

## 基于动态路网的分布式邻近目标查询算法

叶晨<sup>1,2</sup>, 杨振宇<sup>1,2</sup>, 喻剑<sup>2</sup>, 龙其<sup>1,2</sup>

(1. 同济大学 计算机科学与技术系, 上海 201804; 2. 同济大学 嵌入式系统与服务计算教育部重点实验室, 上海 201804)

**摘要:** 提出了一种基于实时路况信息的分布式邻近目标查询算法, 采用基于 Voronoi 图的划分将地理信息存储在离它最近路口的智能摄像头上, 实时路况信息由智能摄像头采集, 通过对路口的畅通程度进行建模, 估算出路口间通行所需要的时间。当有车辆查询邻近目标时, 网络中的智能摄像头根据所在路口的畅通程度和到邻近路口的距离, 在分布式查询过程中加入延时转发机制, 广播目标路径询问的数据分组, 使数据分组的发送能模拟当前的路况进行传输, 从而获得到达邻近目标的路径。基于真实数据的实验结果表明算法是有效的, 处理大量并发查询时的性能优于现有方法。

**关键词:** 动态路网; 最邻近查询;  $k$  邻近查询; 分布式查询; 延迟路由

中图分类号: TP391.9

文献标识码: A

文章编号: 1000-436X(2014)12-0116-08

## Distributed nearneighbor search algorithm based on real-time traffic information in dynamic road network

YE Chen<sup>1,2</sup>, YANG Zhen-yu<sup>1,2</sup>, YU Jian<sup>2</sup>, LONG Qi<sup>1,2</sup>

(1. Computer Science and Technology Department, Tongji University, Shanghai 201804, China;

2. Key Laboratory of Embedded System and Services Computing, Ministry of Education, Tongji University, Shanghai 201804, China)

**Abstract:** A novel distributed near neighbor search algorithm that makes use of real-time traffic information is presented. The geographic information are stored in the nearest smart camera using Voronoi partition, and cameras are located in the intersection. The intersection unimpeded degree is modeled and the time which vehicle travel between adjacent intersections is estimated. When a vehicle search for some near neighbors, smart cameras set a delay to broadcast the near neighbor search packet based on the traffic parameters collected by smart camera networks. In this way, the near neighbor search packet can be transmitted according to current road conditions. Thus get the path to near targets quickly and effectively. Extensive experiments are conducted on real data sets, and the results show that proposed algorithm is efficient and scalable to large number of concurrent query, significantly outperforming state-of-the-art methods.

**Key words:** dynamic road network; nearest neighbor search;  $k$ -nearest neighbor search; distributed search; delay routing

### 1 引言

随着移动互联网的发展和定位技术的日趋成熟, 基于位置的服务(LBS, location based service)<sup>[1,2]</sup>已经深入到各行各业的应用中, 逐步走入人们的日常生活。当开车出行时, 人们经常想要去一些离自己最近的地方, 如最近的加油站、最近的餐馆、最近的银行等。这些查询称为邻近目标查询, 当只查询一个结果时, 称为最邻近查询(NNS, nearest

neighbor search), 查询多个可能的结果时, 称为  $k$  邻近查询( $k$ -NN,  $k$ -nearest neighbor)。已有的研究工作<sup>[3,4]</sup>将 NNS 和  $k$ -NN 算法应用于路网中的邻近兴趣点(POI, point-of-interest)查询, 但它们普遍采用集中式存储和索引的方法。本文创新性的提出一种高效的分布式邻近目标查询算法, 基于 Voronoi 图的划分将地理信息存储在离它最近路口的智能摄像头上, 一方面简化了路网中地理信息索引的复杂性, 另一方面对智能摄像头采集的实时路况信息

收稿日期: 2014-03-26; 修回日期: 2014-08-07

基金项目: 国家国际科技合作专项基金资助项目(2012DFG11580)

Foundation Item: The International S&T Cooperation Program of China (2012DFG11580)

进行路口交通流建模，并引入计算机网络中的路由思想获取到达邻近目标的动态最短路径。

现有的分布式空间索引方法，在划分方式上主要依赖于将二维或多维的空间数据按照树结构进行存储，能用于属性查询、定位查询等一些较简单的空间查询，但很难应用于最邻近查询等复杂的查询分析。这方面的研究主要基于 R-tree<sup>[5]</sup>或对 R-tree 的改进<sup>[6]</sup>，如 R\*-tree、VB-tree、MR-tree 等<sup>[7]</sup>。同时，这些基于欧氏距离的查询方法没有考虑道路网络距离的约束条件。

动态路网是一个时变网络，动态路网中邻近目标的查询问题包含到达目标的最短路径问题，然而，基于时变网络的最短路径问题并没有十分精确的求解方案。Kiseok Sung<sup>[8]</sup>等人提出了一种路段车流随时间变化的路网模型(flow speed model)，并指出路段行驶时间随时间变化。基于此模型，Evangelos Kanoulas<sup>[9]</sup>等人假设在某一段时间内的车流速度是一给定值或已知的时变函数，并将路段车流速度按照不同时间进行了分类。然而现实生活中的动态路网远比这复杂，因此上述模型难以应用于工程实践。

