



(12)发明专利

(10)授权公告号 CN 106019338 B

(45)授权公告日 2019.01.29

(21)申请号 201610297475.0

审查员 陈丹华

(22)申请日 2016.05.06

(65)同一申请的已公布的文献号

申请公布号 CN 106019338 A

(43)申请公布日 2016.10.12

(73)专利权人 上海海事大学

地址 201306 上海市浦东新区临港新城海
港大道1550号

(72)发明人 张家碧 杨智应

(74)专利代理机构 上海三和万国知识产权代理

事务所(普通合伙) 31230

代理人 陈伟勇

(51)Int.Cl.

G01S 19/42(2010.01)

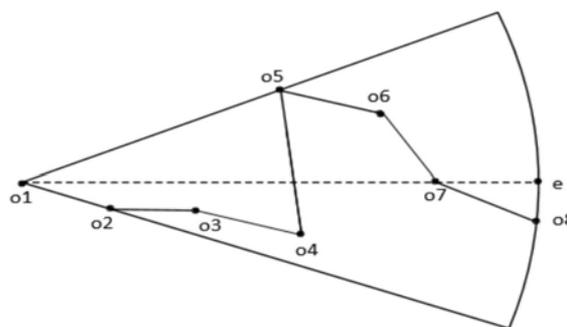
权利要求书1页 说明书5页 附图1页

(54)发明名称

一种船舶轨迹的实时化简方法

(57)摘要

船舶在航行时存在通信设备的干扰,信号衰减等因素,因此来自海上航行船舶的GPS定位数据可能存在信号失真等情况。船舶的航行状态较为稳定,航向是通过缓慢的累计发生改变的,因此其带宽度肥大,曲度平缓。结合以上实际,本文以实时化简为背景,以一种简单的(只要求移动对象的位置信息)新的实时轨迹化简方法为核心,建立一种船舶轨迹的实时化简机制。在坐标下对船舶的位置信息点进行计算,确定需要保留的点,减小了网络通信量和存储空间。



1. 一种船舶轨迹的实时化简方法, 船舶的位置点信息由GPS定位得到, 其特征在于包括如下步骤:

步骤1 打开GPS开始定位, 得到前两个位置点信息 o_1, o_2 ; 视 o_1 为不可化简的点, 发送其位置信息到数据库, 并以 o_1 为极点, 过 o_1, o_2 的直线为极轴建立极坐标系;

步骤2 根据GPS是否关闭判断定位是否结束, 若定位结束跳至第6步; 否则等待下一个位置点信息 o_3 的到来, 构造最小边界扇形, 确定顶角 θ 的大小, 计算 S_{sec} 和 S_{max} 的大小, S_{max} 为最大扇形面积; 如果 $S_{sec} > S_{max}$, 则跳至第4步, 否则继续等待新的位置点信息;

步骤3 根据GPS是否关闭判断定位是否结束, 若定位结束跳至第6步, 否则等待下一位置点 o_i 信息的到来, 其中, $i \geq 4$, 重新构造最小边界扇形, 计算扇形的实际面积 S_{sec} ; 接着判断顶角 θ 的大小是否改变, 若改变则重新计算 S_{max} , 当顶角 θ 不变时 S_{max} 不变; 如果 $S_{sec} > S_{max}$, 则跳至第4步, 否则继续等待新的位置点信息;

步骤4 当前条件已满足向服务器更新条件, 在当前构造的扇形中若某点 o_i 满足其前2个或超过2个的点位于扇形角平分线的一侧或角平分线上, 其中, $3 \leq i \leq n$, 而 o_i 位于角平分线的另一侧或角平分线上, 则将这些点发送到服务器端; 最近接受的点 o_k 为不可化简的点发送到服务器端, 跳至第5步;

步骤5 根据GPS是否关闭判断定位是否结束, 若定位结束跳至第6步, 否则以 o_k 为新的扇形顶点, 继续接收新的位置点信息 o_{k+1}, o_{k+2} ; 构造最小边界扇形, 确定顶角 θ 的大小, 计算 S_{max} 的大小; 跳至第3步;

