# Implementazione di PostgreSQL-PostGIS per l'analisi di immagine satellitare: test di applicabilità per la batimetria marina

Aldo Banni, Monica Deidda

DICAAR, Università di Cagliari, via Marengo 2, 09126 Cagliari, fax 0706755402, tel. 0706755404

**Introduzione**

Nell'ambito del progetto PRIN2008 l'unità di ricerca di Cagliari è impegnata nell'individuazione di metodologie per l'implementazione di una serie di funzioni PostgreSQL-PostGIS per valutazioni statistiche e analisi delle serie temporali/spaziali, sia con algoritmi di uso generale sia con l'applicazione di modelli specifici. Il server PostgreSQL -PostGIS permette, infatti, l'inserimento di nuove funzioni, e librerie di funzioni, in aggiunta a quelle di corredo, quali operatori geometrici, funzioni geometriche, conversioni geometriche, operatori e calcoli matriciali.

Delle nuove funzioni si vuole, in pratica, verificare l'utilità, l'applicabilità e la correttezza in procedure di interazione su interfaccia WEB-GIS.

Procedure intese a fornire analisi e valutazioni non di precisione ma comunque utili come strumento di approccio e sperimentazione di metodologie di analisi. La validità delle funzioni implementate viene effettuata con l'ausilio di un WEB-GIS che, per quanto non di precisione, aiuta a verificare la correttezza dei modelli applicati e, riguardo alla validazione dei parametri, si tiene conto del livello di confidenza prefissato.

Allo scopo di verificare capacità e rendimento delle funzioni implementate in PostgreSQL-PostGIS è stato "trasferito" sul server uno studio di analisi di immagine satellitare per la determinazione delle curve batimetriche, precedentemente sviluppato in ambiente IDL-ENVI.

In particolare si è fatto ricorso alle funzioni GDAL PostGIS Raster per l'immissione delle immagini raster satellitari nel database, previa trasformazione in forma tabellare, così da poter essere in seguito utilizzate sia dalle funzioni originali PostgreSQL/PostGIS che da quelle da noi implementate.

**Impianto e configurazione del sistema**

Il sistema SW atto allo sviluppo della procedura è così configurato: *Linux kernel 2.6.32.1; Web Server: Apache, Tomcat*; DB Server: *PostgreSQL 9.1 + PostGIS 2.0 (raster data support)*; *Piattaforma Pubblicazione Dati Spaziali: MapServer (supporta PostGIS WKT Raster)*; *Piattaforma Sharing Dati Spaziali: GeoServer*; *Desktop GIS: QuantumGIS (supporta PostGIS WKT Raster)*.

La funzione *raster2pgsql* di PostGIS permette di inserire, "traducendolo", un file raster nel database.

Per l'inserimento da remoto dell'immagine da elaborare il server web è implementato con *script*, in *php* , di *uploading* e trasformazione. Gli *script* contengono in sintesi:

- Accesso al database specifico;
- Costruzione di pagina interattiva con "*form action*" diretta all'*uploading/elaborazione* (per caricare il file nella directory scelta si utilizzano semplicemente l'html markup `input type="file"` e la variabile php globale $_FILES);
- Funzione php system($raster2database) per l'esecuzione del comando PostGIS *raster2pgsql*, che costruisce la query per l'inserimento dell'immagine in forma di tabella;
- Esecuzione della query.

**La tabella raster**

La funzione di trasferimento di un file raster (per esempio un file TIFF) è

*raster2pgsql* -s 23032 -I -C -M -r -d /home/aldo/Ricerca/Ricerca_Papers/WEB-GIS/POSTGIS/ikonos_clip_rgbnir.tif ,-t 128x128 ik_128x128 > ikonos_poetto.sql

dove con -s si sceglie lo *Spatial Reference System Identifier;* con -t la dimensione (pixel X x Y) di ciascuna riga nella suddivisione dell'immagine (la necessità di dividere l'immagine in celle sarà esposta in seguito).

La query descritta dal file *ikonos_poetto.sql* serve per la costruzione di una tabella con i campi *rid* (di tipo intero, rappresenta il numero della riga che contiene i dati relativi ad una cella) e *rast* (di tipo raster).

