

# GIS-Technologie-News

SOGI Informationsblatt 4/2010

## Vektor-Raster-Analyse in Datenbanken

Räumliche Datenbanken sind aus modernen Geodateninfrastrukturen nicht mehr wegzudenken. Während heute primär Attribut- und Vektordaten in räumlichen Datenbanken gespeichert und analysiert werden, war die Ablage von Rasterdaten innerhalb von räumlichen Datenbanken oft umstritten. Die reine Datenspeicherung ohne sinnvolle Analysefunktionen war für viele Datenbankadministratoren nicht Anreiz genug, sich mit den um ein Vielfaches grösseren Datenmengen, die Rasterdaten mit sich bringen, abzumühen.

Die zunehmende Verfügbarkeit von Raster-Analysefunktionen (direkt über SQL-Befehle), sowie die Vorteile der Datenadministration, sind aber nun Grund genug die Situation für Rasterdaten neu zu überdenken. Dies vielleicht nicht für alle Anwendungszwecke, sondern primär für solche Ebenen, auf denen kombinierte Vektor- und Rasteranalysen direkt in der Datenbank durchgeführt werden sollen.

Die Speicherung von Rasterdaten innerhalb von Datenbanken ist nicht neu. Verschiedene kommerzielle GIS-Firmen oder kommerzielle Geodatenbanken bieten dies schon seit einigen Jahren an. In den meisten Fällen werden die Analysen und Datenmanipulationen nicht direkt in der Datenbank durchgeführt, sondern mit eigener Software ausserhalb der Datenbank.

Mit der Einführung des PostGIS-Rasterdatentyps und den damit verbundenen Analysemöglichkeiten können aber neu auch komplexe Analysen direkt in der Geodatenbank durchgeführt werden, ohne eigene Analysesoftware im GIS-Client entwickeln zu müssen. PostGIS bietet sich damit als Backend an für Desktop-GIS, Web-GIS, Web Processing Services (WPS), ModelBuilder/FME-ähnliche Umgebungen oder Berichtgeneratoren. Angesprochen sind damit weniger die Endanwender sondern vor allem GIS-Entwickler und Datenbankadministratoren.

Die Vorteile von kombinierten Vektor-Rasteranalysen und von Rasteranalysen direkt in der Datenbank sind:

- Die gleichen oder ähnliche SQL-Befehle können auf Vektor und Raster angewendet werden
- In gemeinsamen SQL-Befehlen können Vektor- und Rasteroperatoren und -Funktionen gleichzeitig verwendet werden
- Viele GIS-Analysen können direkt in der Datenbank ausgeführt werden, ohne eine zusätzliche GIS-Software zu benötigen
- Rastermanipulationen und Analysen können Teil von Datenbanktransaktionen und Triggern sein oder von Triggern ausgelöst werden
- Eigene, komplexere Raster/Vektoranalysen können mit Hilfe von Stored Procedures in verschiedenen Programmier-Sprachen direkt in der Datenbank implementiert werden (z.b. PL/pgSQL, PL/Perl, PL/Python, C, Java, etc.)

## Das PostGIS Raster-Projekt

Gestützt auf Sponsorengelder und das Know-How von PostGIS- und OSGeo-Kernentwicklern wurde das PostGIS Raster-Projekt mit folgenden Zielen ins Leben gerufen:

- Es sollen Operationen auf beliebigen Geodaten ausgeführt werden können, (fast) ohne Rücksicht auf deren Repräsentation (Vektor, Raster, Grid, TIN, etc.).
- Alle Raster-Operationen werden in der Datenbank durchgeführt.

Folgende Überlegungen wurden zur Implementation gemacht:

- Rasterdaten (inkl. Header wie Pixel-Grösse am Boden, Breite und Höhe des Rasters, die Koordinaten-Referenzsystem-Nummer, etc.) werden in einem einzigen Feld verwaltet (keine zweite Tabelle wie z.B. bei Oracle).
- Rasterdaten werden entweder innerhalb der Datenbank (als Binärdaten) als auch ausserhalb (als JPEG, TIFF, etc.) verwaltet.
- Jede PostGIS Raster SQL-Funktion soll sowohl auf Daten innerhalb als auch ausserhalb der Datenbank funktionieren (dort nur read-only).
- Eine Tabelle mit Rasterdaten ist ein Raster-Layer – analog dem Vektor-Layer.
- Ein Feld vom Typ Raster enthält Rasterdaten. Das kann ein komplettes Rasterbild (z.B. eine Landsat-Szene), eine Kachel (image "tile") oder ein Raster-Objekt (ein gerastertes Vektor-Objekt) sein. Es gibt keinen Unterschied zwischen Raster und Kachel.

Aktuell gibt es zur Darstellung von PostGIS Rasterdaten die Software-Werkzeuge OpenJUMP (als Vektor-Layer), sowie QGIS und Mapserver (als Raster-Layer). Geplant ist ein Support für gvSIG und GeoServer.

