

An Online Information System for Apartment Services with Fuzzy Spatial Queries

Huiqing H. Yang
Department of Math & Computer Science
Virginia State University
Petersburg, VA, USA 23836
Phone: 1-804-524-6877
Fax: 1-804-524-5746
hyang@vsu.edu

Geoffrey L. Tranard
Department of Math & Computer Science
Virginia State University
Petersburg, VA, USA 23836
Phone: 1-203-385-1183
Fax: 1-804-524-5746
man_amazin@yahoo.com

Abstract

Spatial data lies at the heart of GIS research. The great strength of the spatial data is to query about the location and direction of spatial objects. This paper is to explore the possibilities and advantages of geo-spatial data to improve online apartment services. In order to support spatial querying in nature, a simple fuzzy query model is presented in the paper. An online spatial information system is developed by utilizing PHP technology with MySQL database. The system will provide visitors and residents with more accurate information about the apartment. Querying examples are also provided.

Keywords: *spatial database, fuzzy logic, spatial query, web application, PHP, MySQL*

1. Introduction

The University Apartment at Ettrick (UAE) is an apartment complex built to provide off campus housing to students. Originally, the apartment provides an official website for its properties, but there is no any spatial information included, for instance, distance or direction to a building in the main campus. The motivation of this research is to develop a spatial information system to improve and increase the amount of information for the users, residents or future residents.

The ability to perform queries on spatial data is

essential in an information retrieval system. An online spatial query requires technique support. The development of new technologies such as new remote sensing, diffusion and miniaturization of GPS receivers has provide an opportunity to improve decision-making based on geospatial information. Recently, 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 [1-3]. In this paper, an online information system is implemented by using an open source database MySQL and PHP script language.

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. A simple fuzzy model for direction query is introduced. The purpose is to improve the service quality and accuracy for the apartment.

The paper is organized as follows. Section 2 presents implementation tools for the spatial information system. A fuzzy model for handling direction queries is introduced in section 3. Section 4 briefly describes a spatial database design. Section 5 shows an online spatial information system. Querying samples and some

of source codes are provided in section 5.

2. Implementation Tools

In order to achieve a useful information system and make the querying results more reliable, PHP script language and MySQL are used in this research. The process of how PHP interacts with the MySQL database to generate dynamic querying pages is shown in Figure 1.

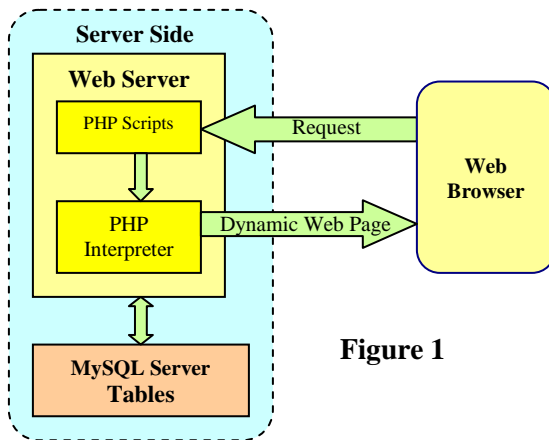


Figure 1

In the system, PHP scripts are mainly used to connect to the database and generate dynamic querying results. Spatial data of objects is stored in a MySQL database.

3. A Fuzzy Query Model

The directional relationships are commonly concerned in our daily life. Many efforts have been made in our previous work [4,5]. In this paper, we mainly deal with four cardinal directions and their refinements. The directional relationships are defined as follows:

$$D = \{E, S, W, N, NE, NW, SE, SW\}$$

E: East NE: North-East
 S: South NW: North-West
 N: North SE: South-East
 W: West SW: South-West

These relations can provide the basic directional information. An example of the spatial queries

might look like:

Math Department is **East** of the Apartment

However, these types of queries can only perform a low-level spatial query since they could not distinguish the similar relationships. For example, there are three objects A (Apartment), R (Register Office) and M (Mathematical Department) shown in the Figure 2.

