

BRUNO DO CARMO GUIMARÃES

**UNISIG - SISTEMA PARA REPRESENTAR E AVALIAR IMAGENS
INTERPOLADAS**

Governador Valadares – MG

Novembro de 2002

**UNIVALE – UNIVERSIDADE VALE DO RIO DOCE
FATEC – FACULDADE DE CIÊNCIAS TECNOLÓGICAS**

BRUNO DO CARMO GUIMARÃES

**UNISIG - SISTEMA PARA REPRESENTAR E AVALIAR IMAGENS
INTERPOLADAS**

Trabalho de Diplomação referente à disciplina de Projeto de Sistema do 4º ano do Curso de Bacharelado em Ciência da Computação da Universidade Vale do Rio Doce. Orientado pelo professor Dr. Brauliro Gonçalves Leal.

Governador Valadares – MG

Novembro de 2002

BRUNO DO CARMO GUIMARÃES

**UNISIG - SISTEMA PARA REPRESENTAR E AVALIAR IMAGENS
INTERPOLADAS**

Aprovação: Governador Valadares – MG, 26 de Novembro de 2002

Prof. Dr. Erwin Doescher

Prof. Msc. Rogério Almeida Meneguim

Prof. Msc. Adelino Tolentino Pereira

Bsc. Bruno Barroso de Pinho

**Prof. Dr. Brauliro Gonçalves Leal
Orientador**

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA UNIVERSIDADE VALE DO RIO DOCE

Guimarães, Bruno do Carmo

UNISIG: sistema para representar e avaliar imagens interpoladas /
Bruno do Carmo Guimarães. – Governador Valadares: UNIVALE,
FATEC, 2002.

50p. : il. ; 29cm.

1. Sistema de Informações Geográficas. I. Título

CDU 004.92

DEDICATÓRIA

Dedico a realização deste trabalho aos meus pais Francisco e Dalva, aos meus irmãos Gibson e Mara, aos meus amigos e familiares.

Obrigado pelo apoio!

AGRADECIMENTOS

Agradeço a Deus, que esteve sempre ao meu lado, aos meus colegas de classe, que compartilharam comigo esta vitória, ao professor doutor Brauliro Gonçalves Leal, por todos os conhecimentos a mim passados, a todos professores do curso Ciência da Computação e, principalmente, aos meus pais.

SUMÁRIO

RESUMO	i
ABSTRACT	ii
1 INTRODUÇÃO	01
2 OBJETIVO	03
3 REVISÃO DE LITERATURA	04
3.1 HISTÓRICO DO GEOPROCESSAMENTO	04
3.2 SISTEMAS DE MEDIDA E DE REFERÊNCIA ESPAÇO-TEMPORAL	06
3.3 UNIVERSO DE REPRESENTAÇÃO	08
3.4 HIERARQUIA DE REPRESENTAÇÕES MATRICIAIS	10
3.5 COMPARAÇÃO ENTRE REPRESENTAÇÃO MATRICIAL E VETORIAL	11
3.6 DEFINIÇÃO DE MODELAGEM NUMÉRICA DE TERRENO	13
3.7 MODELOS DE GRADE REGULAR	14
3.8 INTERPOLADORES PARA GERAÇÃO DE GRADE REGULAR	15
3.8.1 INTERPOLADORES DE TENDÊNCIA	15
3.8.2 INTERPOLAÇÃO POR MEDIA MÓVEL	15
3.9 ANÁLISE SOBRE MODELOS DIGITAIS DE TERRENO	17
3.9.1 GERAÇÃO DE IMAGENS DE MDTs	17
3.9.1.1 IMAGEM MDT EM NÍVEIS DE CINZA	17
3.9.1.2 IMAGEM DE MDT SOMBREADA	18
3.9.1.3 FATIAMENTO DO MODELO	20
3.10 ÁLGEBRA DE MAPAS	22
3.10.1 OPERAÇÃO NUMÉRICA	22
3.10.2 OPERAÇÃO LÓGICA	23
3.10.3 INTERPOLAÇÃO	24
4 MATERIAIS E MÉTODOS	26
4.1 ESTRUTURA DO SISTEMA	26
4.2 INTERFACE DO SISTEMA	26
4.3 PRINCIPAIS FUNÇÕES	29
4.3.1 OPERAÇÕES MATEMÁTICAS	29
4.3.2 INTERPOLAÇÃO – INVERSO DO QUADRADO DA DISTÂNCIA	29
4.3.3 GERAÇÃO DE IMAGEM EM ESCALA DE CORES E EM ESCALA DE CINZA	30
4.3.4 EXPORTAÇÃO E IMPORTAÇÃO	30

5	RESULTADOS E DISCUSSÃO	31
5.1	EXPORTANDO UMA IMAGEM E IMPORTANDO UM ARQUIVO TEXTO	31
5.2	INTERPOLANDO	31
5.3	GERANDO IMAGEM EM ESCALA DE CORES E EM ESCALA DE CINZA	32
5.4	SOMANDO DUAS IMAGENS	32
5.5	SUBTRAINDO DUAS IMAGENS	33
5.6	MULTIPLICANDO UMA IMAGEM POR UM VALOR REAL	33
5.7	DIVIDINDO UMA IMAGEM POR UM VALOR REAL	34
5.8	COMPARANDO DUAS IMAGENS	34
6	CONCLUSÃO	35
7	REFERÊNCIAS BIBLIOGRÁFICAS	36
8	ANEXOS	37

RESUMO

Este trabalho apresenta uma das principais operações do SIG (Sistema de Informação Geográfica), a interpolação.

A partir de dados pontuais conhecidos, obtém-se pontos desconhecidos. Tal matriz de dados pode ser representada por meio de imagens nas quais pode-se aplicar as técnicas da álgebra de mapas.

Foi desenvolvido um sistema para representar e avaliar imagens interpoladas no qual se denominou-se UniSIG. Tal sistema foi desenvolvido na linguagem C++, utilizando a ferramenta C++ Builder 4.

Através do uso do sistema desenvolvido, foram gerados resultados de operações de interpolação e álgebra de mapas.

ABSTRACT

This work introduces one of the main operations of GIS (Geographical Information System), the interpolation.

The unknown points are obtained from known punctual points. The array can be represented by images, which algebra maps techniques could be apply.

A system was developed to represent and to evaluate interpolated images and it was called UniSIG. The system was developed in the C++ language, using the C++ Builder 4.

Using the developed system, results of interpolation and algebra maps were generated.

1 INTRODUÇÃO

Um Sistema de Informação Geográfica (SIG) pode ser definido como uma estrutura organizada de hardware, software, dados geográficos e pessoal envolvido no trabalho, projetado para capturar, armazenar, atualizar, manipular, analisar e apresentar todas as formas de informações referenciadas geograficamente. Conceitualmente, um SIG é um mapeador automatizado ou um sistema de mapeamento temático (CÂMARA, 2002).

O SIG tem causado impacto positivo em todos os campos do conhecimento que manipulam e analisam dados distribuídos espacialmente. O SIG pode torna-se uma extensão do nosso pensamento analítico, permitindo pensar sobre os padrões, o espaço e os processos que nele atuam (EASTMAN, 1992).

Os sistemas de informações geográficas têm desempenhado um papel importante como integrador de tecnologias. Ao invés de ser de natureza completamente nova, eles unem várias tecnologias discretas como um todo.

O SIG vem emergindo como uma poderosa tecnologia, permitindo integrar seus dados e métodos de maneiras que apoiam as formas tradicionais de análise geográfica, tal como análise por sobreposição de mapas, e ainda novos tipos de análises e modelagem que vão além da capacidade de métodos manuais. Com o SIG é possível elaborar mapas, modelar, fazer buscas e analisar grandes quantidades de dados mantidos em banco de dados.

O desenvolvimento do SIG tem se fundamentado em inovações que ocorreram em áreas do conhecimento distintas: Geografia, Cartografia, Fotogrametria, Sensoriamento Remoto, Topografia, Geodésia, Engenharia Civil, Estatística, Ciência da Computação, Pesquisas Operacionais, Inteligência Artificial, Demografia, além de outros ramos das Ciências Sociais, Ciências Naturais e Engenharias.

Na área de engenharia urbana, os Sistemas de Informações Geográficas são úteis na avaliação da situação atual das mais diversas atividades urbanas tais como: serviços e transporte, equipamentos urbanos, uso e ocupação do solo, cadastramento imobiliário gráfico, arruamentos, escolas, hospitais, postos de saúde, áreas de lazer, áreas de proteção ambiental e redes de infra-estrutura urbana.

Neste projeto será apresentado uma das principais operações do SIG, a interpolação. A interpolação é uma técnica que consiste em obter pontos desconhecidos a partir de outros conhecidos. Tal matriz de dados pode ser representada por meio de imagens nas quais pode-se aplicar as técnicas da álgebra de mapas.

O sistema foi desenvolvido na linguagem de programação C++, utilizando a ferramenta C++ Builder 4, devido suas características de orientação a objeto e por ser uma ferramenta RAD (Rapidly Application Development).

2 OBJETIVO

O objetivo deste trabalho é desenvolver um software que implemente as técnicas da álgebra de mapas, obtidos a partir da interpolação de dados pontuais.

3 REVISÃO DE LITERATURA

3.1 HISTÓRICO DO GEOPROCESSAMENTO

As primeiras tentativas de automatizar parte do processamento de dados com características espaciais aconteceram na Inglaterra e nos Estados Unidos, nos anos 50, com o objetivo principal de reduzir os custos de produção e manutenção de mapas (CÂMARA, 2002). Dada a precariedade da informática na época e a especificidade das aplicações desenvolvidas (pesquisa em botânica, na Inglaterra, e estudos de volume de tráfego, nos Estados Unidos), estes sistemas ainda não poderiam ser classificados como “sistemas de informação”.

Os primeiros Sistemas de Informação Geográfica surgiram na década de 60, no Canadá, como parte de um programa governamental para criar um inventário de recursos naturais (CÂMARA, 2002). Estes sistemas, no entanto, eram muito difíceis de usar pois, não existiam monitores gráficos de alta resolução, os computadores necessários eram excessivamente caros, e a mão de obra deveria ser altamente especializada, caríssima naquela época. Não existiam soluções comerciais prontas para uso, e cada interessado precisava desenvolver seus próprios programas, o que demandava muito tempo e, naturalmente, muito investimento financeiro.

