



## (12) 发明专利申请

(10) 申请公布号 CN 116467322 A

(43) 申请公布日 2023. 07. 21

(21) 申请号 202310325455.X

(22) 申请日 2023.03.29

(71) 申请人 北京四维图新科技股份有限公司  
地址 100094 北京市海淀区丰豪东路四维  
图新大厦A座10层

(72) 发明人 盛光伟 李心榕 郭林熹 刘茜  
李兴涛 张建平

(74) 专利代理机构 北京同立钧成知识产权代理  
有限公司 11205  
专利代理师 马明明 黄健

(51) Int.Cl.

G06F 16/23 (2019.01)

G06F 16/29 (2019.01)

G06F 18/22 (2023.01)

G06Q 50/30 (2012.01)

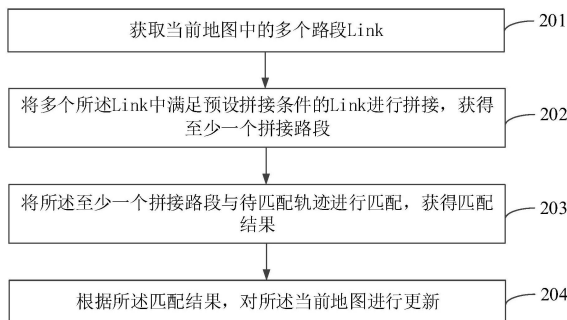
权利要求书2页 说明书10页 附图5页

### (54) 发明名称

地图更新方法、设备、存储介质及程序产品

### (57) 摘要

本申请实施例提供一种地图更新方法、设备、存储介质及程序产品，该方法包括获取当前地图中的多个路段Link，将多个Link中满足预设拼接条件的Link进行拼接，获得至少一个拼接路段，将至少一个拼接路段与待匹配轨迹进行匹配，获得匹配结果，根据匹配结果，对当前地图进行更新。本实施例的方法通过将多个Link进行拼接，得到长度较长的拼接路段，进而基于拼接路段与众包车辆的轨迹进行匹配，能够提高匹配效率和准确率，进而提高要素识别效率和准确率，从而提高地图更新效率和准确率。



1. 一种地图更新方法,其特征在于,包括:

获取当前地图中的多个路段Link;

将多个所述Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段;

将所述至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果;

根据所述匹配结果,对所述当前地图进行更新。

2. 根据权利要求1所述的方法,其特征在于,所述将多个所述Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段,包括:

从多个所述Link中选定目标Link;

针对剩余Link中的每个Link,根据所述Link的端点位置、所述目标Link的端点位置、以及所述Link和所述目标Link之间的方向角夹角,确定所述Link与所述目标Link是否满足预设拼接条件;若是,则将所述Link确定为待拼接Link;

根据至少一个所述待拼接Link和所述目标Link确定拼接路段。

3. 根据权利要求2所述的方法,其特征在于,所述根据所述Link的端点位置、所述目标Link的端点位置、以及所述Link和所述目标Link之间的方向角夹角,确定所述Link与所述目标Link是否满足预设拼接条件,包括:

若所述Link的第一端的端点位置与所述目标Link的第二端的端点位置相一致,且所述Link的第一端和所述目标Link的第二端之间的方向角夹角小于或等于预设阈值,则判定所述Link与所述目标Link满足预设拼接条件。

4. 根据权利要求2所述的方法,其特征在于,所述从多个所述Link中选定目标Link,包括:

将多个所述Link中长度最长的Link选定为目标Link。

5. 根据权利要求2所述的方法,其特征在于,所述根据至少一个所述待拼接Link和所述目标Link确定拼接路段,包括:

将所述剩余Link除所述至少一个待拼接Link之外的Link确定为新的剩余Link;

针对所述至少一个待拼接Link中的每一个待拼接Link,将所述待拼接Link确定为新的目标Link;针对所述新的剩余Link中的每个Link,根据所述Link的端点位置、所述新的目标Link的端点位置、以及所述Link和所述新的目标Link之间的方向角夹角,确定所述Link与所述新的目标Link是否满足预设拼接条件;若是,则将所述Link确定为新的待拼接Link;

根据所述至少一个待拼接Link、至少一个所述新的待拼接Link和所述目标Link确定拼接路段。

6. 根据权利要求2所述的方法,其特征在于,所述根据至少一个所述待拼接Link和所述目标Link确定拼接路段,包括:

将至少一个所述待拼接Link和所述目标Link均转换为字符串;

将所述字符串中的坐标进行合并,获得拼接字符串;

根据所述拼接字符串,基于第一几何构造函数生成拼接路段。

7. 根据权利要求2所述的方法,其特征在于,所述根据至少一个所述待拼接Link和所述目标Link确定拼接路段,包括:

基于第二几何构造函数,将至少一个所述待拼接Link和所述目标Link进行直接拼接,获得拼接路段。

8. 根据权利要求1-7任一项所述的方法,其特征在于,所述获取当前地图中的多个路段Link,包括:

获取所述待匹配轨迹的地图变更请求;

根据所述地图变更请求确定变化区域;

获取当前地图中在所述变化区域内的多个路段Link。

9. 一种地图更新设备,其特征在于,包括:

获取模块,用于获取当前地图中的多个路段Link;

拼接模块,用于将多个所述Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段;

匹配模块,用于将所述至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果;

更新模块,用于根据所述匹配结果,对所述当前地图进行更新。

10. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如权利要求1至8任一项所述的地图更新方法。

11. 一种计算机程序产品,包括计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至8任一项所述的地图更新方法。

## 地图更新方法、设备、存储介质及程序产品

### 技术领域

[0001] 本申请实施例涉及高精度地图技术领域,尤其涉及一种地图更新方法、设备、存储介质及程序产品。

### 背景技术

[0002] 随着L2/L2+智能驾驶的快速普及,如何实现地图的快速、准确更新变得越来越重要。

[0003] 相关技术中,通常采用众包的方式进行地图更新。具体的,可以将道路矢量数据(Link)与众包车辆的轨迹进行匹配,进而基于匹配结果进行更新。

[0004] 然而,实现本申请过程中,发明人发现现有技术中至少存在如下问题:上述地图更新方式中,道路矢量数据中的多个Link长度多样,在与众包车辆的轨迹进行匹配时,匹配效率和准确率较低,影响地图更新效率和准确率。

### 发明内容

[0005] 本申请实施例提供一种地图更新方法、设备、存储介质及程序产品,以提高匹配效率和准确率,进而提高地图更新效率和准确率。

[0006] 第一方面,本申请实施例提供一种地图更新方法,包括:

[0007] 获取当前地图中的多个路段Link;

[0008] 将多个所述Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段;

[0009] 将所述至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果;

[0010] 根据所述匹配结果,对所述当前地图进行更新。

[0011] 在一种可能的设计中,所述将多个所述Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段,包括:

[0012] 从多个所述Link中选定目标Link;