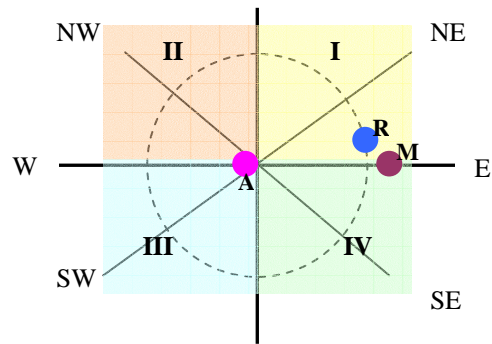


Figure 2

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

Object A and Object R:
 Register Office is **East** of the Apartment.

Object A and Object M:
 Math Department is **East** of the Apartment.

Since the spatial relationships depend on human interpretation, spatial queries should be related to fuzzy concepts in order to support queries of the nature. In order to provide high accurate information about direction, a fuzzy query model for the direction is presented in the paper. Some rules for fuzzy spatial queries are briefly described as follows.

Consider two objects $A(x_1, y_1)$ and $B(x_2, y_2)$. A tuple (x, y) represents the coordinate in the longitude and latitude respectively. We define the directional qualifier DQ as:

$$DQ = \{\text{directly, mostly, slightly}\}.$$

Four rule sets to provide fuzzy spatial querying are defined according to the Figure 2. The rule sets in *I* range and *IV* range are summarized as follows.

Rule I: $x_2 \geq x_1$ and $y_2 \geq y_1$

IF $k_1=0$ THEN "B is **directly** E of A"
 IF $0 < k_1 < 0.2$ THEN "B is **mostly** E of A"
 IF $0.2 \leq k_1 < 0.6$ THEN "B is **slightly** E of A"
 IF $0.6 \leq k_1 < 0.8$ THEN "B is **slightly** NE of A"
 IF $0.8 \leq k_1 < 1.0$ THEN "B is **mostly** NE of A"
 IF $k_1=1.0$ THEN "B is **directly** NE of A"
 IF $1.0 < k_1 < 1.3$ THEN "B is **mostly** NE of A"
 IF $1.3 \leq k_1 < 2.0$ THEN "B is **slightly** NE of A"
 IF $2.0 \leq k_1 < 5.0$ THEN "B is **slightly** N of A"
 IF $5.0 \leq k_1 < 100$ THEN "B is **mostly** N of A"
 IF $(x_2 - x_1) = 0$ THEN "B is **directly** N of A"

where $k_1 = (y_2 - y_1) / (x_2 - x_1)$.

Rule IV: $x_2 \geq x_1$ and $y_1 \geq y_2$

IF $k_4=0$ THEN "B is **directly** E of A"
 IF $0 < k_4 < 0.2$ THEN "B is **mostly** E of A"
 IF $0.2 \leq k_4 < 0.6$ THEN "B is **slightly** E of A"
 IF $0.6 \leq k_4 < 0.8$ THEN "B is **slightly** SE of A"
 IF $0.8 \leq k_4 < 1.0$ THEN "B is **mostly** SE of A"
 IF $k_4=1.0$ THEN "B is **directly** SE of A"
 IF $1.0 < k_4 < 1.3$ THEN "B is **mostly** SE of A"
 IF $1.3 \leq k_4 < 2.0$ THEN "B is **slightly** SE of A"
 IF $2.0 \leq k_4 < 5.0$ THEN "B is **slightly** S of A"
 IF $5.0 \leq k_4 < 100$ THEN "B is **mostly** S of A"
 IF $(x_2 - x_1) = 0$ THEN "B is **directly** S of A"

where $k_4 = |(y_2 - y_1) / (x_2 - x_1)|$.

In the same way, Rule *II*, and Rule *III* can be defined.

The fuzzy spatial querying will provide the directional relationships shown in Figure 2 as:

Math Department is **directly East** of the Apartment
 Register Office is **slightly East** of the Apartment

4. A Spatial Database

The most important feature of Geographic Information Systems is that spatial data are stored in a structured format referred to as spatial database [7]. The spatial queries imply querying

a spatial database based on relationships between particular objects in that database within a particular coordinate system. In order to support spatial queries, a spatial database which includes geospatial locations of the objects such as apartment, buildings in the main campus is designed in terms of the ER (Entity-Relationship) model.

