SPATIAL

April 2009

Oracle Spatial User

Conference

ORACLE®

SPATIAL

Oracle Spatial User Conference

April 23, 2009

Tampa Marriott Waterside Hotel

Tampa, Florida USA

ORACLE
SPATIAL

April 2009

Oracle Spatial User Conference

Oracle Spatial 11g: Technical Update

Dr. Siva Ravada
Director of Development
Oracle

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remain at the sole discretion of Oracle.

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Spatial Feature Enhancements

- GeoRaster
- New Functions in core Spatial
- Geocoding Enhancements
- Web Services
- Network Data Model
- Routing Enhancements

The GeoRaster Java API

- The GeoRaster JAVA API
  - *oracle.spatial.georaster:* Provides a complete mapping of the SDO_GEORASTER object type and its metadata to Java objects, and provides support for the core GeoRaster features
  - *oracle.spatial.georaster.sql:* Provides a Java wrapper of the GeoRaster PL/SQL API for some server-side operations
  - *oracle.spatial.georaster.image:* Provides support for generating Java images from a GeoRaster object and for processing the images

- The core georaster package and the sql package are implemented in pure Java. It doesn't depend upon Java 2D and JAI.
- The image package is based on Java 2D and JAI. This allows users to leverage all the strength and advanced capabilities from Java 2D and JAI. Users can easily develop web applications and other image processing applications.

The GeoRaster Java API (conti.)

- Sample Applications: using this Java API, users can easily develop ETL tools and applications, particularly web-based applications. Source code is provided for four sample applications built with this new Java API

Tools.java	Loader.java	Viewer.java	Exporter.java

Raster CS Transformations

SDO_GEOR.reproject: transform GeoRaster raster data from one projection to another projection. All oracle spatial supported coordinate systems are supported. Five re sampling methods are supported

- **NN, Bilinear, Cubic, Avarage4 and Average16.**

Supports two options

- **Reproject persistently. Reprojects a GeoRaster object and stores the result as a new GeoRaster.**
- **Reproject on-the-fly. This is equivalent to getRasterSubset except the window-based cropping result is transformed into a different projection. The result is stored as a single BLOB.**
- **Both options support window and pyramid level based query (subsetting using AOI)**

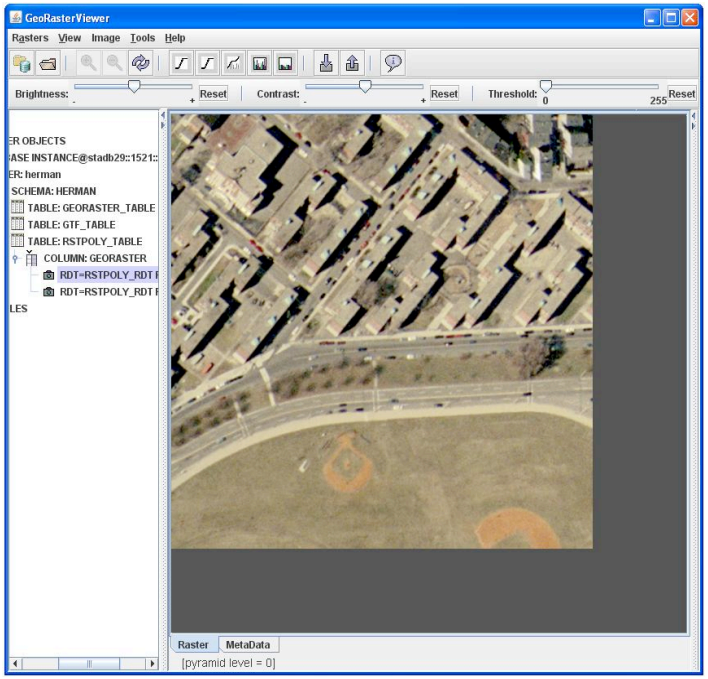
ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

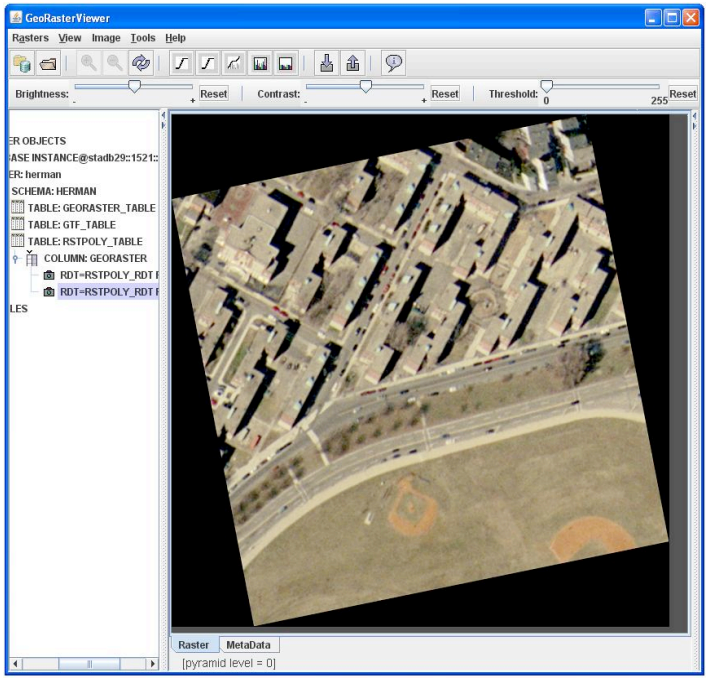
Raster Reprojection – An Example

sdo_geor.reproject (gr1, 'resampling=cubic', 'blocksize=(256,256,3) interleaving=BIP', 26988, gr2);



