UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

EDUARDO ANTONIO SPERANZA

Métodos e ferramentas de interpolação para geração de mapas georreferenciados para tomada de decisão em Irrigação de Precisão

SÃO CARLOS – SP
2008

EDUARDO ANTONIO SPERANZA

Métodos e ferramentas de interpolação para geração de mapas georreferenciados para tomada de decisão em Irrigação de Precisão

Dissertação apresentada ao Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos, Universidade de São Paulo, para obtenção do título de mestre em Engenharia Elétrica.

Área de Concentração: Processamento de Sinais e Instrumentação.

Orientador: Dr. Evandro Luis Linhari Rodrigues

SÃO CARLOS – SP
2008

Dedicatória

À minha noiva Tatiane, pela paciência e compreensão demonstradas durante o desenvolvimento desse trabalho.

À minha família, especialmente aos meus pais, Toninho e Maria, pela formação educacional e cultural durante toda a vida.

Agradecimentos

Ao Dr. André Torre Neto e colegas da Embrapa Instrumentação Agropecuária, pela colaboração e aprendizado no desenvolvimento do projeto Irrigap.

Ao Dr. Evandro, por acreditar no trabalho e se propor a orientá-lo, mesmo que realizado à distância em grande parte do tempo.

A Deus, por me acompanhar nessa caminhada durante todos esses anos.

Resumo

SPERANZA, E.A. **Métodos e ferramentas de interpolação para geração de mapas georreferenciados para tomada de decisão em Irrigação de Precisão.** 2008. 115 f. Dissertação de Mestrado – Escola de Engenharia de São Carlos, Departamento de Engenharia Elétrica, Universidade de São Paulo, São Carlos-SP, 2008.

A qualidade e a produtividade agrícolas estão altamente relacionadas com a disponibilidade de água, já que sua falta pode causar prejuízos significativos em períodos críticos. Por esse motivo, a irrigação torna-se uma maneira importante de manter a disponibilidade de água, com o objetivo de obtenção de altos índices de qualidade e produtividade agrícolas. Os mapas temáticos desempenham papel de fundamental importância no processo de tomada de decisão em agricultura de precisão. Para a geração de mapas desse tipo, estão disponíveis bibliotecas e interfaces de código livre, como a TerraLib, a SOMCode e a aRT, que possuem algoritmos capazes de realizar a interpolação de dados com boa precisão, além de agrupar dados em regiões com características semelhantes. No presente trabalho foram utilizados dados obtidos por uma grade de sensores de uma unidade piloto do sistema Irrigap, desenvolvida pela Embrapa Instrumentação Agropecuária, dados esses referentes a medidas horárias de umidade e temperatura do solo e temperatura do ar. Diferentes métodos de interpolação foram utilizados, com o intuito de verificar qual a melhor maneira de se gerar mapas que possibilitem a visualização de áreas com o mesmo comportamento com relação às variáveis de solo e climatológicas, as chamadas zonas de manejo agrícolas. Os resultados obtidos mostraram que é possível a identificação dessas zonas, desde que sejam utilizados os algoritmos corretos para cada situação. Além disso, foi observado que *softwares* para geração de mapas temáticos utilizando dados reais podem se tornar importantes para a tomada de decisão em agricultura de precisão e proporcionar benefícios como a economia de água e energia durante o período de irrigação.

Palavras-chave: irrigação de precisão, mapas temáticos, instrumentação agropecuária.

Abstract

SPERANZA, E.A**. Interpolation tools and methods for georreferenced maps generation for decision making in Precision Irrigation.** Thesis (Master) – Escola de Engenharia de São Carlos, Departamento de Engenharia Elétrica, Universidade de São Paulo, São Carlos-SP, 2008.

The agricultural quality and yield are highly related with the water readiness, because its privation can cause significant damages in critical periods. Due to that reason, irrigation becomes an important way of readiness water maintenance, with the goal of obtaining high level of agricultural quality and yield. The thematic maps have a very important role in the precision agriculture making decision. For that kind of maps generation, there are free code libraries and interfaces, like TerraLib, SOMCode and aRT, whit algorithms to accomplish data interpolation with reliable precision, and also to group data from areas with similar behavior. In the present work, data were obtained from sensor grid of a pilot unit of the Irrigap system, developed by Embrapa Instrumentação Agropecuária, referring to hourly measures of soil moisture and temperature and air temperature. Different interpolation methods were used, with the intention of verify the best way to generate maps to make possible the visualization of areas with the same behavior related to the soil and the climatological variables, know as management zones. Obtained results show that the identification of these zones is possible, since the correct algorithms are used for each situation. Besides that, was observed that the software developed to generate thematic maps using real data can be important for making decision in precision agriculture and to provide benefits like saving water and energy during the irrigation period.

Keywords: precision irrigation, thematic maps, agricultural instrumentation.

Lista de Figuras

FIGURA 3.1 – ARQUITETURA DA TERRALIB 19
FIGURA 3.2 – USO DA TERRALIB COMO CAMADA DE ACESSO AO BANCO DE DADOS 21
FIGURA 3.3 – MODELO CONCEITUAL DA TERRALIB 24
FIGURA 3.4 – ARMAZENAMENTO DE GEOMETRIAS DO TIPO POLÍGONO 26
FIGURA 3.5 – DIAGRAMA DE CLASSES DA SOMCODE 32
FIGURA 3.6 – DIAGRAMA DE CLASSES PARA LEITURA E ESCRITA DE DADOS 32
FIGURA 3.7 – INTEGRAÇÃO ENTRE R/TERRAVIEW/TERRALIB 35
FIGURA 4.1 – ARQUITETURA BÁSICA DO SISTEMA IRRIGAP 40
FIGURA 4.2 – ÁREA DELIMITADA DA FAZENDA MARINGÁ, EM GAVIÃO PEIXOTO-SP 46
FIGURA 4.3 – ÁREA DA UNIDADE PILOTO INSTALADA 46
FIGURA 5.1 – RELACIONAMENTO ENTRE AS CAMADAS DO SGMAP 59
FIGURA 5.2 – INTEGRAÇÃO ENTRE O SGMAP, BANCO DE DADOS E BIBLIOTECAS EXTERNAS 59
FIGURA 5.3 – INTERFACE PRINCIPAL DO SGMAP 61
FIGURA 5.4 – CONFIGURAÇÕES DOS ALGORITMOS DA TERRALIB 64
FIGURA 5.5 – EXEMPLO DE MAPA TEMÁTICO GERADO 66
FIGURA 5.6 – FORMULÁRIO EDITAR LEGENDA E AGRUPAMENTO 67
FIGURA 5.7 – LEGENDA GERADA PELO SGMAP 69
FIGURA 5.8 – PARÂMETROS DA REDE SOM 70
FIGURA 5.9 – EXEMPLO DE MAPA DE KOHONEN GERADO 72
FIGURA 5.10 - FORMAÇÃO DE AGRUPAMENTO PARA MAPA COM N VARIÁVEIS 73
FIGURA 5.11 – MAPA GERADO APÓS A FORMAÇÃO DO AGRUPAMENTO 74
FIGURA 5.12 – FORMULÁRIO REFERENTE A INTERFACE DE ANIMAÇÃO 75
FIGURA 6.1 - VIZINHO MAIS PRÓXIMO 79
FIGURA 6.2 - N VIZINHOS MAIS PRÓXIMOS 80
FIGURA 6.3 - N VIZINHOS MAIS PRÓXIMOS COM DISTÂNCIA 80
FIGURA 6.4 - AGRUPAMENTO PASSOS IGUAIS 81
FIGURA 6.5 - AGRUPAMENTO QUANTIL 81
FIGURA 6.6 - AGRUPAMENTO DESVIO PADRÃO 82
FIGURA 6.7 – SEQÜÊNCIA DE MAPAS DE UMIDADE DO SOLO COM AGRUPAMENTO QUANTIL 83
FIGURA 6.8 – SEQÜÊNCIA DE MAPAS DE UMIDADE DO SOLO COM AGRUPAMENTO DESVIO PADRÃO 84
FIGURA 6.9 – COMANDOS PARA INICIALIZAÇÃO DA ART 84
FIGURA 6.10 – COMANDOS PARA CARREGAR TEMAS E *LAYERS* 85
FIGURA 6.11 – COMANDOS A SEREM EXECUTADOS PARA A KRIGAGEM 85
FIGURA 6.12 – MAPA GERADO UTILIZANDO KRIGAGEM 86
FIGURA 6.13 – MAPA DE KOHONEN GERADO (15 X 15) 87
FIGURA 6.14 – MAPA GEORREFERENCIADO UTILIZANDO SOMCODE 88
FIGURA 6.15 – MAPA GERADO UTILIZANDO A SOMCODE COM DADOS DE LOCALIZAÇÃO 89
FIGURA 6.16 – MAPA DE KOHONEN GERADO (30 X 30) 90
FIGURA 6.17 – MAPA GEORREFERENCIADO UTILIZANDO SOMCODE 91
FIGURA 6.18 – MAPA DE KOHONEN GERADO (15 X 15) 92
FIGURA 6.19 – MAPA GEORREFERENCIADO UTILIZANDO UMIDADE DO SOLO E ALTITUDE 93
FIGURA A1 – MAPA DE UMIDADE DO SOLO COM O TRATAMENTO DO "EFEITO DE BORDA" 103
FIGURA A2 – MAPA DE UMIDADE DO SOLO SEM O TRATAMENTO DO "EFEITO DE BORDA" 104

Sumário

1. INTRODUÇÃO **1**

2. UTILIZAÇÃO DE MAPAS TEMÁTICOS EM AGRICULTURA DE PRECISÃO **5**

2.1 AGRICULTURA DE PRECISÃO 5
2.2 IRRIGAÇÃO DE PRECISÃO 9
2.3 SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS 10
2.4 MAPAS TEMÁTICOS 11
2.5 IMPORTÂNCIA DA UTILIZAÇÃO DE SISTEMAS DE INFORMAÇÕES GEOGRÁFICAS E MAPAS TEMÁTICOS EM AGRICULTURA DE PRECISÃO 13

3. METODOLOGIAS PARA GERAÇÃO DE MAPAS TEMÁTICOS **15**

3.1 PRINCIPAIS TIPOS DE INTERPOLADORES 17
3.1.1 Krigagem *17*
3.1.2 Inverso da Distância Elevado a uma Potência *18*
3.2 A BIBLIOTECA TERRALIB 18
3.2.1 Introdução *18*
3.2.2 Modelo Conceitual *21*
3.2.3 Modelo do banco de dados *23*
3.2.4 Modelo de Geometrias *24*
3.2.5 Atributos Descritivos *27*
3.2.6 Algoritmos de Interpolação Disponíveis na TerraLib *28*
3.3 A BIBLIOTECA SOMCODE 29
3.3.1 Os Mapas Auto-Organizáveis *30*
3.3.2 Implementação da SOMCode *31*
3.3.3 Características da SOMCode *33*
3.4 ART: A API R-TERRALIB 34
3.4.1 O software R *34*
3.4.2 A Integração Entre R, TerraLib e TerraView *35*
3.4.3 Dependências para Instalação da aRT *36*
3.4.4 Considerações da Integração Entre Estatística Espacial e SIG *37*

4. A PLATAFORMA IRRIGAP **39**

4.1 INTRODUÇÃO 39
4.2 ESPECIFICAÇÕES E CARACTERÍSTICAS DO IRRIGAP 39
4.2.1 Nós Sensores e Atuadores *41*
4.2.2 Estações de Campo *42*
4.2.3 Estação Base *43*
4.2.4 Kit de Instalação *44*
4.2.5 Unidade Piloto do Sistema Irrigap *45*

5. SGMAP – *SOFTWARE* PARA GERAÇÃO DE MAPAS PARA AGRICULTURA DE PRECISÃO **49**

5.1 INTRODUÇÃO 49
5.2 MATERIAIS E MÉTODOS 50
5.2.1 Paradigma de Orientação a Objetos *50*
5.2.2 Principais Conceitos de Programação Orientada a Objetos *51*
5.2.3 Vantagens da Programação Orientada a Objetos e Sua Utilização no Desenvolvimento do SGMAp *53*
5.2.4 A Linguagem C++ *54*
5.2.5 O Banco de Dados PostgreSQL *55*
5.2.6 Arquitetura do SGMAp e Integração com as bibliotecas TerraLib e SOMCode *58*
5.3 INTERFACE E FUNCIONALIDADES DO SGMAP 61
5.3.1 Considerações Iniciais *61*
5.3.2 Geração de Mapas Temáticos por Interpolação Utilizando a TerraLib *64*
5.3.3 Geração e Configuração de Legendas e Agrupamentos *67*
5.3.4 Exibição da Legenda *68*
5.3.5 Geração de Mapas Temáticos Interpolados utilizando a TerraLib em conjunto com a SOMCode *69*

5.3.6 Geração de Animações *74*

6. RESULTADOS E CONCLUSÕES **77**

6.1 INTRODUÇÃO 77
6.2 CONJUNTO DE DADOS UTILIZADO 77
6.3 RESULTADOS E COMPARAÇÃO DE MÉTODOS PARA GERAÇÃO DE MAPAS DE UMIDADE DO SOLO 79
6.4 RESULTADOS OBTIDOS UTILIZANDO AGRUPAMENTO PELA SOMCODE 87
6.5 CONCLUSÕES 93

7. REFERÊNCIAS BIBLIOGRÁFICAS **97**

APÊNDICE A – TRATAMENTO DO "EFEITO DE BORDA" **103**

APÊNDICE B – TRECHOS DE CÓDIGO FONTE EM C++ **105**

1. Introdução

As relações entre água, solo, planta e atmosfera são a base para a produção agrícola, onde seu rendimento e eficiência dependem do equilíbrio dessas relações.

A disponibilidade de água possui um papel determinante no que diz respeito à qualidade e produtividade agrícola, podendo causar prejuízos significativos em virtude da deficiência hídrica em períodos críticos.

Nesse contexto, a irrigação torna-se um método importante para manutenção da disponibilidade de água, com o intuito de se obterem altos índices de qualidade e produtividade agrícola.

Outros fatores também devem ser considerados para justificar o uso da irrigação. Dentre os recursos tecnológicos disponíveis, a técnica da irrigação, sempre que utilizada de forma racional, tanto nos aspectos técnicos quanto econômicos, pode contribuir de forma importante para o eficiente desempenho do agronegócio nacional.

Quando a irrigação é utilizada para suprir as demandas ou necessidades hídricas da planta, o risco da quebra de safra é minimizado, mesmo que ocorra falta de chuva. A redução desse risco é um fator atrativo importante para investimentos, fazendo com que a irrigação se torne um elemento ampliador da disponibilidade de produtos e facilitador de capitalização na agropecuária. Além disso, a presença apenas da precipitação pluviométrica durante quase todo o ciclo da planta não é uma garantia de ocorrência de produção final, pois pode faltar água em momentos críticos do ciclo vegetativo.

Sem a utilização da irrigação, os dispêndios em água, energia, insumos e mão-de-obra podem se transformar em prejuízo e não em receita. O estresse causado pela falta de água reduz sensivelmente a produção vegetal, inviabilizando-a em regiões onde a falta de água é constante e limitante para a atividade agrícola.

Outro fator favorável à irrigação é que algumas espécies de plantas, como frutas e legumes, sob o regime controlado de irrigação e fertilizantes, apresentam melhoria na qualidade do produto final. Nesse caso, as qualidades desejáveis para o consumo, como o tamanho e o teor de açúcar, podem ser determinadas pela irrigação.

A ampliação de safras anuais, passando a realizar o cultivo em diferentes épocas do ano com a possibilidade de colheitas em diversos períodos é mais um fator a ser considerado para a presença de controle de água na produção, melhorando a lucratividade com a colocação do produto no mercado em momentos de falta do mesmo. (MATSURA et al., 2005)

No que diz respeito à utilização nas lavouras, existem quatro métodos de irrigação mais comumente empregados: infiltração, feita por gravidade ou por sulcos, podendo ser superficial ou subterrânea; submersão ou inundação, com o fornecimento de água a porções de terrenos separadas por pequenos diques, podendo ser periódico ou permanente; aspersão ou chuva artificial, com o lançamento de água para o ar, para cima e caindo sob forma de chuva; e gotejamento, com bicos especiais (gotejadores) adaptados às extremidades dos canos de irrigação, que fixados à tubulação principal formam a rede de irrigação. (HERBARIO, 2008).

Diante desse contexto, esforços com a criação de novas metodologias e instrumentos vêm sendo realizados para melhorar o processo de irrigação em culturas que necessitam de água para manter a produtividade e qualidade, evitando o estresse hídrico em períodos de seca. O objetivo desses trabalhos é maximizar o uso eficiente da irrigação.

Os Mapas Temáticos[1] (CÂMARA, 1995) desempenham papel de fundamental importância no processo de tomada de decisão em agricultura de precisão. Especificamente para a irrigação de precisão, a identificação de áreas com diferentes comportamentos com relação à distribuição de água no solo é um dos principais fatores para tomada de decisão com relação à variação da taxa de aplicação de água.

[1] Mapas Temáticos: mapas que contêm regiões geográficas definidas por um ou mais polígonos, como os de uso do solo e aptidão agrícola de uma região.

A disponibilidade de sensores pontualmente distribuídos em campo, com a capacidade de obtenção de dados variáveis de umidade e temperatura em tempo real, possibilita a criação de mapas temáticos para visualização de dados combinados. Para tanto, técnicas de interpolação devem ser utilizadas para a geração de valores intermediários entre os pontos de coleta de dados reais. As coordenadas geográficas desses pontos devem ser conhecidas, e podem ser obtidas por meio de GPS[2], por exemplo. (TORRE-NETO et al., 2005)

Além disso, o armazenamento de dados em uma base espaço-temporal permite a criação de mapas em diferentes intervalos de tempo, possibilitando uma visualização seqüencial do comportamento do solo na linha do tempo (SPERANZA et al., 2007).

O intuito desse trabalho é comparar técnicas de geração de mapas temáticos para agricultura de precisão, utilizando-se de algoritmos presentes em bibliotecas de código livre e de eficiência comprovada. Para tanto, a biblioteca TerraLib desenvolvida pelo INPE (Instituto Nacional de Pesquisas Espaciais) (TERRALIB, 2008) foi largamente utilizada. A biblioteca disponibiliza, em suas últimas versões, técnicas simples para a interpolação de dados georreferenciados, como o inverso do quadrado da distância. No entanto, a TerraLib possibilita apenas a obtenção de mapas relacionando uma única variável.

Para possibilitar a obtenção de mapas utilizando *n* variáveis simultaneamente, foi utilizada juntamente com a TerraLib a biblioteca SOMCode, desenvolvida em um trabalho da Embrapa Tabuleiros Costeiros (CPATC) em parceria com o próprio INPE (SOMCODE, 2008). A SOMCode fornece suporte à Redes Neurais Artificiais, por meio da utilização de mapas auto-organizáveis, os chamados Mapas de Kohonen (KOHONEN, 2001). Desse modo, foi possível a obtenção de mapas temáticos combinando valores de diferentes variáveis em um mesmo intervalo de tempo.

[2] *Global Position System*, ou Sistema de Posicionamento Global: Sistema americano de posicionamento por satélite, utilizado para a determinação da posição de um receptor na superfície da Terra ou em órbita.

A krigagem é um método de regressão utilizado em geoestatística para aproximar ou interpolar dados (KRIGE, 1951). Para efeito de comparação, foram gerados mapas interpolados utilizando esse método, por meio do ambiente computacional estatístico R (AMBIENTE R, 2008). O pacote aRT (ART, 2008), desenvolvido para o ambiente R pela Universidade Federal do Paraná (UFPR), foi utilizado para integrar as funcionalidades do R e as classes da biblioteca TerraLib.

Os dados utilizados para a geração dos mapas foram obtidos por meio de sensores de uma unidade piloto do projeto Irrigap (TORRE-NETO et. al, 2007) (TORRE-NETO et. al, 2005), que constitui uma plataforma tecnológica para a Irrigação de Precisão, desenvolvida pela Embrapa Instrumentação Agropecuária (CNPDIA). Essa plataforma é capaz de fornecer dados pontuais de umidade e temperatura do solo e de temperatura do ar em diferentes intervalos de tempo.

O conteúdo desse trabalho está organizado em seis capítulos, onde esse primeiro contempla uma introdução com a apresentação dos métodos utilizados. O Capítulo 2 descreve a importância da utilização de Mapas Temáticos e Sistemas de Informações Geográficas em agricultura de precisão, no que diz respeito à tomada de decisão. O Capítulo 3 apresenta as metodologias utilizadas na obtenção de mapas, destacando o uso das bibliotecas TerraLib e SOMCode, bem como do pacote aRT e sua integração com a TerraLib. O Capítulo 4 descreve as especificações do projeto Irrigap, que teve sua unidade piloto utilizada como base na obtenção de dados para esse trabalho. O Capítulo 5 exibe as especificações do SGMAp (*Software* para Geração de Mapas para Agricultura de Precisão), desenvolvido no decorrer desse trabalho. O Capítulo 6 mostra os resultados obtidos com o SGMAp, comparações com outros métodos e conclusões do que foi desenvolvido e avaliado durante a execução do trabalho.

2. Utilização de Mapas Temáticos em Agricultura de Precisão

2.1 Agricultura de Precisão

A Agricultura de Precisão pode ser definida de diferentes maneiras, mas o conceito geral é sempre o mesmo. De uma maneira simplificada, a Agricultura de Precisão pode ser definida como uma estratégia ambientalmente amigável e holística, na qual os produtores rurais podem variar entradas utilizadas e métodos de cultivo - incluindo a aplicação de sementes, fertilizantes e água, seleção variada, plantação, lavoura, colheita – para encontrar variações de solo e condições de cultura no campo (SRINIVASAN, 2006). Algumas outras definições de Agricultura de Precisão são descritas a seguir:

• "Adaptação cuidadosa do solo e do manejo da cultura para ajuste às diferentes condições encontradas em cada campo." (JOHANNSEN, 1996)

• "Produtores utilizam as tecnologias GPS que envolvem satélites e sensores de solo, além de gerenciamento intensivo da informação para entender variações nas condições de campo. Essa informação é utilizada para aplicar fertilizantes e outros insumos de maneira mais precisa, prevendo com maior precisão a produtividade da cultura."(THE U.S. HOUSE COMMITEE ON AGRICULTURE, 2000)

• "Aplicações de tecnologias e princípios para gerenciar variabilidades espaciais e temporais associadas com todos os aspectos da produção agrícola." (PIERCE E NOWAK, 1999). Em outras palavras, "fazer o procedimento correto, no momento correto, no local correto, e da maneira correta".

Essas definições sugerem que devem existir pelo menos três elementos críticos para o

sucesso da Agricultura de Precisão: informação, tecnologia e manejo.

Apesar de possuir um conceito geral bem definido, o termo Agricultura de Precisão é ainda muito recente, e tem gerado muito mal-entendido por parte dos muitos usuários dessa tecnologia no Brasil. Há alguns anos, quando surgiu, a Agricultura de Precisão trazia a idéia de uniformidade para a alta produtividade das lavouras, possibilitando a aplicação de insumos a taxas variadas em cada metro quadrado das mesmas. Nessa época, muito se falava da simplicidade eletrônica de todo o processo, com obtenção facilitada de dados que se tornariam insumos para aplicação de adubos e fertilizantes à taxa variada, sendo que isso seria o suficiente para nivelar a produtividade pelo seu máximo.

No entanto, com a disponibilidade dos primeiros dados a nível nacional, passou-se a entender um pouco melhor sobre a variabilidade que ocorria nas lavouras, e verificou-se que as correlações entre produtividade e os fatores individuais são baixas, tornando o desafio da Agricultura de Precisão mais interessante (MOLIN, 2003).

O manejo convencional praticado na agricultura de hoje é realizado na maioria dos casos pela média, ou seja, faz-se a amostragem do solo e um resultado vale para todo o talhão, ou até mesmo para toda a propriedade. Para verificarmos que esse tipo de tratamento pode deixar de considerar muitos aspectos, basta analisarmos um mapa de produtividade de uma lavoura para que seja identificada uma grande variabilidade. Portanto, o que é praticado hoje é apenas uma simplificação, devido à escassez de recursos técnicos para um maior nível de detalhamento (MOLIN, 2002).