The rules can be described as:

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

Some valid attribute values are listed as follows.

Table 1.

Apartment Type	Number of Bedrooms
A	4
B	2
C	1

Table 2.

Closet Type
Slide
Door

Since most GPS receivers are set to use latitude/longitude coordinates as their default setting [8], in this research, we use the latitude longitude coordinate system to describe a position on the surface of the earth. An ER diagram is shown in Figure 3.

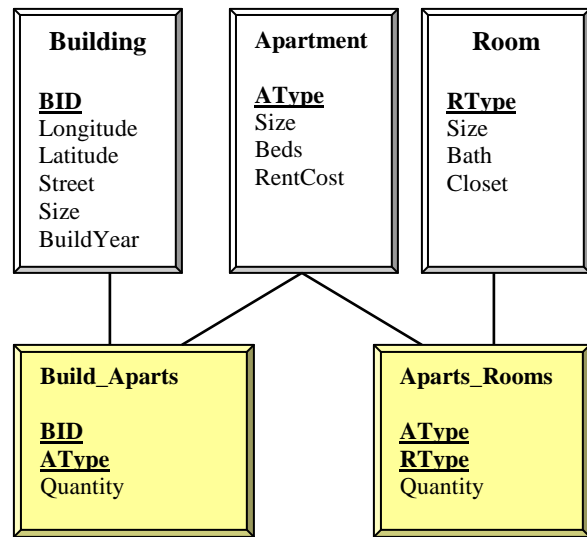


Figure 3. ER Diagram

BID: Building Identification Number
 AType: Apartment Type
 RType: Room Type

Table 3. The keys in the tables

Table Name	PK	FK	Reference Table
Building	BID		
Apartment	AType		
Room	RType		
Build_Aparts	BID AType	BID AType	Building Apartment
Aparts_Rooms	AType RType	AType RType	Apartment Room

PK: Primary Key
 FK: Foreign Key

The database is implemented by using MySQL.

5. An Online Information System

The system information retrieval is based on the spatial database queries. There are two specific types of queries provided in the paper.

a. General Information queries

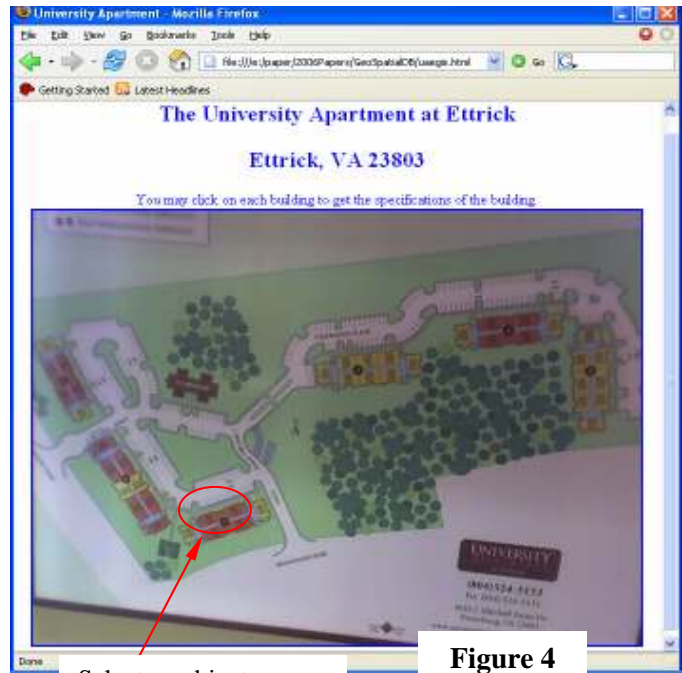
These kinds of queries will provide the users with the basic information about apartment.