From: SRID 26986

"NAD83 / Massachusetts Island"



To: SRID 26988

"NAD83 / Michigan North"

ORACLE SPATIAL

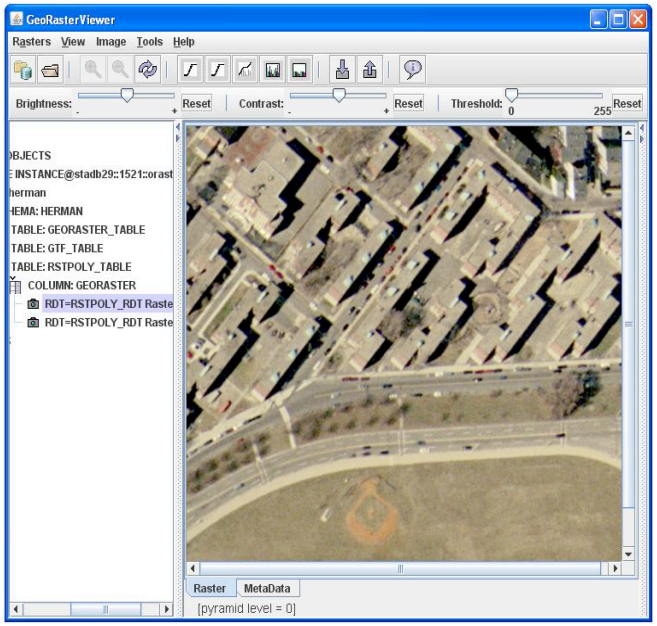
April 2009

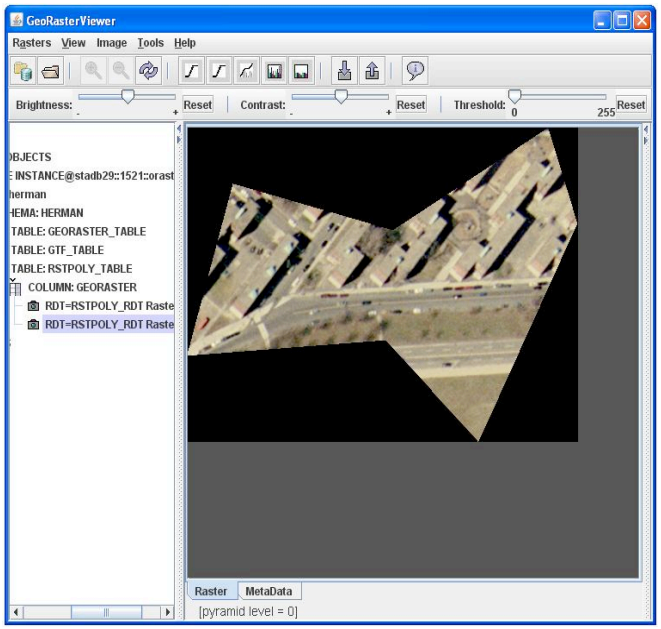
Oracle Spatial User Conference

Polygon-based Raster Clipping

SDO_GEOR.getRasterSusbet is enhanced:

- In previous releases, only the MBR (rectangle) of the polygon is used.
- Now, it allows users to clip the query result along the polygon (irregular) boundary. For example,





Blocking Size Optimizer – Minimize Padding

- GeoRaster supports regular blocking and padding is applied to the right and lower boundary blocks if necessary. However, padding wastes some storage space
- The blocking size optimizer would automatically optimize the blocking size based on the GeoRaster dimension sizes and pyramid levels so that the padding can be minimized:
  - "blocking=optimalPadding" in storageParam. This applies to any functions which use storageParam, such as mosaic, subset, scaleCopy, mergeLayers, reproject. It automatically adjusts blocking sizes.
  - SDO_GEOR_UTL.calcOptimizedBlockSize Users can use it to pre-compute the optimal blocking sizes and then apply.