步骤6 定位结束, 最后一个点是不可化简的点, 发送最后一个点的信息到数据库;

步骤1, 2, 3在极坐标系中以 o_1 为极点和扇形顶点构造扇形, 在顶角 θ 确定时, 结合给定阈值 st_error 能计算出 θ 和 st_error 下的最大扇形面积 S_{max} , 具体为: 以扇形的一条边和角平分线为另一条边的等腰三角形的底长为 d ; 在顶点为 o_1 , 顶角为 θ 的所有扇形中, 选取 $d = st_error$ 的扇形, 并根据扇形面积公式计算其面积为 S_{max} , 其中,

$$S_{max} = \frac{\theta \pi \left(\frac{st_error}{2 \sin \frac{\theta}{4}} \right)^2}{360},$$

扇形的实际面积 S_{sec} 与 S_{max} 的大小作为是否向服务器更新位置信息的判断依据; 实际面积 S_{sec} 大于 S_{max} , 则认为要更新移动对象位置信息。

2. 根据权利要求1所述的一种船舶轨迹的实时化简方法, 其特征是: 步骤4中在满足更新条件时, 判断已接收的点是否满足其前2个或超过2个的点位于扇形角平分线的一侧或角平分线上, 而该点本身位于角平分的另一侧或角平分线上; 若满足则将该点位置信息发送到服务器端。

一种船舶轨迹的实时化简方法

技术领域：

[0001] 本发明涉及船舶轨迹跟踪定位，具体涉及一种船舶轨迹的实时化简方法。

背景技术：

[0002] 随着社会进步和科技进步，物流产业也得到了飞速的发展。船舶是其中的一种重要的载体。在船舶的正常行驶过程中，其位置信息会不断的更替。大量的数据一方面造成了传输和处理的压力，增大了开销。另一方面可能造成数据的冗余。虽然目前有较多的移动对象轨迹化简方法，但一方面轨迹的实时化简算法较少，另一方面针对船舶航迹特点的也很少。本文在此背景下提出一种针对船舶的轨迹实时化简方法。

[0003] 移动对象轨迹化简问题是当下比较热点的问题之一，移动对象本质上是随时间而变化的几何实体。移动对象轨迹具有不确定性，这包括时态不确定性和区域不确定性。在处理这类问题时，我们可以为原始轨迹设置一个阈值 th ，这时我们可以得到一个围绕轨迹的三维圆柱形缓冲区域。假设移动对象的实际位置偏离了预计位置的距离达到或者超过了 th ，移动对象将更新数据库服务器。

[0004] 船舶在航行时存在通信设备的干扰，信号衰减等因素，因此来自海上航行船舶的GPS定位数据可能存在信号失真等情况，因此可对船舶轨迹进行压缩化简。就船舶而言，其航行状态较为稳定，航向是通过缓慢的累计发生改变的，因此其带宽度肥大，曲度平缓。结合以上实际，本文以实时化简为背景，以一种简单的（只要求移动对象的位置信息）新的实时轨迹化简方法为核心，建立一种船舶轨迹的实时化简方法。在坐标下对船舶的位置信息进行计算，确定需要保留的点，减小了网络通信量和存储空间。

发明内容：

[0005] 船舶轨迹化简可认为是对散列的轨迹点进行线性近似。在本方法中选取扇形来匹配扇形的运动轨迹，并考虑到实际场景中GPS本身存在的定位误差。设计了一种针对船舶轨迹的实时化简方法。具体如下：

[0006] 一种船舶轨迹的实时化简方法，船舶的位置点信息由GPS定位得到，其特征在于包括如下步骤：

[0007] 步骤1、打开GPS开始定位，得到前两个位置点信息 o_1, o_2 ；视 o_1 为不可化简的点，发送其位置信息到数据库，并以 o_1 为极点，过 o_1, o_2 的直线为极轴建立极坐标系；

[0008] 步骤2、根据GPS是否关闭判断定位是否结束，若定位结束跳至第6步；否则等待下一个位置点信息 o_3 的到来，构造最小边界扇形，确定顶角 θ 的大小，计算 S_{sec} 和 S_{max} 的大小， S_{max} 为最大扇形面积；如果 $S_{sec} > S_{max}$ ，则跳至第4步，否则继续等待新的位置点信息；

