

## Explorando a extensão *WKT Raster* do PostGIS para armazenamento e manipulação de imagens de sensoriamento remoto

Lúbia Vinhas<sup>1</sup>, Karine Reis Ferreira<sup>1</sup>

<sup>1</sup> Instituto Nacional de Pesquisas Espaciais - INPE  
Caixa Postal 515 - 12227-010 - São José dos Campos - SP, Brasil  
{lubia,karine}@dpi.inpe.br

**Abstract.** Spatial extensions allow DMBS systems to store and query spatial data using high-level query languages capable of expressing complex queries mixing descriptive and spatial criteria. This work explores the WKT Raster, part of PostGIS, the spatial extension for PostgreSQL, to manage raster data, more specifically remote sensing imagery. This spatial extension was chosen, because it is, to the authors' knowledge, the most mature product that is free of license. Some typical remote sensing operations are executed using only the query language available in the WKT Raster. The results show that WKT Raster is a promising tool to handle raster data in object-relational databases. Some interesting features are the integration to the spatial types used to handle vector data and the large number of processing operators.

**Palavras-chave:** remote sensing, image processing, spatial databases, sensoriamento remoto, processamento de imagens, bancos de dados geográficos, PostGIS.

### 1. Introdução

Tradicionalmente, ao se trabalhar com dados geográficos, distingue-se conceitualmente duas categorias: a dos *campos* e a dos *objetos*. Os campos referem-se a observação de variáveis contínuas no espaço, por exemplo os valores de altimetria do terreno. São em geral representados por matrizes de valores, onde o valor de cada elemento da matriz é o valor da variável em uma porção regular do terreno, definida por parâmetros pré-estabelecidos (a chamada resolução espacial). Os objetos referem-se a entidades discretas que possuem uma identificação única que os distingue dos demais, por exemplo um município. São tradicionalmente representados por estruturas geométricas vetoriais: pontos, linhas e polígonos, as quais possibilitam análises baseadas principalmente em relacionamentos topológicos entre os objetos e outras funções geométricas. Para uma definição matemática formal de campos e objetos pode-se consultar o trabalho de Câmara (1995).

Os Sistemas Gerenciadores de Bancos de Dados (SGBD) gerenciam grandes volumes de dados, de maneira persistente, organizada e interrelacionada. Os Sistemas de Informação Geográfica (SIG) são sistemas de informação desenvolvidos para armazenar, processar e visualizar dados geográficos e, em geral, tem uma como componente central um SGBD. Inicialmente, os SGBDs proviam o gerenciamento apenas da componente convencional do dado geográfico enquanto que a componente de localização, ou espacial, era armazenada em arquivos externos e gerenciada somente pelo SIG. A partir dos anos 2000, os SGBDs passaram a oferecer as chamadas *extensões espaciais*, ou seja, um conjunto de tipos, métodos de acesso e mecanismos de indexação espaciais que permitem que o gerenciamento da componente espacial do dado geográfico também seja feita pelo SGBD (Queiroz e Ferreira, 2005). Esse artefato tecnológico permitiu a evolução das arquiteturas para o desenvolvimento de SIGs, ou de aplicações geográficas em geral, para diferentes ambientes como o monousuário, o colaborativo ou ainda o distribuído no ambiente da internet. Nessa arquitetura grande parte das funcionalidades oferecidas pelas aplicações é implementada através de consultas ao SGBD que é responsável por realizar o processamento do dado. A *Structured Query Language* (SQL) é uma linguagem especialmente desenvolvida para o gerenciamento de dados em SGBD's relacionais e as extensões espaciais incluem em sua SQL um vocabulário específico para manipular os dados espaciais (Shekhar e Chawla, 2003).

Todas as extensões espaciais existentes oferecem o suporte para tratar dados com representação vetorial, a maioria seguindo as especificações propostas pelo Open Geospatial Consortium (OGC)<sup>1</sup> como por exemplo a especificação *Simple Feature SQL* (Herring 2011). Porém, poucas oferecem o mesmo nível de suporte para tratar dados com representação matricial, como as imagens de sensoriamento remoto. Para isso foram desenvolvidas soluções baseadas em campos binários longos, onde o dado é armazenado sem semântica e toda a sua manipulação é feita pela camada de acesso ao banco de dados, ou pela própria aplicação (Vinhas et al. 2003; Barclay et al. 2000; Baumann 1994).

