# A Fuzzy Spatial Querying Model for Improving Apartment Web Services

**Huiqing Helen Yang**
**Department of Mathematics and Computer Science, Virginia State University**
**Petersburg, VA 23806, USA**

## ABSTRACT

An apartment web service is a web site intended to provide online services to the residents. With the availability of global positioning systems and the development of new technologies, the access of geo-spatial information has drawn great attention for information retrieves. This research is focused on developing an online service for an apartment to provide geo-spatial information. In this paper, a fuzzy spatial querying model under uncertainty is used to improve spatial querying. An inexact inferring with the certainty factor is investigated. An online spatial database for an apartment is implemented by utilizing PHP with MySQL database. Demonstration shows that the system can provide visitors with the amount of significant geo-spatial information about the apartment. Moreover, fuzziness and uncertainty are mainly concerned. Querying examples are provided.

**Keywords**: fuzzy logic, fuzzy inference, uncertainty, spatial query, spatial database, and web application.

## 1. INTRODUCTION

The University Apartment at Ettrick (UAE) has an official website which intended to provide online services to the residents or future residents. The original website only provided basic information about its properties. There is no geo-spatial information included, for example, distance or direction to a building in the main campus. With the availability of global positioning systems (GPS) and the development of new technologies, such as diffusion and miniaturization of GPS receivers and remote sensing, it is possible to provide geographic information in an apartment web service.

The ability to perform queries on spatial data is essential in an information retrieval system. In our previous work [1], a spatial information system was developed to improve and increase the amount of information for the users, residents or future residents. Although the simple fuzzy model presented in [1] shows the capabilities of spatial queries, the decision rules are defined subjectively. The motivation of this research is to use a binary fuzzy model [2-3] to improve the spatial querying accuracy.

Spatial querying is a more general term in the geographic queries, which can be defined as queries about the spatial relationships such as topology and direction. As a critical component in the spatial retrieval, the directional relationship is mainly concerned in the paper. The system is implemented by using an open source database MySQL and PHP web programming language [4-6].

The paper is organized as follows: section 2 presents system architecture, a spatial data model is given in section 3, a fuzzy querying model under uncertainty follows, and section 5 provides implementations, and samples of querying results are listed in section 6. Finally, our conclusion is given in section 7.

## 2. SYSTEM ARCHITECTURE

The general system architecture is shown in Figure 1.

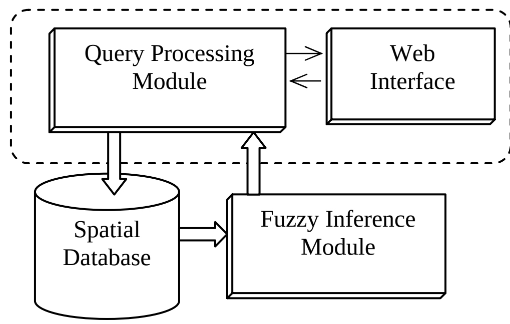


Figure 1. System Architecture

The system is implemented by utilizing PHP web programming language and MySQL database. The main components are described as follows:

### 2.1. Web Interface

The user can access the system through a friendly web interface, which has been implemented by embedding PHP scripts within HTML code.

### 2.2. Query Processing Module

The query processing uses PHP scripts to pass the queries to the PHP interpreter, which interacts with MySQL database and fuzzy inference module to provide answers based on the user queries.

### 2.3. Spatial Database

It stores objects that have spatial characteristics such as street and buildings locations. The spatial relationships among the objects are important when querying a spatial database. Although a spatial database can in general refer to a high dimensional space, we limit our discussion to two-dimension space as illustration. The ER model is used to construct a semantic model of the database. The model is implemented with MySQL database.

### 2.4. Fuzzy Inference Module

This is the main module in the system. Based on the data model of geo-spatial objects, it performs inference under uncertainty and provides answers with a certain degree of belief (Certainty Factor). The degree is represented by a crisp numerical value ranging from 0 to 1. A certainty factor of 1 indicates that it is very certain that a query result is true, and a certainty factor of 0 indicates that it is very uncertain that a query result is true.

