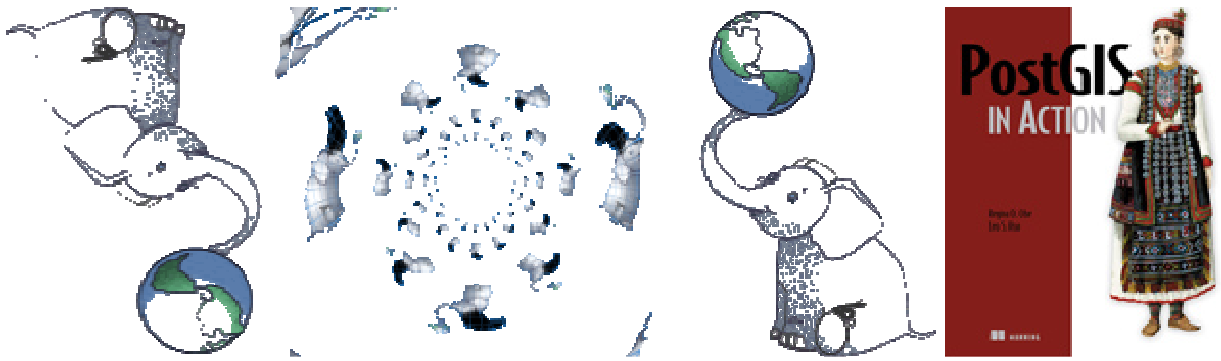
PostGIS 2.0 3D and Raster support enhancements

Regina Obe and Leo Hsu
http://www.postgis.us
http://www.bostongis.com
http://www.postgresonline.com
http://www.paragoncorporation.com



PostGIS goes 3D
New in PostGIS 2.0

- Polyhedral Surfaces and TINS
- Affine Transform support for all 3D
- `ST_AsGML, ST_GeomFromGML` for 3D Polyhedral and TINS
- New 3D relationship /measurement functions currently work for all 3D except for TINS –

  `ST_3DDistance, ST_3DDWithin, ST_3DIntersects, ST_3DClosestPoint,ST_3DLongestLine, ST_3DShortestLine, ST_3DMaxDistance`

3D Geometry
Polyhedral Surface

```
CREATE TABLE test3d(gid SERIAL PRIMARY KEY, geom geometry);

INSERT INTO test3d(geom)

VALUES('POLYHEDRALSURFACE(

((0 0 0,0 0 5,0 15 5,0 15 0, 0 0 0)),

((0 0 0,0 15 0,10 15 0,10 0 0, 0 0 0)),

((0 0 0,10 0 0,10 0 5,0 0 5, 0 0 0)),

((10 0 0,10 15 0,10 15 5,10 0 5, 10 0 0)),

((0 15 0,0 15 5,10 15 5,10 15 0,

0 15 0)))'::geometry

);
```

3D Geometry
Triangular Irregular Network (TIN)

```
INSERT INTO test3d(geom)
VALUES('TIN(((1 2 3,4 5 6,7 8 9,1 2 3)),
    ((10 11 12,13 14 15,16 17 18,10 11 12)),
    ((19 20 21,22 23 24,25 26 27,19 20 21)))'::geometry);
```

3D Geometry
ST_AsGML

```
SELECT gid, ST_AsGML(3, geom) As ogml FROM test3d;


-- result -
1|<gml:PolyhedralSurface><gml:polygonPatches>
  <gml:PolygonPatch>...</gml:PolyhedralSurface>
2|<gml:Tin><gml:trianglePatches>
 <gml:Triangle>...</gml:Triangle>...
 </gml:trianglePatches></gml:Tin>
```

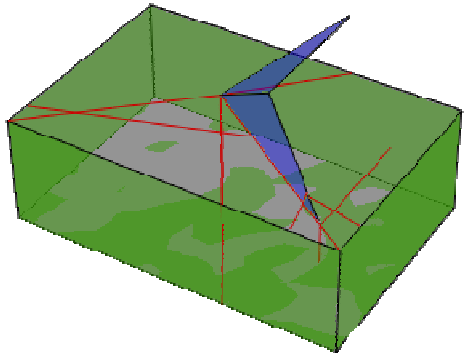
PostGIS goes 3D
Open Source Desktop Viewing

