



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

TerraLib: o software livre chega ao mercado de geoinformação

Gilberto Câmara

INPE

Licença de Uso: Creative Commons Atribuição-Use Não-Comercial-
Compartilhamento

<http://creativecommons.org/licenses/by-nc-sa/2.5/br/>



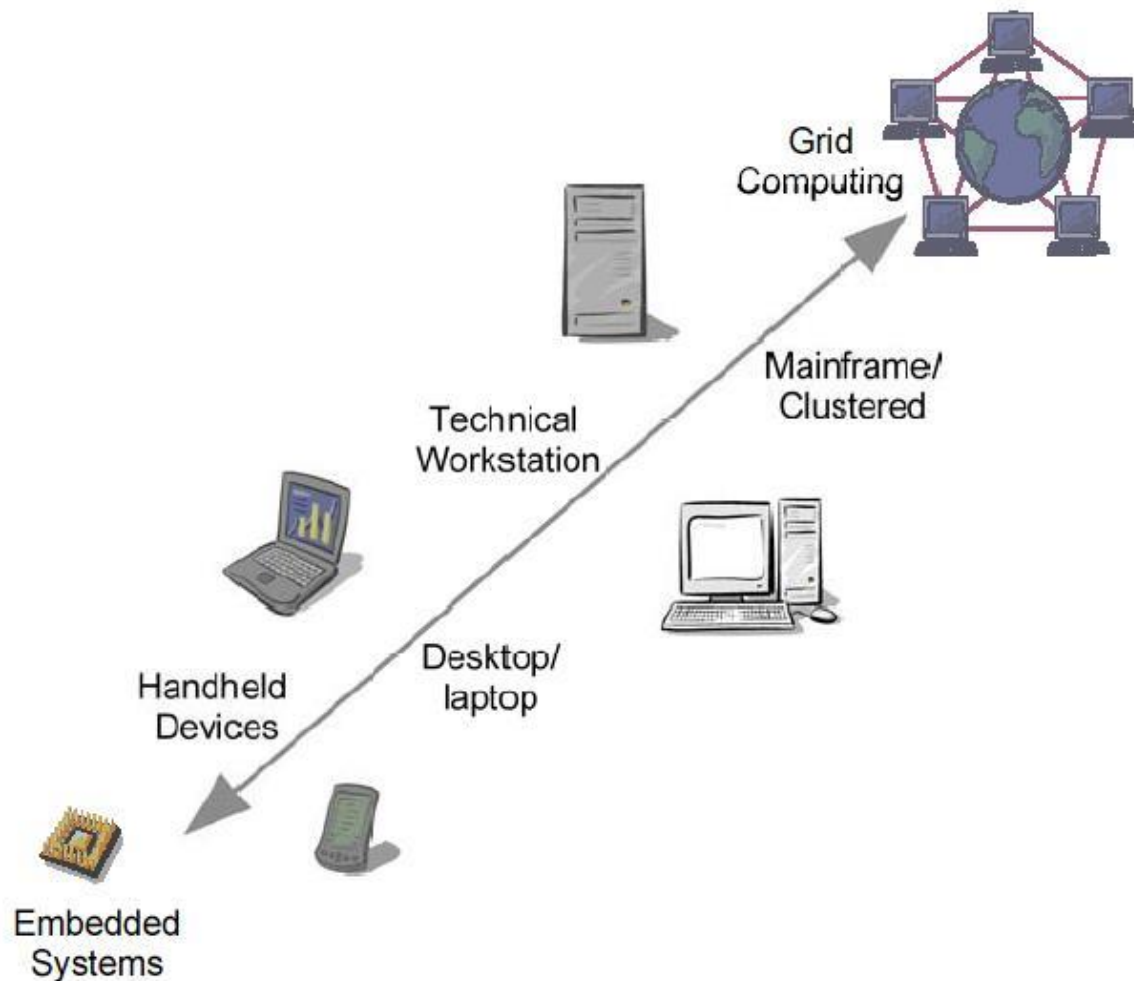
Motivação

- Porque software livre?
- Quais são os desafios do uso da geoinformação?
- O que é a TerraLib?
- Como um software livre pode contribuir para o bom uso da geoinformação?

Porque Software Livre?

- Software livre (*open source software*)
 - SL é software cujo código fonte está disponível e pode ser usado, copiado e distribuído com ou sem mudanças
 - SL pode ser cobrado, mas não escondido
- Exemplos de software livre
 - Linux, Apache, Open Office, PERL
 - Servidores Web
 - 2/3 dos servidores usam Apache

Linux e Escalabilidade



Vantagens Gerais do Software Livre

- Benefício social é maior
 - Software orientado para seu uso, não para maximizar objetivos empresariais
- Independência de tecnologia proprietária
- Não obsolescência de hardware
 - Evitar o “software bloat”
- Possibilidade de adequar aplicativos e redistribuir versão alterada
 - Otimiza uso da competência disponível
- Sistemas e aplicativos geralmente muito configuráveis

Vantagens Específicas do Software Livre

- Quando um projeto de software livre atinge um “tamanho crítico” temos ainda
 - Robustez e segurança:
 - muitos programadores tem acesso ao código, maior capacidade de detectar erros
 - Suporte abundante: soluções comunitárias e privadas
 - No Brasil, suporte comercial para Linux oferecido pela IBM e pela Conectiva

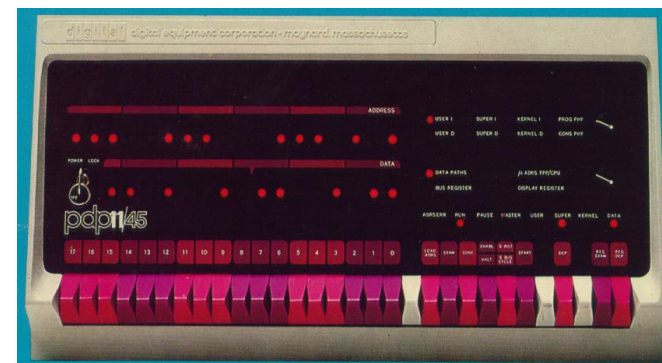
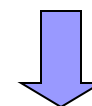
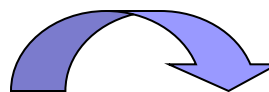
Software Livre é Pouco Confiável?

- Limitação de Responsabilidade (“Danos Indiretos”)
- Sujeito à legislação pertinente, em nenhuma hipótese o Fabricante ou seus fornecedores serão responsáveis por quaisquer danos (incluindo sem limitação danos diretos ou indiretos resultantes de lesão corporal, lucros cessantes, interrupção de negócios, perda de informações ou outros prejuízos pecuniários) decorrentes de uso ou da impossibilidade de usar este produto, ainda que o Fabricante tenha sido alertado quanto a possibilidade destes danos.
- De que software é esta licença?

