的道路段的数目的霍夫曼树

[0449]

--- [霍夫曼树: Nr个线路] ---			
位:	1 值:	3 码	0
位:	2 值:	2 码	10
位:	3 值:	1 码	110
位:	4 值:	4 码	1110
位:	5 值:	5 码	11110
位:	6 值:	6 码	111110
位:	7 值:	7 码	1111110
位:	7 值:	8 码	1111111

[0450] 大多数节点具有 3 个所附接的道路段, 但在所述霍夫曼树中的第 2 及第 3 位置中, 发现具有 2 个及 1 个所附接道路段的节点 (其并非决策节点)。

[0451] 非平凡位向量块的霍夫曼树

[0452] 此为最大的霍夫曼树, 因为平凡位向量的存储块为目前为止最大的地图大小贡献因素 (在比荷卢经济联盟 255 区的实例中, 为 83. 015%)。

[0453]

--- [霍夫曼树: NonTrivialArpFlagBlock (非平凡Arp旗标块)] ---			
位:	1 值:	0 码	0
位:	6 值:	1 码	100000
位:	6 值:	2 码	100001
位:	6 值:	4 码	100010
位:	6 值:	8 码	100011

[0454]

位:	6 值:	16 码	100100
位:	6 值:	32 码	100101
位:	6 值:	64 码	100110
位:	6 值:	512 码	100111
位:	6 值:	1024 码	101000
位:	7 值:	128 码	1010010
位:	7 值:	256 码	1010011
位:	7 值:	384 码	1010100
位:	8 值:	5 码	10101010

...裁剪, 过大...

位:	24 值:	1534 码	11111111111111111111011
位:	24 值:	1717 码	11111111111111111111100
位:	24 值:	1741 码	11111111111111111111101
位:	24 值:	1830 码	11111111111111111111110
位:	24 值:	1973 码	11111111111111111111111

[0455] 存储全部由 0 构成的块为最频繁模式, 且根据上文的霍夫曼树以仅 1 个位来编码 (这意味着即使平凡位向量 000...000 不按块编码, 50% 或 50% 以上的块还编码值 0)。这是因为大多数非平凡位向量含有

[0456] • 大部分 0 (及少数 1)

[0457] • 或大部分 1 (及少数 0)

[0458] 所述编码方案否定 (~) 含有大部分 1 的位向量, 所以最后, 位向量的编码块大部分情况下仅以 1 位编码含有 000...000 的块。次频繁的块为仅设定 1 位的块 (1、2、4、8、16、32、64...)。其具有或多或少相同的频率, 因此具有相同 (或几乎相同) 数目个位。

[0459] 本地区 ID 的霍夫曼树

[0460] 由于将区的列表按频率存储在每一页中, 因此可看到, 与其它所在地的区 ID 相比, 存储本地区 ID 0 占用较少位 (事实上仅 1 位)。不同霍夫曼树对应于具有 3 个区、4 个区、5 个区等等的页。

[0461]

```

--- [ 霍夫曼树: 区_0 ] ---
位:    1 值:  0 码          0
位:    2 值:  1 码          10
位:    2 值:  2 码          11

--- [ 霍夫曼树: 区_1 ] ---

```

[0462]

位:	1	值:	0	码	0
位:	2	值:	1	码	10
位:	3	值:	2	码	110
位:	3	值:	3	码	111

--- [霍夫曼树: 区_2] ---

位:	1	值:	0	码	0
位:	2	值:	1	码	10
位:	3	值:	2	码	110
位:	4	值:	3	码	1110
位:	4	值:	4	码	1111

--- [霍夫曼树: 区_3] ---

位:	1	值:	0	码	0
位:	2	值:	1	码	10
位:	3	值:	2	码	110
位:	4	值:	3	码	1110
位:	5	值:	4	码	11110
位:	5	值:	5	码	11111

...裁剪...

[0463] 位向量历史码的霍夫曼树

[0464] 码 0 (表示平凡位向量 000...000) 为最频繁的 (且在大多数情况下, 仅以 1 位编码)。码 1 (平凡位向量 111...111) 就为次频繁的 (且仅以 1 位编码)。再次频繁的码 (2) 是针对按块编码的非平凡位向量。其它码 (> 2) 针对在历史中找到的位向量。

[0465]

--- [霍夫曼树: Arp 旗标_0] ---

位:	1	值:	0	码	0
位:	2	值:	1	码	10
位:	2	值:	2	码	11

--- [霍夫曼树: Arp 旗标_1] -----

位:	1	值:	0	码	0
位:	2	值:	1	码	10
位:	3	值:	2	码	110
位:	3	值:	3	码	111

--- [霍夫曼树: Arp 旗标_2] ---

位:	1	值:	0	码	0
位:	2	值:	1	码	10
位:	3	值:	2	码	110
位:	4	值:	3	码	1110
位:	4	值:	4	码	1111

[0466]

--- [霍夫曼树: Arp旗标_3] ---				
位:	1	值:	0	码 0
位:	2	值:	1	码 10
位:	3	值:	2	码 110
位:	4	值:	3	码 1110
位:	5	值:	4	码 11110
位:	5	值:	5	码 11111
--- [霍夫曼树: Arp旗标_4] ---				
位:	1	值:	0	码 0
位:	2	值:	1	码 10
位:	3	值:	2	码 110
位:	4	值:	3	码 1110
位:	5	值:	4	码 11110
位:	6	值:	5	码 111110
位:	6	值:	6	码 111111
--- [霍夫曼树: Arp旗标_5]---				
位:	2	值:	0	码 00
位:	2	值:	1	码 01
位:	2	值:	2	码 10
位:	4	值:	3	码 1100
位:	4	值:	4	码 1101
位:	4	值:	5	码 1110
位:	5	值:	6	码 11110
位:	5	值:	7	码 11111
...裁剪...				

- [0467] 输入参数对地图大小的影响
- [0468] 存在若干输入参数,其控制图 11 中所展示的文件格式且可影响地图大小。对所述参数的调整可为地图大小与存储器使用或解压缩速度（视参数而定）之间的折衷。
- [0469] 本发明的实施例可使用以下输入参数：
- [0470] • 区页大小
- [0471] • 位向量页大小
- [0472] • 位向量块大小
- [0473] • 位向量霍夫曼编解码器计数
- [0474] • 区霍夫曼编解码器计数
- [0475] 位向量块大小的影响
- [0476]

值	地图大小 (位)
-----	-----
4	35548192
5	33648792
6	32290344
7	30853616
8	31103200
9	30436696 (默认)
10	30051792
11	29266784
12	28934696

[0477] 增加位向量的霍夫曼块大小会改进地图压缩。块大小越高,压缩越好。但增加块大小是以需要更多存储器来存储具有 2^n 个值的较大霍夫曼树为代价的。上文的表说明此情况。

[0478] 当引入优化以重新映射每源区的区 ID 时,预期此参数的影响变得更重大:在使用较大的位向量块大小时,此优化将有希望导致显著的地图大小减小。

[0479] 位向量页大小的影响

[0480]

值	地图大小 (位)
-----	-----
2^1	55563944
2^2	42502936
2^3	34898840
2^4	30436696 (默认)
2^5	27389952
2^6	25165032
2^7	23635936

[0481] 增加页大小有助于更好地压缩地图。但遗憾的是,较大的页使路线选择方法对文件格式中的数据的解压缩减缓,因为存取随机道路段的位向量需要解码页中的所有道路段。上文的表说明此情况。

[0482] 位向量霍夫曼编解码器计数的影响

[0483]

值	地图大小 (位)
-----	-----
1	30866920
2	30748024
3	30634168

[0484]

5	30504504
7	30467944
9	30436696 (默认)
11	30423688

[0485] 增加位向量霍夫曼编解码器的数目有助于稍微增加压缩,且在上文的表中说明此情况。增加所述值几乎不存在缺点,因为无论如何所述霍夫曼树为较小的。增加到超过 9 个霍夫曼树(默认值)不提供任何显著改进。在位向量页较大的情况下,增加此参数可能更有效。

[0486] 聚结及重新排序位向量中的位

[0487] 位向量具有模式。每一源区(即,存储位向量的节点的区)的所述模式显著不同。可通过存储每一源区的小的转译表来减少用以存储 N 位的位向量的位数目。这些表执行 2 个功能,其进一步描述于此节中:

[0488] • 位聚结

[0489] • 位重新排序

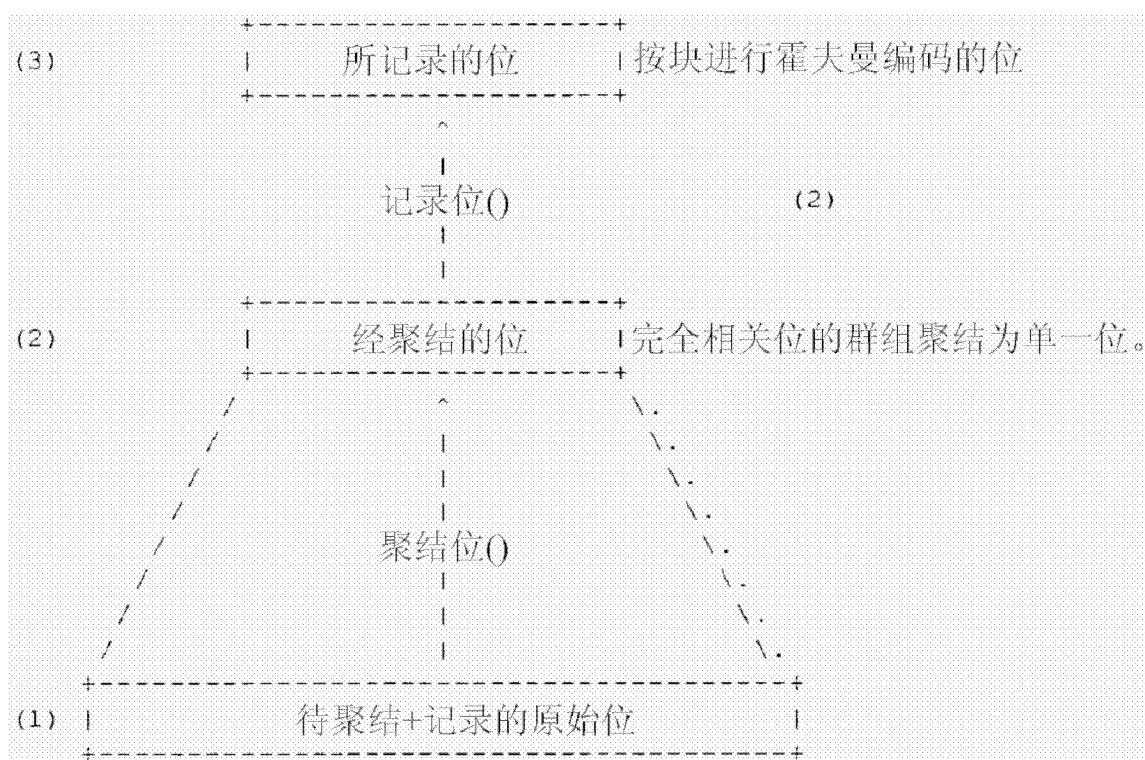
[0490] 可直观地将此想法理解如下:当在西班牙时,清楚的是,通向瑞典的道路段(针对目的地区瑞典的位=1)极可能还通向挪威(即,针对目的地挪威的位是也为 1)。如果另一道路段不通向瑞典(位=0),那么在大多数情况下其也不通向挪威。所以当在西班牙时,目的地区瑞典及挪威的位向量位的值几乎总是相等。事实上,对于许多目的地区来说,其甚至总是严格地相等。哪些位与哪一位充分相关很大程度上取决于始发区。

[0491] 举例来说,当在芬兰时,目的地区挪威及瑞典的位的相关性小得多。另一方面,在芬兰,目的地区西班牙及葡萄牙的位很可能 100% 相关(或至少极接近 100%)。

[0492] 位聚结利用了一些位总是相等(完全相关)的性质。所述位可聚结成单个位。聚结位减少了在每一区中要编码的位数目。

[0493] 位重新排序利用了一些位的相关性相当高(但并非 100% 相关)的性质,以便以优化对位向量的霍夫曼编码(及/或减小霍夫曼树的大小)的方式混洗(shuffle)位。

[0494]



[0495] 在计算用以聚结位及将位重新排序的表之前,计算所有始发区中的所有位对的相关性。相异对的数目为:

$$[0496] \quad C(N, 2) = N! / (2! * (N-2)!) = N * (N-1) / 2$$

[0497] ... 其中N为区的数目。因此当区的数目较高时,此数目相当高。计算所有位相关性为产生图11中所展示的文件格式的方法的最慢部分。所述方法的复杂性为 $n * N * N$, 其中n为位向量的数目,且N为区的数目。对于每一区中的每一位对(即,位向量中的相异列的对),计算三维表:

[0498] 位相关性[始发区Id][位I][位J]。

[0499] 表中的每一条目含有4个字段的结构,其计数:

- [0500] • 始发区Id的所有位向量中位I=0且位J=0的次数
- [0501] • 始发区Id的所有位向量中位I=0且位J=1的次数
- [0502] • 始发区Id的所有位向量中位I=1且位J=0的次数
- [0503] • 始发区Id的所有位向量中位I=1且位J=1的次数

[0504] 虽然在处理器时间方面计算成本高昂(且因此缓慢),但可易于将此过程并行化,因为计算每一始发区的位相关性完全独立于其它始发区。

[0505] 本发明的实施例使用多执行线程以加速所有位的相关性的计算以高效地使用SMP(对称多重处理)机器。在一个系统中,单CPU机器计算具有300个区的比荷卢经济联盟的位相关性花费约8分钟。但并行关系(parallelism)很好地按比例调整,且在启用线程且使用4个CPU时,计算位相关性就花费2分钟(为原来的4分之一)。

[0506] 位聚结

[0507] 当若干位完全相关(即,总是全部为0或全部为1)时,可在不释放任何信息的情况下将其聚结成仅1个位。将在给定区中的完全相关的位集合称为“群组”,那么给定区的位向量由若干群组构成。如果N位的位向量由n个群组构成,那么编码N位的位向量使用

n 个位 + 较小的表来指示每一区中的哪些位为相等的。

[0508] 此位聚结具有由此产生的文件的大小以无损失方式减小的优点。此优点很可能随着区的数目增加而增加,因为更多位可能被聚结。所以地图大小随着区的数目以次线性方式增加。

[0509] 在一个实施例中,获得以下数据:

[0510]

	最小群组数目	平均群组数目	最大群组数目
比荷卢经济联盟的 255 个区	12	84	152
比荷卢经济联盟的 300 个区	13	90.017	163

[0511] 因而,在具有 255 个区的比荷卢经济联盟的实例中,存在至少 1 个甚至在霍夫曼编码之前也仅具有 12 个群组(即,仅需要 12 个位来编码其 255 位的位向量(即,位向量的长度与区的数目相同))的区(且因此,在霍夫曼编码之后具有甚至更少群组)。

[0512] 平均来说,对于比荷卢经济联盟的 255 个区,区需要 84 个位。在此实例中,最差的区在霍夫曼编码之前需要 152 个位(152 个群组)。

[0513] 作为另一实例,且以上述 255 个区的比荷卢经济联盟的实例中的区 Id = 3 为例,其具有从经编码数据取得的以下 18 个群组。

[0514]

```
区 Id=[3]具有[18]个群组:
#1 ( 0 )
#1 ( 1 )
#2 ( 2 )
#141 ( 3 4 5 7 8 11 12 15 16 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 36
      37 38 39 40 42 43 44 45 46 50 51 59 60 62 63 66 67 69 71 72 73
      74 75 76 80 81 82 83 84 86 88 89 90 95 96 97 98 100 101 102 103
      104 105 106 108 110 112 113 119 120 123 127 129 130 133 135 136
      139 142 143 147 148: 149 150 151 154 155 156 157 159 160 165 166
      167 171 172 173 174 176 177 178 179 181 182 183 190 192 200 203
      205 207 210 211 212 213 214 215 218: 219 220 221 222 226 227 229
      235 237 241 242 243 244 245 247 249 250 252 )
```

[0515]

```

#1 ( 6 )
#1 ( 9 )
#1 ( 10 )
#30 ( 13 14 35 48 49 57 68 85 93 109 115 117 118 128 131 134 138 153 158
      164 168 169 187 189 195 209 217 224 238 239 )
#59 ( 17 18 47 53 54 56 65 91 92 99 107 114 116 121 124 125 126 132 137
      140 141 144 145 146 152 161 162 163 170 175 180 184 185 186 191
      193 194 196 198 199 201 202 204 206 208 216 223 225 228 230
      231 232 233 234 236 240 246 248 251 )
#1 ( 34 )
#5 ( 41 78 79 111 197 )
#3 ( 52 77 87 )
#1 ( 55 )
#3 ( 58 61 94 )
#1 ( 64 )
#1 ( 70 )
#1 ( 122 )
#1 ( 188 )

```

[0516] 每一道路段表示一群组（区 $Id = 3$ 的 18 个群组）。括号中的数目为在每一群组中被聚结的位索引。# 之后的数字为每一群组中的位数目。

[0517] 在此实例中，1 个群组为极大的且含有多达 141 个区。这是寻常的。一般来说，与地图的中间的区相比，地图的边缘上的区聚结更多位。

[0518] 在此实例中，位的数目已平均减小为原来的 $\sim 1/3$ ($\sim = 255/84$)。当然，这并不意味着地图大小减小为原来的 $1/3$ ，因为要编码的剩余位比原始位具有更多熵（所以更难以通过霍夫曼块来编码）。但位聚结位仍可减小地图大小（可能显著地减小地图大小）。

