SuperMap Objects Java 6R Technology Documents

——Spatial Query

SuperMap Software Co., Ltd.

Beijing. China

Legal Statement

1. The copyright of this document is covered by SuperMap Software Co., Ltd. in accordance with the Copyright Law of the People's Republic of China and the Universal Copyright Convention. If, without the written permission of the company, any part of the document shall not in any way or any reason be used, copied, modified, transmitted, or bundled with other products to be used, sold, tort reserved.

2. "超图", "SuperMap", and  are the registered trademarks of SuperMap Software Co., Ltd., protected by the Copyright Law of the People's Republic of China. If, without the written permission of the company, the trademarks shall not in any way or any reason be used, copied, modified, transmitted, or bundled with other products to be used, sold, tort reserved.

3. This document represents no responsibilities of any supplier or agent. Without statement, SuperMap Software Co. Ltd. has right to do modifications to this document.

4. The copyright of trademarks mentioned in this document belongs to the corresponding companies. Without the written permission of these companies, the trademarks shall not in any way or any reason be used, copied, modified, or transmitted.

5. The concerned software products and the updated products hereinafter in this document are developed and sold by SuperMap Software Co., Ltd.

Hereby declare

SuperMap Software Co.. Ltd.:

Add: 7/F Tower B, Technology Fortune Center, No. 8 Xueqing Road,

Haidian District, Beijing, 100192, P. R. China

Tel: +86-10-82736655-4170

Fax: +86-10-82734630

HomePage: www.supermap.com

Sales: request@supermap.com

Tech Support: globalsupport@supermap.com

SuperMap Software welcomes all advices and suggestions from you.

Content

1

Definition of Spatial Query

Spatial query produces outcome according to the spatial relationships between the geometry objects. The filter conditions of spatial query could be attributes as well as spatial locations. When you want to know how many five-star hotels around the Olympic Stadiums within 5 kilometers, the conditions should use both of the class field and a 5-kilometer buffer, employing the Contain to query the hotel layer to get the desired result. The following sections will introduce the rich spatial query methods offered by the SuperMap to users for flexible applications.

SuperMap Objects Java provides eight basic operators of spatial query: Identity, Cross, Overlap, Disjoint, Touch, Intersect, Contain, Within. All the operators cover the spatial query operators in the OGC standards1.

Spatial query involves three elements, which are query object A, object B on the query layer, and the outcome record sets. In the diagrams we provided at here, the green color represents A, the black color represents B, and the red color represents the outcome objects.

Note: the geometry objects of points, lines, regions, as well as text, parametric objects are all permitted in spatial queries. We are going to illustrate the points, lines and regions at here only. The spatial relationship judged by the text objects is decided by the external rectangular. For the composite objects, there are operators to support, such as Intersect, Disjoint, Contain (must be a region as the query object), within (must be a region query layer).

---

[1] OGC, short for The Open Geospatial Consortium, Inc., settled a range of standards to share the geographic information, including SLD, WSC, WCS, GML, WMS, WFS and so on.

2

Concepts of Geometry Objects

The geometry objects of points, lines, and regions have explicit Envelop, Boundary, Interior and Exterior. Please look at the table below for their definitions. Comparing between the spatial locations of geometry objects means to compare the Envelop, Boundary, Interior and Exterior relationships. Consequently, understanding these definitions is useful to understand the operators in spatial query.

Table 1 Geometry Objects Definitions

Geometry Objects	Envelop	Boundary	Interior	Exterior
Points	none	none	itself	region excluding the points
Lines	the smallest external rectangle of the line object	endpoints of the line object	the line without its endpoints	the area excluding the line object
region	the smallest external rectangle of the region object	the intersection set of the region object control boundaries	the region object without its boundary	the areas outsides the region object

SuperMap Objects Java applies the 9-intersection model of the OGC to define the spatial relationship between the geometry objects with various dimensions and types. The model uses I(a),

B(a), E(a) to represent the Interior set, Boundary set, and Exterior set of the geometry object a. The intersection operation of the Interior sets, Boundary sets, and Exterior sets of two geometry objects will produce the results in the following table. The 9-intersection model is stored in the computer as a three dimensional matrix, and suitable to nearly all the geometry objects.