Alguns autores relatam as vantagens de se utilizar a Agricultura de Precisão ao invés do sistema de manejo convencional. Para Gentil e Ferreira (1999), a adoção da mesma pode contribuir para reduzir o risco de atividade agrícola e dos custos de produção, facilitar a tomada de decisão rápida e correta, aumentar a produtividade da lavoura e diminuir o uso de defensivos, preservando o meio-ambiente. Para Batchelor et.al (1997), além dos benefícios já

citados, a Agricultura de Precisão pode melhorar o rendimento de colheita e lucros, e ao mesmo tempo reduzir custos com fertilizantes e praguicidas.

Nos últimos anos, a Agricultura de Precisão tem sido definida como um sistema de gestão ou de gerenciamento agrícola, constituindo-se de tecnologias e procedimentos utilizados para a otimização das lavouras e dos sistemas de produção, tendo como base o gerenciamento da variabilidade no espaço.

Como esse sistema de gerenciamento prescinde de muita informação, já se utiliza o termo “tecnologia da informação” para a Agricultura de Precisão, visto que a mesma se torna um agente gerador de bases de dados de alto valor gerencial (MOLIN, 2003).

A técnica de Agricultura de Precisão que tem se tornado bastante popular, e que se soma às informações já conhecidas da cultura, é a geração de mapas individuais para variáveis relacionadas ao solo. Para a obtenção desses mapas, é necessário certo investimento na coleta de amostra de dados, localizadas por GPS. A partir desse campo de coleta, surge a outra fase do processo, logo após o diagnóstico dos dados, que é a aplicação dos insumos na dose certa em cada local.

Uma tendência recente e muito provável de futuro é o gerenciamento agrícola por zonas de manejo, que faz com que o produtor passe a tratar a propriedade em áreas isoladas em todos os sentidos, desde a amostragem do solo até a contabilidade. Para a identificação dessas zonas, ferramentas como os mapas georreferenciados e as imagens aéreas tornam-se indispensáveis.

Para a realização de todas as fases do processo, têm-se notado o papel fundamental da mecânica e da eletrônica, com o fornecimento de máquinas agrícolas, sistemas eletrônicos e hidráulicos, além de programas computacionais. No Brasil, nesses últimos anos, tem-se verificado o crescimento de empresas que atuam no desenvolvimento e comercialização de soluções tecnológicas para Sistemas de Informações Geográficas dedicados à agricultura,

sensores de vários princípios de funcionamento para coleta de indicadores de solo, além de equipamentos dedicados para a automatização da regulagem de máquinas aplicadoras de insumos em geral. A iniciativa das empresas brasileiras em desenvolver produtos relacionados à Agricultura de Precisão se deve principalmente ao custo alto para importação desses produtos, tornando o desenvolvimento e comercialização a nível nacional como um fator de grande importância para torná-los acessíveis ao mercado brasileiro.

Além da disponibilidade de equipamentos e sistemas de informação para o gerenciamento agrícola, vale ressaltar a grande importância do usuário no contexto da Agricultura de Precisão. Basicamente, suas tarefas se resumem à coleta de dados úteis para chegar a um diagnóstico correto para tomada de decisão. No entanto, para que essas tarefas sejam executadas com o maior grau de precisão possível, são necessários empenho e conhecimento que poucos agricultores possuem para o gerenciamento de informações nesse nível de detalhamento. Com isso, a formação de pessoal capacitado com perfil adequado torna-se importante e necessária.

Finalmente, para que a adoção da Agricultura de Precisão aconteça em um ritmo mais acelerado, são necessários esforços na pesquisa, para melhorias no processo de entendimento dos fenômenos associados à variabilidade existente no campo e as formas de intervenção e convivência com a mesma; e a tentativa de redução do custo de adoção dessas tecnologias, que é alto devido à falta de produção de produtos nacionais. Na medida em que a tecnologia evolui, com técnicas mais adaptadas à realidade e acessíveis economicamente, a adoção dos agricultores vai aumentando de forma natural.

Desse modo, em propriedades onde se tem o objetivo de maximizar os lucros e a produtividade, além de minimizar os danos ambientais, a introdução da Agricultura de Precisão se tornará imprescindível.

Dentro do contexto de Agricultura de Precisão, uma das técnicas que surgiram com o

intuito de otimizar a aplicação de água nas lavouras foi a Irrigação de Precisão, descrita em maiores detalhes na próxima seção.

2.2 Irrigação de Precisão

A Irrigação de Precisão constitui uma combinação de tecnologias e práticas associadas de manejo que podem ajudar a superar os problemas de quantidade e tempo de irrigação a serem utilizados. (NATIONAL RESEARCH COUNCIL, 2005).

Deve ser ressaltado que a Irrigação de Precisão não implica em tecnologia de irrigação com custo alto. Pelo contrário, se refere a uma vasta gama de tecnologias e práticas de gerenciamento de água, possibilitando aos produtores com acesso limitado a esse recurso aplicar irrigação em suas culturas no tempo e quantidade exata para aumentar a produtividade da água. Ao contrário do que possa parecer, a Irrigação de Precisão pode ser praticada em conjunto com tecnologias convencionais existentes.

A adoção da Irrigação de Precisão pelas propriedades rurais pode ser sustentada pelo conceito de que, a aplicação de água sem se levar em conta variabilidade espacial do solo, pode fazer com que algumas áreas recebam mais do que a quantidade necessária a ser aplicada no momento, enquanto que outras podem não receber a quantidade suficiente para manter o equilíbrio da cultura. Quando se tem uma aplicação excessiva de água, pode-se estar contribuindo para o escoamento de águas superficiais ou pela lixiviação[3] de nutrientes e substâncias químicas para águas subterrâneas. Já uma aplicação ineficiente pode causar perdas econômicas com a redução da produtividade da cultura. (ESMAT et.al, 2002).

No entanto, a grande vantagem da Irrigação de Precisão, quando comparada com sistemas de irrigação convencionais, sejam estes automatizados ou não, está na possibilidade

[3] Lixiviação: Processo físico de lavagem das rochas e solos pelas águas das fortes chuvas, decompondo as rochas e carregando os sedimentos para outras áreas, extraindo, dessa forma, nutrientes e tornando o solo mais pobre.

de se evitar o desperdício de água e energia elétrica, pois a aplicação de água deverá ocorrer apenas quando necessária e de maneira diferenciada em cada zona de manejo definida na propriedade. Assim, além de manter a cultura sempre no estado ideal quanto à necessidade hídrica, evitando o desgaste prematuro da lavoura e a perda de produtividade, a Irrigação de Precisão pode representar outro ganho de alto valor econômico para o produtor e ambiental para a sociedade, que é a redução da utilização de água e energia.

Os sistemas de Irrigação de Precisão possibilitam também a aplicação espacialmente diferenciada de fertilizantes por meio da água aplicada na irrigação, constituindo os chamados sistemas de Fertirrigação de Precisão. O uso da Fertirrigação de Precisão pelo produtor proporciona a economia de fertilizantes e mão-de-obra, além de maior eficiência na aplicação dos mesmos, tendo como conseqüência o aumento da produtividade.

Como a Irrigação de Precisão é uma das tecnologias presentes no contexto de Agricultura de Precisão, a sua adoção por propriedades que tem o objetivo de maximizar os lucros e a produtividade diminuindo os impactos ambientais se torna imprescindível.

2.3 Sistemas de Informações Geográficas

Um Sistema de Informações Geográficas (SIG) é um sistema de informação baseado em computador que permite capturar, modelar, manipular, recuperar, consultar, analisar e apresentar dados geograficamente georreferenciados (HARA, 1997). A tecnologia do SIG pode trazer enormes benefícios devido à sua capacidade de manipular a informação espacial de forma rápida, precisa e sofisticada.

O aspecto mais fundamental dos dados tratados por um SIG é a dupla natureza da informação: um dado espacial ou geográfico possui uma localização, expressa como coordenadas de um mapa, e atributos descritivos, que podem ser representados em bancos de

dados convencionais.

2.4 Mapas Temáticos

Os Mapas Temáticos exibem padrões espaciais de um tema ou de um conjunto de atributos. Ao contrário dos mapas de referência, que exibem muitas características geográficas, como florestas, estradas e limites de área, os Mapas Temáticos enfatizam a variação espacial de algumas distribuições geográficas. Essas distribuições podem ser fenômenos físicos, como variações climáticas, densidade de população, ou assuntos relacionados à saúde. Para esse tipo de mapa, a localização é algo importante para prover uma base de referência de onde o fenômeno selecionado ocorreu.

Segundo Câmara (1995), os Mapas Temáticos representam dados espaciais do tipo geo-campo[4], que se caracterizam por conter regiões definidas por um ou mais polígonos, como mapas de aptidão agrícola de uma determinada região.

Existem duas maneiras de se armazenar os dados de um mapa temático: na forma vetorial (arcos) ou matricial (raster). Na primeira, os dados constituem uma topologia chamada de arco-nó-região, onde os arcos (limites entre regiões) se conectam entre si por meio de nós (ponto inicial e final); os arcos que circundam uma área definem uma região (polígono). Na segunda, a área correspondente ao mapa é dividida em células de tamanho fixo, onde cada célula possui um valor correspondente ao tema mais freqüente em sua localização espacial.

Existem vantagens e desvantagens de se utilizar uma ou outra maneira de armazenamento de dados, dependendo da aplicação. A Tabela 1, abaixo, faz uma breve

[4] Geo-campos: Variações espaciais contínuas, utilizadas para grandezas espacialmente distribuídas, tais como tipos de solo, topografia e teor de minerais. Na prática, são dados temáticos, imagens e modelos numéricos de terreno.

comparação entre elas:

Tabela 1 – Comparação entre modos de armazenamento de dados temáticos

Fonte: CÂMARA, 1995

Aspecto	Formato Vetorial	Formato Matricial
Relações espaciais entre objetos	Relacionamentos Topológicos Entre Objetos	Relacionamentos espaciais devem ser inferidos
Ligação com o banco de dados	Facilita a associação de atributos a elementos gráficos.	Associa atributos apenas a classes do mapa.
Análise, simulação e modelagem	Representação indireta de fenômenos contínuos, com álgebra de mapas limitada.	Representa melhor fenômeno com variação contínua no espaço. Simulação e modelagem mais fáceis.
Escalas de Trabalho	Adequado a grandes e pequenas escalas.	Mais adequado a pequenas escalas (grandes áreas).
Algoritmos	Problemas com erros geométricos.	Processamento mais rápido e eficiente.
Armazenamento	Por coordenadas	Por matrizes

2.5 Importância da Utilização de Sistemas de Informações Geográficas e Mapas Temáticos em Agricultura de Precisão

A utilização de Sistemas de Posicionamento Global (GPS) e de SIG no setor agrícola tem possibilitado a aplicação de diferentes estratégias de manejo, dentro de uma mesma área cultivada. Com o surgimento da Agricultura de Precisão, conceitos estão sendo revistos e criados, modificando a visão de como manejar o sistema agrícola.

Desse modo, a utilização de tecnologias de informações geográficas permite amparar tecnicamente decisões estratégicas e complexas em relação ao sistema de manejo adotado, seus efeitos ambientais e na produtividade das diferentes culturas (BOLFE E GOMES, 2005).

Dentre os mapas temáticos utilizados para tomada de decisão em Agricultura de Precisão, destacam-se os mapas de produtividade e lucratividade, fertilidade do solo, infestação e doenças, tamanho de fruto e reflectância, dentre outros.

Quando o contexto é a Irrigação de Precisão, destacam-se os mapas de umidade do solo, gerados por métodos de interpolação, com base em medidas diretas de sensores distribuídos em campo. Com base nesses mapas, o produtor pode determinar o momento adequado para realizar a irrigação em uma região, bem como a quantidade de água a ser aplicada na mesma.

3. Metodologias Para Geração de Mapas Temáticos

A elaboração de Mapas Temáticos pode ser realizada de diversas maneiras. Alguns trabalhos em Agricultura de Precisão vêm sendo realizados no sentido de encontrar os métodos mais adequados para geração desses mapas, de acordo com os dados disponíveis e o objetivo da visualização final.

O trabalho realizado por Farias et al. (2003) mostrou a variabilidade espacial e o potencial de produtividade em pomares de citros georreferenciados por meio de mapas temáticos, gerados pelas metodologias de geoestatística. Para tanto, foi utilizada uma área experimental localizada em uma propriedade rural no município de Luiz Antonio-SP, que compreendia uma área irrigada, com 3560 plantas, e outra não irrigada, com 2918 plantas. Por meio de digitalização, foram obtidas imagens das duas quadras, e, a partir dessas imagens, foi selecionado um total de 101 plantas que tiveram sua produtividade medida por meio da quantidade de frutos colhida pelos colhedores. Utilizando técnicas geoestatísticas de krigagem com cálculos realizados a partir desses dados, foram obtidos mapas de produtividade das duas quadras selecionadas. O resultado obtido mostrou que a média de produtividade para a área irrigada foi de 8 caixas de 40,8 kg por planta, enquanto que na área sem irrigação essa média caiu para 3,4 caixas.

Já a pesquisa de Coelho (2006) avaliou a influência conjunta da densidade amostral e do tipo de interpolador na elaboração de mapas temáticos de produtividade da soja. Nesse trabalho, foi realizada uma comparação entre os métodos de interpolação da krigagem, inverso do quadrado, inverso do quadrado da distância e polinomial, em cinco diferentes grades amostrais, utilizando-se amostras coletadas de uma área experimental da região oeste do Paraná. Essas amostras seguiram uma grade regular de 256 parcelas, analisadas de 1998 a 2002, sendo que a produtividade de cada uma foi associada ao seu centro geométrico. Os

resultados desse trabalho mostraram que quanto maior o número de amostras melhor é o detalhamento dos mapas. O interpolador inverso do quadrado da distância foi o que mais se aproximou do método referência (krigagem). Para que seja utilizado todo o potencial da krigagem, é necessário que se tenham mais amostras, para que um bom semivariograma seja construído.

O estudo de Schimalski et. al (2004) gerou mapas de produtividade em cultura de milho utilizando-se cinco interpoladores distintos (krigagem, vizinho natural, inverso da distância, radial e mínima curvatura), resultando na criação de classes de produtividade para serem visualizadas em mapas temáticos. Nesse estudo, foram utilizadas duas áreas experimentais, uma localizada no município de Porto União e outra em Frontin, ambas no estado do Paraná. Foram determinadas, em cada uma das áreas, unidades amostrais georreferenciadas para cálculo da produtividade baseado no comprimento das espigas e no peso dos grãos. O somatório do peso em cada unidade amostral foi extrapolado para produtividade por hectare levando-se em conta a área retangular da amostra. Com a obtenção desses dados, foi possível gerar uma grade retangular com pontos espacialmente distribuídos e com valores de produtividade interpolados, obtendo-se diferentes classes de produtividade. Os interpoladores da krigagem, mínima curvatura e radial, nessa ordem, se mostraram mais aptos para serem utilizados nessa aplicação. Os demais modelos também conseguiram representar a variabilidade, mas não conseguiram definir bem as classes.

O que deve ser notado tanto nos trabalhos citados acima, como em muitos outros, é o uso de métodos de interpolação para geração dos mapas temáticos. Segundo Sharma (1999), a interpolação é um procedimento de estimação de valor de um atributo em locais não amostrados a partir de pontos amostrados na mesma área ou região.

A interpolação espacial converte dados de observações pontuais em campos contínuos, produzindo padrões espaciais que podem ser comparados com outras tabelas espaciais

contínuas (COELHO, 2006). A base da interpolação segue o raciocínio de que valores do atributo, em média, tendem a ser similares em locais mais próximos do que em locais mais afastados.

A grande vantagem de se utilizar a interpolação espacial em Agricultura de Precisão é a de que se essa técnica possibilita a obtenção de mapas georreferenciados de boa qualidade, com a utilização de um número limitado de amostras de dados. Com isso, o custo final do sistema, no que diz respeito ao investimento financeiro em equipamentos eletrônicos torna-se reduzido, visto que as técnicas de interpolação possibilitam, por exemplo, a utilização de um número menor de sensores do que o que seria utilizado se as mesmas não fossem consideradas.

3.1 Principais Tipos de Interpoladores

3.1.1 Krigagem

A krigagem é uma metodologia para a interpolação de dados que tem como principal objetivo minimizar a variância estimada (CRESSIE, 1991). A krigagem tem a capacidade de produzir melhores estimativas em termos de interpolação, porque está embasada em duas premissas: não tendenciosidade do estimador e variância mínima das estimativas (VIEIRA, 2000). Deve-se ressaltar que a krigagem mostra uma ótima qualidade estatística explicita (COELHO, 2006).

Apesar de apresentar na maioria dos casos os melhores resultados, o algoritmo da krigagem exige, para a sua correta utilização e implementação computacional, um conhecimento bastante aprofundado de geoestatística.

3.1.2 Inverso da Distância Elevado a uma Potência

O método do inverso da distância estipula pesos aos pontos que são inversamente proporcionais à distância elevada a uma potência, que normalmente é quadrática (ISAAKS E SRIVASTAVA, 1989). Assim, o interpolador dá mais importância a valores mais próximos do ponto a ser estimado. Se o valor da potência for aumentado, aumenta a influência dos pontos mais próximos e diminui a dos pontos mais distantes.

O inverso da distância é um interpolador simples e que, na maioria dos casos, apresenta resultados satisfatórios.

3.2 A Biblioteca TerraLib

Esta seção descreve a biblioteca TerraLib em termos de bancos de dados geográficos, incluindo o modelo conceitual, de armazenamento de geometrias e dados descritivos, além dos mecanismos de manipulação do banco de dados em diferentes níveis de abstração. Atualmente, a TerraLib encontra-se na versão 3.2.0, disponível para *download* em (TERRALIB, 2008).

3.2.1 Introdução

A TerraLib é uma biblioteca oriunda de um projeto de *software* livre desenvolvido pelo INPE, e que permite o trabalho colaborativo entre a comunidade de desenvolvimento de

aplicações geográficas. A biblioteca atende requisitos que vão desde a prototipação rápida de novas técnicas até o desenvolvimento de aplicações colaborativas (CASANOVA et al., 2005).

O código-fonte da TerraLib constitui-se de um conjunto de classes escritas em C++ para a construção de aplicativos geográficos. Destina-se a servir como base para o desenvolvimento cooperativo na comunidade de usuários ou desenvolvedores de SIG.

A TerraLib fornece funções para a decodificação de dados geográficos, estruturas de dados espaço-temporais, algoritmos de análise espacial, além de propor um modelo para um banco de dados geográficos (CÂMARA et al., 2000). A Figura 3.1 mostra a arquitetura da biblioteca.

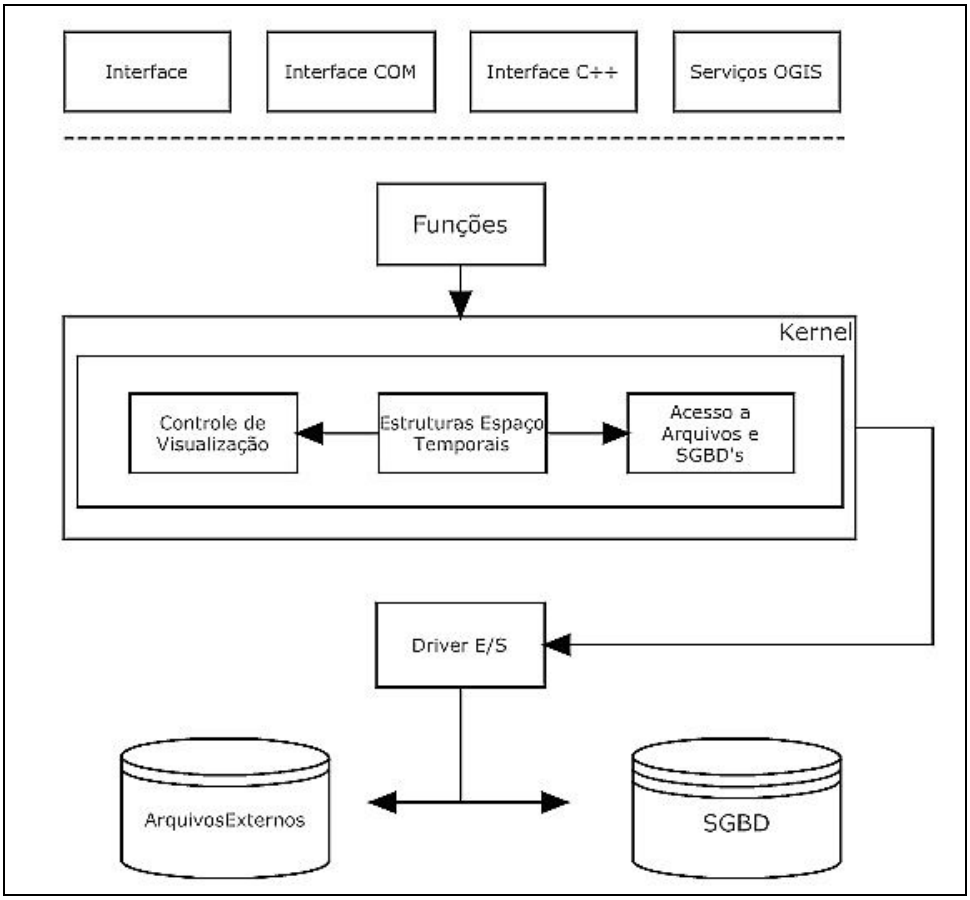

Figura 3.1 – Arquitetura da TerraLib

O módulo central chamado de *kernel*, é composto de estruturas de dados espaço-temporais, suporte a projeções cartográficas, operadores espaciais e uma interface para

armazenamento e recuperação de dados espaço-temporais em bancos de dados objeto-relacionais, além de mecanismos de controle de visualização. A interface de recuperação e armazenamento é implementada em um módulo composto de *drivers*, que também contém rotinas de decodificação de dados geográficos em formatos abertos e proprietários. As funções de análise espacial são implementadas utilizando as estruturas do *kernel*.

Uma característica interessante da TerraLib é que sobre os componentes da mesma podem ser construídas diferentes interfaces em diferentes ambientes de programação, como Java e C++, inclusive para a implementação de serviços do consórcio OpenGIS (OGC)[5].

A TerraLib possui capacidade de integração a Sistemas de Gerenciamento de Bancos de Dados Objeto-Relacionais (SGBD-OR) para armazenar dados geográficos. Essa integração permite o compartilhamento de grandes bases de dados, em ambientes corporativos, por aplicações customizadas para diferentes tipos de usuários. No modelo de arquitetura em camadas (DAVIS E CÂMARA, 2001), a TerraLib funciona como a camada de acesso entre o banco de dados e a aplicação final.

O TerraView (TERRAVIEW, 2008) é um exemplo de aplicativo geográfico construído sobre a TerraLib. Oferece à comunidade de desenvolvimento de SIG um visualizador de dados geográficos com recursos de consulta e análise dos mesmos, manipulando tanto dados vetoriais (pontos, linhas e polígonos) como matriciais (grades e imagens).

A Figura 3.2 ilustra como o TerraView utiliza a TerraLib como camada de acesso a um banco de dados sob o controle de um SGBD-OR.

[5] *Open GeoSpatial Consortium*, ou OpenGIS Consortium: organização voluntária internacional de padrões de consenso na área geoespacial.