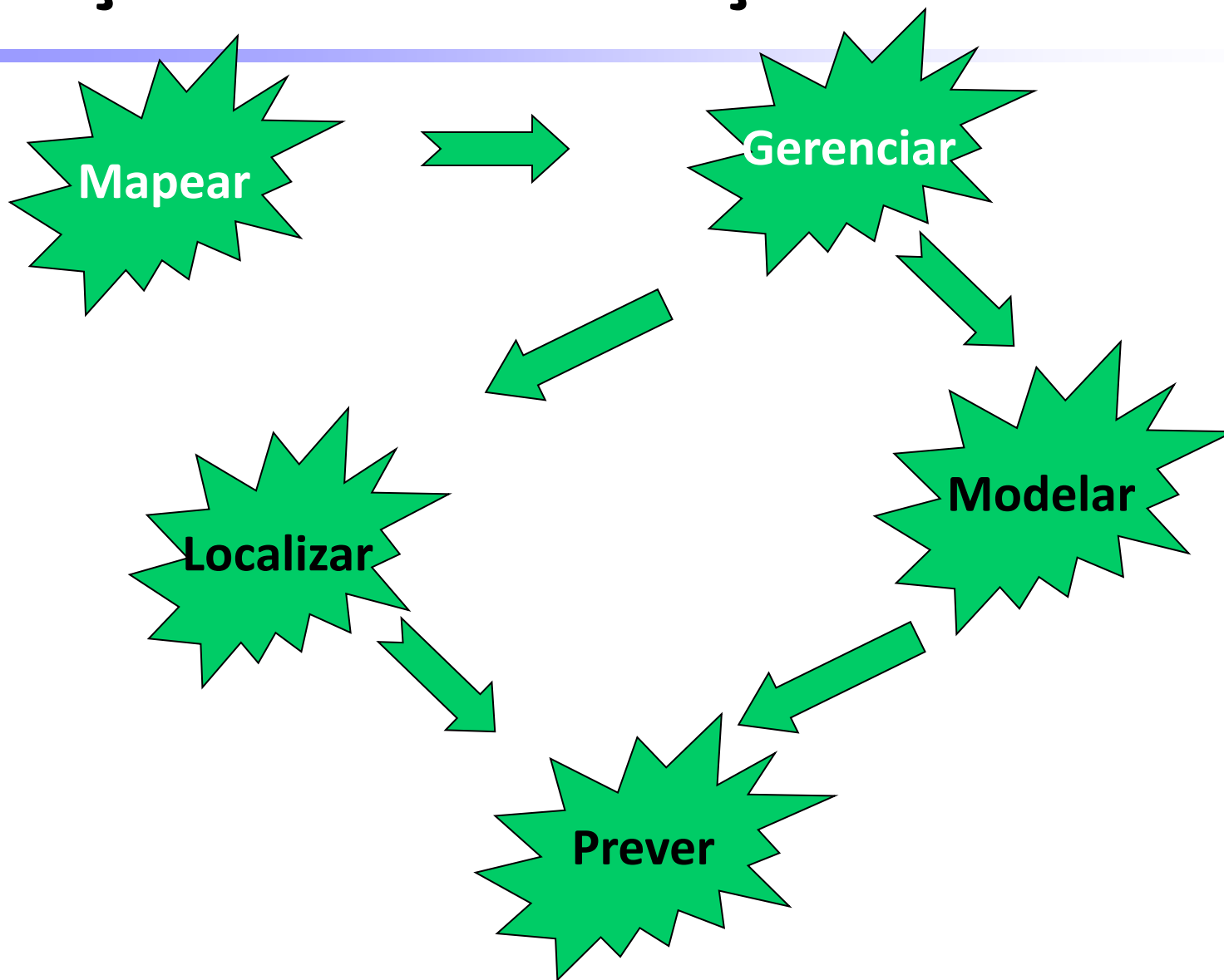
Licenças de Software Livre

- Direitos autorais
 - Ao disponibilizar um programa, seus autores escolhem o grau de liberdade com que modificações e redistribuições podem ser efetuadas.
- Tipos de licenças
 - GNU Public License (“copyleft”): qualquer modificação de SL também deve ser SL
 - Impede que SL seja integrado em software proprietário.
 - BSD-like: poucas restrições sobre a forma de uso, alterações e redistribuição do software licenciado.
 - O software pode ser vendido e não há obrigações quanto a inclusão do código fonte, podendo o mesmo ser incluído em software proprietário.
 - GNU Library License: SL pode ser incluído em software proprietário.
 - Produto final deve ter a parte de SL distribuída livremente

Colocando o Mundo no Computador



Evolução da GeoInformação



Evolução da Geoinformação

■ Mapear

- Novos dispositivos de captura de dados (GPS)
- Imagens de alta resolução

■ Gerenciar

- Bancos de dados geográficos
- Modelos semânticos e interoperabilidade

■ Localizar

- Sistemas baseados em localização (LBS)
- Gerência da informação distribuída

Evolução da Geoinformação

■ Modelar

- Construir descrições da realidade
- Modelos semânticos, matemáticos, lógicos

■ Prever

- Incorporar a dimensão temporal
- Construir cenários de mudança
- Mudanças de uso do solo, população, hidrologia, clima, agricultura

Diferentes Arquiteturas de SIG

- GIS “desktop”
 - Ambiente monousuário
 - Ênfase em interfaces amigáveis e funções de análise

- SIG distribuído
 - Ambiente multiusuário
 - Compartilhamento de dados
 - Ênfase em controle de acesso e manutenção de integridade

- Servidores Web
 - Uso da Internet para disseminar dados
 - Ênfase em eficiência de acesso e interfaces de navegação

Tratamento de Dados Geométricos

- Como armazenar linhas, pontos, polígonos, matrizes, imagens?

- Arquitetura dual (“modelo geo-relacional”)
 - Dados geométricos armazenados fora do banco de dados (em arquivos)
 - Solução mais comum
 - usada em SIG “desktop”

- Arquitetura em camadas (“modelo objeto-relacional”)
 - Dados geométricos armazenados dentro do banco de dados
 - Solução mais recente
 - usada em SIG distribuídos

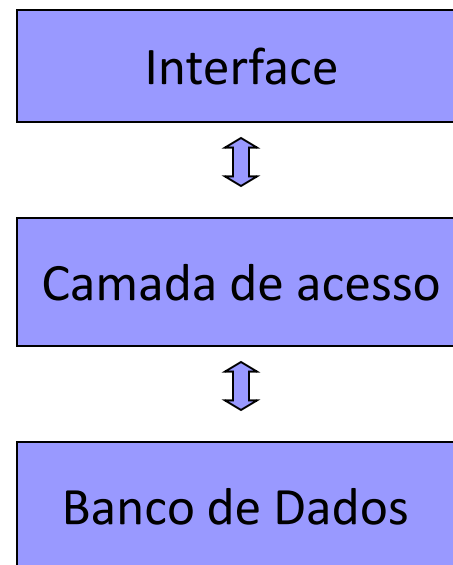
Banco de Dados Geográfico

- Modelo “objeto-relacional”
 - Tratar objetos (e.g. áreas) como partes de relação
 - Colocar os dados geométricos no banco de dados
- Extensões do modelo relacional
 - Usar uma coluna como um registro binário (“campo longo”)

object_id	num_coords	num_holes	parent_id	lower_x	lower_y	upper_x	upper_y	ext_max	spatial_data
35503080	51	0	1	-46.63816	-23.556028	-46.624577	-23.535415	3999973E-02	(bin)
35503066	657	0	2	-46.809763	-23.608743	-46.753759	-23.572728	0000015E-02	(bin)
35503085	161	0	3	-46.749461	-23.643174	-46.700546	-23.610856	0000009E-02	(bin)
35503056	585	0	4	-46.803053	-23.886584	-46.671223	-23.73694	.3999999999	(bin)
35503052	518	0	5	-46.811503	-24.007638	-46.605779	-23.873892	:3999999996	(bin)
35503017	403	0	6	-46.792593	-23.656437	-46.741559	-23.60841	0000014E-02	(bin)
35503042	599	0	7	-46.801392	-23.754089	-46.739429	-23.669322	0000029E-02	(bin)
35503019	295	0	8	-46.801277	-23.695534	-46.75258	-23.646908	0000042E-02	(bin)
35503045	171	0	9	-46.761531	-23.741118	-46.72052	-23.639828	0000000002	(bin)
35503096	320	0	10	-46.760325	-23.616778	-46.71935	-23.583661	4999999996	(bin)
35503022	176	0	11	-46.680056	-23.693784	-46.627618	-23.644098	0000022E-02	(bin)
35503023	112	0	12	-46.726892	-23.767187	-46.647337	-23.687184	0000014E-02	(bin)
35503081	41	0	13	-46.729571	-23.709342	-46.696599	-23.65472	0000019E-02	(bin)
35503030	218	0	14	-46.712146	-23.899163	-46.61458	-23.72638	2999999999	(bin)
35503016	70	0	15	-46.709441	-23.699324	-46.669016	-23.650883	3999968E-02	(bin)
35503059	297	0	16	-46.68546	-23.729181	-46.616361	-23.684599	0000014E-02	(bin)
35503034	117	0	17	-46.701453	-23.62593	-46.668255	-23.576248	0000007E-02	(bin)