Table 2 The 9-Intersection Model Table

	Interior I(b)	**Boundary B(b)**	**Exterior E(b)**
InteriorI(a)	dim(I(a) ∩ I(b))	dim(I(a) ∩ B(b))	dim(I(a) ∩ E(b))
BoundaryB(a)	dim(B(a) ∩ I(b))	dim(B(a) ∩ B(b))	dim(B(a) ∩ E(b))
EE(a)	dim(E(a) ∩ I(b))	dim(E(a) ∩ B(b))	dim(E(a) ∩ E(b))

The function dim() returns the dimensional number of the intersection operation outcome. The -1, 0, 1, 2 will be the value of the 9-intersection model according to the principles below:

(1) T, if dim()=0, 1, 2, the intersection exists

(2) F, if dim()=-1, the intersection does not exist

(3) *, without considering of the stated intersection, no matter if there is a intersection set.

Take the region partly overlay as an example to illustrate the 9-intersection model. Please look at the diagram below, region a and region b partly overlay with each other. The string data array Matrix[9] is going to store the values of the 9-intersection model:

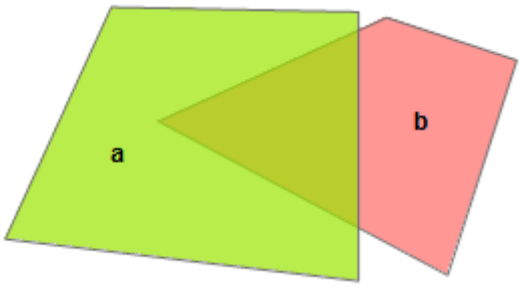
1. dim(I(a)∩I(b))=2, then Matrix(0)=T. If two lines cross to each other, then dim(I(a)∩ I(b))=1; If two points overlap to each other, then dim(I(a)∩I(b))=0.
2. dim(I(a)∩B(b))=1, dim(B(a)∩I(b))=1, if two regions overlap, we don't care about their boundary, interior intersection sets of a and b, hence Matrix(1)= Matrix(3)=*;
3. dim(I(a)∩E(b))=2, then Matrix(2)=T; 9-intersection model

4. dim(B(a)∩B(b))=0, if two regions overlap, we don't care about their boundary intersection sets of a and b, hence Matrix(4)=*;

5. dim(B(a)∩E(b))=1, dim(E(a)∩B(b))=1, if two regions overlap, we don't care about their boundary, exterior intersection sets of a and b, hence Matrix(5)= Matrix(7)=*;

6. dim(E(a)∩I(b))=2, hence, in the 9-intersection model, Matrix(6)=T;

7. dim(E(a)∩E(b))=2, if two regions overlap, we don't care about intersection set between the interior of a and boundary of b, hence, Matrix(8)=*.



	Interior I(b)	Bounder B(b)	Exterior E(b)
Interior I(a)	T	*	T
Bounder B(a)	*	*	*
Exterior E(a)	T	*	*

Diagram 1 "Two regions partly overlap" and its 9-intersection model

3

Operators

This section introduces the definitions of the eight basic operators and illustrates their relationships.

3.1 Identity

Definition: returns the object from the query layer that overlaps completely with the query object, without necessary of totally overlapped sampling points overlapped. As the diagram displayed: ploylines A and B are considered as Identity even the coordinate control points are different.

Diagram 2 The objects of Identity with different sampling points

Illustration of their basic relationship: the type of A and B must be the same, which could be points, lines and regions. The interior intersection of A and B is not null, and the intersection of A's interior and B's exterior is null, and intersection of A's boundary and B's exterior is null, and intersection of A's exterior and B's interior is null, and intersection of A's exterior and B's boundary is null. The 9-intersection model of Identity is displayed in the following table. The right diagram gives an example of region query.

Table 3 The objects of Identity with different sampling points

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	T	*	F
	boundary	*	*	F
	Exterior	F	F	*

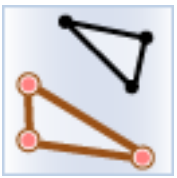


Diagram 3 A Region Identity a Region

3.2 Cross

Definitions: returns all the objects (lines, regions) of the query layer that intersect with query object (lines).

Cross includes two situations as below:

（1） The interior intersection of A and B is not null, and the intersection of A's interior and B's exterior is not null. The lines and regions are suitable to this relationship.

Table 4 Cross The 9-Intersection Model_1 of Cross

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	T	*	T
	boundary	*	*	*
	Exterior	*	*	*

Diagram 4 Lines Cross Regions

（2） The interior intersection of A and B is not null, and must be zero dimensional. This relationship is suitable to line cross line.