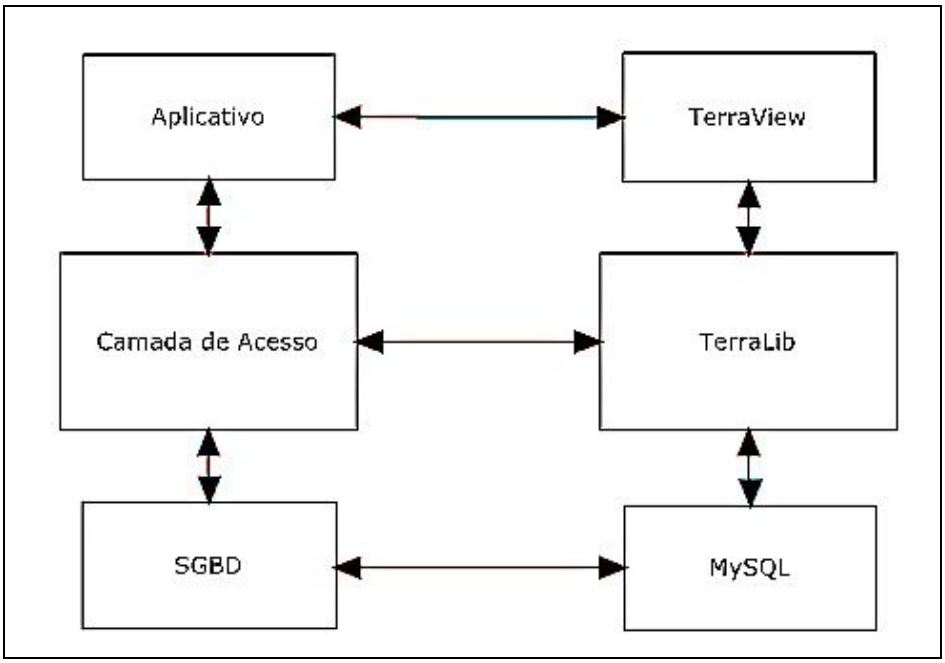

Figura 3.2 – Uso da TerraLib como camada de acesso ao Banco de Dados

No contexto de biblioteca de *software*, a TerraLib é compilada em um ambiente multiplataforma (Windows ou Linux), em diferentes compiladores C++, utilizando exaustivamente mecanismos como a STL[6] (Standard Template Library).

3.2.2 Modelo Conceitual

A TerraLib possui um modelo conceitual de banco de dados geográfico, sobre o qual são escritos os seus algoritmos de processamento de dados. As tabelas que formam o modelo conceitual da TerraLib são:

• Banco de Dados: Repositório de informações contendo os dados geográficos e o modelo de organização. O único requisito da TerraLib é que o SGBD escolhido possua a capacidade de armazenar campos binários longos, ou uma extensão própria capaz de criar

[6] Standard Template Library: biblioteca de funções, parte da biblioteca padrão C++, que descreve containers, iteradores e algoritmos.

tipos abstratos especiais.

• *Layer*: Representa uma estrutura de agregação de um conjunto de informações espaciais que são localizadas sobre uma região geográfica e compartilham um conjunto de atributos. As *layers* são inseridas no banco de dados por meio de importação de arquivos de dados geográficos em formatos como *shapefiles*, ASCII-SPRING, MID/MIF, GeoTiff, JPEG ou dbf. A biblioteca possui as rotinas de importação desses arquivos. *Layers* também podem ser geradas a partir de processamentos executados sobre outras *layers* já existentes no banco de dados.

• Representação: Modelo de representação da componente espacial dos dados de uma *layer*, podendo ser do tipo vetorial, onde a TerraLib distingue entre representações formadas por pontos, linhas, áreas, células ou redes; e do tipo matricial, suportando grades regulares multidimensionais.

• Projeção Cartográfica: Representa a referência geográfica da componente espacial dos dados geográficos, permitindo projetar a superfície terrestre em uma superfície plana.

• Tema: Define uma seleção de dados sobre uma *layer*, podendo essa seleção ser baseada em critérios a serem atendidos pelos atributos descritivos do dado ou da seleção espacial. Um tema pode definir também a forma de apresentação gráfica da componente espacial de seus objetos, bem como formas de agrupamento de dados de uma *layer*, gerando grupos com legendas que os caracterizam.

• Vista: Define uma visão particular de um usuário sobre o banco de dados, agregando os temas que serão processados ou visualizados em conjunto.

• Visual: Conjunto de características de apresentação de primitivas geométricas, como cores de preenchimento e contorno de polígonos, espessuras de contornos e linhas, cores de pontos, símbolos, estilos, etc.

• Legenda: Grupo de dados dentro de um tema, apresentados com o mesmo visual,

quando os dados do tema são agrupados de alguma maneira.

As tabelas que formam o modelo conceitual estão representadas tanto nas classes que compõem a biblioteca quanto em um conjunto de tabelas no banco de dados.

3.2.3 Modelo do banco de dados

Um banco de dados TerraLib é formado pelo armazenamento físico de dados geográficos e informações sobre a organização dos mesmos em um conjunto de tabelas, que podem ser dividas em dois tipos: meta-dados, com nome e formato pré-definidos, utilizadas para armazenar o modelo conceitual da biblioteca; e dados, utilizadas para armazenar os dados em si, tanto na componente espacial quanto na descritiva.

As tabelas de meta-dados são automaticamente criadas quando se cria um novo banco TerraLib. Para acessar bancos já existentes, as aplicações abrem conexões a eles, podendo manter conexões a mais de um banco de dados ao mesmo tempo. Essas tabelas armazenam os conceitos descritos na seção 3.2.2.

Para uma melhor compreensão das tabelas do modelo de meta-dados, é interessante observar na prática o conteúdo das mesmas após uma execução típica de operações:

- Criar um banco de dados;
- Importar um dado geográfico de um arquivo para uma *layer* do banco de dados;
- Criar uma vista;
- Criar um tema sobre a *layer* criada e inseri-lo na vista criada.

A Figura 3.3 mostra as tabelas que representam as entidades do modelo conceitual da TerraLib, bem como o seus relacionamentos após uma execução de operações. Nessa figura, apenas os campos mais importantes de cada tabela são exibidos.

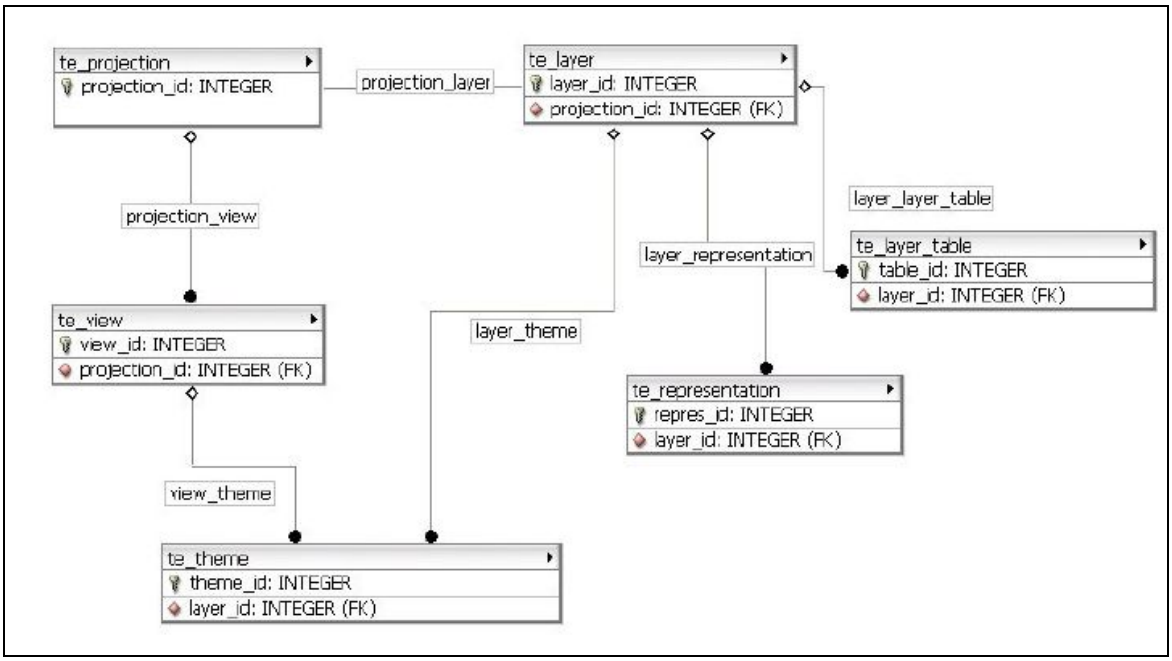

Figura 3.3 – Modelo Conceitual da TerraLib

3.2.4 Modelo de Geometrias

Os dados geográficos da TerraLib, como descrito na seção 3.2.2, são agregados em *layers*. A classe base da qual derivam todas as geometrias da TerraLib é chamada de TeGeometry, onde cada geometria possui uma identificação única, a referência ao seu menor retângulo envolvente e sua projeção, e a identificação do objeto geográfico que representa. A classe TeCoord2D representa as geometrias vetoriais da TerraLib a partir de coordenadas bi-dimensionais, que podem ser :

• Pontos: Instância única de TeCoord2D, representados pela classe TePoint;

• Linhas: Um ou mais segmentos da classe TeLine2D, como sendo um vetor de duas ou mais instâncias de TeCoord2D.

• Anéis: Linhas fechadas, representados pela classe TeLinearRing. Esta é

implementada como uma instância única de TeLine2D, que satisfaz a restrição de que a primeira coordenada seja igual à última.

• Polígonos: Delimitações de áreas, representados pela classe TePolygon, que podem conter nenhum, um ou mais buracos ou ilhas. São implementados como um vetor de TeLinearRing, onde o primeiro anel do vetor é sempre o anel externo e os outros anéis, se existirem, são buracos ou ilhas do anel externo.

As geometrias matriciais também possuem sua representação na TerraLib, onde são considerados como qualquer dado representado em uma estrutura de matriz com M linhas x N colunas. Exemplos desse tipo de dado são grades regulares com valores de determinada grandeza (como uma grade de altimetria) ou imagens de sensoriamento remoto. Essas geometrias são representadas na TerraLib pela classe TeRaster (CASANOVA et al., 2005), que é uma interface genérica de acesso aos elementos de um dado matricial.

A TerraLib implementa também uma estrutura de dados de espaços celulares. Os espaços celulares podem ser vistos de duas maneiras: como uma estrutura matricial generalizada, onde cada célula armazena mais que um valor de atributo; ou como um conjunto de polígonos que não se interceptam.

A vantagem dessa estrutura é a possibilidade de armazenar conjuntamente todas as informações necessárias para descrever um fenômeno espacial complexo, beneficiando os aspectos de visualização e interface. Desse modo, todas as informações podem ser representadas da mesma forma que objetos geográficos com representação vetorial.

A geometria da TerraLib que atende essa estrutura de dados está representada pela classe TeCell, que representa uma célula em um espaço celular materializado na classe TeCellSet. Cada célula possui uma identificação única e uma referência a sua posição dentro do espaço celular, onde podem ser associados diferentes atributos conforme o modelo dinâmico a ser construído.

O armazenamento físico das geometrias em um banco de dados leva em conta questões relativas à eficiência e recuperação, além da existência ou não de um tipo espacial no SGBD. Todas as tabelas do modelo da TerraLib que armazenam as geometrias possuem os seguintes campos:

• geom_id: do tipo inteiro, armazena a identificação única da geometria;

• object_id: do tipo texto, armazena a identificação única do objeto geográfico ao qual a geometria está relacionada;

• spatial_data: armazena o dado geométrico, e seu tipo depende do SGBD onde está armazenado o banco de dados. Para SGBD's com extensão espacial, é o tipo fornecido pela extensão. Para SGBD's sem extensão espacial, é um campo binário longo.

Quando o armazenamento ocorre em SGBD's sem extensão espacial, as tabelas de geometrias do tipo polígono e linha possuem campos adicionais para representar o mínimo retângulo envolvente da geometria, utilizado pelas rotinas de recuperação como indexadores espaciais do dado. Já para os SGBD's com extensão espacial, a coluna com o dado espacial é indexada espacialmente pelo mecanismo próprio da extensão.

A Figura 3.4 ilustra a diferença entre uma tabela representando uma entidade geométrica do tipo polígono criada em um banco sem extensão espacial em relação a um banco com extensão espacial.

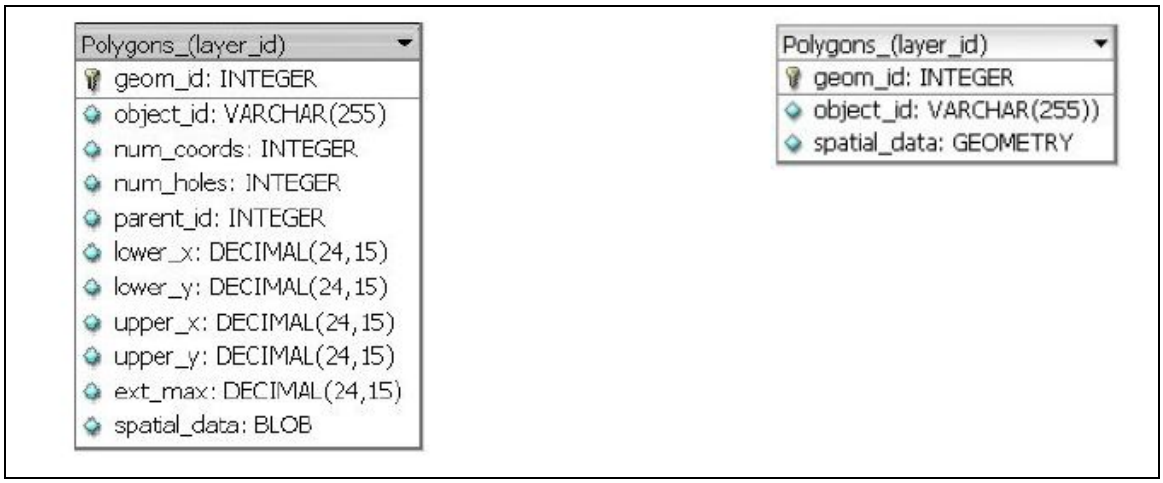

Figura 3.4 – Armazenamento de geometrias do tipo Polígono

No caso da representação por SGBD com extensão espacial, o tipo espacial fornecido pela extensão é representado pelo tipo GEOMETRY.

Existem algumas vantagens em se utilizar bancos de dados com extensão espacial, que adiciona tabelas geográficas ao SGBD. Além de possibilitar a capacidade de armazenamento e recuperação de dados espaciais segundo a especificação *Simple Features Specification*[7] (SFS), do consórcio internacional *Open GeoSpatial* (OGC), implementa diversas funcionalidades topológicas, possibilitando um desenvolvimento de SIG mais prático, e garantindo interoperabilidade com outros sistemas computacionais que também adotam o SFS.

3.2.5 Atributos Descritivos

Os atributos descritivos dos objetos constantes em uma *layer* são representados em tabelas relacionais, onde cada campo representa um atributo do objeto. Um dos campos deve conter a identificação do objeto, repetindo-se os valores na tabela de geometrias, para constituir a ligação entre os atributos descritivos e a própria geometria do objeto. Cada *layer* pode possuir uma ou mais tabelas de atributos, que ficam registradas em uma outra tabela do banco de dados.

As tabelas de atributos podem ser de chamadas de estáticas, quando não possuem nenhuma informação temporal e cada registro representa os atributos de um objeto diferente; e externas, quando não representam nenhum objeto definido de uma *layer*, mas podem possuir algum campo com valores coincidentes com valores de um campo de uma tabela estática de

[7] *Simple Features Specification*: Padrão OpenGIS que especifica o armazenamento digital de dados geográficos

uma *layer*.

A estrutura TeTable é responsável por materializar as tabelas de atributos de uma *layer* em memória. Essa classe sabe qual o nome, tipo, quais são os campos e pode conter também o conteúdo da tabela.

3.2.6 Algoritmos de Interpolação Disponíveis na TerraLib

A partir da versão 3.1.0, foram disponibilizados na biblioteca TerraLib alguns algoritmos para interpolação de dados. Esses algoritmos permitem a obtenção de mapas interpolados para geometrias matriciais (raster) ou vetoriais, sendo que os últimos são constituídos por estruturas celulares.

A estrutura que realiza a interpolação dos dados exige dois temas como entrada: um que representa os limites para a região a ser interpolada, normalmente um polígono; e o outro que representa os pontos contendo os valores que serão utilizados para a interpolação.

Os algoritmos de interpolação disponibilizados na TerraLib são:

• Vizinho Mais Próximo, que como o próprio nome já diz, atribui à célula de saída o valor do ponto mais próximo geograficamente a ela;

• Média dos Vizinhos Mais Próximos, que realiza uma média dos valores dos *k*-pontos geograficamente mais próximos da célula, onde o valor de *k* deve ser determinado.

• Inverso da Distância Elevado a uma Potência, que é um dos algoritmos de interpolação mais conhecidos na área de geoestatística e costuma fornecer resultados satisfatórios. O algoritmo atribui à célula de saída a soma dos valores associados aos *k*-pontos geograficamente mais próximos da célula, divididos pela distância de cada ponto em relação a

célula elevada a uma potencia *n*, onde *n* e *k* devem ser determinados.

• Média dos Elementos em uma Caixa: atribui à célula de saída a média dos pontos contidos em uma caixa de raio *r*, com ponto central localizado exatamente no ponto central da própria célula. O valor de *r* deve ser determinado.

• Inverso da Distância Elevado a uma Potência de Elementos em uma Caixa: é executado da mesma maneira que o Inverso da Distância Elevado a uma Potência, só que ao invés de serem utilizados os valores dos pontos geograficamente mais próximos à célula, são utilizados os valores dos pontos presentes em uma caixa quadrada de lado *r*, com ponto central localizado exatamente no ponto central da própria célula. O valor de *r* deve ser determinado.

Apesar de possuir poucas opções para geração de mapas interpolados, os algoritmos disponíveis para essa finalidade na TerraLib podem produzir resultados satisfatórios em determinadas situações. É o que acontece com os dados utilizados nesse trabalho, e que será ilustrado nos capítulos seguintes.

3.3 A Biblioteca SOMCode

A projeto da biblioteca SOMCode (SILVA et. al, 2006) está focado na implementação de algoritmos e encapsulamento de dados relacionados aos Mapas Auto-Organizáveis (SOM) de Kohonen (KOHONEN, 2001). Devido à necessidade de acesso indireto ou direto a bancos de dados geográficos para a aplicação dos Mapas Auto-Organizáveis nos problemas de análise espacial, a SOMCode possibilita a conexão com a biblioteca TerraLib.

Apesar de o algoritmo de aprendizado padrão dos SOM ser conceitualmente simples, sua implementação requer alguns cuidados. Em seu trabalho, Kohonen (2001) afirma que a

grande maioria das implementações não leva em consideração todos os detalhes envolvidos nos algoritmos SOM. Para minimizar os problemas, a equipe da Finland SOM desenvolveu dois pacotes que implementam todos os algoritmos básicos para SOM: O SOM PAK (KOHONEN, 1995), desenvolvido em C; e o SOM Toolbox (VESANTO, 1999), desenvolvido em Matlab. Ambos são de código livre e disponível.

Entretanto, após uma profunda análise desses pacotes, Silva et. al (2006) entenderam que muito tempo e esforço seriam gastos para conectar essas bibliotecas com a TerraLib, devido ao fato de a implementação das mesmas não terem sido realizadas na linguagem C++, não utilizando assim conceitos modernos de programação, o que torna sua manutenção e reuso mais difíceis. Apesar das vantagens de funcionalidade e confiabilidade que os dois pacotes disponibilizavam, Silva et. al (2006) decidiram desenvolver e implementar um novo algoritmo SOM em C++ usando o paradigma de orientação a objetos.

3.3.1 Os Mapas Auto-Organizáveis

O Mapa Auto-Organizável de Kohonen é uma rede neural artificial competitiva, estruturada em duas camadas (KOHONEN, 2001). A primeira representa os dados de entrada, e a segunda é uma grade de neurônios, normalmente bi-dimensional e interconectada, onde cada neurônio possui um vetor de código associado a ele, indicando a relação de vizinhança.

O processo padrão ou seqüencial de aprendizado pode ser dividido em três etapas. Na primeira, competitiva, cada padrão de entrada é apresentado para todos os neurônios da camada de saída. Os neurônios competem entre si, seguindo algum critério (normalmente a distância Euclidiana), para encontrar um único neurônio vencedor, também chamado de BMU (*Best Match Unit*). Na segunda, cooperativa, é definida a vizinhança desse neurônio. Na terceira, adaptativa, os vetores de código do neurônio vencedor e de sua vizinhança são

ajustados.

O processo de aprendizagem em lote difere da aprendizagem seqüencial no que diz respeito à forma de atualização dos vetores de código, ausência da taxa de aprendizagem e não obrigatoriedade de apresentação aleatória dos padrões. Nesse algoritmo, os vetores de código são atualizados ao final de cada época de treinamento. Em cada época, o conjunto de dados é particionado de acordo com as regiões de Voronoi[8] dos vetores de código do mapa neural, definido segundo critério de proximidade do vetor de código com o conjunto de dados.

Após o processo de aprendizado, os vetores de código devem aproximar, de alguma maneira, os dados de entrada. Adicionalmente, SOM preserva a estrutura topológica dos dados de entrada, onde padrões aproximados no conjunto de dados de entrada são associados com neurônios próximos na grade SOM.

A rede SOM pode variar levando-se em conta o algoritmo de aprendizado utilizado, a estrutura topológica da grade, a função de vizinhança utilizada, os parâmetros iniciais, etc.

3.3.2 Implementação da SOMCode

Após muitos estudos sobre qual padrão de desenvolvimento seria utilizado para as classes da SOMCode, o escolhido foi o padrão Abstract Factory[9]. O diagrama de classes final da SOMCode pode ser ilustrado na Figura 3.5, abaixo:

[8] Regiões de Voronoi: em Matemática, constitui um tipo especial de decomposição do espaço métrico, determinado por distancias para um determinado conjunto de objetos no espaço, isto é, por um conjunto discreto de pontos.

[9] Abstract Factory: Padrão de projeto de software que permite a criação de famílias de objetos relacionados ou dependentes, por meio de uma única interface e sem que a classe concreta seja especificada.

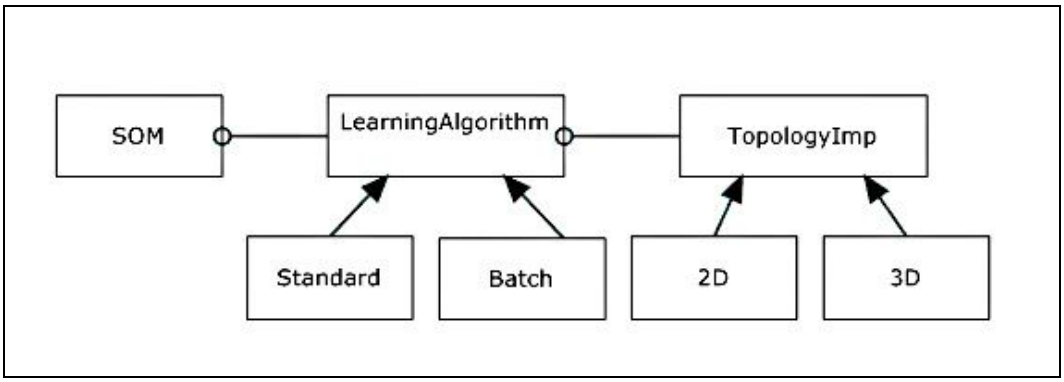

Figura 3.5 – Diagrama de Classes da SOMCode

Para cada classe de aprendizado, foi criada uma classe construtora, LearningFactory, StandardFactory e BatchFactory. Essas duas últimas devem implementar a função construtora da classe Factory.

A Figura 3.6 mostra o diagrama de classes final para a implementação dos métodos de leitura e escrita de dados, chamado SOMData, que alimentarão a rede neural.