[0519] 当聚结每一区中的位时，每一区中的位向量因此具有不同数目个位（但给定始发区的位向量具有相同数目个位），如图 15 中所说明。阴影说明每一位向量的始发区 ID。

[0520] 一旦已计算出表的位相关性 (bitCorrelation)，识别完全相关位的群组就相当容易且快速。在其中找不到 01 或 10 模式的所有位对为完全相关的，且可聚结成 1 位。

[0521] 为了给出位聚结的另一实例，下表给出 6 个位向量（每一位向量具有 15 个位）的实例。这些位向量尚未被聚结。

[0522]

```

10011110111100
000001101110100
011001010010111
111110010001011
011000010000011
000000101100000
...
```

[0523] 这些 15 位的位向量可聚结成仅 4 个群组 – 因此，在霍夫曼编码之前每位向量有 4 个位。这展示于下表中：

[0524]	1	2	3	4	5	6	7	8	9	10	11	→	1+3+9	2+11	5+7	4+8+10
	1	00	11	1	1	0	11	1	1	1	00		1	0	1	1
	0	00	00	1	1	0	11	1	0	1	00		0	0	1	1
	0	11	00	1	0	1	00	1	0	1	11		0	1	0	1
	1	11	11	0	0	1	00	0	1	0	11		1	1	0	0
	0	11	00	0	0	1	00	0	0	0	11		0	1	0	0
	0	00	00	0	1	0	11	0	0	0	00	→	0	0	1	0

[0525] 所述表的右边部分的标题展示所述表的左边部分中的哪些位构成 4 位聚结向量。为了清楚起见,4 位聚结向量的第一位表示左边部分的第 1、第 3 及第 9 列。因而,由于在上述实例中左边部分(即,15 位原始向量)的第 1、第 3 及第 9 位总是相同,因此其可由上表的右边侧的第一列表示。

[0526] 因此,总之,左边侧与右边侧之间的一致性的表为如下:

[0527]	原始位	聚结位
	0	0
	1	1
	2	1
	3	0
	4	0
	5	3
	6	2
	7	1
	8	2
	9	2
	10	3
	11	0
	12	3
	13	1
	14	1

[0528] 聚结位有效地将附近目的地区及在从始发区的相同角度扇区中的目的地区分组,如图 16 中所说明。

[0529] 图 16 中的阴影指示被聚结在一起的区的群组。2 个图(即,图 16a 及 16b)展示从 2 个不同始发区:A 及 B 的观点聚结的相同区。哪些区被聚结在一起取决于始发节点的区 ID,如由所述图所展示。在此图中,8 个区被聚结成 4 个群组。从这些图中的始发区 A 或始发区 B 的观点来看,聚结位不同。注意垂直条状区(1300)如何被聚结,因为当从始发区 A 看时,其处于同一方向上。

[0530] 图 17 中进一步示范此情况,图 17 说明聚结位将附近目的地区及从每一始发区的观点来看在 +/- 相同角度扇区中的目的地区分组。

[0531] 在聚结位之后,每一区的位向量的要编码的位的数目不同。在(例如)比荷卢经济联盟的 255 个区的情况下,每一区的位向量的位数目为:

[0532]

始发区->	聚结位
1 ->	57
2 ->	93
3 ->	18
4 ->	12
5 ->	63
6 ->	75
... 裁剪...	
251 ->	21
252 ->	46
253 ->	117
254 ->	80

[0533] 所述方法为准确的（其并非试误式的）。因此，所述方法找到位的群组的理想数目。

[0534] 在一些实施例中，以无损失方式实施聚结位：仅当区总是具有相同位时才聚结所述区。但在其它实施例中，这可扩展为使得聚结位有损失；即，使得完全恢复原始数据为不可能的。这些有损失实施例可引入阈值：如果一对位的差异小于 X 倍，那么在位向量中的一些位向量中用 1 替换一个或一个以上 0 允许所述位向量被聚结的情况下，可聚结所述位。

[0535] 此有损失聚结的实例将为使 0000001100 与 0000001110 聚结。此方法的优点为，数据被更好地压缩，但缺点为，使用所述位向量的路线选择方法将评估其它道路段，因为额外的 1 将指示道路段为最快速路线的部分。

[0536] 在无损失聚结的情况下，阈值为 0。增加所述阈值将允许聚结其它区，而代价是将少数几个位 1 引入位向量中。但如果阈值被保持为低的，那么其将对可使用所述数据格式执行路线选择的速度几乎没有影响。举例来说，如果在数千个位向量中 2 个位几乎总是相同（除了（例如）仅 1 个位向量之外），那么变更此唯一位向量的不同处以使得有可能聚结更多位可能是可接受的。阈值越高，可预期越多地图压缩。

[0537] 位重新排序

[0538] 在一些实施例中，一旦位已被聚结（如上文所描述），就可将位重新排序。将位重新排序并不（在霍夫曼编码之前）减少要编码的位的数目，而是有助于使得霍夫曼编码更有效率，从而减小地图大小，且还可有助于减小位向量霍夫曼码中的相异码的数目，从而降低霍夫曼树的存储器要求。

[0539] 与位聚结不同，位重新排序为试误式方法，所以其可找到一种较好的重新排序，但其未必可找到最佳的重新排序。

[0540] 按块来编码位向量。块大小为可定制的，且存储在地图标头 800 中。在下文给出的解释中，且作为实例，将块大小设定为 11 个位。位向量中的位的数目可能不可被 11 除尽，所以最后的块可能小于 11 个位。最后的块使用与具有 11 个位的完整块所使用的霍夫曼树不同的霍夫曼树。编码每一位向量因此编码：

[0541] • 使用针对 11 位块（在此实例中）的霍夫曼码的若干完整块

[0542] • 当剩余的位小于 11 个时可使用另一霍夫曼码的最后 1 个块。

[0543] 以下图片描绘要按块进行霍夫曼编码的位，其中位向量在 2 个区（其在聚结之后具有不同数目个位）中：

[0544]

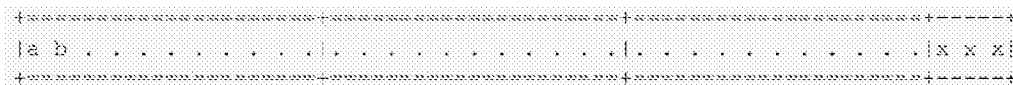


[0545] 每一区均具有霍夫曼树将很可能在存储器方面成本过于高昂。所以在一些实施例中,所有区共享同一霍夫曼树。由于在每一区中相关位为不同的,因此将每一区中的位重新排序以使得在所有区中相关位被系统地放置于相同位置处是有利可图的。所以跨越所有区共享霍夫曼树变得更有效率。

[0546] 将用来将每一区的位重新排序的表存储在地图数据中。注意,无需编码单独的用以使位聚结的表及用以将位重新排序的表,而是可改为仅编码 1 个表,其为两种变换(聚结+重新排序)的合成物。

[0547] 将完整块中的位重新排序的方法如下进行:对于每一区,找到最相关的位对。由于完全相关的位已被聚结,因此剩余位均为不完全相关的。然而,一些位对比其它位对相关得多。在第一完整块中,将最相关的位对(a, b)重新映射到位 #0 及 #1:

[0548]



[0549] 由于分组最相关的位对,因此完整块的霍夫曼树具有较少相异码(此具有使用较少存储器来存储数据的优点),且统计数据为较偏斜的(使得霍夫曼编码更有效率)。在绝大多数情况下,前 2 个位(例如)含有 00 或 11,但几乎从来不含有 10 或 01(其它位也是如此),而不是含有位的随机序列。在未将位重新排序的情况下,前 2 个位将含有所有种类的模式 00、01、10、11。

[0550] 在将前 2 个位重新映射之后,方法接着找出下一最相关的(c, d)位对,且重新映射其以将其存储在第二块的前 2 个位中:

[0551]



[0552] 接着,方法再一次找出下一最相关的(e, f)位对,且重新映射其以将其存储在第

三块的前 2 个位中：

[0553]

```
+-----+
|a b . . . . . |c d . . . . . |e f . . . . . |x x x|
+-----+
```

[0554] 当到达最后的完整块时,所述方法返回到第一个完整块且重新映射下一最相关的位对 (g, h)。如果若干对具有相同相关性,那么连结断路器 (tie breaker) 选择与同一块中的前一对最相关的对 (即,与 (a, b) 最相关的对) :

[0555]

```
+-----+
|a b g h . . . . . |c d . . . . . |e f . . . . . |x x x|
+-----+
```

[0556] 算法方法如上述继续进行,直到完整块中的所有位对均已被重新映射为止 :

[0557]

```
+-----+
|a b g h m n s t y z . |c d i j o p u v A B . |e f k l q r w x C D . |x x x|
+-----+
```

[0558] 在此实例中,由于块大小具有奇数个位 (11 个位),因此每一完整块中仍存在未经重新映射的位。所述方法接着找出与“z”最相关的位,且将其存储在第二块中。接着找出与 B 最相关的位且将其存储在第二块中,等等,直到所有完整块均被重新映射为止 :

[0559]

```
+-----+
|a b g h m n s t y z E |c d i j o p u v A B F |e f k l q r w x C D G |x x x|
+-----+
```

[0560] 此时,完整块中的所有位被重新映射。所述方法接着重新映射最后的块中的位,试图分组最相关的位对 (如针对完整块所进行)。将最后的块中的位重新排序可有助于减小地图大小,但其并不像将完整块中的位重新排序一样有帮助,这是由于 2 个原因 :

[0561] • 完整块为更重要的。在此实例中,每一码对于完整块使用 3 个霍夫曼码,而其对于最后的块仅使用 1 个霍夫曼码,所以完整块比最后的不完全的最后块对总地图大小的贡献更大是正常的,且优化完整块的霍夫曼编码为更有用的。

[0562] • 由于已经挑选完整块中的所有最相关的位来重新映射其,因此最后块中剩下的待重新映射的位为较不相关的。所以最后块中的位的熵因此高于完整块中的位的熵。换句话说,最后块的霍夫曼编码不如完整块的霍夫曼编码有效率。

[0563]

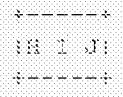
```
+-----+
|a b g h m n s t y z E |c d i j o p u v A B F |e f k l q r w x C D G |H I J|
+-----+
```

[0564] 应记住,对于所有区的所有完整块使用同一霍夫曼树。在上述实例中编码位向量因此以相同霍夫曼码编码所有完整块,且最后以不同霍夫曼编解码器编码最后的块 :

[0565]

```
+-----+
|a b g h m n s t y z E|
+-----+
|c d i j o p u v A B F|
+-----+
|e f k l q r w x C D G|
+-----+
```

(使用同一霍夫曼编解码器的 3 个完整块)

[0566]  (使用另一霍夫曼编解码器的 1 个最后块)

[0567] 当看到上述图时,所述方法重新映射每一块的第一位(而不是在跳到第二块之前重新映射第一块中的所有位)的原因应更清楚。由于针对所有完整块使用同一编解码器,因此需要使所有块的所有码尽可能相同。如果将重新映射第一块中的所有位,接着重新映射第二块中的所有位(等等),那么每一块将具有十分不同的模式:第一块将具有最相关的位,第二块将具有较不相关的位,第三块将具有更不相关的位,等等。将有可能为完整块的每一列创建若干霍夫曼编解码器,但相信所述做法在存储器方面成本过于高昂。目前为止所概述的方法在使所有完整块共享同一霍夫曼编解码器的同时运作良好。

[0568] 可能的其它优化

[0569] 页的页

[0570] 位向量页信息存储增量字段,其用以找出每一位向量页的开始偏移。使用偏移的道路线性内插器(linear interpolator)来存储增量字段。但所述偏移并非非常道路线性的,因为小节点 ID(等级 0,高速公路)周围的位向量比高节点 ID(等级 5,目的地道路)周围的位向量需要更多位。

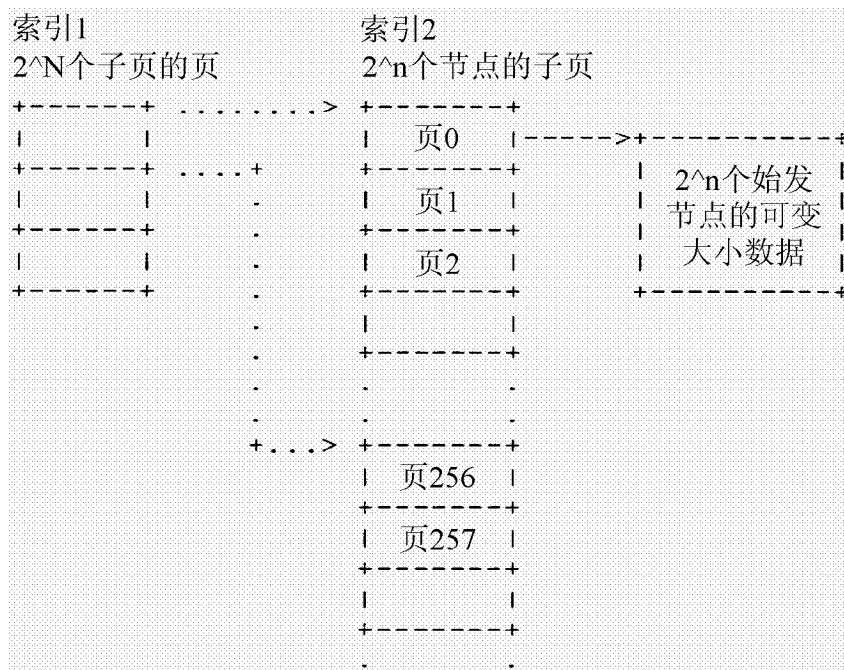
[0571] 位向量页信息的编码不像可能期望地一样有利,因为内插器不能精确地预测实际偏移。通过改进内插器,将有可能改进位向量页信息表的编码以使其更有效率。

[0572] 本发明的一些实施例可使用具有 2 个等级而不是仅具有 1 个等级的位向量页(及可能地,区页)。可将用以分组数据的具有 2^n 个节点的页(称其为子页)分组成具有 2^N 个子页的页。

[0573] 因而,将按页而不是全局地(在标头中)存储线性内插参数。举例来说,索引等级 2 可能早先将 $2^4 = 16$ 个节点分组于子页中,且索引 1 可能将 $2^{10} = 1023$ 个所述子页分组于页中。接着针对 $1024 \times 16 = 16K$ 个节点而不是对全部节点计数(在西欧及中欧的地图中的 40,000,000 个节点)发生线性内插,所以对可变大小偏移的线性内插变得精确得多,且索引 2 中的增量字段因此较小。

[0574] 在页较大的情况下,额外索引 1 的大小较小(小到足以装入存储器中)。能够装入装置的存储器内是有利的,因为其不应减缓使用所述数据的路线选择。可接着针对索引 1 表中的每一条目存储平均页大小,而不是将平均页大小存储在标头中。

[0575]



[0576] 交错位向量页

[0577] 如上文所描述,页偏移的内插器为不精确的,因为重要道路(许多非平凡旗标)及次要道路(许多平凡旗标)的位向量是不同的。使内插器更线性的一种简单方式为使不同网络等级的页交错,且这可在本发明的实施例中使用。

[0578] 以上实施例中所述的文件存储以下页:

[0579]

```
#0
#1
#2
#3
#4
...
#n-3
#n-2
#n-1
```

[0580] 在可能更高效的其它实施例中,有可能以交错的方式存储如下:

[0581]

```
#0
#n-1
#1
#n-2
#2
#n-3
#3
#n-4
...
```

[0582] 为了存取页 #x(例如由路线规划应用程序),通过加载页 #x' 来存取所述页,其中:

[0583] • $x' = 2 * x$ (其中 x 为偶数)

[0584] • $x' = 2*(x-(n-1))$ (其中 x 为奇数)

[0585] 此实施例应为有利的,因为其将使索引的大小每页减小若干个位。然而,数据可能对于高速缓存来说分组地较差,这可能减缓数据存取(在文件系统高速缓冲存储器中较少命中)。

[0586] 不存储所有节点的区 ID

[0587] 不需要存储无出口的路段的节点及具有 2 个所附接道路段的节点的区 ID。对于路线选择来说,可忽略这些节点。进入这些节点中的一者可被变换成进入其邻居决策节点。

[0588] 在页等级处及 / 或节点等级处存储额外信息

[0589] 查看地图数据,存在许多仅含有平凡位向量 000...000 或 111...111 的位向量页。一些实施例可能针对每一页存储 1 位以标记所述页,那么存储所述页中的位向量可为更有效率的,因为对于每一位向量仅需要单个位来指示其为 000...000 还是 111...111。这不仅将减小仅含有平凡位向量的页的大小,而且还将使具有非平凡位向量的页的位向量码的霍夫曼树得到更好的优化(因为所述码的频率将在百分比方面显著增加,所以用来指示非平凡位向量的位数目将减小)。在较精细网络等级(例如,等级 3)中,大多数页仅含有平凡位向量,所以在所述页的约一半页中可能每页仅存在 1 个位向量。

[0590] 仅存储决策节点处的位向量

[0591] 如上文所论述,一些实施例可能不存储具有 1 或 2 个所附接道路段的节点的位向量。然而,其它实施例可为更积极的,且将所述想法推广为仅存储决策节点周围的位向量。

[0592] 引入决策节点及决策道路段的概念,因为其可在编码地图数据的过程中为有利的:不需要编码非决策道路段的位向量,如现在所论述。

[0593] • 决策节点为其中存在用于路线选择的进入道路段的节点,使得存在离开所述节点(而不进行 U 形转弯)的多个选择。

[0594] • 非决策节点是并非决策节点的节点。所以不管路线选择来自哪里,总是仅存在一条离开所述节点的道路。

[0595] • 决策道路段为用以离开决策节点的合法的道路段。

[0596] • 非决策道路段是并非决策道路段的道路段。

[0597] 所有决策道路段 s 因此在决策节点周围。但并非在决策节点周围的所有道路段 s 均为决策道路段。在非决策节点周围的所有道路段为非决策道路段。

[0598] 将仅编码决策道路段的位向量。非决策道路段的位向量为隐含的(000...000 或 111...111),因为使用所述数据的路线选择技术可根据已存在于道路段中的信息而作出确定。

