

空间/时空数据库概念与实战

TST-图贲

- ◆ 空间/时空数据库概念
- ◆ 应用场景
- ◆ 几何模型使用案例
- ◆ 栅格模型使用案例
- ◆ 轨迹模型使用案例
- ◆ 点云模型使用案例

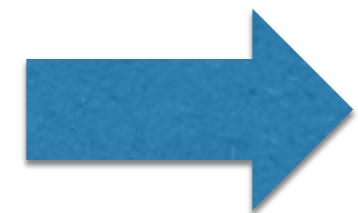
空间/时空数据库概念

空间/时空数据库

阿里云 | 
奥运会全球指定云服务商

数据库直接存储和管理地理空间数据：

- 空间模型（类型）
- 空间关系（拓扑）
- 空间参考（坐标系、投影）
- 空间检索（索引）
- 一系列空间分析和操作的算子



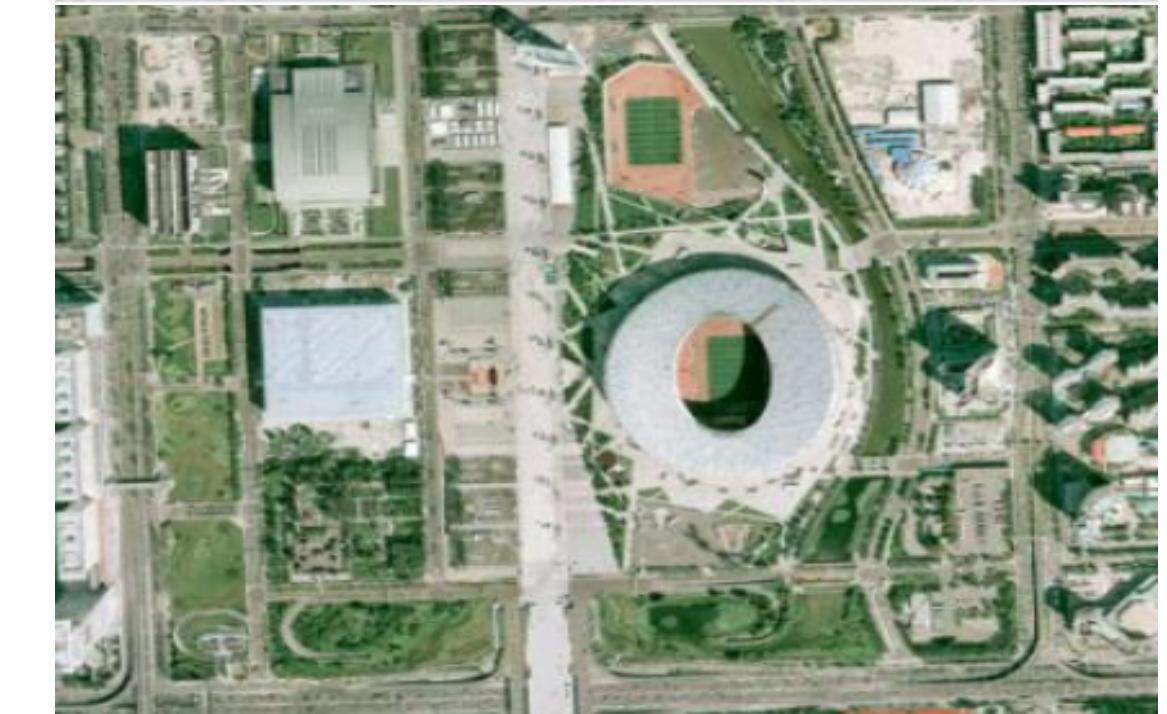
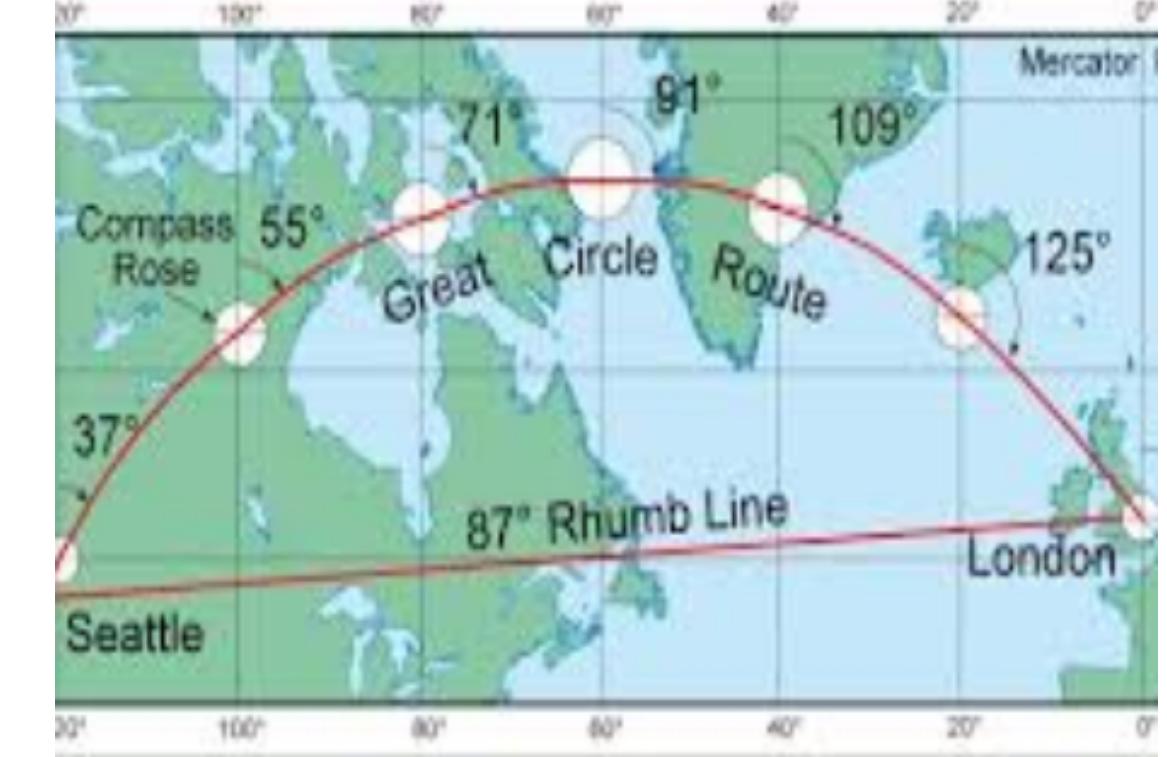
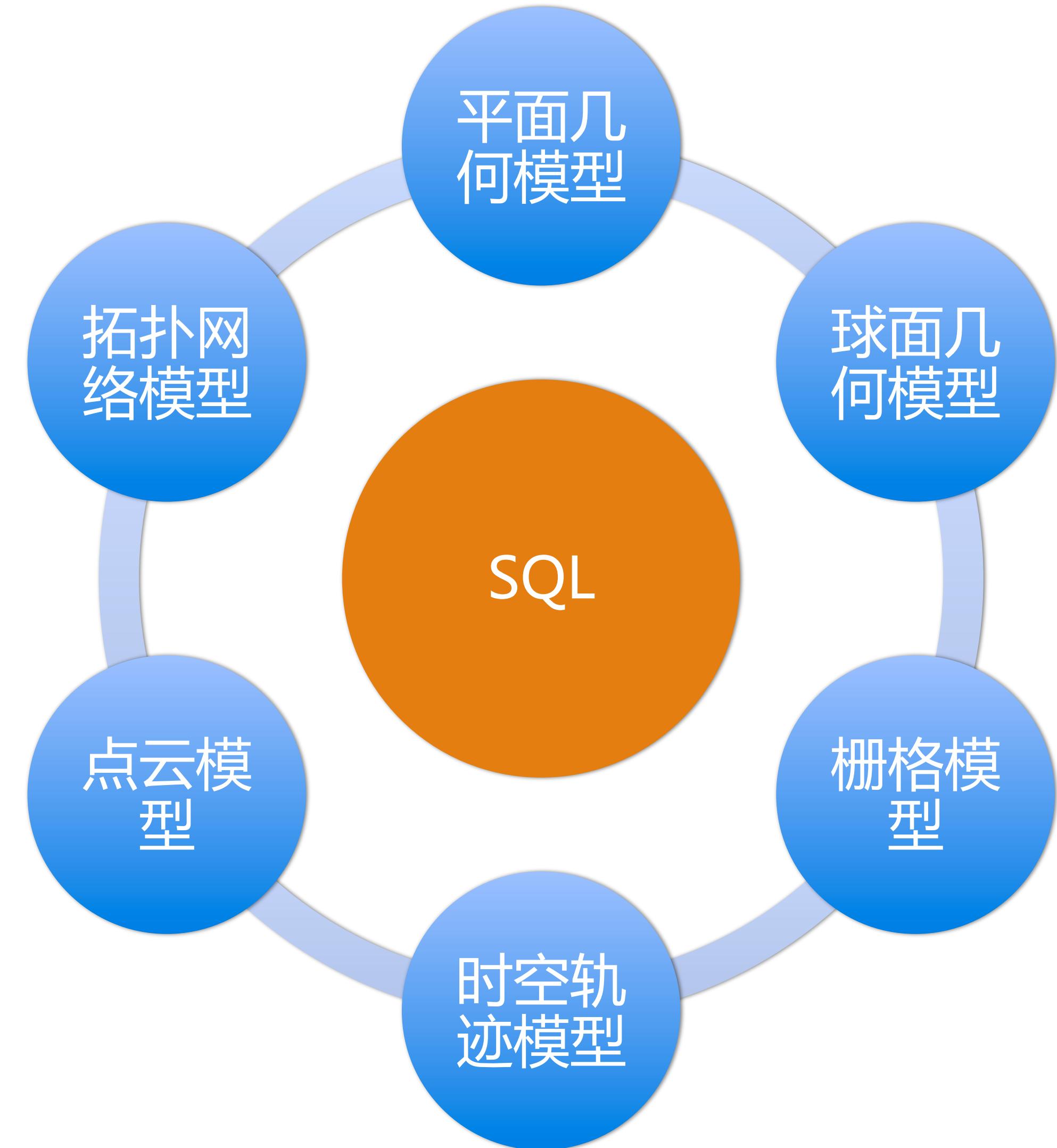
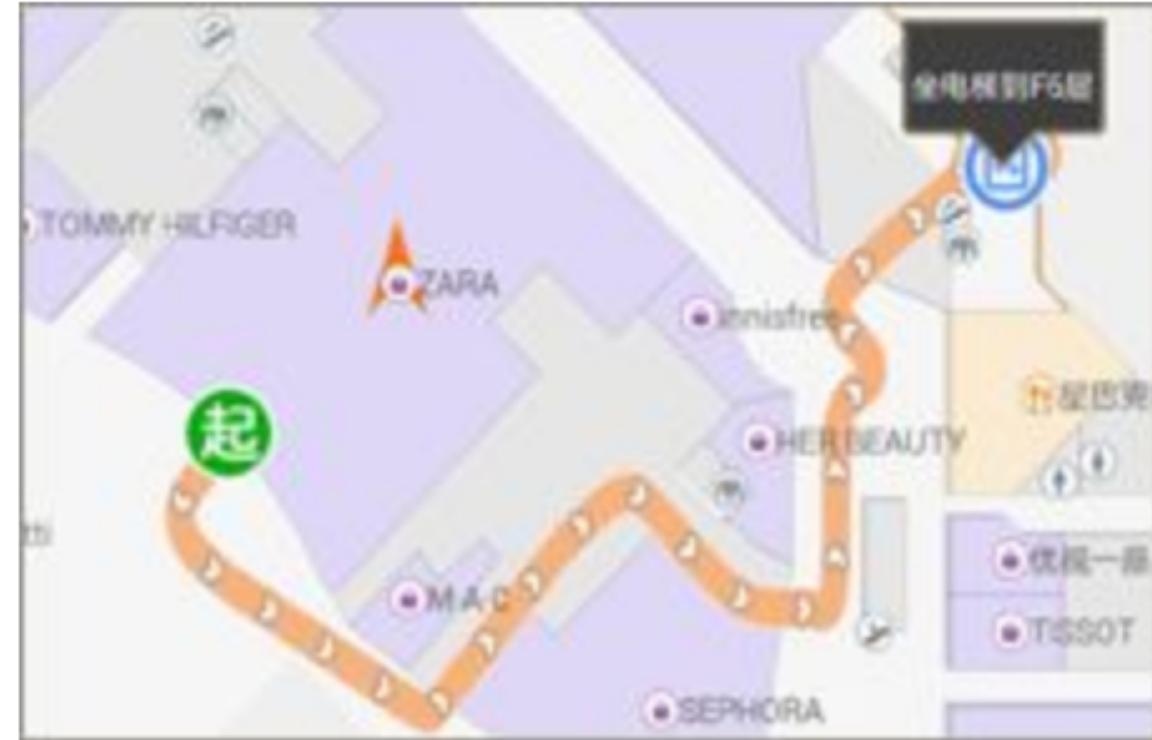
时空数据库：