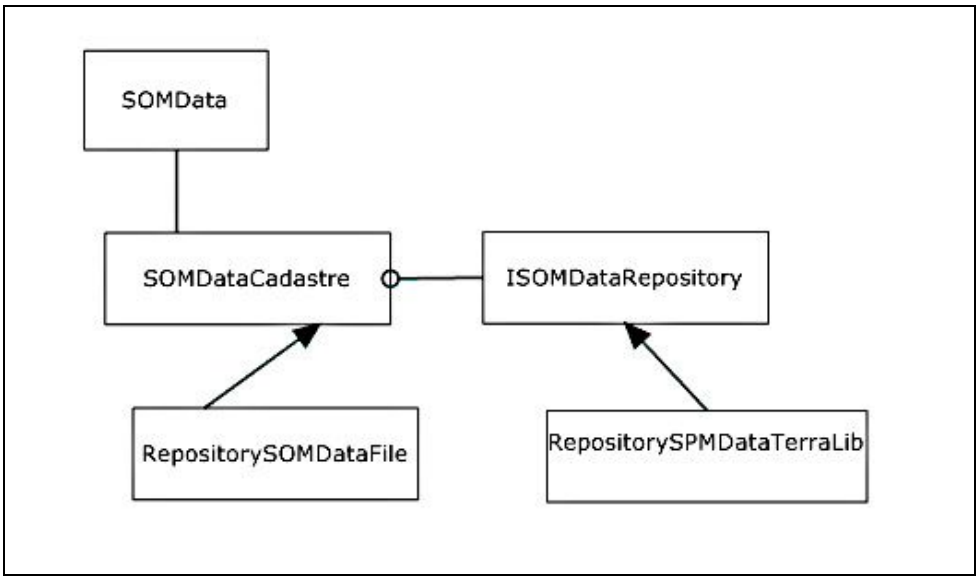

Figura 3.6 – Diagrama de Classes para leitura e escrita de dados

Uma classe concreta foi criada, denominada SOMDataCadastre, para um isolamento completo dos detalhes de armazenamento. Desse modo, a classe SOMData transfere todas as responsabilidades de gerenciamento de dados para a classe SOMDataCadastre. Como existem

muitas maneiras para armazenamento de dados, o padrão de desenvolvimento Strategy[10] foi utilizado para facilitar o processo de implementação de novos algoritmos de acesso. Assim, foi criada a classe abstrata ISOMDataRepository, e as classes concretas que implementam seus métodos de acesso, RepositorySOMDataFile, para arquivos, e RepositorySOMDataTerraLib, para o formato da TerraLib.

3.3.3 Características da SOMCode

Algumas características da SOMCode devem ser destacadas, com relação ao que foi implementado:

- Algoritmos de aprendizado seqüencial e em lote;
- Grades Bi-dimensionais;
- Funções de vizinhança gaussiana e de bolha;
- Lattice[11] Retangular e Hexagonal;
- Erros Métricos Topológicos e de Quantização;
- Inicialização Linear.

Dois bancos de dados de *benchmark* foram testados para avaliar a qualidade da SOMCode: Íris e Wine (BLAKE, 1998). Os resultados indicaram que a biblioteca é capaz de separar classes diferentes e gerar resultados similares ao SOM Toolbox.

[10] Strategy: padrão de projeto de software, que representa uma operação a ser realizada sobre os elementos de uma estrutura de objetos.
[11] Lattice: em Matemática, constitui-se de um conjunto parcialmente ordenado no qual todo par de elementos possui um único supremo e ínfimo.

3.4 aRT: a API R-TerraLib

A aRT (ART, 2008) consiste de um pacote do *software* estatístico R, desenvolvido pela Universidade Federal do Paraná (UFPR), para realizar a integração entre o R e a TerraLib. O intercâmbio desses *softwares* propicia ao analista de dados no R os recursos de acesso aos SGBD's e operações da TerraLib, e incorpora os resultados gerados no R ao SGBD de forma direta e imperceptível ao usuário. Esses resultados podem então ser visualizados pelo TerraView.

3.4.1 O *software* R

O *software* R (AMBIENTE R, 2008) consiste de uma linguagem e de um ambiente que possui uma variedade ampla de técnicas gráficas e estatísticas, constituindo-se em um condutor de métodos iterativos de análise de dados. O R é uma implementação gratuita da linguagem S, base do sistema comercial S-PLUS (S-PLUS, 2008).

Muitas técnicas estatísticas foram implementadas no ambiente R, sendo que algumas delas foram providas por meio de pacotes, como o gstat , geoR, geoRglm, dentre outros.

O R administra toda a computação em memória, e apresenta uma dificuldade considerável para a análise de conjuntos de dados muito grandes. Uma maneira de superar essa dificuldade é utilizar um mecanismo para acessar um SGBD externo, como PostgreSQL e MySQL.

3.4.2 A Integração Entre R, TerraLib e TerraView

A Figura 3.7, abaixo, ilustra o relacionamento entre o ambiente R (aRT), um ambiente SIG, no caso o TerraView, e a biblioteca TerraLib.

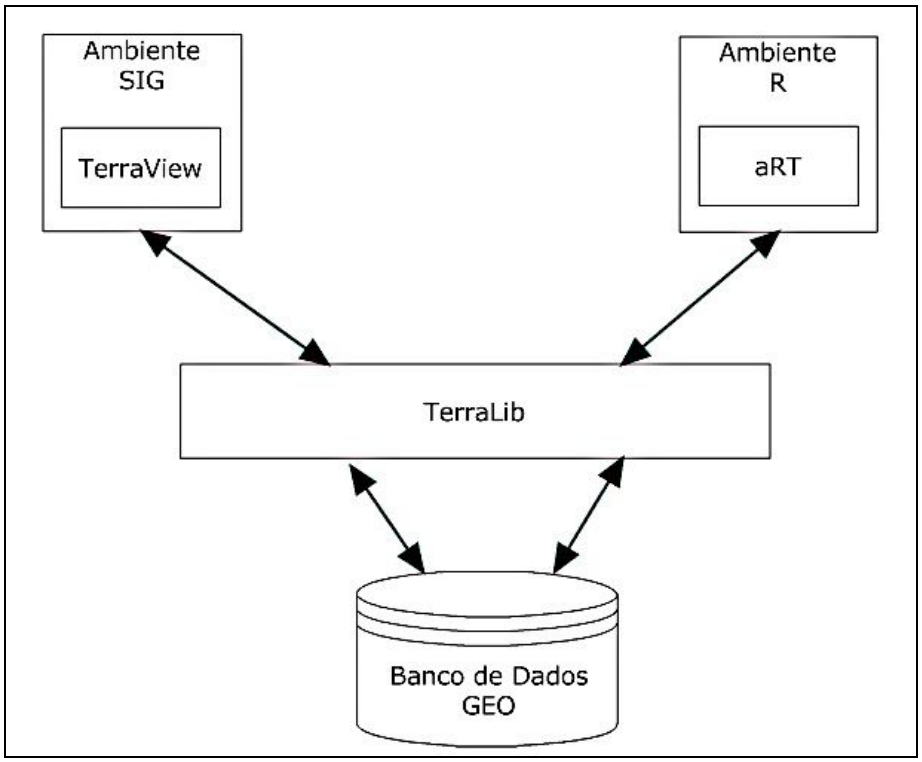

Figura 3.7 – Integração entre R/TerraView/TerraLib

Por meio dessa figura, podem-se observar as operações que podem ser realizadas por meio dessa integração.

Por intermédio do pacote aRT, os dados geográficos são recuperados em memória para o Ambiente R, já que o mesmo possibilita a utilização das estruturas da TerraLib. A partir desses dados, podem ser realizadas operações estatísticas, cuja saída também é armazenada em uma estrutura da TerraLib, e consequentemente, no banco de dados. Por intermédio da TerraLib, o *software* TerraView recupera os dados salvos anteriormente, para possibilitar a visualização em forma de mapas temáticos.

O pacote aRT é constituído de quatro classes para manipular os dados e funções da TerraLib, sendo elas: aRT, aRTdb, aRTlayer e aRTtheme. Objetos pertencentes à classe aRT permitem conexão com o SGBD para a realização de funções de administração do banco de dados. Depois de criado, o objeto armazena uma conexão virtual, onde todas as vezes que o acesso ao banco de dados é requerido, o objeto se conecta , executa a tarefa necessária e então se desconecta.

Um objeto da classe aRTdb, que depende da criação de um objeto aRT, provê o acesso e criação de banco de dados , sendo que objetos desse tipo mantêm todos os objetos da TerraLib que são utilizados pela aRT em memória.

A classe aRTLayer permite a manipulação de *layers*, e os dados devem ser convertidos para o formato aRT antes de sua inserção no banco de dados. A classe aRTtheme é voltada para usuários do TerraView, e de outros SIG's que utilizam a TerraLib como base, propiciando a visualização dos temas em sistemas desse tipo.

Exemplos práticos de utilização da aRT serão descritos no capitulo 6, com dados reais da plataforma Irrigap.

3.4.3 Dependências para Instalação da aRT

A aRT foi inicialmente desenvolvida para Linux. Para essa plataforma, sua versão mais estável é a 1.4.2, disponível para *download* em (ART, 2008). Alguns pré-requisitos de instalação são necessários para que a aRT opere corretamente sobre o Linux, além do pacote de instalação do R:

- SGBD MySQL, atualmente na versão de distribuição 5.0.21;
- Kit de desenvolvimento Qt, para C++, atualmente na versão 3.3.3;
- Biblioteca JPEG para possibilitar o trabalho com o formato de arquivo jpeg;

• Pacote sp, para R, a partir da versão 0.9-1, que possui classes e métodos para manipulação de dados espaciais;

• Biblioteca TerraLib.

O porte da aRT para Windows era algo desejado para usuários do pacote que trabalham com essa plataforma. Diante dessa necessidade, o INPE e a UFPR realizaram um trabalho em conjunto para atendê-la. Para tanto, o pacote aRT foi compilado sob um ambiente que simula de forma mínima, no sistema operacional Windows, o sistema operacional Linux.

No entanto, em outubro de 2006, surgiu a aRT 1.0.0 para Windows, que opera sem a necessidade de *softwares* adicionais como descrito acima. Além do *software* R, a biblioteca necessita dos seguintes pré-requisitos, para o seu correto funcionamento, em ambiente Windows:

• SGBD MySQL, a partir da versão de distribuição 4.1.6-gamma;

• Pacote sp, a partir da versão 0.7-12.

Para a plataforma Windows, a versão do *software* R instalada deve ser anterior à 2.4.

3.4.4 Considerações da Integração Entre Estatística Espacial e SIG

A integração entre Estatística Espacial e SIG propicia o fortalecimento das áreas envolvidas, aumentando a eficácia individual e comum de ambas. Ultimamente, tem-se observado um considerável progresso para integração entre essas áreas, com diferentes propostas de estratégias. Dentre elas, destacam-se a integração completa e os mecanismos de acoplamento fraco e acoplamento forte, além da combinação entre os dois últimos.

Na integração completa, a principal vantagem está no fato de as ferramentas de análise

estatística serem incorporadas diretamente no SIG. No entanto, a inclusão de um grande número de funções de estatística espacial em um SIG pode aumentar a complexidade do sistema para usuários comuns, além de demandar um grande investimento de tempo de programação e manutenção de *software* (FOOK, 2005).

No acoplamento fraco, os dados são exportados do SIG para dentro de um pacote de análise estatística de dados espaciais, e os resultados da análise são novamente retornados para o SIG, também por meio de exportação, para que sejam visualizados. Essa estratégia é computacionalmente simples de ser implementada. Para tanto, basta que os programas trabalhem com um formato de dados comum a ambos os *softwares*. Uma desvantagem para sua utilização é o overhead que pode haver nos processos de exportação e importação dos dados ou dos resultados quando muitos conjuntos de dados forem necessários para realização das análises estatísticas (FOOK, 2005).

No acoplamento forte, há uma ligação maior entre os *softwares* de SIG e análise estatística, pois este permite que rotinas do *software* estatístico sejam chamadas de dentro do SIG, ou vice-versa, ou ainda que o SIG e o *software* de análise compartilhem o mesmo banco de dados comum, sem a perda de estruturas de níveis mais altos, como topologia, identidade do objeto, meta-dados e outros relacionamentos (FOOK, 2005).

A integração entre o R e a TerraLib constitui uma estratégia de acoplamento forte.

4. A Plataforma Irrigap

Esse capítulo descreve as especificações da rede de sensores e atuadores de instrumentação fixa com comunicação sem fio e as ferramentas computacionais para irrigação espacialmente diferenciada em culturas perenes, que constituem o projeto Irrigap.

4.1 Introdução

O projeto Irrigap (TORRE-NETO, et al. 2007) (TORRE-NETO, et al. 2005) vem sendo desenvolvido pela Embrapa Instrumentação Agropecuária (CNPDIA) desde 2004. Seu desenvolvimento foi baseado em trabalhos iniciados por (TORRE-NETO,et al. 2001), com o intuito de criar um sistema de irrigação espacialmente variado, baseado em uma rede de sensores e atuadores sem fios. Este foi idealizado para ser de baixo custo, confiável e compatível com as práticas locais de produção de culturas perenes, visto que as redes de sensores e atuadores sem fios disponíveis no mercado ainda não apresentam baixo custo, principalmente para países que ainda estão em desenvolvimento.

4.2 Especificações e Características do Irrigap

O sistema desenvolvido é constituído dos seguintes componentes: Sensores e Atuadores distribuídos em rede, onde cada ponto de coleta de dados ou atuação constitui um Nó; uma ou mais Estações de Campo (EC); uma Estação Base (EB); um ou mais Kits de Instalação. Além disso, também faz parte do sistema todo o *software* desenvolvido para atender às necessidades.

Os Nós Sensores e Atuadores foram desenvolvidos para irrigação por gotejamento a taxa variável, baseados em parâmetros do solo. Os dados coletados pelos sensores são transmitidos a uma EC, que por sua vez envia comandos de atuação aos nós atuadores, por meio de protocolo de comunicação sem fios.

No que diz respeito à área de cobertura, uma EC deve ser capaz de atender uma área de 100 hectares, e cada uma delas deverá transmitir pacotes com dados coletados para a EB, bem como receber pacotes de atuação da mesma.

A EB constitui-se basicamente de um computador pessoal localizado em um determinado local. No entanto, dependendo do volume de dados trafegados na rede, e da necessidade de utilização do sistema de maneira compartilhada, podem ser instalados servidores de banco de dados e clientes da aplicação de gerenciamento em diferentes computadores.

A Figura 4.1, abaixo, mostra a arquitetura básica do sistema Irrigap.

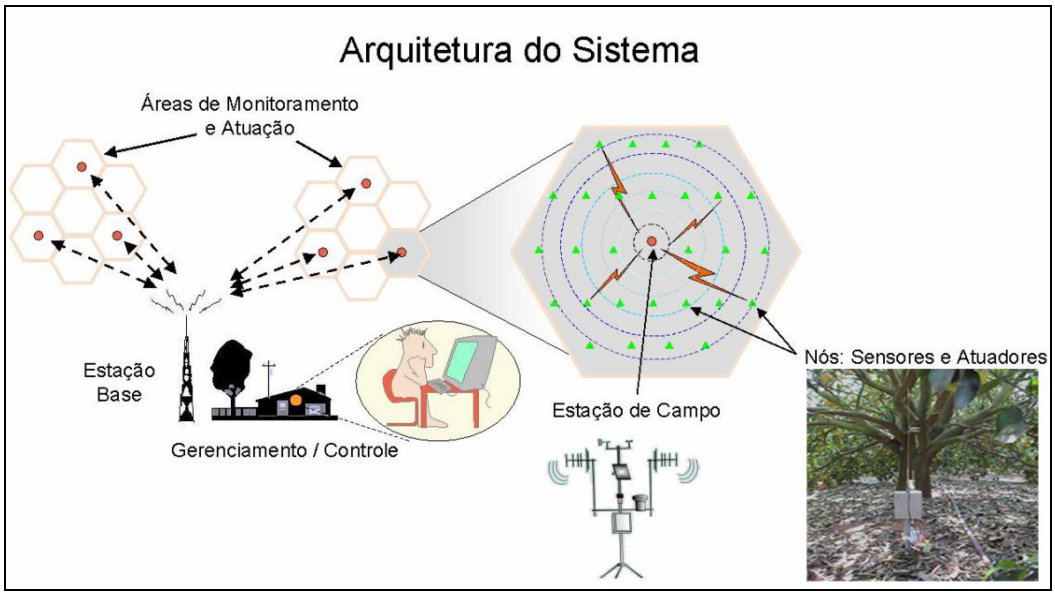

Figura 4.1 – Arquitetura básica do Sistema Irrigap

A seguir, segue uma descrição mais detalhada de cada um dos componentes do sistema Irrigap.

4.2.1 Nós Sensores e Atuadores

O Nó Sensor é o módulo que realiza a aquisição de dados referentes aos parâmetros do solo, utilizados para modelar as zonas de manejo da irrigação. Esses dados são de fundamental importância para a tomada de decisão em um sistema de irrigação. Utilizando-se de uma programação pré-determinada, os Nós Sensores se comunicam periodicamente com sua EC correspondente, enviando os dados coletados.

A montagem básica desse tipo de Nó é constituída por um sensor capacitivo para medida de umidade do solo, e sensores para medidas de temperatura do solo e do ar, uma caixa de alumínio com uma antena instalada na parte superior, e o *hardware* para transmissão de dados, acondicionado no interior da caixa. Seu circuito eletrônico é composto de um micro controlador capaz de coletar, processar e armazenar os dados dos sensores, e um transceptor (transmissor + receptor) para transmissão sem fios por radiofreqüência. A alimentação é realizada por uma bateria de lítio de 3.6V, com autonomia prevista para 12 meses. Os componentes utilizados para a composição do Nó são de tecnologia *nanowatt*, ou seja, de baixo consumo, para priorizar a economia de energia.

O Nó Atuador possui *hardware*, *software* e construção similares ao Nó Sensor. A principal diferença entre eles, está no fato de o Nó Atuador possuir um circuito de interface para válvulas solenóides e medidores de fluxo. Uma ponte H é utilizada para abrir e fechar as válvulas, devido às mesmas serem do tipo *latching*[12], e um circuito é utilizado para condicionar o sinal do medidor de fluxo, conectado a um pino contador de pulsos do micro controlador. Os Nós Atuadores são alimentados com quatro células de bateria NiCd de 1.2V, com capacidade de carga de 1000 mAh, e, por consumirem mais energia que os nós sensores, são equipados com um painel solar foto-voltaico de 1 *watt* para recarga das baterias. O ciclo

[12] Válvula Latching: tipo de válvula utilizado em sistemas de controle, é alternável entre duas posições para controlar o fluxo de trabalho de saída de um fluído para um atuador correspondente

de funcionamento, o intervalo de aquisição e a comunicação de dados são os mesmos dos Nós Sensores.

4.2.2 Estações de Campo

As Estações de Campo (EC) constituem elementos concentradores de dados da rede de sensores sem fios. Sua função principal é receber, processar e enviar dados para a Estação Base (EB). O envio desses dados é realizado por meio de conexões JDBC (*Java Database Connectivity*), mantendo o banco de dados da Estação Base sempre atualizado.

Outra função importante da EC é repassar os comandos para configurar a operação e atuação dos Nós e realizar o roteamento da rede de comunicação. Como a área de cobertura de uma EC é de aproximadamente 100 hectares, é possível que cada EC cubra aproximadamente 400 Nós, com espaçamento de 40 metros entre eles. No entanto, o protocolo de comunicação de dados está preparado para monitorar 65 mil Nós.

As ECs são compostas por uma CPU, uma mini-estação climatológica, uma interface de rede sem fio (WLAN) e um *driver* de comunicação com os Nós. Sua alimentação é realizada por um conjunto de duas baterias automotivas de 55 Ah cada, recarregadas por um painel solar de 70 *watts* de potência.

A CPU constitui-se de um PC104 baseado no processador NS Geode GX1, de 300 MHz, e *chipset* NS CS5530A (PCM-3550 da Advantech), com 256 *Megabytes* de memória RAM e uma memória *Compact Flash* de 1 *Gigabyte* funcionando como disco do sistema.

O sistema operacional utilizado é o Linux, distribuição Slackware 10.2, com *kernel* 2.4. Todo o *software* desenvolvido para as operações da EC utiliza as linguagens de programação ANSI C e Java.

Também compõe a estrutura da EC uma estação climatológica conectada a uma das

portas seriais RS-232 do PC104, além de um *Access Point* como interface de rede sem fios.

Para recepção e envio de dados à Estação Base são utilizados os serviços mais comuns do protocolo TCP/IP, como FTP, SMTP e SSH.

O único componente não comercial da EC é o *driver* de comunicação de dados com os Nós. Esse *driver* é composto de um micro controlador conectado a um módulo de rádio para transmissão e outro para recepção de dados dos nós, além de uma interface serial RS-232 para comunicação com o PC104. O nível de potência de transmissão é aumentado para 20 dB para que a EC possa atingir um raio de abrangência de aproximadamente 750 metros.

4.2.3 Estação Base

A Estação Base é utilizada para centralizar todas as operações e dados do sistema. Pode ser composta por apenas um computador pessoal, com *software* para gerenciamento e servidor de banco de dados instalado, ou possuir uma composição mais robusta, com servidores de banco de dados e clientes do *software* de gerenciamento instalados em diversos computadores. A conexão com as ECs acontece também por meio de um *Acess Point*, conectado à uma antena omni-direcional.

Uma base de dados espaço-temporal foi desenvolvida, juntamente com um *software* de gerenciamento da irrigação com funcionalidades de SIG, capaz de operar sobre o sistema e auxiliar na tomada de decisão para a irrigação à taxa variada.

A base de dados possui todas as informações referentes às unidades do sistema, tais como coordenadas geográficas de posicionamento global dos nós, modos de configuração e operação, e informações sobre a própria propriedade. A base para o desenvolvimento do *software* de gerenciamento foi a linguagem C++ e a biblioteca TerraLib, já descrita anteriormente.

O *software* permite a realização de programações da irrigação em dois modos distintos: convencional e inteligente. No primeiro, as regiões que constituem as unidades de irrigação são fixadas na instalação do sistema e não devem ser modificadas, possibilitando apenas a automação do acionamento das válvulas. Já no segundo, as regiões são definidas de acordo com a observação de seqüências de mapas de umidade do solo gerados pelo próprio *software*, permitindo a caracterização das zonas de manejo pelo próprio usuário do sistema. A partir dessa caracterização, a malha hidráulica do sistema de irrigação deve ser reajustada ou projetada (caso ainda não exista), para que as zonas formadas possam ser irrigadas de maneira espacialmente diferenciada.

É nesse momento que se observa com maior clareza a contribuição do sistema Irrigap para o processo, pois permite a operação sabendo onde, como e quanto irrigar, mantendo cada região irrigada o mais próximo possível da umidade ideal, e minimizando o consumo de água e energia elétrica sem afetar a produtividade.

4.2.4 Kit de Instalação

O Kit de Instalação tem como funções a configuração dos Nós da rede sem fios no ato da instalação e na manutenção, e a realização de coletas imediatas de dados.

Consiste de um conjunto de *softwares* que utilizam comunicação *bluetooth* com GPS e um *driver* para comunicação com os Nós via radiofreqüência. Esses *softwares* proporcionam a coleta de dados geográficos e alfanuméricos, acesso às configurações e funções do Nó e navegação por pontos georreferenciados.

4.2.5 Unidade Piloto do Sistema Irrigap

Com o intuito de avaliar a viabilidade e operabilidade do sistema como um todo, foi instalada e colocada em operação, em campo agrícola, uma Unidade Piloto. A área cedida – uma plantação de laranja (citros), contendo árvores com 10 anos de vida e aproximadamente 3 metros de altura, com espaçamento de aproximadamente 5 metros entre as árvores da mesma fileira e 8 metros entre as fileiras – pertence à Fischer Agropecuária e está localizada na Fazenda Maringá, em Gavião Peixoto-SP. Com latitude e longitude centrais de 21° 45' 28" S e 48° 27' 38" W, respectivamente, a área abrange uma extensão de terra de aproximadamente 25 hectares, que contempla um talhão[13] quadrado com aproximadamente 500 metros de lado.

A área utilizada possui sistema de irrigação instalado do tipo gotejamento, sendo que os bicos gotejadores são espaçados de 1 metro. Esse método proporciona grande eficiência, apresentando algumas vantagens como: economia de água pelo controle do gotejamento; fornecimento de quantidade de água estritamente necessária para cada planta; a água da irrigação pode ser fornecida por gravidade; a irrigação pode ser suspensa a qualquer hora; a rede de encanamentos não atrapalha os serviços normais de uma plantação, como colheitas e capinas.

