

Faculty of Information Technology  
Monash University



Literature Review — Semester 2, 2014

## Isolated Region Spatial Query

Jian Loong Liew 22545727

Supervisor: Associate Professor David Taniar

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Formulating of Definitions . . . . .	3
<b>2</b>	<b>Related Works</b>	<b>4</b>
2.1	Overview of Spatial Queries . . . . .	4
<b>3</b>	<b>Nearest Neighbour</b>	<b>4</b>
3.1	Range Euclidean Restriction . . . . .	4
3.2	Aggregate Nearest Neighbour Query . . . . .	4
3.3	Constrained Nearest Neighbour . . . . .	5
3.4	Reverse $k$ -Nearest Neighbour ( $RkNN$ ) . . . . .	5
<b>4</b>	<b>Constrained Range Search</b>	<b>5</b>
<b>5</b>	<b>Reverse Region Query</b>	<b>5</b>
<b>6</b>	<b>K-Means</b>	<b>5</b>
<b>7</b>	<b>Group Nearest Neighbour</b>	<b>6</b>
7.1	Multiple Query Method (MQM) . . . . .	6
7.2	Single Point Method . . . . .	6
7.3	Minimum Bounding Method (MBM) . . . . .	6
7.4	Group Cloest Pair Method . . . . .	6
7.5	Two Ellipse-based Pruning Method . . . . .	6
<b>8</b>	<b>Probablistic Group Nearest Query</b>	<b>7</b>
<b>9</b>	<b>Group Nearest Group</b>	<b>7</b>
9.1	Exhaustive Hierarchical Combination Algorithm (EHC) . . . . .	7
9.2	Subset Hierarchical Replacement Algorithm (SHR) . . . . .	7
9.3	Comparison of query processing solutions . . . . .	7
9.4	Zone of Influence . . . . .	7
<b>10</b>	<b>Optimum Region Query</b>	<b>7</b>
10.1	Circle Partition and Arcs Superposition (CPAS) . . . . .	8
<b>11</b>	<b>Skyline Query</b>	<b>8</b>
<b>12</b>	<b>Isolated Region</b>	<b>8</b>
<b>13</b>	<b>R-Trees</b>	<b>8</b>
13.1	Rectangle Handling . . . . .	8
<b>14</b>	<b>TPR Tree</b>	<b>8</b>
<b>15</b>	<b>Voronoi Diagram</b>	<b>8</b>
15.1	On-line Progressive Expansion (OPE) . . . . .	9
15.2	Off-line Precalculation (OPC) . . . . .	9



# 1 Introduction

A spatial query is a query that returns features based on their spatial relationship with a query geometry. The query of interest here is the *Isolated Region* query. This is a distinctive spatial query in terms of its requirements and the name of it is coined from its requirements. The objective of this query is to find a region where an object can be placed so that it is not in the state of *isolation* and yet still be in the state where it is in a *crowd*. The input of this query will be a set of points,  $S = \{p_1, \dots, p_n\}$  where each point represents an object of interest (OOI). The output of this query will be a region where any object placed within it satisfies the criteria. In order to properly solve, this each object needs to create a region of influence. The reasoning behind this, is because it is important to know the region of influence of each object in order to properly define a solution for this query.

This query should always pick the best region where the object can be placed in. In other words, the number of solutions to this query can be more than 1 regions.

This query consist of the combination of two requirements which are:

- The object must not be in the state of isolation.
- The object must be in the state where it is in a crowd.

Before attempting to go further the conditions of each scenario will be further defined.

Based on Figure 2a, it can be can seen that when only 1 object is used as the input, the conditions are quite clear. Assuming this object A, has a region of influence denoted by a circle around it, so any object placed inside the circle will be under its zone of influence. It is quite clear that we can place any object that satisfies the condition in this scenario quite easily, because any object placed just outside the circle would be considered to be in the crowd and not isolated.