[0013] 针对剩余Link中的每个Link,根据所述Link的端点位置、所述目标Link的端点位置、以及所述Link和所述目标Link之间的方向角夹角,确定所述Link与所述目标Link是否满足预设拼接条件;若是,则将所述Link确定为待拼接Link;

[0014] 根据至少一个所述待拼接Link和所述目标Link确定拼接路段。

[0015] 在一种可能的设计中,所述根据所述Link的端点位置、所述目标Link的端点位置、以及所述Link和所述目标Link之间的方向角夹角,确定所述Link与所述目标Link是否满足预设拼接条件,包括:

[0016] 若所述Link的第一端的端点位置与所述目标Link的第二端的端点位置相一致,且所述Link的第一端和所述目标Link的第二端之间的方向角夹角小于或等于预设阈值,则判定所述Link与所述目标Link满足预设拼接条件。

[0017] 在一种可能的设计中,所述从多个所述Link中选定目标Link,包括:

[0018] 将多个所述Link中长度最长的Link选定为目标Link。

[0019] 在一种可能的设计中,所述根据至少一个所述待拼接Link和所述目标Link确定拼接路段,包括:

[0020] 将所述剩余Link除所述至少一个待拼接Link之外的Link确定为新的剩余Link;

[0021] 针对所述至少一个待拼接Link中的每一个待拼接Link,将所述待拼接Link确定为新的目标Link;针对所述新的剩余Link中的每个Link,根据所述Link的端点位置、所述新的目标Link的端点位置、以及所述Link和所述新的目标Link之间的方向角夹角,确定所述Link与所述新的目标Link是否满足预设拼接条件;若是,则将所述Link确定为新的待拼接Link;

[0022] 根据所述至少一个待拼接Link、至少一个所述新的待拼接Link和所述目标Link确定拼接路段。

[0023] 在一种可能的设计中,所述根据至少一个所述待拼接Link和所述目标Link确定拼接路段,包括:

[0024] 将至少一个所述待拼接Link和所述目标Link均转换为字符串;

[0025] 将所述字符串中的坐标进行合并,获得拼接字符串;

[0026] 根据所述拼接字符串,基于第一几何构造函数生成拼接路段。

[0027] 在一种可能的设计中,所述根据至少一个所述待拼接Link和所述目标Link确定拼接路段,包括:

[0028] 基于第二几何构造函数,将至少一个所述待拼接Link和所述Link进行直接拼接,获得拼接路段。

[0029] 在一种可能的设计中,所述获取当前地图中的多个路段Link,包括:

[0030] 获取所述待匹配轨迹的地图变更请求;

[0031] 根据所述地图变更请求确定变化区域;

[0032] 获取当前地图中在所述变化区域内的多个路段Link。

[0033] 第二方面,本申请实施例提供一种地图更新设备,包括:

[0034] 获取模块,用于获取当前地图中的多个路段Link;

[0035] 拼接模块,用于将多个所述Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段;

[0036] 匹配模块,用于将所述至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果;

[0037] 更新模块,用于根据所述匹配结果,对所述当前地图进行更新。

[0038] 第三方面,本申请实施例提供一种地图更新设备,包括:至少一个处理器和存储器;

[0039] 所述存储器存储计算机执行指令;

[0040] 所述至少一个处理器执行所述存储器存储的计算机执行指令,使得所述至少一个处理器执行如上第一方面以及第一方面各种可能的设计所述的方法。

[0041] 第四方面,本申请实施例提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上第一方面以及第一方面各种可能的设计所述的方法。

[0042] 第五方面,本申请实施例提供一种计算机程序产品,包括计算机程序,所述计算机程序被处理器执行时,实现如上第一方面以及第一方面各种可能的设计所述的方法。

[0043] 本实施例提供的地图更新方法、设备、存储介质及程序产品,该方法包括获取当前地图中的多个路段Link,将多个Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段,将至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果,根据匹配结果,对当前地图进行更新。本实施例的方法通过将多个Link进行拼接,得到长度较长的拼接路段,进而基于拼接路段与众包车辆的轨迹进行匹配,能够提高匹配效率和准确率,进而提高要素识别效率和准确率,从而提高地图更新效率和准确率。

## 附图说明

[0044] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0045] 图1为本申请实施例提供的路口处轨迹匹配示意图;

[0046] 图2为本申请实施例提供的地图更新方法的流程示意图;

[0047] 图3a为本申请实施例提供的路口的变化区域的示意图;

[0048] 图3b为本申请实施例提供的兴趣点的变化区域的示意图;

[0049] 图4为本申请实施例提供的单向顺方向的Link的结构示意图;

[0050] 图5为本申请实施例提供的单向逆方向的Link的结构示意图;

[0051] 图6为本申请实施例提供的方向角夹角匹配规则的原理示意图一;

[0052] 图7为本申请实施例提供的方向角夹角匹配规则的原理示意图二;

[0053] 图8为本申请实施例提供的方向角夹角匹配规则的原理示意图三;

[0054] 图9为本申请实施例提供的方向角夹角匹配规则的原理示意图四;

[0055] 图10为本申请实施例提供的方向角夹角匹配规则的原理示意图五;

[0056] 图11为本申请实施例提供的方向角夹角匹配规则的原理示意图六;

[0057] 图12为本申请实施例提供的地图更新设备的结构示意图;

[0058] 图13为本申请实施例提供的地图更新设备的硬件结构示意图。

## 具体实施方式

[0059] 为使本申请实施例的目的、技术方案和优点更加清楚,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0060] 随着L2/L2+智能驾驶技术的快速普及,如何实现地图的快速、准确更新变得越来越重要。对于地图提供商来讲,为了保证地图的质量,通常会先利用分布在全国的外业采集团队携带专业设备进行现场数据采集,然后提交到数据库进行地图制作,之后再分发给用户。这种方法虽然能够获取到高质量的地图数据,但是地图的鲜度通常为月度级或者季度级,无法满足智能驾驶对地图鲜度的需求。随着汽车上搭载的传感器越来越多,利用这些车辆进行地图更新,不仅能有效降低制图成本,而且能将地图的鲜度提高到天级或者周级,满足智能驾驶的需求。

[0061] 相关技术中,利用非自有的浮动车进行地图更新通常被称为众包地图更新,这种更新方式主要包括以下步骤:将相关矢量数据(Link)与众包车辆的轨迹进行轨迹匹配;基于得到的匹配结果对当前地图进行更新。

[0062] 然而,通常来讲,如何建立众包车辆轨迹与已有的Link的有效匹配是能否获取到有效的、连续的多源数据的关键,换言之,已有Link是否足够连贯,是否满足正常行车的要求非常重要。而在上述地图更新方式中,已有的Link的长度通常不固定,少则十几米,多则几百米,如果将其直接与众包车辆轨迹进行匹配,后续如果采用图像识别算法进行要素识别,不连贯的数据会对识别效率和识别准确率产生影响;如果采用人工进行要素识别,大量不连贯的数据也会降低人工的作业效率和准确率。