本文通过将复杂的大规模交通地理信息分布式地存储在路口智能摄像头上，一些原来复杂的求解问题得到显著的简化。对于最邻近目标查询，首先定位查询的起始位置，找到与之相关的最近的智能摄像头节点，该节点采用基于延时路由的分布式邻近查询算法进行相应目标的查询，并返回查询结果。而  $k$  邻近查询是最邻近查询的扩展，在使用查询算法进行查询的过程要保留多个结果。另外在某些场合下，查询是以一定的范围进行限制的，比如，某个时候人们想查询离自己 5 km 的范围内有哪些景点可以游玩。针对上述几种邻近目标的查询，在下文中将给出相应的分布式查询算法，并通过实验验证了算法的准确高效。

## 2 分布式交通地理信息存储

### 2.1 智能摄像头网络模型

本文的智能摄像头网络是智能摄像头、通信网络与交通路网的有机结合。以设立在各个路口的智能摄像头为单位，相邻路口的摄像头之间都有网络连接，能够实现相互之间的通信。如图 1 所示，这种形式的智能摄像头网络一定程度上模拟了交通路网的拓扑结构。

用图  $G=(V,E)$  来描述摄像头及其之间的关系， $V$  是各路口的智能摄像头节点， $E$  是摄像头之间的网络连接关系， $E=\{(u,v)|\text{摄像头 } u、v \text{ 间存在直接的网络连接关系}\}$ ，在本文的场景中，对于路网中任意相邻路口的摄像头  $u$  和  $v$ ， $(u,v)\in E$ 。

每个路口的智能摄像头实时监控路口的交通状况，并具有一定的存储空间和计算能力。对于大规模的城市交通地理信息，采用就地存储的分布式存储方式，把交通路网中的各种地理信息，如加油站、宾馆、饭馆、银行等，存储在离它最近的智能摄像头，如图 1 所示。这样，原来大规模的城市交通地理信息被分散存储到城市中成千上万的智能摄像头上，对于每一个智能摄像头而言，只需存储与之相关的少量交通地理信息，相较于大规模交通地理信息数据而言，不需要采用复杂的数据结构，一般的关系型数据库就足以胜任了。

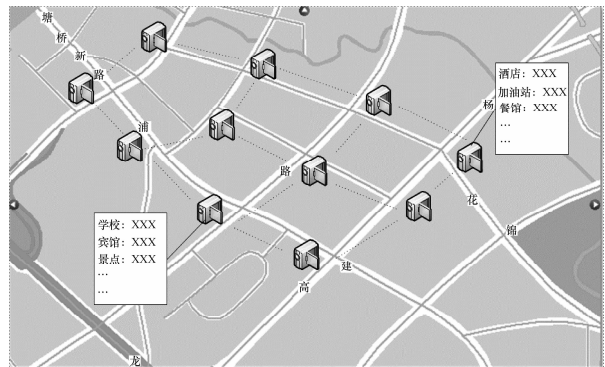


图 1 路口智能摄像头与摄像头网络

### 2.2 基于 Voronoi 图的地理信息划分

分布式的存储大大降低了交通地理信息的组织难度，对于存储在智能摄像头中的每一个地理场所的信息，它的记录形式如下。

(场所 ID, 类型, 名称, 位置, 附加信息)

其中，类型表示一个场所的类型，如酒店、加油站、银行、学校等；名称表示该场所的具体名称，如同济大学；位置表示该场所的具体坐标信息；附加信息则可以记录一些收费、评价等用于其他应用的扩展内容。

针对如何将上述信息存储在离它最近的智能摄像头上，并且消除重复的记录，本文采用基于 Voronoi 图<sup>[8]</sup>的划分来组织这些地理信息。从计算几何的角度来看，Voronoi 图是对平面上的离散点而言的，它把平面分为若干个区域，每个区域包含一个点，该点所在的区域是到该点最近点的集合。位

置坐标点是交通地理信息中最基本的空间数据, 本文从空间中的距离定义出发, 引进空间 Voronoi 图的划分, 把构成 Voronoi 图的基本点称为目标点。

**定义 1** 设  $p_1, p_2$  是二维空间中的 2 点, 其坐标分别为  $(x_1, y_1), (x_2, y_2)$ , 那么  $p_1, p_2$  之间的距离定义为

$$Distance(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

**定义 2** 设  $P_1, P_2, \dots, P_n$  构成目标点集合  $M$ , 那么, 构成  $P_i$  的 Voronoi 图划分的区域多边形定义为

$$Voronoi(P_i) = \{p_k \mid Distance(p_k, P_i) < Distance(p_k, P_j), P_j \in M, P_j \neq P_i\} \quad (2)$$

求目标点集  $M$  的 Voronoi 图划分, 就是求每个目标点的区域多边形。Voronoi 图的构造算法主要包括矢量法与栅格法<sup>[10]</sup>。矢量法根据图的性质, 通过计算顶点坐标获得 Voronoi 图的边界, 获取临近生长源以及 Voronoi 边界轨迹与交点。栅格法通过生长源膨胀, 形成势力范围, 进而获得 Voronoi 图。包括并行构造算法<sup>[11]</sup>在内的 Voronoi 图生成算法已经有了深入的研究, 本文不再赘述。

在路网环境下, 把各个交通路口看成离散的点, 从这些点出发, 生成整个交通路网的 Voronoi 图, 这样整个交通区域将由这些路口点划分成若干个区域, 每个区域中都包含有一个路口点, 并且该路口点是区域中最近的路口, 如图 2 所示。