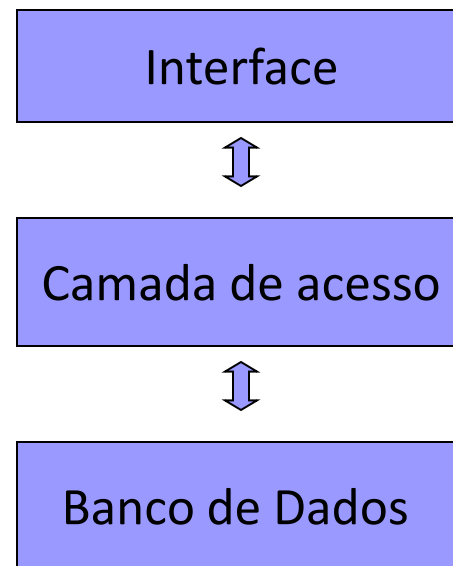
Componentes de Bancos de Dados Geográficos

- Banco de dados (alternativas)
 - Apenas suporte para campos longos (Access)
 - Interface para tipos de dados espaciais (ORACLE)
- Camada de Acesso
 - Bibliotecas de funções
 - TerraLib, ArcSDE
- Interface
 - Integrada com camada de acesso
 - TerraView
 - Cliente-Servidor
 - SIGMUN, ArcGIS 8.0



Arquitetura em camadas: componentes

- Exemplos – TerraView/TerraLib
- Banco de dados
 - ORACLE, mySQL
- Camada de Acesso
 - TerraLib
- Interface
 - TerraView



TerraLib

- Biblioteca de software aberto para GIS
 - Coleção de funções de software
 - Pode ser combinado para gerar programas executáveis
- Componentes básicos
 - Extensão de SGBD objeto-relacionais para dados espaço-temporais
 - Geometrias matriciais e vetoriais
 - ORACLE, Postgres, mySQL, Access
- Ambiente para desenvolvimento de aplicativos geográficos
 - <http://www.terralib.org>

TerraLib

■ Parceiros

- INPE – Divisão de Processamento de Imagens
- TECGRAF-PUC/RIO – Grupo de Tecnologia em Computação Gráfica
- FUNCATE – Gerência de Geoprocessamento
- UFMG – Laboratório de Estatística Espacial
- PRODABEL
- ENSP - FIOCRUZ

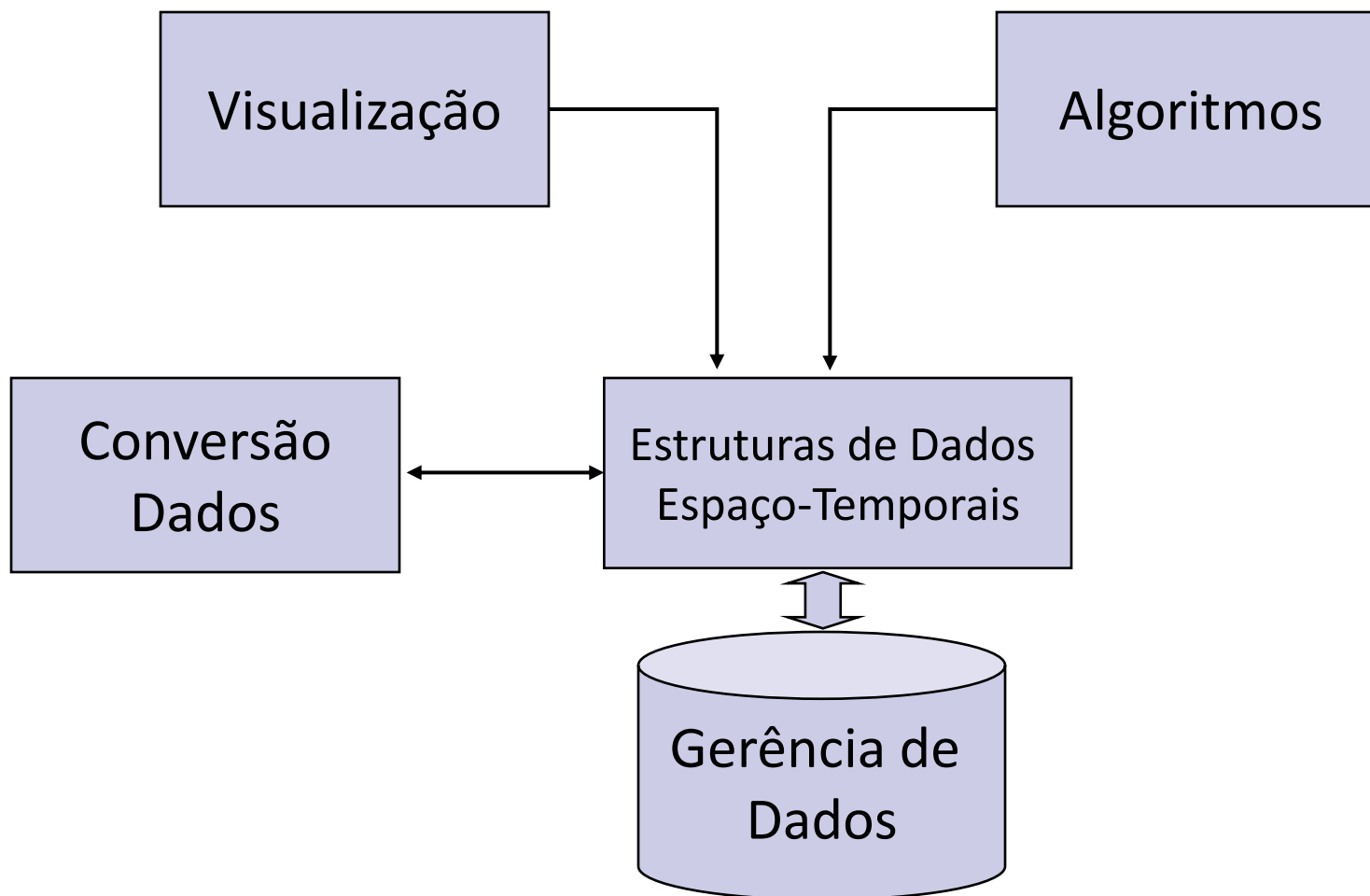
TerraLib: Motivação Científica

- Novos conceitos em geoinformação
 - Ontologias
 - Modelos espaço-temporais
 - Incerteza
 - Geocomputação

- Como construir novas idéias ?
 - Será que a pesquisa em geoinformação será limitada pela indústria?