- 动态监控、轨迹、遥感
- 时间维度

常见的空间/时空数据库（关系型）：

- PostGIS
- Ganos (PostGIS++) (时空数据库)
- Oracle Spatial

空间模型



应用场景



互联网出行

运动社交

城市大脑

智慧物流

测绘遥感

激光点云管理

车联网

外卖递送

新零售/O2O

LBS

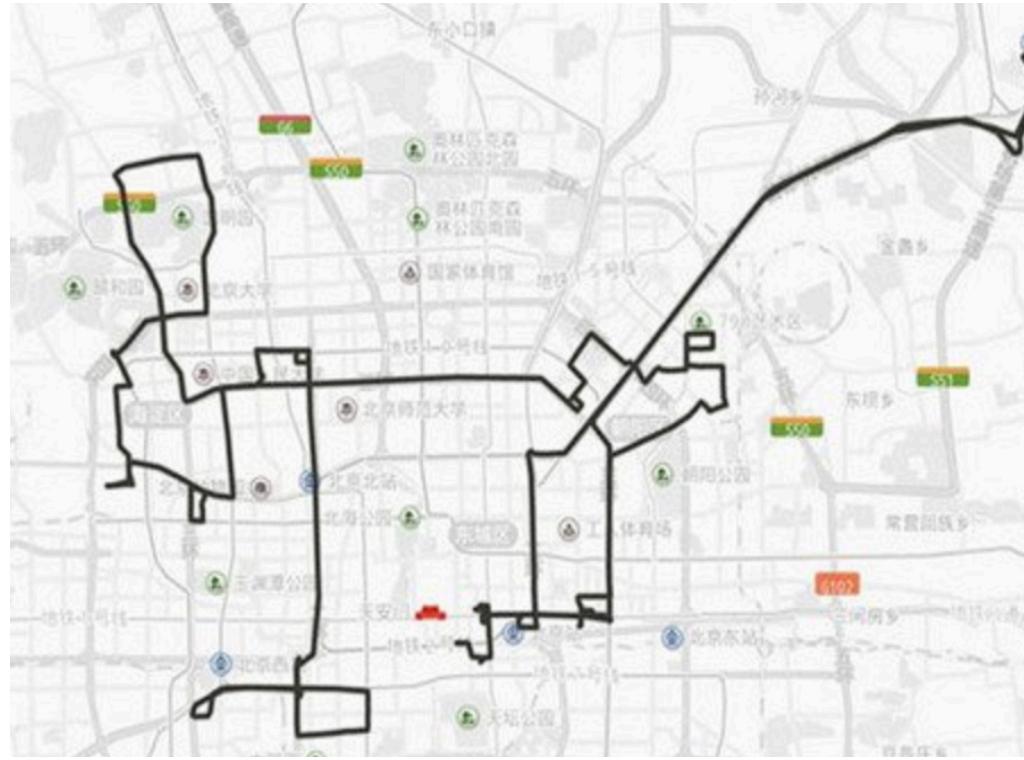
空间/时空数据库

几何模型使用案例

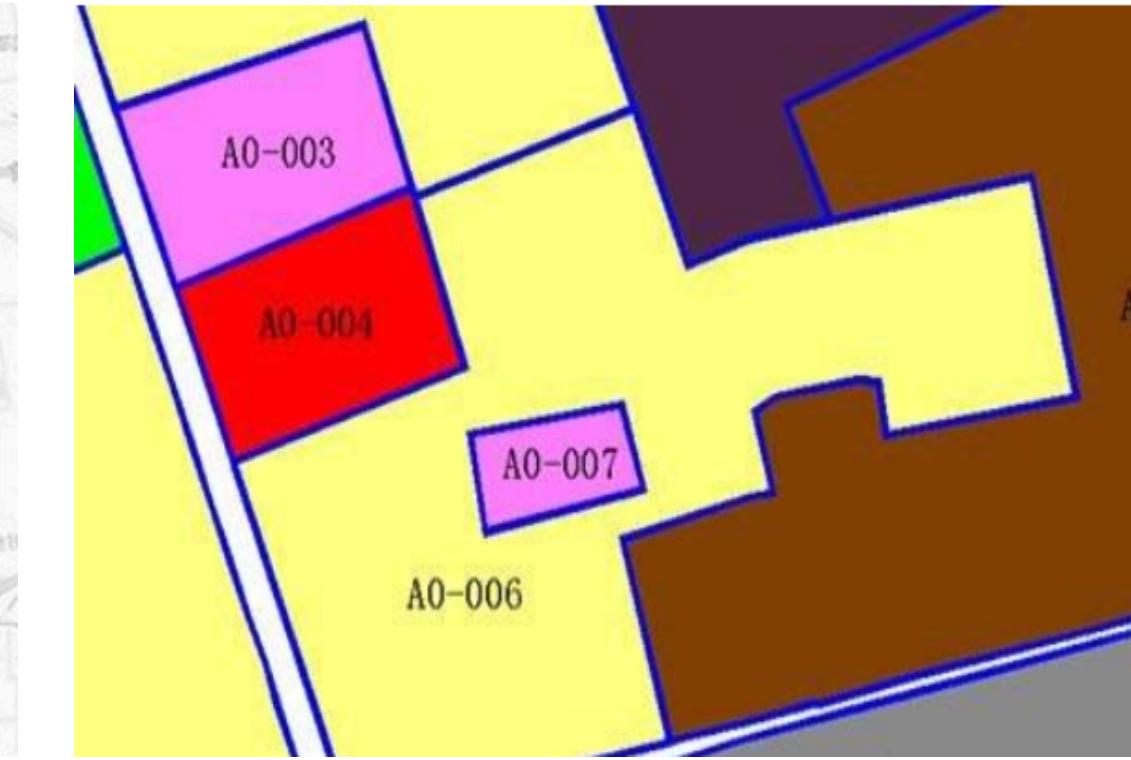
2D/3D/4D几何模型



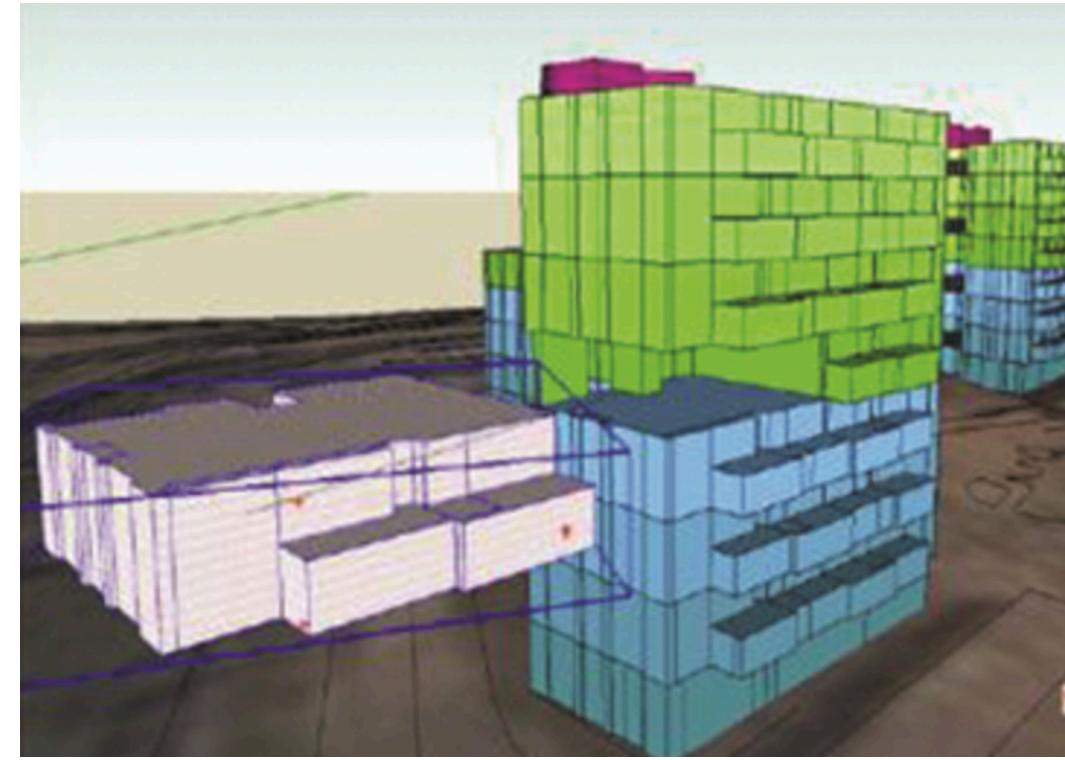
点



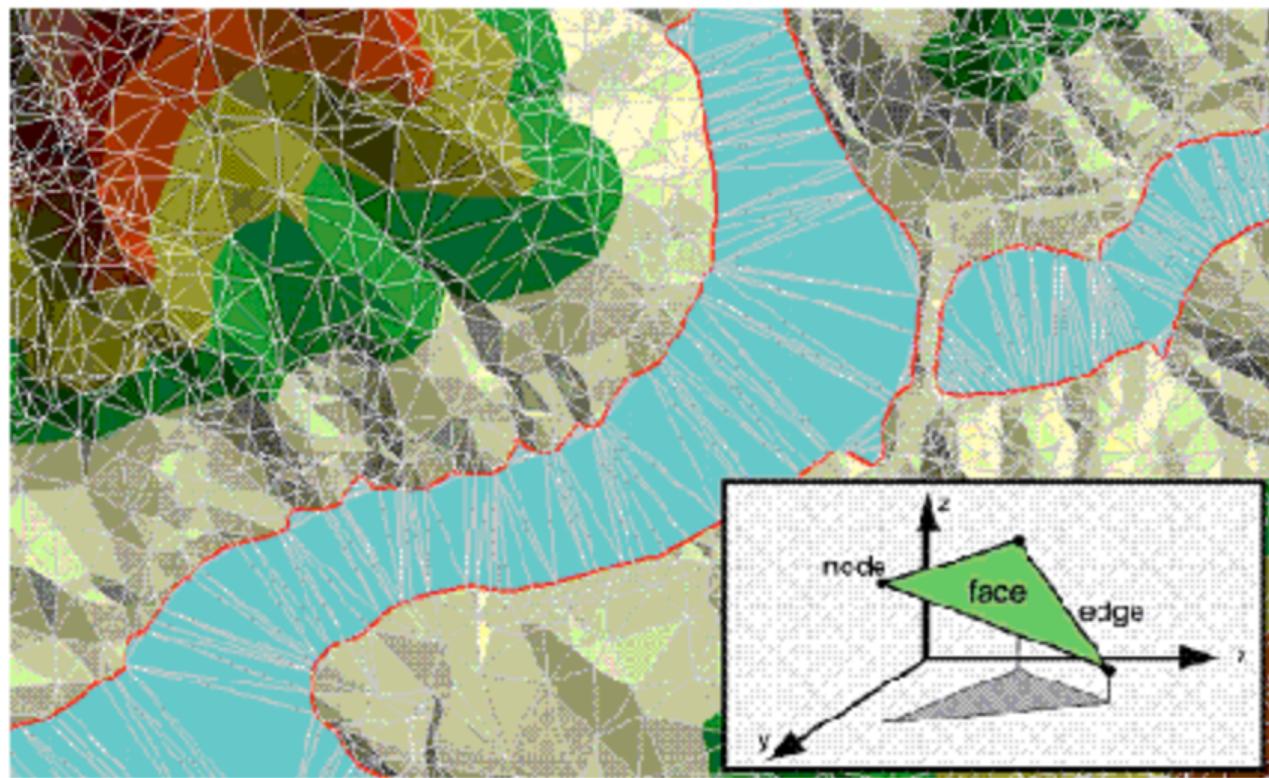
线



面



体



TIN (不规则三角网)

Point
LineString
Polygon
CircularString
CompoundCurve
Curvepolygon

GeometryCollection
MultiPoint
MultiLineString
MultiCurve
MultiPolygon
MultiSurface

PolyhedralSurface
Triangle
Tin

创建几何扩展



Ganos

--创建几何扩展

```
Create extension ganos_geometry cascade;
```

--创建拓扑扩展

```
Create extension ganos_geometry_topology;
```

--创建sfCGAL插件扩展

```
Create extension ganos_geometry_sfCGAL;
```

PostGIS

--创建几何扩展

```
Create extension postgis;
```

--创建拓扑扩展

```
Create extension postgis_topology;