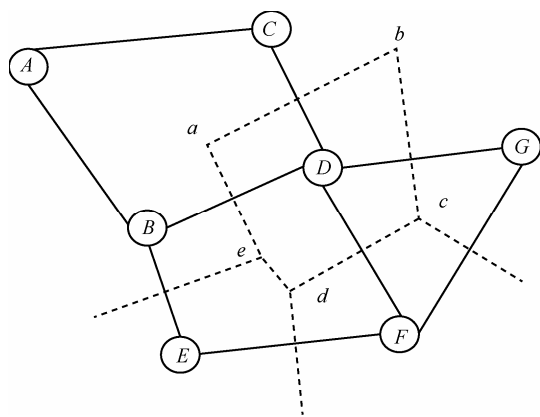


图 2 基于 Voronoi 图的划分

图 2 中, 大写字母代表智能摄像头, 它的坐标根据城市道路规划的规划以  $(x, y)$  来唯一标记。所有的坐标形成一个二维平面点集, 利用栅格化的 Voronoi 图生成算法来求这个平面点集的 Voronoi 图, 然后根据 Voronoi 的划分计算每个点的区域多边形。如图 2 所示, 点  $D$  的区域多边形是多边形

$abcde$ 。计算出每个点的区域多边形后, 将其存储在对应的智能摄像头上, 用于进行交通地理信息存储和更新时的判断依据。

交通地理信息要进行存储时, 向周围摄像头发送场所的位置信息, 收到该信息的智能摄像头判断其是否位于它的区域多边形内。如果位于相应的区域多边形内, 智能摄像头将该场所完整的信息存储在本地。

### 3 基于实时路况的延时路由算法

#### 3.1 基于实时数据的路口交通流建模

布置在各个路口的智能摄像头, 实时监控区域内行驶的车辆, 当摄像头捕获到一辆车时搜集车辆的颜色、车牌等信息, 并将这些特征信息通过智能摄像头网络发送给与之相邻的下一个路口的智能摄像头, 当下一个路口的智能摄像头也发现了同样的车辆时, 可以通过 2 次时间的差异来估计该车通过这段路口所用的时间。当获取大量的通行时间数据时, 可以推算和预测当前两路口间的平均通行时间。

文献[12]提出了一种改进的迭代学习方法, 利用路口摄像头测得车辆在相邻路口间出现的时间, 来估算当前两路口间的平均通行时间。假设上一次统计的平均时间为  $T_{old}$ ; 当前一分钟内统计到的车辆的平均通行时间为  $T_{new}$ ; 则新计算出的平均时间如式(3)所示。

$$T_{cur} = \beta T_{old} + (1 - \beta) T_{new} \quad (3)$$

其中,  $\beta$  是  $(0,1)$  内的一个常数。

迭代流程如下。

- ①初始第 0 分钟, 设置  $T_{old} = 0$ ;
- ②通过智能摄像头获取数据  $T_{new}$ ;
- ③根据式(3)计算  $T_{cur}$ ;
- ④更新  $T_{old} \leftarrow T_{cur}$ ;
- ⑤进入下一分钟, 重复②~④。

如此不停地迭代, 实时估计车辆的通行时间。

#### 3.2 道路通行时间转化为网络传播延时

基于实时数据的交通流模型计算出路口间的平均通行时间后, 为了满足分布式邻近目标查询的需求, 本文将平均通行时间转化成传播延时, 即数据分组在网络中传输的时间。例如, 当前路况下, 车辆从  $A$  到达  $B$  需要  $T$  min, 那么按一定的比例把它转化成数据分组在这 2 个路口的智能摄像头间传递所需的时间为  $t$  ms。

为了保证车辆在道路上行驶的最短路径可以由数据分组传播的最短路径来表示，本文采用了线性函数，将车辆行驶时间转化成传播延时  $t$ ，其中， $k$  为转换比例系数。

$$t = kT \tag{4}$$

这里采用线性函数的原因如下。

1) 以等比例来转化延时，相当于将路网图转化为权值更小的网络同构图，在这样的网络同构图中路由得到的最短路径即为原路网中的最短路径缩小相应比例。

2) 数据分组在相邻摄像头之间传递的速度很快，小于 1 ms。把网络传播延时设置在 10 ms 到 1 000 ms 之间，尽可能地避免了数据处理与传递对于结果的干扰。

通过这种方式转换延时，比例系数  $k$  的选择将对结果有较大的影响，实验一将着重讨论  $k$  值的选取。

### 3.3 传播延时表

把传播延时引入到分布式智能摄像头网络后，网络中的智能摄像头就像一个个的路由器，实时维护着到相邻路口的传播延时，如图 3 所示。这一数据记录在每个智能摄像头的传播延时表中，如表 1 所示。

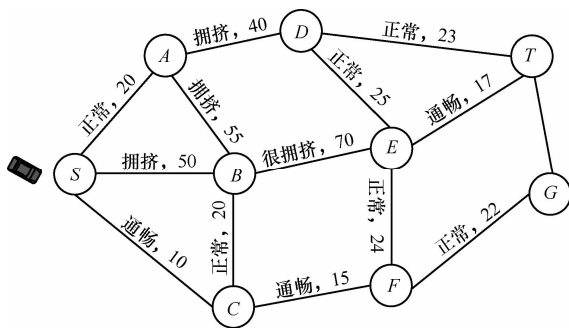


图 3 包含传播延时的智能摄像头网络示例

表 1 节点 S 的传播延时

相邻摄像头 ID	传播延时/ ms
A	20
B	50
C	10

图 3 中，车辆在 S 路口附近需要去往目标所在路口 T。此时，路口间的拥挤程度和通行时间，已通过一定的比例被转化为数据分组的传播延时。这样，最先到达 T 的消息就是沿着 S-C-F-E-T 的路径传输的。这样的路径正确反应出了当前路况下，车