## 3. SPATIAL DATA MODEL

Spatial data plays an important role in GIS research. Since the ability to extract information for query results is dependent on the underlying structure of data, a binary spatial model presented in our previous research [2] is used, which can interpret spatial characteristics of objects stored in the spatial database.

Assume that the spatial objects can be approximated by their minimum bounding rectangles (MBR). Figure 2 shows two objects A(xa1, ya1)(xa2, ya2 and B(xb1, yb1)(xb2, yb2) in two-dimension.

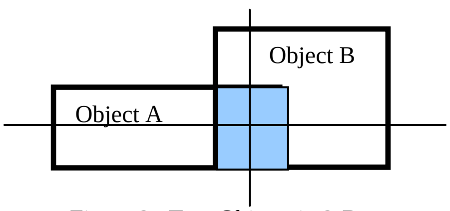


Figure 2. Two Objects in 2-D

A tuple (x, y) represents the coordinate in the longitude and latitude respectively.

## 4. FUZZY QUERYING MODEL UNDER UNCERTAINTY

Because the spatial querying deals with some concepts expressed by verbal language, fuzziness is frequently involved. Hence, the ability to query a spatial data under fuzziness is one of the most important characteristics of any spatial databases. In the paper, the directional relationship is mainly concerned. Based on the binary spatial model, a fuzzy inference model for directional queries is presented.

We define the directional relationships as

D={East, South, West, North, North-East,
North-West, South-East, South-West}.

These relations can provide the basic directional information. An example of the spatial queries might look like:

*Object A is **East** of Object B.*

However, these types of queries could not distinguish the similar relationships. For example, there are three objects, that is, A, R and M shown in Figure 3.

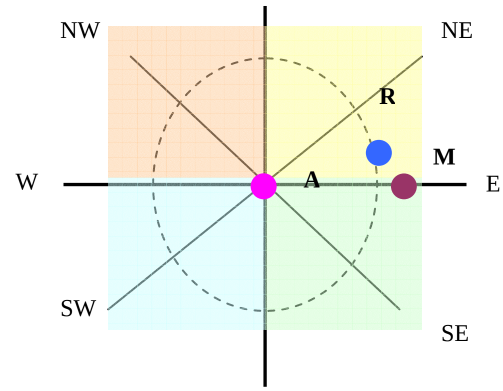


Figure 3. Three Objects A, R and M in 2-D

The basic spatial queries will provide the same directional relationships, that is,

*Object R is **East** of object A.*
*Object M is **East** of object A.*

In order to provide high accurate information such as object M is ***directly*** East of object A, object R is ***mostly*** East of object A, a fuzzy query approach for the direction is applied. We define the directional qualifier DQ as:

DQ={directly, mostly, somewhat, slightly, not}.

### 4.1. Membership Function for Directional Qualifier

Generally, a fuzzy set in a universe is characterized by a membership function $\mu$: U$\rightarrow$[0,1]. A term belongs to a set with a certain grade of membership. In order to perform fuzzy querying for direction, basic strategies are summarized as follows (see the details in [2-3]).

- Partition each object into sub-groups in eight directions based on the reference area (the common part of two objects) shown in Figure 4.
- Map each sub-group to a node, and assign two weights (area and node weights) to each node.
- Calculate two weighs to determine the special degree.

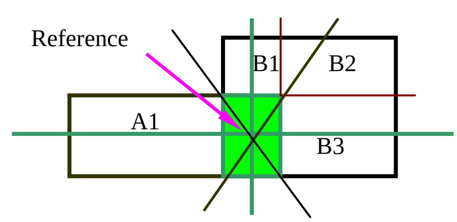


Figure 4. Partitioning Two Objects in 2-D

Where area weight can be calculated by
*AW=(area of sub-group) / (area of the entire object)*

and node weight can be obtained by

*NW=AW · (axis length) / (longest axis length).*

The resulting quantitative NW is mapped to a range that corresponds to a term known as linguistic qualifiers. The fuzzy membership function for the directional qualifiers is defined as follows:

$$\mu_{directly}(NW) = \begin{cases} 1.0 & \text{if } 0.95 \le NW \le 1.0 \\ 20\,(NW - 0.80)/3 & \text{if } 0.8 \le NW \le 0.95 \end{cases}$$

$$\mu_{mostly}(NW) = \begin{cases} 20\,(0.95 - NW)/3 & \text{if } 0.8 \le NW \le 0.95 \\ 1.0 & \text{if } 0.6 \le NW \le 0.80 \\ 10\,(NW - 0.5) & \text{if } 0.5 \le NW \le 0.6 \end{cases}$$

$$\mu_{somewhat}(NW) = \begin{cases} 10\,(0.6 - NW) & \text{if } 0.5 \le NW \le 0.6 \\ 1.0 & \text{if } 0.3 \le NW \le 0.5 \\ 10\,(NW - 0.2) & \text{if } 0.2 \le NW \le 0.3 \end{cases}$$

$$\mu_{slightly}(NW) = \begin{cases} 10\,(0.3 - NW) & \text{if } 0.2 \le NW \le 0.3 \\ 1.0 & \text{if } 0.02 \le NW \le 0.2 \\ 100(NW - 0.01) & \text{if } 0.01 \le NW \le 0.02 \end{cases}$$

$$\mu_{not}(NW) = \begin{cases} 100\,(0.02 - NW) & \text{if } 0.01 \le NW \le 0.02 \\ 1.0 & \text{if } 0.0 \le NW \le 0.01 \end{cases}$$

Figure 5 illustrates the primary terms of DQ. Each term represents a specific fuzzy set.

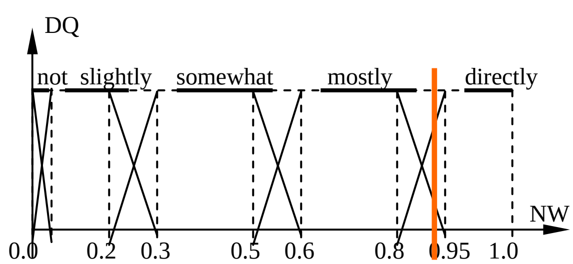


Figure 5. Membership Function

A common feature of the fuzzy sets is overlapping, that is, the qualifiers may be associated with two different terms at the intersect intervals. For instance, DQ may take ‘directly’ and ‘mostly’ at the same time. In such case, the fuzzy querying will give the following querying information:

*Object A is* ***directly*** *east of Object B*
*Object A is* ***mostly*** *east of Object B.*

Which querying information is reliable? This reveals uncertainty – the lack of adequate and correct information to make a decision. In this paper, the inferencing approach for fuzzy spatial querying presented in our previous work [2-3] has been adopted.

### 4.2. Certainty Factor

Uncertainty is an inevitable problem in above fuzzy inference. There are a number of methods to deal with uncertainty, such as decision theory, evidential and probability theory, confirmation theory and the dempster-shager theory, etc. By combing the fuzzy set and confirmation theory, we proposed a fuzzy inference approach under uncertainty, in which a certainty factor (CF) is used to evaluate the degree of the certainty. The main ideas are briefly described as follows.

**Case 1**. Single qualifier

This is a case in which the qualifier only takes one term at given interval, the grade of membership μ( ) can be used as a CF that represents the degree of belief. The query results will look like:

*Object A is* ***directly*** *east of Object B*
with CF=μ( DQ =directly).

**Case 2**. Multiple qualifiers

In the intersect intervals, the qualifier will be associated with two terms. For example, the query results look like:

*Object A is* ***directly*** *east of Object B*
*Object A is* ***mostly*** *east of Object B*

It is acceptable if we take the maximum grade of memberships, that is,

CF=max{ μ(directly), μ(mostly) }.

If μ(directly) is less than μ(mostly), the query result will be:

*Object A is* ***mostly*** *east of Object B*
with CF=μ( DQ =mostly).