[0063] 为解决上述技术问题,发明人研究发现,可以将Link进行拼接得到长度较长的拼接路段,这样基于拼接得到的拼接路段进行匹配时,能够避免因为数据不连贯影响识别效率和准确率。基于此,本申请提供一种地图更新方法,通过将多个Link进行拼接,得到长度较长的拼接路段,进而基于拼接路段与众包车辆的轨迹进行匹配,能够提高匹配效率和准确率,进而提高要素识别效率和准确率,从而提高地图更新效率和准确率。

[0064] 另外,在相关技术中,如图1所示,在路口等需要改变行车方向的场景,车辆不会严格按照道路Link的形状行驶,二者在转弯处很难进行匹配,因此加粗路段较易匹配成功,进而基于匹配结果有效获取到地图更新所需的多源数据进行地图更新,非加粗路段较难匹配成功,从而无法获取到多源数据。

[0065] 针对该技术问题,发明人进一步发现,可以利用道路Link的通行方向和夹角构建道路Link设置拼接条件,保证拼接后的Link符合行车轨迹,避免数据的不连贯,从而提高道路Link与车辆轨迹匹配率。

[0066] 下面以具体地实施例对本申请的技术方案进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例不再赘述。

[0067] 图2为本申请实施例提供的地图更新方法的流程示意图。如图2所示,该方法包括:

[0068] 201、获取当前地图中的多个路段Link。

[0069] 具体的,本实施例的执行主体可以是服务器。例如云端服务器或本地服务器,可以采用多种软件系统,例如Windows系统或Linux系统。

[0070] 本实施例中,Link是指构成地图的元素,各Link可以在经过渲染后,显示于用户终端的显示屏上,视觉呈现上各Link是连续的状态。

[0071] 在一些实施例中,为了减少计算量,可以仅获取变化区域,也即需要进行更新的区域内的Link。具体的,获取当前地图中的多个路段Link,可以包括:获取待匹配轨迹的地图变更请求;根据地图变更请求确定变化区域;获取当前地图中在变化区域内的多个路段Link。

[0072] 示例性的,当路口附近发生变化,需要更新地图时,则将路口一定范围划定为变化区域,获取该变化区域内的Link。具体的,如图3a所示,可以针对每个路口点,以路口点为圆心,以第一预设长度为半径划定该路口点对应的圆形区域。则4个路口点分别对应的圆形区域就构成了变化区域,可以获取当前地图中该变化区域内的多个Link。当然,以上仅为示例,还可以以十字路口的中心点为圆心,以第二预设长度为半径划定圆形区域,以该圆形区域为变化区域。具体划定何种形状的区域,是否划定多个区域进行组合,可根据实际需要进

行设定,本实施例对此不作限定。

[0073] 示例性的,当需要对某个POI点一定范围对应的地图进行更新时,变化区域的确定过程:如图3b所示,从POI点向最近的道路Link做垂线,以垂足为圆心构建半径为X的圆形缓冲区,将给缓冲区确定为变化区域。

[0074] 202、将多个Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段。

[0075] 具体的,考虑到多个Link中大多是长度较短,例如几十米,一两百米的路段,如果将其直接与众包车辆轨迹进行匹配,后续如果采用图像识别算法进行要素识别,长度较短的多个Link构成的不连贯的数据会对识别效率以及识别准确率产生影响;如果采用人工进行要素识别,大量不连贯的数据也会降低人工的作业效率和准确率。因此,可以将多个Link中满足预设拼接条件的Link进行拼接,将两个以上的Link连接为一个新的Link即拼接路段,这样就加长了Link的长度,进而可以在匹配过程中,提高要素识别的效率和准确率。

[0076] 在一些实施例中,可以基于Link的端点位置(例如坐标,或者node值(参见图4所示))将端点位置一致的两个Link(即一个Link的一个端点与另一个Link的一个端点的端点位置一致)进行直接拼接。可选地,两个端点的端点位置一致可以是两个端点之间的距离小于预设距离,例如,第一Link的第一端的端点位置与第二Link的第二端的端点位置之间的距离小于预设距离,则可以将该第一端与该第二端进行拼接,从而实现了第一Link与第二Link的拼接。

[0077] 在一些实施例中,为了保证拼接的准确性以及提高拼接得到的拼接路段与众包车辆的轨迹的匹配率,可以对待拼接的两个Link的方向角夹角进行限定,作为预设拼接条件之一。具体的,将多个Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段,可以包括:从多个Link中选定目标Link;针对剩余Link中的每个Link,根据Link的端点位置、目标Link的端点位置、以及Link和目标Link之间的方向角夹角,确定Link与目标Link是否满足预设拼接条件;若是,则将Link确定为待拼接Link;根据至少一个待拼接Link和目标Link确定拼接路段。

[0078] 可选地,根据Link的端点位置、目标Link的端点位置、以及Link和目标Link之间的方向角夹角,确定Link与目标Link是否满足预设拼接条件,可以包括:若Link的第一端的端点位置与目标Link的第二端的端点位置相一致,且Link的第一端和目标Link的第二端之间的方向角夹角小于或等于预设阈值,则判定Link与目标Link满足预设拼接条件。可选地,根据至少一个待拼接Link和目标Link确定拼接路段,可以包括:将剩余Link除至少一个待拼接Link之外的Link确定为新的剩余Link;针对至少一个待拼接Link中的每一个待拼接Link,将待拼接Link确定为新的目标Link;针对新的剩余Link中的每个Link,根据Link的端点位置、新的目标Link的端点位置、以及Link和新的目标Link之间的方向角夹角,确定Link与新的目标Link是否满足预设拼接条件;若是,则将Link确定为新的待拼接Link;根据至少一个待拼接Link、至少一个新的待拼接Link和目标Link确定拼接路段。

[0079] 在一些实施例中,为了提高拼接效率和有效性,从多个Link中选定目标Link,可以包括:将多个Link中长度最长的Link选定为目标Link。

[0080] 在一些实施例中,在确定了至少一个待拼接Link后,可以将待拼接Link与目标Link进行拼接处理。拼接处理的方式有多种,在一种可实现方式中,可以将待拼接Link和目标Link进行格式转换等预处理后再进行拼接,即实现间接拼接,具体的,根据至少一个待拼



接Link和目标Link确定拼接路段,可以包括:将至少一个待拼接Link和目标Link均转换为字符串;将字符串中的坐标进行合并,获得拼接字符串;根据拼接字符串,基于第一几何构造函数(例如ST\_LinestringFromText函数)生成拼接路段。在另一种方式中,可以无需对待拼接Link和目标Link进行格式转换,直接采用对应几何构造函数进行处理。具体的,根据至少一个待拼接Link和目标Link确定拼接路段,可以包括:基于第二几何构造函数(例如ST\_LineMerge函数),将至少一个待拼接Link和目标Link进行直接拼接,获得拼接路段。具体可参照以下实施例中的步骤2028,此处不再赘述。