- **Precisamos de software livre para compartilhar nossos resultados**

Estrutura da TerraLib



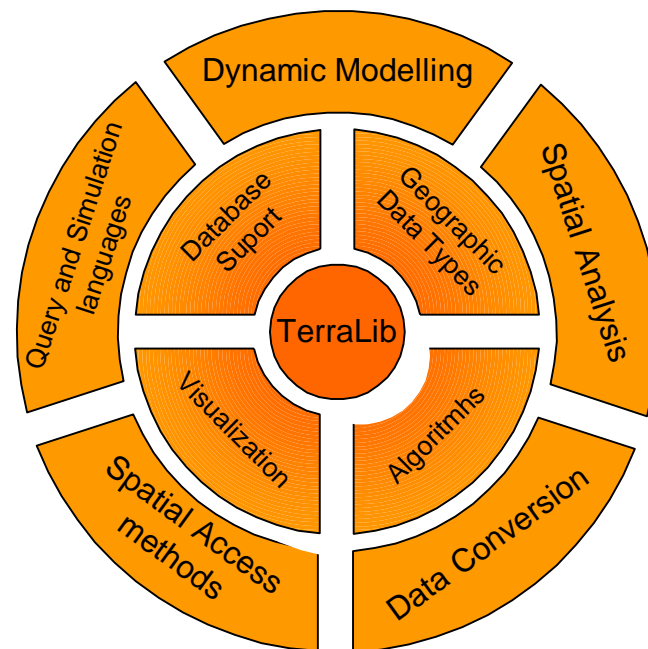
TerraLib: Arquitetura em Camadas

■ Kernel

- Interface com SGDB
- Tipos de dados espaço-temporais
- Indexação espacial

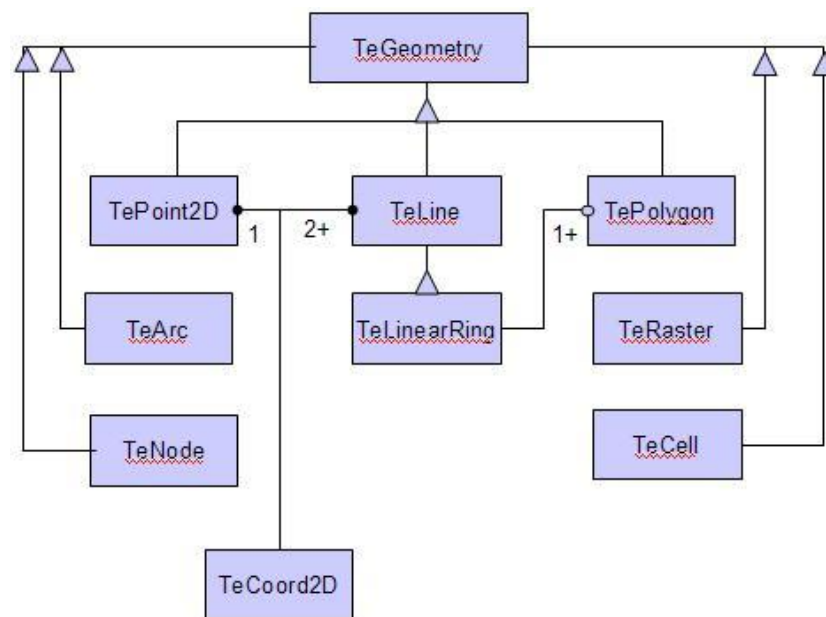
■ Aplicativos

- Visualização
- Análise Espacial
- Modelagem Dinâmica



Modelo de Dados Terralib

- Dados vetoriais
 - Modelo compatível com OpenGIS
- Classe básica: TeGeometry
- Classes derivadas
 - TePoint2D
 - TeLine
 - TePolygon
 - TePolygonSet



Operadores sobre Dados Vetoriais

- SQL functions for vector data :
 - Equals (g1 Geometry, g2 Geometry) : Integer
 - Disjoint (g1 Geometry, g2 Geometry) : Integer
 - Touches (g1 Geometry, g2 Geometry) : Integer
 - Within (g1 Geometry, g2 Geometry) : Integer
 - Overlaps (g1 Geometry, g2 Geometry) : Integer
 - Crosses (g1 Geometry, g2 Geometry) : Integer
 - Intersects (g1 Geometry, g2 Geometry) : Integer
 - Contains (g1 Geometry, g2 Geometry) : Integer
 - Relate (g1 Geometry, g2 Geometry, patternMatrix string) : Int

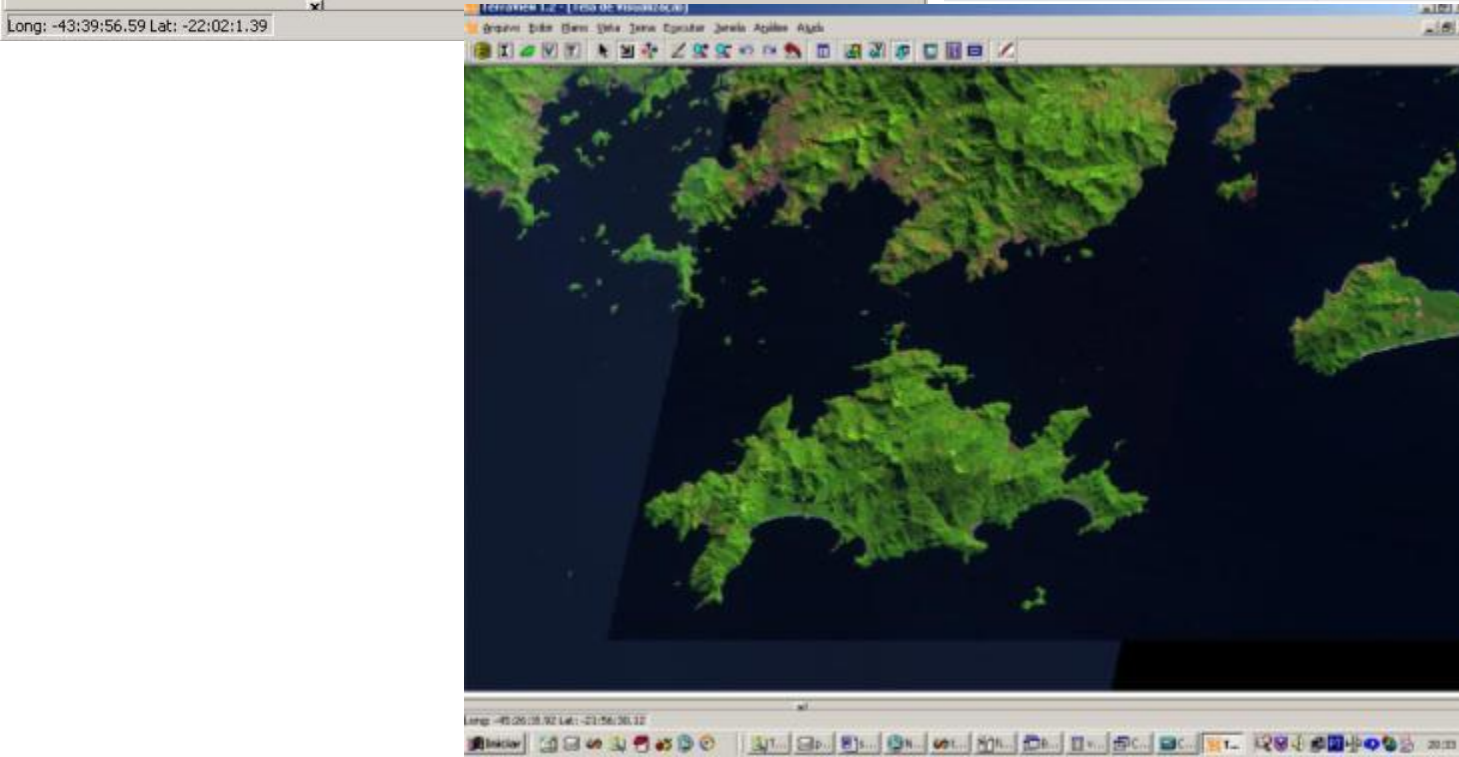
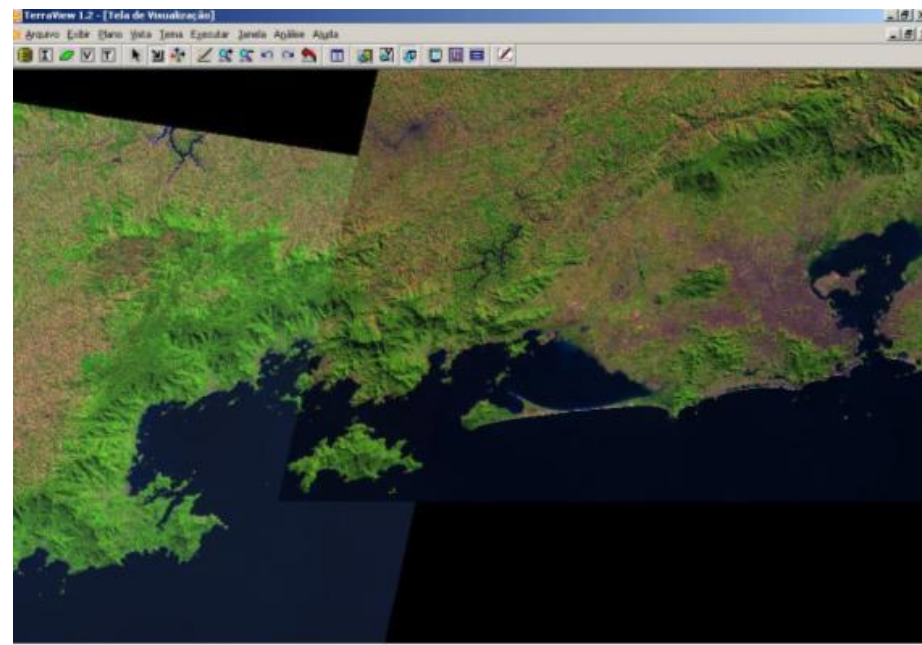
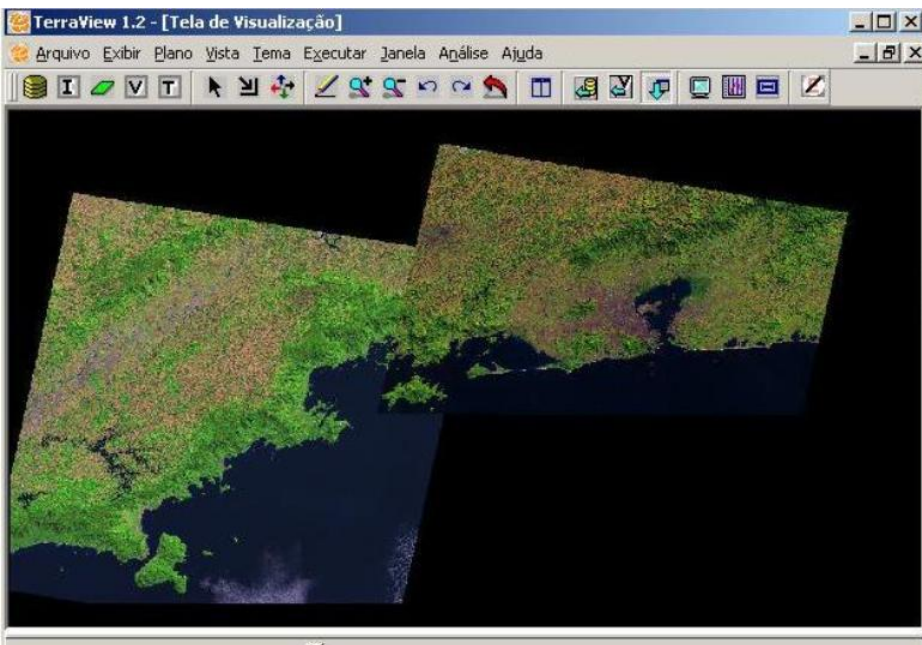
Operadores sobre Dados Vetoriais

