

# Research of Extensions of World Wind Java Spatial Retrieve Capability by ARC SDE Geometric Analysis Function

Xiaowei Xu<sup>1,2</sup>, Jinlei Wang<sup>1,2</sup>, Li Zhang<sup>1,2</sup>, Peng Yang<sup>1,2</sup>, Yang Lin<sup>1,2</sup>, Zhen Zhao<sup>3</sup>

<sup>1</sup>North China Sea Branch of the State Oceanic Administration, Qingdao Shandong

<sup>2</sup>Marine Ecological Environment and Disaster Prevention and Reduction Laboratory of Shandong Province, Qingdao Shandong

<sup>3</sup>School of Information Science & Technology, Qingdao University of Science & Technology, Qingdao Shandong  
Email: xuxiaowei@bhj.gov.cn

Received: Jul. 25<sup>th</sup>, 2017; accepted: Aug. 6<sup>th</sup>, 2017; published: Aug. 14<sup>th</sup>, 2017

---

## Abstract

World Wind Java is an excellent open-source GIS development platform. Similar to other 3-dimensional geographic information platform, World Wind Java faces the same question for element picking up and spatial analysis released in WMS. Spatial data using the ARC SDE provides the retrieval function expansion of the World Wind Java data retrieval function, and practical application in the "North China Sea Integrated Ocean Management Information System" system, and verifies the effectiveness of the proposed method.

## Keywords

World Wind Java, SDE, Vector Data, Spatial Retrieve

---

# 利用ARC SDE空间分析函数扩展World Wind Java空间检索能力的方法研究

徐晓玮<sup>1,2</sup>, 王金磊<sup>1,2</sup>, 张莉<sup>1,2</sup>, 杨鹏<sup>1,2</sup>, 林杨<sup>1,2</sup>, 赵振<sup>3</sup>

<sup>1</sup>国家海洋局北海信息中心, 山东 青岛

<sup>2</sup>山东省海洋生态环境与防灾减灾重点实验室, 山东 青岛

<sup>3</sup>青岛科技大学信息科学技术学院, 山东 青岛

Email: xuxiaowei@bhj.gov.cn

收稿日期: 2017年7月25日; 录用日期: 2017年8月6日; 发布日期: 2017年8月14日

## 摘要

World Wind Java是一个优秀的开源代码三维地理信息开发平台。与其他三维地理信息平台类似, World Wind Java同样面临以WMS格式发布的海量矢量数据拾取与空间检索需求。本文利用ARC SDE提供的空间数据检索函数开发的服务接口扩展实现了World Wind Java的空间数据检索功能,并在“北海区海洋综合信息一张图”系统中进行了实际应用,验证了方法的有效性。

## 关键词

World Wind Java, 空间数据引擎, 矢量数据, 空间检索

Copyright © 2017 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

World Wind 是由 NASA 的科研人员开发的开放源代码三维地图平台。它可以将 Landsat 7、MODIS (Moderate-Resolution Imaging Spectroradiometer, 中分辨率成像光谱仪)、SRTM(Shuttle Radar Topography Mission, 航天飞机雷达地形测图任务)等多颗卫星的数据及其它 WMS(Web 地图位置服务, Web Map Service)服务商提供的图像通过一个三维的地球模型展现[1] [2]。World Wind 软件包括 Java SDK(Java 开发版)和.Net SDK (.Net 框架开发版) 2 个版本,其中 Java SDK 即可以用于开发桌面版系统,也可以开发成为 Java applet 嵌入网页中运行[3]。ARC SDE (空间数据库引擎, Spatial Database Engine)是 ARC GIS(地理信息系统, Geographic Information System)的空间数据引擎,提供空间和非空间数据进行高效率操作的数据库服务,支持多用户同时并发访问和操作同一数据。

本文在 World Wind Java 三维开源平台的基础上,通过 ARC SDE 的空间操作函数扩展了平台对 WMS 图层数据源的空间检索能力,构建具备点、线、复杂面和缓冲查询能力的 WebService 接口,并在“北海区海洋综合信息一张图”系统中得到了应用,证明此类方法有效的解决了海量数据表现与空间检索的矛盾问题。