[0081] 示例性的,如图4和5所示,从通行方向上来看,Link包括单向顺方向、双向、以及单向逆方向等类型。单个Link可以包括两端的node点、形状点,其中,涉及的名词定义可参考以下:

[0082] Node点:Link的端点,相同的Node值代表Link在此处存在挂接关系。用snode表示Link的起始端点,enode表示Link的结束端点。

[0083] 形状点:构成Link形状的点。

[0084] Link:由两端的Node点和中间的形状点按照一定顺序构成,是现实世界道路经数字化后的表现形式。

[0085] 通行方向:当Link为单向通行时,顺方向指通行方向为snode到enode的方向,逆方向指通行方向为enode到snode的方向;当Link为双向通行时,默认为顺方向,即通行方向为snode到enode的方向;

[0086] Link通行起终点:spoint表示Link通行方向的起点,epoint表示Link通行方向的终点。

[0087] 通行方位角:sdegree表示spoint处的通行方位角,edegree表示epoint处的通行方位角。

[0088] 结合图4和图5提供的Link,拼接过程可以包括以下步骤:

[0089] 用  $\Omega = \Omega 1 \cup \omega n \cup \Omega 2$  表示所有的Link集合, $\Phi$  表示Link拼接结果集合,其中  $\Omega 1$  表示还未与其他Link拼接过或者未作为起始Link过的Link集合, $\omega n$  表示正在参与拼接的Link集合, $\Omega 2$  表示已与其他Link拼接过或者未找到可拼接Link的起始Link的Link集合。

[0090] 2021、从  $\Omega 1$  中选择长度最长的一条Link作为起始Link,用1A表示,将其添加到  $\omega n$  中;

[0091] 2022、将1A的epoint设置为起始拼接处,基于epoint的Node值,在  $\Omega 1$  剩下的Link中寻找spoint具有相同Node值的Link,用1B表示;

[0092] 2023、判断1B是否符合拼接条件,用T表示1A和1B拼接处通行方位角的夹角阈值(以下三种情况的划分可以有效的避免在计算过程中出现超出 $0^\circ$ 至 $360^\circ$ 的范围的夹角(例如负数的角,大于 $360^\circ$ 的角),便于服务器进行计算):

[0093] 如图6所示,当  $\text{edegreeA}-T \geq 0^\circ$  且  $\text{edegreeA}+T \leq 360^\circ$  时,用  $\alpha 1$  表示  $\text{edegreeA}-T$ ,  $\alpha 2$  表示  $\text{edegreeA}+T$ ,则  $\text{sdegreeB} \in [\alpha 1, \alpha 2]$ 。

[0094] 如图7所示,当  $\text{edegreeA}-T < 0^\circ$  且  $\text{edegreeA}+T \leq 360^\circ$  时,用  $\alpha 1$  表示  $360^\circ + \text{edegreeA}-T$ ,  $\alpha 2$  表示  $\text{edegreeA}+T$ ,则  $\text{sdegreeB} \in [\alpha 1, 360^\circ] \cup [0^\circ, \alpha 2]$ 。

[0095] 如图8所示,当  $\text{edegreeA}-T \geq 0^\circ$  且  $\text{edegreeA}+T \geq 360^\circ$  时,用  $\alpha 1$  表示  $\text{edegreeA}-T$ ,  $\alpha 2$  表示  $\text{edegreeA}+T-360^\circ$ ,则  $\text{sdegreeB} \in [\alpha 1, 360^\circ] \cup [0^\circ, \alpha 2]$ 。

[0096] 示例性的,因1A在拼接处的通行方位角 $e\text{degreeA}$ 加或减去夹角阈值 $T$ 可能存在小于 $0^\circ$ 或大于 $360^\circ$ 的情况,无法与1B的通行方位角 $s\text{degreeB}$ 直接进行比较,因此需要分情况讨论,假设夹角阈值 $T=45^\circ$ :

[0097] 当 $e\text{degreeA}=60^\circ$ 时,满足步骤(3)的①的规则描述( $e\text{degreeA}-T\geq 0^\circ$ 且 $e\text{degreeA}+T\leq 360^\circ$ ),则 $\alpha_1=e\text{degreeA}-T=15^\circ$ , $\alpha_2=e\text{degreeA}+T=105^\circ$ ,即 $s\text{degreeB}$ 需要位于 $[15^\circ, 105^\circ]$ 的区间内,1A才能与1B拼接;

[0098] 当 $e\text{degreeA}=10^\circ$ 时,满足步骤(3)的②中的规则描述( $e\text{degreeA}-T<0^\circ$ 且 $e\text{degreeA}+T\leq 360^\circ$ ),则 $\alpha_1=360^\circ+e\text{degreeA}-T=325^\circ$ , $\alpha_2=e\text{degreeA}+T=55^\circ$ ,即 $s\text{degreeB}$ 需要位于 $[325^\circ, 360^\circ]$ 或 $[0^\circ, 55^\circ]$ 的区间内,1A才能与1B拼接。

[0099] 当 $e\text{degreeA}=350^\circ$ 时,满足步骤(3)的③中的规则描述( $e\text{degreeA}-T\geq 0^\circ$ 且 $e\text{degreeA}+T\geq 360^\circ$ ),则 $\alpha_1=e\text{degreeA}-T=305^\circ$ , $\alpha_2=e\text{degreeA}+T-360^\circ=35^\circ$ ,即 $s\text{degreeB}$ 需要位于 $[305^\circ, 360^\circ]$ 或 $[0^\circ, 35^\circ]$ 的区间内,1A才能与1B拼接。

[0100] 由上述描述可知,通过界定三个规则,能够避免在计算过程中出现超出 $0^\circ$ 至 $360^\circ$ 的范围的夹角(例如负数的角,大于 $360^\circ$ 的角),便于服务器进行准确的计算。

[0101] 2024、如果存在可拼接的Link(存在多条时,可以任选其中一条),将其添加到 $\omega_n$ 中,并将此Link指定为1A的 $e\text{point}$ 设置为起始拼接处,重复执行步骤2022、2023,直到无可拼接的Link;

[0102] 2025、如果不存在可拼接的Link,将1A的 $s\text{point}$ 设置为起始拼接处,基于 $s\text{point}$ 的Node值,在 $\Omega_1$ 剩下的Link中寻找 $e\text{point}$ 具有相同Node值的Link,用1C表示;

[0103] 2026、判断1C是否符合拼接条件,用 $T$ 表示1A和1C拼接处通行方位角的夹角阈值:

[0104] 如图9所示,当 $s\text{degreeA}-T\geq 0^\circ$ 且 $s\text{degreeA}+T\leq 360^\circ$ 时,用 $\alpha_1$ 表示 $s\text{degreeA}-T$ , $\alpha_2$ 表示 $s\text{degreeA}+T$ ,则 $e\text{degreeC}\in[\alpha_1, \alpha_2]$ 。