As duas principais iniciativas de fornecer suporte a dados matriciais embutidas em uma extensão espacial de maneira similar aos dados vetoriais são o *GeoRaster* que faz parte da extensão *Oracle Spatial* (Murray et al. 2003) e mais recentemente o *WKT raster* da extensão *PostGIS* para o *PostgreSQL* (Obe and Hsu, 2011). Esse trabalho tem por objetivo experimentar e apresentar algumas das funcionalidades da extensão espacial *WKT Raster*, do *PostGIS* no processamento de imagens de sensoriamento remoto. O *WKT Raster* foi escolhido por ser um produto livre de licença, de maneira que esse trabalho possa ser reproduzido total ou parcialmente pelo leitor. Para isso, são reproduzidos os comandos SQL utilizados em cada processamento.

## 2. Metodologia de Trabalho

Foram escolhidas 3 bandas (2, 3 e 4) de 12 cenas do sensor CCD/CBERS-2B (ou seja 36 arquivos de imagens, em formato GeoTIFF), foram usadas para a construção do banco de dados) que recobrem a área do município de Novo Progresso, no estado do Pará. A Figura 1 mostra localização das 12 cenas, tomando como referência o estado do Pará. Em amarelo destaca-se o contorno do município de Novo Progresso.

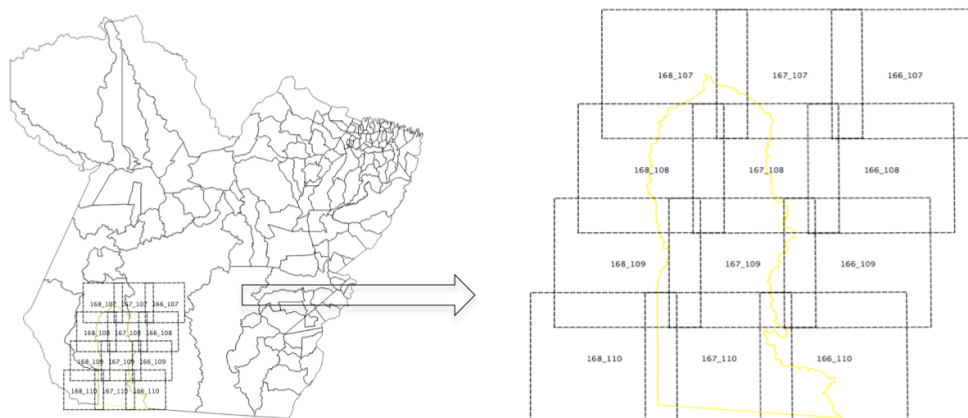


Figura 1. Localização das cenas utilizadas.

O INPE disponibiliza as imagens CBERS-2B em nível 2 de correção geométrica, o que implica dizer que para sua utilização é necessário a execução de um procedimento de registro (por ponto de controles, por exemplo) para que se obtenha uma precisão geométrica aceitável. No contexto desse trabalho, considera-se que as informações de localização registradas no arquivos em formato GeoTIFF é suficiente. As imagens estão georeferenciadas para uma um sistema de referência espacial plano usando projeção UTM, sobre Datum WGS84. As imagens tem resolução espacial horizontal e vertical de 20 metros. Essas informações estão escritas no formato GeoTIFF. Quanto a resolução radiométrica, as imagens CBERS-2B/CCD são codificadas em 8 bits sem sinal, ou seja, os valores de cada pixel podem variar de 0 a 255.

<sup>1</sup> <http://www.opengeospatial.org/>

O sistema de processamento do INPE utiliza o valor 0 para indicar “ausência de informação” que nesse trabalho é chamado de *nodata*. Ou seja, não existem pixels com valor 0 em áreas úteis da imagem, como pode ser visto na Figura 2.

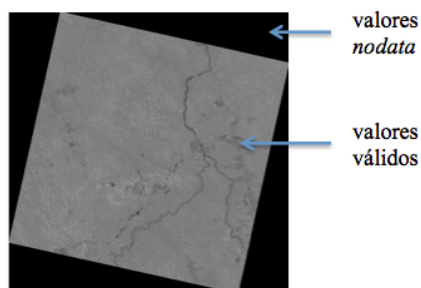


Figura 2. Representação de valores *nodata*.