辆行驶的最优路径。在实际应用场景中，每个智能摄像头的邻居一般 4 到 6 个，因此传播延时表可以采用简单的线性表，支持快速查询与更新。

## 4 基于延时路由的分布式邻近查询算法

### 4.1 分布式最邻近查询

人们在出行过程中，经常有一些突发的查询需求。最简单地，车辆在行驶过程中提示燃油不多了，这时人们通常会想要到离自己最近的加油站去加油。常规的方法要在大量的交通地理数据中进行搜索，或者是用复杂的数据结构对地理信息进行组织，并在此基础上研究高效的方法。本文提出的方法中，大量的地理信息被分布式地存储于智能摄像头，各个路口的智能摄像头一起协同工作来获取查询结果。在一次查询请求的过程中，每个摄像头依据传播延时表转发数据分组，单个摄像头节点的计算量非常小。数据分组利用摄像头之间相互连接的网络，同一时间并行地尝试各种走法，找到的最近目标将会是当前路况下最近的有效目标，同时返回到达该目标的最优路径。

#### 4.1.1 传播消息格式与查询消息表

分布式最邻近查询算法需要在智能摄像头网络中传播查询消息和与之对应的回应消息。

查询消息的格式定义为  $\langle \text{TYPE}, \text{GUID}, \text{LASTNODE}, \text{TARGET} \rangle$ ，其中，TYPE 是消息的类型，如最邻近目标查询消息，最邻近回应消息等；GUID 唯一标记了这一次邻近查询的相关消息数据分组，是随机产生的一个全局唯一的量；LASTNODE 是指消息的发送方；TARGET 是指要查询的最邻近目标是什么，如加油站、酒店等，用来判断是否已经找到目标。

回应消息格式为  $\langle \text{TYPE}, \text{GUID}, \text{COORDINATE}, \text{路径节点数 } n, \text{节点 } 1、2、\dots、n \rangle$ 。回应消息最终返回给提出查询的节点，其中，TYPE 和 GUID 同查询消息，COORDINATE 是查询到的最近目标的地理位置，之后是到达该目标的最短路径需要经过的路口节点总数以及依次经过的节点 ID。

为了避免消息的重复发送并简化查询过程，每个智能摄像头维护着一个查询消息表。如表 2 所示。该表记录了  $\langle \text{GUID}, \text{上一跳节点}, \text{消息到达时间} \rangle$ ，其中 GUID 是主键，唯一确定了每次邻近查询的消息。

GUID	上一跳节点	消息到达时间
123456	B	12:57:23
124245	D	12:56:49
...	...	...

### 4.1.2 分布式最邻近查询过程

当智能摄像头接收到附近车辆发出的最邻近查询请求后，处理流程如图 4 和图 5 所示。

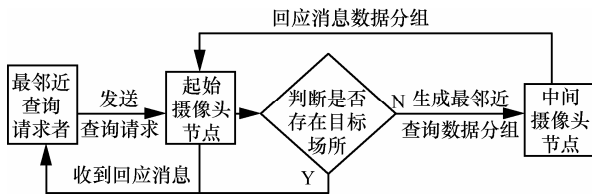


图 4 分布式最邻近查询起始节点处理流程

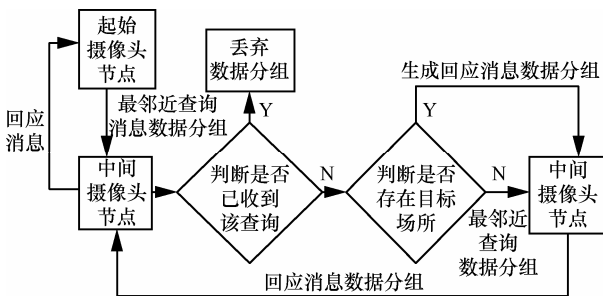


图 5 分布式最邻近查询中间节点处理流程

①接收查询请求的节点，根据查询目标的类型，首先在自己的交通地理信息数据库中查询是否存在相应类型的记录，若存在，直接返回找到记录的位置和名称发送给查询申请者，完成本次查询。若没有相关的记录，则创建查询消息，并遍历传播延时表，将查询消息延时发送给相应的相邻摄像头。如图 3 所示，节点 S 收到了查询请求，若本地不存在相应类型的记录，则将查询消息数据分组延迟 12 ms 发送给节点 C，再延迟 3 ms 发送给节点 A，再延迟 30 ms 发送给节点 B。

②中间节点收到最邻近查询消息数据分组后，先检查自己的消息查询表中是否已有对应的查询消息：若有，则直接丢弃该消息，否则在自己的交通地理信息数据库中查询是否存在要查询的场所类型的记录，如果存在，就创建最邻近回应消息数据分组，将查询到的场所名称和位置记录在数据分组中，沿着上一跳路径传送回起始节点。如果不存在，则将最邻近查询消息数据分组按传播延时表中的延时发送给与之相邻的摄像头。

③当创建了最邻近回应数据分组后，数据分组

要沿着走过的路径传递回去，中间智能摄像头节点收到最邻近回应消息数据分组后，首先判断是否已经收到过相同 GUID 的回应消息数据分组；若收到过，则直接丢弃该数据分组，因为最邻近的目标一定会最先返回；若没有收到过，则根据数据分组的 GUID 从查询消息表中获取上一跳地址，并把自己加入最短路径序列中，同时标记该 GUID 的回应消息数据分组已经收到过，然后将最邻近回应消息数据分组发往上一跳智能摄像头节点。