[0009] 步骤3、根据GPS是否关闭判断定位是否结束，若定位结束跳至第6步，否则等待下一位置点 o_i ($i \geq 4$) 信息的到来，重新构造最小边界扇形，计算扇形的实际面积 S_{sec} ；接着判断顶角 θ 的大小是否改变，若改变则重新计算 S_{max} ，当顶角 θ 不变时 S_{max} 不变；如果 $S_{sec} > S_{max}$ ，则跳至第4步，否则继续等待新的位置点信息；其中，结合给定阈值 st_error 能计算出 θ 和

st_error下的最大扇形面积 S_{\max} ,具体为:以扇形的一条边和角平分线为另一条边的等腰三角形的底长为d;在顶点为 o_1 ,顶角为 θ 的所有扇形中,选取 $d=st_error$ 的扇形,并根据扇形

$$S_{\max} \left(S_{\max} = \frac{\theta \pi \left(\frac{st_error}{2 \sin \frac{\theta}{4}} \right)^2}{360} \right);$$

[0010] 步骤4、当前条件已满足向服务器更新条件,在当前构造的扇形中若某点 o_i ($3 \leq i \leq n$) 满足其前2个或超过2个的点位于扇形角平分线的一侧或角平分线上,而 o_i 位于角平分线的另一侧或角平分线上,则将这些点发送到服务器端;最近接受的点 o_k 为不可化简的点发送到服务器端,跳至第5步;

[0011] 步骤5、根据GPS是否关闭判断定位是否结束,若定位结束跳至第6步,否则以 o_k 为新的扇形顶点,继续接收新的位置点信息 o_{k+1}, o_{k+2} ;构造最小边界扇形,确定顶角 θ 的大小,计算 S_{\max} 的大小;跳至第3步;

[0012] 步骤6、定位结束,最后一个点是不可化简的点,发送最后一个点的信息到数据库。

[0013] 优选的,所述步骤1,2,3在极坐标系中以 o_1 为极点和扇形顶点构造扇形,在顶角 θ 确定时,结合给定阈值st_error能计算出 θ 和st_error下的最大扇形面积 S_{\max} ,之后以扇形的实际面积 S_{sec} 与 S_{\max} 的大小作为是否向服务器更新位置信息的判断依据;实际面积 S_{sec} 大于 S_{\max} ,则认为要更新移动对象位置信息。

[0014] 优选的,所述步骤4中在满足更新条件时,判断已接收的点是否满足其前2个或超过2个的点位于扇形角平分线的一侧或角平分线上,而改点本身位于角平分的另一侧或角平分线上;若满足则将该点位置信息发送到服务器端。

[0015] 有益效果:

[0016] 1.基于目前的轨迹化简算法的现状,以实时化简和船舶轨迹为背景,建立了一套船舶轨迹的实时化简机制,这套机制有较强的运用背景。

[0017] 2.本方法对硬件设备要求低,仅要求船舶的基本位置信息,对船舶速度、方向等信息并无要求。是一种简单高效的轨迹化简方法。

[0018] 3.本方法的计算量较小,在整体开销无明显增加的前提下,保留了更多、更关键的轨迹特征,提高了化简轨迹的还原度。

附图说明

[0019] 图1为本发明一种船舶轨迹的实时化简方法的流程示意图。

具体实施方式

[0020] 在本化简方法中,为了减小计算规模和总体开销。将GPS定位得到的船舶的位置信息(由经纬度表示)转换成极坐标系下的各个点,并以初始定位得到的点作为极坐标系的极点。该化简方法要求在化简开始之前手动输入误差阈值st_error,此处需要说明的是对于同一串船舶的轨迹信息,不同的st_error有不同的结果。船舶在开始定位得到自己的前两个初始位置 o_1, o_2 时,以 o_1 为极点建立极坐标系,并计算 o_2 与 o_1 的关系将 o_2 加入极坐标系。当接收到第三个位置点信息 o_3 时,以 o_1 为扇形顶点,过 o_1, o_2 和 o_1, o_3 的两条线段为边,并以两者