Figure 1

Based on Figure 2a, it can be can seen that when only 1 object is used as the input, the conditions are quite clear. Assuming this object A, has a region of influence denoted by a circle around it, so any object placed inside the circle will be under its zone of influence. IT is quite clear that we can place any object that satisfies the condition in this scenario quite easily, because any object placed just outside the circle would be considered to be in the crowd and not isolated.

However this situation becomes more complex when the number of object that are the input are increased. Based on Figure 1b, it can be seen that with 2 objects as the input, the situation of being crowded will be changed. There are also many variations of how these 2 inputs can be. For example, if the distance between these two objects are changed, the state of being in the *crowd* will be different. This is because as it moves further it will be in a state of isolation. The main question here, would be how far out before it considered to be in the isolated state.



Figure 2

This is an interesting scenario because depending on the way the algorithm is handled, results will be different depending on the approach.



Figure 3

In Figure 3a, it can be seen that the state of being crowded will be in 2 possible places represented by C and D. This is because both of these states are similar in nature. This however is not true when the two objects are far apart. In Figure 4b, it is clear that the crowded state can be satisfied by placing the object in the middle between these two objects. However this situation is not as simple as it seems, because as two objects move further away, it is clear placing the object in between these two will not satisfy the condition of being the crowd.

This situation further increases in complexity as the number of objects are increased. For example, when the number of input objects are 3, there are more possible variations to this scenario. The reason for this is because the scenario of 3 objects is more interesting. In Figure 3a, shows the possible variations when there are 3 objects of interest. For example, in Figure 4a, the potential placement that can satisfy the condition is below the 3 objects of interest. It is quite simple to see this solution in this scenario, however assuming the objects move further away from each other, this will change the scenario. If this happens, it is obvious that the solution to this query is changed as well.



Figure 4

Previous work has been done in order to satisfy the requirements individually, however no work has been done where the two constraints are placed together. Placing the two requirements together changes the way the query needs to be processed and thus this literature review aims to explore the various algorithms and approaches that can be used to solve the query.

The definitions of the terms used are as follow.

**Definition 1** *The state of being a crowd is where the query point is the zone of influence of the maximum number of objects of interest. However, in this query the condition of being in the crowd must be relaxed in order to provide a beneficial solution. Thus, being in a crowd means that the query point must be near the zone of influence of the maximum number of objects.*

**Definition 2** *The state of isolated or isolation means that the object is not under the influence of any objects of interest. This means that it has the least numbers nearest neighbours.*

**Definition 3** *An Isolated Region is where the object is far away from any related object.*

**Definition 4** *A safe region is an area where the query point can move freely without effecting the query result. For this query, the safe region is where an object can be placed so that it does not violate the conditions.*

**Definition 5** *A zone of influence is a region where by any points or locations or coordinates or pixels in this region will consider this object as its closest object, compared to any other object.*

## 1.1 Formulating of Definitions

Here the state of being in the “crowd” will be detailed. For example, if only one query point is given as the input where  $S = \{p_1\}$ , the condition of the crowd needs to be defined.

With that, the formalisation of the *Isolated Region* query is as follows. Given a set of points,  $p_1, \dots, p_n$ , which are the input. The output of this will be a region in which, it is a safe zone which obeys the two requirements in which the state of any object placed within this safe zone will be in the state of being in a crowd and isolated. This output region maybe of of any shape.

Xuan in [31] formalised the definition of an *Optimum Region* in which, “Given a finite set of point objects and a positive value  $r$ , the optimum region includes all points which can cover the maximum number of objects in the finite set as the center of a circle with radius  $r$ .”. This query will satisfy the requirement where the region will has the maximum number of nearest neighbours, however the input of this query will require the value  $r$ . The proposed algorithm to answer this query suggested by Xuan, is the Circle Partition and Ars Superposition. This will be futher detailed in Section xx. Coined to be the “Optimum Region”, this query aims to determine the best possible position to place an object. Knowing this it would be possible to

Korn and Muthukrishnan in [16] formalised the notion of influence based on reverse nearest neighbour queries and its variants. This was called *Influence Set*. This is because the reverse nearest neighbour is not symmetric. However. this concept of influence is often changed depending the application at hand and difficult to formalise.