④当起始节点收到最邻近回应消息后，提取查询到的最邻近目标的名称，位置以及前往该目标的最短路径信息，发送给查询申请者。至此，基于分布式的最邻近查询已经基本完成。

### 4.2 分布式 k 邻近查询

在临近目标查询中，有时希望返回多个查询结果，这样能提供一定的选择空间。将分布式最邻近查询算法修改一下，就可以得到分布式 k 邻近查询算法。与最邻近查询相比，k 邻近只是要求保留多个查询结果。

把分布式最邻近查询的过程修改成能进行分布式 k 邻近查询，需要修改的地方主要有 2 处。

首先，在最邻近的查询过程中，每个节点在收到最邻近查询消息数据分组后，如果在自己的交通地理信息数据库中找到了要查询的场所，就不会继续转发该查询的消息数据分组了。由于继续转发得到的查询结果一定不会比现在的结果更好，因此可以停止继续转发。而 k 邻近的查询中，则仍然需要将该查询消息数据分组转发出去，这样才能保证不会漏掉比较好的结果。

其次，当节点收到最邻近回应消息数据分组时，对于相同 GUID 的回应消息数据分组，只保留一个进行返回，而且是只返回最先到到达的回应消息。而对于分布式 k 邻近查询，由于要返回 k 个结果，因此最先到到达该节点的 k 个回应消息数据分组，都有可能成为最终的结果。于是每个节点对每个 GUID 的查询，要设置一个回应消息计数器，当收到该 GUID 的回应消息后，如果计数器小于 k 则返回该回应消息，并把计数器增加 1，如图 6 所示。

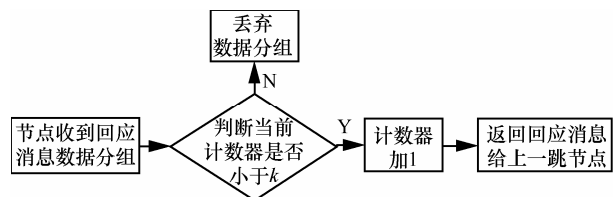


图 6 分布式 k 邻近查询中间节点回应消息计数

修改以上 2 点不同之处后, 原来分布式最邻近查询的流程也适用于分布式  $k$  邻近查询了。这时, 起始节点可能会收到较多的回应消息, 保留最先收到的  $k$  个结果, 发送给  $k$  邻近查询的请求者即可。分布式  $k$  邻近查询算法的伪代码见算法 1。

#### 算法 1 分布式 $k$ 邻近查询算法

输入: 接收到的消息  $M$ , 本地消息查询表  $List$ , 邻近查询数  $k$

```
function Distributed_kNN( $M, List, k$ )
if  $M.Type = REQUEST$  then //收到查询消息
    if  $List.Find(M.Guid) < 0$  then //判断是否已收到过该查询
         $List.Add(M)$  //消息查询表中追加一条记录
         $targets \leftarrow Map.Find(M.Target)$  //在本地地理信息数据库中查找目标场所
         $count \leftarrow targets.Count$ 
        if  $count < k$  then
            DelayedForward( $Neighbours, M$ ) //延时转发给相邻节点
        else
             $count \leftarrow k$ 
        end if
        for  $i = 0 \rightarrow count-1$  do
             $newM \leftarrow (RESPONSE, m.Guid, m.lastNode, targets[i].Coordinate, Nodes.Add(myID))$  //构造回应消息
            SendMessage( $M.lastNode, newM$ ) //向上一跳节点发送回应消息
        end for
    end if
else //  $M.Type = RESPONSE$  收到回应消息
    if ( $index \leftarrow List.Find(M.Guid) \geq 0$ ) then
        if  $List[index].Count < k$  then
             $List[index].Count ++$  //计数器加 1
             $M.Nodes.Add(myID)$  //将本节点加入到所需要经过的路口节点数组中
            SendMessage( $List[index].lastNode, M$ ) //向上一跳节点发送回应消息
        end if
    end if
end if
end function
```

### 4.3 区域限制下的分布式邻近查询

进行邻近目标查询时, 有时希望把查询结果的位置限定在一定的区域范围内, 比如查询离自己 5 km 范围内有哪些景点可以游玩。这种情形下的查询不是以最短距离来限制查询结果, 而是根据一定的区域限制, 如圆或矩形等区域把查询结果限制起来。

相对于分布式最邻近查询和分布式  $k$  邻近查询, 区域限制的分布式邻近查询不需要使用基于延时路由的分布式查询算法, 只要在每个节点进行限制区域检测。算法的大致流程如下。

①起始节点接收到查询请求后, 根据请求者位置和要求生成区域限制查询消息数据分组, 数据分组类似于最邻近消息查询数据分组, 与之相比多了限制区域, 用于各个节点进行判断的依据。然后起始节点将区域限制查询消息数据分组直接传递给与之相邻的节点。

②中间节点收到区域限制查询消息数据分组后, 如果未收到该数据分组对应 GUID 的查询消息, 根据数据分组中的区域信息判断其是否位于限制区域中, 如果不在, 则丢弃数据分组; 否则, 首先, 将数据分组继续转发, 然后查询本地地理信息数据库, 找到相应的查询目标后生成回应数据分组返回上一跳节点。