- Distance between geometries
 - Distance (g1 Geometry, g2 Geometry) : Double Precision

- Spatial Operators:
 - Intersection (g1 Geometry, g2 Geometry) : Geometry
 - Difference (g1 Geometry, g2 Geometry) : Geometry
 - Union (g1 Geometry, g2 Geometry) : Geometry
 - SymDifference (g1 Geometry, g2 Geometry) : Geometry
 - Buffer (g1 Geometry, d double) : Geometry
 - ConvexHull (g1 Geometry) : Geometry

Suporte para Imagens

- Imagens de satélite
 - Importância crescente em GIS
 - Grandes volumes de dados
- Imagens na TerraLib
 - Particionamento em blocos para melhor indexação
 - Estrutura multi-nível para visualização eficiente



SQL extensions

- Vector x Raster (statistic values):
 - Count (g1 Geometry, r1 Raster) : Integer
 - Minimum (g1 Geometry, r1 Raster) : Double
 - Maximum (g1 Geometry, r1 Raster) : Double
 - Average (g1 Geometry, r1 Raster) : Double
 - Variance (g1 Geometry, r1 Raster) : Double
 - StdDeviation (g1 Geometry, r1 Raster) : Double
 - Median (g1 Geometry, r1 Raster) : Double
 - Value (point Geometry,r1 Raster) : Double

 - Others: assymetry, curtosis, coefficient of variation, mode

SQL extensions - Raster

- Function

- Histogram (r1 Raster) : Integer Array

- Spatial operators

- WC2RC (wc PointGeometry, r1 Raster) : PointGeometry
- RC2WC (rc PointGeometry, r1 Raster) : PointGeometry
- Mask (r1 Raster, r2 Raster) : Raster
- Mask (g1 Geometry, r1 Raster) : Raster
- Reclassify (r1 Raster, rl Rules) : Raster
- Slice (r1 Raster, rl Rules) : Raster
- Weight (r1 Raster,rl Rules) : Raster
- Calculate (r1 Raster,...,rn Raster,mathexp String) : Raster