Cheema et al (2011), suggested a more generic concept called *influence zone* and showed that the influence zone can be used to efficiently compute the influence sets. This influence zone has various applications in location based services and decision support systems. This is because the influence zone may be used for market analysis as well as targeted marketing [5]. This concept is more generic compared to the notation of influence set proposed by Korn et al (2000).

## 2 Related Works

[11]

[4].

- Reverse Nearest Neighbour by Korn et al. (2000) [16]
- Group Nearest Neighbour by Papadias et al. (2004) [20].
- Group Nearest Group by Deng et al. (2012) [7]
- Optimum Region by Xuan (2012) [31]

In order to process the various queries in the spatial road network are as follows

- Voronoi Diagrams
- VN<sup>3</sup> [15]
- Network Voronoi Diagram [33]

### 2.1 Overview of Spatial Queries

This diagram is based on deriving information from [28].

## 3 Nearest Neighbour

[14]

[23]

The nearest neighbour problem is considered to be one of the best known problems in computer science.

[24]

### 3.1 Range Euclidean Restriction

[22].

### 3.2 Aggregate Nearest Neighbour Query

Aggregate nearest neighbour queries returns the object that minimizes an aggregate distance function with respect to a set of query points [34]. A scenario for ANN is as follows: Assuming for example,  $n$  users at location  $(q_1, \dots, q_2)$ . An ANN query outputs the facility that minimises the sum of distances that the users have to travel in order to meet there [21]. Thus, the input for an ANN query will be the set of static data points given to it.

### 3.3 Constrained Nearest Neighbour

[10].

### 3.4 Reverse $k$ -Nearest Neighbour (R $k$ NN)

The goal of a reverse  $k$ -nearest neighbour query is to identify the “influence” of query object on the whole data set [1].

A Reverse Nearest Neighbour (RNN) search is a method to retrieve all objects that consider the query point as the nearest neighbour. An example would be like when a marketing application in which the issue is to determine the business impact of opening an outlet of Company  $A$  at a given location. A simple task is to determine the segment of  $A$ ’s customer who would be likely to use this new facility [16]. Korn et al (2000) broken this down into two cases which are static and dynamic cases and also formalised a novel notion of influence based on reverse nearest neighbour queries and its variants. Influence sets based on reverse nearest neighbour (RNN) queries seem to capture the notion of influence based on the example stated by them.

This methodology to generate the influence zone of each location will be used in order to generate the influence zone of each object of interest.

## 4 Constrained Range Search

[32].

[15]

[35]

[13]

## 5 Reverse Region Query

The input of this are objects and the output of this query is a region.

## 6 K-Means

The K-Means algorithm, which is part of a supervised machine learning algorithm can potentially be used to solve this problem. By applying the concept of k-means, which uses the query points as the input which is  $Q = \{q_1, \dots, q_n\}$ , and a value of  $n$ , it is possible to cluster all the object based on a certain number of clusters. This bears similarities to the GNN query however instead of using candidate positions as the input, this uses a  $n$  value that determines the number of clusters generated. The benefit of using this algorithm is that it is comparatively faster compared to the algorithms. Even though so, it depends on the situation because applying the concept of Voronoi diagrams to this algorithm might alter its complexity.



## 7 Group Nearest Neighbour

This was proposed by Papadias et al [20]. It is considered to be a novel form of Nearest Neighbour search. The GNN query is different from the traditional  $knn$  query which only specifies a single query point, the GNN query has multiple query points. The input of this problem consist of a set of  $P = \{p_1, \dots, p_n\}$  of static data points in multi dimensional space and a group of query points.  $Q = \{q_1, \dots, q_n\}$ . The ouput of this contains the  $k > 1$  data points [20].