中较长的一个作为半径构造最小边界扇形。此时的实际扇形面积记为 S_{sec} ，扇形的顶角记为 θ 。假设以扇形的一条边和角平分线为另一条边的等腰三角形的底长为 d 。在顶点为 o_1 ，顶角为 θ 的所有扇形中，选取 $d = st_error$ 的扇形，并根据扇形面积公式计算其面积为

$$s_{max} \left(s_{max} = \frac{\theta \pi \left(\frac{st_error}{2 \sin \frac{\theta}{4}} \right)^2}{360} \right)。$$

判断 S_{sec} 和 S_{max} 的大小，若 $S_{sec} > S_{max}$ ，则向服务器更新该移动对

象的位置信息，否则继续等待接受下一位置点信息。从接收到的第四个位置点 o_4 开始，首先重新构造最小边界扇形，判断扇形的顶角是否还是 θ 。在 θ 不变的情况下，若 $S_{sec} > S_{max}$ ，则向服务器更新该移动对象的位置信息，否则继续等待接受下一位置点信息。假设 θ 的大小发生变化，则重新计算 S_{max} ，再与 S_{sec} 比较大小。

[0021] 本化简方法的具体实施步骤可表示如下：

[0022] 输入：船舶实时定位得到的各位置点信息；

[0023] 输出：筛选后将要发送到数据库的位置点信息；

[0024] 预设值：距离误差 st_error ；

[0025] 步骤1、打开GPS开始定位，得到前两个位置点信息 o_1, o_2 。视 o_1 为不可化简的点，发送其位置信息到数据库。并以 o_1 为极点，过 o_1, o_2 的直线为极轴建立极坐标系；

[0026] 步骤2、根据GPS是否关闭判断定位是否结束，若定位结束跳至第6步。否则等待下一个位置点信息 o_3 的到来，构造最小边界扇形，确定顶角 θ 的大小，计算 S_{sec} 和 S_{max} 的大小。如果 $S_{sec} > S_{max}$ ，则跳至第4步。否则继续等待新的位置点信息；

[0027] 步骤3、根据GPS是否关闭判断定位是否结束，若定位结束跳至第6步。否则等待下一位置点 $o_i (i \geq 4)$ 信息的到来，重新构造最小边界扇形，计算扇形的实际面积 S_{sec} ；接着判断顶角 θ 的大小是否改变，若改变则重新计算 S_{max} ，当顶角 θ 不变时 S_{max} 不变。如果 $S_{sec} > S_{max}$ ，则跳至第4步。否则继续等待新的位置点信息；

[0028] 步骤4、当前条件已满足向服务器更新条件，在当前构造的扇形中若某点 $o_i (3 \leq i \leq n)$ 满足其前2个或2个以上的点位于扇形角平分线的一侧或角平分线上，而 o_i 位于角平分的另一侧或角平分线上，则将这些点也发送到服务器端；最近接受的点 o_k 为不可化简的点发送到服务器端，调至第5步。

[0029] 步骤5、根据GPS是否关闭判断定位是否结束，若定位结束跳至第6步。否则以 o_k 为新的扇形顶点，继续接收新的位置点信息 o_{k+1}, o_{k+2} 。构造最小边界扇形，确定顶角 θ 的大小，计算 s_{max} 的大小。跳至第3步；