Além disto, a capacidade de armazenamento e a velocidade de processamento eram insuficientes. Ao longo dos anos 70, foram desenvolvidos novos recursos de hardware, tornando viável o desenvolvimento de sistemas comerciais. Foi então que a expressão Geographic Information System foi criada. Surgiu também nesta os primeiros sistemas comerciais de CAD (Computer Aided Design, ou projeto assistido por computador), que melhoraram substancialmente as condições para a produção de desenhos e plantas para engenharia, e serviram de base para os primeiros sistemas de cartografia automatizada. Também nos anos 70, foram desenvolvidos alguns fundamentos matemáticos voltados para a cartografia, incluindo questões de geometria computacional. No entanto, devido aos custos e ao fato destes proto-sistemas ainda utilizarem exclusivamente computadores de grande porte, apenas grandes organizações tinham acesso tal tecnologia (CÂMARA, 2002).

A década de 80 representa o avanço acelerado na tecnologia de sistemas de informação geográfica que inicia um período que perdura até os dias de hoje. Até então, limitados pelo alto custo do hardware e pela pouca quantidade de pesquisa específica sobre o

tema, o SIG se beneficiou grandemente pela massificação causada pelos avanços dos computadores pessoais e do estabelecimento de centros de estudos sobre o assunto. Nos EUA, a criação dos centros de pesquisa que formam o NCGIA – National Centre for Geographical Information and Analysis, marca o estabelecimento do Geoprocessamento como área científica independente.

No decorrer dos anos 80, com a grande popularização e redução de custos das estações de trabalho gráficas, além do surgimento e evolução dos computadores pessoais e dos sistemas gerenciadores de bancos de dados relacionais, ocorreu uma grande difusão do uso de SCSIG (Softwares Computacionais para SIG) e suas aplicações. A incorporação de muitas funções de análise espacial proporcionou também um aumento das aplicações de SCSIG. Na década de 90, observou-se um grande crescimento do ritmo de utilização do SIG pelas organizações, sempre impulsionados pelos custos decrescentes dos hardware e software, e também pelo surgimento de alternativas de menor custo para a construção de bases de dados geográficas.

3.2 SISTEMAS DE MEDIDA E DE REFERÊNCIA ESPAÇO-TEMPORAL

O aspecto central no uso da tecnologia de Geoprocessamento advém da característica lógico-matemática de sistemas de informação. Para ser representada em ambiente computacional, tem-se que associar a cada tipo de informação geográfica uma escala de medida e de referência, que é utilizada pelo SIG para caracterizá-lo.

A técnica mais aplicada em Geoprocessamento utiliza uma escala de medida, que permite associar grandezas numéricas a cada objeto a ser representado. Esta visão deriva do conceito “representacionalista” proposto pelo filósofo Bertrand Russel, em que as propriedades não são intrínsecas aos objetos, mas são obtidas a partir de medidas. Assim, a representação de um objeto geográfico num SIG dependerá da escala utilizada (CÂMARA, 2002).

Uma característica importante dos níveis de medidas temáticas é que elas não determinam magnitude. Quando o estudo necessita de uma descrição mais detalhada, que permita comparar intervalo e ordem de grandeza entre eventos, recorre-se aos níveis de medidas denominados numéricos, onde as regras de atribuição de valores baseiam-se numa escala numérica, geralmente um sub conjunto dos números reais.

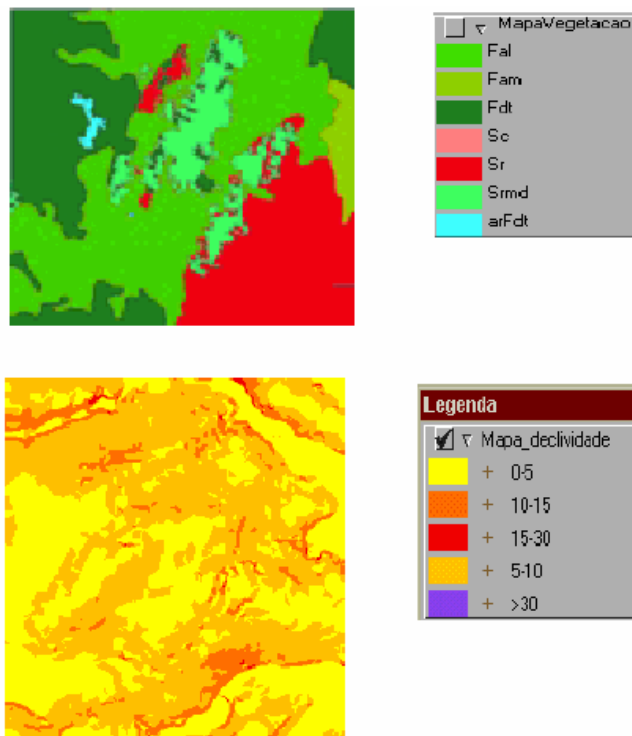


Figura 1 - Exemplos de medida nominal (mapa de vegetação) e medida ordinal (mapa de classes de declividade).

Existem dois níveis de medidas baseados em escala de números reais: o nível por intervalo e o nível por razão. No nível de medida por intervalo, o ponto de referência zero é definido de forma arbitrária, permitindo a atribuição de valores negativos, positivos e nulos para as amostras. O equador e o meridiano de Greenwich, usados com referência na determinação de posições sobre a superfície da Terra, são exemplos de referência zero arbitrária e a localização geográfica em latitude/longitude, o são para variáveis descritas no nível de medida por intervalo. Por ter um ponto de referência zero arbitrária, valores medidos no nível por intervalo não podem ser usados para estimar proporções (CÂMARA, 2002).

No nível de medida por razão, o ponto de referência zero não é arbitrário, mas determinado por alguma condição natural. Por exemplo, na descrição de atributos como o peso, a distância entre dois pontos, a área e o volume de objetos, não faz sentido físicos valores negativos, sendo a ausência destes atributos o ponto de origem zero na escala de medida correspondente. Por ter como referência um ponto de zero absoluto, as medidas feitas no nível de medida por razão permitem estimar proporções e podem ser usadas em operações de multiplicação, divisão e subtração entre amostras.

As medidas temáticas e as numéricas por intervalo não devem ser usadas diretamente em expressões matemáticas. Entretanto, na prática, os modelos ambientais combinam valores por razão com valores por intervalo. Nestes casos, parâmetros devem ser incluídos para permitir a conversão de valores medidos no nível por intervalo para o nível por razão, em unidades apropriadas (CÂMARA, 2002).

3.3 UNIVERSO DE REPRESENTAÇÃO

No universo de representação, definem-se as possíveis representações geométricas que podem estar associadas às classes do universo conceitual. Inicialmente, deve-se considerar as duas grandes classes de representações geométricas: representação vetorial e matricial.

Através da representação vetorial, a representação de um elemento ou objeto busca reproduzi-lo o mais exatamente possível. Qualquer entidade ou elemento gráfico de um mapa é reduzido a três formas básicas: pontos, linhas, áreas ou polígonos.

A representação matricial consiste no uso de uma malha quadriculada regular sobre a qual se constrói, célula a célula, o elemento que está sendo representado. A cada célula, atribui-se um código referente ao atributo estudado, de tal forma que o computador saiba a que elemento ou objeto pertence determinada célula.

Vale ressaltar que as representações estão associadas aos tipos de dados, a saber:

- dados temáticos: admitem tanto representação matricial quanto vetorial;
- dados cadastrais: sua parte gráfica é armazenada em forma de coordenadas vetoriais e seus atributos não gráficos são armazenados em um banco de dados;
- redes: sua parte gráfica é armazenada em forma de coordenadas vetoriais, com a topologia arco-nó e seus atributos não gráficos são armazenados em um banco de dados;
- imagens de sensoriamento remoto: armazenadas em representação matricial;
- modelos numéricos de terreno: podem ser armazenados em grades regulares (representação matricial), grades triangulares (representação vetorial com topologia arco-nó) ou isolinhas (representação vetorial sem topologia).

Através da representação matricial, o espaço é representado como uma matriz $P(m, n)$ composto de m colunas e n linhas, onde cada célula possui um número de linha, um número de coluna e um valor correspondente ao atributo estudado, e cada célula é individualmente acessada pelas suas coordenadas.

A representação matricial supõe que o espaço pode ser tratado como uma superfície plana, onde cada célula está associada a uma porção do terreno. A resolução do sistema é dada pela relação entre o tamanho da célula no mapa ou documento e a área por ela coberta no terreno. A Figura 2 apresenta um mapa representado por células de diferentes tamanhos (diferentes resoluções), representando diferentes áreas no terreno.

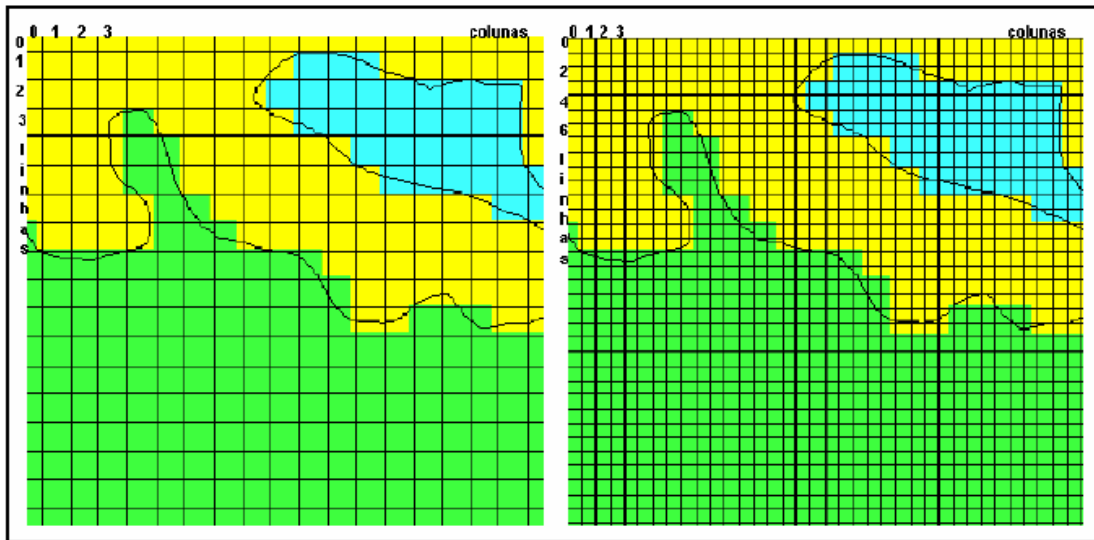


Figura 2 - Diferentes representações matriciais para um mapa.

Como o mapa do lado esquerdo da Figura possui uma resolução quatro vezes menor que o do mapa do lado direito, as avaliações de áreas e distâncias serão menos exatas que no primeiro. Em contrapartida, o espaço de armazenamento necessário para o mapa da direita será quatro vezes maior que o da esquerda.

Os dados são codificados, célula a célula, atribuindo a cada uma, o código correspondente à uma classe referente ao fenômeno estudado. Para fazer isto, é necessário estabelecer um critério a ser obedecido em toda a operação.