An example scenario for the GNN would would be to select a meeting place from all available meeting places. For example, for three executive directors who are located in three different places, to meet at the closest meeting place [28].

This is considered to be an expensive problem by definition because of the number of data points as well as query points it needs to process as it is considerably more complex than the traditional  $knn$  queries. The reason for this complexity is mainly due to two reasons [17]. The first is because there are multiple query points being specified which requires more distance computation and the other is because the fact that the query point can be distributed within the data space in arbitrary ways, creating a large search region.

### 7.1 Multiple Query Method (MQM)

This algorithm utilises the threshold algorithm where it performs incremental NN queries for each point in  $Q$  and combines their results [20]. The MQM retrieves the NN for every point in query set  $Q$ , it sometimes accesses the same tree nodes for different query points and this causes its cost to increase fast with the query set cardinality.

### 7.2 Single Point Method

The single point method introduced by Papadias et al in [20].

### 7.3 Minimum Bounding Method (MBM)

### 7.4 Group Cloest Pair Method

### 7.5 Two Ellipse-based Pruning Method

Li et al (2005) suggested a distance pruning method using an ellipse. The pruning method is “If a point or an MBR is far away enough with respect to the two points we choose as the approximate ellipse, they cannot be in the final answer.” This method is compared to the SPM method and the MBM method. The authors claim that this method works more efficiently than both SPM and MBM because the ellipse used in the methods are less distance computational during the search and can prune unqualified nodes more efficiently. IT is noted that the two ellipse-based pruning method can be used in both the depth-first and best-first travel paradigms.

## 8 Probablistic Group Nearest Query

[18].

## 9 Group Nearest Group

Group Nearest Group (GNG) query can be defined as a query which finds one data point  $p$  from a data point set  $D$  such that the total distance from  $p$  to the points in a query point set  $Q$  is minimal [7]. This is regarded as the generic version of the GNN query. When  $k = 1$ , a GNG query is reduced to a GNN query. A GNG query can also be called as a k-median clustering in operations search which is a partition based clustering problem with group data points into  $k$  which is a given number of clusters based on an optimization objective function.

The scenario of a GNG query is as follows. A security service provider plans to set up several new branches to serve several business districts which can be represented by a set of land marks, such as well-known buildings. The max number of branches is usually on constrained by factors such as business cost. Under this constraint the provider wishes to select branch locations from many choices such that the reponse time to security alarms can be minimised that is the average distance between these landmarks to the nearest branch is minimal.

The GNG algorithm has its useful because it will find a meeting point from a set of groups, however it requires 2 set of inputs which are the a set of data points.

Query points with different weights are also useful in many situations [7].

### 9.1 Exhaustive Hierarchical Combination Algorithm (EHC)

### 9.2 Subset Hierarchical Replacement Algorithm (SHR)

### 9.3 Comparison of query processing solutions

### 9.4 Zone of Influence

## 10 Optimum Region Query

Another spatial query of interest is the *Optimum Region*. This problem appears in Euclidean space however its methodology can be applied to the spatial road networking using Voronoi Diagrams. Optimum region is defined as a region which includes all the points which can cover the maximum number of objects in the finite set as the center of a radius  $r$  [31]. The objective of the optimum region search is to find a region where the center of a circle with a fixed radius can be located to cover most objects of interest in the given set. An example of an optimum region query is as follows:

- Build a hospital in a community and let most resident reach to the hospital within 20 minutes or 15 km.
- A company plans to establish a new Wi-Fi base station, the coverage range of which is 20 meters, to cover most Wi-Fi devices in the company.

It is noted however, the optimum region query does not really hand the problem scenario at hand. This is because the optimum region aims to retrieve the area with the maximum number of intersection regions. An example of an output from the optimum region is as follows.