```

--创建sfCGAL插件扩展

```
Create extension postgis_sfCGAL;
```

创建表

--方式一：直接创建带geometry字段的表

```
CREATE TABLE ROADS ( ID int4, ROAD_NAME varchar(25), geom geometry(LINESTRING,3857) );
```

--方式二：先创建普通表，再附加几何字段

```
CREATE TABLE ROADS ( ID int4, ROAD_NAME varchar(25) );
SELECT AddGeometryColumn( 'roads', 'geom', 3857, 'LINESTRING', 2);
```

数据入库

--方式一：入库工具

shp2pgsql

--方式二：sql

```
INSERT INTO roads (id, geom, road_name)
VALUES (1,ST_GeomFromText('LINESTRING(191232 243118,191108 243242)',3857),'**高速');
```

空间测量

--长度计算

```
SELECT sum(ST_Length(the_geom))/1000 AS km_roads FROM bc_roads;
```

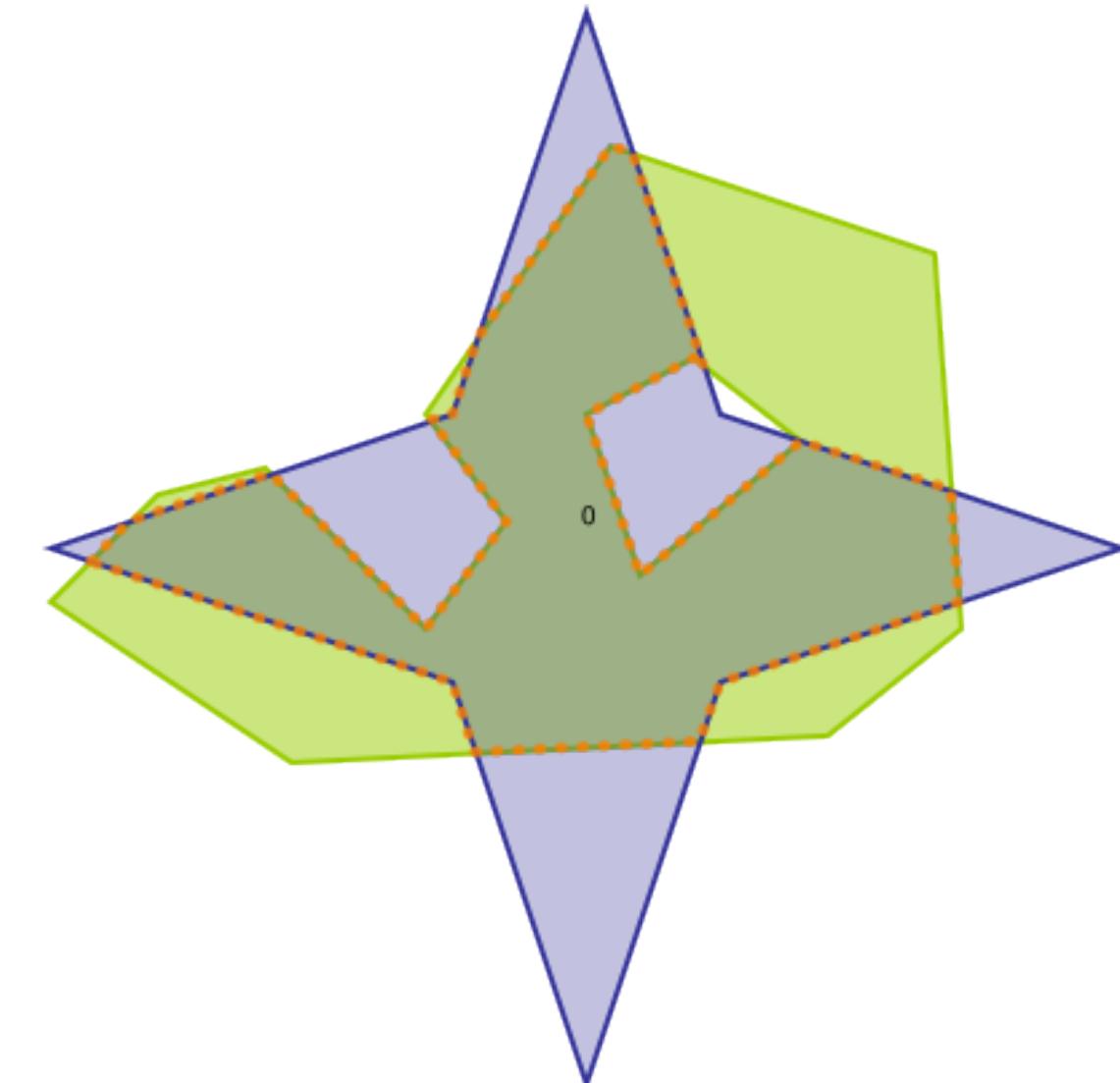
--面积计算

```
SELECT ST_Area(the_geom)/10000 AS hectares FROM bc_municipality WHERE name = 'PRINCE GEORGE';
```

空间分析

```
SELECT  
sum(ST_Area(CASE  
WHEN ST_CONTAINS(foo.geom, p.geom) THEN p.geom  
ELSE ST_INTERSECTION(foo.geom, p.geom)  
END  
))
```