Pode-se, por exemplo, atribuir a cada célula o código da classe sobre a qual estiver o centro da quadrícula. Outra possibilidade é adotar o critério da maior ocorrência. Neste caso, o código corresponde ao da classe que ocupar a maior parte da célula.

3.4 HIERARQUIA DE REPRESENTAÇÕES MATRICIAIS

A representação geométrica matricial pode ser especializada segundo a hierarquia de classes apresentada na Figura 3.

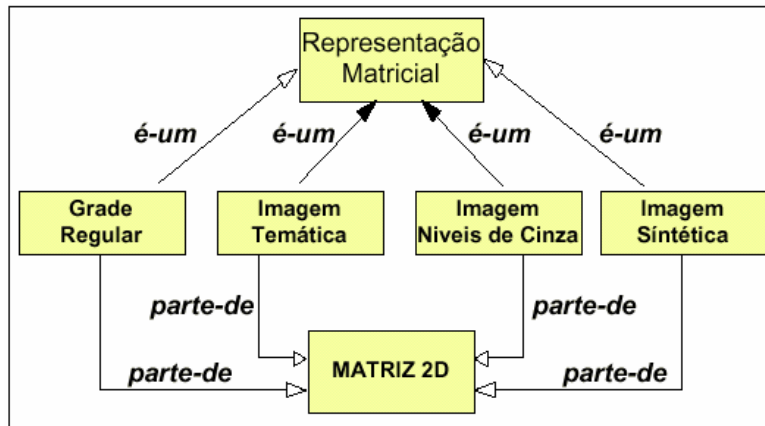


Figura 3 - Hierarquia de classes para representação matricial.

Os tipos possíveis de representação matricial são:

- Grade Regular: uma grade regular é uma matriz de números reais;
- Imagem em Tons de Cinza: imagem representada através de uma matriz onde os valores da matriz representam os valores de cinza da imagem;
- Imagem Temática: representação matricial de um geo-campo temático.
- Imagem Sintética (ou Codificada): representação de uma imagem em cores, utilizada para mostrar imagens em composição colorida em placas gráficas falsa-cor.

3.5 COMPARAÇÃO ENTRE REPRESENTAÇÃO MATRICIAL E VETORIAL

Como observamos anteriormente, dados temáticos admitem tanto a representação matricial quanto a vetorial, deste modo, é relevante compará-las.

Para a produção de cartas e em operações onde se requer maior precisão, a representação vetorial é mais adequada. As operações de álgebra de mapas são mais facilmente realizadas no formato matricial. No entanto, para um mesmo grau de precisão, o espaço de armazenamento requerido por uma representação matricial é substancialmente maior. Isto é apresentado na Figura 4.

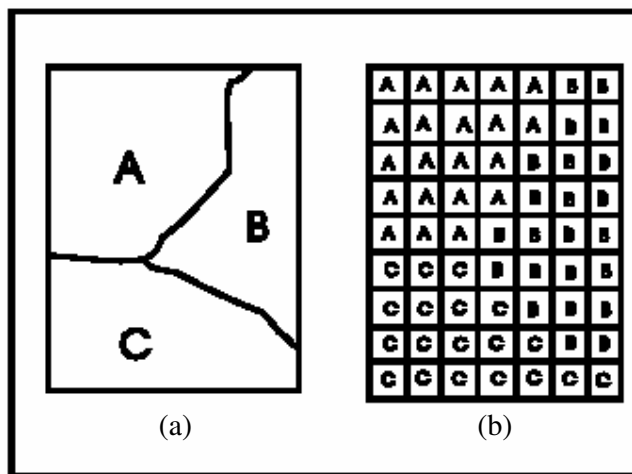


Figura 4 - Representação vetorial (a) e matricial (b) de um mapa temático.

A Tabela 1 apresenta uma comparação entre as vantagens e limitações de armazenamento matricial e vetorial para mapas temáticos. Esta comparação considera vários aspectos como relacionamentos espaciais, análise, armazenamento. Nesta tabela, o formato mais vantajoso para cada caso é apresentado em destaque.

Tabela 1 - Comparação entre representações para mapas temáticos (CÂMARA, 2002).

Aspecto	Representação Vetorial	Representação Matricial
Relações espaciais entre objetos	Relacionamento topológicos entre objetos disponíveis	Relacionamentos espaciais devem ser inferidos
Ligação com banco de dados	Facilita associar atributos a elementos gráficos	Associa atributos apenas a classes do mapa
Análise, Simulação e Modelagem	Representação indireta de fenômenos contínuos. Álgebra de mapas é limitada	Representa melhor fenômenos com a variação contínua no espaço Simulação e modelagem mais fáceis
Escalas de trabalho	Adequado tanto a grandes quanto a pequenas escalas	Mais adequado para pequenas escalas (1:25.000 e menores)
Algoritmos	Problemas com erros geométricos	Processamento mais rápido e eficiente
Armazenamento	Por coordenadas (mais eficiente)	Por matrizes

3.6 DEFINIÇÃO DE MODELAGEM NUMÉRICA DE TERRENO

Um Modelo Numérico de Terreno (MNT) é uma representação matemática computacional da distribuição de um fenômeno espacial que ocorre dentro de uma região da superfície terrestre. Dados de relevo, informação geológicas, levantamentos de profundidades do mar ou cursos d'água, informação meteorológicas e dados geofísicos e geoquímicos, são exemplos típicos de fenômenos representados por um MNT.

Segundo Burrough (1986), os principais usos do MNT são:

- Armazenamento de dados de altimetria para gerar mapas topográficos;
- Análises de corte-aterro para projeto de estradas e barragens;
- Elaboração de mapas de declividade e exposição para apoio a análise de geomorfologia e erodibilidade;
- Análise de variáveis geofísicas e geoquímicas;
- Apresentação tridimensional (em combinação com outras variáveis).

Para representar superfícies reais em modelos computacionais é indispensável a elaboração e criação de modelos digitais, que pode estar representado por equações analíticas ou uma rede (grade) de pontos, de modo a transmitir ao usuário as características espaciais do terreno.

A criação de um modelo numérico de terreno corresponde a uma nova maneira de focar o problema da elaboração e implantação de projetos. A partir dos modelos (grades) pode-se calcular diretamente volumes, áreas, desenhar perfis e secções transversais, gerar imagens sombreadas ou em níveis de cinza, gerar mapas de declividade e aspecto, gerar fatiamentos nos intervalos desejados e perspectivas tridimensionais.

3.7 MODELOS DE GRADE REGULAR

Uma grade regular retangular é um poliedro de faces retangulares. O processo de geração de uma grade regular consiste em estimar os valores de cota de cada ponto da grade a partir do conjunto de amostras de entrada. Quando se faz uso de todas as amostras para interpolar cada ponto da grade diz-se que a interpolação é global. Em geral, o conjunto de amostras pode ser muito grande e não homogêneo, tornando a interpolação global pouco apropriada em relação ao tempo de processamento computacional e precisão do modelo. Por exemplo, para uma grande quantidade de amostras, fica bastante difícil definir-se uma função polinomial capaz de representar satisfatoriamente todas as variações espaciais do fenômeno em estudo. Assim, é muito comum utilizar-se interpolação local para o cálculo de cota de cada elemento da grade. Neste caso, o valor de cota de cada elemento da grade é estimado a partir de uma quantidade preestabelecida de amostras vizinhas mais próximas do elemento.

Um modelo de grade regular retangular pode ser gerado a partir de um conjunto de amostras regularmente espaçadas ou a partir de um conjunto de amostras irregularmente espaçadas. O processo de geração de uma grade regular retangular a partir de outra grade também regular retangular, objetivando uma melhora na resolução da grade, é conhecido como refinamento da grade.

Para se gerar grades regulares retangulares a partir de amostras irregularmente espaçadas é comum definir-se funções interpolantes simples (por exemplo, médias ponderadas, splines, entre outras) e utilizá-las localmente. É importante notar que a estimativa da cota z de cada ponto da grade é realizada a partir de amostras vizinhas locais, mas uma análise global das amostras é sempre necessária para se encontrar essa vizinhança.

3.8 INTERPOLADORES PARA GERAÇÃO DE GRADE REGULAR

3.8.1 INTERPOLADORES DE TENDÊNCIA

Os interpoladores de tendência representam as variações do fenômeno estudado a partir de regressões polinomiais. A idéia é ajustar uma superfície polinomial sobre todos os dados amostrais minimizando-se o erro de ajuste global. Geralmente utiliza-se uma minimização do erro médio quadrático. A Figura 5 apresenta esse conceito para o caso de ajuste de curvas.

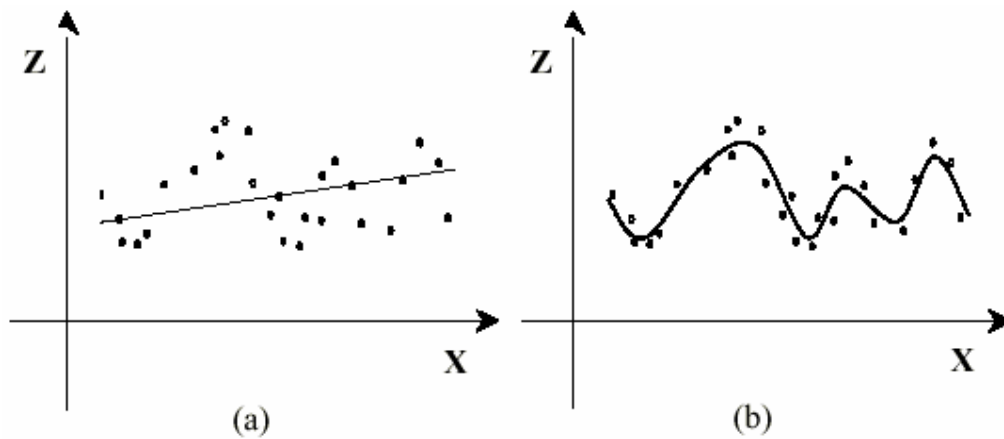


Figura 5 - Curva polinomial ajustada para representar a variação de um conjunto de amostra: (a) linear e (b) não linear

3.8.2 INTERPOLAÇÃO POR MEDIA MÓVEL

A média das cotas das amostras vizinhas, também conhecido como média móvel, é um dos esquemas de interpolação mais simples para estimação dos valores de cota dos pontos de uma grade regular retangular.

A formulação geral para este tipo de interpolação é dada pela Equação 1:

$$z_i = \frac{\sum_{j=1}^n w_{ij} z_{ij}}{\sum_{j=1}^n w_{ij}} \quad (1)$$

Em que:

- Z_i é o valor de cota de um ponto i qualquer da grade;
- Z_j é a cota de uma amostra j vizinha do ponto i da grade e;
- W_{ij} é um fator de ponderação.