## 5. IMPLEMENTATION

As a powerful database server, MySQL has been used to meet the web application requirements. It performs well on the majority of queries, and has a large support base for access from PHP, a powerful script language. In this paper, an information system for an apartment is implemented by using an open source database MySQL and PHP script language.

### 5.1. A Spatial Database

In order to support spatial queries, a spatial database is designed in terms of ER model, which includes geospatial locations of the objects [1]. Figure 6 shows the diagram of the spatial database.

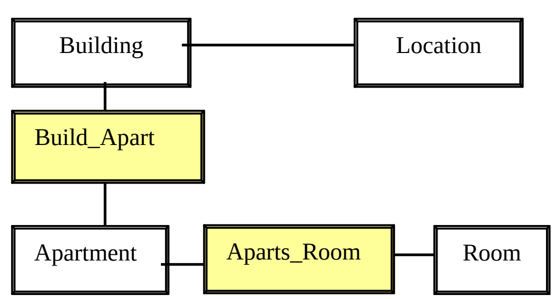


Figure 6. Spatial Database Diagram

The rules can be described as:

- Each building may have many apartments.
- Each apartment may have different types of rooms.

By means of MySQL, an apartment service database is created. Samples of table structure and data are provided below:

A. All tables in the database

```
mysql> show tables;
+-------------------+
| Tables_in_spatial |
+-------------------+
| Apartment         |
| Apart_Room        |
| Building          |
| Building_Apart    |
| Location          |
| Room              |
+-------------------+
6 rows in set (0.00 sec)
```

B. Table Structure

The table structure for each table is displayed as follows.

```
mysql> DESC Apartment;
+---------------+---------+------+-----+
| Field         | Type    | Null | Key |
+---------------+---------+------+-----+
| AptType       | char(2) |      | PRI |
| NumberofBedrm | char(2) | YES  |     |
| Size          | int(11) | YES  |     |
| Rent          | float   | YES  |     |
+---------------+---------+------+-----+
4 rows in set (0.00 sec)
```

```
mysql> DESC Apart_Room;
+--------------+------------+----+-----+
| Field        |Type        |Null| Key |
+--------------+------------+----+-----+
| AptType      |char(2)     |    | PRI |
| RoomType     |char(2)     |    | PRI |
| Room_Location|varchar(10) |YES |     |
+--------------+------------+----+-----+
3 rows in set (0.00 sec)
```

```
mysql> DESC Building;
+-------------+------------+------+-----+
| Field       | Type       | Null | Key |
+-------------+------------+------+-----+
| BID         | varchar(5) |      | PRI |
| BType       | varchar(15)| YES  |     |
| StreetName  | varchar(20)| YES  |     |
| Size        | int(11)    | YES  |     |
| NumApts     | char(2)    | YES  |     |
+-------------+------------+------+-----+
5 rows in set (0.00 sec)
```

```
mysql> DESC Building_Apart;
+----------+-----------+------+-----+
| Field    | Type      | Null | Key |
+----------+-----------+------+-----+
| BID      | varchar(5) |     | PRI |
| AptType  | char(2)   |      | PRI |
| Quantity | int(11)   | YES  |     |
+----------+-----------+------+-----+
3 rows in set (0.00 sec)
```

```
mysql> DESC Location;
+-----------+-----------+----+-----+
| Field     | Type      |Null| Key |
+-----------+-----------+----+-----+
| BID       | varchar(5) |   | PRI |
| BuildName | varchar(18)|YES |    |
| X1Long    | double    |YES |     |
| Y1Latitude| double    |YES |     |
| X2Long    | double    |YES |     |
| Y3Latitude| double    |YES |     |
+-----------+-----------+----+-----+
6 rows in set (0.00 sec)
```

```
mysql> DESC Room;
+------------+-----------+------+-----+
| Field      | Type      | Null | Key |
+------------+-----------+------+-----+
| RoomType   | char(2)   |      | PRI |
| Size       | int(11)   | YES  |     |
| RoomCloset | varchar(6) | YES |     |
| Bath       | char(3)   | YES  |     |
+------------+-----------+------+-----+
4 rows in set (0.00 sec)
```

BID: Building Identification Number
(X1Long, Y1Latitude) and (X2Long, Y2Latitude) represent the longitude and latitude coordinates of the building.