[0599] 如何确定节点是否为决策节点? 可将准则缩减为:

[0600]

```
isDecisionNode= (lineCount>=3) && (lineGoingOutCount >=2)
```

其中：

lineCount: 为附接到节点的线路的总数

忽略不可进行路线选择的线路类型（铁路、参考线路），

忽略在两个方向上封闭的线路且忽略禁止通行的道路（住宅区域）。

lineGoingOutCount: 为附接到节点的可合法选取以离开所述节点的线路的数目。

[0601] 选取一道路段以离开所述节点是否为合法是取决于道路段属性：

[0602] • 道路段类型（铁路及参考道路段始终为非法的）

[0603] • 道路的向前 / 向后流动（存储在道路段旗标中）

[0604] • 道路段旗标中的禁止通行属性（禁止通行道路段 s 不具有任何位向量）

[0605] 在一些实施例中，忽略非决策道路段可丢弃约略 40% 的道路段。已发现，此百分比为相当一致的而与地图无关。避免编码 40% 的位向量为有利的，但其节省了小于 40% 的地图大小，因为其主要移除平凡位向量。

[0606] 移除在具有小于 3 个所附接道路段的节点（虚拟节点）周围的位向量会移除非平凡位向量，所以针对此类别的非决策道路段的地图大小节省可比针对非决策道路段的地图大小节省大。另一方面，筛选需要解码器（例如，使用地图的路线选择方法）来解码道路段的道路段类型及道路段旗标，且对其应用逻辑以便弄清为隐含的位向量，这可减缓过程。

[0607] 理论上，实施例还可查看机动（即，转弯限制）来决定离开的道路段是否为合法的，但此技术增加复杂性。忽略机动意味着实施例可能编码比严格必需的位向量多的位向量，但实现了方法的简化。

[0608] 非决策节点的实例

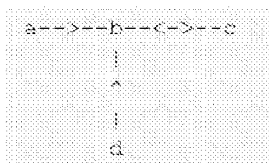
[0609]



[0610] 在此实例中，(b) 附接到可在两个方向上通行的 2 个道路段。(b) 不是决策节点，因为存在 ≤ 2 个所附接道路段。

[0611] 所以将不编码离开节点 (b) 的两个道路段的位向量。解码器可隐含地将其设定成 111...111。

[0612]

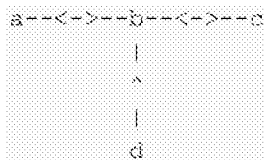


[0613] 此实例中的箭头 > 展示合法的流动方向。(b) 不是决策节点，因为仅存在一个出口。所以 (b) 周围的所有道路段均不需要位向量。

[0614] 不编码离开节点 (b) 的道路段 (b) - > (c) 的位向量，其将隐含地为 111...111。也不编码离开 (b) 的非法道路段的位向量，其将隐含地为 000...000。

[0615] 决策节点的实例

[0616]



[0617] (b) 为决策节点,因为当来自 (d) 时,存在选择:路线选择可朝向 (a) 或朝向 (c) 继续。

[0618] 注意,在此实例中,当来自 (a) 时,没有选择:路线选择仅可朝向 (c) 继续。但节点 (b) 仍为决策节点,因为存在当来自 (d) 时的至少一个选择,所以应存储节点 (b) 周围的 2 个决策道路段的位向量。

[0619] 存储道路段 (b) - > (a) 及道路段 (b) - > (a) 的位向量,因为其为决策道路段。

[0620] 不存储道路段 (b) - > (d) 的位向量,因为根据交通的向后 / 向前流动,选取此道路段为非法的。所述位向量隐含地为 000...000。

[0621] 对位向量进行“或”运算

[0622] 假设节点具有 3 个所附接道路段,且前 2 个经解码的道路段具有以下位向量:

[0623]

```
00000000000000000000
00000000000000000000
```

[0624] 那么第三个位向量不需编码,因为其只能为:

[0625]

```
11111111111111111111
```

[0626] 这只能在节点周围的道路段的位向量碰巧呈此次序时起作用:所有位向量为 000...000 且最后一个位向量为 111...111。实际上,看来此情况相当频繁地发生在较精细等级(例如,等级 0)网络(其为大多数道路段所在之处)中。

[0627] 将前 2 个位向量取为:

[0628]

```
000000000000000000110
000000000000000000010
```

[0629] 那么除了为未知的且需要以某种方式编码的 2 个位之外,第三位向量的所有位只能设定成 1。

[0630]

```
11111111111111111111??1
```

[0631] 由于在上述实例中,已经知晓大多数位,因此应可能使用此信息比编码整个位向量更有效率地编码所述 2 个未知位。因此,在此实例中,仅必需编码 2 个位。

[0632] 使用此性质的可能的快速解码方案将为通过对道路段中的所有先前的位向量进行“或”运算来计算当前节点的道路段中的所有经解码的位向量的位掩码。在使用与早先的实例相同的实例的情况下,如果节点具有 3 个所附接道路段,且如果先前 2 个道路段具有以下位向量:

[0633]

```
000000000000000000110
000000000000000000010
```


2	1	????????????
2	3	????????????
3	4	000000000000

[0655] 不存储在较精细网络等级处的节点的位向量

[0656] 除了在接近目的地或起点之处以外,路线选择几乎从不经过不重要的道路段网络等级(例如,网络等级 5)。因此,有可能不存储等级 5 处的位向量,且因而,对路线选择的影响最小(即使在等级 5 处的道路段的数目较大的情况下)。在大多数情况下,路线选择将仅探索少数在路线的开始或末端处的在最不重要的网络等级处的道路段,因为其将迅速地到达一些更重要的道路段网络等级,且关于所述节点的搜索加速数据将几乎总是告知选路器(router)跳过返回到最不重要的网络等级的导航段,并使用通向或继续停留在较重要的网络等级的导航段。这在最重要的网络等级处是成立的,因为在目的地仍然遥远时使用所述网络等级。

[0657] 除了丢弃最不重要的道路段(例如,等级 5)处的位向量之外,还可不编码所述网络处的区 ID。

[0658] 将位向量中的少数几个 0 变换成 1

[0659] 这也是有损压缩方案。

[0660] 如果位向量仅含有一个 0(或可能地,小于小阈值),那么在产生以下结果的情况下将其变换成 1 可为有可能的(即,将所述道路段设定为最快速路线的部分):

[0661] • 平凡位向量 111...111(因为以比编码非平凡位向量紧凑的方式来编码平凡位向量)

[0662] • 或已在当前位向量页的历史中发现的位向量(因为所述位向量也被更高效地编码)

[0663] 将位向量中的 0 变换成 1 并不影响路线选择结果,其仅会由于使路线选择考虑更多道路段(因为其现在被设定为最快速路线的部分)而使路线选择较慢。然而,本发明的一些实施例可使用此方法—尤其是在地图节省为较大的,同时在路线选择速度方面的性能影响(performance hit)较小的情况下。

[0664] 路线选择

[0665] 图 18 展示地图 1600,其覆盖一区域且具有开始节点 1602 及目的地节点 1604,以及已由现有技术 A*搜索方法探索的多个道路段。由虚线 1606 展示选定路线。

[0666] 图 19 展示地图 1650,其与图 18 的地图覆盖相同的区域且还展示相同的开始节点 1602 及目的地节点 1604。所述图还醒目提示使用本发明的实施例而探索的道路,其利用了与用以产生图 18 的路线相同的道路网络及相同的准则(例如,两者均乘汽车行进,希望将速度用作要最小化的成本函数,等等)。再次以虚线道路段 1606 展示由所述方法选定的路线,且应注意,所述道路段是与图 18 中所计算的线路相同。因而,应注意,通过本发明的实施例计算的路线可为所谓的数学上准确的,且可找到关于已被选择以最小化的给定成本模型的最好/最佳路线。因此,将明白,当与现有技术相比时,由本发明的实施例探索的道路段为显著较少的。因而,与现有技术 A*方法相比,本发明的实施例的方法较迅速,且通常显著较迅速。

[0667] 还应注意,随着路线 1606 逼近目的地 1604,与开始相比,更多路线被探索;即,当

路线 1606 前进超过点 1652 时,探索除了最低成本路线之外的道路。此情况的一种解释为,所述方法在点 1652 处已从等级 0 区切换到等级 1 区。应进一步看出,在目的地 1604 附近探索其它道路段。此外,可通过注意到路线规划方法将在此附近从等级 1 切换到等级 2 来解释此情况。

[0668] 因此,当进行长程路线选择时,通常将仅存在一条最快速路线。然而,随着方法切换到较精细等级(例如,从等级 0 到 1),可能存在被标记为“最低成本”且因此需要被探索的一条以上路线。

[0669] 一旦地图数据已被创建且位向量被计算并存储,其就可用于计算两个点之间的路线。

[0670] 图 20 展示通过路线规划应用程序探索网络的现有技术 A*方法,其中在一节点处作出应进一步探索哪一道路段的决策。为了实现此,将行进到下一节点的成本添加到从所述下一节点移到目标的估计成本。接着进一步探索从给定节点起的具有最小成本的路径。所述方法保持随着所述方法的进行目前为止已招致的成本的总和,且在每次反复时考虑最小成本。

[0671] 举例来说,在节点 B 处开始,可看出,节点 C 与 D 两者均与 B(开始节点)相距 2 个成本单位。然而,从节点 D 起,到达节点 F 的估计成本为 3 个单位,且从节点 C 起,到达节点 F 的估计成本为 5 个单位。因此,所述 A*方法将比探索到节点 C 的路线更为优先地探索到节点 D 的路线,因为对于到节点 D 的道路段来说,到下一节点的成本加到目标的估计成本较小(即, $BC = 2+5 = 7$,而 $BD = 2+3 = 5$)。

[0672] 当考虑使用本发明的实施例的最小成本位向量来执行路线规划时,每一道路段的位(即,图 10 的列 704)识别哪些道路段是到目的地区的最小成本路径的部分的候选者。因而,将关于图 20 所描述的 A*方法修改成图 21 中所展示的方法。

[0673] 通常,使用位向量执行路线规划,如关于图 21 所描述。然而,本发明的一些实施例可允许在各种情况下使路线规划退回到 A*(或其它现有技术方法)。不同实施例可利用以下各项中的任一者:位向量信息缺失(藉此允许系统甚至在不存在位向量的情况下起作用);使用成本函数;除了针对其计算了位向量的成本函数之外的路线选择(通常最快速路线,但未必如此);用户指定了与用于预先计算位向量的准则不同的准则(例如,用户希望避免高速公路);交通工具类型不同于用于预先计算位向量的交通工具类型(例如,用户步行且不乘汽车);及路线低于阈值长度(例如,路线的起点及终点在同一区内)。

[0674] 与图 20 相比,在图 21 上展示两个额外数字 1800、1801。另外,在所述图上以虚线道路段 1802 及 1804 来表示各区。“1”1800 指示道路段 BD 是从节点 B 到目标节点所位于的区的一个最小成本路径的部分。“0”1801 指示道路段 BC 不是从节点 B 到目标节点所位于的区的任何最小低成本路径的部分。

[0675] 因而,当从节点 A 开始时,本发明的实施例将使用上文所概述的 A*来从 A 起探索节点。然而,一旦已到达节点 B,就无需进一步使用所述探索,因为存在仅道路段 BD 为最小成本路径的可能部分的明确标记。因而,已使用一旦 A*方法且找到最小成本路径,就随后将通过查看位向量内的相关位来选择最小成本路径。

[0676] 然而,如果在不同时间在一对区之间存在不同的最小成本路径,那么最小成本数据可识别所述对区之间的一个以上最小成本路径。因此,开始节点与目标节点(起点与目

的地)之间的路线选择可能找到一个以上最小成本路径,且必须在竞争的最小成本路径之间做出选择,例如在道路段 BD 的数字 1800 及道路段 BC 的数字 1801 均具有指示每一道路段在特定时间为最小成本路径的部分的值“1”的情况下。在此实施例中,位向量不识别每一道路段为最小成本路径的部分的时间。因此,路线选择算法基于行进于所述路径的段上的时间来实行对两个可能的最小成本路径的成本分析。可根据从起点/开始节点的出发时间确定此时间。

[0677] 速度概况与每一段相关联(如图 10a 中所展示),所述速度概况涉及行进经由所述段的预期速度如何随着时间而变化。根据这些速度概况,可确定在相关时间的预期速度,且这可用以确定在所述时间沿着所述道路段行进的成本。通过合计沿着每一最小成本路径的道路段行进的成本,可确定每一最小成本路径的成本。导航装置可接着将在所述时间具有最低成本的最小成本路线选择作为路线。

[0678] 可在显示器 206 上显示所确定的路线。图 23 为如何可将路线显示于地图数据上的实例。

[0679] 在此实施例中,显示器包含在路线的可导航段中的至少一些可导航段上的预期速度的指示。对于此实施例,这些指示为对路线的着色及在地图的图像的侧面的时间条 2000 以展示沿着所述路线的预期速度。举例来说,如果预期速度超过第一阈值,那么可将可导航段着色为第一颜色(例如,绿色),且如果预期速度低于第二阈值,那么可将可导航段着色为第二颜色(例如,红色)。

[0680] 现在将参考图 10a 及 24 到图 31 而论述确定路线的替代方法。可确定表示随着时间而改变的成本的成本概况,而不是确定在特定时间路线的成本。在此实施例中,将每一可导航段的速度概况的多个值而不是单个值用以确定路线的成本概况。

[0681] 举例来说,参看图 21,每一路线的成本为成本如何随着时间而变化的分布,而不是具有每一道路段 BC、CE 等等的单个值。将这些成本分布求和以确定路线的成本概况。还在图 10a 中展示成本概况经由树的传播。

[0682] 在此实施例中,每一可导航段具有与其相关联的速度概况,其对于每小时包含 12 个时间组(time bin),每一时间组含有在所述时间行进经由可导航段的预期速度。确定每一组的预期速度的成本,且将路线的可导航段的每一相关组的成本求和以确定成本概况。每一段的所求和的值可能并非来自相同时间的组,因为沿着每一段行进的预期时间将视预测用户何时到达每一可导航段而变化。举例来说,可预测,用户将花 5 分钟沿着可导航段 BD 行进,因此,针对段 BD 及 DF 而求和到一起的组值可为相隔 5 分钟的组值,即将 BC 的时间组 T 的值与 DF 的时间组 T+5 的值相加。

[0683] 仅探索为最小成本路径的部分的可导航段,而不探索并非最小成本路径的部分的竞争可导航段。因此,对于待考虑的路径 BDF 及 BCEF 两者,两个路径必须被最小成本数据识别为最小成本路径。

[0684] 如果两个路径 BDF 及 BCEF 为在不同时间的最小成本路径,那么在两条路线相交的节点 F 处,BDF 及 BCEF 的成本概况需要在某种程度上被留存且可能被进一步传播。为了实现此,处理器实行与将两个成本概况(C1 及 C2)彼此叠加等效的计算,且确定上界分布 UB(以实线展示)及下界分布 LB(以点线展示)。这在图 30 及 31 中展示。UB 及 LB 分布接着被用于进一步传播,从而允许确定每一路线的最大及最小成本概况。

[0685] 以与后续可导航路线的成本值被添加到这些概况类似的方式发生上界分布及下界分布的进一步传播。如果所探索的路线分开,且接着又在类似于节点 F 的以后的节点处相交,那么通过将每一进入路径的两个上界概况重叠以确定单个新的上界概况且通过将每一进入路径的两个下界概况重叠以确定单个新的下界概况来再次处理所述两个上界概况及所述两个下界概况。以此方式,将概况的数目的增加保持在不会过度阻碍确定路线所花费的时间或不会需要大量存储器(因为在 PND 上存储器为有限的)的等级。

[0686] 在另一实施例中,仅留存上界概况及下界概况中的一者以供沿着路线进一步传播。在另一实施例中,确定两个进入概况的平均值以产生用于沿着路线进一步传播的平均概况。

[0687] 一旦已确定整条路线的成本概况,就将上概况或下概况中的一者选择为最后成本概况,其用以确定在导航装置上的显示,如现在将加以解释。

[0688] 图 24 及 25 说明可显示成本概况的多个值的方式。

[0689] 在图 24 中,通过显示从起点的出发时间及在目的地的到达时间以及隔开预定量(在此实施例中为 15 分钟)的离散出发时间的时间条来说明成本概况。与图 23 中的时间条 2000 相似,每一时间条经色彩译码以指示沿着路线的不同部分的预期速度。提供箭头 2001、2002,当用户选择所述箭头时,致使显示器显示出发时间及到达时间的对,以及较早或较迟开始时间的时间条(选择左箭头 2001 使得显示较早开始时间,及选择右箭头 2002 使得显示较迟开始时间)。选择时间条可使得显示针对选定开始时间的显示画面(类似图 23 中所展示的内容)。

[0690] 在图 25 中,将成本概况显示为在一组时间段上(在此实例中,在 168 个小时(等同于一周)中)的连续曲线图。将此曲线图连同地图一起显示,所述地图说明有助于形成此成本概况的从起点到目的地的路线。将曲线图及路线进行色彩译码(如由实线及点线所说明),使得用户可确定在什么时间将使用哪些路线。举例来说,将 7 小时 40 分钟与 9 小时 15 分钟之间的曲线图着色以展示在路线的开始,应遵循虚线路线 2003,而不是非点线路线 2004 的第一部分。此显示允许用户容易地看到在起点与目的地之间行进的成本如何随着时间而改变及这如何影响路线。