③节点收到回应消息后都要返回给上一跳节点, 起始节点收到所有的回应消息, 将其整理之后返回给邻近目标查询者。

## 5 实验与分析

本实验采用上海市的路网结构数据进行测试。如图 7 所示, 路网中包含约 4 500 个路口节点, 并用历史的路况数据生成不同时间段的路口间车辆通行时间。

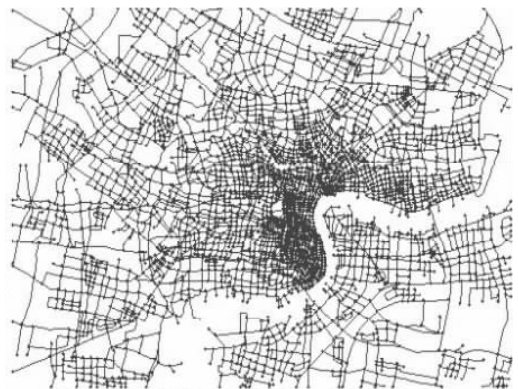


图 7 2009 年上海市路网结构

### 5.1 实验 1

实验 1 主要是验证分布式延时路由算法的正确性，并测试了延时转换比例  $k$  对邻近查询结果的影响。

根据式(4)将车辆在路网中行驶的时间按比例转化为数据分组在网络中传播的延时。第一组实验中，传播延时 10 ms 到 1 000 ms，分别对应了行驶时间为 6 s 到 30 min，可计算得出  $k = 33.11$ 。从地图中随机选取了 100 组出发路口到目标路口的行车时间进行测试，比较了基于分布式延时路由算法的最短时间和 Dijkstra 算法计算出的实际最短时间，结果如图 8 所示。

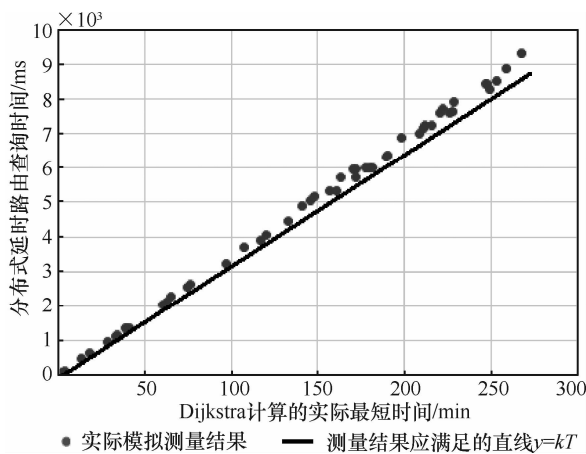


图 8 最短路径查询的测试结果 1

从图 8 可以看出，依据本文算法计算所得的最短时间与实际最短时间符合正比例函数，且斜率与前述转换比例  $k$  十分接近，说明查询所得的最短路径与实际最短路径相符。实验中的数据点略在直线  $t=kT$  的上方，是由于数据分组在节点间传递所带来的延时引起的，由于差距很小，并不影响最短路径的计算。

第二组实验，将行驶时间 6 s 到 30 min 对应到传播延时 10 ms 到 600 ms，可计算出  $k = 20.07$ 。采用第一组实验方法，随机测试了另 100 组点对，得到的结果如图 9 所示，相比第一组实验，在相同的实际最短路径时，虽查询的时间更短，但误差更大。

实验中，对  $k$  的不同取值进行与第一、第二组相同的实验，并计算它们平均查询时间和误差，结果如表 3 所示。

从表 3 中可以看出，比例系数  $k$  对于分布式延时路由查询的结果影响较大， $k$  取值太大会使一次查询的时间增加，而  $k$  太小，查询的误差会变大。

通过实验，在一般局域网条件下（相邻路口的网络延时小于 1 ms）。 $k$  取值 15 到 37 之间是可行的。当  $k < 15$  时查询误差已经高于 10%，而当  $k > 37$  平均查询时间超过了 5 s，查询效率慢。

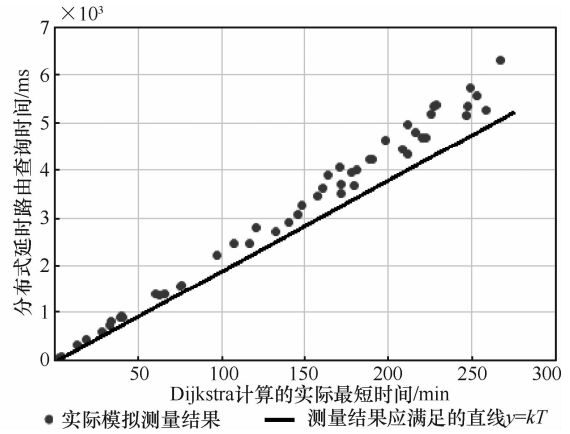


图 9 最短路径查询的测试结果 2

表 3 不同  $k$  值下的平均查询时间与误差

$k$ 取值	平均查询时间/ms	平均查询误差/%
13	2 190.26	11.97
14	2 298.18	10.07
15	2 422.48	9.10
17	2 710.90	8.21
20	3 180.94	7.88
25	3 863.88	5.86
33	4 974.22	3.46
37	5 475.74	2.24
40	5 908.60	2.01
45	6 629.06	1.86

### 5.2 实验 2

实验 2 主要是验证分布式邻近目标查询算法的并发查询效率。

实验 2 的模拟实验中，分别采用集中式的 Dijkstra 算法和分布式延时路由算法，针对同一时刻发起的随机不同次数的并发查询进行实验，分别统计了 2 种算法完成所有查询的总时间。实验结果如图 10 所示。