## 2. 需求分析和设计

### 2.1. 需求分析

World Wind Java 使用 Applet 和 Application 两种框架,支持多种文件格式和网络数据源,可以满足构建实用的三维地理信息应用系统的需求。World Wind Java 支持 KML/KMZ、SHP、GPX 等多种格式的文件数据源,它使用 UserSurfaceIcon 类表示点图元,使用 UserPolyLines 和 UserPolygons 类分别表示线和面图元,这些图元都可以通过实现鼠标和键盘接口响应用户操作,使程序具有良好的地图空间交互能力[4]。

World Wind Java 对以上三类图元提供了巧妙的拾取操作支持能力。由于三维地理信息系统在表现过程中存在一个三维向二维的投影,要从用户点击的屏幕点(像素位置)获取图元是存在一定困难的。屏幕是

一个二维的画布，而用户真正要提取的实际上是三维球体上的位置，这个二维像素坐标与三维地理坐标并不存在明确的一对一的关系。如果先从二维像素坐标转换成三维的地理坐标，再通过复杂的空间逻辑运算决定与哪些图元相交的话，效率是非常低的。为了克服这个问题，World Wind Java 非常巧妙地使用了窗口颜色缓冲相交检验的方法来处理图元拾取过程中的判断，使拾取操作的处理变得简单和可行。

World Wind Java 在拾取(Pick)操作时，World Wind 调用每一个图层的 AbstractLayer 接口的 doPick 方法，就像它绘图时要逐个调用 DrawContext 方法一样。World Wind Java 在调用这两种接口的顺序恰好相反，绘图时从最底层开始，而拾取时从最表层开始。当应用程序发出指定点(区域)的拾取消息时，各图层顺序负责根据指定区域检查是否有相交对象，并提交对象描述，在全部图层循环后，应用程序就获得了指定点的图元清单。在这一过程中，控制器首先将当前的选取点(区域)对应的范围颜色缓存清除，这个操作是在后台缓存中进行的，所以并不影响用户的视觉效果。然后控制器遍历每一个图层，每个图层用特定的拾取标志色进行绘制，这个过程同样是在后台进行。图层中每个可拾取(Pickable)的图元使用不同的标识颜色，通过读取指定区域的颜色缓冲区的值，就可以判断出该图元是否应当被拾取。所有图层循环过后，应用程序就可以获得一份拾取图元的清单，这种方法避免了复杂的空间逻辑运算，效率非常高。

这种空间要素拾取的解决方法存在一个缺陷，对 WMS 服务提供的栅格数据的数据源支持能力不足。针对此问题，目前主流方法是采用在栅格图上直接绘制矢量图形的办法来解决，但是该方法无法使用大数据，且不适合本文应用平台技术体系。World Wind Java 对 WMS 等空间栅格数据格式支持非常好，可以兼容大部分 WMS 服务器发布的空间影像数据服务。当我们面临大量的遥感影像数据和几十万节点乃至几百万节点的空间图元数据时，使用矢量数据源通过 WMS 发布是保证系统效率的最好办法。这种数据表现方式面临着一个问题：所有图元都被栅格化成了空间影像，World Wind Java 无法再通过“渲染”的方法区分用户点击的是哪一个图元，也就失去了高效实现空间检索的一个重要途径。

国家海洋局北海信息中心在构建“北海区海洋综合信息一张图”系统时面临海量数据的检索和操作问题，由于确权用海、海洋功能区划、道路、居民点等图层空间数据量非常大，系统使用了 WMS 图层来发布这些空间要素。同时针对用户提出的空间要素的地图检索功能，系统通过 ARC SDE 提供函数实现了这一需求，从而通过两种技术的结合，为用户提供了优良的用户体验。