Considere-se o seguinte conjunto de processamentos inspirado em aplicações de mapeamento de uso e cobertura da Terra:

1. Organização das cenas que recobrem a área de interesse em um banco de dados;
2. Análise exploratória dos valores da imagem;
3. Realce de contraste das imagens;
4. Classificação das imagens;
5. Cálculo de área de classes;
6. Recorte das cenas pela área de interesse.

Nesse trabalho foi avaliados se, e como, esses processamentos podem ser executadas usando os tipos e funções espaciais disponíveis na extensão *PostGIS/WKT Raster*, usando comandos SQL submetidos ao gerenciador.

Os comandos utilizados para executar os processamentos são mostrados nos itens com a legenda “Consulta” e os resultados dessas consultas que não são gráficos são mostrados nesses mesmos itens. Para mostrar graficamente as imagens originais, ou aquelas resultantes de manipulações usou-se a função *ST\_AsJPEG* que exporta um dado *raster* para um arquivo em formato JPEG.

### 3. Resultados e Discussão

#### 3.1 Organização do banco de dados

O *WKT Raster* permite diferentes arranjos para organização dos dados *raster* dentro do banco. Primeiramente, a extensão permite que apenas os metadados sobre um dado matricial sejam armazenados no banco juntamente com uma referencia externa ao arquivo de dados.

Nesse trabalho explora-se a capacidade o uso do banco de dados para armazenar efetivamente os dados de imagem. Nesse caso, o banco pode conter uma ou mais tabelas com colunas com tipos espaciais (como *geometry* ou *raster*); e por sua vez, uma tabela pode conter uma ou mais colunas do tipo *raster*. Cada valor em uma coluna do tipo *raster* representa um dado completo, ou seja, pode ser manipulado independentemente. Assim diferentes arranjos podem ser construídos de acordo com as demandas da aplicação final, os dados disponíveis ou ainda considerando questões de otimização. A Figura 3 mostra 2 exemplos de arranjos, considerando todos os arquivos de imagem disponíveis nesse trabalho. Cada quadrado representa uma célula de uma tabela, e as cores representam o arquivo de dados que deu origem ao conteúdo da célula.



Figura 3. Arranjos de tabelas para armazenamento das imagens: a) um arquivo por registro; b) cada arquivo dividido em blocos de 512 x 512 pixels.

Nesse trabalho, foi utilizado o arranjo mostrado na Figura 3 b) onde cada banda de imagem foi armazenada em uma tabela, com uma coluna do tipo *raster* e cada arquivo referente a cada banda, foi dividido em blocos regulares de 512 x 512 pixels. Para a inserção das imagens no banco de dados usou-se o script de *raster2pgsql* que é disponibilizado junto com o *WKT Raster*. O script processa um ou mais arquivos de imagens e produz um arquivo de comandos SQL que deve ser submetido ao gerenciador. A Figura 4 exemplifica os comandos necessários executar essas duas ações: a partir de um diretório contendo todas as imagens relativas a banda 2, é produzido um arquivo com os comandos SQL para a inserção dos dados no banco (no primeiro comando), que ao ser processado (segundo comando) cria e preenche a tabela no banco de dados.

```
/Dados/CBERS/Banda2/$ raster2pgsql -t 512x512 -F -N 0*.tif public.band2 > band2.sql
/Dados/CBERS/Banda2/$ psql -h localhost -U postgres -w -d cbers -f band2.sql
```

Figura 4. Comandos para inserção de um conjunto arquivos de imagens para um tabela no PostGIS.

Tabelas que possuem colunas com tipos espaciais tem seus metadados registrados em tabelas de controle da extensão espacial, no caso de colunas *raster* na tabela chamada *raster\_columns*. O *script* de importação automaticamente extrai as informações sobre as imagens que serão armazenadas na coluna com o tipo *raster*. A Consulta 1 mostra um exemplo de consulta a essa tabela e o resultado obtido, após a importação das 3 bandas.

```
SELECT r_table_name as table, srid, scale_x, scale_y, no_data_values, pixel_types FROM raster_columns;
```

table	srid	scale_x	scale_y	no_data_values	pixel_types
band2	32721	20	-20	{0}	{8BUI}
band3	32721	20	-20	{0}	{8BUI}
band4	32721	20	-20	{0}	{8BUI}

(3 registros)

Consulta 1. Comandos para consulta a tabela de metadados de sobre dados matriciais.