Il tipo raster è una struttura contenente, tra l'altro, il numero di bande (4 nel nostro esempio), il *pixel-type* (nel nostro caso 16BUI: 16-bit unsigned integer) e i valori radiometrici per ciascuna banda per tutti i pixel della cella (compreso il valore *NULL* in caso di assenza di valore, nel partizionamento in celle alcune presenteranno dei pixel esterni all'immagine, e quindi senza valori radiometrici). La suddivisione in celle è funzione delle analisi che si vogliono fare, infatti nel nostro caso bisogna creare delle "*DOP Zone*", celle di estensione congrua alla omogeneità, e su queste effettuare calcoli statistici. Nel caso che la tabella raster sia stata costruita con celle troppo grandi il tempo di calcolo aumenta notevolmente se le *DOP zone* hanno estensione inferiore a quelle delle celle, dovendo estrarre quindi delle sotto-celle. Come con la generazione di serie di NxN pixel (costruzione di un poligono con la query PostGIS "*ST_UpperLeftX, ST_UpperLeftY, ST_UpperLeftX+ST_ScaleX*N, …, CROSS JOIN generate_series"*).

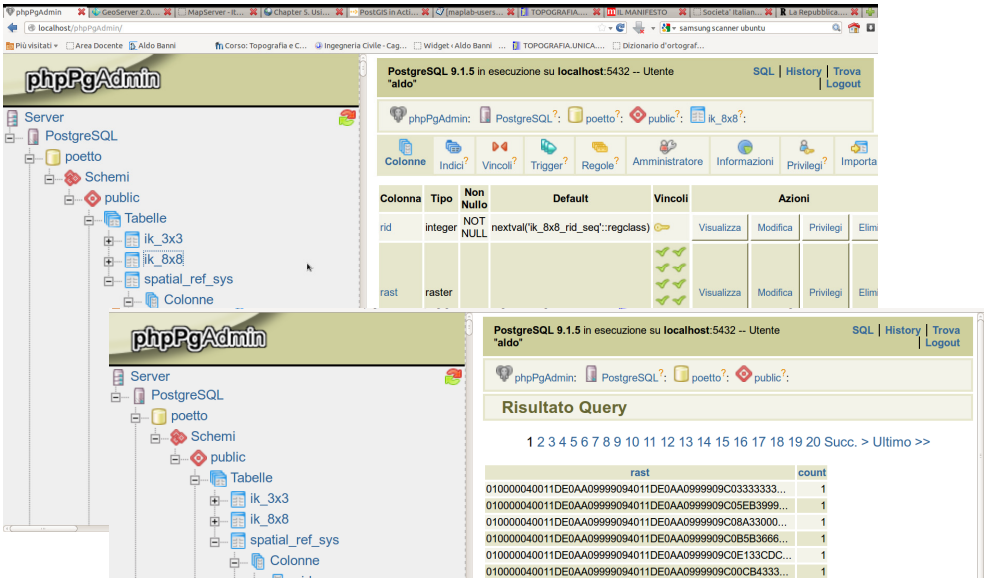


*Figura 1. Immagine trasferita su raster con celle (righe rid) di dimensione 8x8 pixel (~ 25m x 25m).*

**Nuove funzioni di analisi ed elaborazione**

L'esistente libreria di funzioni Post-GIS permette già molte operazioni e analisi sui raster: costruzione di poligoni da unione di pixel secondo il valore di una o più bande; campionamento e riduzione in scala; statistiche e istogrammi; operazioni algebriche tra i valori delle bande; restituzione del raster in formato GDAL e, quindi, in immagini JPEG, GTIFF o PNG.

Nel caso di una elaborazione matematica complessa il server PostgreSQL-PostGIS può essere implementato con nuove funzioni.

Infatti una delle principali caratteristiche di PostgreSQL è il supporto per l'implementazione di funzioni definibili dall'utente e sviluppabili in vari linguaggi di programmazione, quali script SQL, Perl, PHP, Python o compilati in C. Con l'ausilio di linguaggi genericamente chiamati *Procedural Language* (PL), PL/pgSQL, PL/Tcl, PL/Perl e PL/Python, gli script di programmazione sono implementati correttamente nella libreria di PostreSQL.

Per quanto riguarda le funzioni scritte in C/C++, vengono fornite le librerie dinamiche di interfaccia con il *DB server*.

Il test di applicabilità proposto riguarda il modello per la determinazione dei valori batimetrici in prossimità di coste marine mediante l'analisi di immagini satellitari.

Considerando che l'attenuazione della radiazione solare in acqua è una funzione esponenziale della profondità e ipotizzando che la qualità dell'acqua, e quindi il coefficiente di attenuazione e l'albedo del fondale siano costanti in tutta la porzione dell'immagine satellitare esaminata, i valori batimetrici vengono ottenuti attraverso interpolazione dalla serie osservativa relativa alle *DOP Zone*. Gli algoritmi delle funzioni di "lettura" delle immagini e di applicazione del modello sono già stati scritti e testati in linguaggio IDL-ENVI (Deidda M., Sanna G., 2012), quindi si è proceduto alla traduzione in linguaggio C++ utilizzando il Procedural Language (PL) per la loro implementazione come funzione PostgreSQL/PostGIS.