[0691] 图 26 到 28 说明替代实施例。在此实施例中,已确定成本概况,如上文所描述,但仅显示针对由用户指定的行进时间而确定的路线的图像。然而,显示器上为框 2010,其展示所显示的路线可适用的日及时间。可由用户在检视展示所确定的路线的地图数据的图像的同时选择设在日及时间的任一侧的箭头以改变日及/或时间。改变日及/或时间可引起路线改变,且显示器被更新以显示新路线以及在所述路线上行进所花费的时间及所行进的距离的任何改变。在图 27 及 28 中展示这些经更新的显示的实例。

[0692] “实时”地(例如,在毫秒的数量级)发生显示的更新,例如,小于 1000 毫秒且优选小于 500 毫秒。可经由使用预处理的最小成本数据,及/或因为针对这些其它时间的路线已经由概况搜索而确定来实现此些快速重新计算时间。

[0693] 图 29 说明使用概况搜索的导航装置的另一功能。在此实施例中,在选择了行进时间(例如,开始时间)时,导航装置的处理器的比较在所述时间行进的成本与在所述行进时间周围的窗(例如,行进时间 ± 30 分钟)中成本概况的值。如果处理器发现所述窗内的行进时间具有较低成本,那么处理器可致使显示器显示给出较佳结果的行进时间的指示。举

例来说,在所说明的实施例中,显示备注 2015,所述备注 2015 告知用户如果他 / 她 15 分钟后行进,那么行进时间将为较短的(在此情况下,短 10 分钟)。

[0694] 为了执行目前为止所描述的路径选择,使用解码过程从关于图 11 到 13 所描述的经编码的侧文件获得位向量信息。

[0695] 上文所描述的预处理的输出包含:

- [0696] • 对大多数节点的层级式区指派,及
- [0697] • 对这些节点处的外出道路段的位向量指派。

[0698] 地图加载

[0699] 一致性检查

[0700] 当加载地图数据时,侧文件的集合应呈现于地图目录中,否则,解码器将使搜索加速数据去活,使得在不具有搜索加速数据的情况下实行路线搜索。存在有助于确保数据完整性的若干检查,现在列出所述检查。

[0701] 侧文件遵循命名惯例。

[0702] 等级的数目由侧文件的数目给出,但等级的数目还存储在每一侧文件的标头中且等于侧文件的数目。

[0703] 每一侧文件存储在其相应等级处的区的数目,其与在侧文件名称中指定的数目相同。此外,侧文件存储以下信息(其对于所有侧文件应相同):

- [0704] • 侧文件版本。
- [0705] • 每“位向量页”的节点的数目(下文解释)。
- [0706] 每一侧文件中还存在校验和,其识别
- [0707] • 作为整体的特定侧文件集合
- [0708] • 作为整体的地图。

[0709] 对于给定电子地图的相关联的侧文件,这些信息应为正确的。如果上述检查中的任一者失败,那么将不针对此地图激活搜索加速数据特征。

[0710] 加载到存储器中的数据结构

[0711] 解码器读取外部存储器实施方案中的侧文件。这意味着位向量侧文件的内容不被完全加载到存储器中,而是仅根据需要来读取。然而,在开始时将一些一般性数据加载到存储器中,且只要所述地图被加载,所述数据就被保持于其中。

[0712] 标头信息

[0713] 每一侧文件以标头区段开始,所述标头区段含有上文关于图 8 到 10 所描述的数据。每一侧文件的此信息存储在存储器中。

[0714] 霍夫曼树定义

[0715] 侧文件含有若干霍夫曼树的定义。每一霍夫曼树定义给出对特定霍夫曼编码的完整描述,且稍后被用以将侧文件位流的一部分解码成原始数据(即,将来自侧文件的位序列解码成数目或某其它特定值)。从每一侧文件读取以下霍夫曼树定义且将其保持于存储器中。

[0716] • 用于解码每一节点处的道路段的数目的霍夫曼树。(经编码的道路段数目可比基础地图中的道路段的数目小。注意,解码器独立于基础地图,即,其不需要读取地图)。此霍夫曼树仅存储在等级 0 的侧文件中。

[0717] • 用于解码页码的霍夫曼树（下文解释）。

[0718] • 用于解码位向量块的若干霍夫曼树。树定义的数目是由如存储在侧文件标头中的规则位向量块长度给出；其等于块长度减去一。（道路段的位向量的整个位串被分裂成块。如果位向量位串的长度不是规则块长度的倍数，那么最终块较短。对于从 2 直到规则块长度的每一块长度，存在一个霍夫曼树。霍夫曼树不被用于长度为一的块，因为这仅为一位且被直接存储）。

[0719] • 用于为位向量选择解码方法的若干霍夫曼树。在标头中指定这些树的数目；在下文解释其使用。

[0720] 邻居列表

[0721] 在除了最精细等级外的每一等级处，侧文件编码邻居列表。等级 k 处的区的邻居列表为零或零个以上被称为等级 k 区的邻居的等级 $(k+1)$ 区的列表。等级 k 的侧文件处的邻居列表区段具有两个部分。

[0722] • 第一部分含有每一等级 k 子区的邻居区的数目。举例来说 ($k = 1$)，如果在全局等级 0 处存在 4000 个区，且每一全局区被再分成 10 个等级 1 区，那么等级 1 侧文件含有 4000×10 个等级 1 子区。对于这些子区中的每一者，给出个别邻居列表（其由等级 2 区组成）的长度。

[0723] • 第二部分含有实际邻居列表，其相应长度从第一部分获知。每一邻居列表为等级 $(k+1)$ 子区的列表。

[0724] 区重新映射表

[0725] 侧文件（例如）通过聚结及重新排序等等而以压缩格式存储位向量。等级 k 处的侧文件具有用以解压缩按块编码的位向量的区重新映射表。其由两部分组成：

[0726] • 第一部分为按等级 k 子区编索引的阵列。对于等级 k 处的每一子区，其含有经压缩的位向量的长度。这对于按块编码位向量的相应子区中的节点的所述外出道路段为相关的。

[0727] • 第二部分为按 (1) 未压缩的位向量中的位位置及 (2) 等级 k 子区编索引的二维阵列。所述阵列条目指定了按块读取的未压缩位串中的给定位位置在经压缩位向量中的位位置。

[0728] 注意，仅第一部分被保持于存储器中；第二部分仅在解码期间被按需使用，因为其较大。当前，仅在全局等级侧文件 ($k = 0$) 处存储区重新映射表。

[0729] 起始路线查询

[0730] 解码器是用以加速从出发位置到目的地节点集合的路线搜索。（目的地由一个以上节点组成的实例是将整个道路路段用作目的地的情况）。目的地节点界定一个或一个以上目标区的集合。

[0731] 注意，目标区为区 ID 的序列，每一分割等级有一区 ID（开始于等级 0 处）。在每一新搜索的开始，通过清除先前目标区集合且将新的目的地节点传递到解码器而建立目标区的集合。解码器将确定不重复的目标区的列表。用于找到给定节点的区的机制与在搜索期间相同；细节见下文。

[0732] 在路线搜索期间扫描节点

[0733] 剩余论述假定已设定目标区的集合。解码器的特征为当被提供道路网络的节点 ID

时传回按离开此节点的道路段编索引的位阵列（此节点及给定目标区的位向量）的功能。节点及节点处的道路段的排序由地图界定。每当位值为零时，这意味着在搜索期间可忽略对应道路段。（这通常导致搜索空间的大量减少，且因此搜索的运行时间的大量减少）。

[0734] 对于少数节点，区信息或位向量数据均不存储在侧文件中。对于这些节点，解码器传回所有位皆为 1 的位串。（这将防止路线搜索跳过此节点处的任何道路段）。解码器具有指示是否此为针对给定节点的情况的布尔 (Boolean) 查询功能。此外，存在指示给定节点是否位于先前所定位的目标区中的一者中的布尔查询功能。由解码器传回的在目标区中的节点的位向量再一次为所有位为 1 的位串。这些布尔查询功能是用于路线搜索中的优化。

[0735] 根据侧文件格式，按若干步骤解码给定节点的位向量数据。在每一侧文件中，将区及位向量信息组织成所谓的页。每页的节点的数目为 2 的固定幂（例如，16），且对于侧文件集合中的每一侧文件为相同的。这意味着，对于给定节点 ID，可通过简单的位移位来计算位向量页的页索引。节点的信息被按节点 ID 连续存储。

[0736] 找出给定节点的页偏移

[0737] 对于每一侧文件，每页的字节数的平均数目被存储在侧文件的标头中。所述平均数目被用以通过使页索引乘以平均大小而近似得出页的字节偏移。将校正项存储在按页索引来编索引的表中。此表被存储在侧文件的单独区段中。当查询页时，在表中查找校正项，且将所述校正项与近似页偏移相加，从而给出所述页在侧文件中的位置。

[0738] 解码页

[0739] 高速缓存

[0740] 当第一次查询页时，解码（相对于固定目标区集合）并高速缓存所述页中的节点的位向量串。下一次从同一页查询节点的数据时，可使用经高速缓存的数据而不进行任何侧文件存取。经解码的位向量位串中的特殊标记位是用以记住所述节点是否为无信息节点。

[0741] 页码

[0742] 对于每一页，所谓的页码位指定所述页中的所有节点是否具有相同区 ID。页码对于每等级含有一个位，但所有位仅在等级 0 侧文件中的页的开始处被存储为共同霍夫曼符号。

[0743] 解码外出道路段计数

[0744] 如上文所提到，每一页含有固定数目个节点的信息。节点的此数目存储在每一侧文件的标头中。在一页的开始（或针对等级 0，在页码之后），所述页列出所述页中的每一节点的外出道路段的数目。每当外出道路段的数目为零时，这意味着根本未存储对应节点的信息。在解码所述页时，数目被存储在临时阵列中。

[0745] 解码区

[0746] 在道路段计数区段之后是区区段。节点的区由区 ID 的序列给出，每一等级处有一个区 ID。特定等级的区 ID 存储在对应侧文件中。解码器读取页中的所有节点在所有等级处的区。当未存储节点的信息（即，节点的道路段计数为零）时，使区信息为空的。解码器读取道路段计数大于零的第一个节点的区 ID 序列。如果给定等级处的页码指定所述级处的所有区 ID 为相同的，那么在所述级处为所有节点设定相同区 ID。否则，读取具有正道路段计数的所有节点的在对应等级处的区 ID。在此过程的结束时，解码器已用页的所有节点

的完整区 ID 序列填充表（其中一些序列可为空的）。

[0747] 解码位向量

[0748] 找出相关位位置的集合

[0749] 对于给定节点及目标区来说，特定等级处的特定位置确定离开此节点的道路段的位向量位的值。当存在一个以上目标区时，将所得位一起进行“或”运算。对于每一节点，解码器计算相关位位置的集合。对于所述节点处的每一外出道路段，相关位位置的集合相同。这仅视节点的区及目标区的集合而定。如果仅存在一个目标区，那么在一等级处将仅存在一个相关位位置；换句话说，对于此节点，可忽略其它等级处所存储的信息。在一个以上目标区的情况下，一些相关位位置可重合，所以总是至多存在与存在的目标区一样多的相关位位置。在下文中，将论述解码器如何确定一个目标区的相关位位置。对于一个以上目标区来说，以相同方式找到相关位位置且将相关位位置组合成一集合。

[0750] 当未界定邻居区时，对于每一等级处的每一目标区存在一个位向量位（每外出道路段）。（为简单起见，忽略对于节点自身的区 ID 不存储位的事实）。相关位处于第一等级（从等级 0 计数），在所述级处，节点的区 ID 不同于目标区的区 ID。举例来说，如果全局等级处的节点的区 ID 等于全局处的目标区 ID，但所述两个区 ID 在等级 1 处为不同的，那么相关位位置处于等级 1，且其等于目标区 ID。所述相关位位置为所述级处的未压缩的位向量串中的位位置；此串针对每一可能的目标区 ID 含有一个位。区重新映射表是用以将此位置变换成经压缩位向量串（在所述级实际上被编码的串）中的位置。

[0751] 当界定了邻居区时，那么相关位被确定在“最精细”等级（在其中目标区为所述节点的区的邻居）处。将四个等级作为实例，令目标区 ID 序列为 (a, b, c, d)，且令节点的区 ID 序列为 (e, f, g, h)。如果 (a, b, c, d) 为 (e, f, g) 的邻居（如在邻居列表区段中所定义），且 (a, b) 为 (e) 的邻居，那么相关位由 (a, b, c, d) 确定且位于等级 2 处，即含有作为邻居的 (a, b, c, d) 的 (e, f, g) 前缀的等级。更精确地说，相关位位置为等级 2 侧文件中作为区 (e, f, g) 的邻居的 (a, b, c, d) 的索引。在此情况下，区 ID 的规则位 h（如先前段中所解释）为不相关的。此外，此相关位位置是相对于未压缩的位向量串。解码器使用区重新映射表以将其变换到等级 2 处的经压缩位串中的位位置。关于邻居列表的更多信息包含于“增强的多等级图分割的提案 (Proposal for enhanced multi-level graph partitioning)”的文献中。

[0752] 解码位向量位

[0753] 在给定目标区的固定集合的情况下，页中的每一节点的位向量将由每外出道路段一个位组成。如果节点的道路段计数为零（即，节点为无信息节点），那么将把每一位设定成 1，而对于所述节点不进行进一步解码。如果节点位于目标区中的一者中，那么位向量将又为全部 1；在此情况下，可能必须跳过经编码的数据以便解码后续节点的数据。

[0754] 如果当前节点的位向量并非平凡地均为 1，那么解码器确定相关位位置的集合，如较早所解释。每一位位置是相对于特定等级，即，位位于特定侧文件中。在每一等级处，侧文件含有完全经压缩位串，但解码器仅需要提取相关位位置处的位。在解码期间，读取（跳过）不使用的信息（但仅在其后为实际上需要的信息的情况下）。否则，根本不读取位流的不使用的部分。

[0755] 根据等级对相关位位置的集合分组。将相关位置变换成经压缩位串中的位置，而

不是解码经压缩位串且将其解压缩以读取相关位置处的位。如果在某等级处存在相关位，那么首先在适当时（解码器记得其已在每一等级处的侧文件中处理到多远）跳过前置节点的数据。当在给定侧文件中到达正被讨论的节点的区段时，逐行解码位向量。对于每一外出道路段来说，存储一个位；其为所有相关位位置的经解码位的“或”运算的结果。当前节点处的特定道路段的信息以指定解码方法的霍夫曼符号开始。其具有以下值中的一者：

[0756] •指定此等级处的道路段的所有位向量位为 0（包括此等级处的所有邻居位）的符号。对于此道路段，不必解码其它位。

[0757] •指定此等级处的道路段的所有位向量位为 1（包括此等级处的所有邻居位）的符号。对于此道路段，不必解码其它位。

[0758] •指定位向量位串被按块明确给出的符号。在下文解释按块对经压缩位向量位串的解码。将道路段的位向量位置于在整个页的编码期间建立的“历史堆叠”中。

[0759] •指定对历史堆叠的索引的符号。在给定索引处，历史堆叠含有道路段的所要位向量位值。

[0760] 视历史堆叠的当前大小，使用用于解码方法选择器的不同霍夫曼树。在侧文件的标头中存在指定霍夫曼树的数目的限制值；当历史堆叠的大小超过此值时，就重新使用上次的霍夫曼树。

[0761] 如果方法选择器符号指定位向量串应被明确地编码，那么解码器逐块地读取经压缩位向量位串。解码器收集此等级处的相关位位置（相对于经压缩位向量串）处的位值。对所述位的值进行“或”运算。所述“或”的中间结果一变成 1，就可跳过此道路段的所有其它块中的剩余位。用于解码每一块的霍夫曼树视所述块中的位的数目而定。针对如指定于侧文件标头中的规则块长度存在一个霍夫曼树。道路段的最后块可为较短的；视其大小，将适当霍夫曼树用于解码。长度为一的块为仅一位，将其直接读取而无需霍夫曼解码。

[0762] 如果块的数目为至少 2，那么从位向量产生的位流含有在所述块的霍夫曼符号之前的一额外位。所述位的值指定是否将整个经解码的串理解为实际值的逐位否定。（此否定的目的是更好的霍夫曼压缩）。