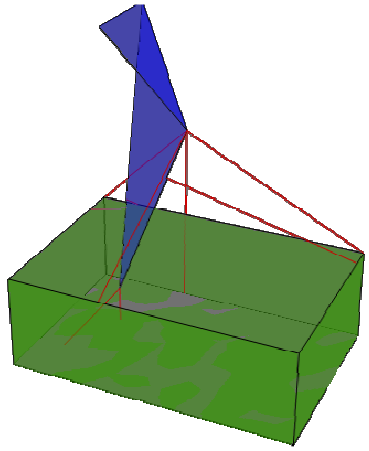
- None yet, but GvSig upcoming version will have at least 3D support to view simple 3D (not TINS/Polyhedral). With ST_3DShortestLine and Geometry Dump tricks can get to display Polyhedral Surfaces. Using PostGIS Affine Transform functions e.g ST_Affine*, ST_Rotate*, ST_Translate – can move TINS/Polyhedral Surfaces and other 3D geometries
- Snapshots from Nicklas Avén's PostGIS post:

http://www.postgis.org/pipermail/postgis-users/2011-January/028658.html

With associated SQL:

http://www.postgis.org/pipermail/postgis-devel/2011-January/011302.html





PostGIS 2.0 /PostgreSQL 3D Use Case

Resource Management

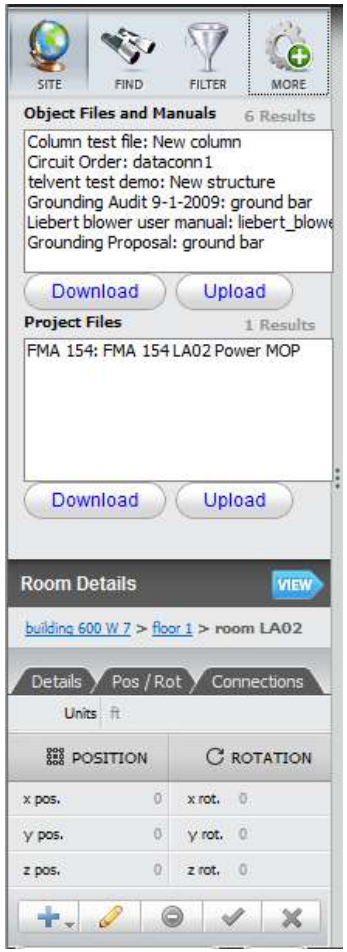
Arrival 3D (new venture): http://www.arrival3d.com

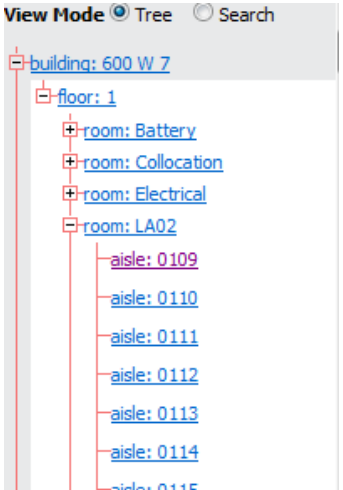
Collaboration with Network Optics Engineer and VRML/X3D Expert.

Web-based PHP/JQuery/X3D

- X3D scenes autogenerated from database objects viewed with BS Contact X3D web-viewer
- PostgreSQL 9.0 resource repository with LTree for managing resource node relationships / incorporating PostGIS for more analytics (right now just 3D resource node points and a PostgreSQL inventory model server to place the models centered at the nodes)
- Cataloging cable paths, summaries of terminations etc., Closest point for determining optimal paths.
- Integration with existing Telecom Provisioning and Alarm Systems

PostGIS / PostgreSQL 3D Use Case





Filtering Objects on the fly

PostGIS 2.0 Raster

For more information:
http://www.postgis.org/documentation/manual-svn/RT_reference.html