Atualmente, estão instalados e em operação 55 Nós sensores, formando uma matriz de dimensão 7x7 (49 Nós), complementada por uma oitava coluna contendo apenas 6 Nós. Cada um desses Nós possui três sensores em operação, coletando dados de umidade e temperatura do solo e temperatura do ar em intervalos de uma hora. As Figuras 4.2 e 4.3 mostram, respectivamente, a área total da fazenda delimitada, e uma aproximação da área da unidade piloto, com a distribuição dos 49 Nós sensores da matriz 7x7.

[13] Talhão: Terreno delimitado para cultura

Figura 4.2 – Área delimitada da Fazenda Maringá, em Gavião Peixoto-SP

Figura 4.3 – Área da Unidade Piloto Instalada

Os mapas foram gerados com a utilização do *software* Quantum GIS, de código aberto, em sua versão 0.8.1-Titan, disponível em (QUANTUMGIS, 2008). As imagens geométricas georreferenciadas da Fazenda e da área da Unidade Piloto, bem como dos Nós

instalados, foram geradas pelo Kit Instalação descrito na seção anterior. A imagem de fundo foi obtida acessando-se o servidor de mapas *web* (WMS) da Nasa, disponível gratuitamente em (NASA, 2008).

5. SGMAp – *Software* para Geração de Mapas para Agricultura de Precisão

5.1 Introdução

O motivo maior para a execução desse trabalho é a comparação de diferentes métodos para a geração de mapas temáticos georreferenciados para apoio à tomada de decisão em Irrigação de Precisão. Na verdade, o estudo foi dirigido para a Irrigação de Precisão em virtude da disponibilidade de dados reais fornecidos por uma unidade piloto do sistema Irrigap, descrito no capítulo anterior. No entanto, nada impede que esse trabalho seja utilizado por outras áreas que compõem a Agricultura de Precisão.

Sendo assim, fez-se necessária a criação de um *software* capaz de gerar esses mapas, utilizando diferentes técnicas de interpolação de dados disponíveis. O SGMAp – *Software* para Geração de Mapas para Agricultura de Precisão - disponibiliza os algoritmos de interpolação fornecidos pela biblioteca TerraLib, além dos métodos de agrupamento disponibilizados pela mesma. Também possibilita a geração de mapas interpolados com a combinação de *n*-variáveis, onde o agrupamento é realizado pelos algoritmos disponibilizados pela SOMCode.

As seções a seguir descrevem as características e funcionalidades disponíveis nesse *software*.

5.2 Materiais e Métodos

Essa seção descreve os materiais e métodos utilizados para a criação do SGMAp, bem como o motivo pelos quais os mesmos foram utilizados.

5.2.1 Paradigma de Orientação a Objetos

A Orientação a Objetos tem sua origem nos anos 60, na Noruega, com Kristen Nygaard e Ole-Johan Dahl, no Centro Norueguês de Computação. No entanto, essa técnica foi mais bem conceituada no laboratório da Xerox, em Palo Alto (EUA), sendo refinada numa seqüência de protótipos da linguagem SmallTalk[14]. O líder desse projeto foi Alan Curtis Kay[15], considerado um dos criadores do termo "Programação Orientada a Objetos".

Alan Kay observou que o conceito de objetos tinha enorme potencial como uma ferramenta cognitiva: havia uma boa correspondência com a maneira de pensar das pessoas sobre o mundo. Ele percebeu que um substantivo isolado faz aparecer uma imagem concreta na mente das pessoas, efeito esse que um verbo isolado não possui. Na verdade, os verbos são propriedades dos substantivos: garotos correm; cachorro corre; água corre; trem corre. Na orientação a objetos isso é chamado de polimorfismo.

Kay pensou então em como construir um sistema de *software* a partir de agentes autônomos que interagissem entre si, estabelecendo os seguintes princípios de orientação a objetos:

• Qualquer coisa é um objeto;

[14] Smalltalk: Linguagem de programação orientada a objetos fracamente tipada, ou seja, em Smalltalk tudo é objeto, e não há tipos primitivos como em outras linguagens orientadas a objetos, como Java.
[15] Alan Cutis Kay: Cientista da computação norte-americano, conhecido por ter sido um dos pais da programação orientada a objetos, que lhe valeu o Turing Award de 2003.

• Objetos realizam tarefas por meio de requisição de serviços;

• Cada objeto pertence a uma determinada classe;

• Uma classe agrupa objetos similares;

• Uma classe possui comportamentos associados a um objeto;

• Classes são organizadas em hierarquias.

Depois de reunir conceitos de diversas áreas do conhecimento e com base em sua experiência como pesquisador, Alan Kay criou o paradigma de análise e programação mais eficiente da atualidade (HADDAD, 2008).

5.2.2 Principais Conceitos de Programação Orientada a Objetos

A idéia fundamental de linguagens orientadas a objetos é a possibilidade de combinar num único registro campos que conterão dados e campos que são funções para operar os campos de dados do registro. Uma unidade desse tipo é chamada de Classe, que é considerada um tipo de dado como os tipos que existem pré-definidos em compiladores de diversas linguagens de programação. Uma variável, ou instância de uma classe, é chamada de Objeto e contem campos de dados e funções.

As funções de um Objeto são chamadas de funções membro ou métodos e, geralmente, são o único meio de acesso aos campos de dados também chamados de variáveis de instancia.

Se o programa necessita atribuir valor a alguma variável de instância, deve chamar uma função-membro que recebe o valor como argumento e faz essa alteração, ou seja, variáveis de instância não podem ser acessadas diretamente. Dessa forma, os campos de dados estarão escondidos para prevenir alterações acidentais, caracterizando o conceito de Encapsulamento. O Encapsulamento simplifica a escrita, manutenção e alteração de

programas.

A programação orientada a objetos oferece uma maneira de relacionar classes umas com as outras por meio de hierarquias. Assim, uma classe pode ser dividida em subclasses, mantendo-se o princípio de que cada subclasse herda as características da classe da qual foi derivada. Esse conceito é chamado de Herança, onde a classe de origem é chamada de classe-base e as classes que compartilham as características de uma classe-base são chamadas de classes derivadas.

Da mesma maneira que se pode criar uma biblioteca de funções úteis a diversos programas, é possível formar uma biblioteca de classes que poderão vir a ser o núcleo de muitos programas. O uso de uma biblioteca de classes oferece uma grande vantagem sobre a de funções, pois o programador pode criar classes derivadas da classe-base da biblioteca. Assim, sem alterar a classe base, é possível adicionar a ela características diferentes que a tornarão capaz de executar exatamente o desejado.

O uso de classes derivadas aumenta a eficiência da programação pela não necessidade da repetição de código. A facilidade com que as classes existentes podem ser reutilizadas sem serem alteradas é um dos maiores benefícios da programação orientada a objetos.

Finalmente, o Polimorfismo consiste na criação de uma família de funções que compartilham do mesmo nome, sendo que cada uma possui código independente. O resultado da ação de cada uma das funções da família é o mesmo, mas a maneira de atingir esse resultado é distinta.

A Sobrecarga é um tipo particular de Polimorfismo. Um exemplo de Sobrecarga é a utilização de um mesmo símbolo para a execução de operações distintas, como acontece com a operação de soma pelo computador para números inteiros e números reais (MIZRAHI, 1995).

5.2.3 Vantagens da Programação Orientada a Objetos e Sua Utilização no Desenvolvimento do SGMAp

A Orientação a Objetos traz vários benefícios no desenvolvimento e manutenção de *software*. As vantagens que representam conseqüências diretas da adoção desse paradigma são:

• Facilidade para reutilização de código e, consequentemente, do projeto;

• Possibilidade de desenvolvimento em um nível mais alto de abstração;

• Utilização de um único padrão conceitual durante todo o processo de criação do *software*;

• Melhor adequação à arquitetura cliente/servidor;

• Facilidade de comunicação com os usuários e outros profissionais de informática.

Já as principais vantagens desse paradigma, com relação ao que de fato é esperado se obter com essa tecnologia são:

• Ciclo de vida mais longo para os sistemas;

• Desenvolvimento acelerado dos sistemas;

• Possibilidade de construção de sistemas muito mais complexos, pela incorporação de funções prontas;

• Menor custo para desenvolvimento e manutenção de sistemas. (FACCAMP, 2008)

O desenvolvimento do SGMAp foi baseado nesse paradigma por conta de todas as vantagens apresentadas acima, e pelo fato de as bibliotecas TerraLib e SOMCode o utilizarem em seu desenvolvimento.

5.2.4 A Linguagem C++

5.2.4.1 Introdução

A linguagem C++ foi desenvolvida inicialmente por Bjarne Stroustrup, na AT&T, entre os anos de 1979 e 1983. Surgiu a partir da linguagem C, com a idéia de agregar à mesma o conceito de orientação a objetos. Por diversas razões, C++ deve ser encarada como uma outra linguagem, e não como um super-conjunto de C, dentre elas:

• Todo *software* escrito em C nem sempre é compilado em C++ sem conter erros, podendo nem mesmo gerar o mesmo resultado, pois a sintaxe e a semântica de algumas construções são diferentes;

• Ao serem unidas partes de um *software* escrito em C++ com partes de um *software* escrito em C estas devem ser bem especificadas, pois as convenções de funcionamento de código compilado também são diferentes;

• A linguagem C++ dispõe de um conjunto de mecanismos básicos não presentes em C, que devem ser utilizados para a produção de *software* mais modular e confiável;

• Os mecanismos de C++ devem inspirar o desenvolvimento seguindo o paradigma de orientação a objetos.

A partir da primeira versão, de 1983, a linguagem foi sendo revisada e tornou-se disponível fora da AT&T em 1985, e padronizada pela ISO somente no final de 1997, pelo padrão ISO/IEC 14882[16]. (JOHANN, 2004).

Com relação à utilização da linguagem C++ no desenvolvimento do *software* SGMAp, a mesma foi escolhida devido ao fato de as bibliotecas TerraLib e SOMCode terem sido desenvolvidas nessa linguagem, facilitando o reaproveitamento de classes já escritas nas

[16] ISO/IEC 14882: Padrão oficial da linguagem de programação C++, inicialmente publicada em 1998 e revisada em 2003.

mesmas.

5.2.4.2 O IDE Borland C++ Builder

O IDE[17] Borland C++ Builder (BORLAND, 2008) é um ambiente visual de desenvolvimento de aplicações orientado a objetos, e que permite desenvolver de forma rápida aplicações para Windows. Além disso, disponibiliza um ambiente de ferramentas RAD (Desenvolvimento Rápido de Aplicações) e um extenso suporte à programação orientada a objeto, formado por duas bibliotecas de componentes reutilizáveis (VCL - *Visual Components Library* e CLX - *Components Library for Cross-Plataform*). Essas bibliotecas contêm os objetos que encapsulam as técnicas utilizadas para o desenvolvimento de aplicações.

Atualmente, encontra-se na versão 2006. A versão utilizada nesse projeto é a 6.0. A presença da VCL e da CLX foi o principal motivo da escolha desse IDE para o desenvolvimento do *software* SGMAp, pois essas bibliotecas possibilitaram um desenvolvimento rápido e eficiente desse *software*, principalmente dos componentes relacionados à camada de interface visual.

5.2.5 O Banco de Dados PostgreSQL

5.2.5.1 Introdução

Devido à grande quantidade de dados disponibilizados pelo sistema Irrigap, com atualizações que ocorrem de hora em hora, se fazia necessária a utilização de um SGBD

[17] IDE: *Integrated Development Environment* – Ambiente Integrado para Desenvolvimento de Software, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar o processo.

(Sistema de Gerenciamento de Banco de Dados) para exibi-los no *software* SGMAp. O SGBD PostgreSQL foi o escolhido para essa implementação por fornecer algumas vantagens, dentre elas a sua total integração com a biblioteca TerraLib tanto na forma nativa, quanto com a utilização da extensão espacial PostGIS.

5.2.5.2 Características do PostgreSQL

O PostgreSQL é um SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional de código aberto, com mais de quinze anos de desenvolvimento. É extremamente robusto e confiável, além de ser bastante flexível e rico em recursos. Além disso, é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados. A equipe de desenvolvimento do PostgreSQL sempre teve uma grande preocupação em manter a compatibilidade com os padrões SQL92/SQL99[18].

Pela riqueza de recursos e conformidade com os padrões, é um SGBD muito adequado para o estudo universitário do modelo relacional, além de ser uma ótima opção para empresas implementarem soluções de alta confiabilidade sem altos custos de licenciamento. É um programa distribuído sob a BSD *License*[19], o que torna o seu código fonte disponível e o seu uso livre para diversos tipos de aplicações, sejam elas comerciais ou não. O PostgreSQL foi implementado em diversos ambientes de produção no mundo, entre eles, o banco de dados que armazena os registros de domínio .org, mantido pela empresa Afilias.

[18] Padrões da *Structured Query Language* (SQL), que é uma linguagem de pesquisa declarativa para banco de dados relacional.

[19] Licença de software livre inicialmente utilizada nos sistemas operacionais do tipo *Berkeley Software Distribution* (um sistema derivado do Unix). Apesar de ter sido criada para os sistemas BSD, atualmente vários outros sistemas são distribuídos sob esta licença.

As versões mais recentes possuem, dentre outros, os seguintes recursos:

- Sub-consultas;
- Controle de concorrência multi-versão (MVCC);
- Integridade Referencial;
- Funções armazenadas (*Stored Procedures*), que podem ser escritas em várias linguagens de programação (PL/PgSQL, Perl, Python, Ruby, e outras);
- Gatilhos (*Triggers*);
- Tipos definidos pelo usuário;
- Esquemas (*Schemas*);
- Conexões SSL. (POSTGRESQL, 2008)

No que diz respeito ao desempenho, o PostgreSQL tem obtido bons resultados, quando comparados a outros SGBD's, como Oracle, MySQL e Firebird. Um *benchmark* envolvendo junções de tabelas para esses SGBD's foi realizado pela Universidade Estadual de Campinas (Unicamp), e está disponível em (GUIMARÃES, 2006). Nesse teste, ficou provado que o PostgreSQL teve desempenho superior aos outros na maioria dos casos, mostrando assim a eficiência de sua utilização em bancos de dados que possuem consultas envolvendo, por exemplo, dados de diferentes tabelas. Para a administração e gerenciamento das bases de dados do PostgreSQL, é utilizado o *software* PgAdmin, que faz parte do pacote de distribuição desse SGBD.

5.2.5.3 A Extensão Espacial PostGIS

PostGIS é uma extensão espacial gratuita e de código fonte aberto ao Sistema de Banco de Dados Objeto-Relacional PostgreSQL, permitindo que objetos geográficos sejam

armazenados no banco. Inclui suporte para índices baseados em GiST e índices espaciais R-Tree, além de funções para análise básica e processamento de objetos geográficos.

Resumindo, PostGIS "habilita espacialmente" o servidor PostgreSQL para ser utilizado como um banco de dados espacial para Sistemas de Informações Geográficas (SIG), como faz a extensão *Spatial* do Oracle.

PostGIS segue a especificação OpenGIS para SQL "*Simple Features*", e é distribuído sobre a GNU *General Public License*[20].

5.2.6 Arquitetura do SGMAp e Integração com as bibliotecas TerraLib e SOMCode

O SGMAp foi desenvolvido utilizando diferentes camadas, de forma a facilitar as manutenções e implementações futuras.

A camada de objetos representativos de tabelas possui classes que materializam cada uma das tabelas presentes no banco de dados, como Talhão, Nós e Sensores, em objetos capazes de armazenar as propriedades dessas tabelas em memória.

A camada de acesso ao banco de dados permite recuperar as informações necessárias à camada anterior por meio de objetos DAO[21], responsáveis pela implementação de mecanismos de busca de dados relacionados às tabelas presentes no banco de dados.

[20] Licença para software livre idealizada por Richard Stallman no final da década de 1980, no âmbito do projeto GNU da *Free Software Foundation*. Permite que os programas sejam distribuídos e reaproveitados, mantendo, porém, os direitos do autor de forma a não permitir que essa informação seja usada de uma maneira que limite as liberdades originais.

[21] DAO: *Data Acesss Object* - Objeto que provê uma interface abstrata para alguns tipos de bancos de dados, fornecendo operações específicas sem expor detalhes sobre o banco de dados.

A camada de interface com o usuário e bibliotecas externas armazena formulários de entrada de dados e, por intermédio dos objetos que representam esses formulários, permite o acesso às classes das bibliotecas TerraLib e SOMCode.

A Figura 5.1, abaixo, ilustra o relacionamento entre essas camadas, bem como a troca de dados entre as mesmas, indicadas pelas setas bi-direcionais:

Figura 5.1 – Relacionamento entre as camadas do SGMAp

A Figura 5.2, abaixo, ilustra a integração existente entre o SGMAp, o banco de dados e as bibliotecas TerraLib e SOMCode.

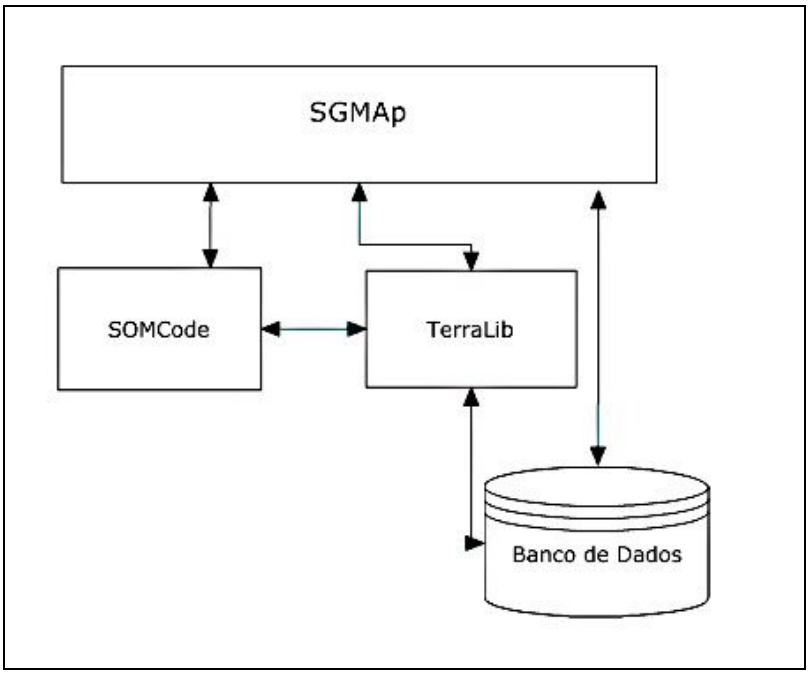

Figura 5.2 – Integração entre o SGMAp, banco de dados e bibliotecas externas

Por meio dessa Figura, pode ser observado que o SGMAp possui acesso direto ao banco de dados. Esse acesso é realizado por meio de uma conexão ODBC[22] e se faz necessário para a recuperação de informações de entidades do sistema, tais como Talhões, Nós e Sensores.

Já a conexão com a biblioteca TerraLib se faz necessária quando a operação a ser realizada no sistema envolve dados geográficos como polígonos, pontos e células. Esse tipo de operação é realizado, por exemplo, quando um mapa temático é exibido. A conexão com a TerraLib se inicia com a instanciação de um objeto da classe TeDatabase, que abre uma conexão com o banco de dados no modelo da TerraLib. No caso do SGMAp, os objetos TeDatabase são criados por meio da classe filha TePostGIS, que implementa métodos de acesso ao SGDB PostgreSQL com extensão espacial PostGIS, utilizado nesse trabalho.

Finalmente, a conexão com a biblioteca SOMCode restringe-se apenas ao uso dos algoritmos de treinamento da rede neural de Kohonen, utilizados para a formação de agrupamentos para a exibição dos mapas temáticos.

A SOMCode também está integrada diretamente à TerraLib, no sentido de receber os dados provenientes do banco necessários para a execução dos seus algoritmos. Dessa maneira, a SOMCode não realiza acesso direto ao banco de dados.

[22] Do inglês *Open Data Base Connectivity*: É uma tecnologia padrão de programação para o acesso a banco de dados por meio de uma biblioteca de funções pré-definida, criada pelo SQL *Access Group*.

5.3 Interface e Funcionalidades do SGMAp

5.3.1 Considerações Iniciais

A interface com o usuário do SGMAp foi desenvolvida de maneira a atender requisitos de usabilidade, com parâmetros de configuração de algoritmos já pré-definidos. A Figura 5.3, abaixo, mostra a interface inicial do sistema, que já disponibiliza a maioria dos parâmetros a serem escolhidos pelo usuário.

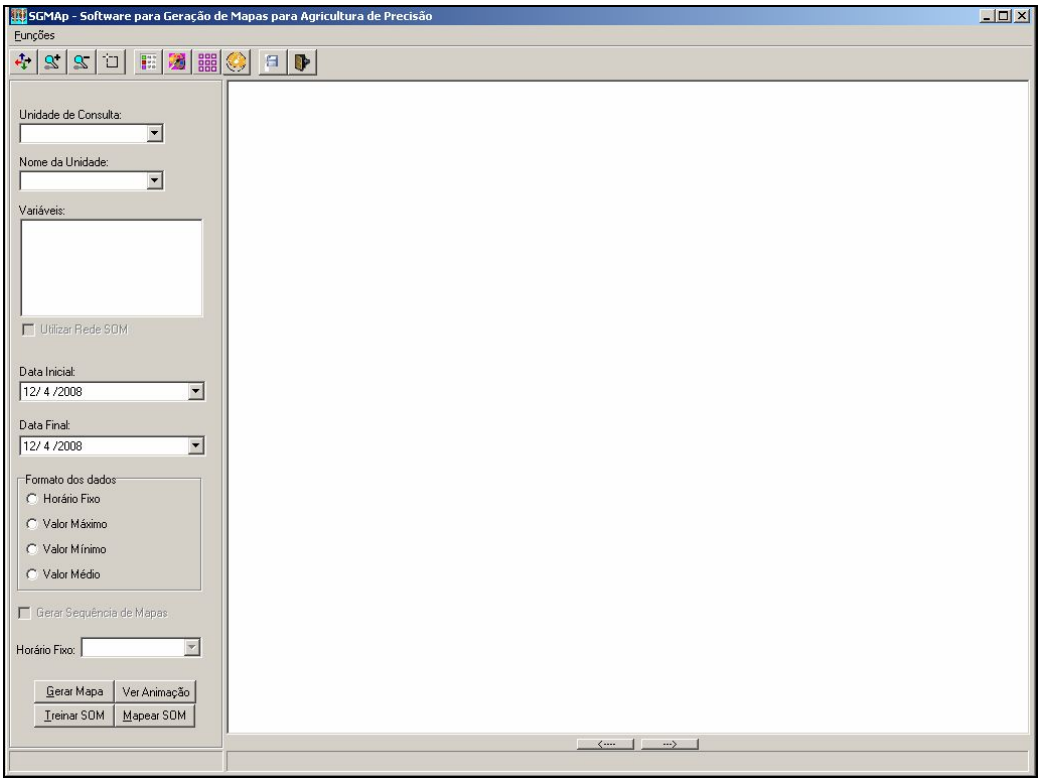

Figura 5.3 – Interface principal do SGMAp

A Barra de Ferramentas, localizada logo abaixo do menu “Funções” , possui botões de acesso à funcionalidades relacionadas à visualização dos mapas, como diferentes tipos de *zoom* e movimentação, além de acesso às configurações dos algoritmos da TerraLib e da SOMCode. Essas configurações podem ser acessadas também pelo menu “Funções”.

Para a descrição das opções da interface principal, vamos considerar o caminho usual para escolha das mesmas. Todas as funcionalidades do SGMAp serão descritas com mais detalhes nas seções seguintes.

Primeiramente, o usuário tem a opção de escolher qual a unidade de consulta de seu interesse para a geração dos mapas, por meio da caixa de seleção “Unidade de Consulta”. As opções disponíveis são:

• “Fazenda”, que possibilita a geração de mapas que consideram a área total da propriedade rural;

• “Variedade”, que possibilita a geração de mapas que consideram apenas uma variedade do produto cultivado na propriedade rural;

• “Talhão”, que possibilita a geração de mapas que consideram apenas um talhão da propriedade rural.