An example of the basic querying

The interface for the apartment is shown in Figure 4. A user can select any building to view its information. After the users submit the request, the querying results will be displayed. Figure 5 shows an example of the basic apartment query.

b. Spatial information queries

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



Select an object

Figure 4

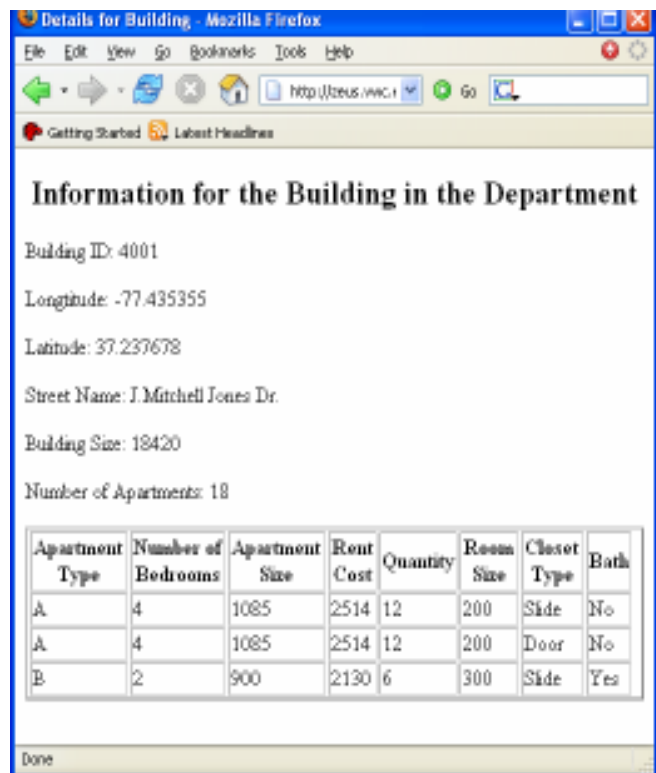


Figure 5

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:

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

By using the fuzzy spatial query mode presented in section 3, the system will provide direction information.

An example of the spatial querying

The interface shown in Figure 6 provides a map of the main campus and the apartment.

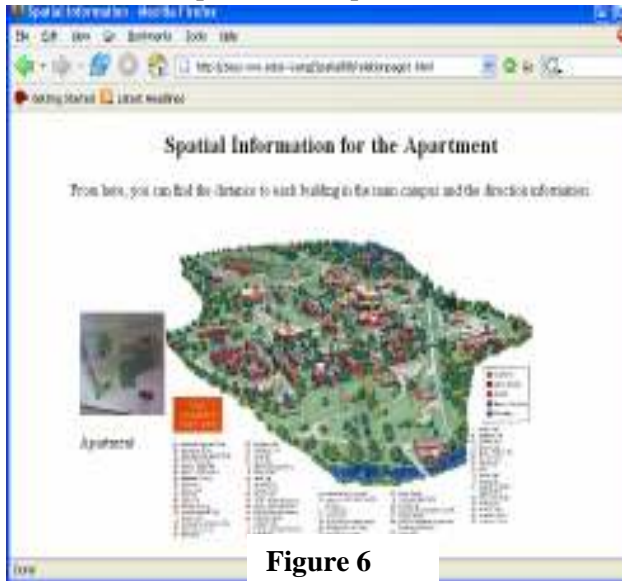


Figure 6

The source is the apartment. The user can select the destination in the campus. For example, if the Library is clicked, the distance from the apartment to the library is given in Figure 7.

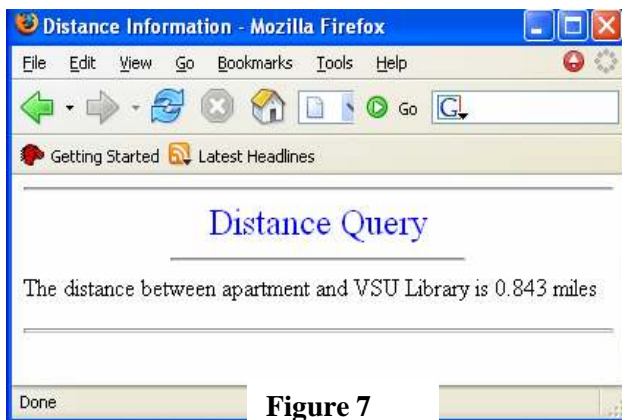


Figure 7

The partial source code which uses PHP to connect to MySQL database is listed in Table 4.

Table 4. Code for the distance querying

```

<?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);
?>

```

If the destination is Football Field, the direction information is shown in Figure 8.