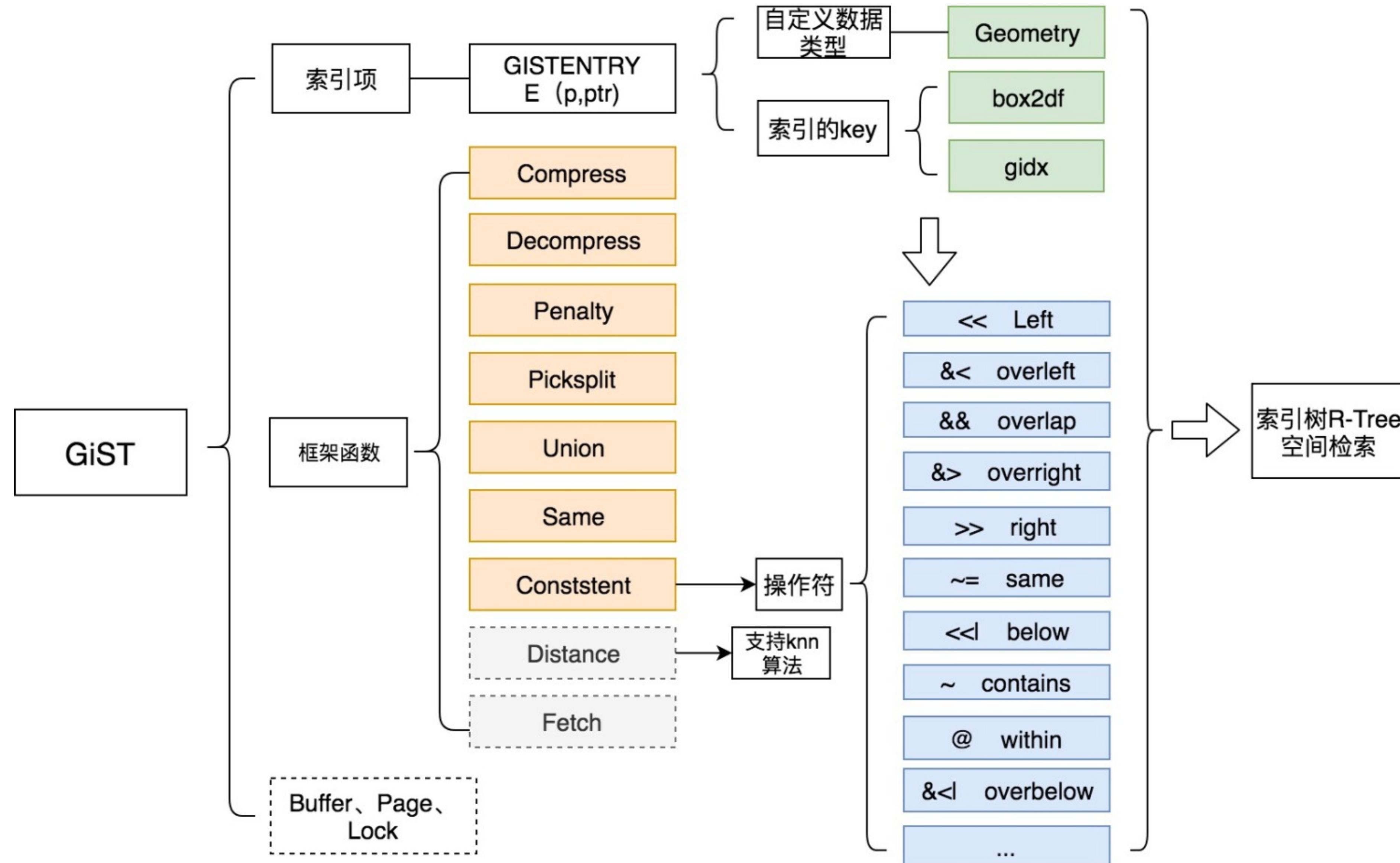
```
FROM poly180w p , boua_for_180w foo  
where ST_INTERSECTS(foo.geom, p.geom) = TRUE;
```



空间索引



奥运会全球指定云服务商



空间搜索



```
select * from tbl_pos where att1=:att1 and att2=:att2 and
st_contains(
    geometry(
        ST_Buffer( -- 圆形转换为对称多边形。
            geography(
                st_setsrid(st_makepoint(:x,:y), 4326) -- 中心点
            ),
            5000 -- st_buffer生成以某个POINT为中心的，半径距离为5000米的polygon空间对对象
        )
    ), -- 将geography转换为geometry
    pos -- st_contains, polygon包含point
) order by pos <-> st_setsrid(st_makepoint(:x,:y), 4326) limit 100;
```

栅格模型使用案例

栅格模型-大规模遥感影像

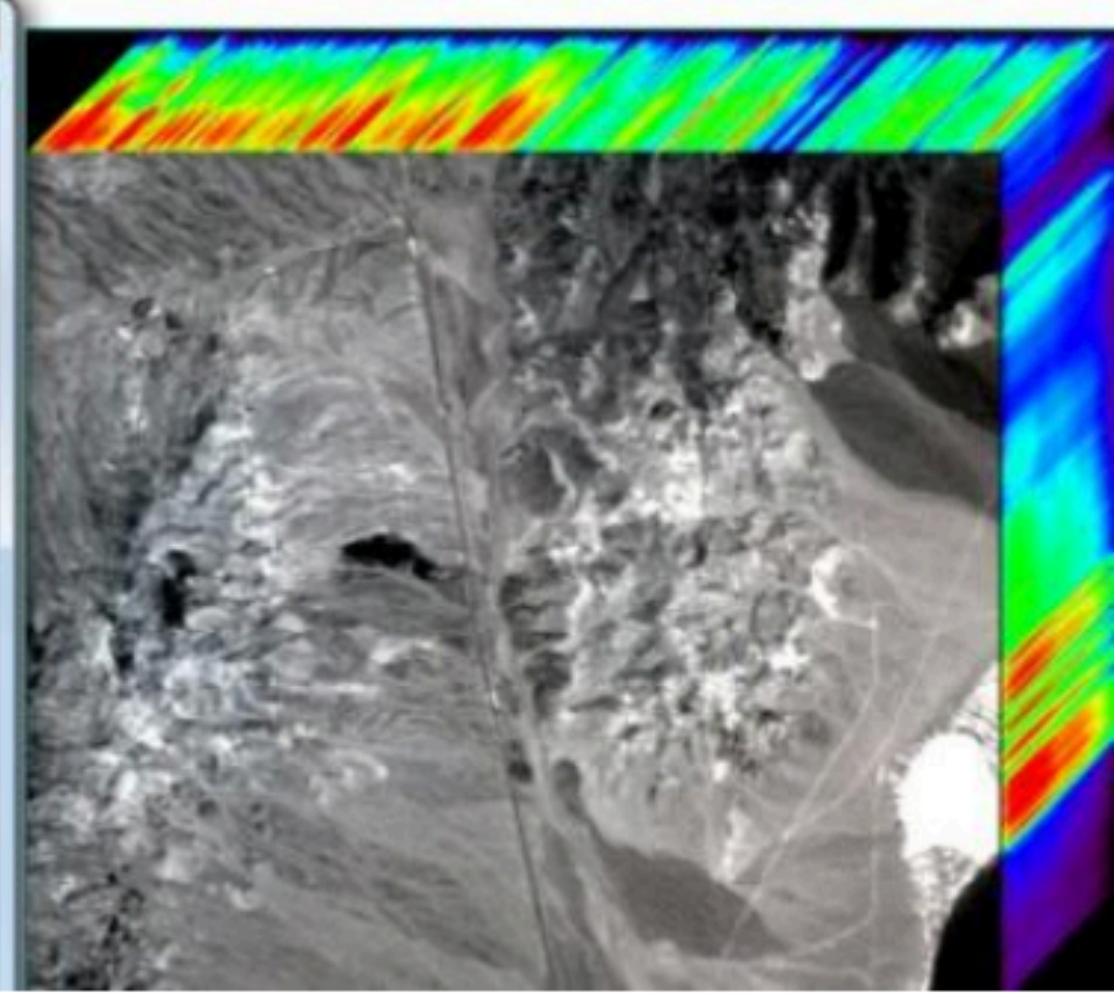
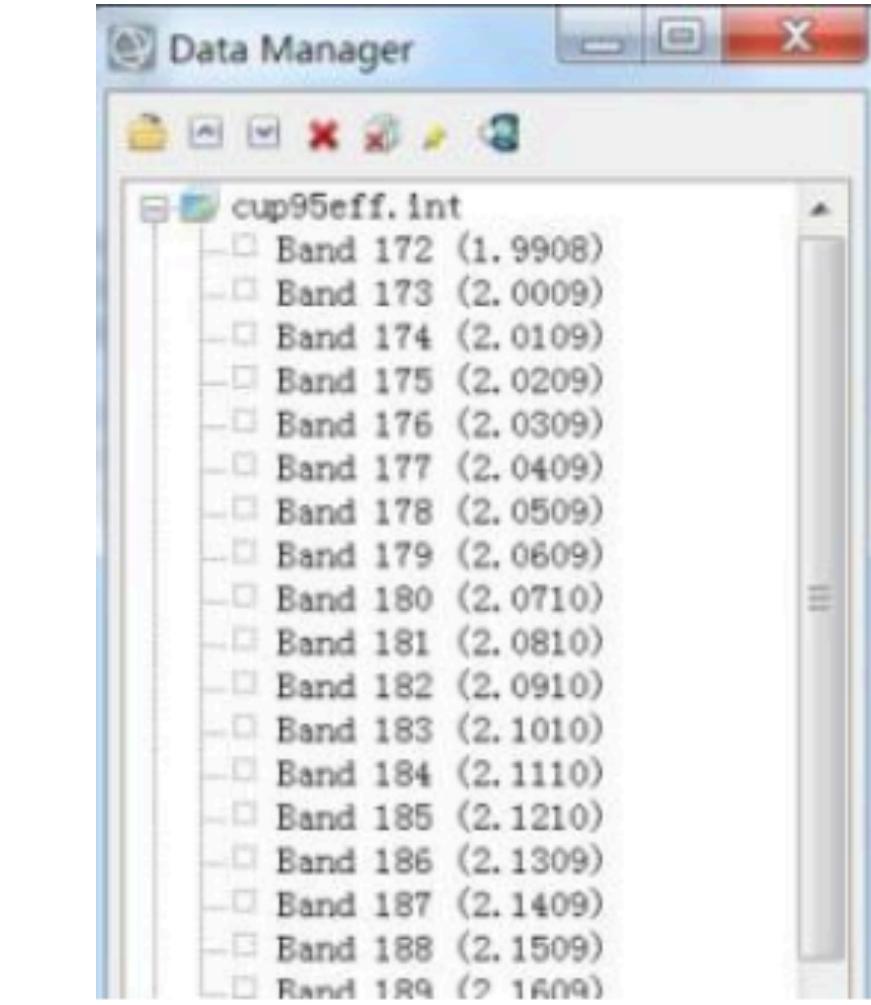
阿里云 | 
奥运会全球指定云服务商