Variações desse esquema básico são: interpolador por vizinho mais próximo; por média simples; por média ponderada; por média ponderada por quadrante, e; interpolador por média ponderada por quadrante e por cota.

A interpolação por vizinho mais próximo é definida pela escolha de apenas uma amostra vizinha para cada ponto da grade, ou seja, quando j igual a 1.

A interpolação por média simples considera o valor de cota z do elemento da grade igual a média aritmética dos valores de cota das amostras vizinhas. Neste caso considera-se que o fator de ponderação w_{ij} é igual a 1 para qualquer amostra vizinha.

Na interpolação por média ponderada, o valor de cota de cada elemento da grade é definido pela média ponderada dos valores de cota das amostras vizinhas. A ponderação mais usada na prática é o inverso da distância euclidiana do ponto da grade à amostra considerada, Equação 2.

$$w_{ij} = \frac{1}{d_{ij}^k} \quad (2)$$

Em que:

- k é o expoente da distância, geralmente igual a 1 ou 2, e;
- d_{ij} é o valor de distância da amostra j ao ponto i da grade, Equação 3.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3)$$

É intuitivo considerar-se que, quanto mais distante, uma amostra se encontra do ponto da grade, menor deve ser a sua influência (peso) no cômputo final do valor de cota do ponto da grade.

3.9 ANÁLISE SOBRE MODELOS DIGITAIS DE TERRENO

3.9.1 GERAÇÃO DE IMAGENS DE MDTs

A partir de um modelo digital de terreno (MDT) pode-se criar imagens em níveis de cinza, imagens sombreadas e escala de cores.

3.9.1.1 IMAGEM MDT EM NÍVEIS DE CINZA

São geradas diretamente do mapeamento dos valores de cota do modelo para valores de nível de cinza. Supondo-se que cada pixel da imagem de nível de cinza é representada por 8 bits, pode-se mapear os valores de cota para valores de nível de cinza de 1 a 255. A Figura 6 apresenta o mapeamento de um valor de cota Z_i para um valor de nível de cinza NC_i ;

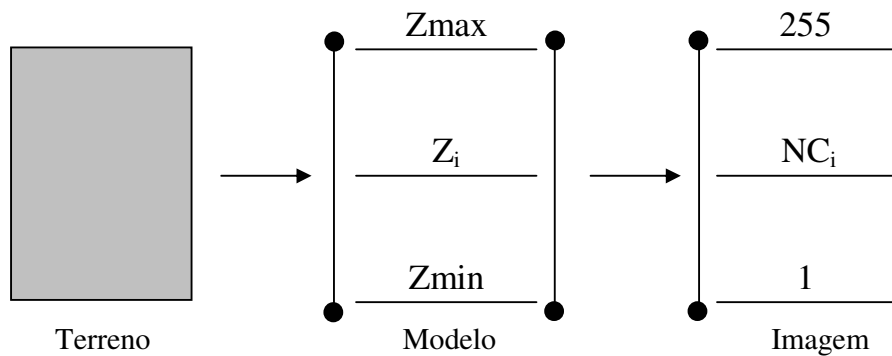


Figura 6 - Mapeamento dos valores de cota para níveis de cinza

Considerando-se um mapeamento linear dos valores de cota do modelo para valores de níveis de cinza, quantização linear, pode-se calcular o valor NC_i em função de Z_i por meio da Equação 4:

$$NC_i = \frac{254(Z_i - Z_{min})}{(Z_{max} - Z_{min})} + 1 \quad (4)$$

A Equação 4 mapeia os valores de cota do fenômeno representado para valores de 1 a 255 da imagem. O valor de nível de cinza igual a 0 é usado em áreas onde não existe definição do valor de cota para o modelo.

A Figura 7 apresenta uma imagem em nível de cinza obtida a partir de um modelo digital de terreno, onde os pontos claros representam um nível altimétrico menos acentuado e os pontos escuros, um nível mais acentuado. Essa imagem é muito útil para se obter uma

percepção qualitativa global da variação do fenômeno representado pelo modelo digital de terreno.

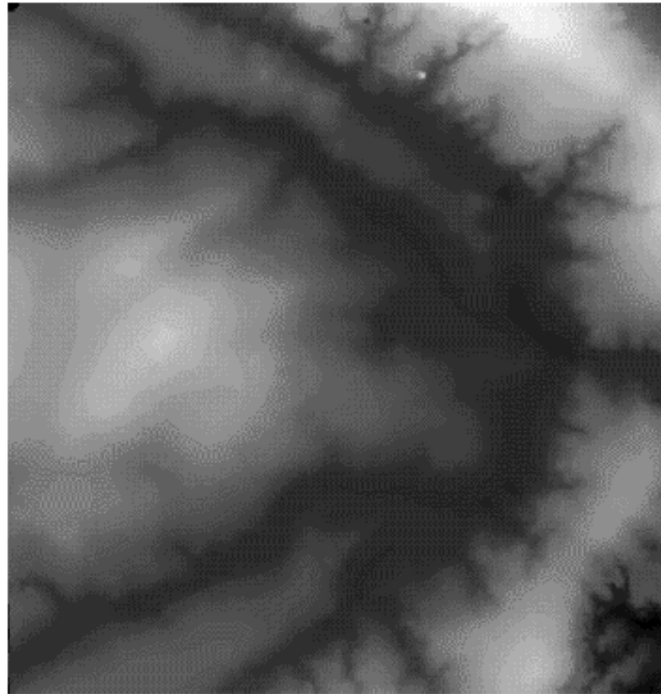


Figura 7 - Modelo de grade regular representado como uma imagem em níveis de cinza (CÂMARA, 2002).

3.9.1.2 IMAGEM DE MDT SOMBREADA

Uma imagem de MDT sombreada é gerada a partir do modelo e do posicionamento, em relação à superfície, de uma fonte de iluminação local. Assim, para cada ponto do modelo pode-se definir um vetor normal a superfície N e um vetor de iluminação I que parte do ponto da superfície e aponta para a fonte de iluminação. A partir desses dois valores pode-se calcular um valor de intensidade de iluminação utilizando-se, por exemplo, o seguinte modelo de iluminação de reflexão difusa, Equação 5.

$$NC_i = (I_a + I_l) K_d (N_i L_i) = (I_a + I_l) K_d (\cos \theta) \quad (5)$$

Esse modelo estabelece que o nível de cinza da imagem sombreada NC_i , na posição i da superfície, é igual a uma intensidade constante de iluminação ambiente I_a somado a uma componente de iluminação local. Essa componente local depende da intensidade da fonte luminosa I_l , de uma constante de reflexão difusa K_d e do ângulo θ formado entre o vetor

unitário normal N e o vetor unitário de iluminação L na posição i da superfície. A Figura 8 apresenta esses conceitos.

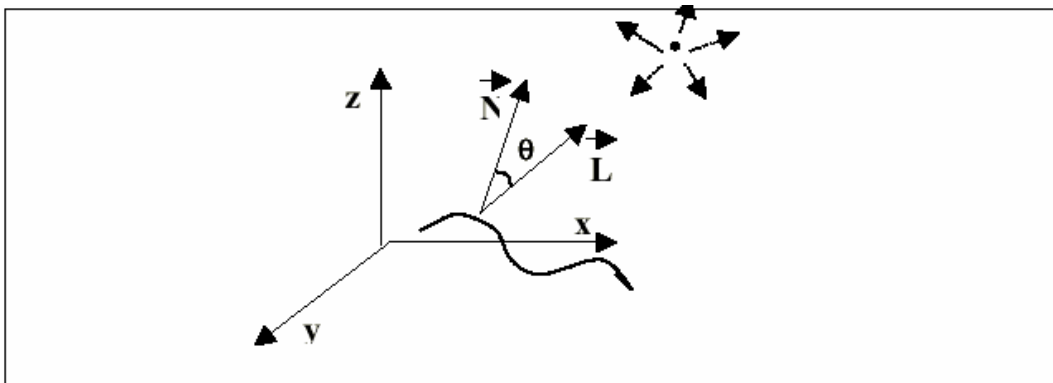


Figura 8 - Componentes do modelo de iluminação de uma superfície por reflexão difusa

O modelo de reflexão difusa, ou modelo lambertiano, é um dos mais simples de ser implementado computacionalmente. Outros modelos mais complexos podem ser utilizados quando se deseja um efeito de sombreado mais realista. A imagem sombreada é muito útil como imagem de textura para compor uma projeção geométrica planar utilizando-se o modelo digital de terreno. A Figura 9 apresenta uma imagem sombreada gerada a partir de um modelo numérico de terreno, onde as sombras mais claras representam maiores cotas de Z e as mais escuras, menores cotas de Z .

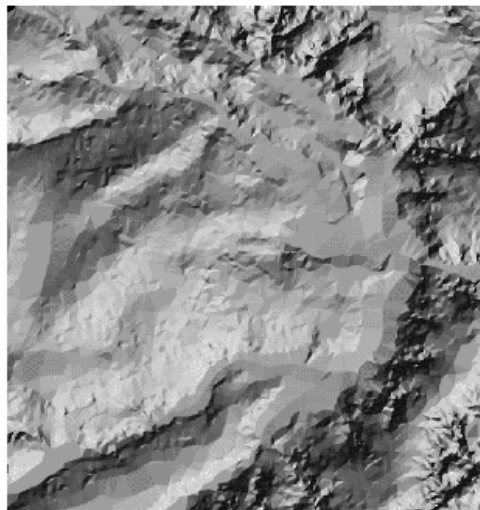


Figura 9 - Modelo de grade regular representado como uma imagem sombreada (CÂMARA, 2002).

3.9.1.3 FATIAMENTO DO MODELO

O fatiamento de um modelo consiste em se definir intervalos, ou fatias, de cotas com a finalidade de se gerar uma imagem temática a partir do modelo. Assim, cada tema, ou classe, da imagem temática é associado a um intervalo de cotas dentro dos valores atribuídos ao fenômeno modelado. O usuário do SIG é responsável pela definição das fatias e também da associação dessas fatias com classes predefinidas no sistema. As fatias são definidas de acordo com intervalos de cotas que são úteis para uma determinada aplicação. A Figura 10 apresenta o processo de fatiamento do modelo.