ORACLE SPATIAL

April 2009

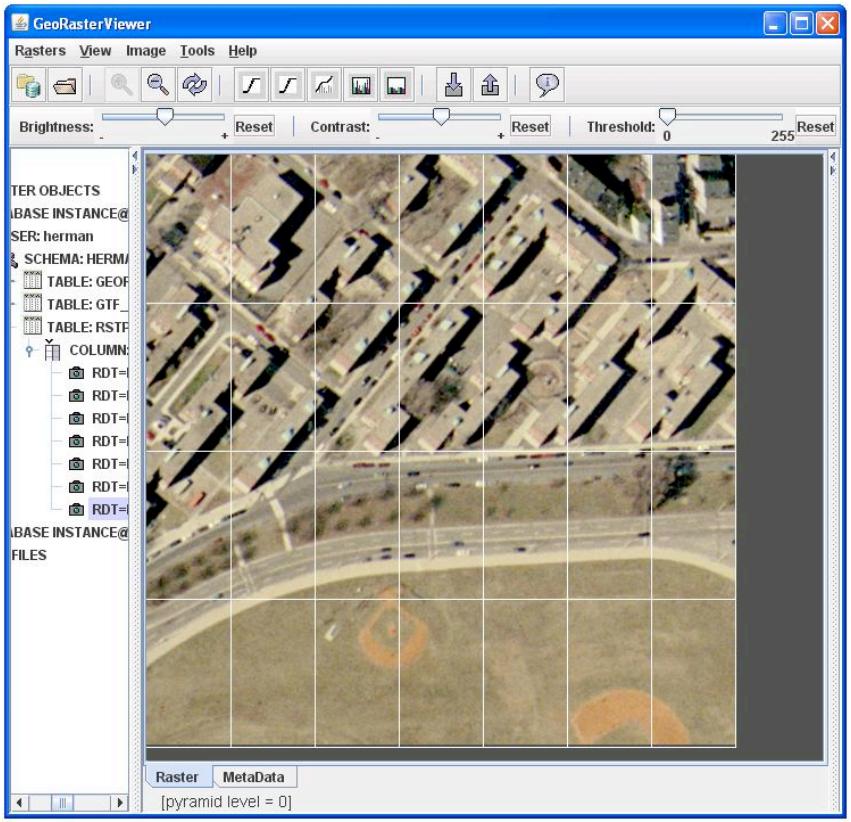
Oracle Spatial User Conference

Blocking Size Optimizer – An Example

-- original image size 518x518x3 blocked into 256x256x3

```
sdo_geor.changeFormatCopy
    (gr1,
     'blocking=optimalPadding
      blocksize=(128,96,1)',
      gr2);
```

-- result image blocked into 130x74, near zero padding



Raster Interpolations – Coverage Evaluate

- A GeoRaster object can represent a continuous grid coverage thus the need for evaluate function
- Given any point within the spatial extent of the GeoRaster object, two new functions are provided to return the value at that point based on cell value interpolations:
  - SDO_GEOR.evaluate: returns interpolated value in the celldepth of the GeoRaster object
  - SDO_GEOR.evaluateDouble: always returns the interpolated value in double precision
- Five interpolation methods: NN (=”none” and default), Bilinear, Cubic, Average4, Average16; they all use the neighboring cell values and are the same as used in GeoRaster re sampling
- NODATA: either considered (TRUE) or not considered (FALSE) during interpolations

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Advanced Georeferencing Using GCP

- GCP: stands for Ground Control Points
- GCP Model: GeoRaster supports a generic GCP model. In the current release, 2D cell coordinates, 2D and 3D model coordinates are supported.
- GCP Storage: GeoRaster defines a GCP XML schema and can (optionally) store GCP natively in the metadata of GeoRaster objects.
- GCP Manipulation: GeoRaster provides a set of update and query functions to manipulate GCP's and related data
    - SDO_GEOR.getControlPoint
    - SDO_GEOR.setControlPoint
    - SDO_GEOR.deleteControlPoint

Advanced Georeferencing Using GCP (cont.)

- Supports georeferencing using GCP's:
  - SDO_GEOR.georeference
  - It generates functional fitting models using GCP points, which are either retrieved from the GeoRaster object or provided by user when they call the function.
- Supports the following functional fitting methods (models created by sdo_geor.georeference using GCP's):
  - Affine
  - QuadraticPolynomial
  - CubicPolynomial
  - DLT
  - QuadraticRational
  - RPC

Other Enhancements

- SDO_GEOR.setModelCoordLocation: This function allows users to change the cell coordinate system from CENTER to UPPERLEFT or the reverse.
- It only applies to georeferenced GeoRaster objects and will automatically adjust the functional fitting coefficients of the GeoRaster SRS accordingly to reflect the change
  - to make sure the relationship between cell coordinates and model model coordinates doesn't change

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

A new ETL tool – the GDAL GeoRaster Driver

- GDAL is the best open source geospatial ETL tool/API for raster data. It now natively supports importing and exporting many formats, to/from SDO_GEORASTER, including GeoTIFF, JPEG2000, ECW, NITF, HDF, NetCDF, ERDAS IMG, USGS DEM, SPOT, and more.
- GDAL is written in C++, and runs much faster than Oracle's JAVA GeoRaster loader/exporter. Oracle's loader/exporter is based on SUN's JAI libraries.
- It provides C/C++, Java, Python API for accessing GeoRaster
- It provides many tools. Two of them are:
  - gdal_translate – utility to translate raster formats to/from GeoRaster objects
  - gdalinfo – utility to view information about a raster, such as a GeoRaster object