实验结果表明，当并发查询很少时，分布式路由查询并没有优势甚至不如经典的 Dijkstra 算法，这是因为分布式延时路由算法需要在路网中延时广播数据分组，而后者主要在本地执行查询。但是，当并发查询数量增多时，分布式延时路由算法的总查询时间的增长很缓

慢，而 Dijkstra 算法的时间是个叠加的过程。因此，在大规模的动态路网中，当并发查询的次数多时，基于延时路由的分布式邻近目标查询算法更有效率。

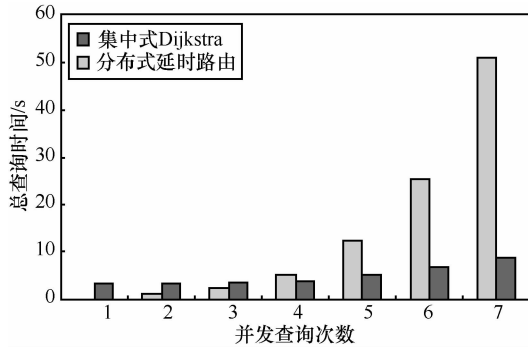


图 10 分布式延时路由与 Dijkstra 算法对比

### 5.3 实验 3

实验 3 将本文提出的分布式邻近查询算法与传统集中式查询 A\*算法<sup>[13]</sup>进行比较。在图 7 所示的动态路网中，随机生成一些并发请求，分别是最邻近查询请求和  $k$  邻近查询请求，实验结果如图 11 和图 12 所示。

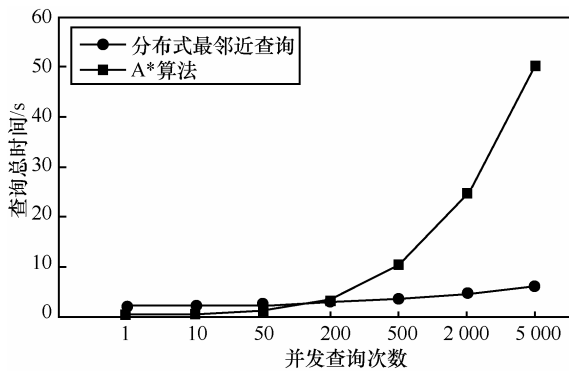


图 11 分布式最邻近查询性能比较

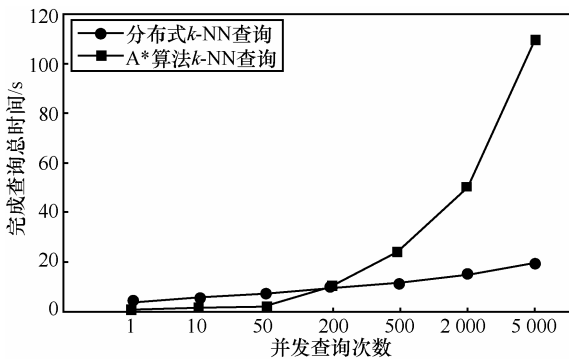


图 12 分布式 k-NN 性能比较

从图 11 和图 12 可以看出，分布式最邻近查询

和分布式  $k$  邻近查询一样，在并发查询请求较少时 ( $<200$ )，与传统方式相比没有优势可言，结果可能更差，然而当路网中查询请求很多时，查询总时间远小于传统方法。因此分布式邻近查询算法也有着特定的适用场景，在并发查询多的场合下，如交通路网中，大量车辆同时进行查询请求，分布式邻近查询算法有着更优的表现。

## 6 结束语

在本文提出的分布式邻近目标查询算法中，地理信息按照地理位置分布式的存储在就近的路口智能摄像头。对于每个路口智能摄像头，无需处理复杂的大规模交通地理数据，只需要存储和查询与之相关的少量交通地理信息。与传统的集中式存储方式对比，大大提高了存储和查询的效率。而基于此进行的分布式邻近目标查询，相对于传统邻近目标查询而言，算法实现上更加简洁高效，借助智能摄像头网络，将查询问题转化为网络中的数据分组扩散问题。实验结果表明，在大量并发查询的场合中，分布式邻近目标查询算法效率较高，查询结果与动态路网中车辆到达目标的最短路径一致。

### 参考文献：

- [1] SCHILLER J H, VOISARD A. Location-Based Services[M]. San Francisco: Morgan Kaufmann, 2004.
- [2] XU H, TEO H H, TAN B C Y, *et al.* Research note-effects of individual self-protection, industry self-regulation, and government regulation on privacy concerns: a study of location-based services[J]. Information Systems Research, 2012, 23(4): 1342-1363.
- [3] ZHANG N, WANG Y, FENG J. Saturn: a Fast Keyword  $k$ -NN Search System in Road Networks[M]. Web-Age Information Management. Springer Berlin Heidelberg, 2013: 203-215.
- [4] KU W S, ZIMMERMANN R, WANG H, *et al.* Adaptive nearest neighbor queries in travel time networks[A]. Proceedings of the 13th annual ACM international workshop on Geographic information systems[C]. ACM, 2005.210-219.
- [5] GUTTMAN A. R-Trees: a Dynamic Index Structure for Spatial Searching[M]. ACM, 1984.
- [6] KAMEL I, FALOUTSOS C. Hilbert R-Tree: an improved R-Tree using fractals[A]. 20th International Conference on Very Large Data Bases[C]. 1994.500-509.
- [7] YANG C, RASKIN R. Introduction to distributed geographic information processing research[J]. International Journal of Geographical Information Science, 2009, 23(5): 553-560.
- [8] SUNG K, BELL M G H, SEONG M, *et al.* Shortest paths in a network with time-dependent flow speeds[J]. European Journal of Operational Research, 2000, 121(1): 32-39.