单波段全色影像



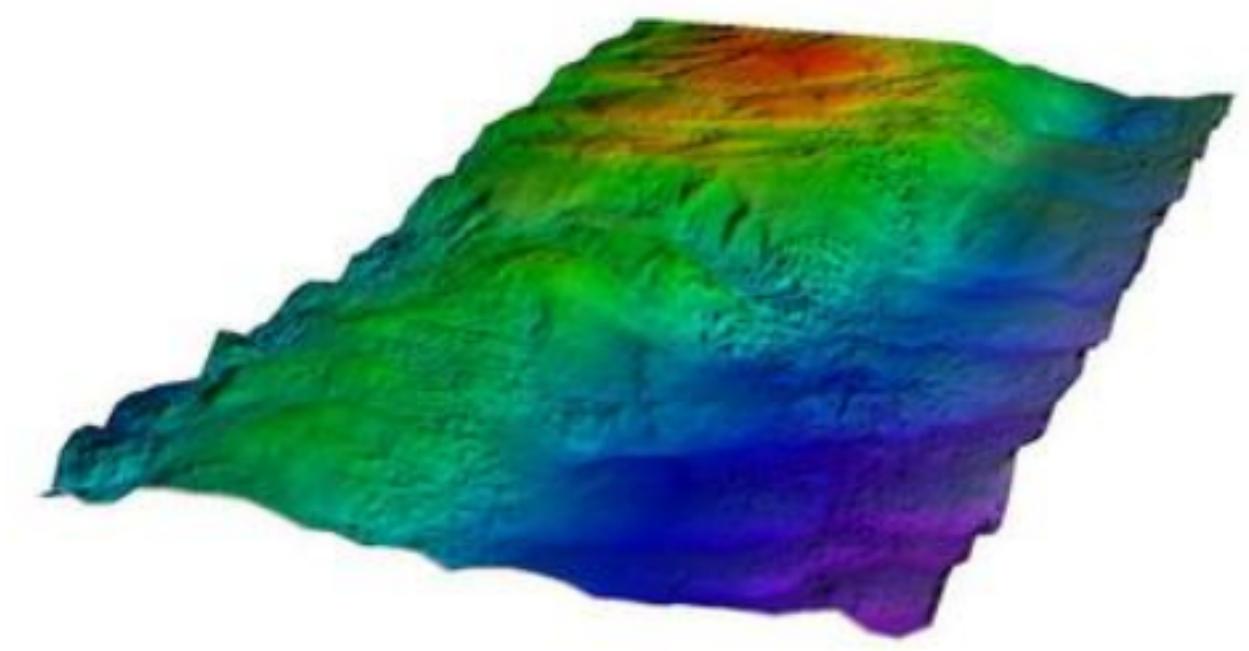
多光谱影像



高光谱影像



全景照片



数字高程模型(DEM)

栅格模型使用案例



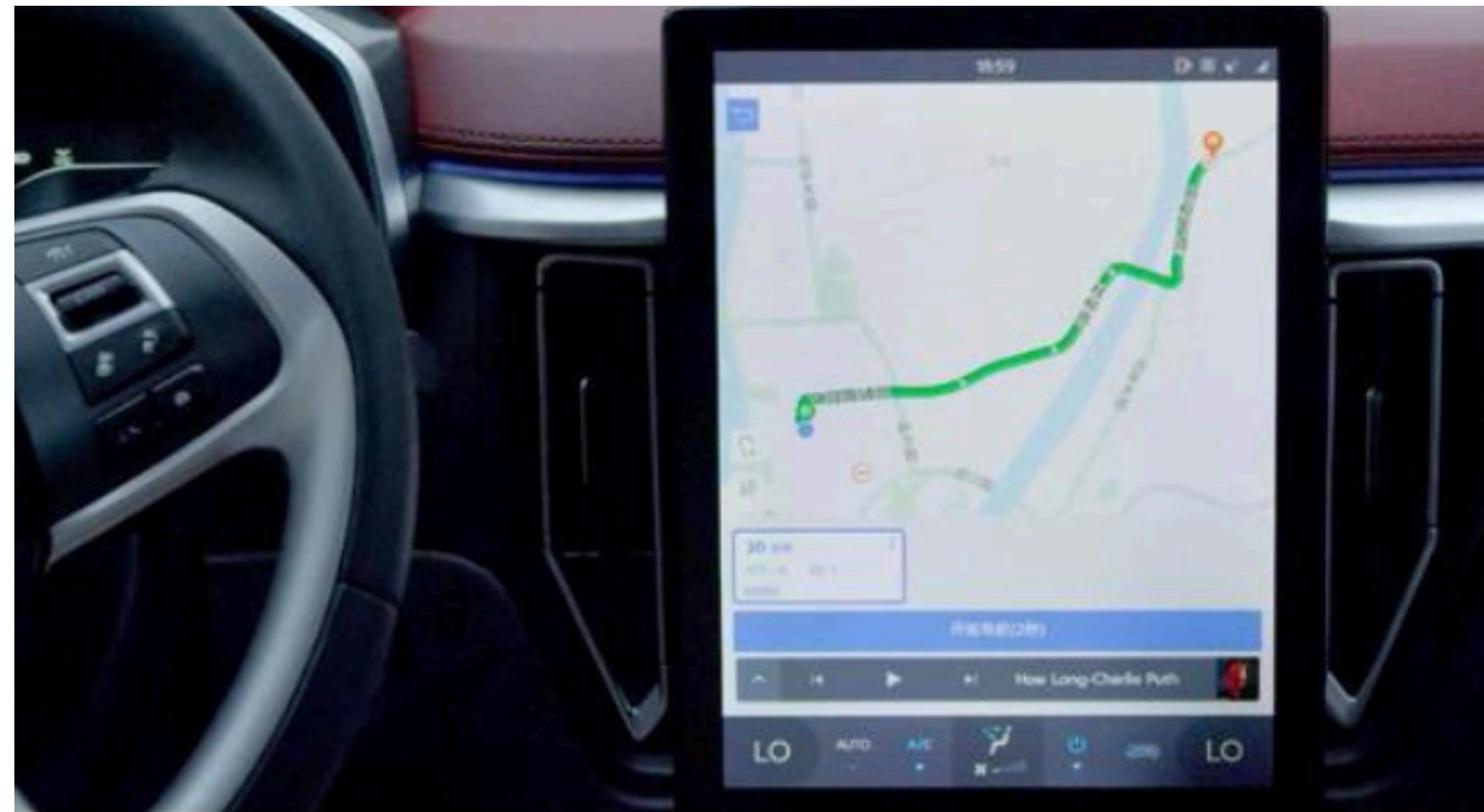
链接云栖社区文章：

使用Ganos实现洪涝灾害承灾体损失综合评估

<https://yq.aliyun.com/articles/738524>

轨迹模型使用案例

MOD轨迹模型



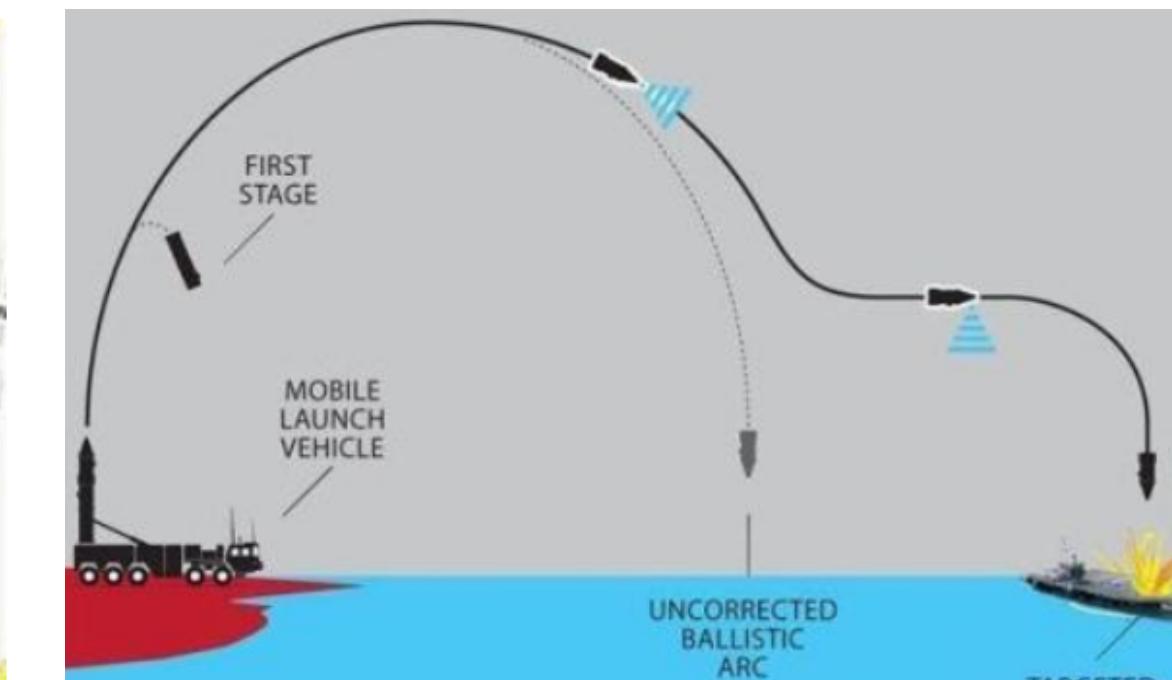
- Moving Objects Database
- 高维度移动对象时空轨迹建模
位置(x,y,z)、时间(t)、属性(速度、方向.....)
- 轨迹时空事件语义(event)
- 轨迹存储、时空查询、分析计算(穿越分析、OD分析、相似度计算.....)