Na seqüência, a própria unidade que será utilizada para a geração dos mapas deverá ser escolhida, por meio da caixa de seleção “Nome da Unidade”. Geralmente, nas propriedades rurais, os talhões são numerados, ou seja, o nome do talhão corresponde na verdade a um número. Entretanto, o nome de cada unidade deverá ser informado no momento do seu cadastro no banco de dados, da maneira como foi definido no sistema de cadastro da propriedade. Para esse trabalho, foi utilizado o cadastro realizado pelo *software* de gerenciamento da unidade piloto do sistema Irrigap.

Após ser delimitada a área para a geração dos mapas, as variáveis que serão utilizadas para o procedimento devem ser selecionadas, por meio da caixa de opções “Variáveis”. O principal motivo para se ter a opção de seleção de *n* variáveis se deve à possibilidade de utilização da SOMCode, que realiza o agrupamento dos dados a serem exibidos. A caixa de opção “Utilizar Rede SOM”, quando selecionada, habilita o SGMAp para utilizar o algoritmo configurado para a SOMCode no momento da visualização dos dados em forma de mapa.

As caixas de seleção de data “Data Inicial” e “Data Final”, como os próprios nomes já dizem, solicitam ao usuário a escolha do período a ser considerado para os dados de entrada, visto que os mesmos se encontram armazenados em uma base de dados espaço-temporal. Dessa forma, poderão ser gerados mapas em diferentes intervalos de tempo.

O grupo de escolha “Formato dos Dados” possibilita ao usuário escolher como os dados serão tratados para a geração dos mapas, e está intimamente ligado à opção “Gerar Seqüência de Mapas” e à caixa de seleção “Horário Fixo”, possibilitando diferentes combinações para a geração do resultado.

Se o usuário escolher o formato de dados com horário fixo, os mapas serão gerados levando em conta apenas os valores das variáveis coletados no horário definido em “Horário Fixo”. Já as opções de valor máximo, mínimo e médio, se escolhidas, fazem com que os mapas gerados levem em conta, respectivamente, os valores máximos, mínimos e médios coletados das variáveis escolhidas, no período especificado.

A opção “Gerar Seqüências de Mapas”, se selecionada, faz com que sejam gerados mapas diários dentro do período especificado, caso este seja de dois ou mais dias. Os mapas diários de uma seqüência podem ser visualizados um a um, por meio dos botões localizados logo abaixo da área de exibição dos mapas.

Os quatro botões localizados abaixo da opção “Horário Fixo” são responsáveis pela execução das principais funções do sistema.

O botão “Gerar Mapa” executa o algoritmo de geração dos mapas de acordo com as opções selecionadas, e exibe os mapas na área branca, localizada à direita, também de acordo com as opções de agrupamento selecionadas.

O botão “Ver Animação” abre uma nova janela, que possibilita a exibição seqüencial dos mapas gerados.

O botão “Treinar SOM” realiza o treinamento da Rede SOM, de acordo com os

parâmetros escolhidos pelo usuário.

Finalmente, o botão “Mapear SOM” executa o algoritmo para geração do Mapa de Kohonen, considerando o último treinamento realizado pela Rede SOM.

5.3.2 Geração de Mapas Temáticos por Interpolação Utilizando a TerraLib

A primeira funcionalidade desenvolvida para o SGMAp é a que permite a geração de mapas temáticos obtidos por interpolação, por meio de um dos cinco algoritmos descritos na seção 3.2.6. A configuração do algoritmo a ser utilizado, bem como dos parâmetros que devem ser determinados pelo usuário, são definidos utilizando-se o formulário “Configurações da TerraLib”, que pode ser acessado tanto pelo seu botão correspondente na barra de ferramentas, quanto pelo menu “Funções”. A Figura 5.4, abaixo, ilustra esse formulário:

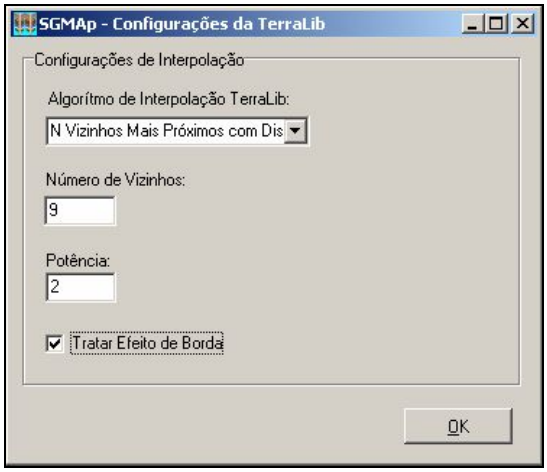

Figura 5.4 – Configurações dos Algoritmos da TerraLib

Por meio dessa figura, pode-se observar que, além da escolha do algoritmo de interpolação a ser utilizado, deve também ser especificado o número de vizinhos e a potência, no caso de algoritmos que necessitem dessa configuração. Além disso, a opção “Tratar Efeito de Borda” permite realizar a interpolação dos dados utilizando um algoritmo para tratamento

desse problema.

Os “efeitos de borda” são erros gerados durante o processo de interpolação, em valores muito próximos ao limite delimitado para a região de interesse. Esse efeito ocorre porque o processo de interpolação calcula os valores de cada célula de saída com base nas menores distâncias dos pontos amostrais, e quando a célula está próxima ao limite, tende a se basear em valores de amostras internas, aumentando consideravelmente a variação nessas regiões (NAMHEE, 2008).

O algoritmo utilizado pelo SGMAp realiza o tratamento desse problema considerando no processo de interpolação pontos localizados na região externa à região de interesse, sendo estes com valores reais ou simulados quando não existem.

Foi definido, a partir dos resultados obtidos nos testes realizados com todos os algoritmos, que o algoritmo dos *n* vizinhos mais próximos com distância, que equivale ao algoritmo tradicional de interpolação do inverso da distância elevado a uma potência, considerando nove vizinhos e potência quadrática, seria considerado como padrão, sem a necessidade de indicação pelo usuário.

Além da definição dessas configurações, também devem ser definidos para a execução dessa funcionalidade os parâmetros relacionados aos dados espaço-temporais, como demonstrado na seção 5.3.1. A Figura 5.5, abaixo, mostra o resultado da geração de um mapa de umidade do solo em uma determinada data, utilizando o algoritmo definido como padrão, além do tratamento do “efeito de borda”.

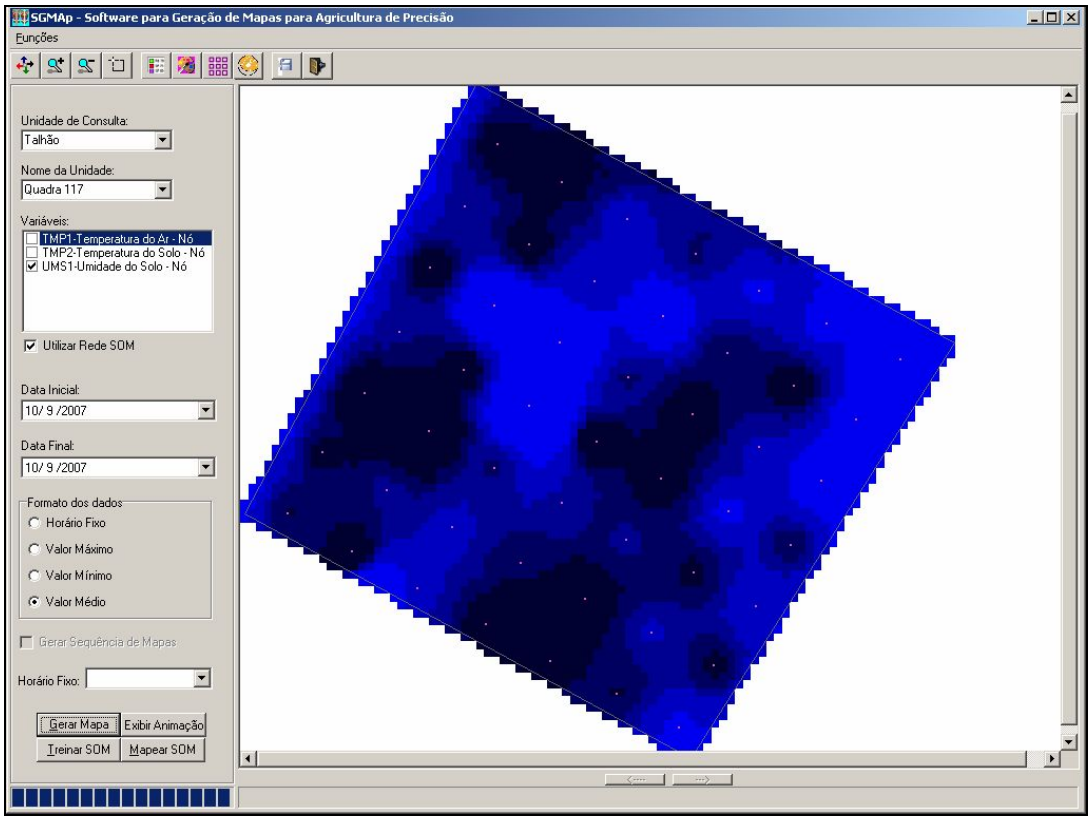

Figura 5.5 – Exemplo de mapa temático gerado

A operação de interpolação de dados é executada por meio do botão “Gerar Mapa”. O algoritmo tem a função de recuperar os dados persistidos no banco de acordo com as opções selecionadas pelo usuário - que nesse caso são o contorno referente ao talhão e os dados médios de umidade do solo nos pontos de medida inseridos dentro desse contorno – e repassa-los no formato correto ao algoritmo de interpolação da TerraLib. Por sua vez, a TerraLib devolve uma estrutura de dados celular contendo os valores resultantes da interpolação, possibilitando ao algoritmo de geração de mapas exibi-lo na área de desenho, conforme mostra a Figura 5.5.

Outros resultados obtidos com a execução dessa funcionalidade serão exibidos na seção 6.3.

5.3.3 Geração e Configuração de Legendas e Agrupamentos

Além da interpolação de dados pontuais para a obtenção dos valores correspondentes às células de saída de um mapa temático georreferenciado, é de extrema importância a configuração do formato de exibição dos mesmos. Para tanto, a biblioteca TerraLib disponibiliza classes e funcionalidades que têm a função de proporcionar variados tipos de configuração para legendas e agrupamentos.

A Figura 5.6, abaixo, ilustra o formulário "Editar Legenda e Agrupamento", disponibilizado pelo SGMAp e que pode ser acessado por meio do sub-menu "Definir Cores" do menu "Funções", ou pelo seu botão correspondente na barra de ferramentas.

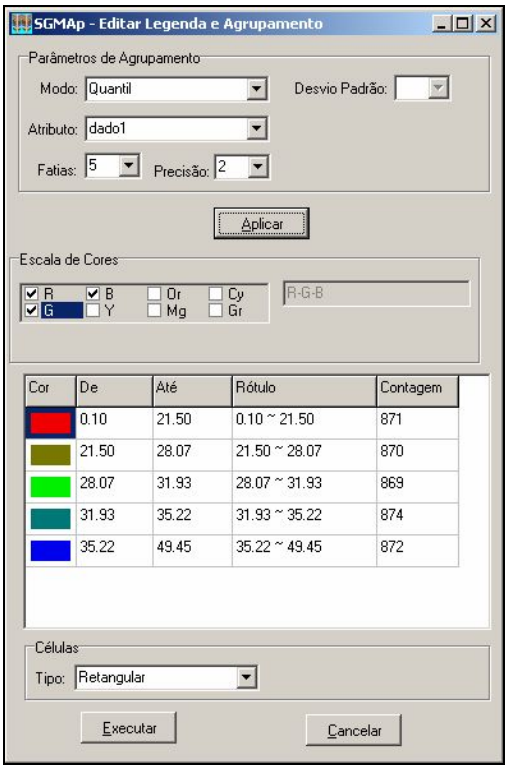

Figura 5.6 – Formulário Editar Legenda e Agrupamento

O grupo "Parâmetros de Agrupamento" tem como principal função possibilitar ao usuário escolher o modo de agrupamento que deseja, baseado nos modos disponibilizados

pela TerraLib, que são:

• Passos Iguais: Divide as células geradas pelo algoritmo de interpolação em passos iguais com relação aos intervalos a serem criados para a legenda. O usuário deve selecionar para esse modo o número de fatias que deseja para a legenda.

• Quantil: Divide as células geradas pelo algoritmo de interpolação de forma que todos os intervalos a serem criados para a legenda possuam um número aproximadamente igual de células. Para esse modo, o usuário deve selecionar também o número de fatias que deseja para a legenda.

• Desvio Padrão: Divide as células geradas pelo algoritmo de interpolação de forma que todos os intervalos a serem criados para a legenda levem em conta um desvio padrão, que deve ser determinado pelo usuário. Nesse modo, o número de fatias é gerado automaticamente.

• Valor Único: Divide as células geradas pelo algoritmo de interpolação de forma que seja criada uma fatia da legenda para cada valor atribuído às células. Nesse modo, o número de fatias também é gerado automaticamente.

Para todos os modos listados acima, é necessário que o usuário escolha a precisão de casas decimais e o atributo a ser associado às células.

5.3.4 Exibição da Legenda

Uma das características do SGMAp, que está ligada ao modelo de dados da TerraLib, é a geração de legendas para os mapas temáticos. Conforme descrito na seção 5.3.3, o SGMAp disponibiliza várias modos para a geração de legenda, por meio da formação de agrupamentos. Todo mapa gerado por ele possui uma legenda associada, e que pode ser

redefinida pelo usuário.

A legenda do mapa que está sendo exibido pode ser visualizada de duas maneiras: pelo botão “Exibir Legenda”, localizado na barra de ferramentas, ou pelo sub-menu de mesmo nome, localizado no menu “Funções”. A Figura 5.7, abaixo, exibe uma legenda gerada para um mapa.

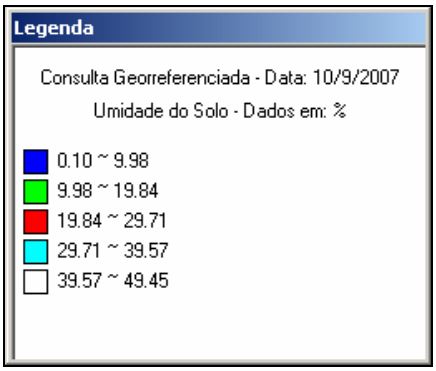

Figura 5.7 – Legenda gerada pelo SGMAp

Os dados referentes às faixas e cores utilizadas pela legenda, bem como os agrupamentos formados, são armazenados no banco de dados pela TerraLib, por meio da classe TeLegend.

5.3.5 Geração de Mapas Temáticos Interpolados utilizando a TerraLib em conjunto com a SOMCode

O SGMAp permite ao usuário, quando o mesmo escolhe mais de uma variável para a geração dos mapas temáticos, decidir se vai utilizar ou não a Rede SOM para combinar os dados, por meio da caixa de seleção “Utilizar Rede SOM”. Para tanto, os parâmetros da Rede SOM deve sem configurados, por meio do formulário “Parâmetros da Rede SOM”, que pode

ser acessado pelo botão “Parâmetros da Rede SOM”, presente na barra de tarefas, ou pelo sub-menu de mesmo nome, presente no menu “Funções”. A Figura 5.8 ilustra esse formulário.

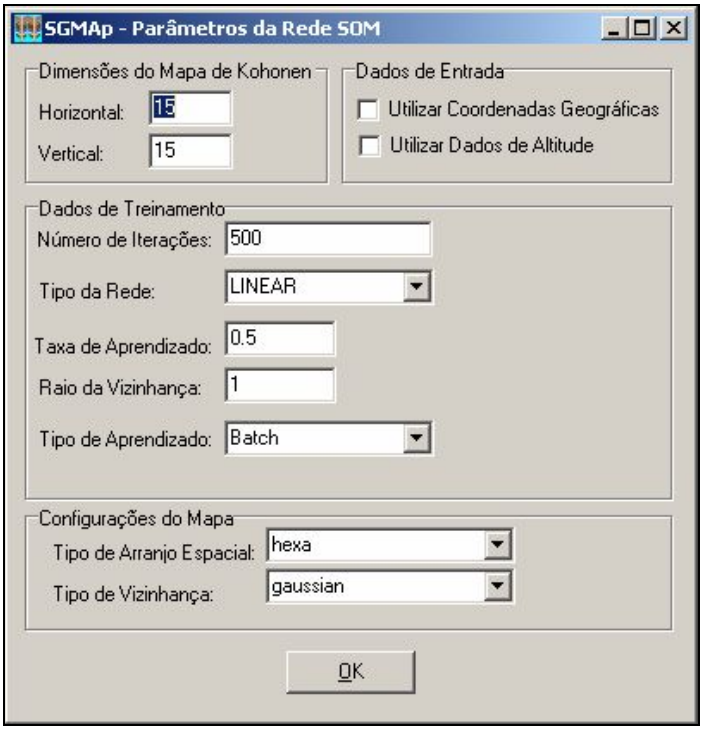

Figura 5.8 – Parâmetros da Rede SOM

O primeiro grupo, denominado “Dimensões do Mapa de Kohonen”, define as dimensões horizontal e vertical do mapa de Kohonen a ser gerado. O grupo “Dados de Entrada” fornece ao usuário a opção de incluir as coordenadas geográficas das células como dados de entrada para o algoritmo da Rede SOM.

O grupo “Dados de Treinamento” permite ao usuário configurar parâmetros para o treinamento da Rede SOM, tais como o número de iterações a ser realizado; o tipo da rede, que pode ser linear, aleatória ou janelada; a taxa de aprendizado e o raio da vizinhança a serem utilizados; e o tipo de aprendizado, que pode ser o padrão, em lote ou on-line.

Finalmente, o grupo “Configurações do Mapa” permite configurar o Mapa de Kohonen a ser gerado, com relação ao tipo de arranjo espacial, que pode ser hexagonal ou retangular; e o tipo de vizinhança, que pode ser gaussiana, *bubble* ou *cutgauss*.

Todas as opções disponibilizadas nesse formulário fazem parte dos parâmetros de configuração utilizados pela SOMCode.

O SGMAp possibilita a geração de dados de treinamento antes da geração do mapa temático. No entanto, essa operação não é necessária, e se a mesma não for executada, serão utilizados os dados de treinamento gerados pela última vez. Essa funcionalidade é executada por meio do sub-menu “Gerar Dados de Treinamento”, do menu funções. O algoritmo para geração desses dados escolhe, aleatoriamente, um conjunto pequeno de amostras pertencente ao conjunto de dados a ser utilizado.

Com o conjunto de dados de treinamento selecionado, a Rede SOM pode ser treinada, por meio do botão “Treinar SOM”, e mapeada, por meio do botão “Mapear SOM”. As configurações a serem utilizadas pelos respectivos algoritmos de execução serão aquelas que foram definidas no formulário “Parâmetros da Rede SOM”.

Ao final da execução desses dois algoritmos, o Mapa de Kohonen gerado pode ser visualizado por meio do formulário “Mapa de Kohonen Gerado”, como mostra a Figura 5.9. Esse formulário pode ser acessado por meio do sub-menu “Visualizar Mapa de Kohonen, presente no menu “Funções”.

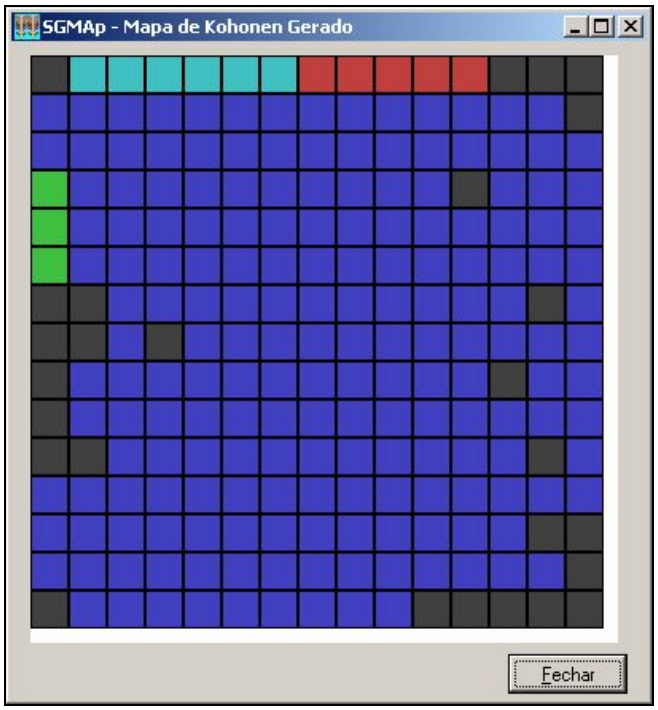

Figura 5.9 – Exemplo de Mapa de Kohonen Gerado

No exemplo acima, foi gerado um mapa com dimensões 15x15. As diferentes cores exibidas no mapa correspondem aos agrupamentos gerados. Para esse caso, foram identificados cinco tipos diferentes de agrupamentos.

Finalmente, o mapa temático utilizando-se de *n* variáveis pode ser gerado, por meio do botão "Gerar Mapa". Com a opção de utilização da Rede SOM, o SGMAp deverá executar o procedimento de interpolação de maneira diferente da utilizada para a interpolação por uma única variável. Assim, o algoritmo de interpolação da TerraLib escolhido será executado *n* vezes, para cada uma das variáveis escolhidas, gerando *n* mapas de células interpolados.

A função da Rede SOM nesse momento é facilitar a geração de um único mapa, a partir dos *n* mapas gerados, de acordo com o Mapa de Kohonen obtido no treinamento. Assim, a Rede SOM fica configurada da seguinte maneira, com relação às camadas de entrada e saída:

• Camada de entrada: valores interpolados associados às células após a execução dos

algoritmos de interpolação da TerraLib. A dimensão do espaço de entrada corresponde ao número de variáveis escolhidas.

• Camada de saída: células com valores associados de acordo com o agrupamento a que a mesma deve pertencer, determinado pelo mapeamento realizado pela Rede SOM.

A etapa de treinamento é realizada com a escolha de amostras aleatórias de valores interpolados para células dos *n* mapas gerados. Já na etapa de mapeamento, é utilizado o Mapa de Kohonen gerado pelo treinamento, e os dados interpolados obtidos para cada uma das células de cada uma das variáveis, na data ou período escolhidos.

As Figuras 5.10 e 5.11 mostram, respectivamente, a configuração de um agrupamento e o mapa temático gerado após essa configuração.

Por meio da Figura 5.10, pode ser observado que o agrupamento, no caso de utilização de *n* variáveis, deve ser formado utilizando-se o atributo "rótulo", no modo "Valor Único". Isso porque, quando utiliza a Rede SOM, o algoritmo de geração de mapas atribui, para cada amostra de entrada, que constitui um vetor com *n* valores de entrada, o grupo correspondente à mesma no mapa de Kohonen gerado.