[0105] 如图10所示,当 $s\text{degreeA}-T<0^\circ$ 且 $s\text{degreeA}+T\leq 360^\circ$ 时,用 $\alpha_1$ 表示 $360^\circ+s\text{degreeA}-T$ , $\alpha_2$ 表示 $s\text{degreeA}+T$ ,则 $e\text{degreeC}\in[\alpha_1, 360^\circ]\cup[0^\circ, \alpha_2]$ 。

[0106] 如图11所示,当 $s\text{degreeA}-T\geq 0^\circ$ 且 $s\text{degreeA}+T\geq 360^\circ$ 时,用 $\alpha_1$ 表示 $s\text{degreeA}-T$ , $\alpha_2$ 表示 $s\text{degreeA}+T-360^\circ$ ,则 $e\text{degreeC}\in[\alpha_1, 360^\circ]\cup[0^\circ, \alpha_2]$ 。

[0107] 示例性的,因1A在拼接处的通行方位角 $s\text{degreeA}$ 加或减去夹角阈值 $T$ 可能存在小于 $0^\circ$ 或大于 $360^\circ$ 的情况,无法与1C的通行方位角 $e\text{degreeC}$ 直接进行比较,因此需要分情况讨论,假设夹角阈值 $T=45^\circ$ :

[0108] 当 $s\text{degreeA}=60^\circ$ 时,满足规则描述( $s\text{degreeA}-T\geq 0^\circ$ 且 $s\text{degreeA}+T\leq 360^\circ$ ),则 $\alpha_1=s\text{degreeA}-T=15^\circ$ , $\alpha_2=s\text{degreeA}+T=105^\circ$ ,即 $e\text{degreeC}$ 需要位于 $[15^\circ, 105^\circ]$ 的区间内,1A才能与1C拼接。

[0109] 当 $s\text{degreeA}=10^\circ$ 时,满足规则描述( $s\text{degreeA}-T<0^\circ$ 且 $s\text{degreeA}+T\leq 360^\circ$ ),则 $\alpha_1=360^\circ+s\text{degreeA}-T=325^\circ$ , $\alpha_2=s\text{degreeA}+T=55^\circ$ ,即 $e\text{degreeC}$ 需要位于 $[325^\circ, 360^\circ]$ 或 $[0^\circ, 55^\circ]$ 的区间内,1A才能与1C拼接。

[0110] 当 $s\text{degreeA}=350^\circ$ 时,满足规则描述( $s\text{degreeA}-T\geq 0^\circ$ 且 $s\text{degreeA}+T\geq 360^\circ$ ),则 $\alpha_1=s\text{degreeA}-T=305^\circ$ , $\alpha_2=s\text{degreeA}+T-360^\circ=35^\circ$ ,即 $e\text{degreeC}$ 需要位于 $[305^\circ, 360^\circ]$ 或 $[0^\circ, 35^\circ]$ 的区间内,1A才能与1C拼接。

[0111] 由上述描述可知,通过界定三个规则,能够避免在计算过程中出现超出 $0^\circ$ 至 $360^\circ$

的范围的夹角(例如负数的角,大于 $360^{\circ}$ 的角),便于服务器进行准确的计算。

[0112] 2027、如果存在可拼接的Link(存在多条时,任选其中一条),将其添加到 $\omega_n$ 中,并将此Link指定为 $l_A$ 的spoint设置为起始拼接处,重复执行步骤(5)、(6),直到无可拼接的Link;

[0113] 2028、将 $\omega_n$ 中所有的Link从 $\Omega_1$ 中移除,并添加到 $\Omega_2$ 中,使用开源库PostGIS中的ST\_LineMerge函数将 $\omega_n$ 中所有的Link拼接成一条,将该条拼接结果添加到 $\Phi$ 中;或者,还可以将 $\omega_n$ 中的link转换为WKT字符串,然后合并WKT字符串中的坐标为一条WKT字符串,最终通过几何构造函数,例如,ST\_LinestringFromText构造出一条Link。

[0114] 2029、重复步骤2021-2028,直到 $\Omega_1$ 中不存在Link为止,此时 $\Phi$ 即是 $\Omega$ 中所有Link的拼接结果,即得到的至少一个待拼接路段。

[0115] 203、将至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果。

[0116] 204、根据匹配结果,对当前地图进行更新。

[0117] 具体的,在获得至少一个待拼接路段后,可以基于该至少一个待拼接路段与待匹配轨迹进行匹配,获得匹配结果(例如与待匹配轨迹的匹配率大于预设阈值的拼接路段)并基于匹配结果获取众包车辆中对应的多源数据(例如图像、点云等)进而可以采用图像识别等自动化算法或者人工识别出标牌、红绿灯、电子眼等地图要素,从而可以将识别到的要素与地图数据库进行差分,将变化的要素更新到地图数据库中,实现对当前地图的更新,得到最新版本的地图,以将该最新版本的地图分发给用户。

[0118] 本实施例提供的地图更新方法,通过将多个Link进行拼接,得到长度较长的拼接路段,进而基于拼接路段与众包车辆的轨迹进行匹配,能够提高匹配效率和准确率,进而提高要素识别效率和准确率,从而提高地图更新效率和准确率。

[0119] 图12为本申请实施例提供的地图更新设备的结构示意图一。如图12所示,该地图更新设备120包括:获取模块1201、拼接模块1202、匹配模块1203和更新模块1204。

[0120] 获取模块1201,用于获取当前地图中的多个路段Link。

[0121] 拼接模块1202,用于将多个Link中满足预设拼接条件的Link进行拼接,获得至少一个拼接路段。

[0122] 匹配模块1203,用于将至少一个拼接路段与待匹配轨迹进行匹配,获得匹配结果。

[0123] 更新模块1204,用于根据匹配结果,对当前地图进行更新。

[0124] 本申请实施例提供的地图更新设备,通过将多个Link进行拼接,得到长度较长的拼接路段,进而基于拼接路段与众包车辆的轨迹进行匹配,能够提高匹配效率和准确率,进而提高要素识别效率和准确率,从而提高地图更新效率和准确率。

[0125] 本申请实施例提供的地图更新设备,可用于执行上述的方法实施例,其实现原理和技术效果类似,本实施例此处不再赘述。

[0126] 图13为本申请实施例提供的地图更新设备的硬件结构示意图,该设备可以是计算机,服务器等。

[0127] 设备130可以包括以下一个或多个组件:处理组件1301,存储器1302,电源组件1303,多媒体组件1304,音频组件1305,输入/输出(I/O)接口1306,传感器组件1307、以及通信组件1308。