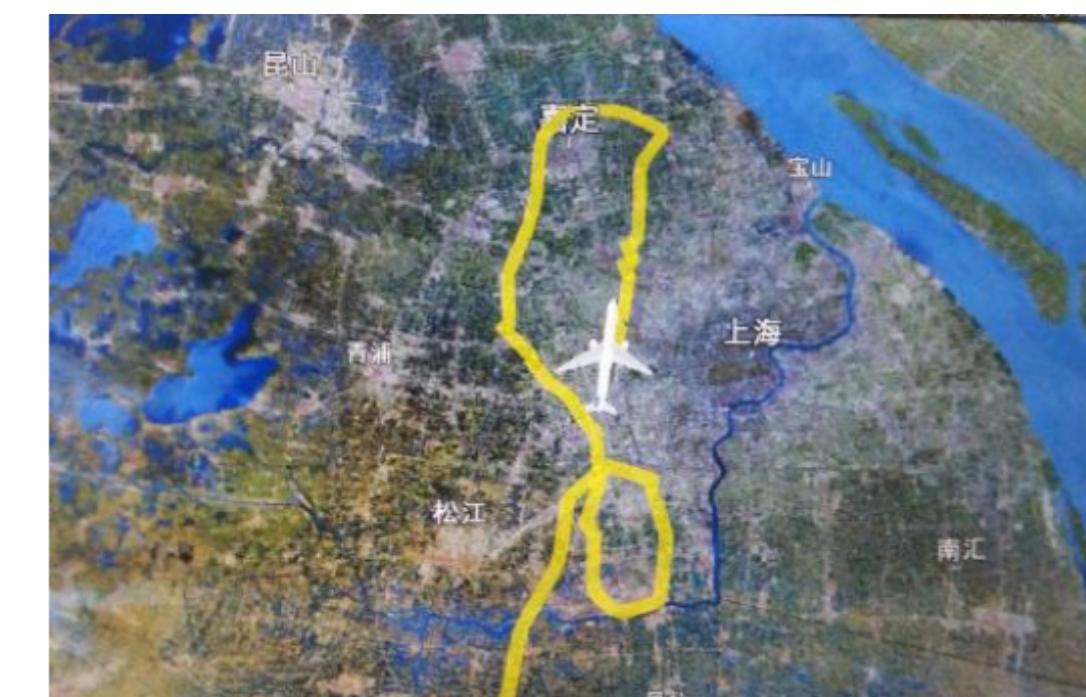
车辆轨迹



人员轨迹



导弹轨迹



飞行轨迹

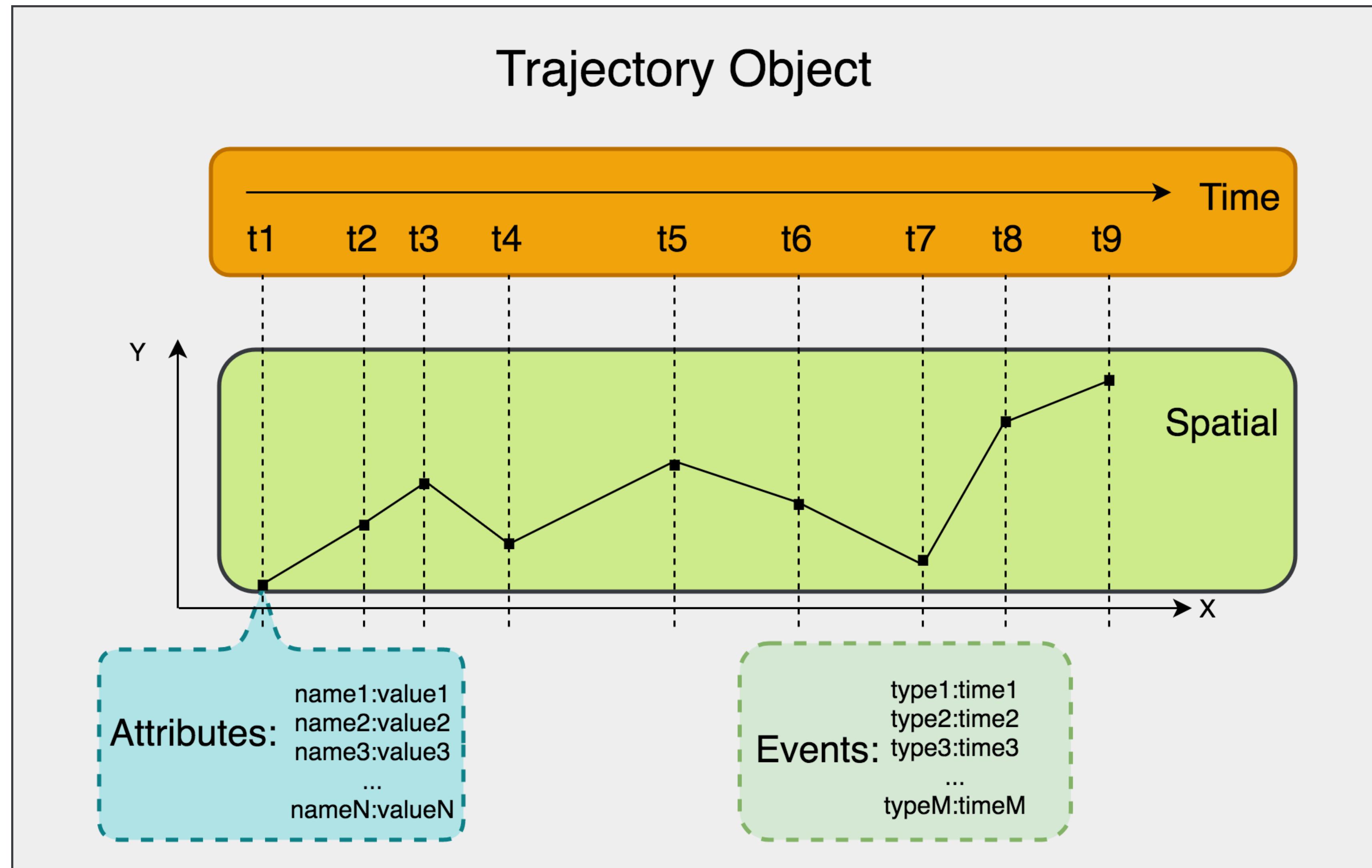
轨迹基本概念

■ Trajectory Point- 轨迹点

- 时空对象
- 某个时刻
- 所在的空间位置
- 附带的属性值

■ Trajectory Object- 轨迹对象

- 一系列轨迹点
- 轨迹事件



■ 创建轨迹模型扩展

Create extension ganos_trajectory cascade;

■ 构造轨迹对象

*trajectory ST_makeTrajectory (leafype type, geometry spatial, timestamp start, timestamp end ,
cstring attrs_json);*

*trajectory ST_makeTrajectory (leafype type, geometry spatial, timestamp[] timeline,
cstring attrs_json);*

■ 示例

```
select ST_makeTrajectory('STPOINT'::leafype,  
st_geomfromtext('LINESTRING (114 35, 115 36, 116 37)', 4326),  
ARRAY['2010-01-01 14:30'::timestamp, '2010-01-01 15:00'::timestamp, '2010-01-01 15:30'::timestamp],  
'{"leafcount":3,"attributes":{"velocity": {"type": "integer", "length": 2,"nullable" : true,"value": [120, 130, 140]}, "accuracy": {"type": "float", "length": 4, "nullable" : false,"value": [120, 130, 140]}, "vesname": {"type": "string", "length": 20, "nullable" : true,"value": ["adsf", "sdf", "sdfff"]}, "active": {"type": "timestamp", "nullable" : false,"value": ["Fri Jan 01 14:30:00 2010", "Fri Jan 01 15:00:00 2010", "Fri Jan 01 15:30:00 2010"]}},  
"events": [{"1" : "Fri Jan 01 14:30:00 2010"}, {"2" : "Fri Jan 01 15:00:00 2010"}, {"3" : "Fri Jan 01 15:30:00 2010"}]);
```

轨迹管理与查询

创建航班轨迹表

```
create table flight_track (
dynamic_id text,
 trajectory ,
info text,
fdst text,
scheduled_deptime bigint,
scheduled_arrrtime bigint,
actual_deptime bigint,
actual_arrrtime bigint,
aircraft_number text,
org_timezone text,
dst_timezone text,
icaoId text
);
```

轨迹时空查询SQL

```
select dynamic_id, traj, st_leafcount(traj) from flight_track
where ST_intersects(traj,
'2019-04-01 12:00:00'::timestamp, --开始时间
'2019-04-01 14:00:00'::timestamp, --结束时间
ST_GeomFromText('Polygon((115.3564 21.4893,122.3438 21.4893,
122.3438 41.2646,115.3564 41.2646,115.3564 21.4893))', --空间范围
4326)::geometry);
```

[演示Demo](#)



轨迹压缩

■ 轨迹三大特征

- 位置
- 速度
- 方向

■ 轨迹压缩目的

- 保留重要轨迹点
- 删减冗余或不重要的轨迹点。

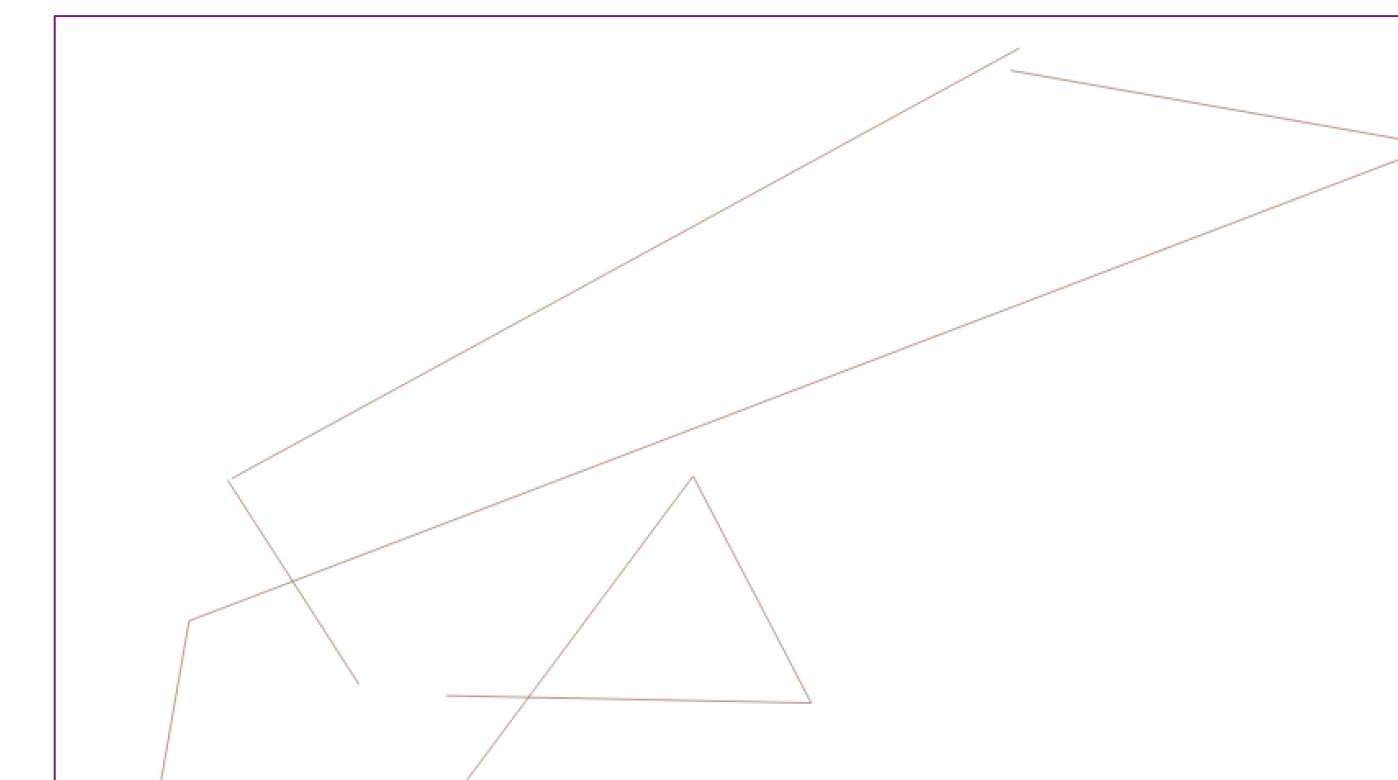
■ 重要轨迹点

- 起始点、终止点
- 拐弯点
- 折返点
- 徘徊点（近似驻点）
- 速度改变的点
- ...