[0030] 步骤6、定位结束，最后一个点是不可化简的点，发送最后一个点的信息到数据库。

[0031] 本方法的实现步骤可由以下伪代码表示，其中 st_error 表示用户输入的误差阈值， $Traj$ 表示原始数据集合；函数 $CalcMBS(buf, point)$ 用于计算当前的最小边界扇形；其中 buf 表示存储历史轨迹点的集合， $point$ 表示最新得到的位置信息。函数 $area(newMBS)$ 用于计算当前构造的扇形面积。 $newMBS$ 表示当前扇形信息，函数 $CalcCen(newMBS)$ 用于计算当前扇形的圆心角大小； $Calcarea(\theta, st_error)$ 用于计算在给定阈值 st_error 和圆心角 θ 下的理论面积；函数 $keypoint(point)$ 用于判断满足更新条件时，当前点是否要向服务器端更新。函数 $Boundinfof(newMBS)$ 用于获取当前扇形的边界信息包括扇形中心点坐标，以及距离该坐标点的最大值和到中心点与X轴所成最大角、最小角四个数据。

```

1.   function calcMinBoundingSector (st_error, Traj)
2.   {
3.       newMBS=null;//保存接收到的轨迹点位置集合
4.       key=null;//关键点集合初始为空
5.       WHILE(Traj.hasNext()) DO
6.           point=Traj.next();
7.           IF(buf.Count>=2) THEN//两个点不能构造扇形
[0032]           newMBS=CalcMBS(buf, point);//计算加入新位置信息后的最小
           边界扇形
8.            $\theta$ =CalcCen(newMBS);//计算圆心角的大小
9.           IF(area(newMBS)<=Calcarea( $\theta$ , st_error)) THEN
10.              //向服务器端更新扇形信息
11.              output(key);//输出关键点
12.              output(Boundinfor(newMBS));
13.              END IF
14.          END IF
15.      END WHILE
[0033] 16.      END WHILE
17.  }

```

[0034] 如图1所示,以初始的位置点,作为极坐标系原点,建立极坐标系。坐标系中每个点代表一船舶一次定位得到的位置点。

[0035] 开始定位,并接收前2点 O_1, O_2 的位置点信息,并以 O_1 为极点,过 O_1, O_2 的直线为极轴建立极坐标。向服务器更新 O_1 的位置点信息。接收 O_3 的信息后构造最小边界扇形,计算顶角 θ 和理论最大面积 S_{max} 以及扇形面积 S_{sec} ,此时 $S_{sec} < S_{max}$ 不满足更新条件。等待接收 O_4 的位置点信息,构造最小边界扇形顶角 θ 大小不变,计算得 $S_{sec} < S_{max}$ 不满足更新条件。等待接收 O_5 ,构造最小边界扇形后发现 θ 大小改变,重新计算在新的顶角 θ 下的最大面积 S_{max} ,比较得 $S_{sec} < S_{max}$,继续接收 O_6 位置点信息。接收 O_6 后构造的扇形顶角 θ 不变, $S_{sec} < S_{max}$,不更新位置点信息。接收 O_7 ,新构造的扇形顶角仍不变,且 $S_{sec} < S_{max}$,不更新移动对象位置信息。接收 O_8 后构造的扇形顶角 θ 不变,但 $S_{sec} > S_{max}$ 。满足更新条件,判断发现 O_5, O_7, O_8 满足步骤4中描述的条件。因此向服务器发送点 O_5, O_7, O_8 的相关信息;此时的定位尚未结束。以 O_8 为定位起点和扇

形的顶点,等待 O_9, O_{10} 的相关信息的到来,重新构造最小边界扇形,计算顶角 θ 的大小。计算 S_{\max} 和 S_{\sec} 的大小并进行比较。重复上述步骤至定位结束。

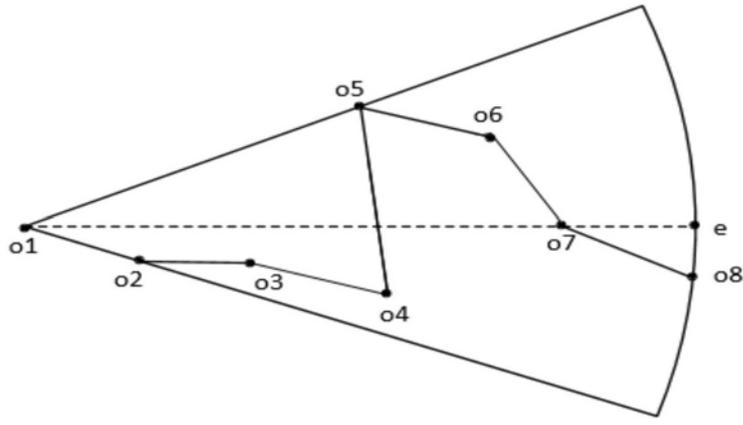


图1