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Spatial Feature Enhancements

- GeoRaster
- New Functions in core Spatial
- Geocoding Enhancements
- Web Services
- Network Data Model
- Routing Enhancements

ORACLE SPATIAL

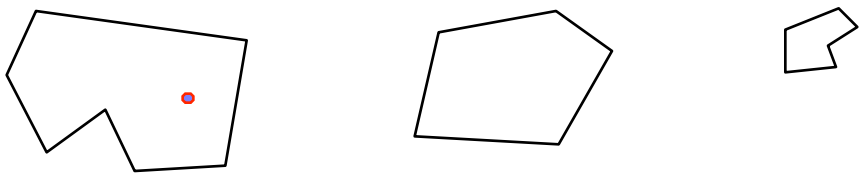
April 2009

Oracle Spatial User Conference

Interior Point

SDO_UTIL.INTERIOR_POINT (geom, tol)
RETURNS SDO_GEOMETRY

- This will compute a point that is interior to the given polygon
- Input geom should have a type 2003 or 2007
- Other geometry types will raise an error
- The tol is the tolerance at which the given geometry is valid
- An interior point is returned which is guaranteed to be INSIDE the given polygon
- This is different from sdo_geom.point_onsurface which is not guaranteed to return a point INSIDE the given geometry



Delaunay Triangulation

SDO_GEOM.SDO_TRIANGULATE (geom, tol)
RETURNS SDO_GEOMETRY

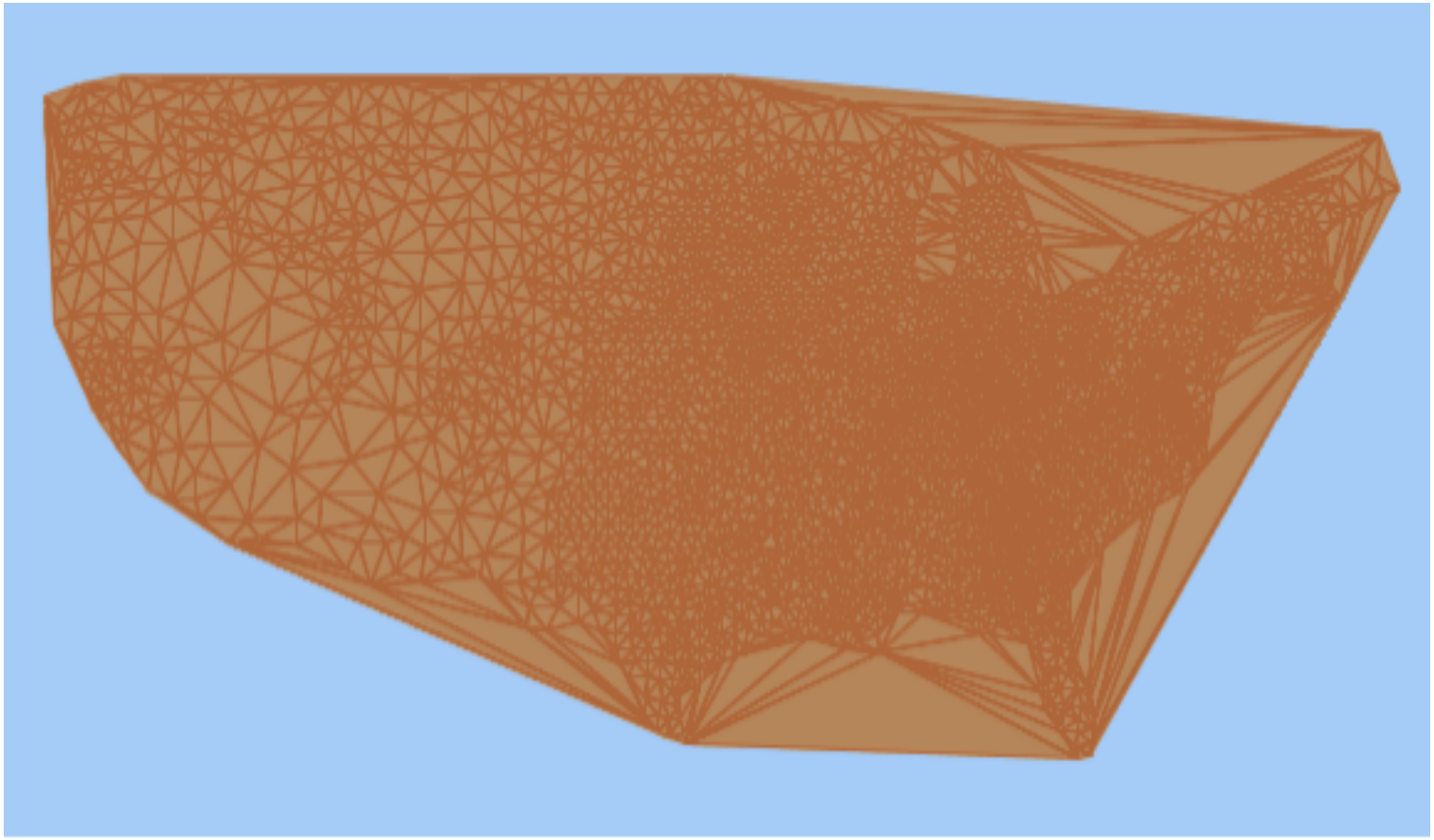
- Returns a geometry with triangular elements that result from Delaunay triangulation of the input geometry
- Input geometry is typically a point set; but any geometry is accepted
- The tol is the tolerance at which the given geometry is valid
- Different from SDO_TIN which is suitable for large collections of points (millions of points)