C. Sample data in the tables

```
mysql> select * from Building;
+-------+---------+---------------+-------+-------+
| BID   | BType   | StreetNam     | Size  |NumApts|
+-------+---------+---------------+-------+-------+
| 4001  | Dual    | J.Mitchell Dr. | 18420 | 18   |
| 4021  | Dual    | J.Mitchell Dr. | 24930 | 24   |
| 4020  | Dual    | J.Mitchell Dr. | 18420 | 18   |
| 20521 | Single  | Foundation Pl. | 21600 | 24   |
| 20511 | Dual    | Foundation Pl. | 18420 | 18   |
| 20501 | Single  | Foundation Pl. | 21600 | 24   |
| 4010  | ClubHouse| J.Mitchell Dr. |  5425 | 0   |
+-------+---------+---------------+-------+-------+
```

```
mysql> SELECT * FROM Room;
+----------+------+------------+------+
| RoomType | Size | RoomCloset | Bath |
+----------+------+------------+------+
| A        |  200 | Slide      | No   |
| B        |  200 | Door       | No   |
| C        |  300 | Slide      | Yes  |
+----------+------+------------+------+
3 rows in set (0.00 sec)
```

### 5.2. Query for Distance

Spatial queries require the geographic location of each object. The geographic coordinate of each building in the system is represented as (x, y), where x and y are coordinates of the object.

Given two objects A(x1, y1) and B(x2, y2), where x1 and y1 represent the longitude and latitude coordinate of the object A, and x2 and y2 represent the longitude and latitude coordinate of the object B.

The distance between A and B can be calculated as:

$$dist\ (A, B) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

The partial source code, which uses PHP to connect to MySQL database, and calculates the distance, is listed below.

```
<?php

$dest=$_REQUEST['dest'];
$src=$_REQUEST['source'];
$server= 'localhost';
$user= 'xxxxxx'; $pass= 'xxxxxx';
$mydb= 'xxxxx';
$table1='Location';
$table2='Building';
$connect = mysql_connect($server,
                       $user, $pass);

if(!$connect)
  die("cannot connect to $server using");

else
{
  mysql_select_db($mydb);
  $query1 = "select Longitude, Latitude
             from $table1
             where BuildName='$dest';";

  $result=mysql_query($query1, $connect)
  if($result)
  {
       $row=mysql_fetch_row($result);
       $x1=53.0*$row[0];
       $y1=69.1*$row[1];

       mysql_select_db($mydb);
       $query2="select Longitude,Latitude
               from $table2
               where BID='$src';";

       $result2=mysql_query($query2);
       if($result2)
       {
         $row2=mysql_fetch_row($result2);
          $x2=53.0*$row2[0];
          $y2=69.1*$row2[1];

         $dist=sqrt(($x1-$x2)*($x1-$x2)
                  +($y1-$y2)*($y1-$y2));
         $dist=round($dist, 3);

         print "The distance between
                  $src and $dest is
                  $dist miles<p> ";
     }
     else
        print "No result from
                  Building table. <p>";
   }
   else
    print "No result form $table1 <p>";
}
   mysql_close($connect);
?>
```