In particolare la funzione *batimetria* provvede alla procedura di calcolo batimetrico delle *DOP zone* e alla generazione di una tabella raster con una banda "profondità". Tabella raster poi visualizzabile (viene restituita un'immagine tiff o jpg o png).

```
#include "postgres.h"
#include "fmgr.h"
#include <string.h>
#include <math.h>

---------------------------

#ifdef PG_MODULE_MAGIC
PG_MODULE_MAGIC;
#endif

Datum batm( PG_FUNCTION_ARGS );

PG_FUNCTION_INFO_V1( batm );
Datum batm( PG_FUNCTION_ARGS )
{

----------------------------------------
```

```
CREATE OR REPLACE FUNCTION
  batimetria( RASTER,GEOMETRY,DOUBLE PRECISION )
RETURNS
  INTEGER
AS
  'batimetria.so', 'batimetria'
LANGUAGE
  C
STRICT
IMMUTABLE;
```

*Figura 2 a. Sorgente C++ della funzione "batimetria" e query per il suo inserimento nella libreria PostgreSQL.*

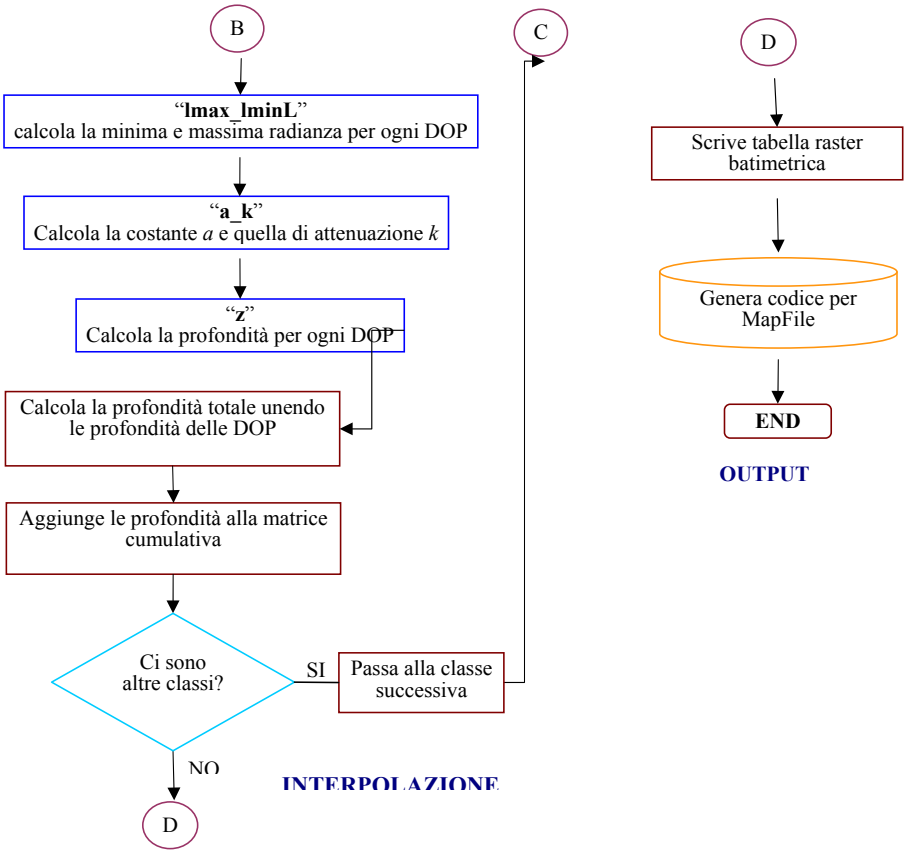


*Figura 2 b. Diagramma di flusso della funzione "batimetria".*

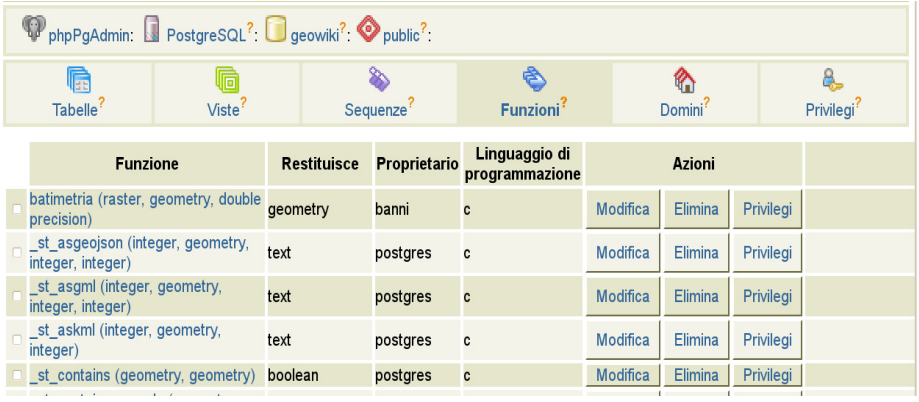


*Figura 3. Libreria funzioni PostgreSQL implementata con "batimetria".*

Come interfaccia WEB-GIS viene utilizzato MapServer con i supporti GDAL e PostGIS-Raster (in teoria qualunque pacchetto che utilizzi GDAL può supportare i dati PostGIS-Raster). La visualizzazione del raster batimetrico è permessa dalla costruzione del realtivo *layer* nel *MapFile,* come descritto in sintesi:

```
LAYER
        NAME batim_raster
        TYPE raster
        STATUS ON
        DATA "PG:host=localhost port=5432 dbname='ik_8x8' user='banni' password='5cher7o0'