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Delaunay Triangulation



SDO_AGGR_SET_UNION

SDO_AGGR_SET_UNION
(Geometry mdsys.SDO_Geometry_array, tol number)
RETURN SDO_GEOMETRY

- MDSYS.SDO_Geometry_array is a VARRAY of SDO_GEOMETRY type
- This function takes a VARRAY of SDO_GEOMETRY objects as input and returns the Aggregate Union of all the geometry objects in the VARRAY
- This is a faster way to create UNION of several Geometry objects (compared to SDO_AGGR_UNION)
- USAGE
  - Since this requires a VARRAY of SDO_GEOMETRY as input, this function is not as flexible as the SDO_AGGR_UNION function which can be used with any arbitrary SQL GROUP BY statements
  - This function is useful when the geometries to be grouped are easily gathered into a collection

SDO_AGGR_SET_UNION

- For example, to find the union of all geometries in the cola_markets table except cola_d (in this case, cola_a, cola_b, and cola_c geometries), users need to write this piece of code:

```
create or replace function Set_Geometry return SDO_GEOMETRY_ARRAY deterministic AS
type                cursor_type is REF CURSOR;
query_crs              cursor_type ;
g sdo_geometry;
GeometryArr    sdo_geometry_array;
begin
  GeometryArr := SDO_GEOMETRY_ARRAY();
  OPEN query_crs FOR 'select shape FROM cola_markets c WHERE c.name <> ‘‘cola_d’’ ';
   LOOP
    FETCH query_crs into g;
     EXIT when query_crs%NOTFOUND ;
     GeometryArr.extend;
     GeometryArr(GeometryArr.count) := g;
   END LOOP;
   return GeometryArr;
end;
/
```

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

SDO_AGGR_SET_UNION

- Use the Set_Geometry function to compute the Aggregate Union with this SQL

  select SDO_AGGR_SET_UNION(Set_Geometry) from dual;
- When to use SDO_AGGR_UNION/SDO_AGGR_SET_UNION
  - The SDO_AGGR_UNION is SQL aggregate function: hence it very flexible and can be used with complex SQL group by clauses
  - But it is also slower compared SDO_AGGR_SET_UNION due to the overhead of SQL aggregation framework
  - When performance is not the highest priority, but flexibility is requried, we recommend the SDO_AGGR_UNION function
  - When the SQL used to collect the geometries is simple, use the SDO_AGGR_SET_UNION as in the above example as this is usually an order of magnitude faster than using SDO_AGGR_UNION
  - Both SDO_AGGR_UNION and SDO_AGGR_SET_UNION work for Geodetic and non-Geodetic data

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

SDO_AGGR_SET_UNION

- Performance Comparison
- States Table with 50 geometries
  - AGGR_SET_UNION: 3.3 Seconds
  - SDO_AGGR_UNION: 27.55 Seconds
- Counties Table with 3230 geometries
  - AGGR_SET_UNION: 44 Seconds
  - SDO_AGGR_UNION: 26 Minutes

Concave Hull

SDO_GEOM.CONCAVEHULL (geom, tol)
RETURNS SDO_GEOMETRY

- This will compute a polygon that represents the area occupied by a set of points in the plane
- Input geometry should be a 2D Geometry of any type
  - This is typically a multipoint geometry
- The tol is the tolerance at which the given geometry is valid
- A CONCAVEHULL is returned which is guaranteed to be a valid polygon (gtype 2003/2007)
- This is different from CONVEXHULL