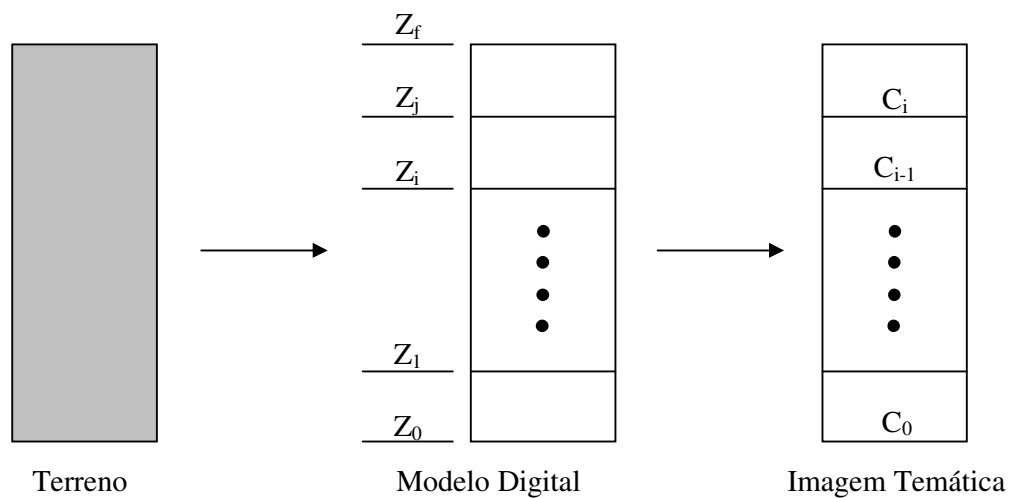


Figura 10 - Ilustração do processo de fatiamento do MDT

Considerando a Figura 10, tem-se que a classe C_0 está associada ao intervalo de cotas entre Z_0 e Z_1 , a classe C_i está associada ao intervalo de cotas entre Z_i e Z_j , e assim por diante. Os valores de cota Z_0 e Z_f representam os extremos desse modelo. Quando um intervalo de cota não for associado a nenhuma classe deve-se atribuir o valor de nível de cinza 0 para os pixels da imagem que representam valores de cota nesse intervalo.

A imagem temática gerada pelo fatiamento do modelo é frequentemente utilizada em análises espaciais como operações lógicas de interseção e união. A Figura 11 apresenta uma imagem temática criada a partir de um modelo digital de terreno.

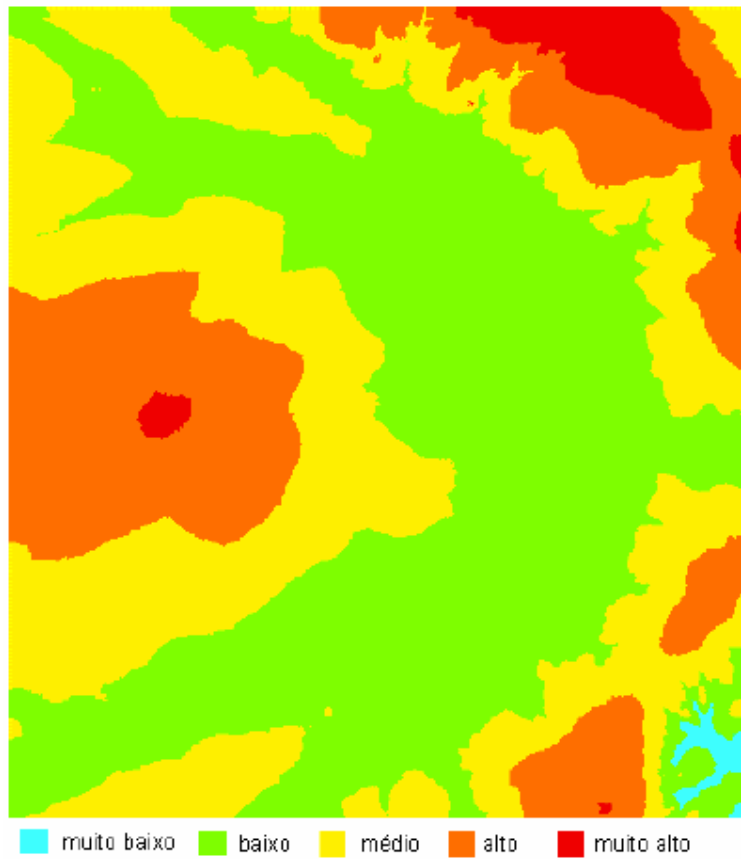


Figura 11 - Imagem temática gerada a partir do fatiamento de um modelo digital de terreno (CÂMARA, 2002).

3.10 ÁLGEBRA DE MAPAS

3.10.1 OPERAÇÃO NUMÉRICA

As definições das operações numéricas aplicadas em matrizes são também utilizadas na álgebra de mapas.

ADIÇÃO DE MATRIZES

Seja $A = [a_{ij}]$ e $B = [b_{ij}]$ são matrizes de mesma dimensão, então a soma $A + B$ é a matriz obtida adicionando-se os elementos correspondentes das matrizes A e B , Equação 6.

$$A + B = [a_{ij} + b_{ij}], \quad (6)$$

onde a notação à direita significa que o elemento da i -ésima linha e j -ésima coluna da matriz $A + B$ é $a_{ij} + b_{ij}$.

MULTIPLICAÇÃO DE MATRIZES POR UM NÚMERO

Seja $A = [a_{ij}]$ é uma matriz e c é um número real, então cA é a matriz obtida pela multiplicação de cada elemento por A por c , Equação 7.

$$cA = [ca_{ij}]. \quad (7)$$

3.10.2 OPERAÇÃO LÓGICA

Estas funções utilizam operadores lógicos (booleanos) e permitem realizar cruzamentos entre dois ou mais planos de informação. A operação lógica do tipo A AND B retorna todos os elementos contidos na intersecção entre A e B; A NOT B retorna somente os elementos contidos exclusivamente em A; A OR B retorna todos elementos contidos tanto em A como em B; A XOR B retorna todos os elementos contido em A e B não incluídos na intersecção de A e B.

O apelo da abordagem Booleana é a sua simplicidade e fácil aplicação, com a combinação lógica de mapas em um SIG diretamente análoga ao tradicional método de sobreposição empregado nas mesas de luz. Porém, na prática pode não ser apropriado atribuir igual importância para cada um dos critérios combinados, as evidências precisam ser pesadas dependendo da sua relativa importância (Bonham-Carter, 1994).

Nas operações booleanas, o valor resultante em cada posição é uma função apenas do valor da posição correspondente nos geo-campos de entrada. Na saída de uma operação booleana, os valores do contra-domínio necessitam ser especificados em função dos valores dos campos de entrada.

A Figura 12 apresenta um programa que cruza dois geo-campos (Solos e Declividade) e calcula o mapa de Aptidão de Uso.

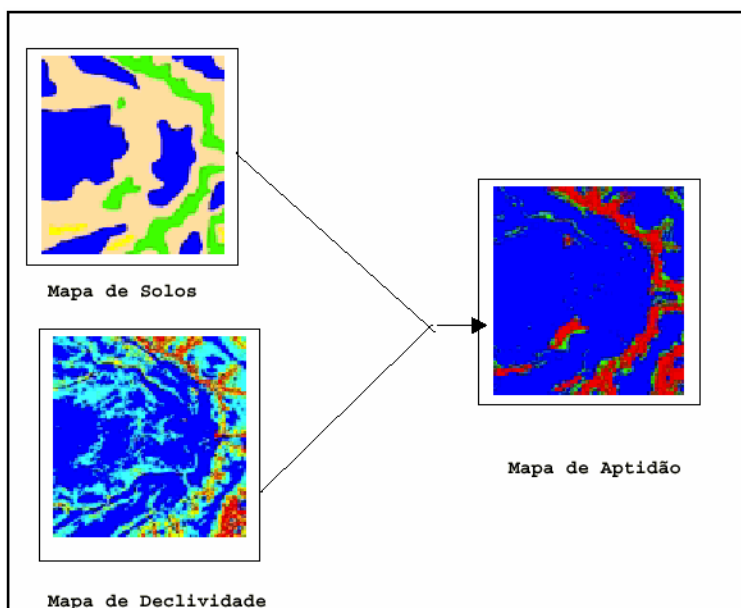


Figura 12 – Exemplo de uma operação lógica (CÂMARA, 2002).

3.10.3 INTERPOLAÇÃO

A interpolação é uma técnica utilizada para casos onde é realmente difícil calcular o valor da função ou ainda quando não é conhecida a expressão da função. Consiste em determinar uma função que assume valores conhecidos em certos pontos. A classe de funções escolhida para a interpolação é a priori arbitrária, e deve ser adequada às características pretendidas pela função.

A interpolação pode-se revelar inadequada se os nós de interpolação não forem escolhidos convenientemente. De um modo geral, o conjunto das funções interpoladoras é determinado por um número finito de parâmetros que deverá ser igual ao número de condições impostas, para que haja apenas uma solução.

Dentre as técnicas de interpolação, será apresentado dois métodos de interpolação: Inverso do quadrado da distância e Gradiente pelo inverso do quadrado da distância.

O Inverso do quadrado da distância é um método que calcula a média dos pontos circunvizinhos com maior peso aos pontos mais próximos. A função é calculada de forma que o valor desconhecido é determinado pela Equação 8:

$$Z = \frac{\sum_{i=1}^N \frac{Z_i}{d_i^2}}{\sum_{i=1}^N \frac{1}{d_i^2}} \quad (8)$$

onde Z é a variável desconhecida, Z_i é o valor da variável conhecida, d é a distância euclideana entre a variável conhecida e a desconhecida, e N é o número total de pontos conhecidos para a interpolação.

O método gradiente pelo inverso do quadrado da distância calcula cada ponto desconhecido baseado nos pontos conhecidos restantes. Os valores são calculados de forma que o valor desconhecido é determinado pela Equação 9:

$$Z = \frac{\sum_{i=1}^N \frac{Z_i + C_x(X - X_i) + C_y(Y - Y_i) + C_e(E - E_i)}{d_i^2}}{\sum_{i=1}^N \frac{1}{d_i^2}} \quad (9)$$

onde X e X_i são as abscissas dos pontos desconhecidos e dos pontos conhecidos, respectivamente, Y e Y_i são as ordenadas dos pontos desconhecidos e dos pontos conhecidos,

respectivamente, E e E_i são os valores dos pontos desconhecidos e dos pontos conhecidos, respectivamente, e C_x , C_y e C_e são os coeficientes de regressão para X , Y e E , respectivamente.

4 MATERIAIS E MÉTODOS

4.1 ESTRUTURA DO SISTEMA

Um sistema para representar e avaliar imagens interpoladas, o qual denominou-se UniSIG foi desenvolvido na linguagem C++, utilizando uma ferramenta RAD (Rapidly Application Development) C++ Builder 4 for Windows.