### 10.1 Circle Partition and Arcs Superposition (CPAS)

This algorithm relies on the polar coordinates of the system.

## 11 Skyline Query

## 12 Isolated Region

## 13 R-Trees

In order to efficiently solve the query processing in the multi dimensional splace, an efficient indexing system has been proposed by various authors. An example would be the *R-trees* [12]. This has then been further developed and enhanced by various authors. For example, a variation of the *R-trees* is the  $R^+$  - *trees* that avoids overlapping rectangles in intermediate nodes of the tree that is introduced. [3].

### 13.1 Rectangle Handling

Table from  $R^+$  tree goes here.

## 14 TPR Tree

[30].

## 15 Voronoi Diagram

[26]

Okabe et al in [19] presented a thorough discussion on regular and network Voronoi diagrams.

[6].

[9]

[2]

[25].

[27]

[29]

[8]

Kolahdouzan et al in [15] suggested an approach termed Voronoi based Network Nearest Neighbour ( $VN^3$ ), which reduces the problem of distance computation in a very large network, in to the problem distance computation in a number of much smaller networks plus some additional table lookups. This is achieved by generating a first-order network Voronoi diagram over the points of interest.

### 15.1 On-line Progressive Expansion (OPE)

### 15.2 Off-line Precalculation (OPC)

## 16 Comparison

The table below list the various spatial queries based on their input, output as well as the various algorithms used.

Item		
Name	Input & Output	Algorithm
Aggregate Nearest Neighbour	per gram	13.65
Reverse Nearest Neighbour	each	0.01
Gnu	stuffed	92.50
Group Nearest Group (GNG)	stuffed	33.33
Optimum Region	frozen	Circle Partition and Arcs Superposition (CPAS)

## References

- [1] Elke Achtert, Christian Böhm, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. Efficient reverse k-nearest neighbor search in arbitrary metric spaces. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 515–526. ACM, 2006.
- [2] Franz Aurenhammer. Voronoi diagrams a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.
- [3] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. *The R\*-tree: an efficient and robust access method for points and rectangles*, volume 19. ACM, 1990.
- [4] Stefan Berchtold, Bernhard Ertl, Daniel A Keim, H-P Kriegel, and Thomas Seidl. Fast nearest neighbor search in high-dimensional space. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 209–218. IEEE, 1998.
- [5] Muhammad Aamir Cheema, Xuemin Lin, Wenjie Zhang, and Ying Zhang. Influence zone: Efficiently processing reverse k nearest neighbors queries. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 577–588. IEEE, 2011.
- [6] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. *Computational geometry*. Springer, 2000.
- [7] Ke Deng, Shazia Sadiq, Xiaofang Zhou, Hu Xu, Gabriel Pui Cheong Fung, and Yansheng Lu. On group nearest group query processing. *Knowledge and Data Engineering, IEEE Transactions on*, 24(2):295–308, 2012.
- [8] Ke Deng, Xiaofang Zhou, and Heng Tao Shen. Multi-source skyline query processing in road networks. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 796–805. IEEE, 2007.
- [9] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM review*, 41(4):637–676, 1999.
- [10] Hakan Ferhatosmanoglu, Ioanna Stanoi, Divyakant Agrawal, and Amr El Abbadi. Constrained nearest neighbor queries. In *Advances in Spatial and Temporal Databases*, pages 257–276. Springer, 2001.
- [11] Ralf Hartmut Güting. An introduction to spatial database systems. *The VLDB Journal The International Journal on Very Large Data Bases*, 3(4):357–399, 1994.
- [12] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.
- [13] Haibo Hu, Dik Lun Lee, and Jianliang Xu. Fast nearest neighbor search on road networks. In *Advances in Database Technology-EDBT 2006*, pages 186–203. Springer, 2006.
- [14] Christian S Jensen, Jan Kolářvr, Torben Bach Pedersen, and Igor Timko. Nearest neighbor queries in road networks. In *Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, pages 1–8. ACM, 2003.
- [15] Mohammad Kolahdouzan and Cyrus Shahabi. Voronoi-based k nearest neighbor search for spatial network databases. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 840–851. VLDB Endowment, 2004.