ORACLE
SPATIAL

April 2009

Oracle Spatial User

Conference

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

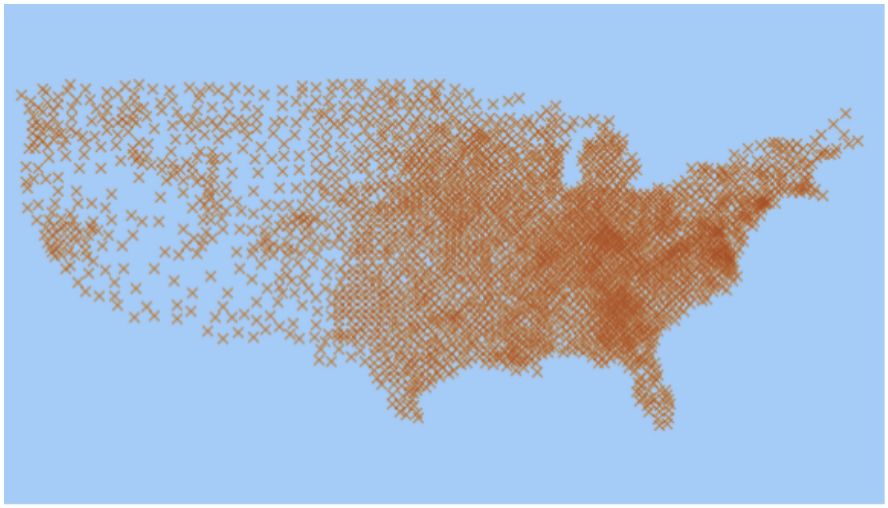
Concave Hull

ORACLE SPATIAL

Concave Hull

April 2009

Oracle Spatial User

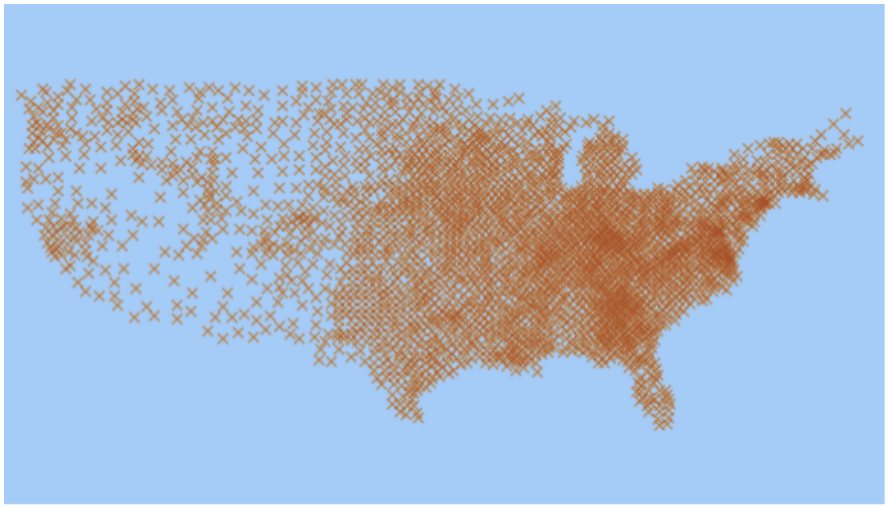


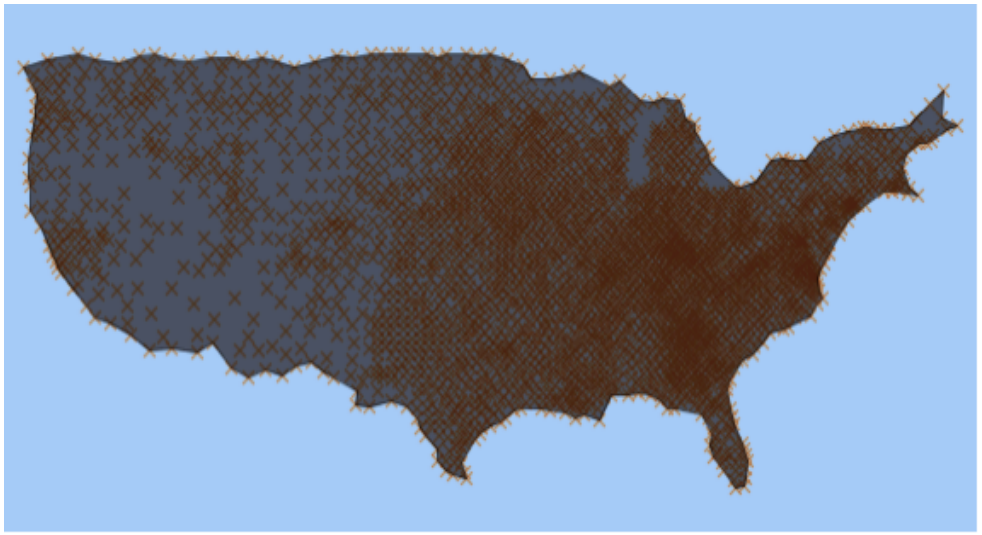
ORACLE SPATIAL

Concave Hull

April 2009

Oracle Spatial User



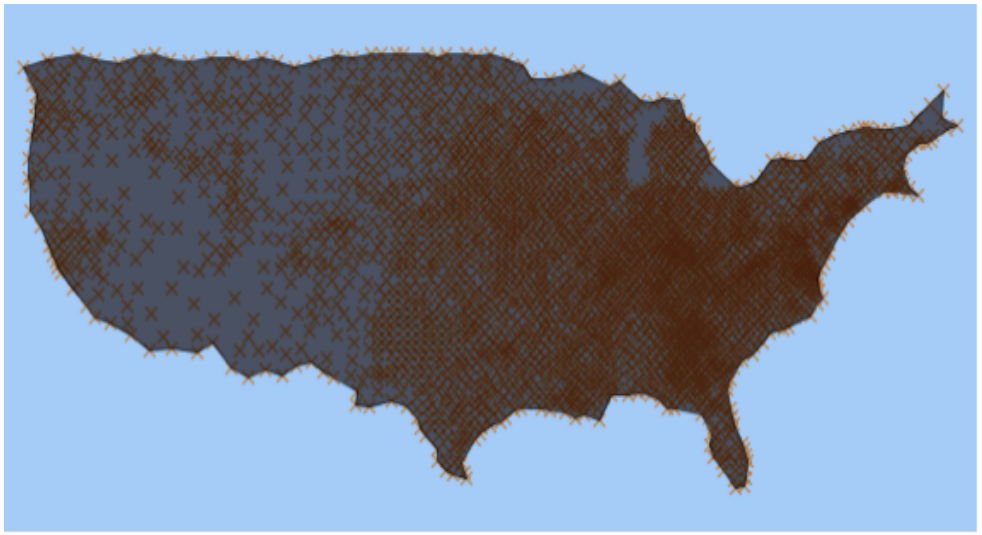


ORACLE SPATIAL

April 2009

Oracle Spatial User

Convex Hull vs Concave Hull



ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Faster Coordinate System Transformations

- SDO_CS.Transform is up to 10 times faster
  - This speedup is seen if 1000s of transformations are done in a session
  - In the prior releases, the transformation context was created for each transform() call
  - now we use that context between transform() calls with the same source and target SRIDs
- SDO_CS.Transform_Layer is not affected

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Cross-Endian support for TTS and Spatial Index

- SDO_UTIL. INITIALIZE_INDEXES_FOR_TTS now automatically fixes the index if the TTS export is done in a different endian format than the target format
- RMAN process does this conversion for the table data, but not for Spatial indexes

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

SDO_PC and SDO_TIN Storage enhancements

- A RESULT table for SDO_PC is used map the input points to the PTN_ID and POINT_ID columns
- Sometimes, this process of copying the input table and generating this RESULT table can be very expensive
  - If there are 200M points in the input table, this process can take more than 10 times the cost to create the SDO_PC blocks
- A RESULT table is not required if the PK of the input table (the RID value) is numeric
- If it is not numeric, then the RESULT table is required

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

SDO_PC and SDO_TIN Storage enhancements

- The RESULT table for SDO_PC was required to have the same number of columns as the INPUT table in addition to the PTN_ID and POINT_ID columns
- This is not required in the next release
  - The result table can only have three columns: RID, PTN_ID, and POINT_ID