                schema='public' table='ikonos_band2' mode='2'"
        PROCESSING "NODATA=0"
        PROCESSING "SCALE=AUTO"
                CLASS
                NAME "curva1"
                EXPRESSION ([pixel] < 20)
                COLOR 250 250 250
        END
        CLASS
                NAME "curva2"
                EXPRESSION ([pixel] > 20 AND [pixel] < 1000)
                COLOR 255 0 0
        END
        CLASS
                NAME "curva3"
                EXPRESSION ([pixel] >= 1000)
                COLOR 0 255 0
        END
END
```

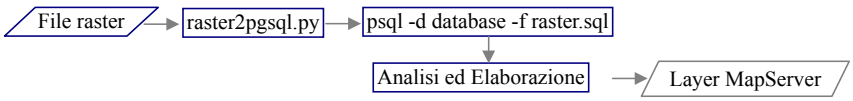


*Figura 4. Diagramma di flusso dall'immagine satellitare al Layer MapServer.*

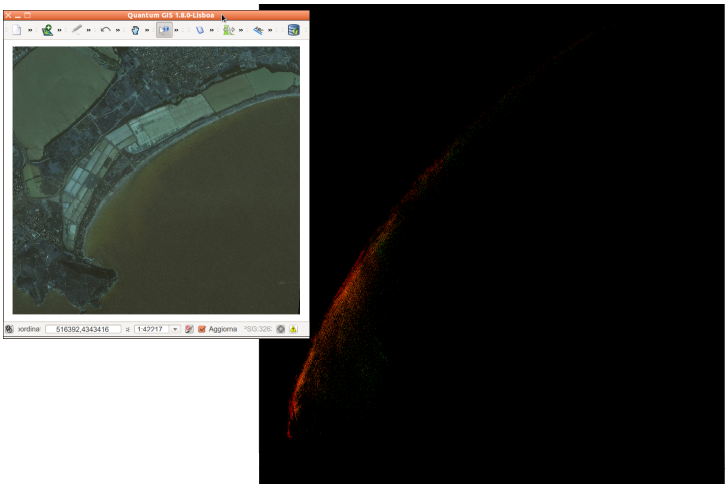

*Figura 5. Dal raster rgbIr al raster "DOP zone" ottenuto utilizzando la funzione "batimetria" nella query.*

**Bibliografia**

http://www.postgresql.org
http://postgis.refractions.net
http://trac.osgeo.org/postgis/wiki/WKTRaster
http://mapserver.org
Deidda M., Sanna G. (2012) "Bathymetric extraction using WorldView-2 high resolution images". *ISPRS Archives*, Volume XXXIX-B8
Deidda M., Sanna G. (2012) "Pre-processing of high resolution satellite images for sea bottom". *Italian Journal of Remote Sensing*, 44(1), pp. 83-95
Jupp, D.L.B., (1988), "Background and extensions to depth of penetration (DOP) mapping in shallow coastal waters". *Proceedings of the Symposium on Remote Sensing of the Coastal Zone*, Gold Coast, Queensland, September 1988, IV.2.1-IV.2.19