[0128] 处理组件1301通常控制设备130的整体操作,诸如与显示,电话呼叫,数据通信,相

机操作和记录操作相关联的操作。处理组件1301可以包括一个或多个处理器1309来执行指令,以完成上述的方法的全部或部分步骤。此外,处理组件1301可以包括一个或多个模块,便于处理组件1301和其他组件之间的交互。例如,处理组件1301可以包括多媒体模块,以方便多媒体组件1304和处理组件1301之间的交互。

[0129] 存储器1302被配置为存储各种类型的数据以支持在设备130的操作。这些数据的示例包括用于在设备130上操作的任何应用程序或方法的指令,联系人数据,电话簿数据,消息,图片,视频等。存储器1302可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。

[0130] 电源组件1303为设备130的各种组件提供电力。电源组件1303可以包括电源管理系统,一个或多个电源,及其他与为设备130生成、管理和分配电力相关联的组件。

[0131] 多媒体组件1304包括在设备130和用户之间提供一个输出接口的屏幕。在一些实施例中,屏幕可以包括液晶显示器(LCD)和触摸面板(TP)。如果屏幕包括触摸面板,屏幕可以被实现为触摸屏,以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。触摸传感器可以不仅感测触摸或滑动动作的边界,而且还检测与触摸或滑动操作相关的持续时间和压力。在一些实施例中,多媒体组件1304包括一个前置摄像头和/或后置摄像头。当设备130处于操作模式,如拍摄模式或视频模式时,前置摄像头和/或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0132] 音频组件1305被配置为输出和/或输入音频信号。例如,音频组件1305包括一个麦克风(MIC),当设备130处于操作模式,如呼叫模式、记录模式和语音识别模式时,麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器1302或经由通信组件1308发送。在一些实施例中,音频组件1305还包括一个扬声器,用于输出音频信号。

[0133] I/O接口1306为处理组件1301和外围接口模块之间提供接口,上述外围接口模块可以是键盘,点击轮,按钮等。这些按钮可包括但不限于:主页按钮、音量按钮、启动按钮和锁定按钮。

[0134] 传感器组件1307包括一个或多个传感器,用于为设备130提供各个方面的状态评估。例如,传感器组件1307可以检测到设备130的打开/关闭状态,组件的相对定位,例如组件为设备130的显示器和小键盘,传感器组件1307还可以检测设备130或设备130一个组件的位置改变,用户与设备130接触的存在或不存在,设备130方位或加速/减速和设备130的温度变化。传感器组件1307可以包括接近传感器,被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件1307还可以包括光传感器,如CMOS或CCD图像传感器,用于在成像应用中使用。在一些实施例中,该传感器组件1307还可以包括加速度传感器,陀螺仪传感器,磁传感器,压力传感器或温度传感器。

[0135] 通信组件1308被配置为便于设备130和其他设备之间有线或无线方式的通信。设备130可以接入基于通信标准的无线网络,如WiFi,2G或3G,或它们的组合。在一个示例性实施例中,通信组件1308经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中,通信组件1308还包括近场通信(NFC)模块,以促进短程通信。例

如,在NFC模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

[0136] 在示例性实施例中,设备130可以被一个或多个应用专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件实现,用于执行上述方法。

[0137] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器1302,上述指令可由设备130的处理器1309执行以完成上述方法。例如,非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0138] 上述的计算机可读存储介质,上述可读存储介质可以是由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。可读存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0139] 一种示例性的可读存储介质耦合至处理器,从而使处理器能够从该可读存储介质读取信息,且可向该可读存储介质写入信息。当然,可读存储介质也可以是处理器的组成部分。处理器和可读存储介质可以位于专用集成电路(Application Specific Integrated Circuits,简称:ASIC)中。当然,处理器和可读存储介质也可以作为分立组件存在于设备中。

[0140] 本领域普通技术人员可以理解:实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时,执行包括上述各方法实施例的步骤;而前述的存储介质包括:ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0141] 本申请实施例还提供一种计算机程序产品,包括计算机程序,计算机程序被处理器执行时,实现如上地图更新设备执行的地图更新方法。

[0142] 最后应说明的是:以上各实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述各实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