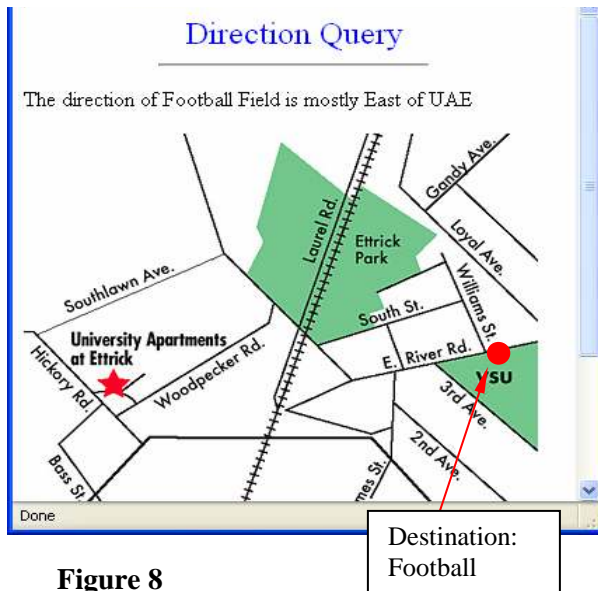


Figure 8

Table 5 shows partial PHP source code for fuzzy direction querying.

Table 5. Code for the direction querying

```
// (x1, y1) and (x2, y2) are known
if($x2!=$x1) $k=(y2-$y1)/($x2-$x1);

if(($x2>$x1)&&($y2>$y1))
{
  if(($y2-$y1)==0)
    print "The direction of $dest is
    directly E of $src <p>";
  else if (($k==1)
    print "The direction of $dest is
    directly NE of $src <p>";
  else if (($x2-$x1)==0)
    print "The direction of $dest is
    directly N of $src <p>";
  else if($k>0 && $k<0.2)
    print "The direction of $dest is
    mostly E of $src <p>";
  else if ((0.2<=$k && $k<0.6)
    print "The direction of $dest is
    slightly E of $src <p>";
  else if (0.6<=$k && $k<0.8)
    print "The direction of $dest is
    slightly NE of $src <p>";
  else if (0.8<=$k && $k<1.0)
    print "The direction of $dest is
    mostly NE of $src <p>";
  else if (1.0<$k && $k<1.3)
    print "The direction of $dest is
    mostly NE of $src <p>";
}
```

```
else if (1.3<=$k && $k<2)
  print "The direction of $dest is
  slightly NE of $src <p>";
else if (2<=$k && $k<5 )
  print "The direction of $dest is
  slightly N of $src <p>";
else if (5<=$k && $k<100)
  print "The direction of $dest is
  mostly N of $src <p>";
}
```

6. Conclusion

Based on an apartment web service application, this paper investigates a method which uses geospatial data to improve accuracy and quality of online information. Querying examples have shown the capabilities of a spatial database to support fuzzy directional queries. In the future, fuzzy queries may be improved by membership functions.

7. References

- [1]. David A. Lash, The Web Wizard's Guide to PHP, Addison Wesley, 2003
- [2]. Leon Atkinson, Core PHP Programming, Prentice Hall, 2004.
- [3]. Luke Welling and Laura Thomson, PHP and MySQL Web Development, Third Edition, Sams Publishing, 2005.
- [4]. H. Yang, M. Cobb, et al., "A Clips-Based Implementation for Querying Binary Spatial Relationships", Proceeding of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada. 2001.
- [5]. H. Yang, M. Cobb, "Fuzzy Spatial Querying with Inexact Inference", Proceeding of the North American Fuzzy Information Processing Society, NAFIPS-FLINT International Conference (IEEE), Tulane University, New Orleans LA, 2002.
- [6]. H. Yang, K. Outlaw, "Secure Consideration of PHP and MySQL in an Online Banking System", ISAS 2005 and CITSA 2005, P290-295, Orlando, Florida, 2005.
- [7]. Shashi Shekhar and Sanjay Chawla, "Spatial Databases: A Tour" Prentice Hall, 2003.
- [8]. Map Tools, <http://www.maptools.com/UsingLatLon/>