- [16] Flip Korn and S Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *ACM SIGMOD Record*, volume 29, pages 201–212. ACM, 2000.
- [17] Hongga Li, Hua Lu, Bo Huang, and Zhiyong Huang. Two ellipse-based pruning methods for group nearest neighbor queries. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 192–199. ACM, 2005.
- [18] Xiang Lian and Lei Chen. Probabilistic group nearest neighbor queries in uncertain databases. *Knowledge and Data Engineering, IEEE Transactions on*, 20(6):809–824, 2008.
- [19] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.
- [20] Dimitris Papadias, Qiongmao Shen, Yufei Tao, and Kyriakos Mouratidis. Group nearest neighbor queries. In *Data Engineering, 2004. Proceedings. 20th International Conference on*, pages 301–312. IEEE, 2004.
- [21] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems (TODS)*, 30(2):529–576, 2005.
- [22] Dimitris Papadias, Jun Zhang, Nikos Mamoulis, and Yufei Tao. Query processing in spatial network databases. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 802–813. VLDB Endowment, 2003.
- [23] Franco P Preparat and Michael Ian Shamos. Computational geometry: an introduction. 1985.
- [24] Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent. Nearest neighbor queries. In *ACM sigmod record*, volume 24, pages 71–79. ACM, 1995.
- [25] Maytham Safar, Dariush Ibrahimi, and David Taniar. Voronoi-based reverse nearest neighbor query processing on spatial networks. *Multimedia systems*, 15(5):295–308, 2009.
- [26] Cyrus Shahabi, Mohammad R Kolahdouzan, and Mehdi Sharifzadeh. A road network embedding technique for k-nearest neighbor search in moving object databases. *GeoInformatica*, 7(3):255–273, 2003.
- [27] Robin Sibson. A brief description of natural neighbour interpolation. *Interpreting multivariate data*, 21, 1981.
- [28] David Taniar and Wenny Rahayu. A taxonomy for nearest neighbour queries in spatial databases. *Journal of Computer and System Sciences*, 79(7):1017–1039, 2013.
- [29] David Taniar, Maytham Safar, Quoc Thai Tran, Wenny Rahayu, and Jong Hyuk Park. Spatial network rnn queries in gis. *The Computer Journal*, 54(4):617–627, 2011.
- [30] Yufei Tao, Dimitris Papadias, and Jimeng Sun. The tpr\*-tree: an optimized spatio-temporal access method for predictive queries. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 790–801. VLDB Endowment, 2003.
- [31] Kefeng Xuan. Range / Region-Related Queries Processing in Spatial Databases by Clayton School of Information Technology Monash University. (0190), 2012.
- [32] Kefeng Xuan, Geng Zhao, David Taniar, Maytham Safar, and Bala Srinivasan. Constrained range search query processing on road networks. *Concurrency and Computation: Practice and Experience*, 23(5):491–504, 2011.

- [33] Kefeng Xuan, Geng Zhao, David Taniar, Bala Srinivasan, Maytham Safar, and Marina Gavrilova. Network voronoi diagram based range search. In *Advanced Information Networking and Applications, 2009. AINA'09. International Conference on*, pages 741–748. IEEE, 2009.
- [34] Man Lung Yiu, Nikos Mamoulis, and Dimitris Papadias. Aggregate nearest neighbor queries in road networks. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):820–833, 2005.
- [35] Geng Zhao, Kefeng Xuan, David Taniar, and Bala Srinivasan. Incremental k-nearest-neighbor search on road networks. *Journal of Interconnection Networks*, 9(04):455–470, 2008.