A base de cálculo do sistema está fundada em representar os pontos de imagem sob a forma matricial. No caso da interpolação, a matriz conterá os pontos previamente conhecidos e os demais pontos são estimados pela Equação 8.

A partir desta matriz, aplica-se as técnicas de geração de imagem em escala de cores e em escala de cinza. Os dados de imagem em formato BMP são lidos e mapeados sobre outra matriz. Cada célula possui um valor numérico que pode ser avaliado como a cor do pixel. Este tipo de imagem é denominado raster. Após a representação da imagem final, o sistema permite analisá-las por meio das técnicas da álgebra de mapas.

4.2 INTERFACE DO SISTEMA

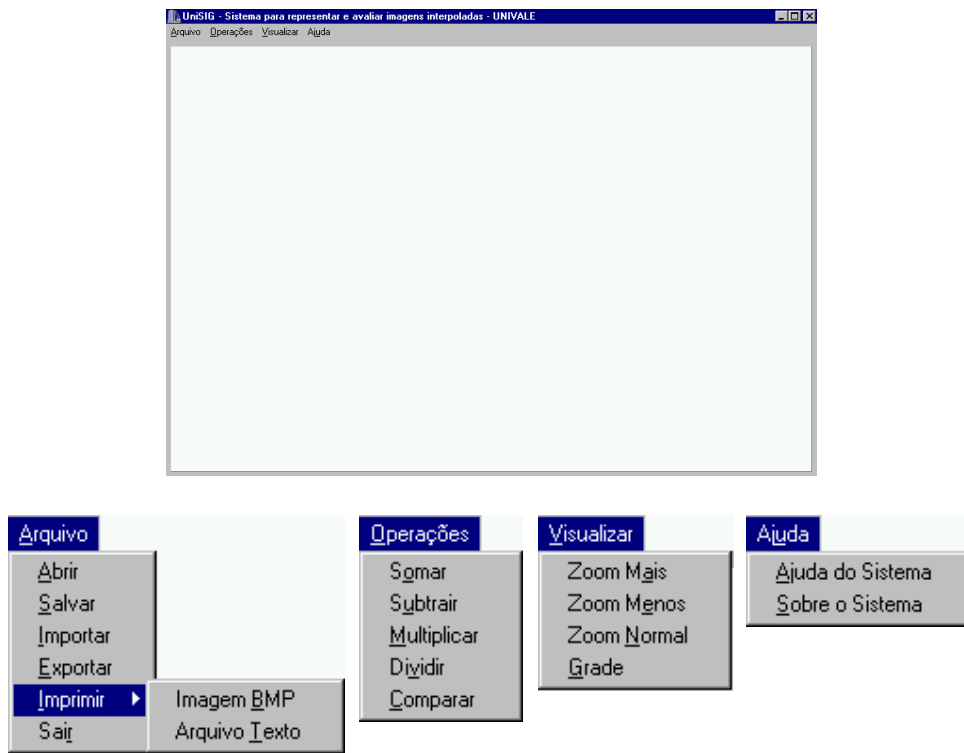


Figura 13 – Interface do sistema

A tela principal do sistema contém o 4 menus: Arquivo, Operação, Visualizar e Ajuda, Figura 13.

O menu Arquivo oferece as opções de: Abrir uma imagem, Salvar uma imagem, Importar um arquivo texto em imagem (Figura 14-a e 15-b), Exportar uma imagem em um arquivo texto e Imprimir Imagem BMP ou Arquivo Texto.

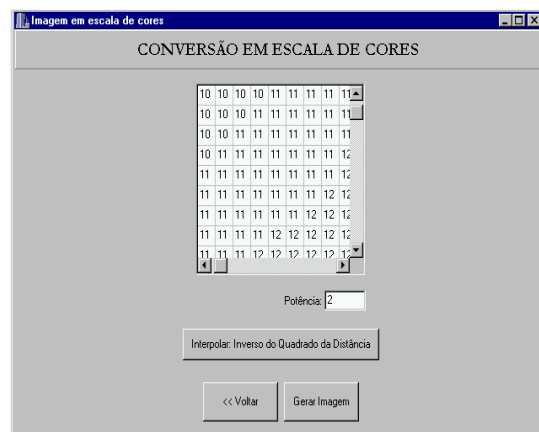
O menu Operações oferece as opções de: Somar duas imagens (Figura 14-c), Subtrair duas imagens, Multiplicar uma imagem por um valor real, Dividir uma imagem por um valor real e Comparar duas imagens, gerando a cor branca para os pontos com valores semelhantes e a cor preta para os pontos com valores diferentes.

O menu Visualizar oferece as opções de: Zoom mais (Figura 14-d), que aproxima duas vezes o tamanho da imagem, Zoom menos, que distancia duas vezes o tamanho da imagem, Zoom Normal, que volta a imagem ao tamanho original e Grade, que ativa e desativa grade na imagem.

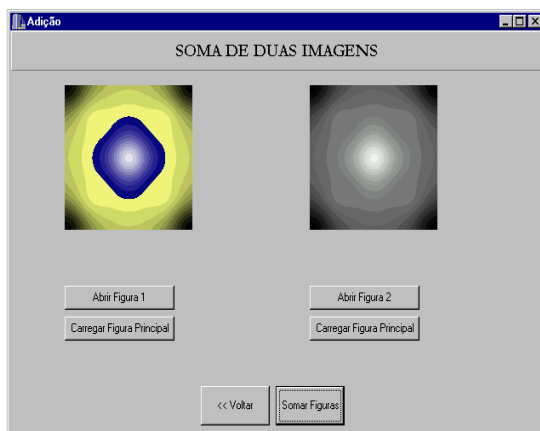
O menu Ajuda oferece as opções de: Ajuda do Sistema (Figura 14-e), que descreve todas as funções do sistema e Sobre o Sistema (Figura 14-f), que apresenta o sistema.



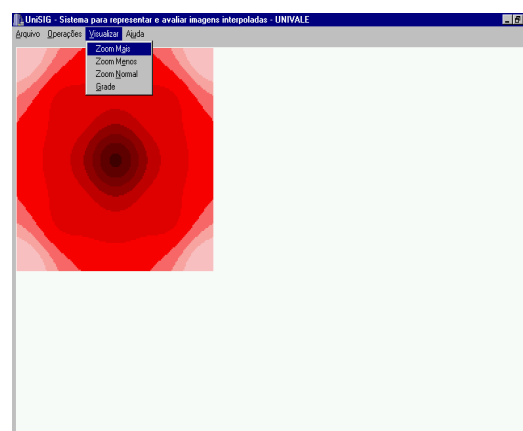
(a)



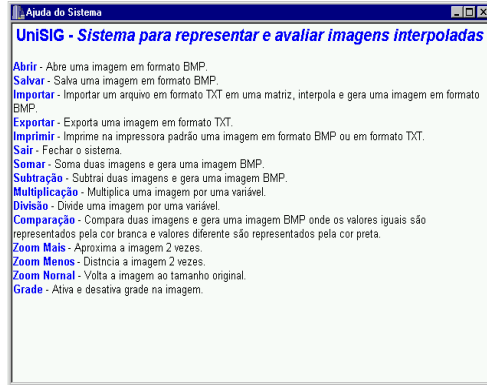
(b)



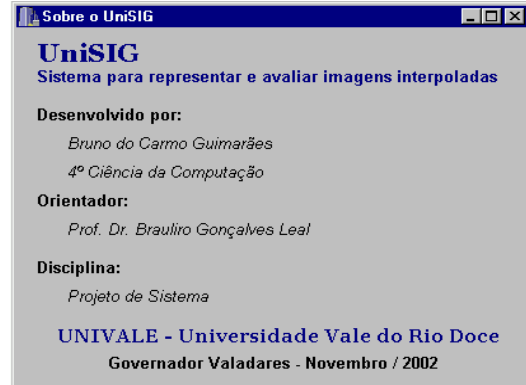
(c)



(d)



(e)



(f)

Figura 14 – Principais telas

4.3 PRINCIPAIS FUNÇÕES

4.3.1 OPERAÇÕES MATEMÁTICAS

Os resultados obtidos no sistema são apresentados sob a forma de imagem, denominada Imagem Resultado (IR), apresentada na Figura 15.

Uma IR é obtida a partir de uma matriz que teve seus elementos calculados por outras matrizes por meio das operações de adição, subtração, multiplicação, divisão e comparação.

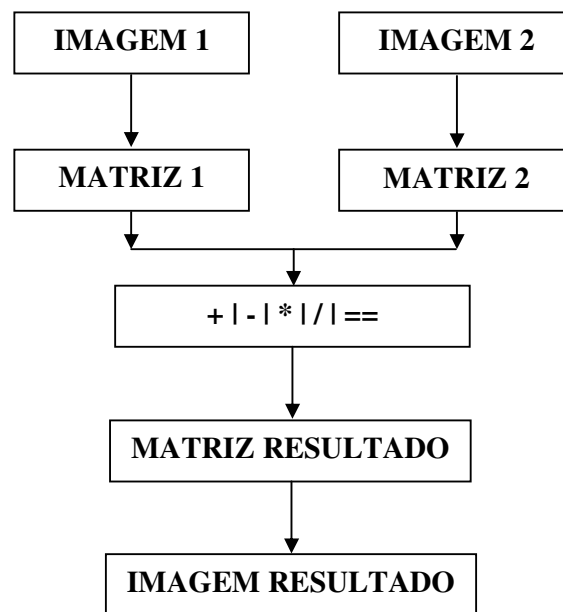


Figura 15 – Diagrama das Operações

4.3.2 INTERPOLAÇÃO – INVERSO DO QUADRADO DA DISTÂNCIA

O método de interpolação é o cerne do sistema. Nela obtém-se pontos através da distância entre os pontos conhecidos e os demais pontos. A implementação é feita utilizando os pontos conhecidos e suas respectivas coordenadas X e Y, os quais são armazenados em três vetores.

A partir destes vetores, aplica-se a Equação 8 para a obtenção dos valores desconhecidos, Anexo I. Assim, é gerada a matriz resultado, apresentada na Figura 16.



Figura 16- Diagrama de interpolação

4.3.3 GERAÇÃO DE IMAGEM EM ESCALA DE CORES E EM ESCALA DE CINZA

Para gerar imagens em escala de cores e em escala de cinza, deve-se segmentar os valores em classes e atribuir valores que representam tonalidades diferentes de cores, obtidas em função dos valores máximos e mínimos da matriz. Cada classe recebe um valor que representa uma faixa de cor, Anexo II e III.