Modelos Temporais na TerraLib

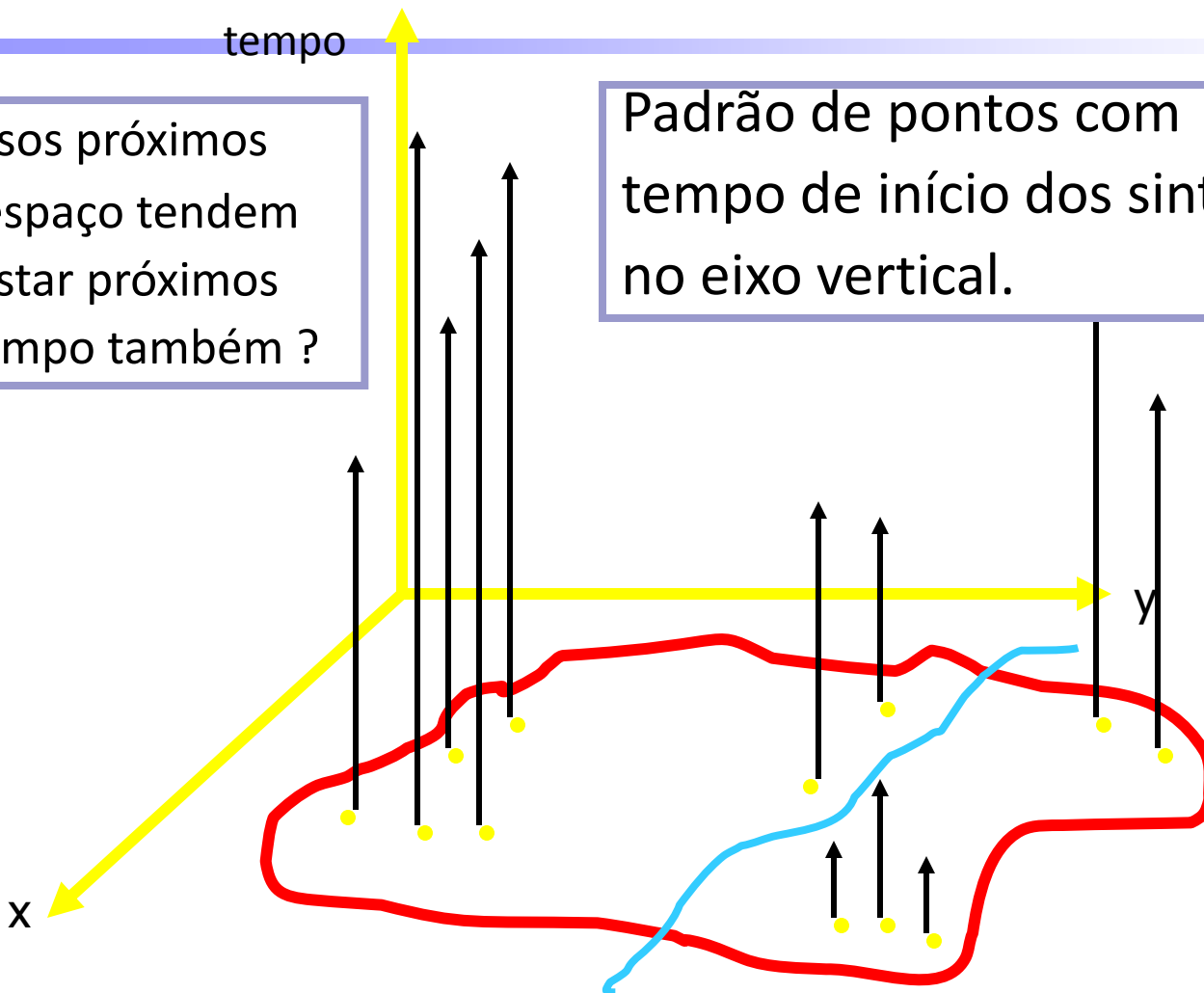
- Dados estáticos
 - Geometria fixa, atributos fixos
 - Ex: mapas tradicionais
- Eventos
 - Ocorrências independentes no tempo
 - Ex: Criminalidade
- Objetos dinâmicos
 - Geometria fixa, atributos variáveis
 - Ex: espaços celulares, censo
- Objetos móveis
 - Geometria variável, atributos variáveis
 - Ex: evolução de lotes num cadastro

Eventos

tempo

Casos próximos no espaço tendem a estar próximos no tempo também ?

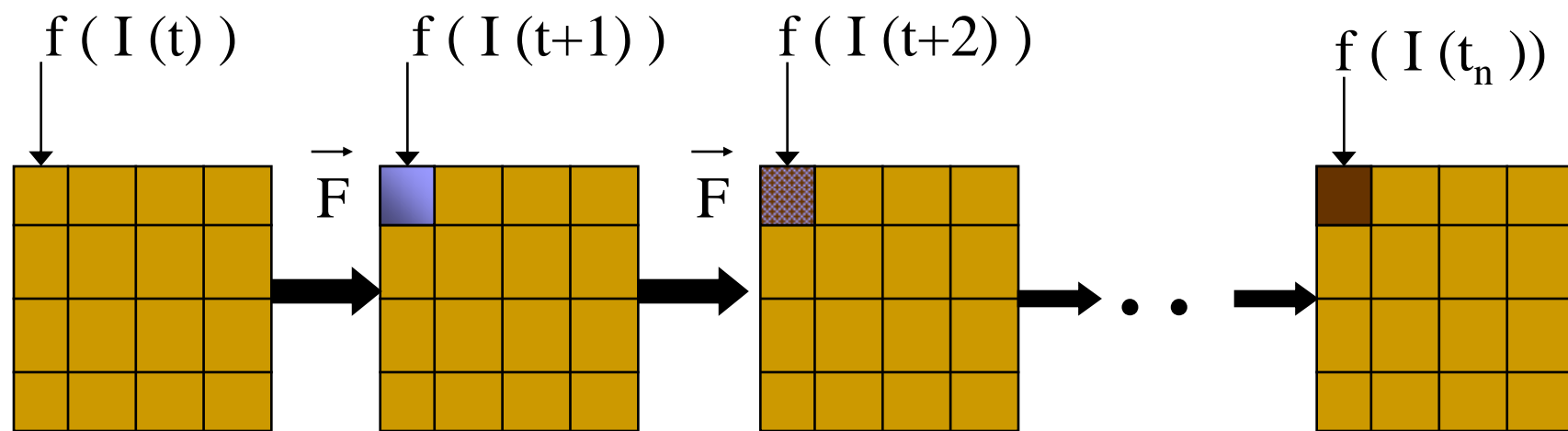
Padrão de pontos com tempo de início dos sintomas no eixo vertical.



Objetos Dinâmicos

- Geometria fixa, mudança em atributos
 - Partição do espaço constante
 - Suporte 0D (ponto), 1D (rede), 2D (celulas ou poligonos)
 - Modelos celulares
 - Aplicações em processos contínuos (“campos”)
 - Exemplos
 - Desmatamento na Amazônia
 - Processo hidrológico
 - Modelos de biodiversidade

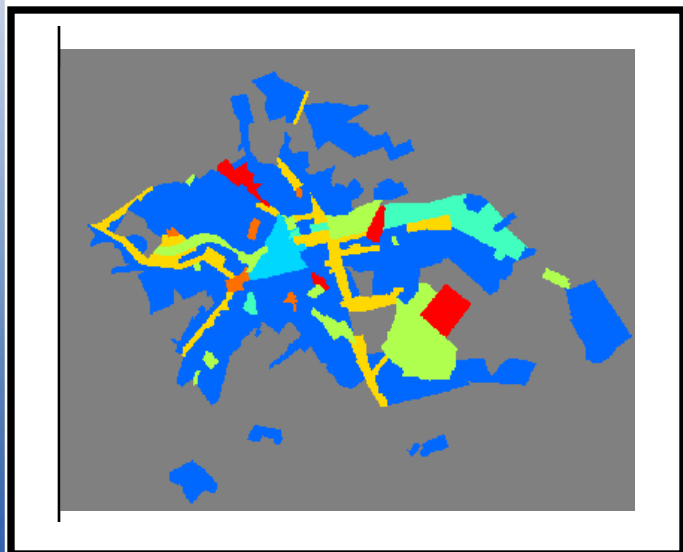
Modelos Espaciais Dinâmicos



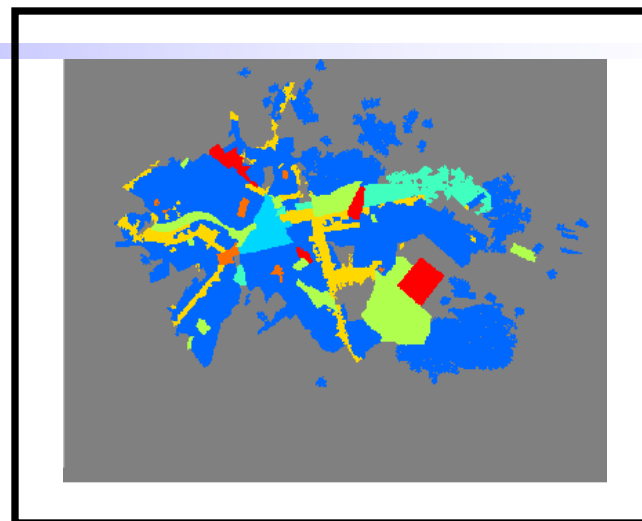
“Um modelo espacial dinâmico é uma representação matemática de um processo do mundo real em que uma localização na superfície terrestre muda em resposta a variações nas forças dirigidas” (Burrough)