## 2.2. 解决方法

“北海区海洋综合信息一张图”系统使用 oracle 数据库存储数据，并使用 ARC SDE for Oracle 提供了 ST\_Geometry 类型来存储和处理几何数据，st\_geometry 是一种遵循 ISO(国际标准化组织，International Organization for Standardization)和 OGC(开放地理信息系统协会，Open GIS Consortium)规范的，可以通过 SQL(结构化查询语言，Structured Query Language)直接读取的空间信息存储类型。采用这种存储方式能够更好地利用 oracle 的资源并兼容 oracle 的特征。使用 st\_geometry 存储空间数据，可以把业务数据和空间数据存储到一张表中，因此可以很方便的在业务数据中增加空间数据。

如前所述，可以通过监听 World Wind Java 在屏幕上的鼠标操作，系统将二维屏幕坐标通过 World Wind Java 画布提供的方法转为三维坐标，并判断三维坐标的合法性。当用户点击的是一个合法的三维坐标时，通过 ARC SDE 的空间检索函数就可以获取保存在 ARC SDE 数据表中的记录集。使用这种方法的优点在于：首先，它扩展了 World Wind Java 的空间数据检索能力，使得我们对海量数据进行表现和操作时，在充分发挥 WMS 服务的用户操作快速响应能力的同时提供高效的空间数据检索能力；其次，利用这种方法可以将多个数据表进行“UNION”或“UNION ALL”检索并在后台形成一个数据集，从而将一些业务逻辑的调整从前端移到服务器端，符合分层开发的设计趋势，降低了系统的耦合度。

### 3. 功能实现

#### 3.1. ARC SDE 空间检索函数

st\_geometry 的存储格式是由 ESRI(美国环境系统研究所公司, Environmental Systems Research Institute)提出的一种新空间数据存储标准, 在空间数据检索效率、复杂空间数据的支持方面比 oracle spatial 表现更专业。要通过 SQL 函数访问存储在 Oracle 中的 st\_geometry 类型的数据, 需要通过扩展 Oracle 的 external procedure agent 或者 extproc 来实现[5]。

ARC SDE 提供了一系列空间检索、数据操作和空间运算函数, 本文仅关注实现 World Wind Java 空间数据检索操作相关部分方法。在空间检索过程中, 系统最关注的是用户的点击位置检索、图元缓冲检索、用户指定位置和半径的范围检索以及用户指定区域的交叉检索(如矩形、多边形区域), 相关主要函数如表 1 所示。

根据不同用户的操作要求, 使用 st\_intersects, 或利用 st\_geometry 构建空间图元作为指定的检索区域, 也可以在前两者的基础上再使用 st\_buffer 进行缓冲查询。ARC SDE 提供了相当多的空间数据操作方法, 但是只要掌握这几个方法就完全可以满足我们利用 ARC SDE 扩展 World Wind Java 空间检索能力的需要。

#### 3.2. 空间检索效率分析

在实际应用中, 为了提高数据检索效率, 我们要针对不同的用户需求优化检索方法, 尤其是面对海量空间数据操作时。尽管用户的空间检索包含了点、线、面和缓冲检索, 归纳起来其实只有两种情况:

- (1) 用户指定的空间范围(线、面, 含缓冲面)是否与数据表空间记录有交叉;
- (2) 用户指定的空间点是否在数据表空间记录的范围内;

前者只能使用使用 st\_intersects 函数, 后者还可以使用 st\_within 函数和 st\_contains 函数, 三者的使用方法与效率存在一些差距。st\_intersects 函数和 st\_within 函数是通过空间索引进行数据检索比对, 效率相对较高, 而 st\_contains 函数则是利用全表扫描方式进行数据检索比对, 效率相对较低。但 st\_intersects 函数执行时会将对后一个传入参数先进行全表扫描后再利用空间索引进行比对, 所以应用该函数时应当将数据量小的参数作为后面的传入参数以提高检索效率。下面进行一些比较, 假设在 ARC SDE 的空间数据库中已经存在空间数据类型为面图元的数据表 TABLEA, 记录约 25 万左右, 表 2 分别列出了针对点、线、面的语句和运行时间。