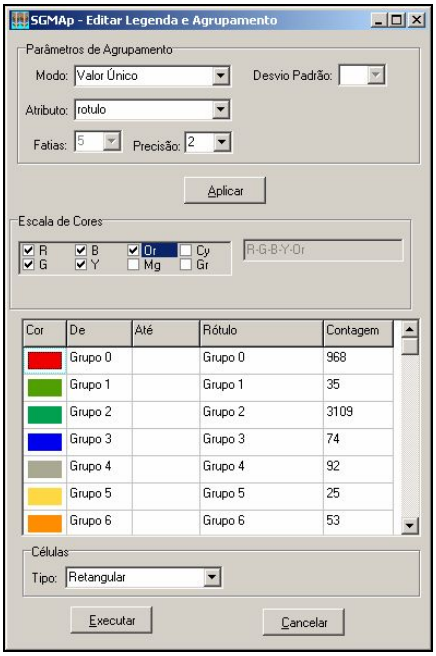

Figura 5.10 - Formação de agrupamento para mapa com n variáveis

Ao confirmar a formação desse agrupamento, o mapa é gerado conforme mostra a Figura 5.11:

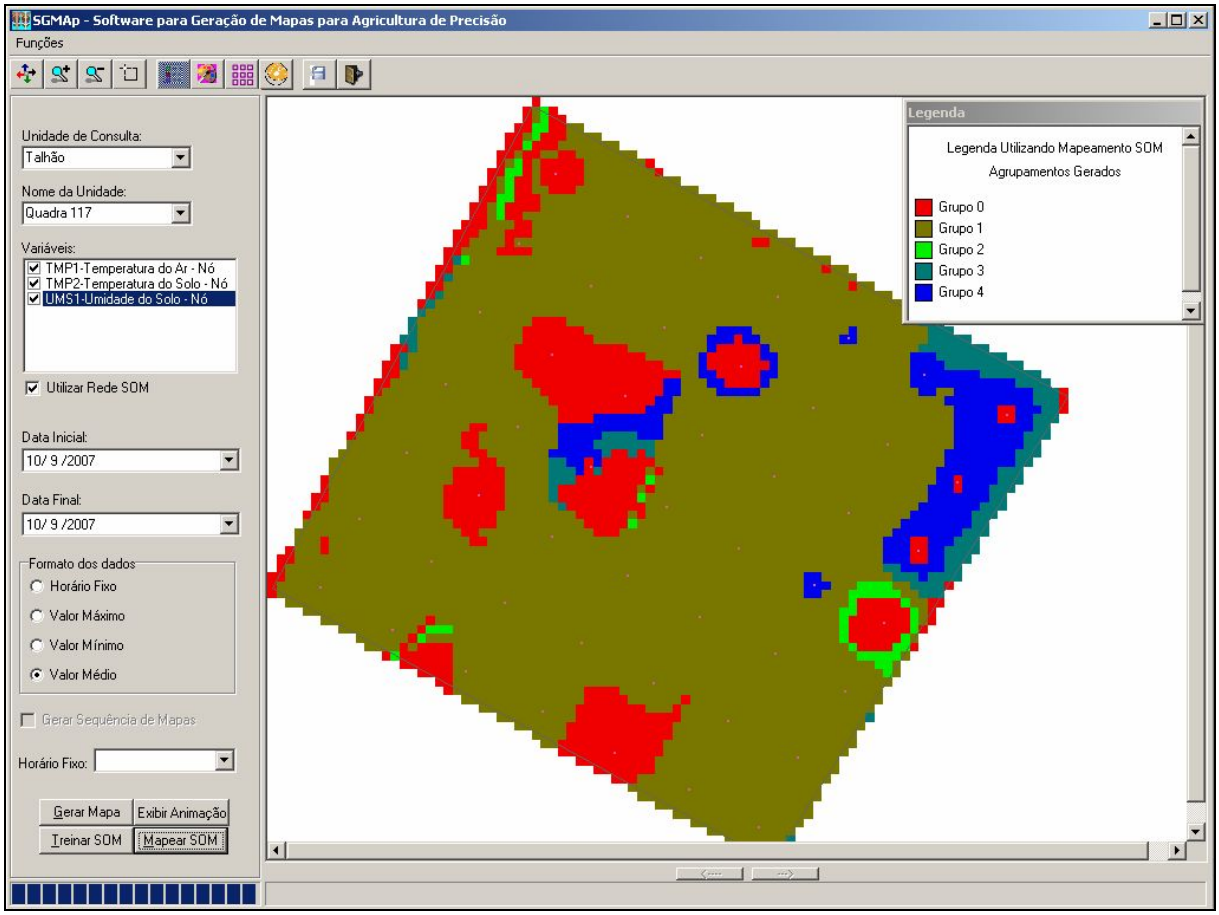

Figura 5.11 – Mapa gerado após a formação do agrupamento

De acordo com a figura acima, pode-se observar cinco agrupamentos diferentes, ou seja, cinco diferentes configurações dentro da área estudada, levando-se em conta dados de umidade e temperatura do solo e temperatura do ar.

5.3.6 Geração de Animações

Com o intuito de facilitar a visualização de uma seqüência de mapas, seja ela horária, diária, semanal, etc., foi criada uma funcionalidade no SGMAp que permite essa visualização

por meio de animações. O botão “Exibir Animação” dá acesso a uma interface que permite essa visualização, levando-se em conta as configurações escolhidas para a última consulta realizada. A Figura 5.12 mostra o formulário referente a essa interface.

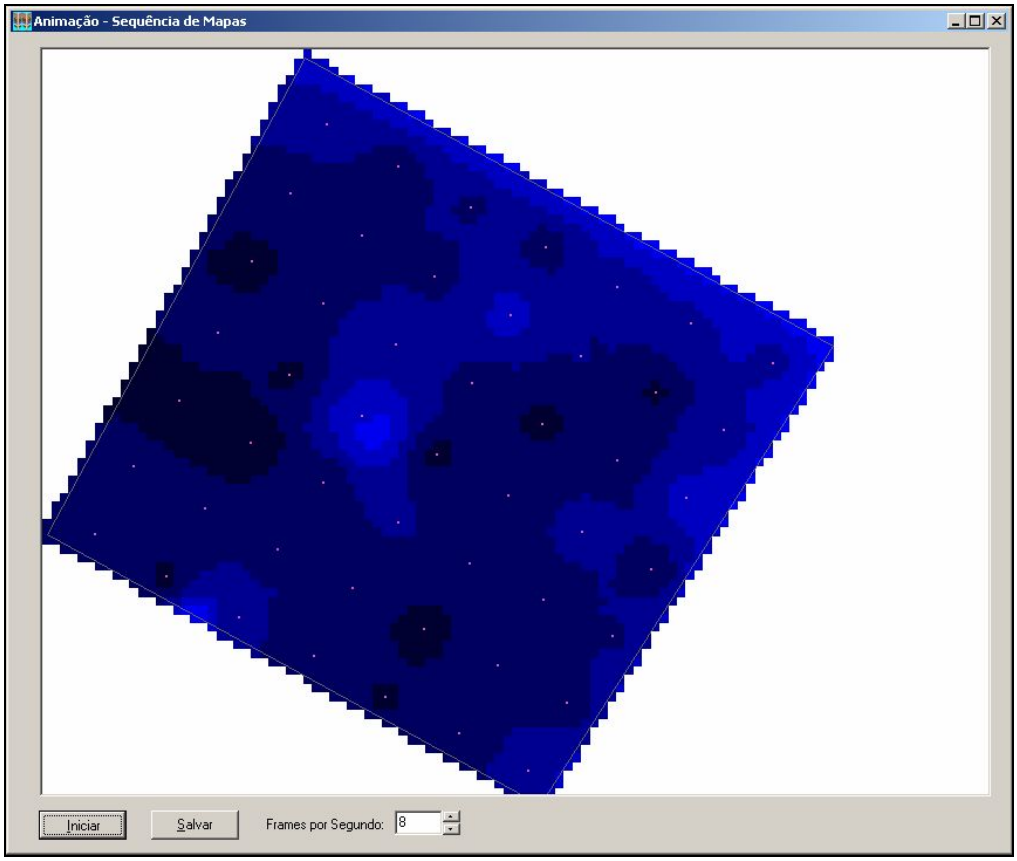

Figura 5.12 – Formulário referente a interface de animação

O botão “Iniciar” inicia a exibição da animação, que pode ser gravada em formato AVI[23] por meio do botão “Salvar”. O usuário pode também determinar a velocidade em que a animação é exibida, escolhendo o número de *frames*[24] por segundo.

Essa funcionalidade pode ser considerada como uma ferramenta importante na tomada de decisão em irrigação de precisão, pois facilita a visualização do comportamento do solo na linha do tempo, permitindo ao produtor rural determinar com maior precisão quais serão as zonas de manejo formadas naquela região.

[23] AVI (*Audio Video Interleave*): trata-se de um formato encapsulador de áudio e vídeo criado pela Microsoft, cuja extensão oficial é .avi.
[24] *Frame*: conjunto de quadros (imagens) que formam um filme por inteiro.

6. Resultados e Conclusões

6.1 Introdução

Esse capítulo descreve os testes realizados com o SGMAp, utilizando dados provenientes da unidade piloto do sistema Irrigap. Além disso, serão exibidos resultados obtidos com a utilização do *software* R, em conjunto com a API aRT e a biblioteca TerraLib, visualizados no *software* TerraView. Também serão relatados alguns problemas encontrados no desenvolvimento do SGMAp, bem como algumas sugestões para melhorias e implementações futuras.

6.2 Conjunto de Dados Utilizado

Para a realização dos testes com o SGMAp, foram utilizados dados coletados pela unidade piloto do sistema Irrigap, durante o período de 24/07 à 18/09/2007. Esses dados são provenientes de 49 nós, com três sensores cada um, capazes de medir umidade e temperatura do solo e temperatura do ar. O intervalo de tempo de leitura dos dados é de uma hora.

Para possibilitar a expansão dos nós da rede, de forma a possibilitar o tratamento do chamado “efeito de borda” pelo SGMAp, foram utilizados nós externos ao limite utilizado pela unidade piloto de estudos. Como a quantidade de nós instalados para realizar esse tipo de tratamento não era suficiente (seria necessária uma grade 9 x 9, ou seja, 81 nós), alguns dados foram simulados. A Figura 6.1 mostra a configuração utilizada para a obtenção dos resultados com o tratamento do “efeito de borda”.

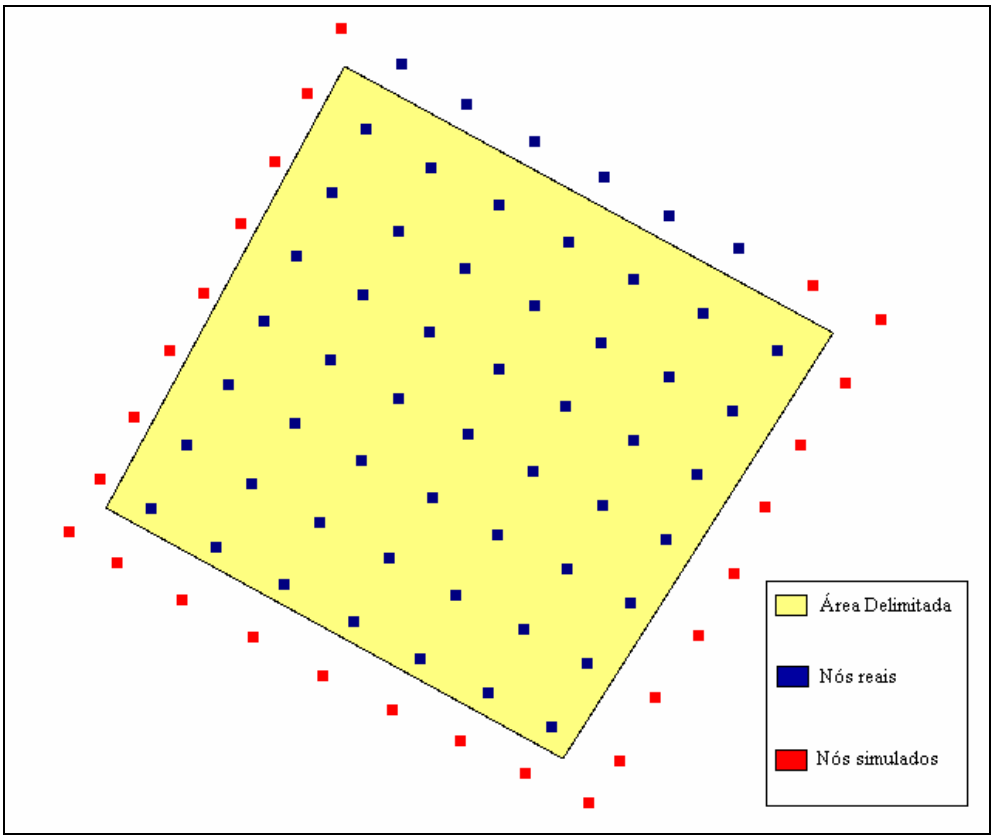

Figura 6.1 – Área delimitada e configuração de nós utilizada

De acordo com a ilustração acima, pode-se observar que foram utilizados 55 nós sensores com obtenção de dados reais e 26 nós sensores com dados simulados de umidade e temperatura do solo e temperatura do ar.

Os dados disponibilizados pelo banco de dados do sistema Irrigap, bem como os que foram simulados, correspondem apenas aos valores medidos naquele exato momento. O SGMAp realiza, dentro dos algoritmos para geração dos mapas temáticos, a modificação desses dados, em memória, para a obtenção de valores mínimos, médios e máximos, nos períodos solicitados. Desse modo, os dados originais são mantidos sem alteração alguma.

Durante a geração de cada mapa, o valor atribuído a cada célula é gravado diretamente no banco de dados, para que, ao término dessa operação, os mesmos possam ser recuperados pelas estruturas de dados da TerraLib, e posteriormente sejam exibidos na tela.

Como o sistema Irrigap encontra-se em desenvolvimento, os dados fornecidos pela unidade piloto podem não refletir exatamente a realidade do solo na região, devido a alguns problemas operacionais encontrados durante a instalação dos nós. No entanto, possibilitam verificar a diferença do comportamento das variáveis em diferentes localidades, mesmo estando estas separadas por poucos metros.

6.3 Resultados e Comparação de Métodos para Geração de Mapas de Umidade do Solo

Os resultados exibidos nessa seção nos mostram o que foi obtido com a utilização dos métodos de interpolação da biblioteca TerraLib. As Figuras 6.1, 6.2, 6.3 nos mostram mapas de umidade do solo, gerados com o mesmo conjunto de dados em uma mesma data, utilizando diferentes métodos de interpolação, todos eles com o devido tratamento do “efeito de borda”.

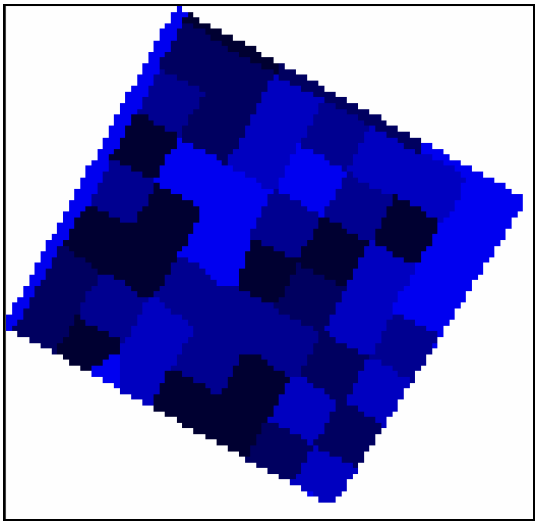

Figura 6.1 - Vizinho Mais Próximo

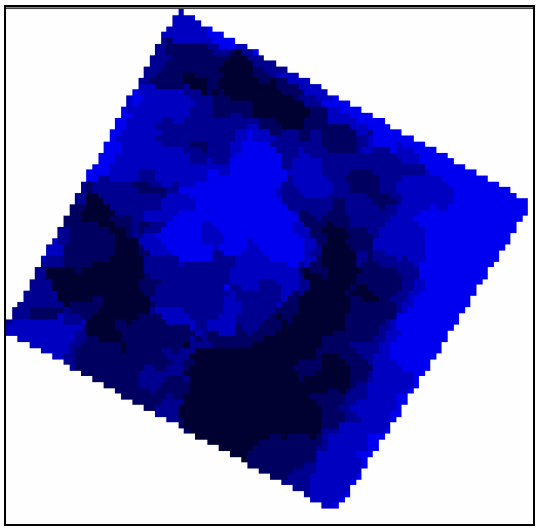

Figura 6.2 - N Vizinhos Mais Próximos

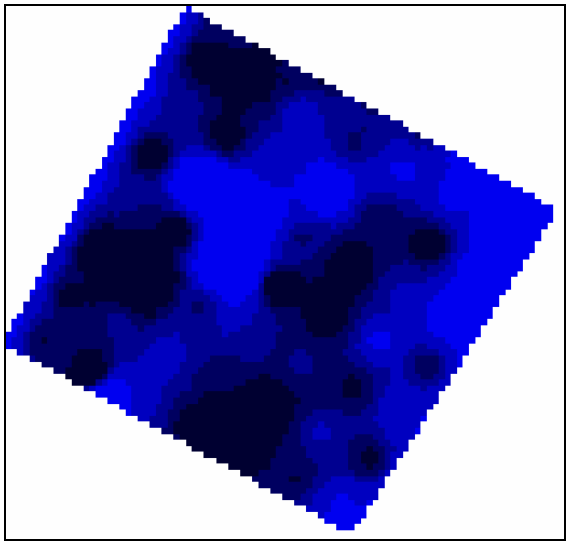

Figura 6.3 - N Vizinhos Mais Próximos Com Distância

Os dados contidos nos mapas acima estão distribuídos da seguinte maneira: quanto mais escuro o tom de azul, maior a umidade do solo no local.

Por meio dessas figuras, pode-se observar que o método utilizado para a geração do mapa da Figura 6.3 apresentou os melhores resultados, pois define melhor as regiões onde o comportamento da umidade do solo é considerado semelhante. Para todos os mapas foi utilizado o método de agrupamento quantitativo (quantil), em cinco fatias com precisão de duas casas decimais. Para o mapa da Figura 6.2 foram utilizados nove vizinhos, e para o mapa da Figura 6.3, nove vizinhos e potência quadrática.

Os outros dois algoritmos de interpolação da TerraLib – Média em Caixa e Média em Caixa com Distância – não forneceram resultados passíveis de exibição nesse contexto.

As Figuras 6.4, 6.5 e 6.6 exibem o mesmo mapa obtido com a utilização do algoritmo que forneceu os melhores resultados na execução anterior, mas nesse momento utilizando três diferentes tipos de agrupamento.

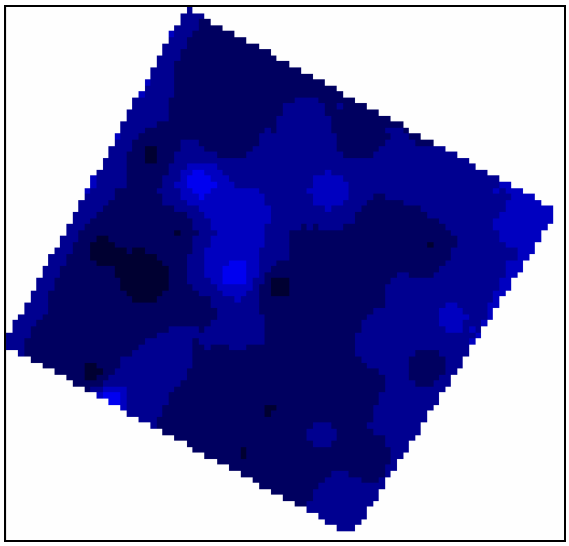

Figura 6.4 - Agrupamento Passos Iguais

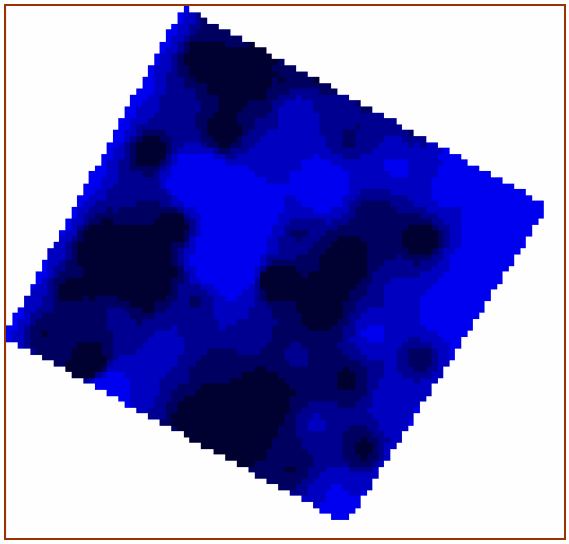

Figura 6.5 - Agrupamento Quantil

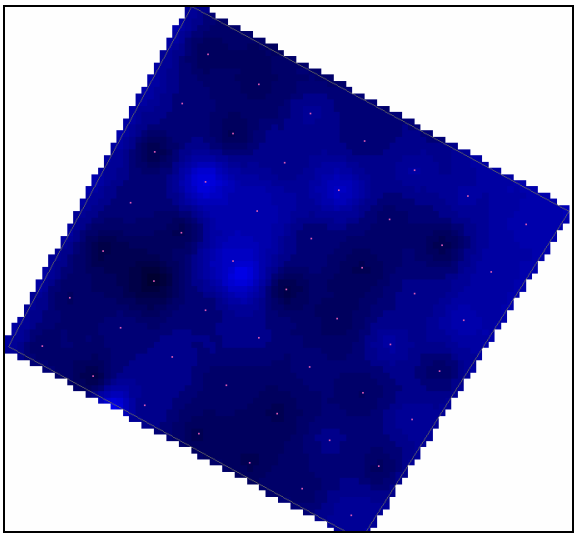

Figura 6.6 - Agrupamento Desvio Padrão

De acordo com essas imagens, o agrupamento das figuras 6.5 e 6.6 apresentaram melhores resultados no agrupamento dos dados, permitindo uma visualização melhor das regiões com o mesmo comportamento com relação à umidade do solo. A diferença entre elas é que o agrupamento por desvio padrão exibe uma transição mais suave entre uma região e outra quando comparado ao agrupamento por quantil. Tanto no agrupamento por passos iguais, na Figura 6.4, quanto o agrupamento por quantil, na Figura 6.5, foi utilizada uma divisão em cinco grupos, com precisão de duas casas decimais. O agrupamento por desvio padrão, na Figura 6.6, foi gerado utilizando-se um desvio padrão de 0,5.

Para que pudesse ser observada a formação de zonas de manejo, seqüências de mapas foram geradas, utilizando diferentes formatos de agrupamento. A Figura 6.7 mostra uma seqüência de mapas de cinco dias diferentes, levando-se em conta o valor médio de cada dia da seqüência, e um agrupamento por quantil.

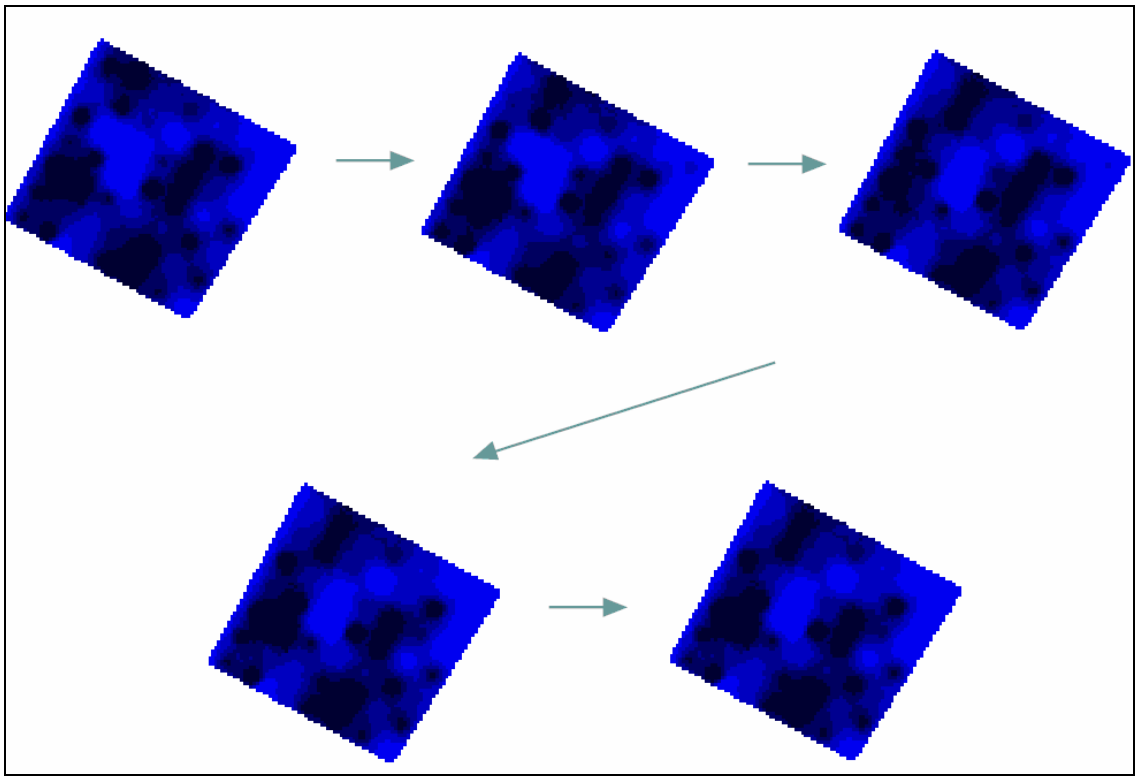

Figura 6.7 – Seqüência de Mapas de Umidade do Solo com Agrupamento Quantil

Pela figura acima, onde as setas indicam a seqüência, com o primeiro dia sendo contemplado pelo mapa na primeira linha, mais à esquerda, e o último dia sendo contemplado pelo mapa na segunda linha, mais à direita, pode-se observar que, dentro dos cinco dias, o comportamento do solo quanto à umidade se manteve parecido. Não se podem observar diferenças significativas entre o primeiro e o último mapa, facilitando a determinação das zonas de manejo. O mesmo pode ser observado na Figura 6.8, com uma seqüência de mapas gerados da mesma maneira que na Figura 6.7, porém utilizando-se o agrupamento com desvio padrão 0,5.