Table 5 The 9-Intersection Model_2 of Cross

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	0	*	*
	boundary	*	*	*
	Exterior	*	*	*

Diagram 5 Lines Cross lines

3.3 Contain

Definition: returns object contained by the queried object in the query layer. Note: objects has intersection on the boundary are accepted, which is the same with Within.

Illustration of their relationships: the interior intersection of A and B is not null, and the intersection of A's exterior and B's interior is null. A contains B, hence the number of A's dimensional should be higher than B's dimensional.

Table 6 The 9-Intersection Model of Contain

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	T	*	*
	boundary	*	*	*
	Exterior	F	*	*

Diagram 6 Regions Contain Regions

3.4 Within

Definition: returns object of the query map containing the query object. If a region is returned, which must contains the query object, including on the boundary. If a line is returned, which must

contains the query object totally. If a point is returned, it must be on the query object, which is the inverse operator of Contain. Object type: query object: points, lines, regions: queried object: points, lines, regions.

Table 7 The 9-Intersection Model of Contained

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	T	*	F
	boundary	*	*	*
	Exterior	*	*	*

Diagram 7 Lines Within Regions

3.5 Overlap

Definition: returns the object from query layer that partly overlapped with query object. The two objects must have the same dimensions, as well as the intersection.

Illustration of the relationship: query object A overlaps with queried object B; A and B have the same dimension; the interior intersection C of A and B is not null, A's exterior intersects with B's interior, B's exterior intersects with A's interior; C is the same type with A.

Table 8 The 9-Intersection Model- Overlap

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	T	*	T
	boundary	*	*	*
	Exterior	T	*	*

Diagram 8 Line OverLap line

3.6 Disjoint

Definition: returns the objects of the query layer that is disjoint with query object.

Illustration of the relationship: the objects could be points, lines, regions, and the interior intersection of A and B is null, and the intersection of A's interior and B's boundary is null, and the intersection of A's boundary and B's interior is null, and the boundary intersection of A and B is null.

Table 9 The 9-Intersection Model Disjoint

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	F	F	*
	boundary	F	F	*
	Exterior	*	*	*

Diagram 9 Regions Disjoint Regions

3.7 Touch

Definition: returns objects from the query layer that touch the boundary of the query boundary.

Touch includes three situations as below:

（1） The interior intersection of A and B is null, and the intersection of A's interior and B's boundary is not null.

Table 10 The 9-Intersection Model_1 Touch

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	F	T	*
	boundary	*	*	*
	Exterior	*	*	*

Diagram 10 Lines Touch Regions

（2） The interior intersection of A and B is null, and the intersection of A's boundary and B's intersection is not null.

Table 11 The 9-Intersection Model_2 Touch

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	F	*	*
	boundary	T	*	*
	Exterior	*	*	*

Diagram 11 Regions Touch Lines

（3） The interior intersection of A and B is null, and the boundary intersection of A and B is not null.

Table 12 The 9-Intersection Model_3 Touch

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	F	*	*
	boundary	*	T	*
	Exterior	*	*	*

Diagram 12 Regions Touch Regions

3.8 Intersect

Definition: returns all objects from the query layer which is not null in the intersection with the query objects.

Illustration of the relationship: the three situations list below are intersection:

（1） The interior intersection of A and B is not null.

Table 13 The 9-Intersection Model_1 Intersect

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	T	*	*
	boundary	*	*	*
	Exterior	*	*	*

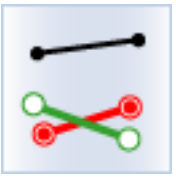


Diagram 13 Lines Intersect Lines

（2） The intersection of A's interior and B's boundary is not null.

Table 14 The 9-Intersection Model_2 Intersect

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	*	T	*
	boundary	*	*	*
	Exterior	*	*	*

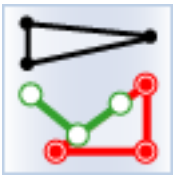


Diagram 14 Lines Intersect Regions

（3） The intersection of A's boundary and B's interior is not null.

Table 15 The 9-Intersection Model_3 Intersect

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	*	*	*
	boundary	T	*	*
	Exterior	*	*	*

Diagram 15 Regions Intersect Regions

（4） The boundary intersection of A and B is not null.

Table 16 The 9-Intersection Model_4 Intersect

	B			
A		**Interior**	**boundary**	**Exterior**
	Interior	*	*	*
	boundary	*	T	*
	Exterior	*	*	*

Diagram 16 Regions Intersect Regions