### Beispiele

Im folgenden soll mittels zwei Beispielen der kombinierte Einsatz von Vektor- und Rasterdaten in der gleichen SQL-Abfrage illustriert werden:

Beispiel 1 (aus WKTRaster-Tutorial)

Gesucht seien alle Höhenzellen eines DHM, auf denen Gämsen beobachtet wurden. Gegeben seien Gämsen-Beobachtungen, um die ein Puffer generiert wurde (Vektoren <gaemse\_point\_buffer>) sowie ein DHM (Raster) als Input. Damit kann folgende SQL-Anfrage formuliert werden:

```
SELECT id,  
       (ST_Intersection(raster, vektor)).geom AS geom,  
       (ST_Intersection(raster, vektor)).val  
FROM gaemse_point_buffer as vektor,  
     dhm_tiled as raster  
WHERE ST_Intersects(raster, vektor);
```

Das Resultat ist ein Vektor-Layer, der Polygone mit gleicher Höhe (vgl. Abbildung 1) enthält. Man beachte dabei die Funktion „ST\_Intersects(raster, vektor)“, welche als Argumente einen Raster- und eine Vektor-Tabelle enthält.

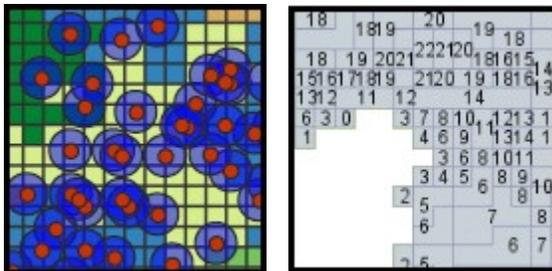


Abbildung 1: Input (Gämsen-Beobachtungen mit Puffer-Vektoren) und ein fiktives Resultat (Vektor-Layer).

Beispiel 2:

Im zweiten Beispiel wird der Vektorlayer av.gebaeude mit dem Rasterlayer dtm.dom verschnitten mit der Einschränkung auf das Gebäude mit der Assekuranznummer (geb.nummer) 1884. Aus

der Liste von Verschnittresultaten wird der Minimum-Wert, der Maximum-Wert und der Durchschnittswert ausgegeben. Weil das DOM nicht scharf mit der Gebäudekante übereinstimmt, wird ein Buffer von einem halben Meter nach innen vorgenommen. Der Subselect (verschachteltes SQL) und die ST\_Intersects()-WHERE-Bedingung dienen der Beschleunigung der Abfrage.

```
SELECT MIN((int_res).val), MAX((int_res).val), AVG((int_res).val)
FROM
  (SELECT ST_Intersection(dom.rast, ST_BUFFER(geb.the_geom,-0.5)) AS int_res
   FROM dom, gebaeude geb
   WHERE geb.nummer = 1884 AND ST_Intersects(dom.rast, geb.the_geom)
  ) foo;
```

### Weitere Ressourcen

- Buch „PostGIS in Action“ von Regina Obe und Leo Hsu – <http://www.manning.com/obe/>
- PostGIS Raster Homepage: <http://trac.osgeo.org/postgis/wiki/WKTRaster>
- PostGIS Raster Tutorial: <http://trac.osgeo.org/postgis/wiki/WKTRasterTutorial01>
- Oracle Georaster Referenz:  
[http://download.oracle.com/docs/html/B10827\\_01/geor\\_intro.htm](http://download.oracle.com/docs/html/B10827_01/geor_intro.htm)
- PostGIS Raster/Oracle Georaster Vergleich:  
<http://gis4free.wordpress.com/2010/07/19/oracle-georaster-part-i/>

### Diskussion und Ausblick

Kritisch zu bemerken ist, dass die Grösse der Kacheln (Pyramide/Tile) im Vorhinein - d.h. zum Zeitpunkt des Ladens - festgelegt werden muss. Die Wahl der Kachelgrösse beeinflusst die Geschwindigkeit von Analysen. Je nach Art der Analyse wäre eine andere Kachelgrösse sinnvoll.

PostGIS Raster ist noch nicht vollständig implementiert. Vieles funktioniert bereits, insbesondere die folgenden vier interessanten SQL-Befehle sind jedoch noch in Entwicklung:

- ST\_Resample(raster,pixelsize,method) – Resampling von existierenden Rastern mit verschiedenen Resampling-Methoden
- ST\_Reclass(raster|geometry,reclass-string) – Reklassifizierung von Raster- und Vektorgeometrien über die Definitione von Wertebereichen und zugehörigen Mappings
- ST\_SelectByValue(raster|geometry,'expression') – Auswahl von einzelnen Pixeln über beliebige Ausdrücke (Formeln, Wertebereiche, etc.)
- ST\_MapAlgebra(raster1|geometry1,raster2|geometry2,'mathematical expression') – beliebige Berechnungen aufgrund von zwei verschiedenen Rasterdatensätzen

Ebenso sind die Ausgabefunktionen noch in Arbeit (ST\_AsJPEG(), ST\_AsTiff(), ST\_AsPNG()). Derzeit können lediglich binäre Rohdaten ausgegeben werden oder ein Export über gdal in ein beliebiges anderes Raster- oder Vektor-Format durchgeführt werden.

Der Release PostGIS 2.0 auf das 1. Quartal 2011 geplant. Ein Prerelease für Windows und Linux kann jetzt schon heruntergeladen werden.

Die Analyse von Coverages wurde also nicht neu erfunden. Es ist jedoch eine bisher einzigartige Realisierung von „transparenten“ Operationen auf Raster- und Vektordaten sowie eine Implementation nach dem Prinzip der "Berechnung nur dort und dann, wo und wann es benötigt wird". Die Grenze zwischen Raster und Vektor verwischt dabei, was durchaus die eine oder andere Bemerkung auch in GIS-Einführungen Wert sein könnte.

Fachgruppe GIS-Technologie  
[technologie@sogi.ch](mailto:technologie@sogi.ch)  
Stefan Keller und Andreas Neumann