Computers and Electrical Engineering 68 (2018) 526–535

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Intelligent optimal route recommendation among heterogeneous objects with keywords☆

Ashokkumar P.*,[a], Arunkumar N.[b], Don S.[c]

[a] School of Computer Science and Engineering, VIT, Vellore, Tamil Nadu, India
[b] School of EEE, Sastra University, Thanjavur, Tamil Nadu, India
[c] School of Computer Science and Engineering & TIFAC CORE in Automobile Infotronics, VIT, Vellore, Tamil Nadu, India

ARTICLE INFO

Keywords:
Heterogeneous search
K-nearest search
Keyword search
Route recommendation
Shortest path

ABSTRACT

Textual information is often added to spatial objects, and thus a spatial search query includes both location-based and textual-based information. For example, a user may want to visit a dental hospital with orthodontic appliances and dental lasers, before going to a pharmacy to buy some drugs (ibuprofen and tylenol). Currently, a brute force technique combined with a spatial inverted index is used to find an optimal set of spatial objects that satisfy both location and textual constraints. In this study, we identify some problems with the brute force technique and propose two algorithms that recommend an optimal shortest path among heterogeneous objects. Our method outputs the shortest path from the user point to all the heterogeneous objects. Results obtained using our algorithm demonstrated that it produces the optimal shortest path compared with existing spatial-textual search algorithms.

1. Introduction

Spatial databases are particular types of databases that are mainly used for storing and retrieving geometrical data, which may be simple points, complex lines, polygons, or three-dimensional (3D) shapes. A small geometric object such as a building can be represented as points, whereas objects representing an area such as a city or a country may be stored as polygons in a spatial database [1,2]. There are many ways of retrieving a list of nearest spatial objects, such as by measuring the distance between the current point and points in the database, or by calculating the distance between the current position and any point in a polygon present inside the database [3,4]. These methods are highly suitable for finding the nearest geometrical objects, but spatial queries are now increasingly performed using search engines [5]. Thus, finding the nearest geometrical object is not sufficient to meet the needs of search engines, so small amounts of textual information are also considered as well as location-based information. The textual information may comprise some properties of a geometric object, e.g., a restaurant can have textual properties such as burgers, pizza, and salads, whereas a textile shop may have textual properties such as formal suits, streetwear, and outfits.

Finding the k-nearest neighbours among homogeneous groups is very easy, e.g., we could find five nearest musical shops that sell guitars, pianos, and violins. This search can be conducted using two methods. The first method involves sorting all of the music shops in ascending order based on their distance from the user point until five shops remain that match all three keywords in terms of their textual properties. The second method is the opposite of the first method, where the spatial database filters the available objects based on three keywords, before performing a distance calculation to find the five nearest objects. These two methods may increase

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. G. R. Gonzalez.
* Corresponding author.
E-mail addresses: ashokkumar.p@vit.ac.in (A. P.), don.sasikumar@vit.ac.in (D. S.).

https://doi.org/10.1016/j.compeleceng.2018.05.004
Received 20 February 2018; Received in revised form 2 May 2018; Accepted 3 May 2018
Available online 16 May 2018
0045-7906/

the time required for searching in some cases, such as when the correct geometrical point is far away from the user or when the nearest objects are missing at least one keyword. The geometrical points can also be fetched incrementally one by one [6].

Several optimized algorithms have been proposed for finding the k-nearest objects in homogeneous groups, such as IR^2 tree [4], SI-algorithm [7], and RSTkNN [8]. These algorithms find the k-nearest neighbours among homogeneous objects by considering both location-based and text-based constraints. However, these algorithms fail to find the k-nearest heterogeneous neighbours.

The IR^2 algorithm is problematic because it generates many false hits. If a geometrical object is filtered out by the IR^2 algorithm, the object does not possess any of the textual properties, but when a geometrical object is accepted by the algorithm, the object may or may not satisfy all of the textual constraints. False hits are likely to occur with the IR^2 algorithm but this drawback is overcome by the spatial-inverted indexing algorithm (SI-Index), which produces no false hits. Another problem with these algorithms is they will not find objects that satisfy partial textual constraints. Thus, these algorithms may return the nearest neighbour that satisfies all of the textual constraints even if it is far away, but two or more nearby geometrical objects might satisfy the textual constraints partially and when the partial objects are combined they could satisfy all of the textual constraints. Our algorithm also finds these partial objects.

Finding k-nearest neighbours among heterogeneous objects is challenging because we need to find N sets of k homogeneous objects, where each homogeneous set is nearest to the previous homogeneous set and the first homogeneous set is nearest to the user point. For example, if a user wants to go to a cinema (to see a Star Wars movie), go shopping (to buy a pen, notebook, and calculator), and then eat in a restaurant (to eat fried rice and noodles) before going home, he can submit this query to a search engine, which finds three different homogeneous groups (cinema theatres, shopping complexes, and restaurants) and then determines the optimal shortest path closest to the user location. If $k = 5$, then the search engine should find the five nearest cinema theatres to the user point that are showing the Star Wars movie, the five nearest toy shops to the cinema theatres that sell all the three textual search constraints (pen, notebook, and calculator), and the five nearest restaurants that sell both fried rice and noodles. The geometrical objects returned by the search query may either satisfy all of the textual constraints or a set of partially satisfactory objects, which can be combined to satisfy all of the textual constraints. Finally, an optimal route should be found that starts from the user point and ends at a restaurant. The route should be both the nearest (by location) and satisfy all of the textual constraints.

1.1. Contributions

We propose a hybrid optimized algorithm to find the k-nearest neighbours that satisfy both location and textual properties. This algorithm improves the search efficiency by outputting the specific objects near to the user point. Our method comprises the following steps.

- The first step involves splitting the user query to find the unique homogeneous items.
- In the next step, the k-nearest neighbours are found in each homogeneous group. The k-points should satisfy both the spatial and textual constraints.
- The final step involves finding a route from the user point that connects all of the homogeneous group so the distance is minimized and all of the textual constraints are met.

The remainder of this paper is organized as follows. In Section 2, we describe some related research into search with keywords. In Section 2.5, we explain the drawbacks of the SI algorithm. In Section 3, we state two problems and their solutions are defined in Section 4 and Section 5, respectively. Our experiments show that the algorithms determine the minimum distance between objects compared with existing algorithms and with a shorter query execution time. The results are discussed in Section 6.

2. Related work

2.1. Signature files

Signature files [9] are a traditional method used for checking that a text feature is present in a document. Signature files are generated using a superimposed coding method. In the signature files, the feature list is divided into unique small blocks [10,11]. These blocks are hashed into word signatures, which are then superimposed to form a block signature. When a search query is

Table 1
Example of keyword searching using SI. The table shows how random bit striping is conducted where a unique textual string is assigned to each and every word.

Words	Textual string
a	11,000
b	01100
c	00110
d	00011
e	01010