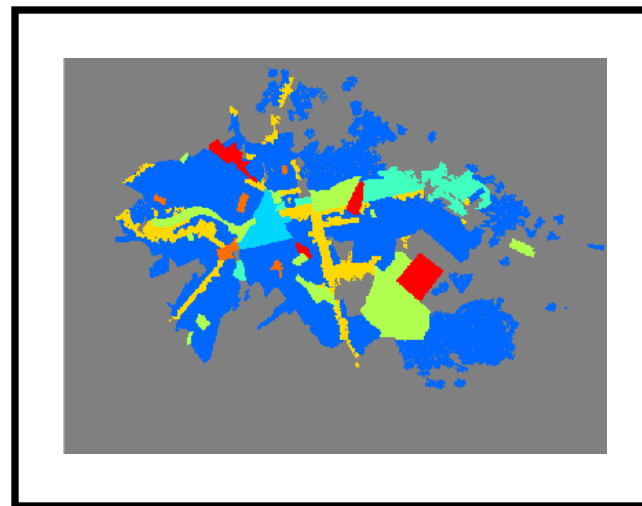
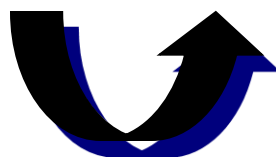
SIMULATIONS OUTPUTS



Reality - Bauru in 1988



S 2



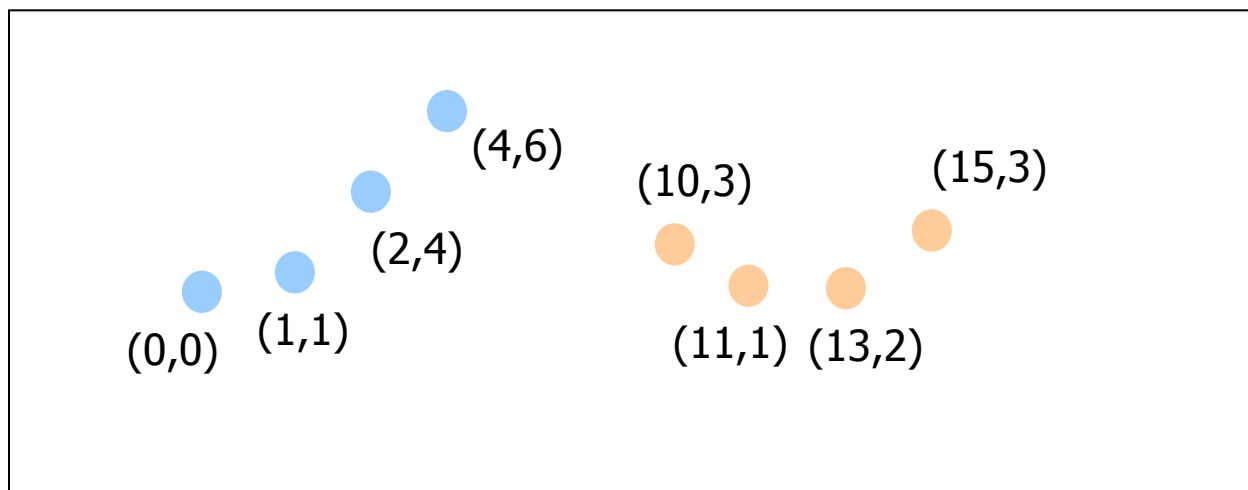
S 3

Objetos móveis

- Monitoramento de uma frota de caminhões: atributos e geometrias (posições do caminhão) mudam com o tempo
 - Em um intervalo de tempo regular (de 2 em 2 horas), a posição de cada caminhão e seus atributos (combustível e velocidade) são medidos; (síncrono)
 - Em outro intervalo de tempo regular (de 12 em 12 horas), a carga é medida; (síncrono)
 - A substituição de motoristas NÃO acontece em intervalos de tempo regulares; (assíncrono)
 - A cor do caminhão pode eventualmente mudar

Objetos móveis

- Monitoramento de uma frota de caminhões: atributos e geometrias (posições do caminhão) mudam com o tempo



Predicados Temporais

precedência

(-----)(.....)

(.....) intervalo

igualdade

(-----)

(.....)

sobreposição

(-----)

(.....)

pertinência

(-----)

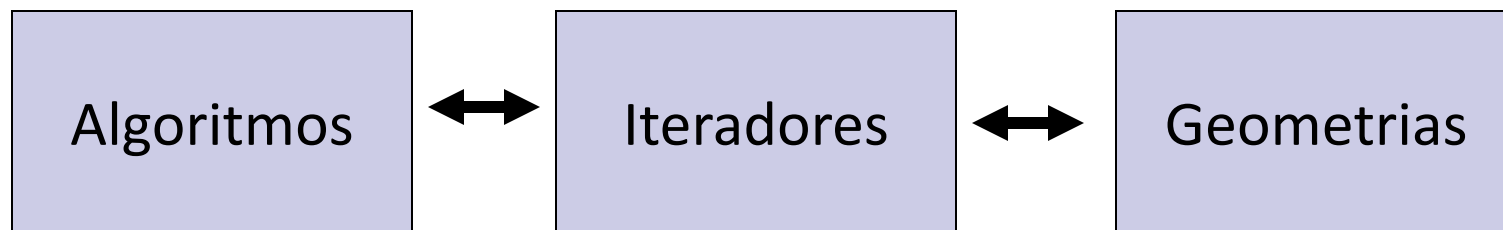
(.....)



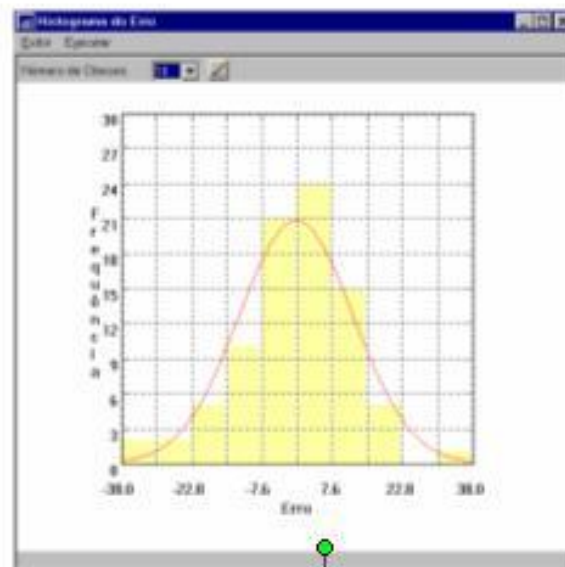
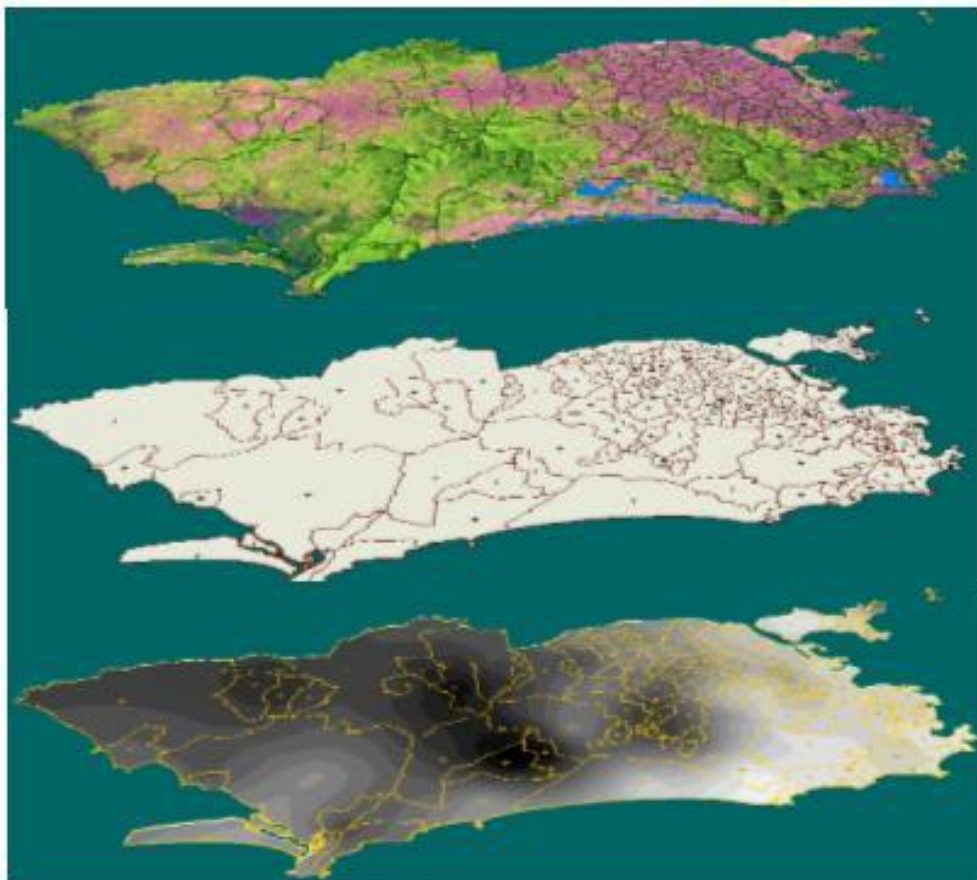
t

