

# A Proposal of Spatial Operators for a Collaborative Map Search System Yuanyuan Wang<sup>1</sup> Panote Siriaraya<sup>2</sup>, Yukiko Kawai<sup>2,3</sup>, Keishi Tajima<sup>4</sup> <sup>1</sup>Yamaguchi University <sup>2</sup>Kyoto Sangyo University <sup>3</sup>Osaka University <sup>4</sup>Kyoto University

### "Meet Up Application" a collaborative map search system: https://yklab.kyoto-su.ac.jp/~ichimura/spatialQuery/ Start: current location cafe Spot: Start (T): Kyoto Aquarium meeting area 🛊 Add a Start (T) Start (T): target location avoid area car 🛊 イオンモール京都五条 市立七条第三小 😵 都市立西京極中 京都南端院 🔮 ○ 根来寺 智借 中恭天皇 九燦陸 😋 Google

Find **cafe shops** that are equally near to all of Find **restaurants** located within 700m of **Times Square** and within1km users' start locations by walking (Start: Kyoto from Grand Central and Pennsylvania station **station**, Start (T): **Kyoto Aquarium**)

# The Spatial Query Language

- **Rule 1** The syntax of the most primitive unit of spatial queries is defined as follows: A\_distance  $\alpha$ . The distance by using a unit distance (e.g., 200m) or by using continuous spaces (e.g., the N nearest  $\alpha$ ). Ex.) A 800m  $\alpha$ : identify the  $\alpha$  objects which exist inside the region 800m from the origin point A  $A_{-}\alpha$ : identify the 3 nearest  $\alpha$  objects from the origin point A
- **Rule 2** The keywords (e.g., A and  $\alpha$ ) used in the primitive unit of the spatial query are encapsulated within a double quotation mark (e.g., "Tokyo tower" for A or "pizza shop" for  $\alpha$ ).
- Rule 3 Each primitive unit of the spatial query can be combined with other units through the use of spatial, directional and distance operators in a mathematical expression format. Ex.)  $(A_800m_{\alpha}) + (B_300m_{\alpha})$



## **Overview of Proposed Map Search Applications**

### **Spatial Query Search**

https://yklab.kyoto-su.ac.jp/~sakata/spatialQueryDemo/



Acknowledgments: The work in this paper is partially supported by JSPS KAKENHI Grant Numbers 16H01722, 15K00162, 17K12686.

## Query Language Instructions

### Primitive Search Unit

#### Supported Spatial Operators

• Syntax: Each unit consists of "Start Location" Distance "property keyword". For example "Times square" 500m "book shop" The start location and property keywords have to be placed in double quotations, else an error will be returned.

• Search by Distance: The distance could be in meters (use m only) or kilometers (use km only). For example "Times square" 1km

· Search by Space key: If no distance is entered, the query would identify and return the nearest shops based on the number of empty spaces. For example "Times square" "book shop" would show the three nearest book shops from Times square

· Addition Operator (+): Identifies all the venues located between two start locations. For example "Times square" 500m "book shop" + "Empire State Building" 500m "book shop" will find all the book shop within 500m of Times square and the Empire state building.

• Difference Operator (-): Identifies all the venues located within one



Find 4 nearest cafe shops from current location by \_\_\_\_ - Details of results (address, review score etc.) A link of the route to the destination

# **Evaluation and Prototype**

5 students completed 2 search tasks through proposed system and Google Maps by **SUS Average accuracy**: proposed system (90%) > Google Maps (10%) **Average satisfaction**: proposed system (**3.92**) > Google Maps (2.08) Confirmed proposed system greatly improves both accuracy and satisfaction