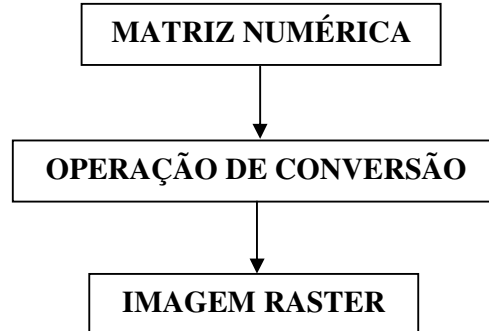


Figura 17- Diagrama de conversão

4.3.4 EXPORTAÇÃO E IMPORTAÇÃO

A exportação de uma imagem em formato texto foi implementada da seguinte forma: a matriz de imagem é percorrida, suas coordenadas X e Y, e seus valores, são armazenados em um arquivo texto com seus respectivos valores separados por espaço.

A importação é o inverso da exportação. Através de um arquivo texto, os valores são lidos e armazenados nas respectivas coordenadas da matriz, Anexo IV e V.

5 RESULTADOS E DISCUSSÃO

5.1 EXPORTANDO UMA IMAGEM E IMPORTANDO UM ARQUIVO TEXTO

As Figuras 18-a e 18-b apresentam a imagem e seu correspondente arquivo em disco, importado pelo UniSIG.

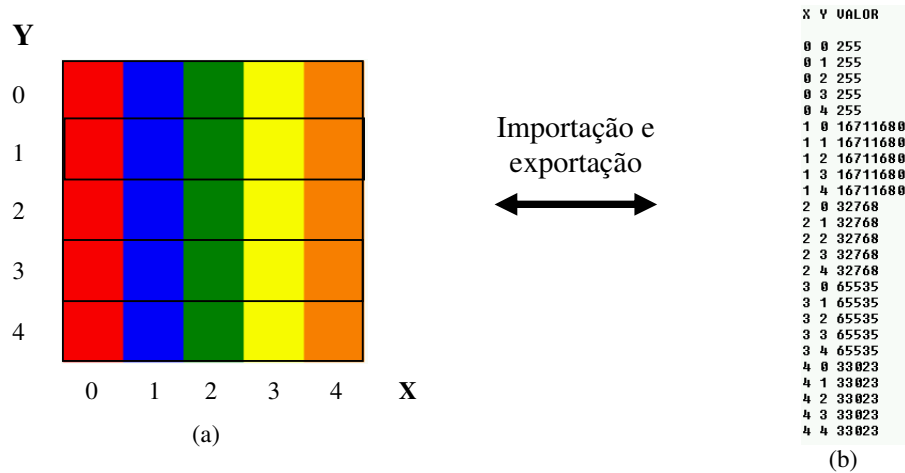


Figura 18 – Exemplo de importação e exportação: a) imagem e b) arquivo texto correspondente

Pode-se observar através da Figura 18, que o arquivo texto representa as coordenadas com seus respectivos valores numéricos correspondentes às cores.

5.2 INTERPOLANDO

A Figura 19 apresenta uma matriz com pontos conhecidos e pontos desconhecidos, e sua correspondente interpolada.

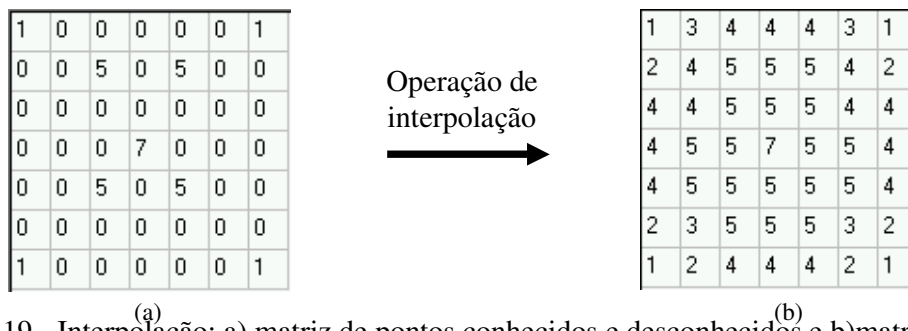


Figura 19 - Interpolação: a) matriz de pontos conhecidos e desconhecidos e b) matriz resultado

Pode-se observar através da Figura 19, que os valores iguais a zero (pontos desconhecidos) são sobrepostos pelo valor resultado da Equação 8 (pontos conhecidos).

5.3 GERANDO IMAGEM EM ESCALA DE CORES E EM ESCALA DE CINZA

A Figura 20 apresenta uma matriz numérica e sua correspondente imagem.



Figura 20 – Conversão de imagem: a) matriz e b) imagem gerada em escala de cinza

Pode-se observar através da Figura 20, que cada valor foi representado por uma cor. Os menores valores se aproximaram da cor preta e os maiores valores se aproximaram da cor branca.

5.4 SOMANDO DUAS IMAGENS

A Figura 21 apresenta a soma de duas matrizes de imagens.

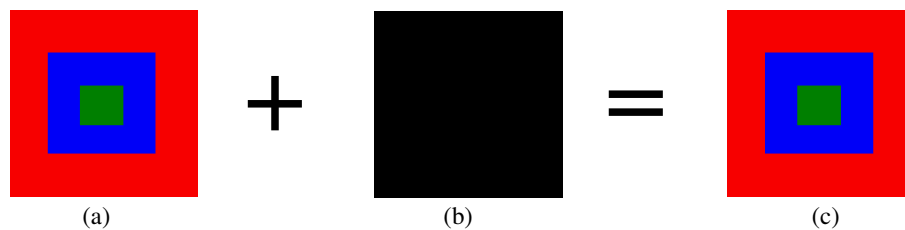


Figura 21 – Soma de imagens

Pode-se observar através da Figura 21, que a soma de uma imagem qualquer com uma imagem de cor preta resulta na mesma imagem, pois todos os elementos da matriz de cor preta possuem valor zero.

5.5 SUBTRAINDO DUAS IMAGENS

A Figura 22 apresenta a subtração de duas matrizes de imagens.

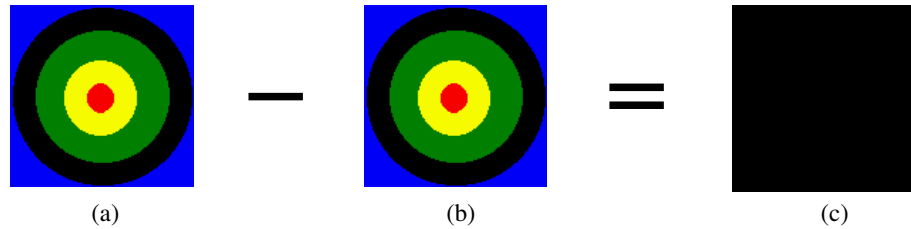


Figura 22 – Subtração de imagens

Pode-se observar através da Figura 22, que subtraindo duas figuras iguais, obtém-se uma matriz com todos seus elementos com valor zero, representada pela cor preta.

5.6 MULTIPLICANDO UMA IMAGEM POR UM VALOR REAL

A Figura 23 apresenta a multiplicação de uma imagem por um valor real.

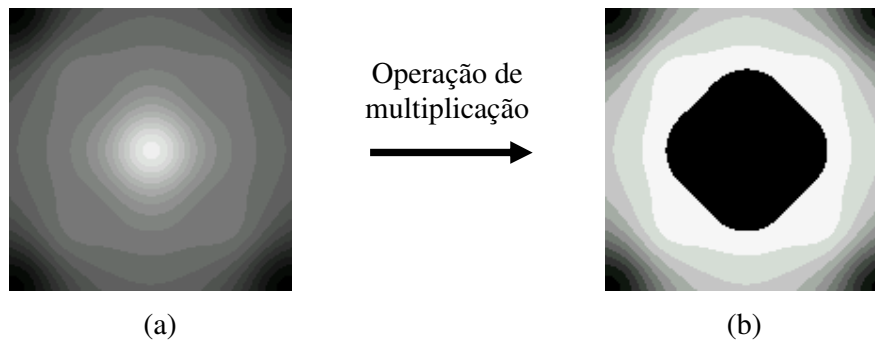


Figura 23 – Multiplicação de imagem por um valor real

Pode-se observar através da Figura 23, que multiplicando uma imagem por 2, a imagem resultante intensifica seus valores que são representados por cores mais claras. Observa-se também que o centro da imagem resultado ficou com cor preta, isso acontece quando o valor da cor ultrapassa o valor 16777215, que é o valor máximo da escala de cores, representada pela cor branca.

5.7 DIVIDINDO UMA IMAGEM POR UM VALOR REAL

A Figura 24 apresenta a divisão de uma imagem por um valor real.



Figura 24 – Divisão de imagem por um valor real

Pode-se observar através da Figura 24, que dividindo uma imagem por 2, a imagem resultante diminui seus valores que são representados por cores mais escuras.

5.8 COMPARANDO DUAS IMAGENS

A Figura 25 apresenta a comparação de duas imagens.



Figura 25 – Comparação de duas imagens

Pode-se observar através da Figura 25, que comparando duas imagens, os valores semelhantes são representados, na imagem resultado, pela cor branca, e as cores diferentes pela cor preta.

6 CONCLUSÃO

Foi desenvolvido um software na linguagem de programação C++ utilizando a ferramenta C++ Builder 4. O mesmo foi submetido a fases de desenvolvimento, testes até chegar em um sistema refinado capaz de representa e analisar imagens interpoladas, utilizando as técnicas de conversão de imagens, interpolação e álgebra de mapas.

A utilização da ferramenta C++ Builder 4 torna ainda mais adequado a implementação deste tipo de sistema, por ser uma ferramenta robusta e bastante simples de ser utilizada.

A criação de um sistema que representa e avalia imagens interpoladas é o princípio dos sistemas de informações geográficas.

O grande valor deste projeto foi o conhecimento obtido através de todos os obstáculos e desafios superados de forma bem sucedida.

O sistema apresentado é uma pequena parte de um SIG, porém, o conhecimento desta tecnologia abre caminhos para a implementação de sistemas capazes de representar e analisar imagens raster mais sofisticados, bem como imagens vetoriais, que envolvem uma tecnologia mais complexa.

7 REFERÊNCIAS BIBLIOGRÁFICAS

Barroso, Leônidas Conceição; Barroso, Magali Maria de Araújo; Filho, Frederico Ferreira Campos; Carvalho, Márcio Luiz Buntede; Maia, Mírian Lourenço. *Cálculo Numérico (com aplicações)*. São Paulo: Harbra. 1987. 254p.

Bonham-Carter, G.F. *Geographic Information System for Geoscientists*. Terrytown. Pergamon/Elsevier Science, 1994.

Calijuri, Maria Lúcia; Röhm, Sergio Antonio. *Sistema de Informações Geográficas*. Viçosa: UFV, 1995. 34p.