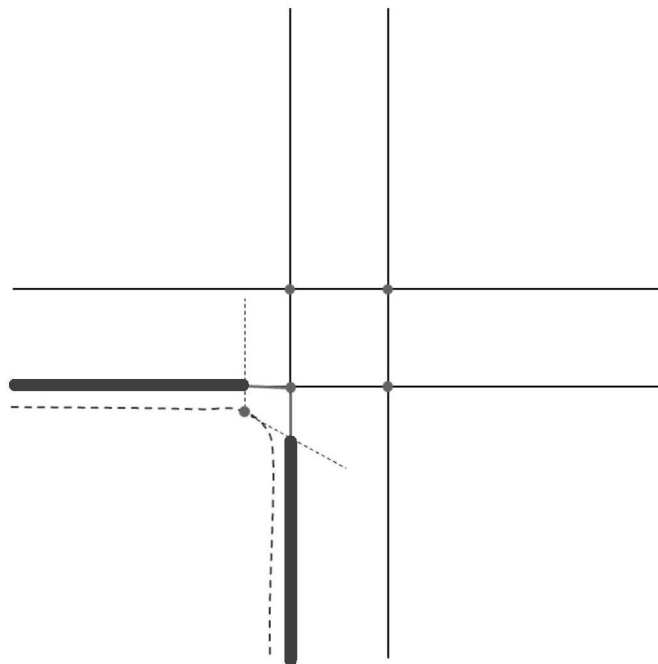


图1

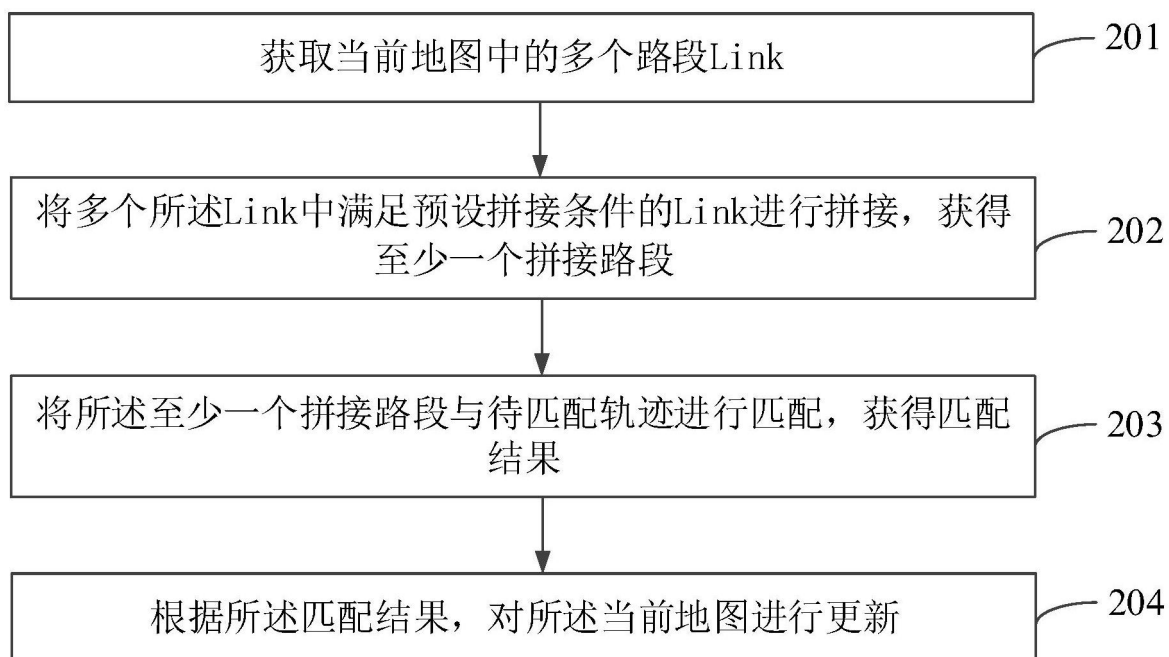


图2

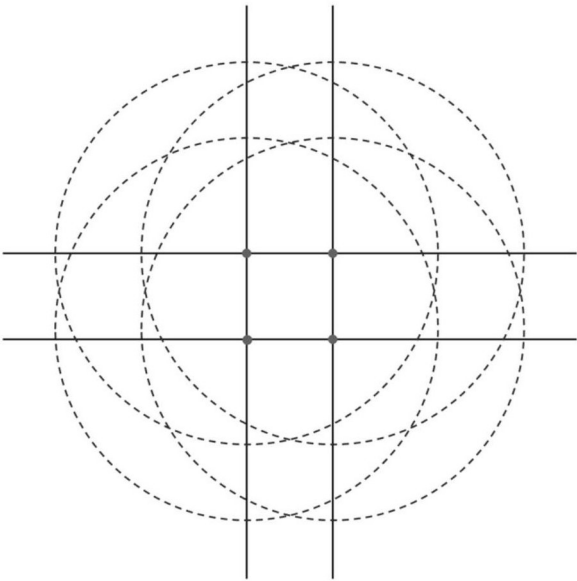


图3a

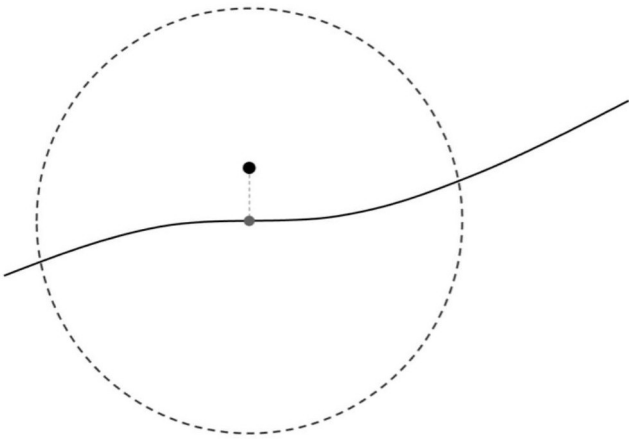


图3b

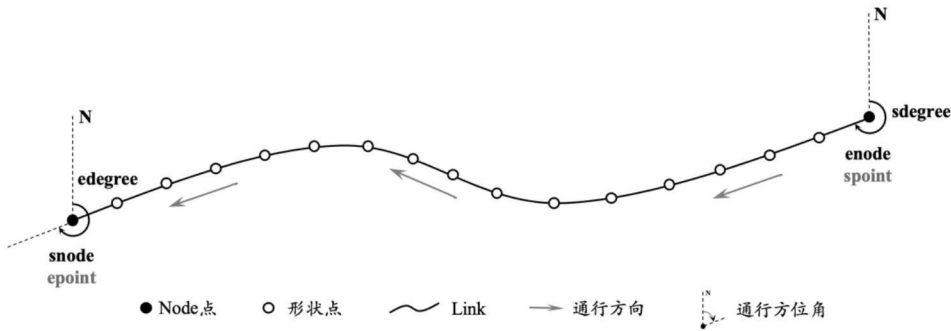


图4

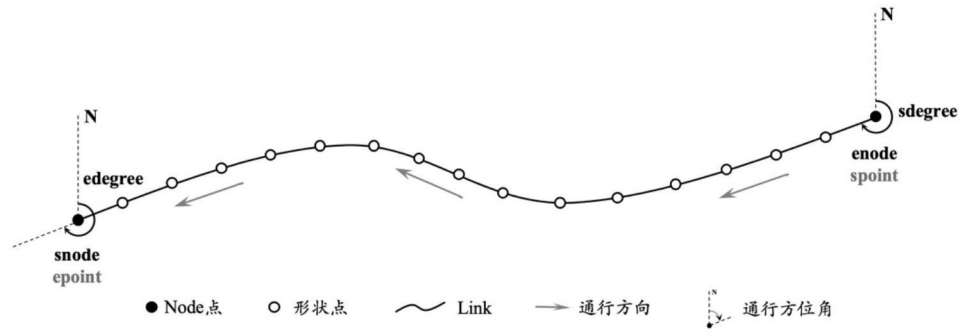


图5

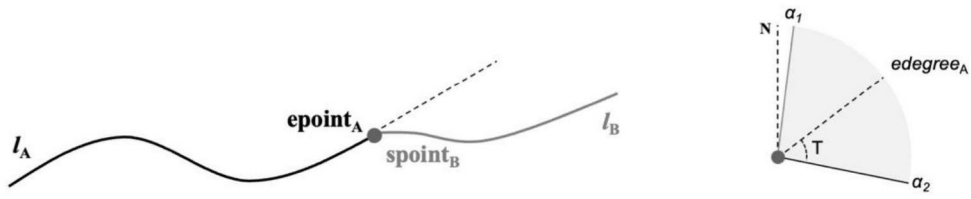


图6