由表 2 可以看到, 实现相同功能的不同检索语句的执行时间效率差距巨大, 有些空间检索甚至可以达到近 2000 倍, 最少的也达到了 160 倍左右。oracle 在执行 SQL 时, 数据表是从后向前按顺序检索的。

**Table 1.** Spatial retrieve function of ARC SDE

**表 1.** ARC SDE 空间检索相关函数表

序号	方法—参数	方法说明
1.	sde.st_intersects (g1, g2)	如果两个 ST_Geometries 的交集生成空集, 则 ST_Intersects 返回 1 或 t, 反之返回 0 或 f
2.	Sde.st_within (g1, g2) Sde.st_contains (g1, g2)	这是一对互补方法, 如果 g1 包含于(st_within)g2 或 g1 包含 (st_contains)g2 则返回 1 或 t, 反之返回 0 或 f
3.	sde.st_buffer (g1, distance)	ST_Buffer 获取几何对象和距离, 然后返回表示围绕源对象的缓冲区的几何对象
4.	sde.st_geometry (x, y, z, m, srid) sde.st_geometry (x, y, z, m, radius, numPts, srid)	st_geometry 通过文本表示构造几何, 利用 st_geometry 可以创建所有的 ARC SDE 支持的空间图元

**Table 2.** Efficiency of spatial retrieve function of ARC SDE  
**表 2.** ARC SDE 空间检索函数效率表

类别	SQL 语句	查询时间(秒)
点	SELECT COUNT(ID) FROM SDE.TABLEA A WHERE SDE.ST_INTERSECTS (A.SHAPE, SDE.ST_POINT (123.36, 39.15, 11)) = 1	0.062
	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.ST_INTERSECTS (SDE.ST_POINT (123.36, 39.15, 11), A.SHAPE) = 1	122
	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.WITHIN (SDE.ST_POINT (123.36, 39.15,11), A.SHAPE) = 1	0.062
	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.ST_CONTAINS (A.SHAPE, SDE.ST_POINT (123.36, 39.15, 11)) = 1	113
线	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.ST_INTERSECTS (A.SHAPE, SDE.ST_LINEFROMTEXT ('LINESTRING (123.36 39.15, 122.53 37.55)', 11)) = 1	1.201
	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.ST_INTERSECTS (SDE.ST_LINEFROMTEXT ('LINESTRING (123.36 39.15, 122.53 37.55)', 11), A.SHAPE) = 1	190.7
缓冲区 (面)	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.ST_INTERSECTS (A.SHAPE, SDE.ST_BUFFER (SDE.ST_POINT (123.36, 39.15, 11), 0.1)) = 1	0.265
	SELECT COUNT (ID) FROM SDE.TABLEA A WHERE SDE.ST_INTERSECTS (SDE.ST_BUFFER (SDE.ST_POINT (123.36, 39.15, 11), 0.1), A.SHAPE) = 1	329.6

执行表 2 的第一条 SQL 语句时, oracle 先生成一个 point 对象, 然后与 TABLEA 中的每条记录的索引对比; 执行表 2 的第二条 SQL 语句时, oracle 会从 TABLEA 中取出每个图元, 再与生成的 POINT 对象进行空间运行对比, 这样在执行一个对比 25 万条记录的 SQL 语句时, 就要生成 25 万次对象, 而且不能充分利用 ARC SDE 的索引进行加速。

### 3.3. 发布空间数据检索的 Web Service 服务

为了提高集成系统模块的复用性, 降低系统间各模块的耦合程度, 我们在 Apache Tomcat 平台上使用 Apache CXF(Apache Celtix + XFire)对刚才描述的 ARC SDE 的存储过程进行了封装。CXF 建立的 Web Service 分为接口部分和实现部分[6], 系统对确权用海项目封装的 Web Service 如图 1 所示。图 1 中的 ISDESpatialElementsFactory.java 是 Web Service 实现部分, 用户通过调用该方法获取需要检索的经纬度坐标列表, SDESpatialElementsFactoryImpl.java 是 Web Sevice 接口部分, 此接口首先利用 st\_buffer 方法获取用户选择区域, 再利用 st\_within 方法获取与用户选择区域相交的图元。参考 World Wind Java 的空间检索方式, 系统对指定的空间点进行了一定的扩展, 也就是对用户选择位置设置了一定的缓冲半径, 避免要求用户必须进行“精确点击”才能拾取图元。