Dentre as informações que constam na tabela de metadados (mencionadas na consulta acima), é interessante notar as seguintes: *srid* que guarda o código do sistema de referência espacial da imagem (no caso o código 32721 representa o sistema de referência projetado em projeção UTM, usando o Datum WGS84); *scale\_x* e *scale\_y* que armazenam a resolução espacial da imagem (no caso 20 x 20 metros); *pixel\_types* que armazena a resolução radiométrica da imagem (no caso “8BUI” significa 8 bits sem sinal); e o *no\_data\_values* que armazena o valor *no\_data*, nesse caso o valor 0. Uma vez criado o banco de dados nas seções seguintes explora-se as capacidades de análise da extensão espacial.

### 3.2 Processamento envolvendo dados vetoriais

Uma das vantagens de se ter uma extensão espacial que trate dados matriciais é a possibilidade de expressão, na linguagem de consulta, de processamentos que envolvem dados com representação vetorial e matricial. Para exemplificar esse processamento mostra-se como fazer o recorte das imagens pelo polígono que delimita o município de interesse. Importou-se os dados dos municípios do Brasil para uma tabela chamada *munic\_2001* com uma coluna espacial chamada *geom* (do tipo *geometry*) que armazena os limites do município.

O recorte é feito usando a função *ST\_Clip*, e armazenando o resultado na mesma tabela em uma coluna do tipo *raster* chamada *clip\_rast*, como mostrado na Consulta 2 para recortar um dos blocos da imagem que tem sobreposição com a fronteira do município. E a Figura 5 mostra o resultado do recorte. É interessante notar que o dado de municípios está em um sistema de referência espacial geodésico (longitudes e latitudes) sobre o Datum WGS84. A função de recorte reconhece essa diferença e faz o remapeamento necessário antes do recorte. O valor *nodata* é atribuído aos pixels eliminados pelo recorte.

```
ALTER TABLE band2 ADD COLUMN clip_rast raster;
UPDATE band2
SET clip_rast = ST_Clip(enh_rast, 1, munic_2001.geom)
FROM munic_2001
WHERE band2.rid=86 AND munic_2001.nome = 'Novo Progresso';
```

Consulta 2. Comandos SQL para gerar as imagens recortadas pelo município.

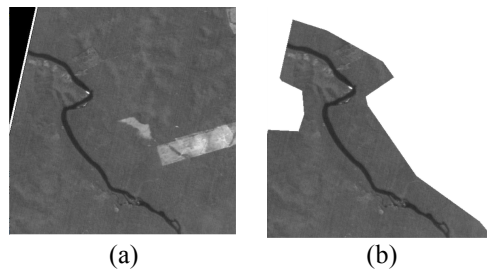


Figura 5. Recorte: a) imagem original e b) imagem recortada.

### 3.3 Análises e estatísticas

Para uma análise exploratória das imagens, pode-se usar as funções para calcular estatísticas e análises sobre os valores de um *raster* ou de um recobrimento formado por vários *raster*. Essas funções podem considerar os valores *nodata* ou não. A Consulta 3 mostra uma consulta simples, a contagem do número de pixels em bloco de imagem arbitrário, considerando-se e desconsiderando-se os valores *nodata*. Além disso, as funções de análise de valores de um dado *raster* podem operar sobre um bloco (Consulta 4a), ou sobre uma tabela inteira, ou seja, sobre uma cobertura (Consulta 4b).

```
SELECT rid, ST_Count(rast, true) AS exclui_nodata
       ST_Count(rast, false) As inclui_nodata
FROM band2 WHERE rid=3;
```

```
rid | exclui_nodata | inclui_nodata
-----+-----+-----
1 | 167086 | 262144
(1 registro)
```

Consulta 3. Consulta o número de pixels em um bloco de imagem.

```
SELECT rid, (stats).*
FROM
( SELECT rid, ST_SummaryStats(rast) AS stats
  FROM band2 WHERE rid=3)
AS aux;
```

rid	count	sum	mean	stddev	min	max
3	167086	6722299	40.2325	26.1698	30	255

(1 registro)

(a)

```
SELECT (stats).*
FROM (SELECT ST_SummaryStats('band2', 'rast', 1, true) AS stats)
AS aux;
```

count	sum	mean	stddev	min	max
378714055	18624477959	49.1782	20.6254	1	255

(1 registro)

(b)

Consulta 4. Consulta estatísticas de valores: a) de um bloco e b) de uma tabela representando uma cobertura.