  - This table can be used as a JOIN table to extract the additional attributes of the Points in the PC by joining back to the INPUT table
- Same storage enhancement is done for SDO_TIN as well

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Spatial Feature Enhancements

- GeoRaster
- New Functions in core Spatial
- Geocoding Enhancements
- Web Services
- Network Data Model
- Routing Enhancements

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Point Address Geocoding

- Point Addressing data has an exact long/lat for each address
  - This is different from the range based addressing where each road segment has an address range
- Support for this feature requires a new data table in addition to the current set of Geocoder tables
- New table: GC_ADDRESS_POINT_NVT
- No interface change required, if this table exists we use it to refine the result using the exact long/lat provided in the table

Point Address Geocoding Table

- GC_ADDRESS_POINT_NVT (

```
ADDRESS_POINT_ID        NUMBER(10)
ROAD_ID                 NUMBER    (req)
ROAD_SEGMENT_ID         NUMBER(10) (req)
SIDE                    VARCHAR2(1)  (req)
PERCENT                 NUMBER    (req)
LANG_CODE               VARCHAR2(3)
HOUSE_NUMBER            VARCHAR2(600 CHAR) (req)
HN_TYPE                 NUMBER(2),
ADDR_LONG               NUMBER(10) (req)
ADDR_LAT                NUMBER(10) (req)
BUILDING_NAME           VARCHAR2(600 CHAR)
ARRIVAL_LINK_ID         NUMBER(10)
ARRIVAL_SIDE            VARCHAR2(1)
DISPLAY_LONG            NUMBER(10)
DISPLAY_LAT             NUMBER(10)
ENHANCED                VARCHAR2(1)
COUNTRY_CODE_2          VARCHAR2(2) )
```

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Point Address Geocoding Data

- Customers can buy this new data set from NAVTEQ
- NAVTEQ Point Addressing data product
- This is in addition to the NAVTEQ Geocoding data in Oracle Data Format (ODF)

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Spatial Feature Enhancements

- GeoRaster
- New Functions in core Spatial
- Storage improvements for SDO_PC/SDO_TIN
- Geocoding Enhancements
- Web Services
- Network Data Model
- Routing Enhancements