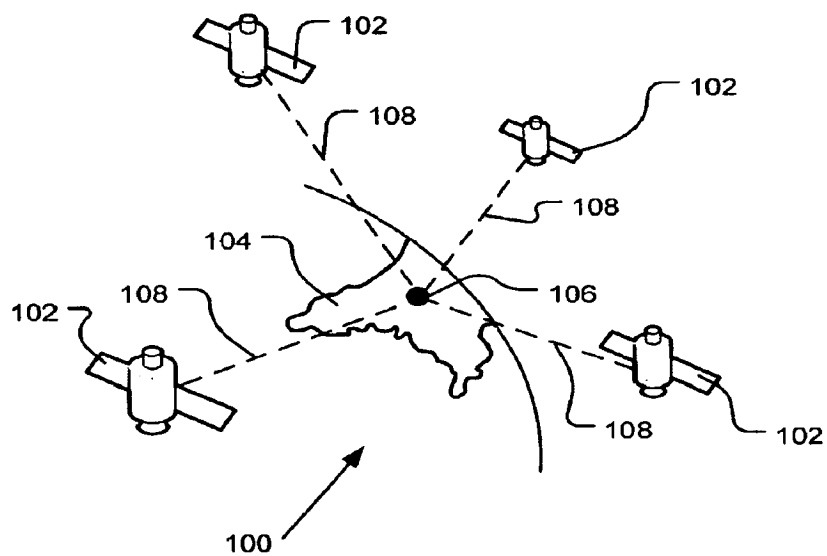


图 1

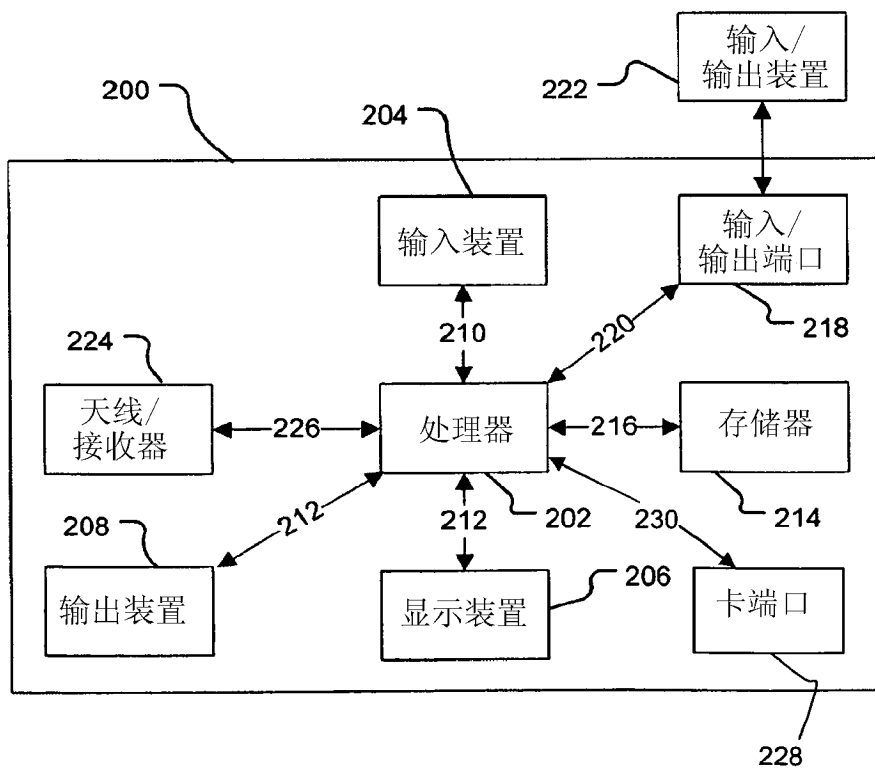


图 3a

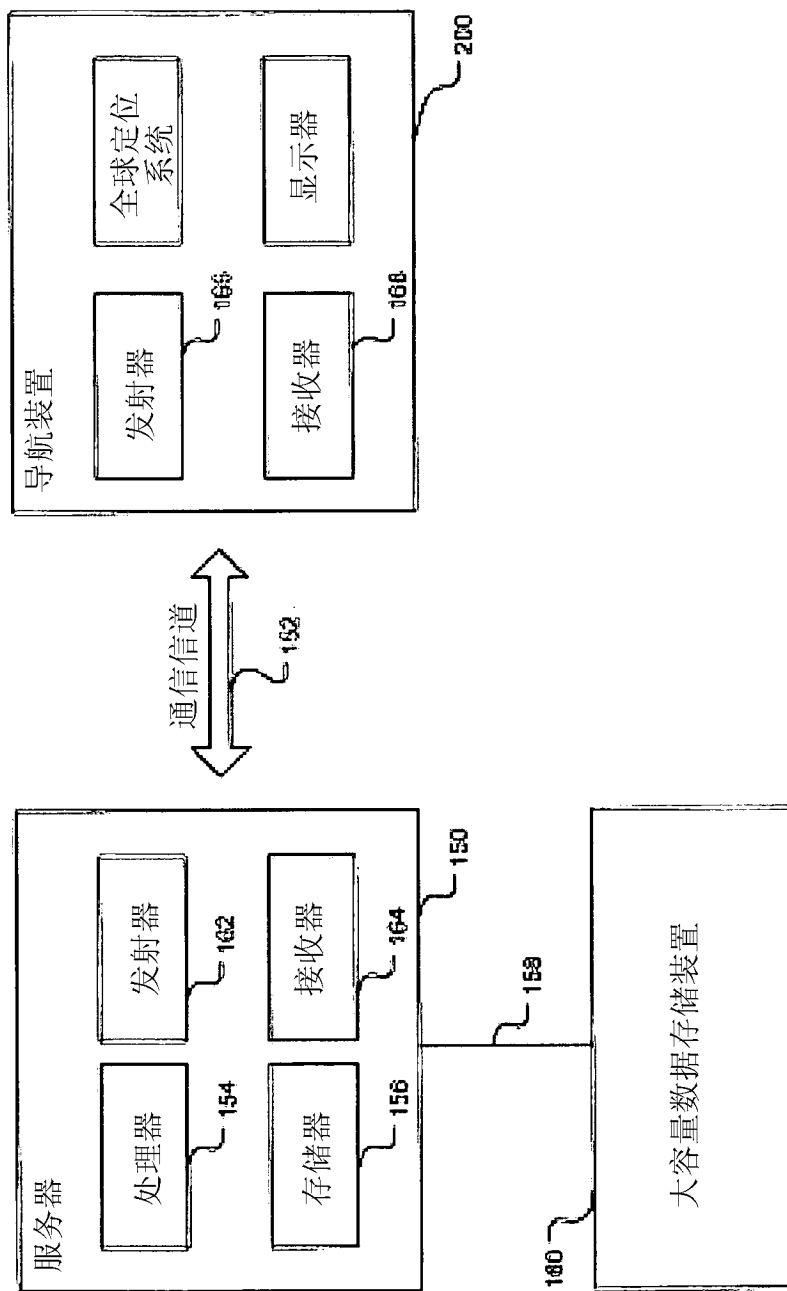


图 2

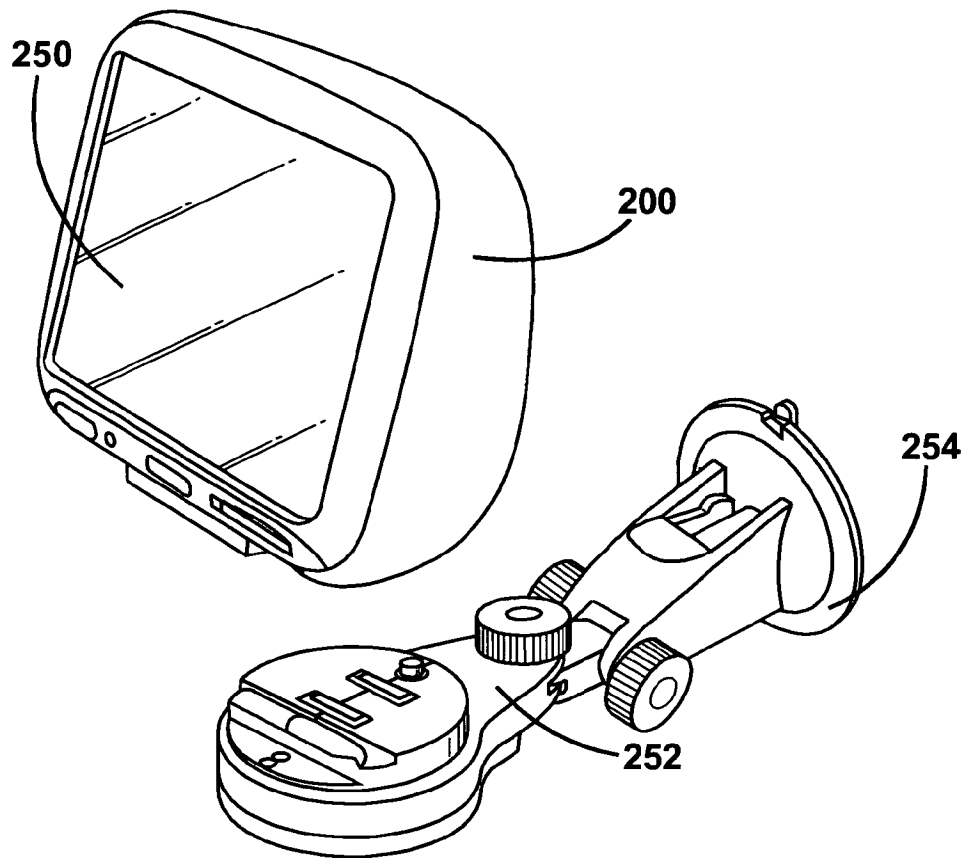


图 4

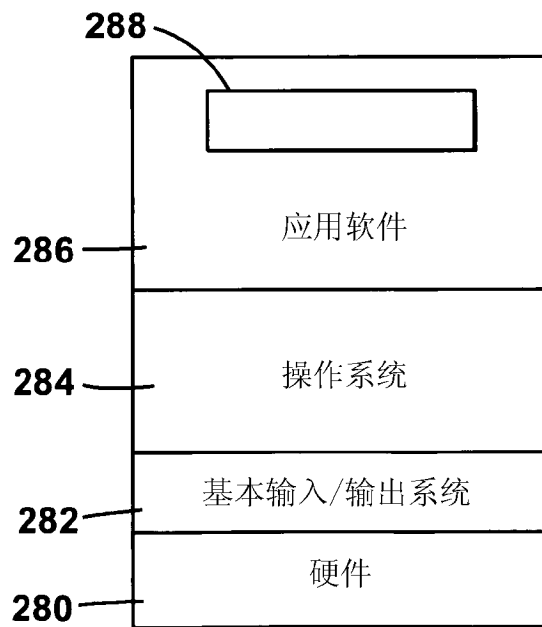


图 3b

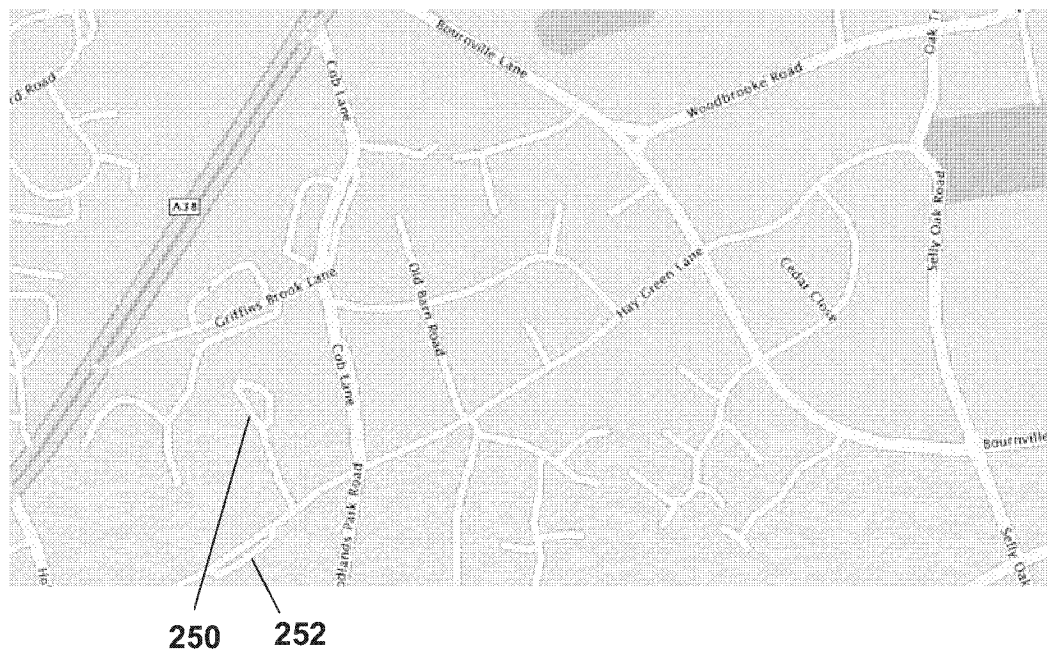


图 5

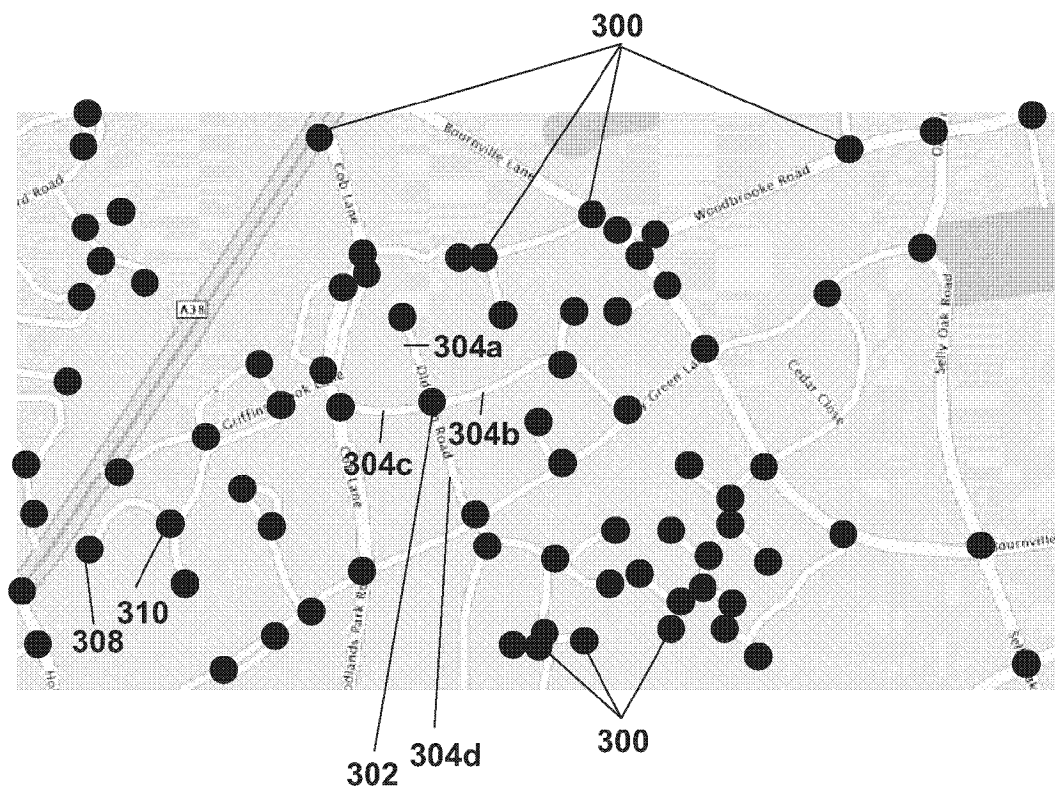


图 6

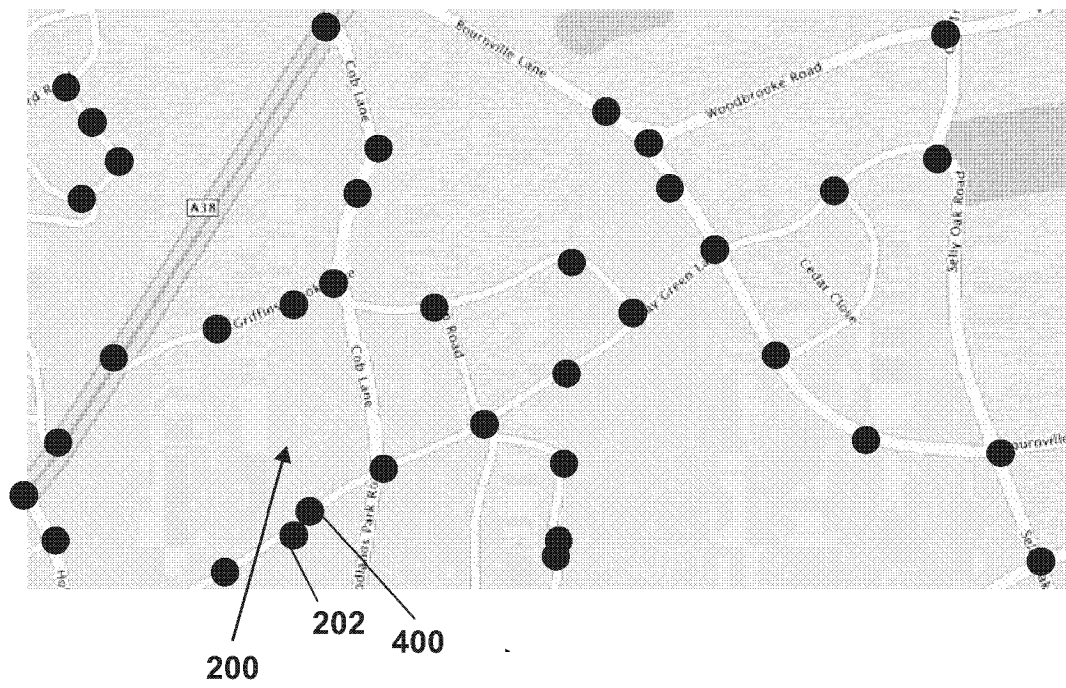