在完成 Webs Service 发布后, 集成框架内的应用系统都可以通过 Web Service 接口来检索指定位置的确权用海数据, 其返回值可以通过两种方法调用:

(1) 通过 AJAX 进行无刷新检索, 然后更新系统网页;

(2) 通过 Applet 的 JavaScript 接口, 也就是 Object.getWindow(this).eval (javaScriptString)方法获取返回值后直接在 Applet 中解析应用;

通过 Web Service 服务的发布, ARC SDE 的空间数据检索函数已经能够被 World Wind Java 和其他 Web 应用系统所调用, 通过这种方法解决了大数据量下 WMS 空间数据服务与 ARC SDE 数据源的显示效率和空间检索功能的矛盾, 图 2 显示了利用这种技术构建的“北海区海洋综合信息一张图”中的空间数据检索结果。

```

ISDESpatialElementsFactory.java
package cn.gov.bhjt.ncss.webservice.sde;

import java.util.List;

@WebService
public interface ISDESpatialElementsFactory extends IFactory {
    /**
     * 本Factory的标识名
     */
    public static String FACTORY_NAME = "SDESpatialElementsFactory";

    @WebMethod(operationName = "getListByType")
    public List<SpatialIndexElements> getList(String uuid, String spatialType, Double longitude, Double latitude, Double radius);

    @WebMethod(operationName = "getListByTypeList")
    public List<SpatialIndexElements> getList(String uuid, List<String> spatialType, Double longitude, Double latitude, Double radius);
}

SDESpatialElementsFactoryImpl.java
package cn.gov.bhjt.ncss.webservice.sde;

import java.lang.reflect.InvocationTargetException;

@WebService(endpointInterface = "cn.gov.bhjt.ncss.webservice.sde.ISDESpatialElementsFactory", serviceName = "SDESpatialElementsUtilis")
public class SDESpatialElementsFactoryImpl extends FactoryClass implements ISDESpatialElementsFactory {

    private Logger databaseLogger = LoggerFactory.getDatabaseLogger();

    public SDESpatialElementsFactoryImpl() {
    }

    @Override
    @WebMethod(operationName = "getListByType")
    public List<SpatialIndexElements> getList(String uuid, String spatialType, Double longitude, Double latitude, Double radius) {
        String sql;
        WhereSQLAppender whereSQLAppender = (WhereSQLAppender) getWhereSQLAppender().clone();
        DatabaseSQLCreator databaseSQLCreator = (DatabaseSQLCreator) getDatabaseSQLCreator().clone();
        List<SpatialIndexElements> list = new ArrayList<>();
    }
}

```

Figure 1. Web service of spatial retrieve for point by CXF  
 图 1. 使用 CXF 封装的指定空间点检索的 Web Service 服务