### 5.3. Fuzzy Spatial Query under Uncertainty

The querying is implemented by using PHP web programming language. Part of PHP code is given as follows:

```
//Receive the locations of two objects
//(xa1, ya1) (xa2, ya2) – Object A
//(xb1, yb1) (xb2, yb2) – Object B

//Calculate area weights for each node
//AW_e, AW_n, AW_s, AW_w, AW_sw, AW_se
//AW_ne, AW_nw
//Calculate node weights for each node
//NW_e, NW_n, NW_s, NW_w, NW_sw, NW_se
//NW_ne, NW_nw

//Find the weights and its direction
// For example, if the weights node is
// located in the East direction,
// the CF is calculate as follows

$direction=“East”;
if($nw>=0.95 && $nw<=1.0)
{
    $qualifier=“directly”;
    $CF=1.0;
}
else if($nw>=0.8 && $nw<0.95)
{   $CF1=20*($nw-0.80)/3;
    $CF2=20*(0.95-$nw)/3;
    if($CF1>$CF2)
    {  $qualifier=“directly”;
       $CF=$CF1;
    }
    else
    {  $qualifier=“mostly”;
       $CF=$CF2;
    }
}
else if($nw>=0.6 && $nw<0.8)
{   $qualifier=“mostly”;
    $CF=1.0;
}
```

```
else if($nw>=0.5 && $nw<0.6)
{   $CF1=10*($nw-0.5);
    $CF2=10*(0.6-$nw);
    if($CF1>$CF2)
    { $qualifier="mostly";
      $CF=$CF1;
    }
    else
    { $qualifier="somewhat";
      $CF=$CF2;
    }
}
else if($nw>=0.3 && $nw<0.5)
{   $qualifier="somewhat";
    $CF=1.0;
}
else if($nw>=0.2 && $nw<0.3)
{   $CF1=10*($nw-0.2);
    $CF2=10*(0.3-$nw);
    if($CF1>$CF2)
    { $qualifier="somewhat";
      $CF=$CF1;
    }
    else
    { $qualifier="slightly";
      $CF=$CF2;
    }
}
else if($nw>=0.02 && $nw<0.2)
{   $qualifier="slightly";
    $CF=1.0;
}
else if($nw>=0.01 && $nw<0.02)
{   $CF1=100*($nw-0.01);
    $CF2=100*(0.02-$nw);
    if($CF1>$CF2)
    { $qualifier="somewhat";
      $CF=$CF1;
    }
    else
    { $qualifier="not";
      $CF=$CF2;
    }
}
else if($nw>=0 && $nw<0.01)
{   $qualifier="not";
    $CF=1.0;
}
```

## 6. SAMPLE QUERIES

The system information retrieval is based on the spatial database queries. In this part, we provide samples to show general information queries and the fuzzy spatial query under uncertainty.

### 6.1. General Information Queries

The queries will provide the users with the basic information about the apartment. Figure 7 shows a map of the apartment. The user can select any building to view its information. After the user submits the request, the querying results will be displayed. Figure 8 shows an example of the basic querying results for the selected building.