Programação Genérica em GIS

- Como desacoplar algoritmos de geometrias ?
 - Idea: Programação genérica (disp em C++)
 - *“Decide which algorithms you want; parametrize them so they work for a variety of suitable types and data structures”* (Stroustrup)



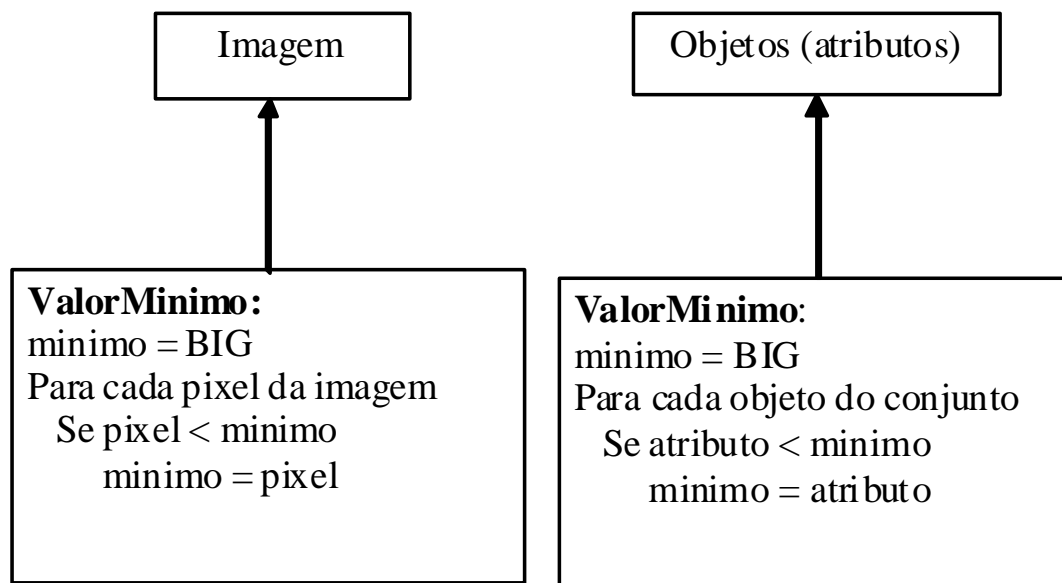
Mesmo Algoritmo, Geometrias Distintas



histogram

Algoritmos Genéricos para GIS

- Exemplo: calcular o valor mínimo
 - Tudo o que precisamos é acessar todos os valores
 - Vale tanto para imagens como para conjuntos de objetos

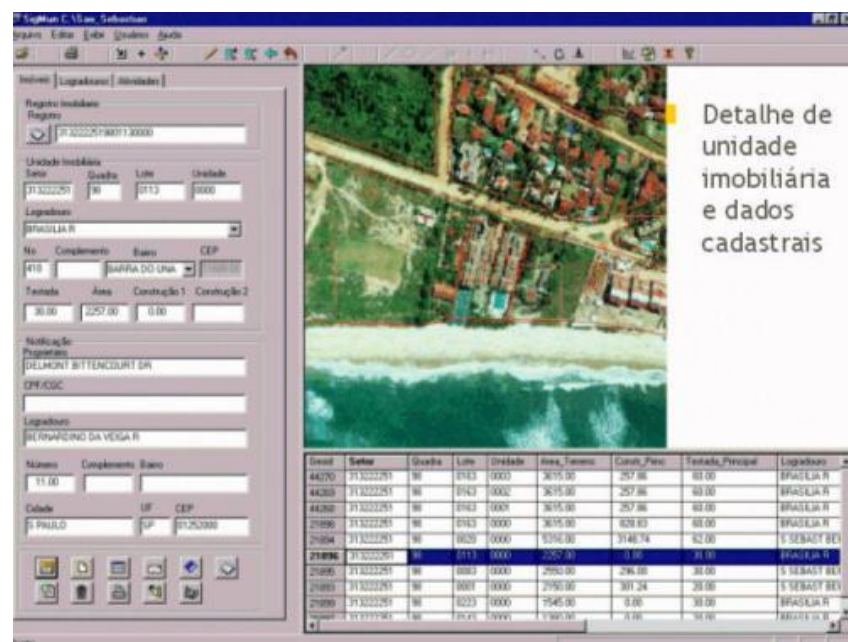


Análise Espacial na TerraLib

- Algoritmos em desenvolvimento
 - Análise de dados pontuais
 - Clustering espaço-tempo (UFMG)
 - Regionalização (EMBRAPA)
 - Geoestatística (INPE)
 - Regressão Espacial e não-espacial (INPE)

Aplicações TerraLib

- TerraView
 - Ambiente simples de visualização
- Mapeamento Urbano
 - Cadastro e Imagens
- Saúde Pública
 - Epidemiologia espacial
- Modelagem de uso do solo
 - Modelos espaço-temporais de desmatamento na Amazônia
- Plano de Ação de Emergência
 - Refinarias e Oleodutos (Petrobras)



Software Livre e Autonomia Tecnológica

- “Software livre” não substitui autonomia tecnológica

- Capacidade de desenvolvimento
 - formular agenda própria de P&D
 - realizar apropriação tecnológica de software livre
 - entender os produtos do mercado

- Software livre é mais que software de graça
 - Oportunidade histórica para capacitação nacional
 - Desenvolvimento de empresas de serviço em GIS independente de fabricantes

O modelo “caixa-branca”

Resultados = Software + Metodologia + Capacitação

■ Software

- Inovação apropriada aos usuários brasileiros
- Software aberto e acessível

■ Metodologia

- Traduzir conceitos em procedimentos operacionais
- Desenvolver exemplos de aplicações reais

■ Capacitação

- Tecnologia como produção social
- Grande ênfase em material didático e formação avançada

O modelo “caixa-branca”

- Quais as lições da história?
- Como os países desenvolvidos se tornaram desenvolvidos?

- Escolha crucial na área de tecnologia
 - “learning-by-doing” x “learning-by-using”
 - Quem aprende fazendo, aprende para sempre...
 - Quem aprende usando, precisa sempre aprender tudo de novo...
 - David Landes (“The Unbound Prometheus”)