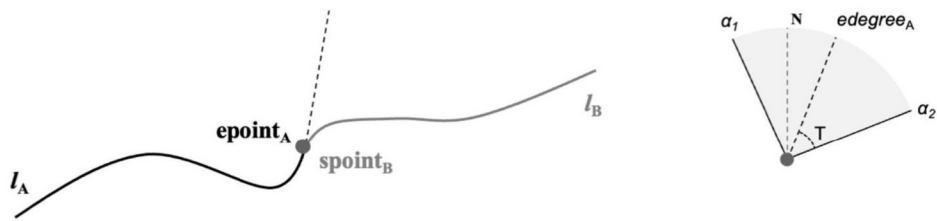


图7

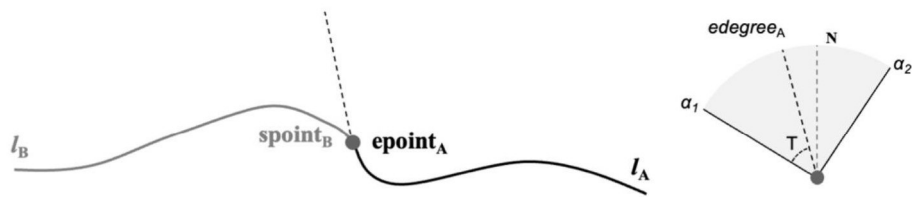


图8



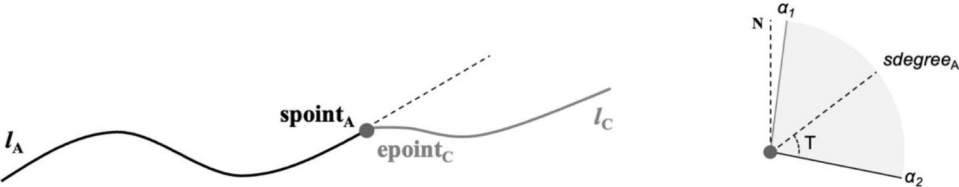


图9

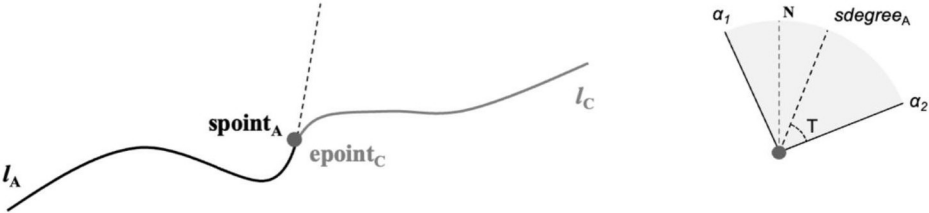


图10

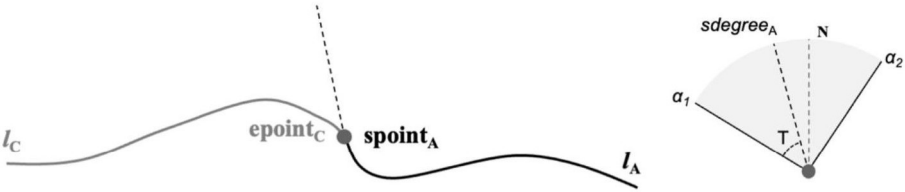


图11

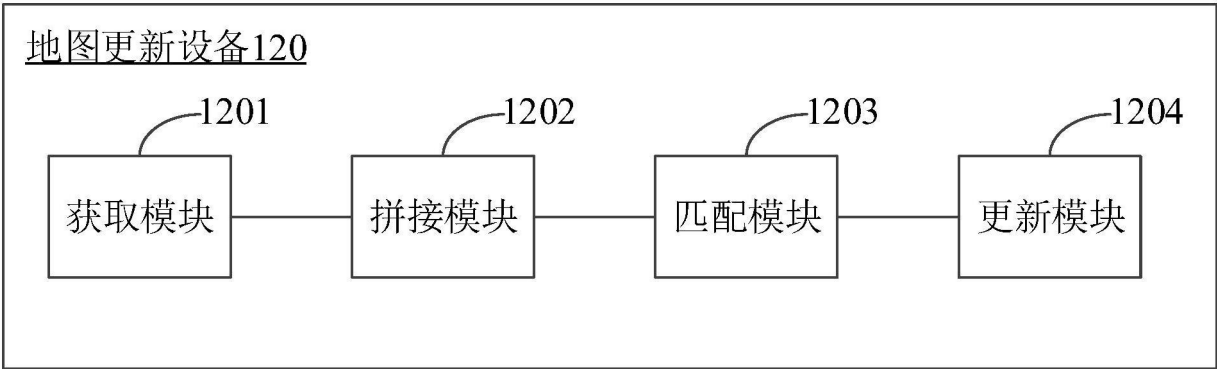


图12

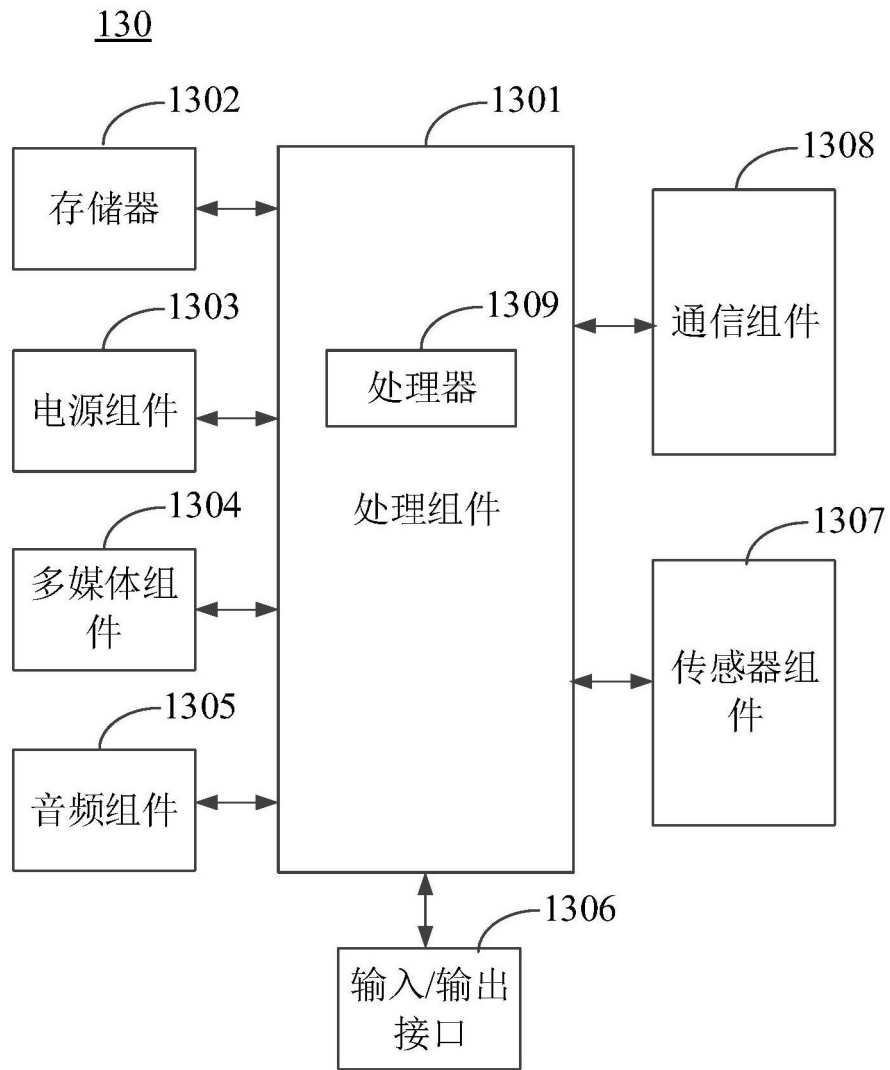


图13