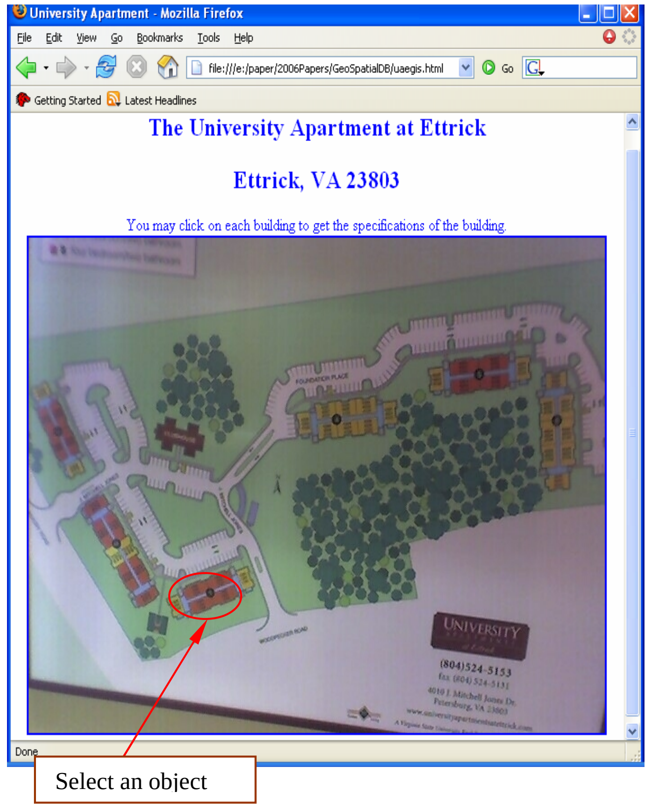


Figure 7. Apartment Map

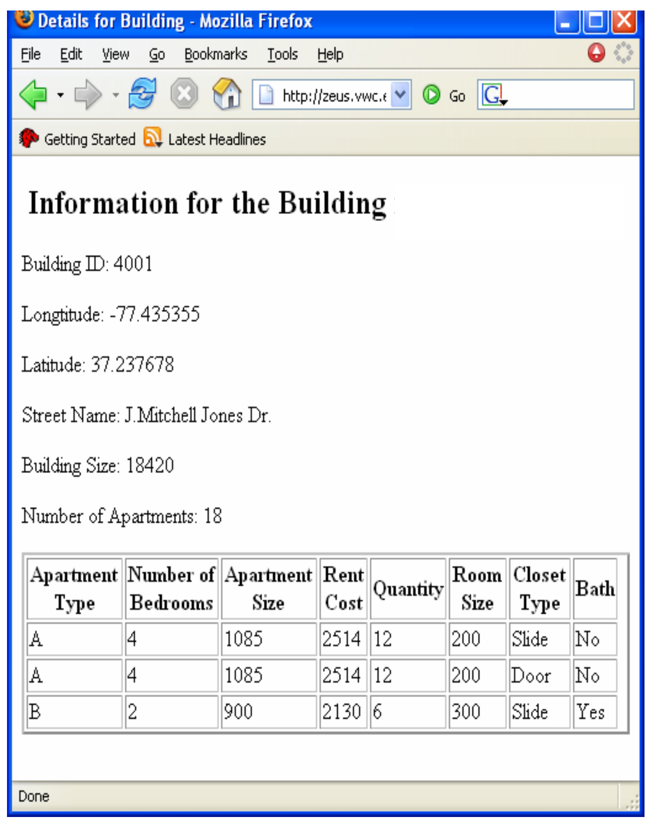


Figure 8. Basic Querying Results

### 6.2. Spatial Information Queries

By using the fuzzy spatial query model presented in section 4, the system can provide direction information.

The interface shown in Figure 9 provides a map of the main campus and the apartment.

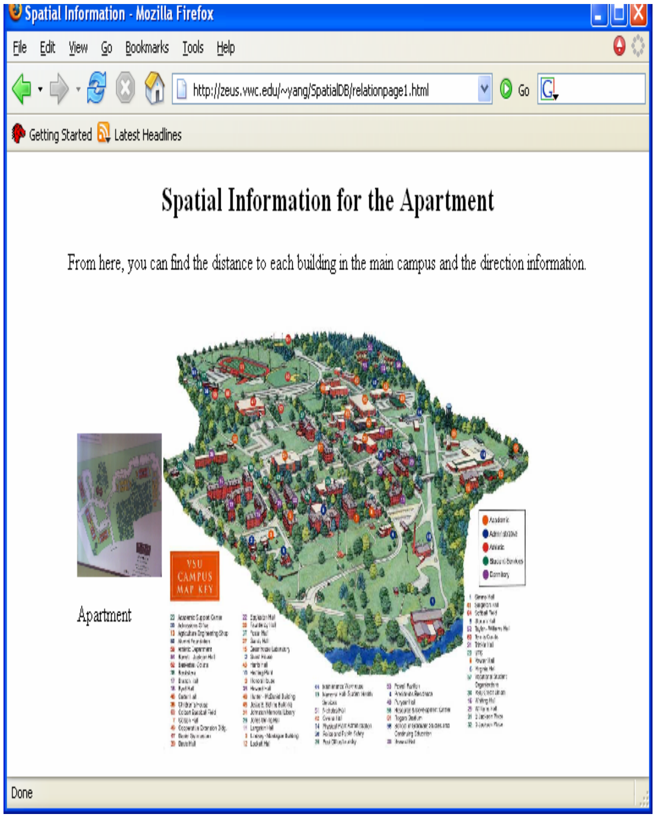


Figure 9. Main Campus Map

The source object is the apartment. The user can select the destination in the campus to perform a query. For example, if the library is clicked, the distance from apartment to the library is provided (see Figure 10).