图 7

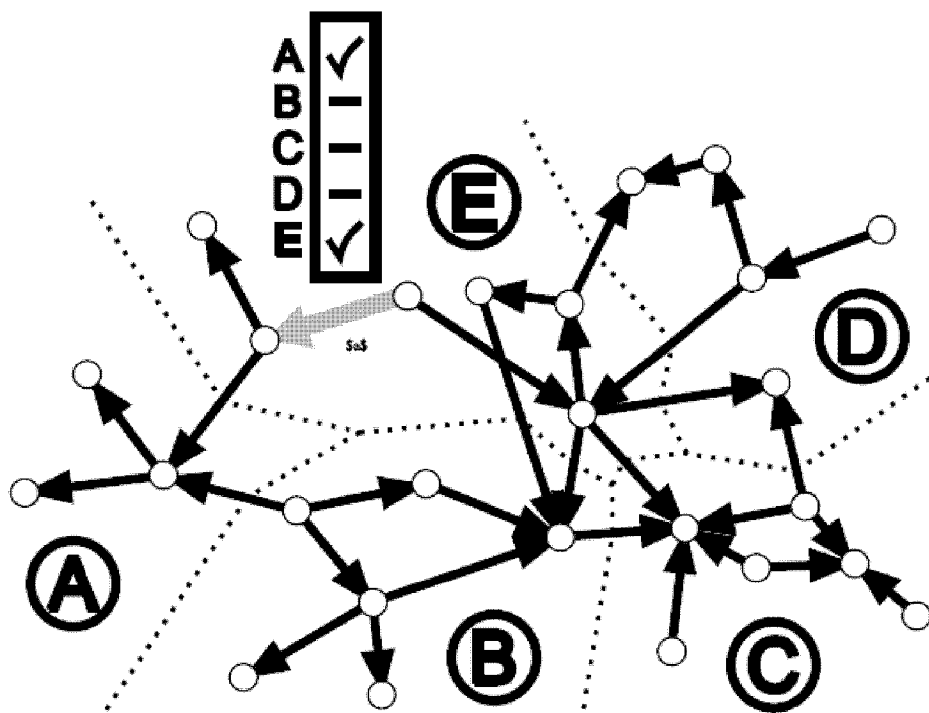


图 8

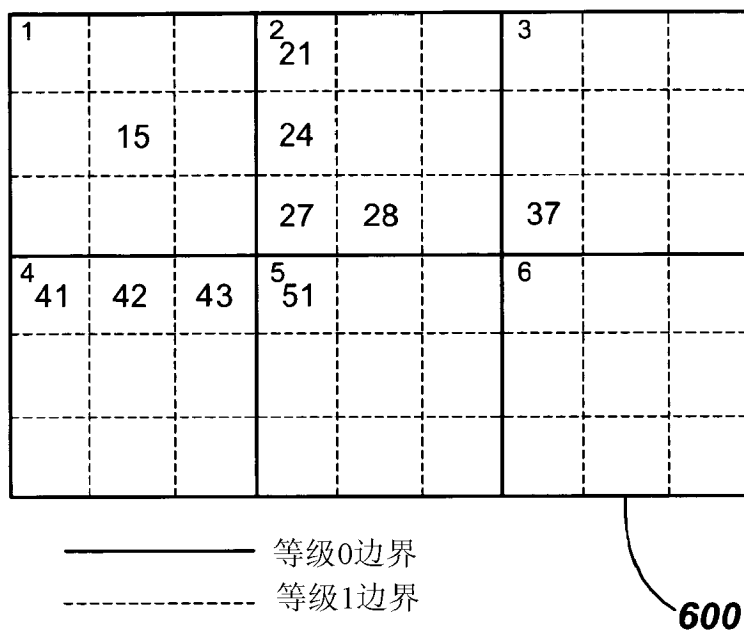


图 9

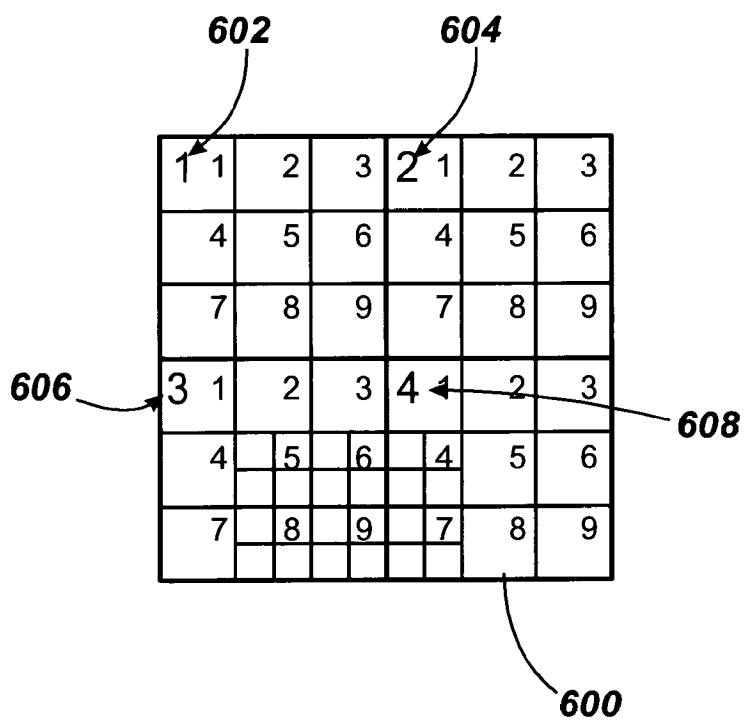


图 9a

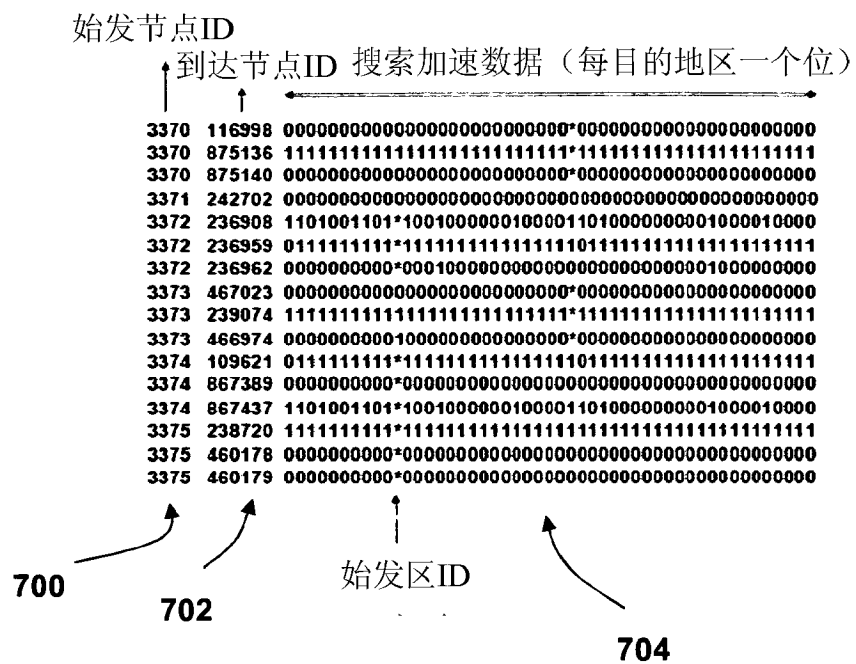


图 10

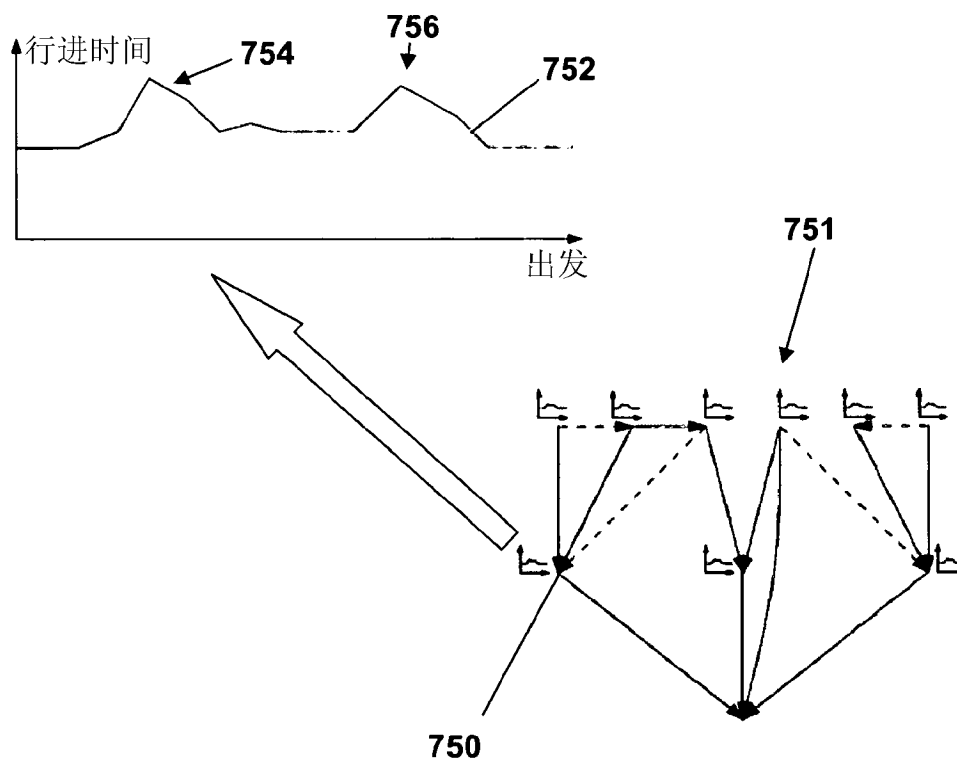


图 10a

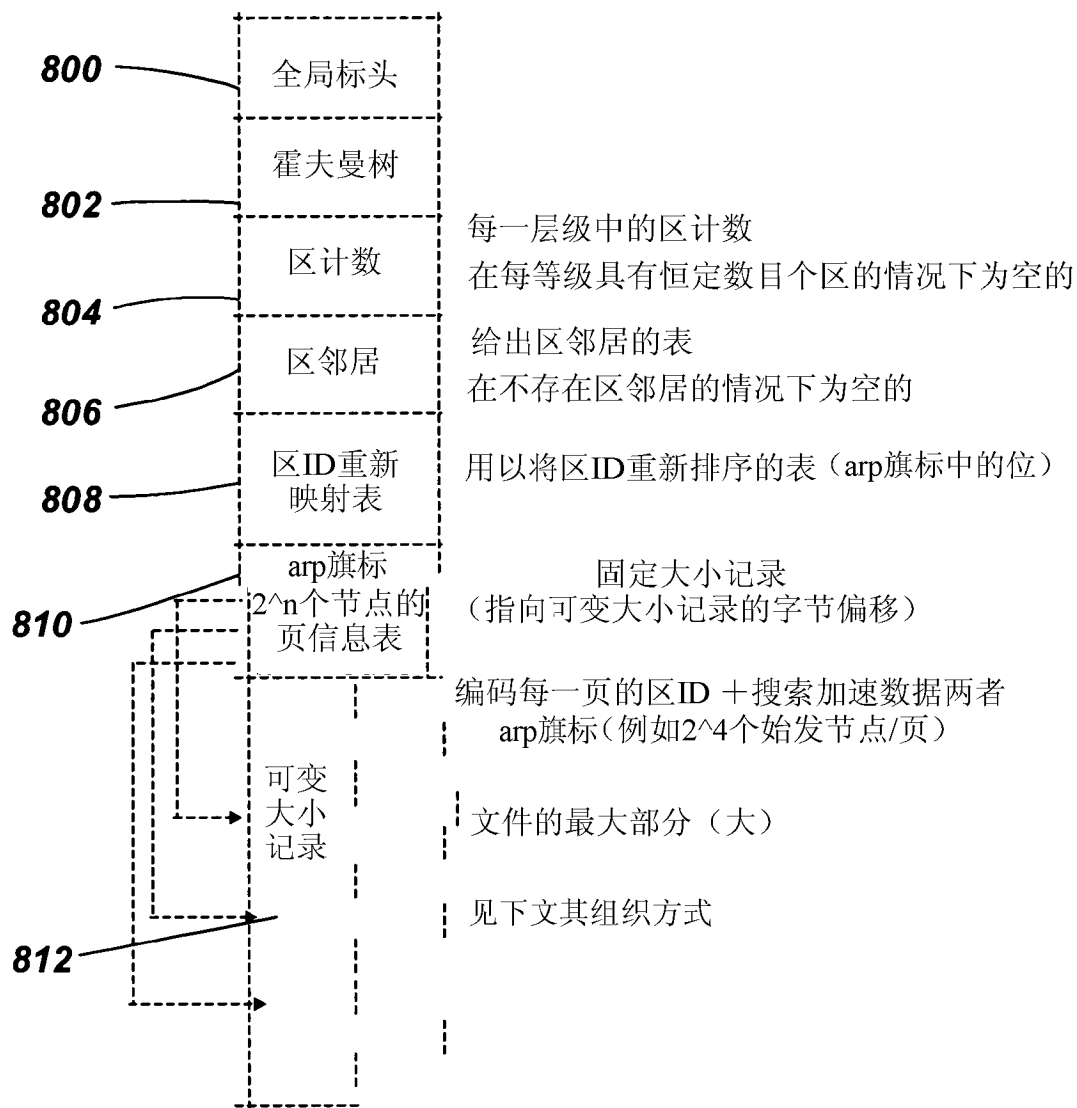


图 11

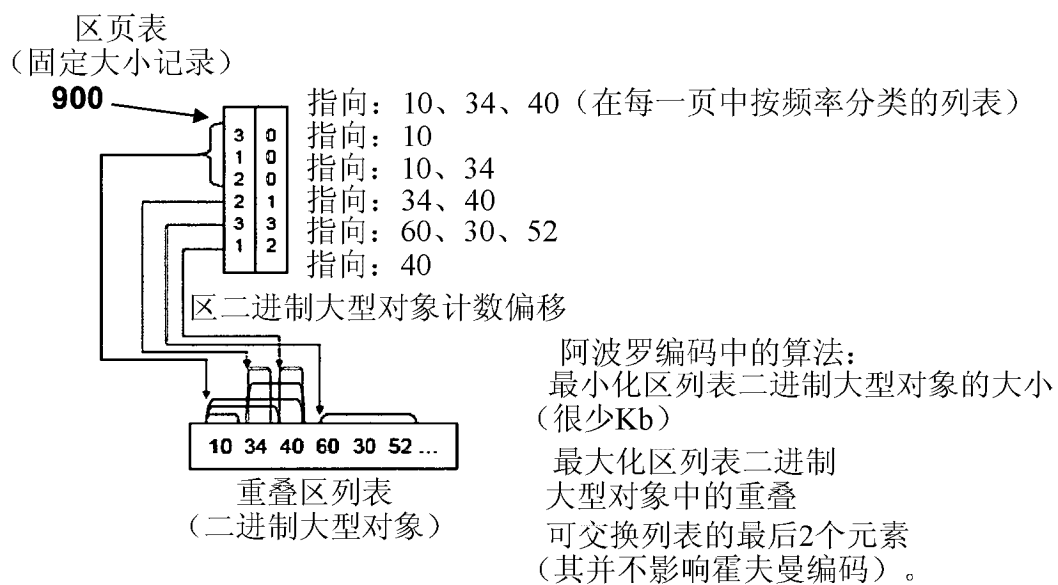


图 12