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Web Services Enhancements

- Full support for DB txns on WFS feature tables
- Full Support for Workspace Manager and WFS Feature tables
- WFS 1.1 support

Support For DB Trasactionsx on WFS-T Feature Tables

- In 11gR1, if a table is WFS-T enabled using the mdsys.sdo_wfs_lock.registerFeatureTable procedure then direct updates/deletes are not allowed on that table (via SQL)
- Updates/Deletes are only allowed via WFS-T transactions
- We now allow updates/deletes on the WFS-T feature tables via SQL
  - Need to execute this procedure in a session to enable this: sdo_wfs_lock.enableDBtxns

Support for WorkSpace Manager and WFS

- In 11gR1, WFS and Workspace Manager work together for some cases
  - We support publishing WFS features from different workspaces (READONLY mode)
- Now we fully support updatable workspaces and WFS transactions
  - Users can now do WFS transactions into workspace enabled features tables
  - Needs sdo_wfs_lock.enableDBtxns procedure in a session before any Workspace maintenance operations, such as refreshing a workspace or merging workspace are done

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Spatial Feature Enhancements

- GeoRaster
- New Functions in core Spatial
- Geocoding Enhancements
- Web Services
- Network Data Model
- Routing Enhancements

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Multiple Cost Support in Path Analysis

- Path/Subpath analysis now supports multiple costs in a single analysis
- First cost is the one NDM optimizes
- Examples
  - Shortest distance paths with travel time
  - Fastest paths with travel distance

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

User Object Support in Network Constraints

- User defined object can be accumulated and used in network constraints during network analysis
- Available in network constraint implementation
- Examples:
  - Previously visited link/node information

Network Synchronization

- NDM now captures changes of network made in the database
- Users can synchronize these changes to network partitions in DB and in-memory
- This can be done incrementally instead of reading the whole network again from the DB

Lean User Data Support

- User data contains application information that is needed in network constraints or cost calculators
- Stored in network partition blobs
- New representation has smaller footprints than 11gR1 representation
- More partitions can fit into memory, so requires less memory to run the network engine

Logical Network Partitioning

- A new partitioning utility to help partition logical networks
- Minimize the number of links among network partitions
- Load balancing (no. of nodes/partition) with a predefined threshold
- This can partition very large networks (millions of nodes/links)

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Hierarchical A* shortest path Analysis

- Support A* shortest path algorithm
- Provide user defined heuristic cost function
- Support hierarchical shortest path analysis
- Better performance than Dijkstra Algorithm as less nodes are explored

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Traveling Sales Man (TSP) Analysis

- Minimum cost tour that includes all given nodes
- Support Points on Network as nodes to be visited
- Node visit order can be enforced using network constraints
- This is useful scheduling problems where a service representative has to visit a number of customers

Drive Time Polygon Generation

- A spatial representation (polygon) based on minimum cost network coverage
- Concavehull polygon or convexhull polygon (accuracy and performance)
- Example
  - Compute Drive Time Polygon (with travel time as link cost) from a service station
  - Use the polygon to determine if a given address can be reached within a given time using point in polygon operation

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Spatial Feature Enhancements

- GeoRaster
- New Functions in core Spatial
- Geocoding Enhancements
- Web Services
- Network Data Model
- Routing Enhancements

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Routing Engine

- Routing Application is now built on top of NDM LOD engine
- Routing Engine uses all the extensibility of the underlying NDM engine
- Truck routing is an example of customizing the routing engine specific to an application based on user data

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Truck Routing

- <?xml version="1.0" standalone="yes"?>
  <route_request id="8" route_preference="shortest" road_preference="highway" vehicle_type="truck" return_driving_directions="true" distance_unit="mile" time_unit="hour" return_route_geometry="false" >
- Route Request vehicle_type
  - (auto|truck) optional, defaults to auto
- Route Request truck_type
  - truck_type: (delivery|public|trailer) optional, no default

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Truck Routing

- truck_height: (positive float) optional, no default
- truck_length: (positive float) optional, no default
- truck_per_axle_weight: (positive float) optional, no default
- truck_weight: (positive float) optional, no default
- truck_width: (positive float) optional, no default
- length_unit: (metric|us) optional, default US
- weight_unit: (metric|us) optional, default US
- Truck height, length and width are specified in length_unit units
- Truck per axle weight and weight and specified in weight_unit units

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

Truck Routing Data

- Table definition for trucking data table
- This table contains data from the NVT_TRANSPORT table from NAVTEQ
  - NAVTEQ Transport Product
- Trucking User Data table

```
ROUTER_TRUCK_DATA (
 edge_id NUMBER,
 maintype NUMBER(2),
 subtype NUMBER(2),
 value NUMBER(2));
```

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

ORACLE SPATIAL

April 2009

Oracle Spatial User Conference

D E M O N S T R A T I O N

Truck Routing

ORACLE
SPATIAL

April 2009

Oracle Spatial User

Conference

QUESTIONS

&

ANSWERS


ORACLE®
SPATIAL