### 3.4 Realce de contraste

Operações de realce de contraste tem por objetivo melhorar a discriminação visual de pixels com valores diferentes. Uma opção simples de realce de contraste pode ser feita através da aplicação de uma função linear que altera os valores de cada pixel de acordo com um parâmetro de ganho e *offset* definidos a priori. Para definir os parâmetros das funções de realce de contraste, o usuário observa as estatísticas básicas e o histograma de valores dos pixels da imagem. O histograma da imagem pode ser calculado via SQL na extensão espacial, como pode ser visto na Consulta 5. Um dos parâmetros da consulta é o número de intervalos de valores para os quais será feita a contagem de pixels. Nesse exemplo são solicitados 25 intervalos.

```
SELECT (stats).*
FROM (SELECT ST_Histogram('band2', 'rast', 1, 25,true) As stats)
AS aux;
```

Consulta 5. Cálculo do histograma com 25 de uma tabela representando uma cobertura.

A função de realce linear pode ser executada através da operação `ST_MapAlgebraExpr` do *WKT Raster* que permite a aplicação de uma expressão algébrica sobre valores dos pixels da imagem. Assim, para cada tabela relativa a cada banda, foi criado uma nova coluna *raster* chamada `enh_rast` que armazena o resultado da aplicação da expressão algébrica. Esse procedimento é exemplificado na Consulta 6.

```
ALTER TABLE band2 ADD COLUMN enh_rast raster;
UPDATE band2
SET enh_rast = ST_MapAlgebraExpr(rast,NULL,['rast']*5.9 - 128.95')
WHERE rid = 128;
```

Consulta 6. Aplicação da expressão algébrica que representa o realce linear de contraste.

### 3.5 Classificação

A classificação de imagens de sensoriamento remoto é um procedimento complexo, que em geral depende de uma grande interação com o usuário. A extensão espacial *WKT Raster* ainda não possui algoritmos de classificação que são tradicionais em sistemas de

processamento de imagens. Uma simplificação desse processamento é o fatiamento de intervalos de valores da imagem para um valor que representa uma classe temática. Para esse procedimento, pode-se usar a função `ST_Reclass`. A Consulta 7 exemplifica o uso dessa operação que fatia a imagem realçada em 3 classes (com a definição arbitrária das fatias).

```
ALTER TABLE band2 ADD COLUMN rast_class raster;
UPDATE band2
SET clas_rast= ST_Reclass(enh_rast,1,'0-80:1, 81-150:128, 151-255:254', '8BUI',0)
WHERE rid=128;
```

Consulta 7. Fatiamento de valores da imagem.

Supondo que a imagem fatiada representa uma classificação em 3 classes razoável, é possível consultar o número de pixels com cada um dos valores e portanto uma estimativa da área associada a cada classe encontrada no fatiamento. A Consulta 8 exemplifica esse procedimento.

```
SELECT (pvc).value AS classe, (pvc).count AS pixels, (pvc).count*20 AS area_m2
FROM (SELECT ST_ValueCount(clas_rast) As pvc
FROM band2 WHERE rid=128) As aux
ORDER BY (pvc).value;
```

classe	pixels	area_m2
1	1856	37120
128	250940	5018800
254	9348	186960

(3 registros)

Consulta 8. Contagem de pixels por classe.

É possível também transformar a representação matricial em uma representação vetorial, ou seja, fazer a vetorização da imagem através de uma consulta ao banco como pode ser visto na Consulta 9. Uma nova tabela chamada `segs` foi criada para guardar todos os segmentos e a classe associada.

```
INSERT INTO segs (class, geom)
SELECT val, ST_GeomFromText(ST_AsText(geom),32721)
FROM
( SELECT (ST_DumpAsPolygons(clas_rast)).*
FROM band2 WHERE rid=128
) AS aux ORDER val;
```

Consulta 9. Vetorização.

A Figura 6 mostra os resultados dessa sequência de processamentos.