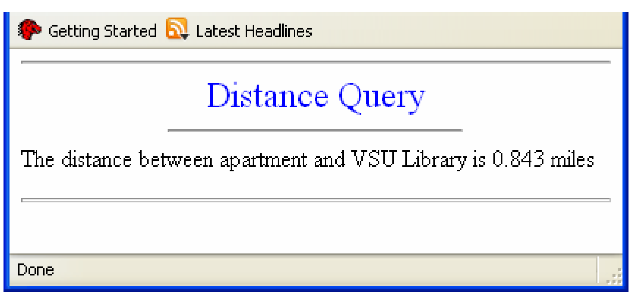


Figure 10. Distance Query Result

If the destination is Football Field, when the Football Field is clicked, direction information is shown in Figure 11.

More demonstration results show that the system can provide reliable spatial information for the web services of the apartment.

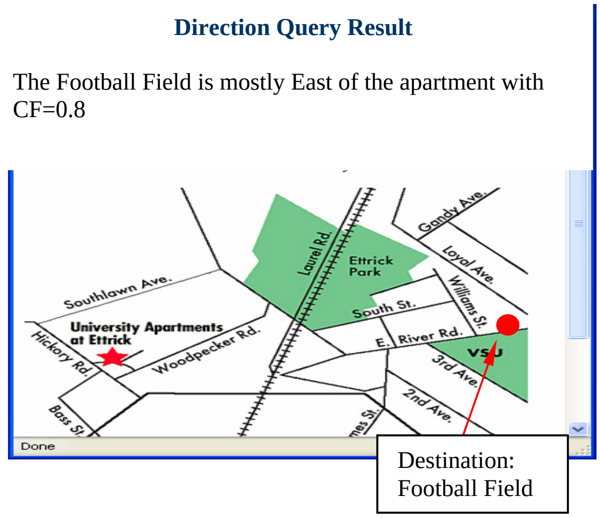


Figure 11. Direction Query Result

## 7. CONCLUSION

Regarding to an apartment web service application, we investigate a fuzzy spatial querying model to improve quality of online information retrieves. In a real world, fuzziness and uncertainty can occur simultaneously. To improve spatial querying accuracy, our research has been focused on a fuzzy inference under uncertainty. The reliability of querying information is judged by a certainty factor (CF). The improved fuzzy querying is very flexible, and the system can meet the goals of information retrieves for the apartment web services.

## 8. REFERENCES


[1]. H. H. Yang and G. L. Tranard, "An Online Information System for Apartment Services with Fuzzy Spatial Queries", **Proceedings of the 2006 international Conference on Information and Knowledge Engineering** (IKE'06), Las Vegas, Nevada, USA, CSREA Press 2006, pp.361-366.

[2]. M. A. Cobb, "Modeling Spatial Relationships within a Fuzzy Framework", **Journal of the American Society for Information Science**, Vol. 49, No. 3, 1998, pp.253-266.

[3]. H. H. Yang and M. A. Cobb, "Fuzzy Spatial Querying with Inexact Inference", **Proceedings of the North American Fuzzy Information Processing Society, NAFIPS-FLINT International Conference (IEEE)**, Tulane University, New Orleans, LA, 2002, pp.377-382.

[4]. D. A. Lash, **The Web Wizard's Guide to PHP**, Addison Wesley, 2003.

[5]. L. Atkinson, **Core PHP Programming**, Prentice Hall, 2004.

[6]. L. Welling and L. Thomson, **PHP and MySQL Web Development**, third edition, Sams Publishing, 2005

[7]. M. Goodchild and S. Gopal, **The accuracy of Spatial Databases**, Basingstoke, UK: Taylor and Francis, 1990.

[8]. S. Shekhar and S. Chawla, **Spatial Databases: A Tour**, Prentice Hall, 2003.