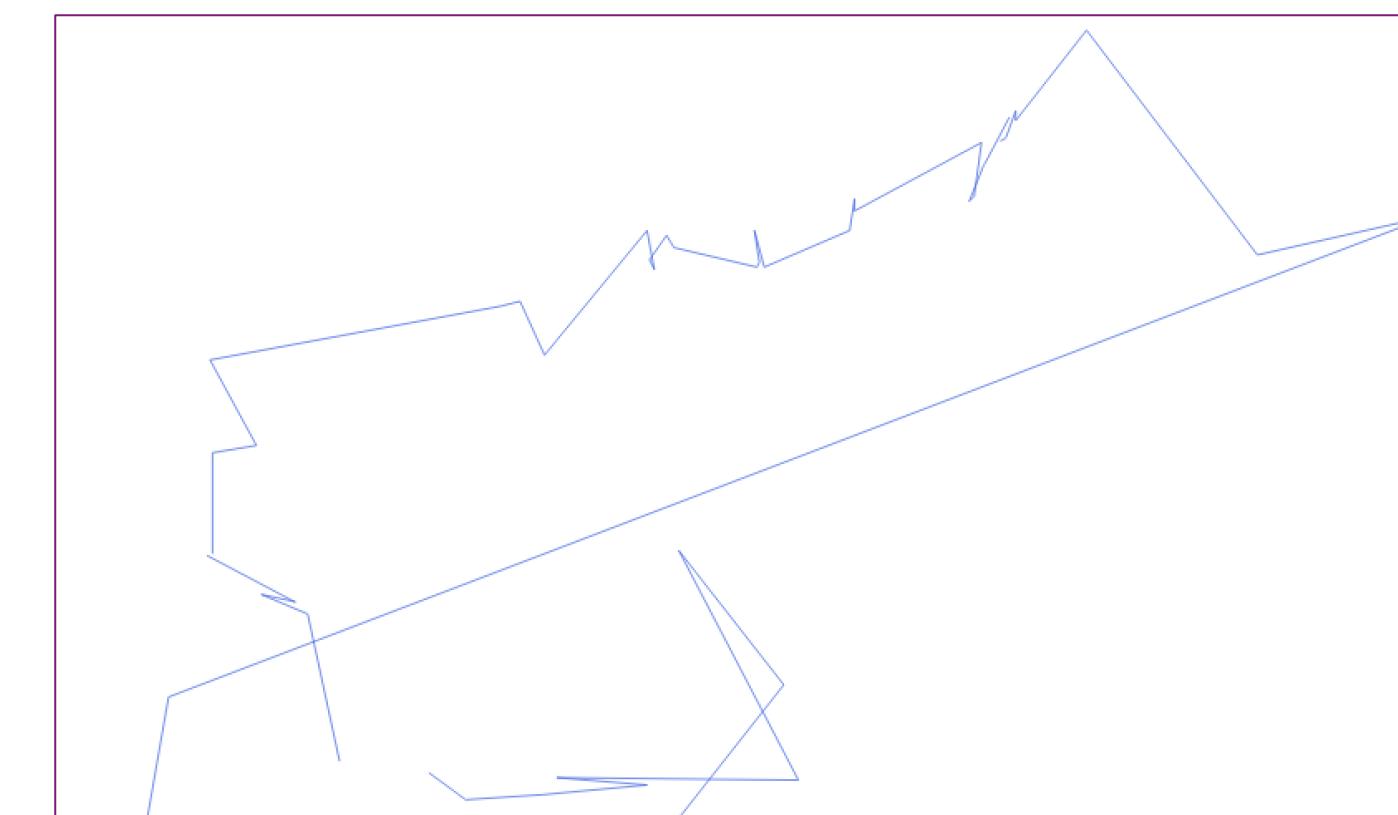
原始轨迹



`st_simplify`
只有空间位置上的
压缩



`st_compress`
• 空间位置
• 方向
• 速度变化



轨迹相似性判断

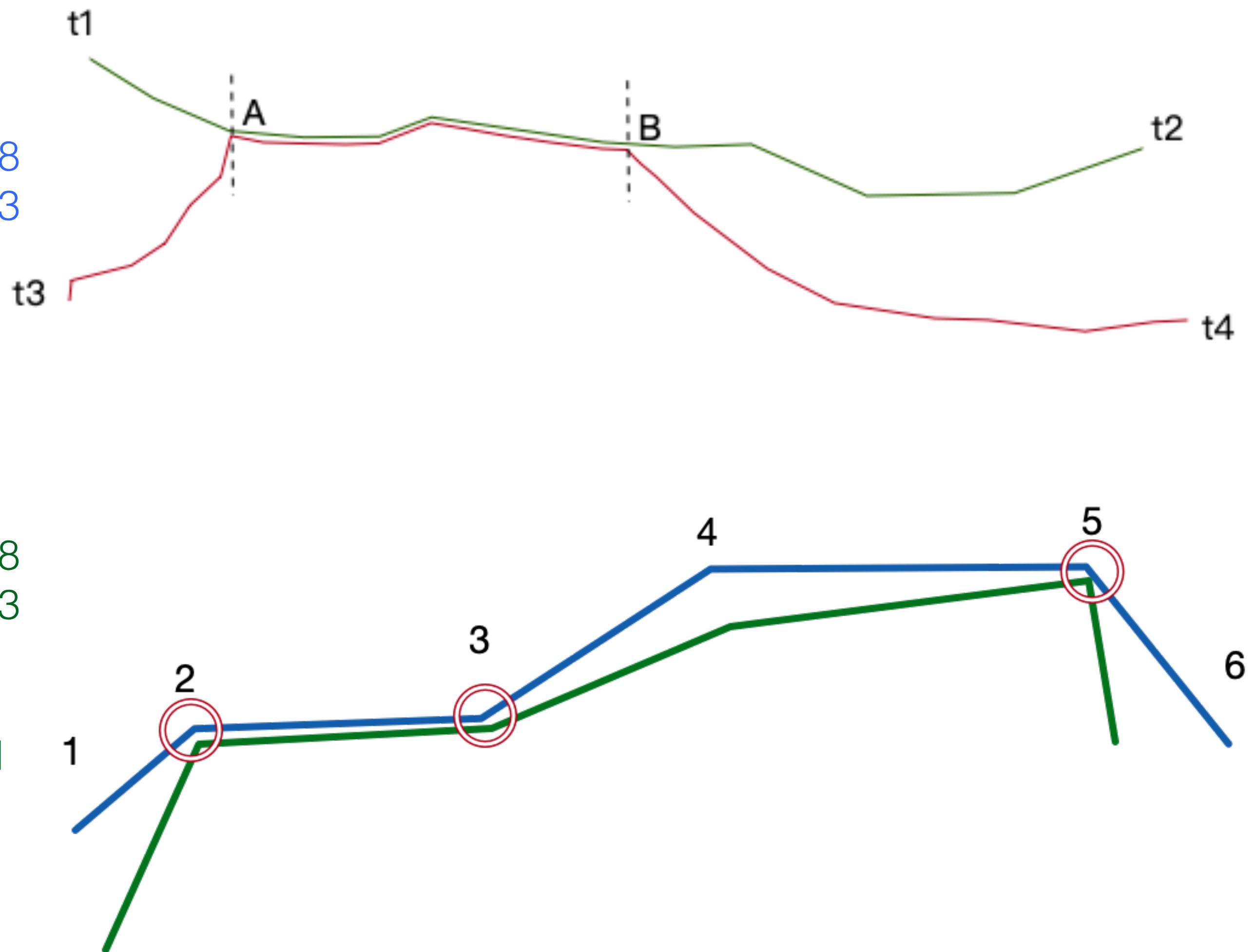
With traj AS (

```
Select ST_makeTrajectory('STPOINT',
'LINESTRINGZ(114.000528 33.588163 54.87 , 114.000535
33.588235 54.85 , 114.000447 33.588272 54.69 , 114.000348
33.588287 54.73 , 114.000245 33.588305 55.26 , 114.000153
33.588305 55.3)::geometry, ARRAY['2010-01-01
11:30'::timestamp, '2010-01-01 11:31', '2010-01-01 11:32',
'2010-01-01 11:33','2010-01-01 11:34','2010-01-01 11:35'],
NULL) a,
```

```
ST_makeTrajectory('STPOINT',
'LINESTRINGZ(114.000529 33.588163 54.87 , 114.000535
33.578235 54.85 , 114.000447 33.578272 54.69 , 114.000348
33.578287 54.73 , 114.000245 33.578305 55.26 , 114.000163
33.588305 55.3)::geometry,ARRAY['2010-01-01
11:29:58'::timestamp, '2010-01-01 11:31:02', '2010-01-01
11:33', '2010-01-01 11:33:09','2010-01-01 11:34','2010-01-01
11:34:30'], NULL) b)
```

```
Select st_LCSSimilarity(a, b, 100) from traj;
st_lcssimilarity
```

3
(1 row)