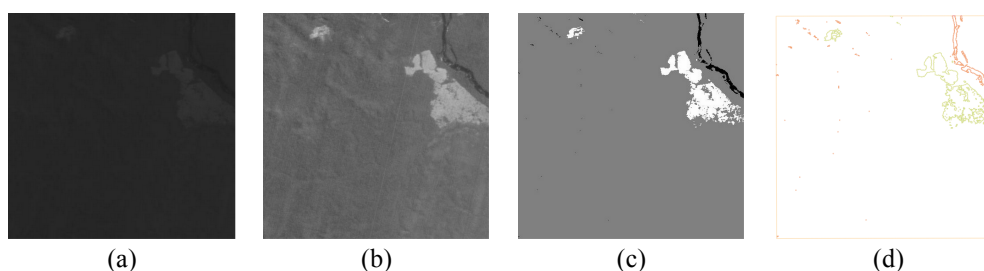


Figura 6. Exemplo de resultados das operações sobre imagem: a) imagem original, b) imagem realçada, (c) imagem classificada e d) imagem vetorizada.

Os processamentos descritos nas seções acima foram executados também para os recobrimentos, ou seja, para as tabelas completas de cada banda.

#### 4. Conclusões

Esse trabalho explorou a extensão *WKT Raster do PostGIS 2.0* para executar alguns processamentos típicos de aplicações em sensoriamento remoto, usando apenas consultas SQL. Os pontos interessantes dessa extensão são a flexibilidade na construção de arranjos que representam recobrimentos completos, e a possibilidade de executar processamentos que envolvem dados matriciais e dados vetoriais na mesma consulta. As consultas que realizam processamentos sobre blocos pequenos (nesse trabalho foram usados blocos de 512 x 512 pixels) são realizadas eficientemente, no entanto, foram observadas algumas limitações quando foram executadas sobre os recobrimentos ou seja, sobre uma tabela completa.

As extensões espaciais para SGBDs permitem que o gerenciamento da componente espacial do dado geográfico seja feita pelo SGBD. Para dados com representação vetorial as extensões espaciais já se mostraram eficientes e importantes em termos de interoperabilidade entre sistemas. No entanto, para o caso dos dados com representação matricial ainda existe um longo caminho a ser percorrido. Esse trabalho mostrou que o *WKT Raster do PostGIS 2.0* é uma solução livre de licença viável que dá mais um passo nesse caminho.

#### Referências Bibliográficas

- Barclay, T.; Gray, J.; Slutz, D. Microsoft TerraServer: A Spatial Data Warehouse. In: ACM SIGMOD International Conference on Management of Data, 2000, Dallas, Texas, USA. **Anais...** ACM Press, p. 307-318.
- Baumann, P. **On the Management of Multidimensional Discrete Data**. VLDB Journal, v. 4, n. 3, Special Issue on Spatial Database Systems, p. 401-444, 1994.
- Câmara, G. **Modelos, linguagens e arquiteturas para bancos de dados geográficos**. 1995. 282 p. (INPE--TDI). Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1995. Disponível em: <<http://urlib.net/sid.inpe.br/iris@1905/2005/07.29.07.25>>. Acesso em: 27 set. 2012.
- Herring, J. R. (2011). Open Geospatial Consortium Inc . **OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1 : Common architecture**. Disponível em <<http://www.opengeospatial.org> >. Acesso em: 27.set.2012.
- Queiroz, G. R.; Ferreira, K. R. SGBDs com extensões espaciais. In: Casanova, M. A.; Câmara, G.; Davis-Jr, C. A.; Vinhas, L.; Queiroz, G. R. (Org.). **Bancos de Dados Geográficos**. Curitiba: Editora MundoGEO, 2005. cap. 8, p. 267-303.
- Murray C.; Blowney J.; Xie J.; Xu T.; Yuditskaya S. **Oracle Spatial GeoRaster, 10 g Release 1**, Oracle Corporation, p. 2-31, 2003.
- Obe, R.; Hsu, L. **PostGIS in action** (chapter 3). Stamford, CT, USA, Manning Publications, 2011.
- Shekhar, S.; Chawla, S. **Spatial databases - a tour**. Upper Saddle River, NJ, USA, Prentice-Hall, 2003.
- Vinhas, L. ; Souza, R. C. M. ; Câmara, G. Image Handling in Spatial Databases. In: Simpósio Brasileiro de GeoInformática (GeoInfo), 5., 2003, Campos do Jordão. **Anais...** São José dos Campos: INPE, 2003.