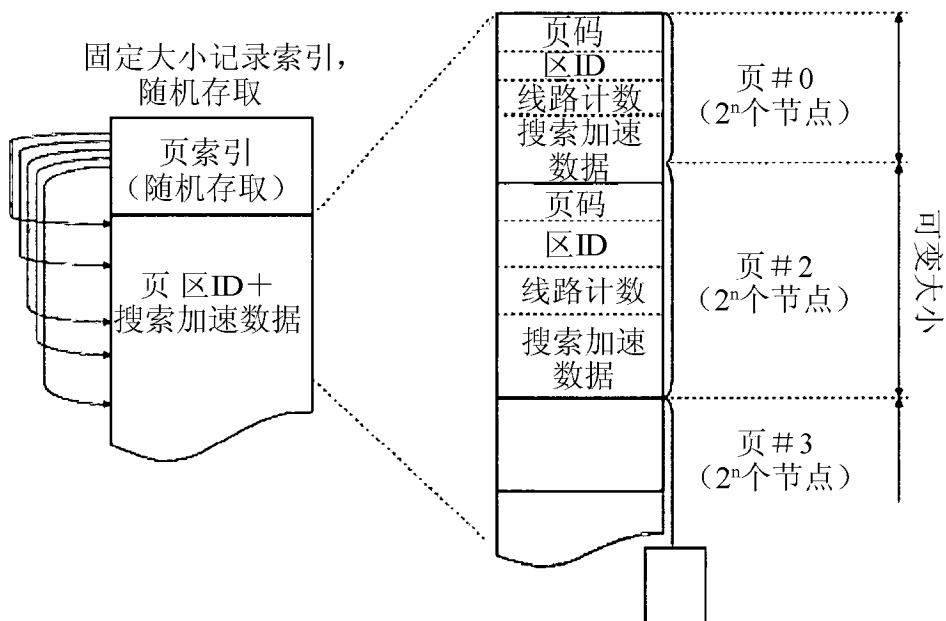


图 13

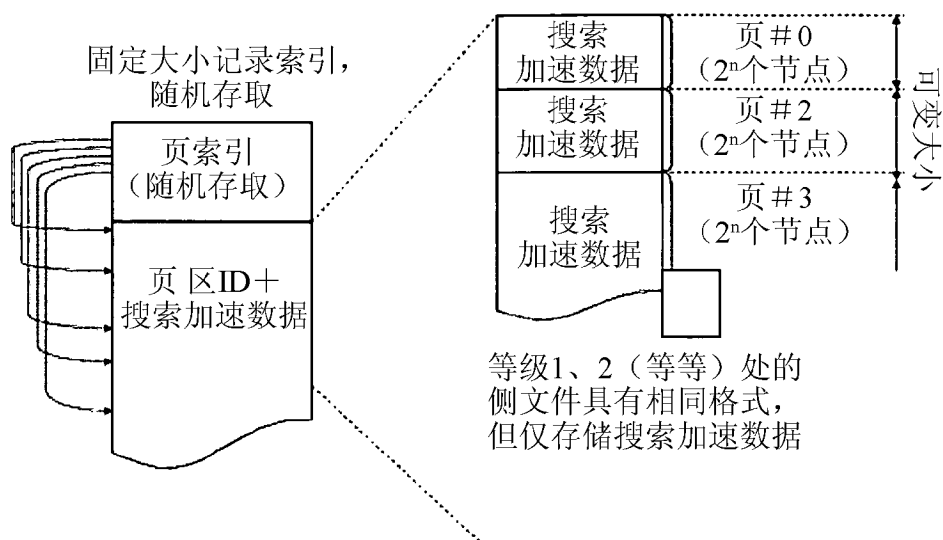


图 14

在聚结之后的可变大小数据:
霍夫曼块大小

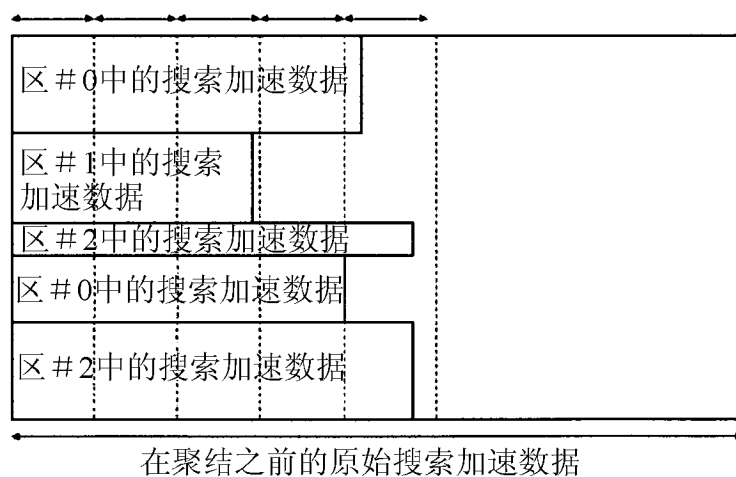


图 15

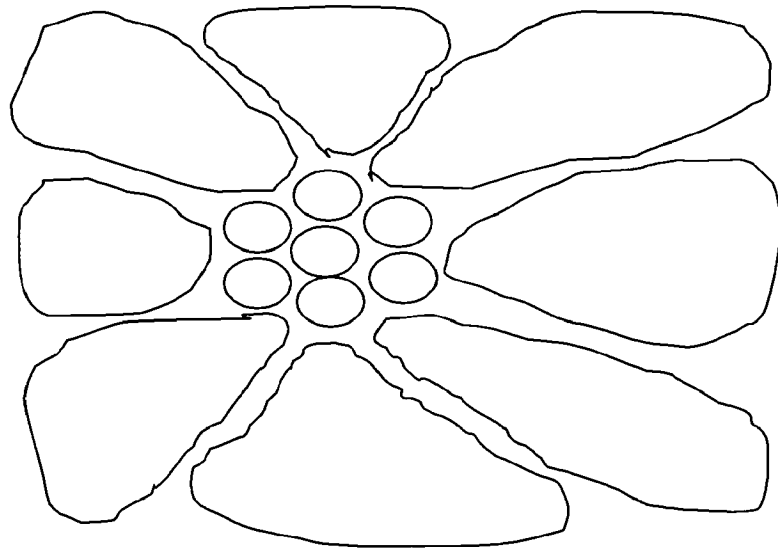
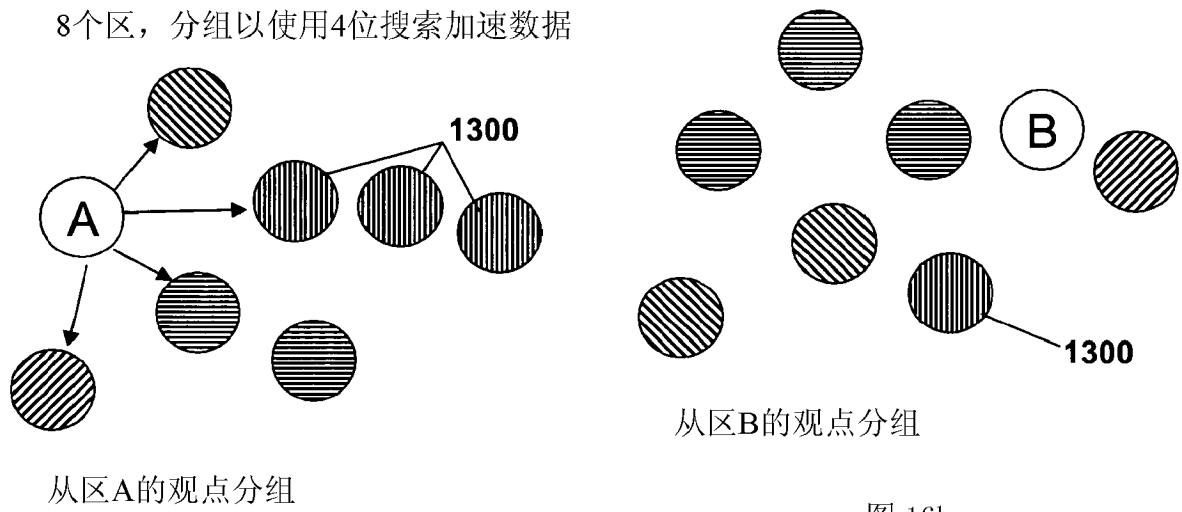


图 17

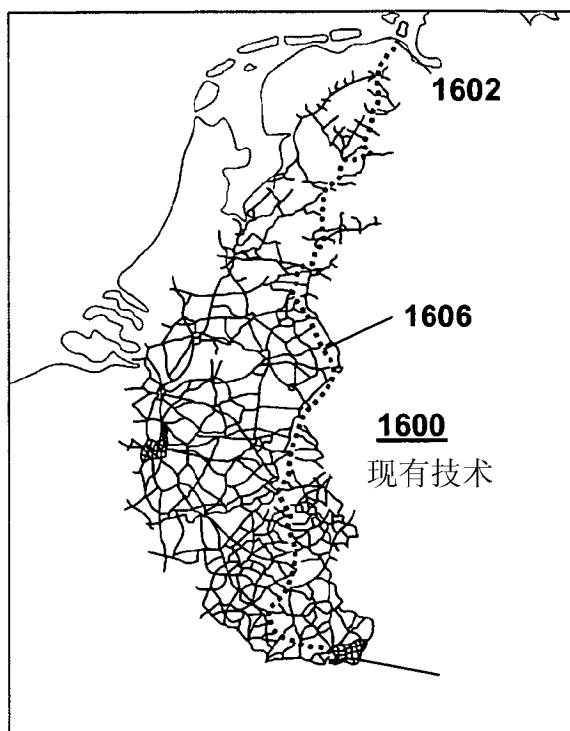


图 18

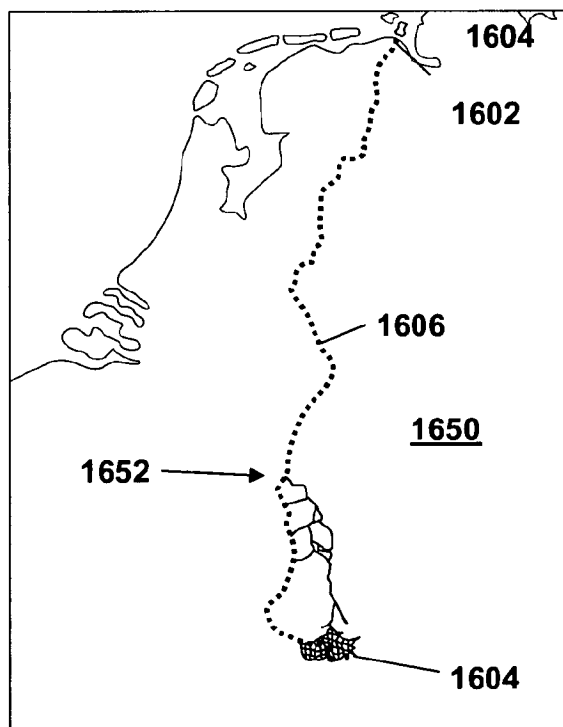
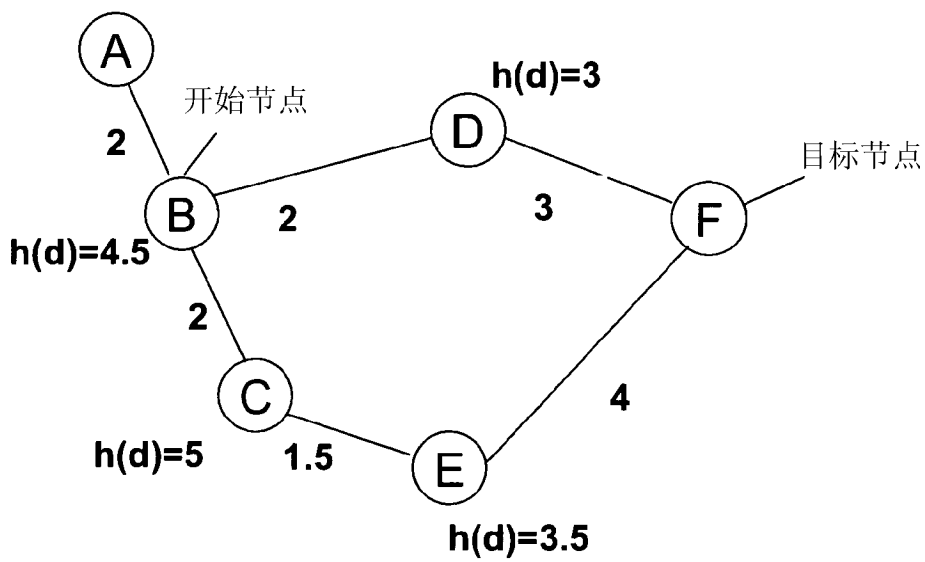