Câmara, Gilberto; Davis, Clodoveu; Monteiro, Antônio Miguel Vieira. *Introdução à Ciência da Geoinformação*. São José dos Campos: INPE, 2002. 343p.

Cyranka, Lúcia F. Mendonça; Souza, Vânia Pinheiro de. *Orientações para Normalização de Trabalhos Acadêmicos*. Juiz de Fora: UFJF, 2000. 89p.

Deitel, H.M.; Deitel, P.J.. *C++ Como programar*. Porto Alegre: Bookman, 2001. 1098p.

Eastman, et al.1995 Eastaman, J.R.; Jin, W.; Kyem, P.A.K.; Toledano, J. Raster produceres for multi-criteria/multi-objective decesion. *Photogrammetric Engineering and Remote Sensing*. 1995. 547p.

Foley, James; Dam, Andries; Feiner, Steven; Hug. *Computer Graphics: Principles and Practice in C*. Pearson, 1995. 1200p.

Pressman, Roger S. *Engenharia de Software*. São Paulo: Makron Books, 1995. 1056p.

Price, David T.; McKenney, Daniel W.; Nalder, Ian A.; Hutchinson, Michael F.; Kesteven, Jennifer L.. *A comparison of two statistical methods for spatial interpolation of Canadian monthly mean climate data*. University of Alberta, Edmonton, Canada. 1999. 81-94p.

Nalder, Ian A.; Wein, Ross W.. *Spatial interpolation of climatic Normals - test of a new method in the Canadian boreal forest*. University of Alberta, Edmonton, Canada. 1998. 211-225p.

8 ANEXOS

ANEXO I – CÓDIGO DA FUNÇÃO INTERPOLAÇÃO

```
int k = 0, n=StrToInt(Edit1->Text);
double dist = 0, numerador = 0, denominador = 0, M[300][300];
double vet_x[13000], vet_y[13000], vet_val[13000];

for ( int i = 0 ; i < StringGrid1->ColCount ; i++ )
    for ( int j = 0 ; j < StringGrid1->RowCount ; j++ )
        M[i][j]=StrToInt(StringGrid1->Cells[i][j]);

for ( int i = 0 ; i < StringGrid1->ColCount ; i++ )
    for ( int j = 0 ; j < StringGrid1->RowCount ; j++ )
        if ( M[i][j] != 0 )
            {
                vet_x[k] = i;
                vet_y[k] = j;
                vet_val[k] = M[i][j];
                k++;
            }
if ( k > 0 )
    {
        for ( int i = 0 ; i < StringGrid1->ColCount ; i++ )
            for ( int j = 0 ; j < StringGrid1->RowCount ; j++ )
                {
                    if ( M[i][j] == 0 )
                        {
                            for ( int a = 0 ; a < k ; a++ )
                                {
                                    dist = sqrt ( pow ( ( vet_x[a] - i ) , 2 ) + pow ( ( vet_y[a] - j ) , 2 ) ) );
                                    numerador += ( vet_val[a] / ( pow ( dist , n ) ) );
                                    denominador += ( 1 / ( pow ( dist , n ) ) );
                                }
                            M[i][j] = int((numerador / denominador) + 0.5);
                            numerador = 0;
                            denominador = 0;
                        }
                }
    }
for ( int i = 0 ; i < StringGrid1->ColCount ; i++ )
    for ( int j = 0 ; j < StringGrid1->RowCount ; j++ )
        StringGrid1->Cells[i][j] = FloatToStr(M[i][j]);
```

ANEXO II – CÓDIGO DA FUNÇÃO GERAÇÃO DE IMAGEM EM ESCALA DE CINZA

```
Form1->Image3->Height=StringGrid1->RowCount;
Form1->Image3->Width=StringGrid1->ColCount;

//Varre toda a matriz a procura do maior e menor valor
Form1->DrawGrid1->Visible = False;
double maior=StrToInt(StringGrid1->Cells[0][0]),
menor=StrToInt(StringGrid1->Cells[0][0]);
int i, j;
for (i=0; i <StringGrid1->ColCount; i++)
    for (j=0; j <StringGrid1->RowCount; j++)
        {
            if (StrToInt64(StringGrid1->Cells[i][j]) > maior)
                maior = StrToInt64(StringGrid1->Cells[i][j]);
```

```

        if (StrToInt64(StringGrid1->Cells[i][j]) < menor)
            menor = StrToInt64(StringGrid1->Cells[i][j]);
    }
//Calcula as Classes
    double classe = ((maior - menor)/10);

//gera os intervalos em escala de cinza
    for (int i=0; i<StringGrid1->ColCount; i++)
        for (int j=0; j<StringGrid1->RowCount; j++)
            {
                double teste=StrToInt(StringGrid1->Cells[i][j]);
                if (StrToInt(StringGrid1->Cells[i][j]) < classe)
                    Form1->Image3->Canvas->Pixels[i][j] = 16777215;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 2 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*9;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 3 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*8;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 4 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*7;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 5 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*6;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 6 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*5;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 7 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*4;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 8 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*3;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 9 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825*2;
                else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 10 + menor)
                    Form1->Image3->Canvas->Pixels[i][j] = 1644825;
                else
                    Form1->Image3->Canvas->Pixels[i][j] = 0;
            }
    Form1->imagem = true;
    Form8->Close();

```

ANEXO III – CÓDIGO DA FUNÇÃO GERAÇÃO DE IMAGEM EM ESCALA DE CORES

```

    Form1->Image3->Height=StringGrid1->RowCount;
    Form1->Image3->Width=StringGrid1->ColCount;

//Varre toda a matriz a procura do maior e menor valor
    Form1->DrawGrid1->Visible = False;
    double maior=StrToInt(StringGrid1->Cells[0][0]),
        menor=StrToInt(StringGrid1->Cells[0][0]);
    int i, j;
    for (i=0; i <StringGrid1->ColCount; i++)
        for (j=0; j <StringGrid1->RowCount; j++)
            {
                if (StrToInt64(StringGrid1->Cells[i][j]) > maior)
                    maior = StrToInt64(StringGrid1->Cells[i][j]);
                if (StrToInt64(StringGrid1->Cells[i][j]) < menor)
                    menor = StrToInt64(StringGrid1->Cells[i][j]);
            }
//Calcula as Classes
    double classe = ((maior - menor)/10);

```

```

//gera os intervalos em escala de cores
for (int i=0; i<StringGrid1->ColCount; i++)
for (int j=0; j<StringGrid1->RowCount; j++)
{
double teste=StrToInt(StringGrid1->Cells[i][j]);
if (StrToInt(StringGrid1->Cells[i][j]) < classe)
Form1->Image3->Canvas->Pixels[i][j] = 15197183;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 2 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 13026303;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 3 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 10856191;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 4 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 7039487;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 5 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 255;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 6 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 231;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 7 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 198;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 8 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 148;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 9 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 123;
else if (StrToInt(StringGrid1->Cells[i][j]) < classe * 10 + menor)
Form1->Image3->Canvas->Pixels[i][j] = 99;
else
Form1->Image3->Canvas->Pixels[i][j] = 66;
}

Form1->imagem = true;
Form1->Close();

```

ANEXO IV – CÓDIGO DA FUNÇÃO EXPORTAÇÃO DE UMA IMAGEM EM FORMATO TEXTO

```

if (imagem=true)
{
SaveDialog1->InitialDir = "C:\\Escola\\4ºano-Superior\\Projeto de Sistemas\\UniSIG\\Arquivos\\";
SaveDialog1->Filter = "Arquivo Texto (*.txt)*.txt";
SaveDialog1->Execute();
Memo1->Clear();
Image3->AutoSize=true;

for (int i=0; i<Image3->Width; i++)
for (int j=0; j<Image3->Height; j++)
Memo1->Lines->Add(AnsiString("") + i + " " + j + " " + Image3->Canvas->Pixels[i][j]);

Image3->AutoSize=false;
ShowMessage("Imagem Salva como: " + SaveDialog1->FileName);
Memo1->Lines->SaveToFile(SaveDialog1->FileName);
}
else
ShowMessage("Imagem não carregada - Operação Cancelada!");

```

ANEXO V – CÓDIGO DA FUNÇÃO IMPORTAÇÃO DE UM TEXTO EM UMA IMAGEM

```

int x, y, valor, max_x, max_y;
OpenDialog1->InitialDir = "C:\\Escola\\4ºano-Superior \\";
OpenDialog1->Filter = "Arquivo Texto (*.txt)*.TXT";
OpenDialog1->Execute();
Memo1->Lines->LoadFromFile(OpenDialog1->FileName);
StringGrid1->RowCount = Memo1->Lines->Count;
StringGrid1->ColCount = Memo1->Lines->Count;
//Varre a primeira linha para atribuir os primeiros valores a max_x e max_y
sscanf(Memo1->Lines->Strings[1].c_str(), "%d %d %d", &x, &y, &valor);
max_x=x;
max_y=y;

for (int i=0; i<Memo1->Lines->Count; i++)
{
    sscanf(Memo1->Lines->Strings[i].c_str(), "%d %d %d", &x, &y, &valor);
    StringGrid1->Cells[x][y]=valor;
    if (max_x < x)
        max_x=x;
    if (max_y < y)
        max_y=y;
}

StringGrid1->ColCount = max_x+1;
StringGrid1->RowCount = max_y+1;

if (MessageBox(0, "Gerar imagem Colorida ou Cinza?", "Importar", MB_YESNO |
MB_ICONQUESTION) == IDYES)
{
    Form8->StringGrid1->RowCount = StringGrid1->RowCount;
    Form8->StringGrid1->ColCount = StringGrid1->ColCount;
    for (int i=0; i<StringGrid1->ColCount; i++)
        for (int j=0; j<StringGrid1->RowCount; j++)
            Form8->StringGrid1->Cells[i][j]=0;
    for (int i=0; i<StringGrid1->ColCount; i++)
        for (int j=0; j<StringGrid1->RowCount; j++)
            if (StringGrid1->Cells[i][j] > 0)
                Form8->StringGrid1->Cells[i][j]=StringGrid1->Cells[i][j];

    Form8->Show();
}
else
{
    Form11->StringGrid1->RowCount = StringGrid1->RowCount;
    Form11->StringGrid1->ColCount = StringGrid1->ColCount;
    for (int i=0; i<StringGrid1->ColCount; i++)
        for (int j=0; j<StringGrid1->RowCount; j++)
            Form11->StringGrid1->Cells[i][j]=0;
    for (int i=0; i<StringGrid1->ColCount; i++)
        for (int j=0; j<StringGrid1->RowCount; j++)
            if (StringGrid1->Cells[i][j] > 0)
                Form11->StringGrid1->Cells[i][j]=StringGrid1->Cells[i][j];

    Form11->Show();
}

```