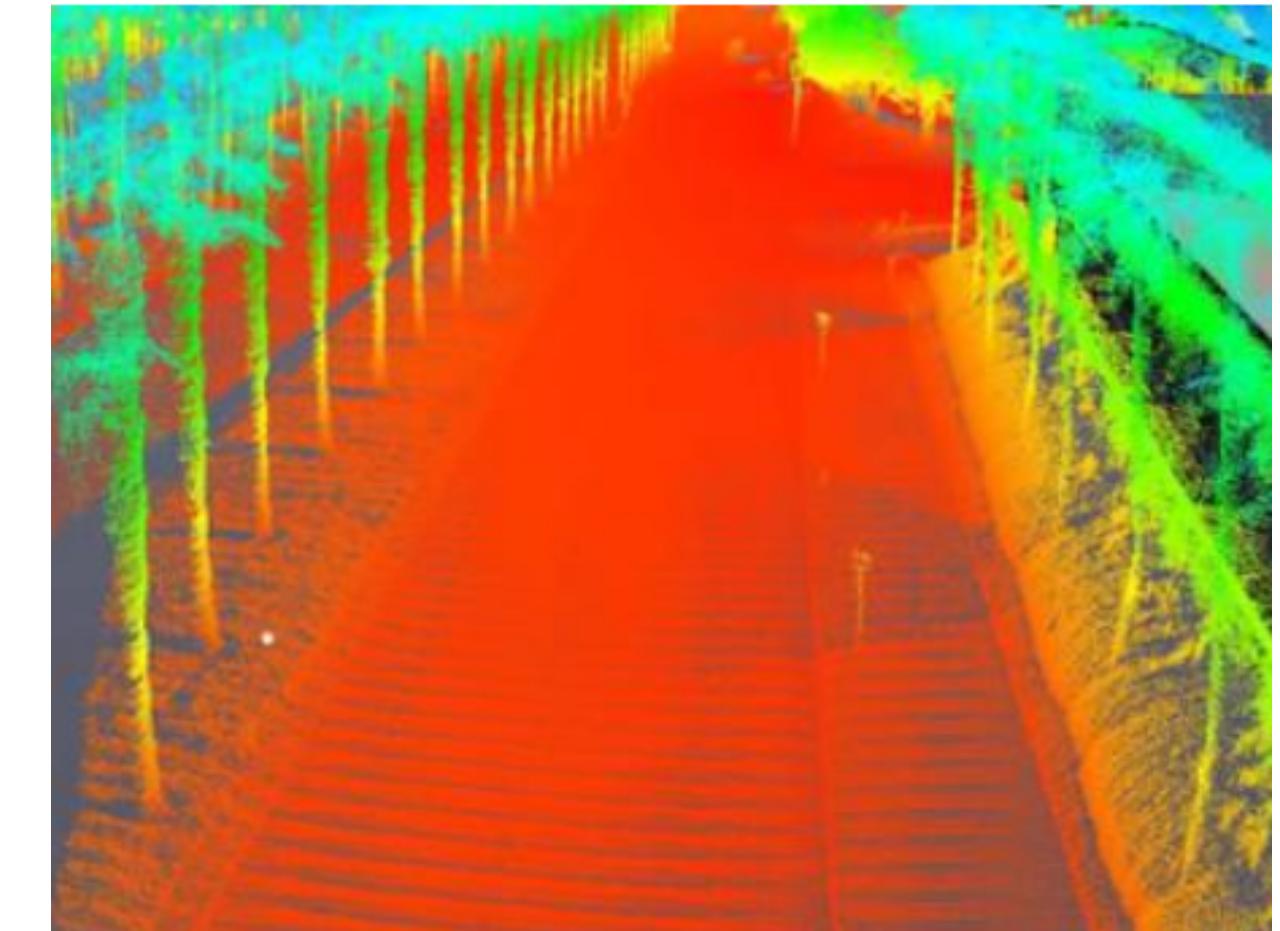
点云模型使用案例

点云模型

激光点云(LiDAR)：位置+传感属性信息，如 RGB、时间、温度、湿度

点云数据的挑战：

- Huge volume : billions (十亿) and even trillions (万亿)
- Multi-dimensional : X.Y. Z + N-D Attr.

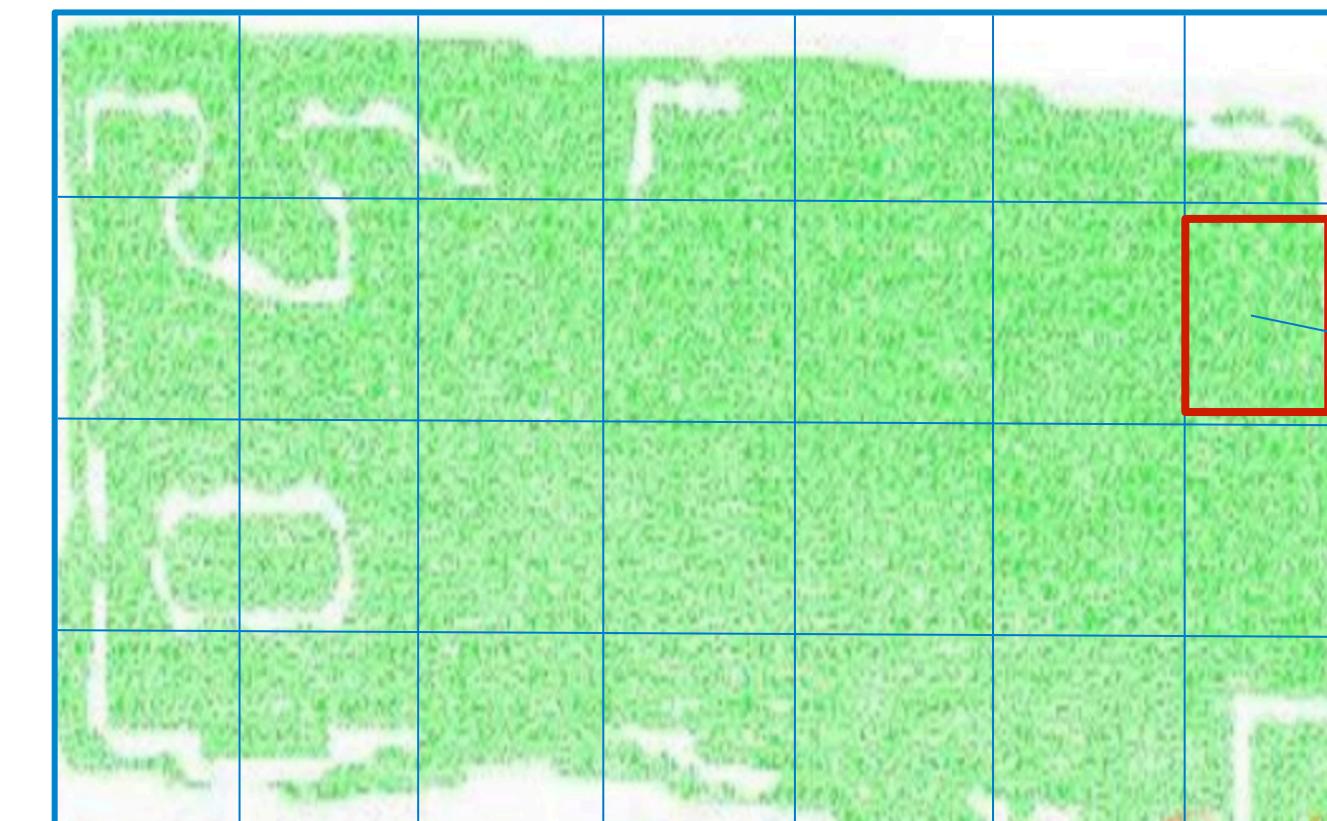


PcPatch类型：基于XML Schema + 分块存储模型

✓ 支持40多个空间关系、操作、统计值计算函数

✓ 支持高效压缩算法：

- Dimensional压缩：20-30%，无损
- LASzip：7-20%，无损



compatible with PDAL drivers to
import data directly from .las

创建点云扩展



创建点云模块扩展：

```
Create extension ganos_pointcloud cascade ;  
Create extension ganos_pointcloud_geometry;
```

--点类型 pcpoint

```
CREATE TABLE points (  
    id SERIAL PRIMARY KEY,  
    pt PCPOINT(1)  --(1)代表选择pointcloud_formats中pcid值为1的schema  
);
```

--点集类型 pcpatch

```
CREATE TABLE patches (  
    id SERIAL PRIMARY KEY,  
    pa PCPATCH(1)  
);
```

点云入库

PDAL入库

Pgpoint.json:

```
{  
  "pipeline": [  
    {  
      "type" : "readers.text" , --源数据  
      "filename":"/home/data/rds_test.csv",  
      "header":"x,y,z,reflectance,utc_time",  
      "spatialreference":"EPSG:4490"  
    },  
    {  
      "type":"filters.chipper",  
      "capacity" :400 --单个pcpatch含多少个point  
    },  
    {  
      "type":"writers.pgpointcloud", --PostgreSQL + Ganos  
      "connection": "host= '192.168.1.1' dbname='pointcloud' user='dianyun'  
                  password='Dianyun123' port='3433'",  
      "table": "pointcloud_pc400",  
      "srid": "4490"  
    } ]}
```

Plugins supporting *.csv files found.

The file is too large: 1.96 GB. Showing in read-only mode.

```
| -2180817.1760587483,4381508.017167045,4076209.462578869,-821,1.5663287852500148E9  
| -2180817.130963512,4381508.067355784,4076209.5123898634,-804,1.5663287852500167E9  
| -2180817.2518439074,4381508.005566918,4076209.6136636436,-881,1.5663287852500184E9  
| -2180820.1408385723,4381506.066820826,4076210.5415347177,-1196,1.5663287852500203E9  
| -2180820.461963807,4381505.870459932,4076210.7067733184,-1146,1.5663287852500222E9  
| -2180820.5894906567,4381505.807994845,4076210.822745904,-1037,1.5663287852500238E9  
| -2180820.598039107,4381505.825703129,4076210.9008709844,-1134,1.5663287852500257E9  
| -2180820.5788523513,4381505.862302282,4076210.972030544,-902,1.5663287852500277E9  
| -2180820.5570191997,4381505.898636014,4076211.0351518434,-897,1.5663287852500293E9  
| -2180820.5304087293,4381505.9405161,4076211.104234701,-963,1.5663287852500312E9  
| -2180820.500126522,4381505.984556782,4076211.1706295903,-937,1.566328785250033E9  
| -2180820.4800947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.455006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.4200947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.385006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.3500947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.315006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.2800947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.245006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.2100947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.175006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.1400947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.105006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.0700947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.035006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.0000947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.-0.50006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.-0.1000947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.035006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.0700947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.105006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.1400947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.175006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.2100947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.245006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.2800947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.315006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.3500947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.385006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.4200947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.455006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.4800947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.515006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.5500947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.585006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.6200947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.655006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.6900947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.725006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.7600947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.795006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.8300947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.865006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.9000947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.935006722,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.9700947304,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.1000947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.135006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.1700947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.205006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.2400947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.275006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.3100947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.345006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.3800947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.415006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.4500947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.485006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.5200947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.555006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.5900947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.625006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.6600947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.695006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.7300947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.765006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.8000947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.835006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.8700947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.905006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.9400947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.975006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.1000947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.135006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.1700947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.205006722,4381506.020021169,4076211.2360556326,-920,1.5663287852500348E9  
| -2180820.2400947304,4381506.060072500,4076211.202725245,-870,1.5663287852500367E9  
| -2180820.275006722,4381506.020021169,4076211.2360
```

点云查询与管理



创建空间索引

```
CREATE INDEX pointcloud_pc400_idx ON pointcloud_pc400 USING GIST  
  (ST_BoundingDiagonalGeometry(pa) gist_geometry_ops_nd);
```

查询落入指定范围的所有点云数据

```
\set x random(-2180834.55325373, -2180607.70494644)  
\set y random(4381273.4162077, 4381565.42591964)  
\set z random(4076197.15295099, 4076559.92900519)
```

```
SELECT ST_Intersection3D(pa, ST_3DMakeBox(ST_MakePoint(:x-0.3, :y-0.3, :z-0.3),  
ST_MakePoint(:x+0.3, :y+0.3, :z+0.3)))  
FROM pointcloud_pc400  
WHERE ST_BoundingDiagonalGeometry(pa) &&&  
ST_3DMakeBox(ST_MakePoint(:x-0.3, :y-0.3, :z-0.3), ST_MakePoint(:x+0.3, :y+0.3, :z+0.3));
```

Ganos时空数据库引擎

欢迎钉钉扫码入群
Ganos时空云计算