图 19



现有技术图 20

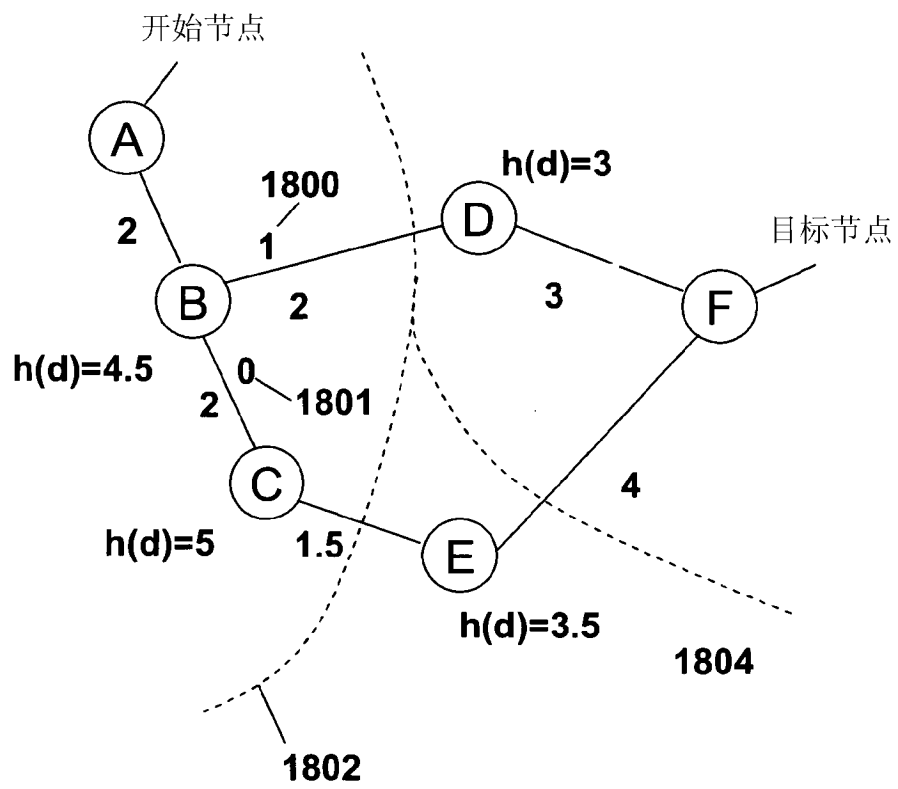


图 21

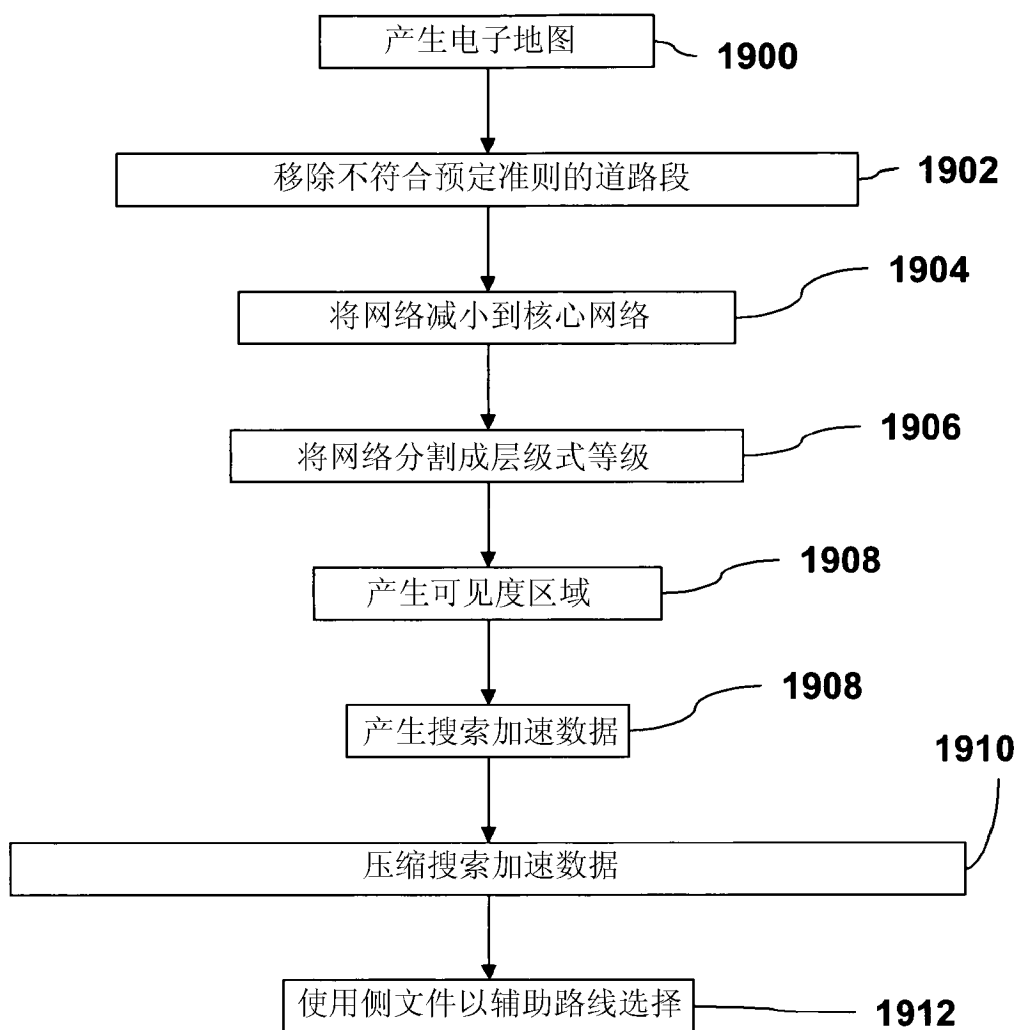


图 22

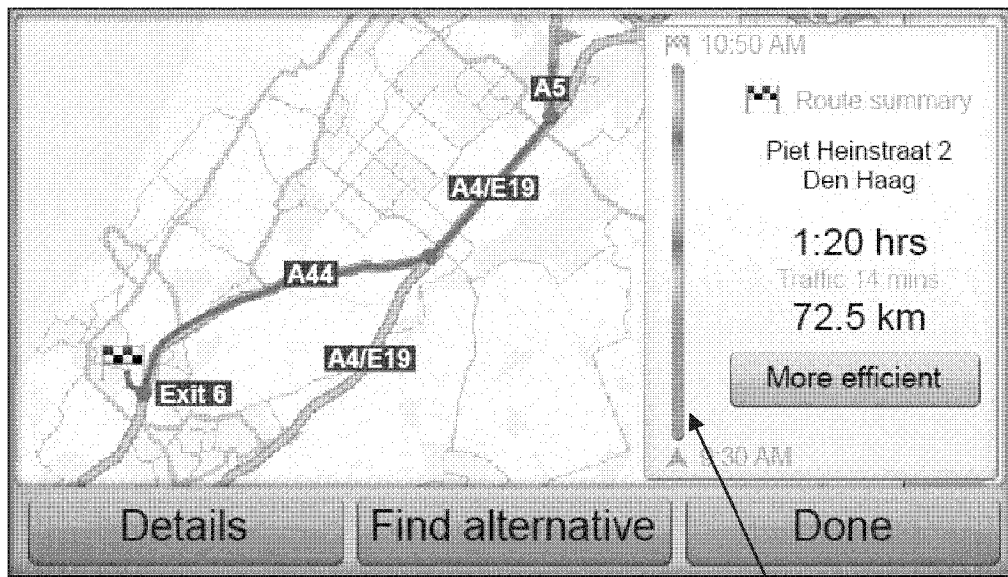


图 23

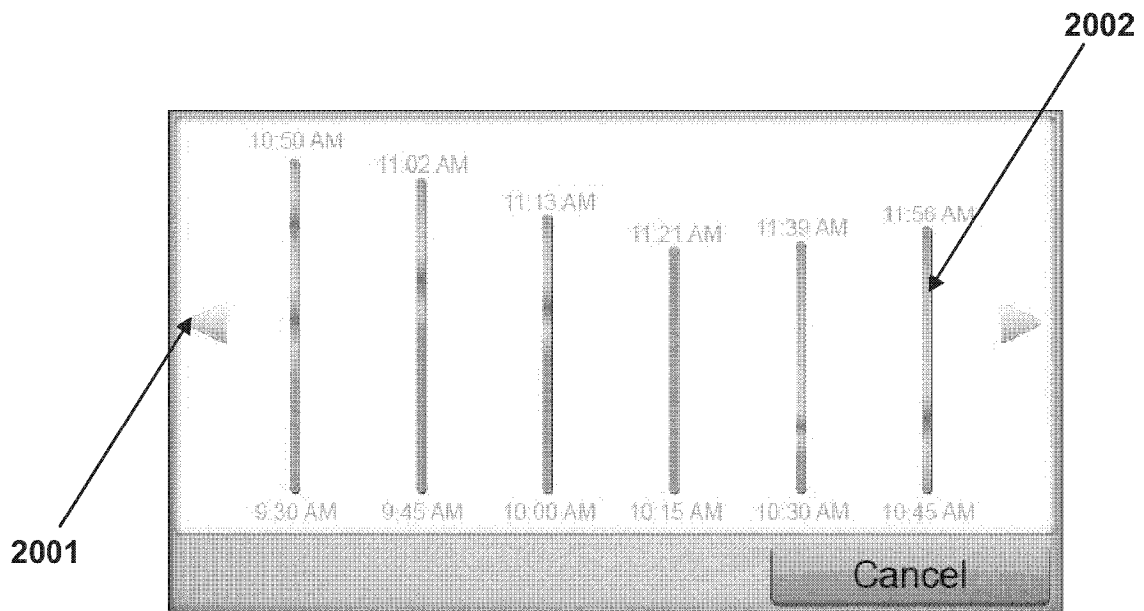


图 24

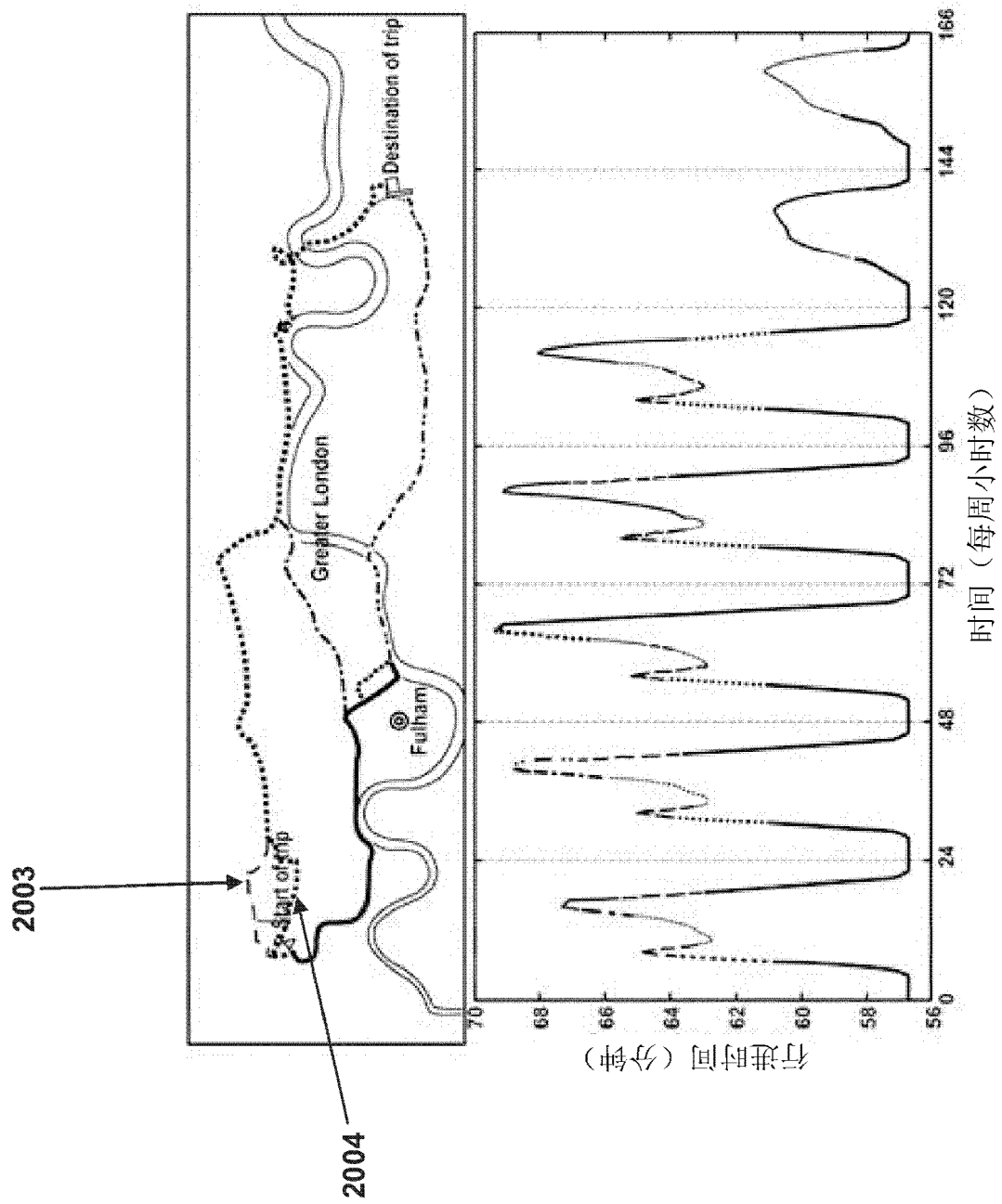


图 25

时间机器=天及时间选择器

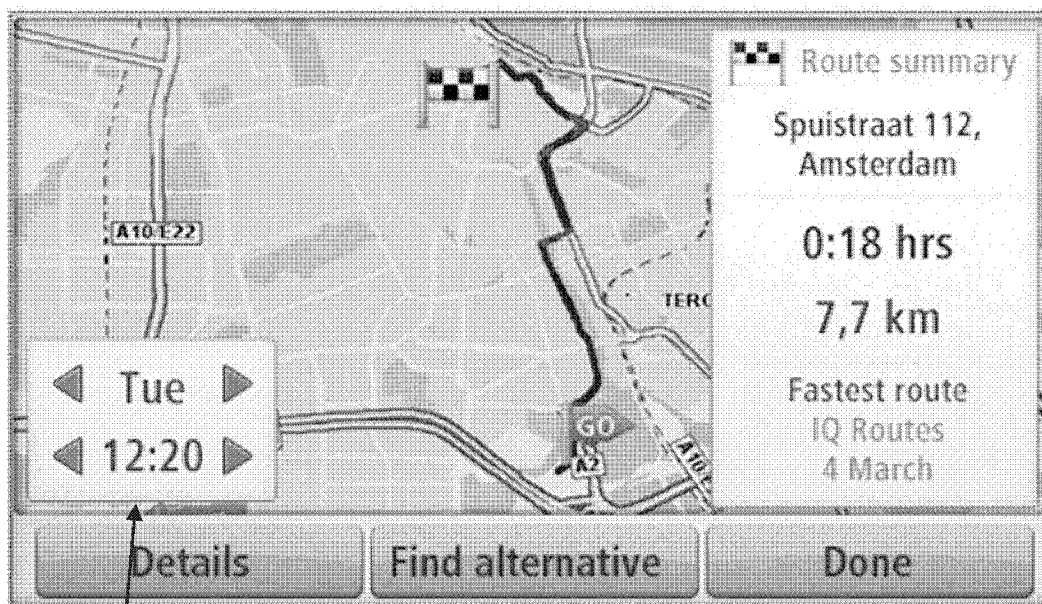


图 26

时间机器=天及时间选择器

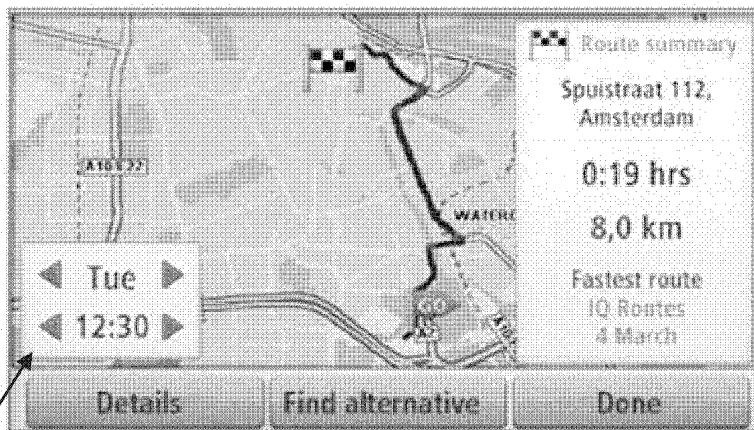


图27

2010

时间机器=天及时间选择器

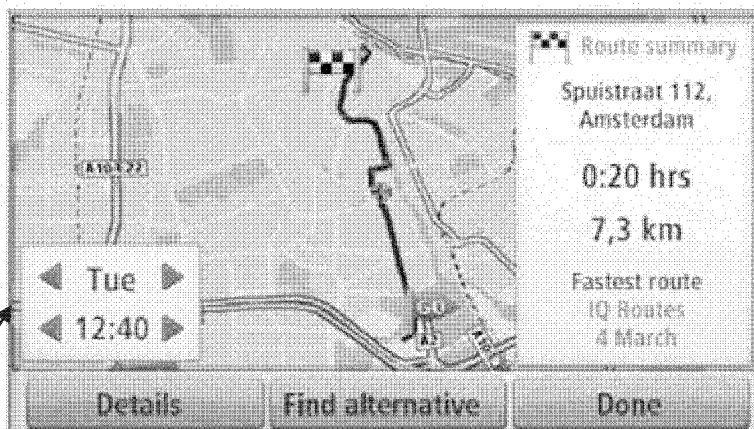


图28

2010

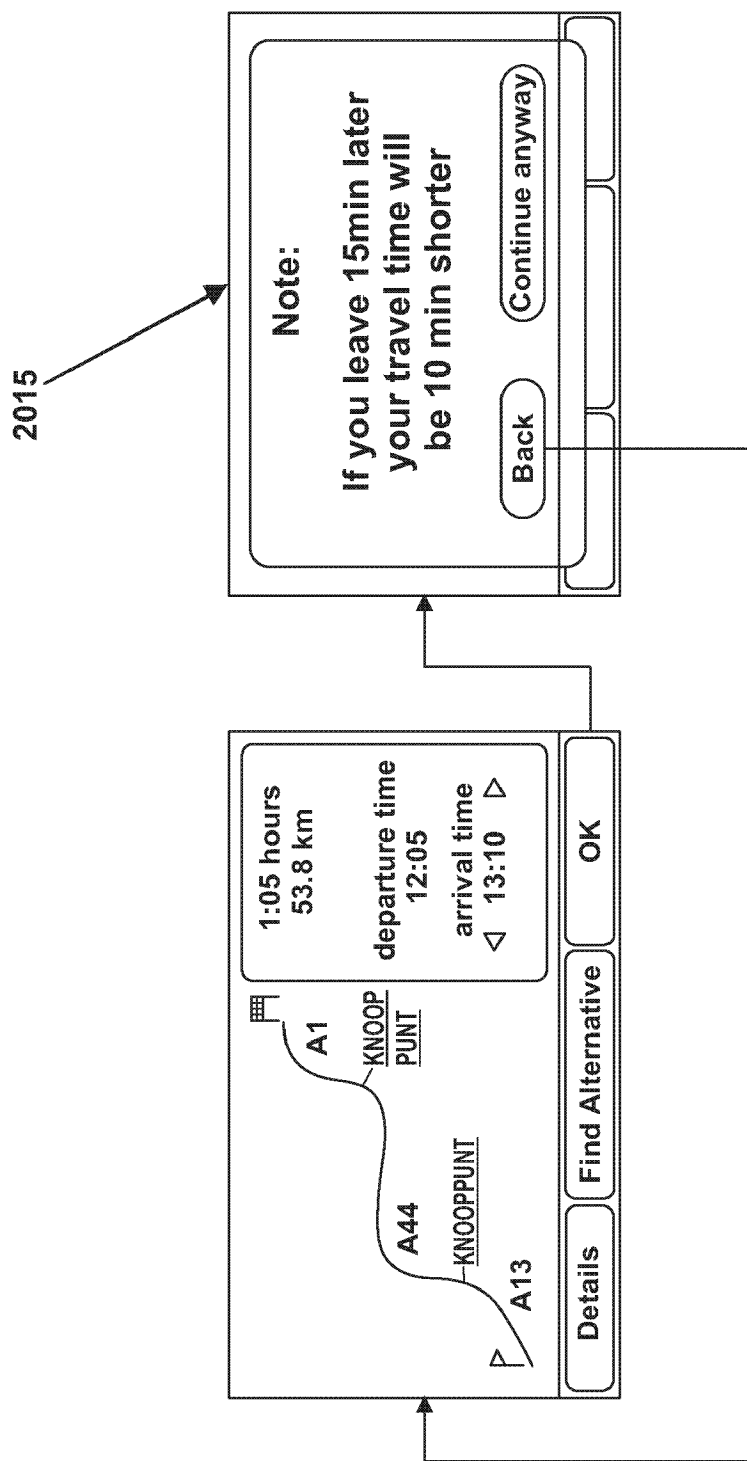


图 29

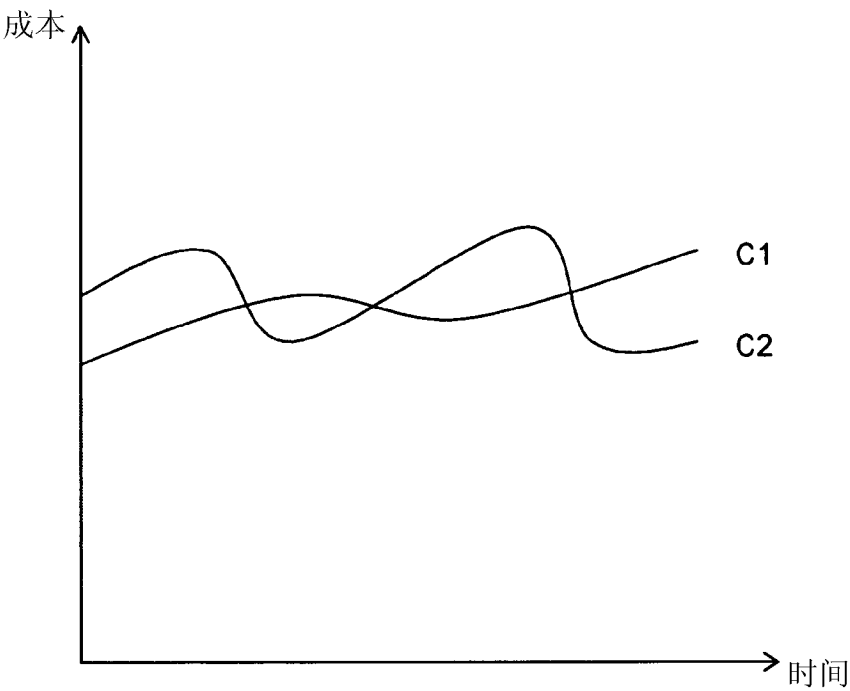


图 30

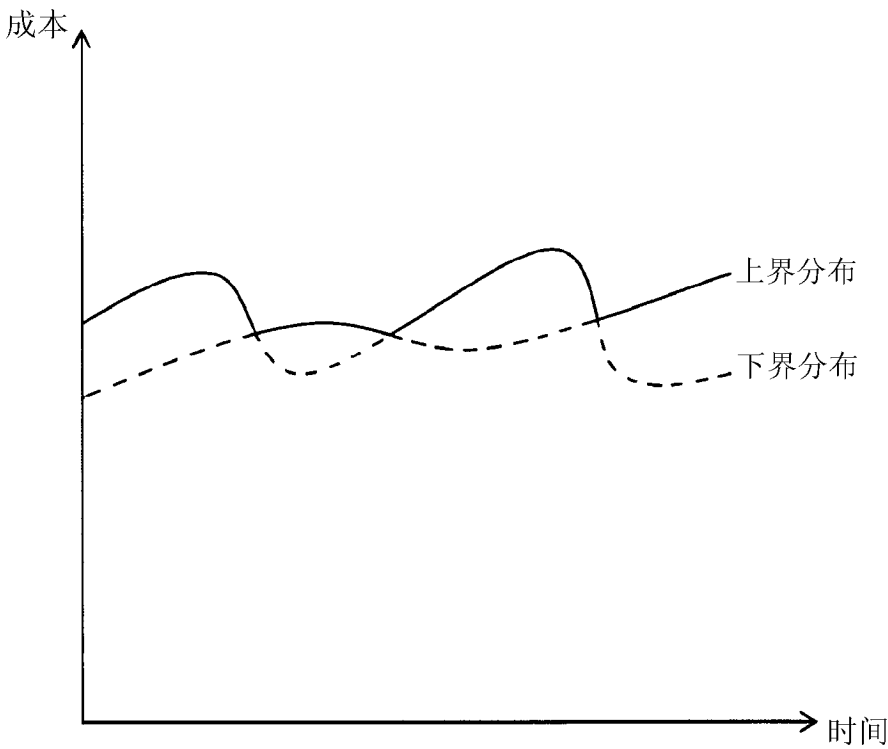


图 31