Key Features

- Georeferenced rasters in the database uses GIST index like Geometry
- New data type called “raster” – one row = raster tile,
  One table = raster coverage
- Python Loader utilities built on GDAL – can load in any kind of raster and bulk load many raster files
- Intersections, Intersects with Geometry data
- Extrude raster regions as geometry
- Ability to create pyramid (overview) tables on load
- Analysis – do averaging of pixel ranges in areas, extrude individual pixels
- Can export raster data to any raster formats supported by GDAL
- In place edit on rasters and ability to create new rasters
- Single band MapAlgebra, just completed this week.
- Rendering tools already available and undergoing fine-tuning

PostGIS 2.0 Raster
Load Data Basic

This generates an sql file that will load all the jpegs in current folder into a new table called aerials.boston (Massachusetts State Plane Meters (26986)), with each raster record 100x100 pixels width / height. The `-F` will create a column called filename in the table which will list
The jpeg file each raster record tile came from.
The `-I` will create a gist index on convex hull of the raster.

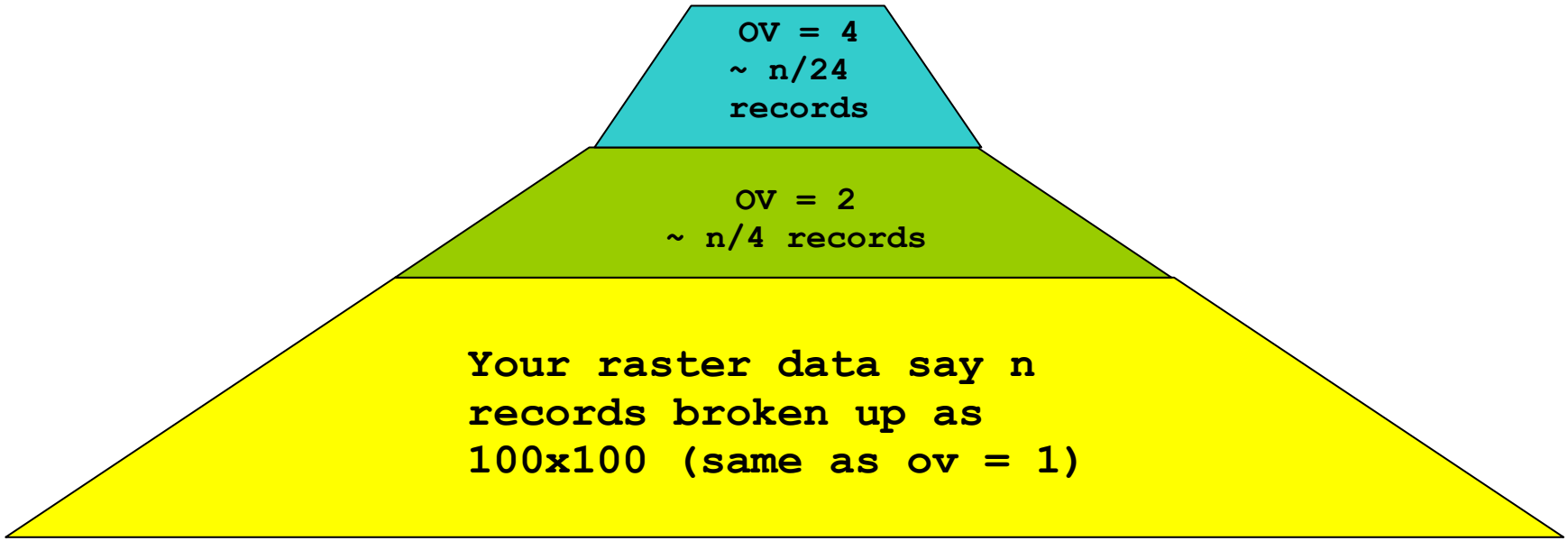
```
python raster2pgsql.py -r *.jpg \
   -t aerials.boston -s 26986 -k 100x100 \
   -F -I -o aerials.sql
```

This runs the script loading the data into mygisdb

```
psql -d mygisdb -f aerials.sql
```

PostGIS 2.0 Raster
Raster Overviews (aka Pyramid)

These are lower resolution raster tables of your primary tables. These are registered in a table called: `raster_overviews` and created using the loader with –l level switch
It works kind of like this: (assuming all you set your overviews
as same block size as your regular)



PostGIS 2.0 Raster
Load Data Overview (Pyramid)