(下转第 135 页)

- Modeling and Simulation, Second International Conference[C]. 2010.411-415.
- [8] HE N, GAN W Y, LI D Y. Evaluate nodes importance in the network using data field theory[A]. International Conference on Convergence Information Technology[C]. 2007.1225-1230.
- [9] 肖俐平, 孟晖, 李德毅. 基于拓扑势的网络节点重要性排序及评价方法[J]. 武汉大学学报(信息科学版), 2008, 33(4): 379-383.  
XIAO L P, MENG H, LI D Y. Approach to node ranking in a network based on topology potential[J]. Geometrics and Information Science of Wuhao University, 2008, 33(4):379-383.
- [10] 淦文燕, 赫南, 李德毅等. 一种基于拓扑势的网络社区发现方法[J]. 软件学报, 2009, 20(8): 2241-2254.  
GAN W Y, HE N, LI D Y, *et al.* Community discovery method in networks based on topology potential[J]. Journal of Software, 2009,20(8): 2241-2254.
- [11] 张健沛, 李泓波, 杨静等. 基于拓扑势的网络社区结点重要度排序算法[J]. 哈尔滨工程大学学报, 2012, 33(6): 745-752.  
ZHANG J P, LI H B, YANG J, *et al.* An importance-sorting algorithm of network community nodes based on topological potential[J]. Journal of Harbin Engineering University, 2012,33(6):745-752.
- [12] 张健沛, 李泓波, 杨静等. 基于归属不确定性的变规模网络重叠社区识别[J]. 电子学报, 2012, 40(12): 2512-2518.  
ZHANG J P, LI H B, YNAG J, *et al.* Variable scale network overlapping community identification based on identity uncertainty[J]. Acta Electronica Sinica, 2012, 40(12):2512-2518.
- [13] 李泓波, 张健沛, 杨静等. 基于社区节点重要性的社会网络压缩方法[J]. 北京大学学报(自然科学版), 2013, 49(1):117-125.  
LI H B, ZHANG J P, YANG J, *et al.* Social network compression based on the importance of the community nodes[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2013, 49(1):117-125.
- [14] 赵东杰, 王华, 李德毅等. 基于拓扑势熵的维基百科词条编辑演化研究[J]. 科技导报, 2012, 30(4): 71-74.
- ZHAO D J, WANG H, LI D Y, *et al.* Article edit evolution in wikipedia based on topology potential entropy[J]. Sciena & Technology Review, 2012, 30(4):71-74.
- [15] LI D. Artificial Intelligence with Uncertainty[M]. New York: CRC Press, 2007.
- [16] YE Q, ZHU T, HU D Y, *et al.* Cell phone mini challenge award: social network accuracy-exploring temporal communication in mobile call graphs[A]. IEEE VAST 2008[C]. 2008.2007-2008.

#### 作者简介:



段松青 (1987-), 男, 湖南郴州人, 北京邮电大学博士生, 主要研究方向为云计算、文本分析、大数据。

于兴隆 (1989-), 男, 河北邢台人, 北京邮电大学硕士生, 主要研究方向为云计算、复杂网络、数据挖掘。

吴斌 (1969-), 男, 湖南长沙人, 北京邮电大学教授、博士生导师, 主要研究方向为云计算、复杂网络、智能信息处理。

王柏 (1962-), 女, 吉林省吉林市人, 北京邮电大学教授、博士生导师, 主要研究方向为通信软件工程、智能信息处理。

(上接第 123 页)

- [9] KANOULAS E, DU Y, XIA T, *et al.* Finding fastest paths on a road network with speed patterns[A]. Data Engineering, 2006, ICDE'06, Proceedings of the 22nd International Conference[C]. IEEE, 2006.10.
- [10] DU Q, FABER V, GUNZBURGER M. Centroidal Voronoi tessellations: applications and algorithms[J]. SIAM Review, 1999, 41(4): 637-676.
- [11] WANG J, CUI C, *et al.* A parallel algorithm for constructing Voronoi diagrams based on point-set adaptive grouping[J]. Concurrency Computat: Pract Exper, 2013, 26: 434-446.
- [12] 龙其, 叶晨, 张亚英. 动态路网中基于实时路况信息的分布式路径生成算法[J]. 计算机科学, 2014, 41(9):259-262,278.  
LONG Q, YE C, ZHANG Y Y. Distributed path generation algorithm based on real-time traffic information in dynamic road network[J]. Computer Science, 2014, 41(9):259-262,278.
- [13] 张翼, 唐国金, 陈磊. 时相关车辆路径规划问题的改进 A\*算法[J]. 控制工程, 2012, 19(5):750-756.  
ZHANG Y, TANG G J, CHEN L. Improved A\* algorithm for time - dependent vehicle routing problem[J]. Control Engineering of China, 2012, 19(5):750-756

#### 作者简介:



叶晨 (1980-), 男, 安徽天长人, 同济大学博士生、讲师, 主要研究方向为嵌入式计算、智能交通、无线网络等。

杨振宇 (1988-), 男, 安徽淮北人, 同济大学博士生, 主要研究方向为智能交通、控制算法等。

喻剑 (1975-), 男, 浙江义乌人, 同济大学讲师, 主要研究方向为物联网、RFID 及应用。

龙其 (1987-), 男, 湖南长沙人, 同济大学硕士生, 主要研究方向为嵌入式计算、智能交通。