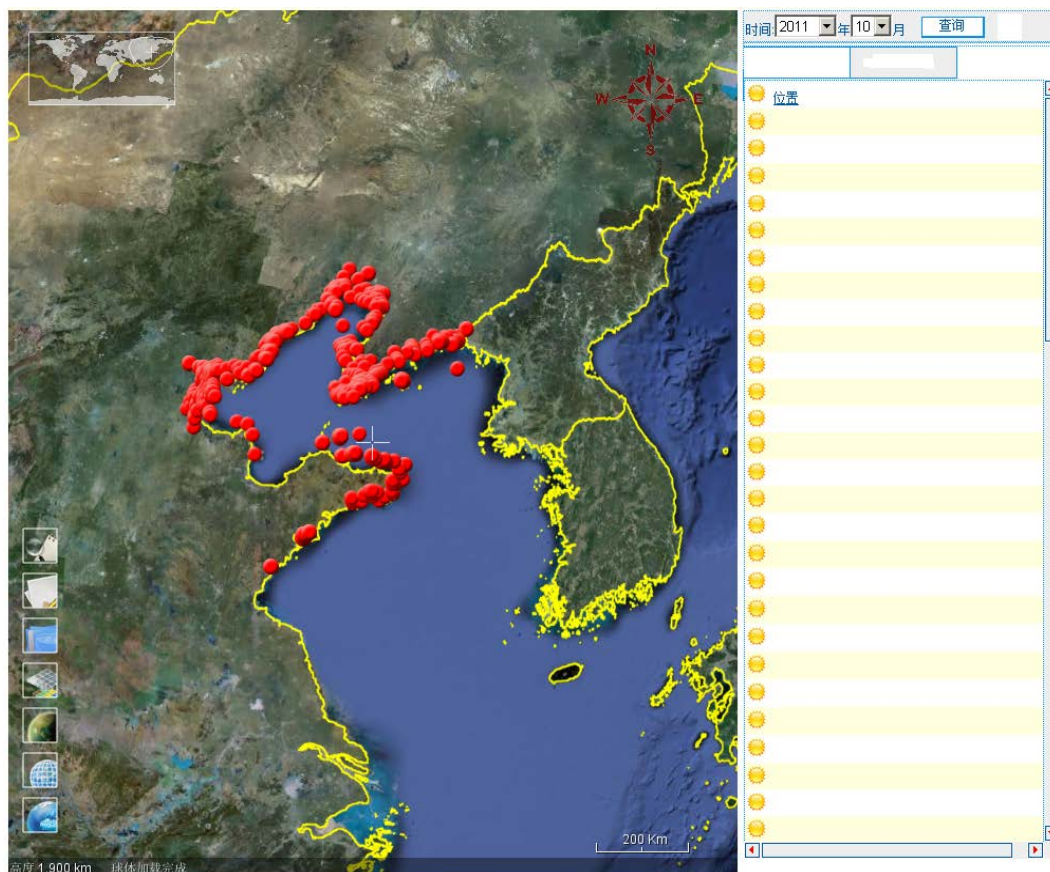


Figure 2. User view of application world wind java  
 图 2. 利用本文方法实现的 World Wind Java 应用界面

## 4. 结束语

系统利用 ARC SDE 的空间检索函数扩展了 World Wind Java 中的 WMS 服务图层的空间数据操作能力, 在 World Wind Java 平台上实现了点、线、面检索和空间缓冲检索等多种功能, 并利用 Web Service 服务提供了可复用的空间数据检索与操作接口, 降低了系统耦合度, 提高了开发和应用的灵活性。利用该方法所构建的系统已经在国家海洋局北海分局建立的“北海区海洋综合信息一张图”得到了应用, 为北海区海洋管理信息化和集成框架建设提供了有力的支持。

## 基金项目

本项目受国家海洋局海洋公益性行业科研专项基金 201305026、山东省海洋生态环境与防灾减灾重点实验室开放基金 201605 支持。

## 参考文献 (References)

- [1] Humphrey, L. (2011) ARC SDE. Betascript Publishing, US.
- [2] 邱洪钢, 张青莲, 熊友谊. ArcGIS Engine 地理信息系统开发从入门到精通[M]. 北京: 人民邮电出版社, 2013.
- [3] 周玲, 沙颖平, 高延铭, 等. 基于 World Wind Java 的 3D 海洋石油平台管理系统的设计与实现[J]. 测绘与空间地理信息. 2012, 35(7): 35-37, 40.
- [4] 周玲, 高延铭, 王海红, 等, 著. World Wind Java 三维地理信息系统开发技术指南[M]. 北京: 机械工业出版社, 2013.
- [5] Peters, D. (2012) Building a GIS: System Architecture Design Strategies for Managers. Environmental Systems Research Institute Inc., US.
- [6] 任刚. 基于 Apache CXF 构建 SOA 应用. 北京: 电子工业出版社, 2013.

### 期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: [csa@hanspub.org](mailto:csa@hanspub.org)