This generates an sql file that will load all the jpegs in current folder into a new table called aerials.o2_boston (Massachusetts State Plane Meters (26986)) for our table aerials.boston, with each raster record
100x100 pixels width / height but lower res.
The –F will create a column called filename in the table
which will list
The jpeg file each raster record tile came from.
The `-l` will create an overview table for aerials.boston with ov level (in this case `4`)
Note: The table will be called `aerials.o_4_boston (`not `aerials.boston`), but will be Registered in `raster_overviews` table and associated with `aerials.boston`

```
python raster2pgsql.py -r *.jpg \
   -t aerials.boston -s 26986 -l 4 -k 100x100 \
   -F -I -o aerials_overview4.sql
```

This runs the script loading the data into mygisdb

```
psql -d mygisdb -f aerials_overview4.sql
```

PostGIS 2.0 Raster
Regular to Overviews

Overviews are good for zoom out and also doing faster but less high res calculations:

For our small sample:

```
--result: 845 records
SELECT COUNT(*) FROM aerials.o_4_boston;
--result: 3,125 records
SELECT COUNT(*) FROM aerials.o_2_boston;
--result: 20,000 records
SELECT COUNT(*) FROM aerials.boston;
```

PostGIS 2.0 Raster
Intersects with geometry

How many parcels intersect our loaded raster tiles

```
SELECT COUNT(DISTINCT p.map_id)
from massgis.parcels_boston As p INNER JOIN aerials.boston As r
        ON ST_Intersects(p.geom, r.rast);
```

PostGIS 2.0 Raster Intersection with geometry

Pick a parcel / show average pixel value – faster to work with lower res but less accurate

```
-- band 3 average for overview - (avg pixval: 89.12 – 991 ms)
SELECT SUM(ST_Area((gv).geom)*(gv).val)/SUM(ST_Area((gv).geom))
FROM (
        SELECT ST_Intersection(r.rast,3, p.geom) As gv
        FROM massgis.parcels_boston As p INNER JOIN aerials.o_4_boston As r
        ON ST_Intersects(p.geom, r.rast)
WHERE p.map_id = '2010306000') As foo;

-- band 3 average for overview - (avg pixval: 136.7 – 3 secs)
SELECT SUM(ST_Area((gv).geom)*(gv).val)/SUM(ST_Area((gv).geom))
FROM (
        SELECT ST_Intersection(r.rast,3, p.geom) As gv
        FROM massgis.parcels_boston As p INNER JOIN aerials.o_2_boston As r
        ON ST_Intersects(p.geom, r.rast)
WHERE p.map_id = '2010306000') As foo;

-- band 3 average for full - (avg pixval: 137.8 -- 12 secs)
SELECT SUM(ST_Area((gv).geom)*(gv).val)/SUM(ST_Area((gv).geom))
FROM (
        SELECT ST_Intersection(r.rast,3, p.geom) As gv
        FROM massgis.parcels_boston As p INNER JOIN aerials.boston As r
        ON ST_Intersects(p.geom, r.rast)
WHERE p.map_id = '2010306000') As foo;
```

Open Source Tools that work with PostGIS raster

GDAL – 1.8+ has PostGIS raster driver (looking for funding to improve performance)

http://trac.osgeo.org/postgis/wiki/WKTRaster/GDALDriverSpecificationWorking

QGIS beta support now via plug-in

GvSig beta support will be integrated in next release available as a plug-in now for current (but only works with older WKT Raster (0.1.6))

MapServer – the first to work – via GDAL driver 1.7+ (better to use 1.8+ GDAL driver)

Mapserver Layer

```
LAYER
    NAME boston_aerials
    TYPE raster
    STATUS ON
    DATA "PG:host=‘localhost' port='5432'
     dbname='ma' user='ma' password=‘test'
     schema='aerials' table='o_2_boston' mode='2'"
    PROJECTION
     "init=epsg:26986"
    END
 END
```

Using aerials.o_2_boston

Using aerials.boston

PostGIS in Action
use promo: postgis40

Get 40% off PostGIS in Action purchase if buy directly from Manning:
http://www.postgis.us
postgis40 is good till March 3rd

Questions