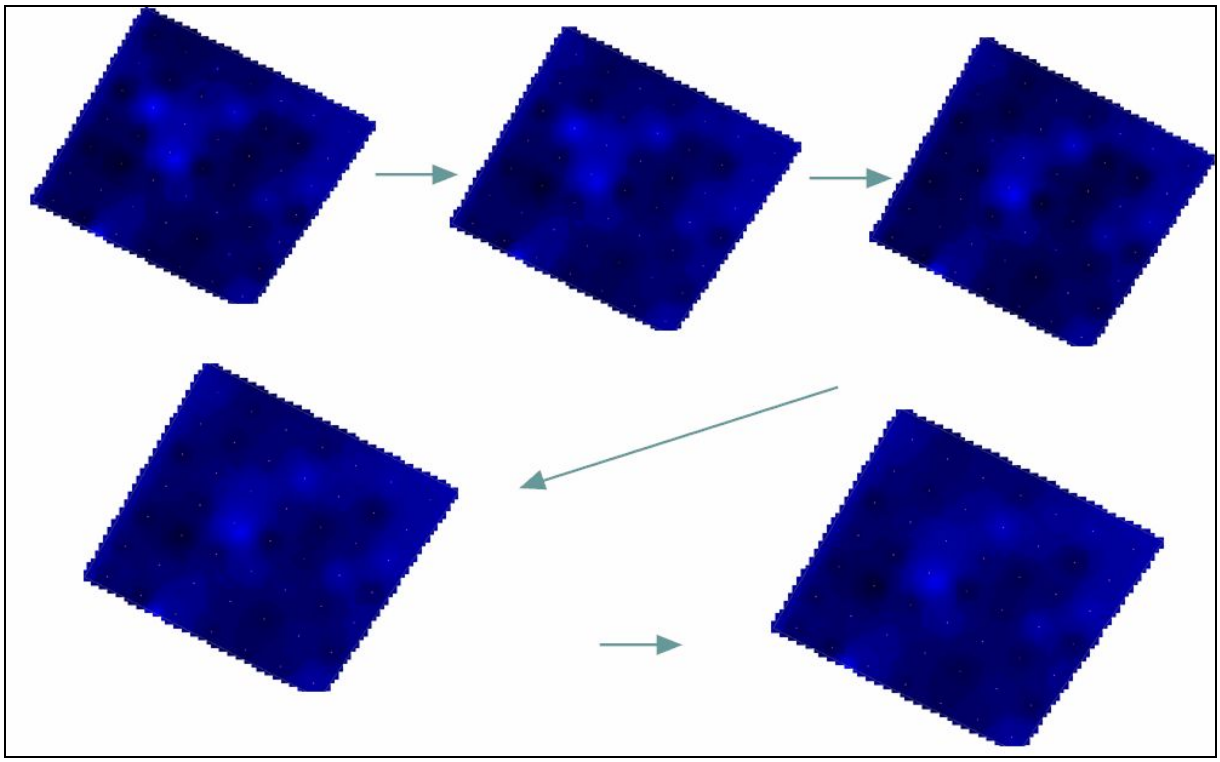

Figura 6.8 – Seqüência de Mapas de Umidade do Solo com Agrupamento Desvio Padrão

Para efeito comparativo, foi utilizado, por intermédio do *software* R em conjunto com a aRT e a TerraLib, o algoritmo da krigagem. A geração de mapas dessa maneira exige muitos passos manuais, tornando o processo bastante trabalhoso e sujeito a erros. Para tanto, foram gerados três arquivos de *scripts* para facilitar essa operação, que podem ser executados utilizando-se a interface do *software* R, a RGui.

O primeiro *script* possui a função de carregar o pacote aRT no ambiente, e criar uma conexão com o banco de dados. A Figura 6.9 exibe a seqüência de comandos executados.

```
require(aRT);
con = openConn(host="localhost", user="root", pass="root", port=3306);
db = openDb(con, "irrigap_terralib_final", update="TRUE");
```

Figura 6.9 – Comandos para inicialização da aRT

O primeiro comando carrega o pacote aRT no R, e o segundo e terceiro comandos são responsáveis por criar e abrir uma conexão com o banco de dados TerraLib, respectivamente.

Já o segundo *script*, carrega os temas e *layers* que serão necessários para a execução do algoritmo. A Figura 6.10 exibe esses comandos.

```
layerContorno = openLayer(db, "talhao_teste_pol");
temaContorno = openTheme(layerContorno, "talhao_teste_pol_1");
layerPontos = openLayer(db, "talhao_point");
temaPontos = openTheme(layerPontos, "talhao_point");
```

Figura 6.10 – Comandos para carregar temas e *layers*

A primeira *layer* e o primeiro tema carregados correspondem ao contorno da área a ser analisada. No caso desse trabalho, corresponde ao polígono que delimita o talhão onde estão localizados os sensores.

A segunda *layer* e o segundo tema carregados correspondem aos pontos de localização dos sensores, além de possuírem valores de umidade do solo ou qualquer outra variável coletados em cada ponto.

Finalmente, o terceiro *script* é o responsável por executar os comandos necessários para a krigagem.

```
require(geoR);
dataPontos = getData(temaPontos);
geoDataPontos = as.geodata(dataPontos, data.col=4);
dataContorno = getData(temaContorno);
poligono = dataContorno[1,];
tmp = slot(poligono, 'polygons');
sub.tmp =  slot(tmp[[1]],'Polygons');
bordas = sub.tmp[[1]]@coords;

ml = likfit(geoDataPontos, trend="1st", ini=c(0.005, 0.001));
x1 = min(bordas[,1]);
x2 = max(bordas[,1]);
y1 = min(bordas[,2]);
y2 = max(bordas[,2]);
gx = seq(x1,x2, by=0.0005);
gy = seq(y1,y2, by=0.0005);
loc0 = expand.grid(gx,gy);
KC = krige.control(trend.d="1st", trend.l="1st", obj=ml);
kc = krige.conv(geoDataPontos, loc=loc0, krige= KC, bor=bordas);
georpred = .prepare.graph.kriging(locations = loc0,borders = bordas, values = kc$pred);
names(georpred)[3] = "z";
addRaster(lraster, georpred);
```

Figura 6.11 – Comandos a serem executados para a krigagem

Primeiramente, é carregado o pacote geoR, que possui a implementação do algoritmo da krigagem. Na seqüência, da linha 2 até a linha 7, os dados são transformados, de forma que o algoritmo utilizado possa processa-los. Ao final do processamento, o mapa final é salvo em

uma *layer* da TerraLib para facilitar sua visualização.

A Figura 6.12 exibe um mapa de umidade do solo gerado utilizando-se o algoritmo da krigagem do pacote geoR. O mapa é exibido utilizando-se o software TerraView.

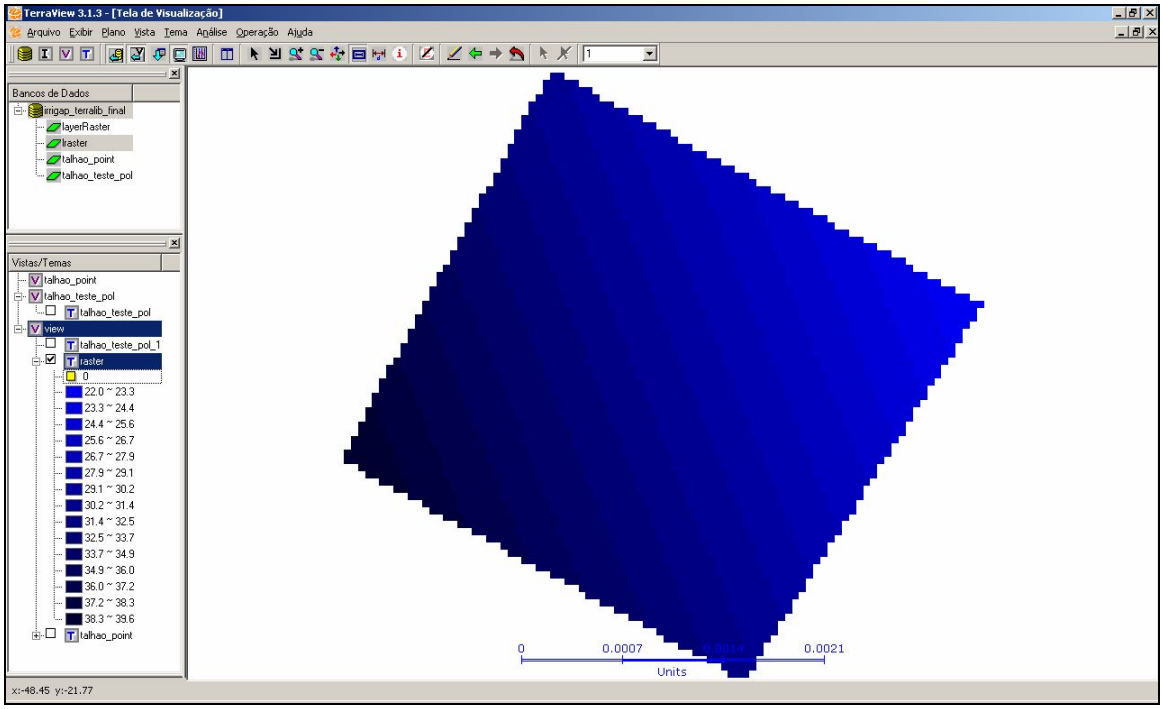

Figura 6.12 – Mapa Gerado utilizando Krigagem

Pode-se observar, por meio desse mapa, que o resultado utilizando-se a krigagem não possibilitou uma boa identificação de agrupamentos, como ocorreu com os algoritmos da TerraLib. No entanto, o algoritmo da krigagem exige um estudo mais detalhado para que os seus parâmetros sejam configurados corretamente, possibilitando a obtenção de resultados até mesmo melhores do que os obtidos com os algoritmos da TerraLib.

6.4 Resultados Obtidos Utilizando Agrupamento pela SOMCode

Essa seção descreve os resultados obtidos com a utilização da biblioteca SOMCode para a formação dos agrupamentos de dados. Para a geração de mapas desse tipo, é necessário que sejam utilizadas duas ou mais variáveis. A Figura 6.13 exibe o Mapa de Kohonen gerado utilizando-se como entrada da rede os dados médios de uma determinada data das variáveis umidade e temperatura do solo e temperatura do ar, previamente interpolados pelo algoritmo de interpolação definido como padrão pelo SGMAp, com tratamento do "efeito de borda" realizado.

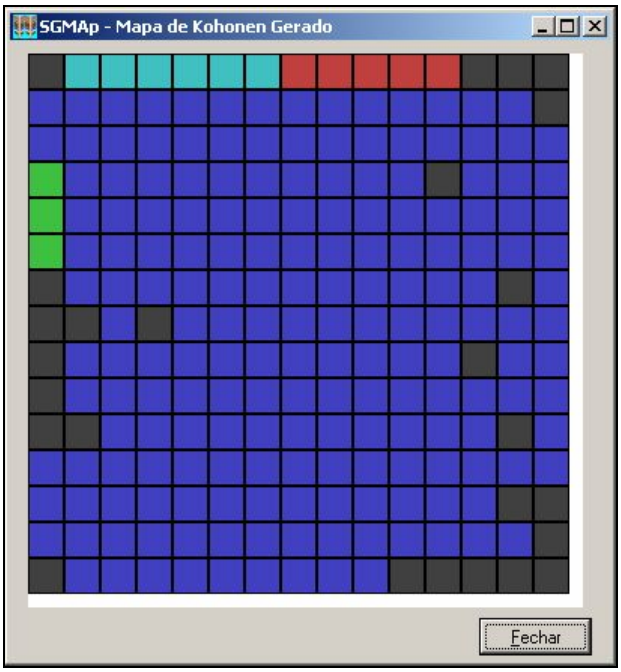

Figura 6.13 – Mapa de Kohonen gerado (15 x 15)

O mapa da figura acima foi gerado após a execução dos algoritmos de treinamento e mapeamento da Rede SOM, com os parâmetros pré-definidos de arranjo espacial do tipo "retangular" e tipo de vizinhança "gaussiana", que são os mais utilizados na maioria das aplicações. O número de iterações utilizado no treinamento foi 500, com taxa de aprendizado 0,5, raio de vizinhança 1 e tipo de aprendizado "em lote".

Ainda nessa figura, pode ser observada a existência de cinco agrupamentos formados pela Rede SOM. O mapa georreferenciado resultante da aplicação da Rede SOM para o agrupamento das regiões semelhantes é exibido na Figura 6.14, abaixo:

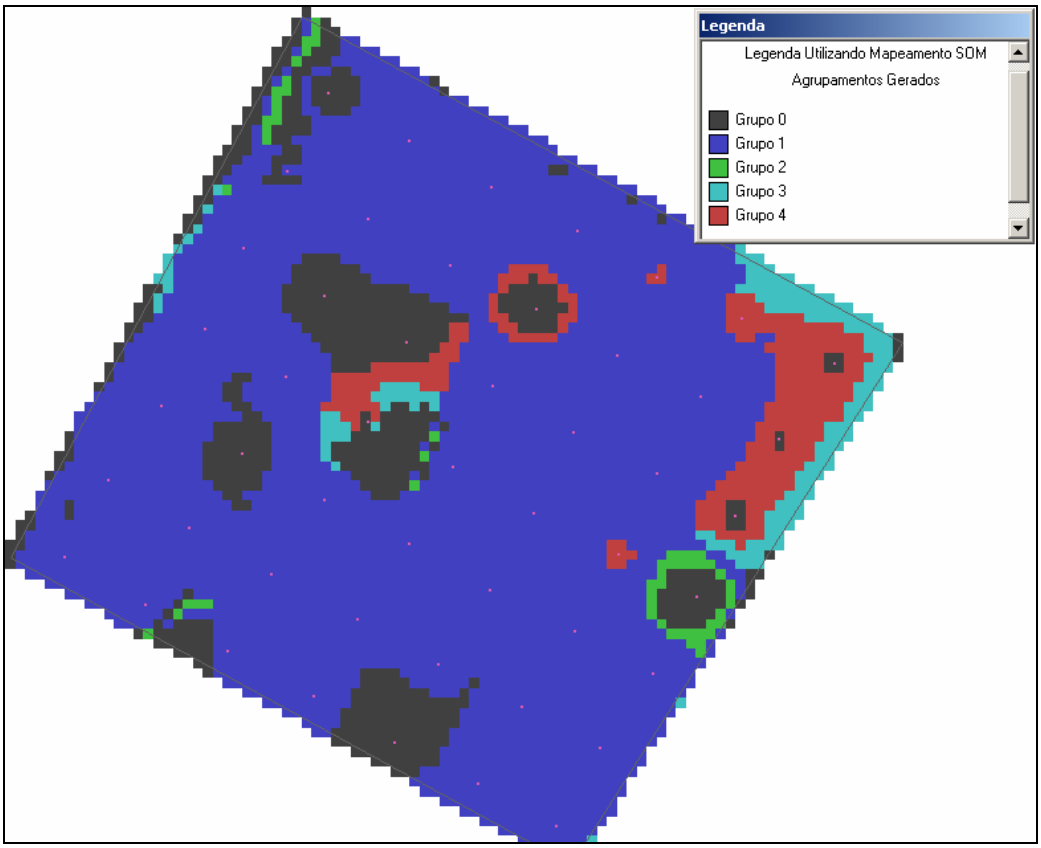

Figura 6.14 – Mapa Georreferenciado utilizando SOMCode

Por meio da legenda, exibida no canto superior direito da figura acima, pode-se observar que os cinco agrupamentos identificados pelos algoritmos da SOMCode aparecem no mapa georreferenciado. Pode ser observado ainda que existe uma grande predominância de amostras pertencentes ao chamado “Grupo 1”, seguida por algumas regiões menores identificadas como pertencentes ao “Grupo 0” e ao “Grupo 2”. As cores utilizadas para representar os agrupamentos são as mesmas nas figuras 6.13 e 6.14, sendo associadas inclusive aos mesmos agrupamentos.

A Figura 6.15 exibe um mapa gerado com as mesmas configurações utilizadas na figura anterior, mas acrescentando como dados de entrada as coordenadas geográficas centrais de cada amostra de célula e a altitude de cada célula, obtida por interpolação a partir dos pontos georreferenciados pertencentes aos nós da rede de sensores.

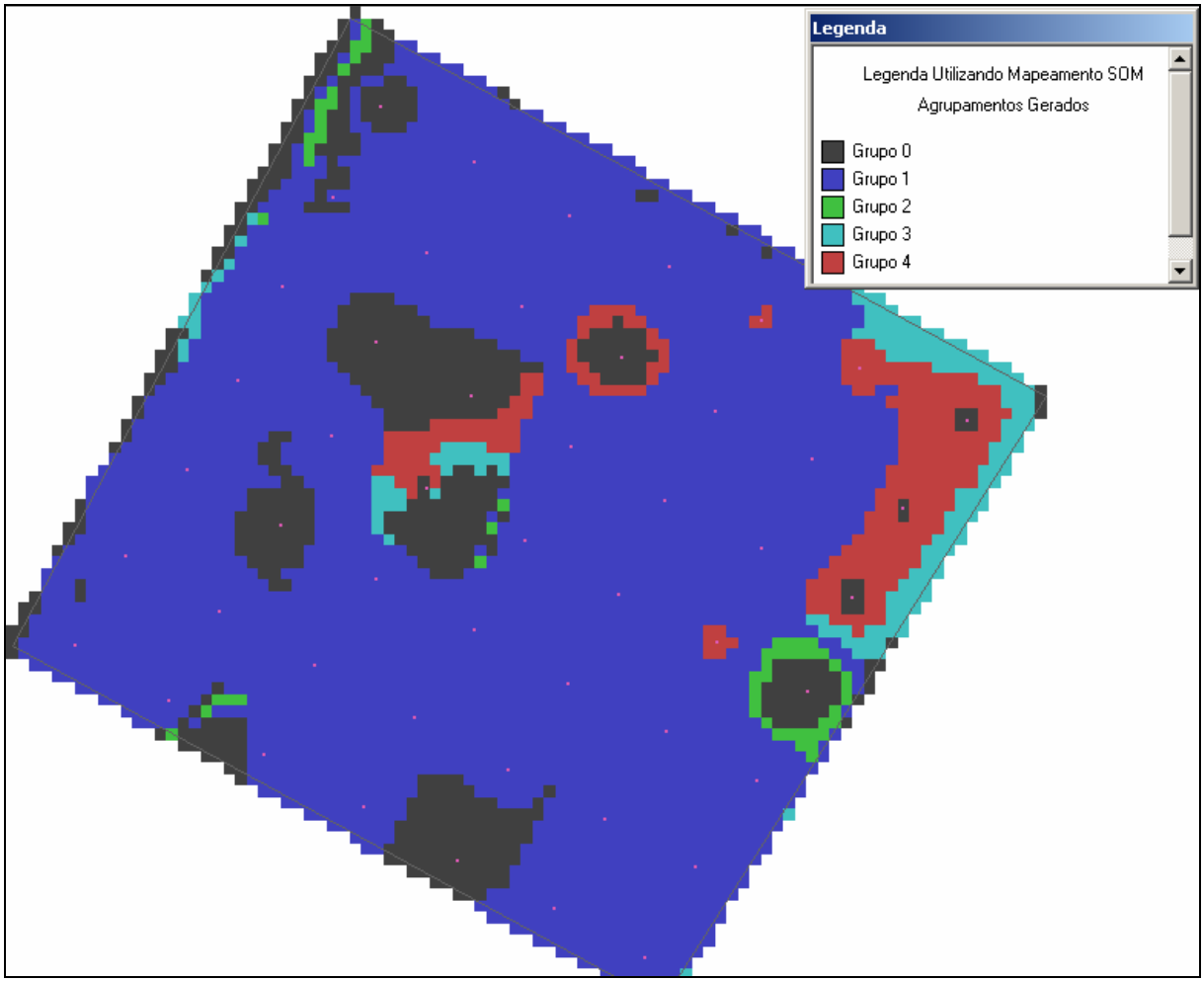

Figura 6.15 – Mapa gerado utilizando a SOMCode com dados de localização

De acordo com a figura acima, e comparando-a com a Figura 6.14, pode-se notar que a inclusão das coordenadas geográficas e dos valores de altitude não influenciou na formação dos agrupamentos, e o mapa de Kohonen gerado foi exatamente o mesmo da Figura 6.13. Um dos motivos para que isso tenha acontecido pode estar relacionado ao fato de os pontos de medida serem muito próximos uns dos outros, fazendo com que as coordenadas geográficas sejam diferentes apenas a partir da quarta ou quinta casa decimal. Além disso, os valores de

altitude disponíveis, mesmo após sofrerem uma normalização, possuíam variação pequena entre os pontos, partindo do valor mínimo de 568 m até o valor máximo de 583 m.

Nas duas figuras (6.14 e 6.15), o agrupamento foi gerado levando-se em conta o rótulo gravado em cada célula, obtido por meio da execução do algoritmo configurado na SOMCode.

Para que fossem verificadas as diferenças no resultado realizando-se alterações dos parâmetros pré-definidos da Rede SOM, foram obtidos mapas georreferenciados variando-se o parâmetro de maior influência no treinamento e mapeamento da mesma: a dimensão do mapa. A Figura 6.17 mostra o resultado com as mesmas entradas utilizadas na Figura 6.14, mas utilizando-se um mapa de Kohonen de dimensões 30 x 30, exibo na Figura 6.16.

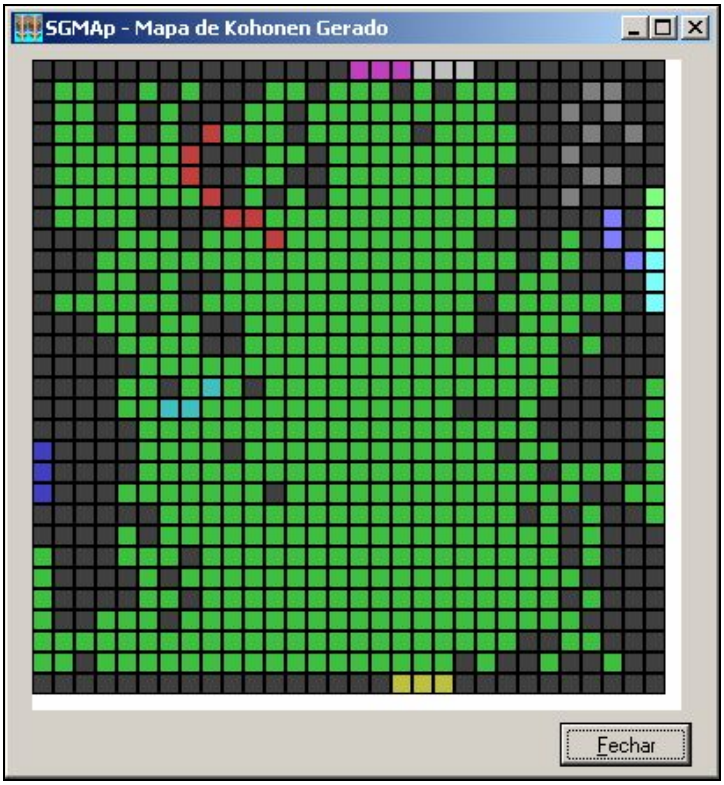

Figura 6.16 – Mapa de Kohonen Gerado (30 x 30)

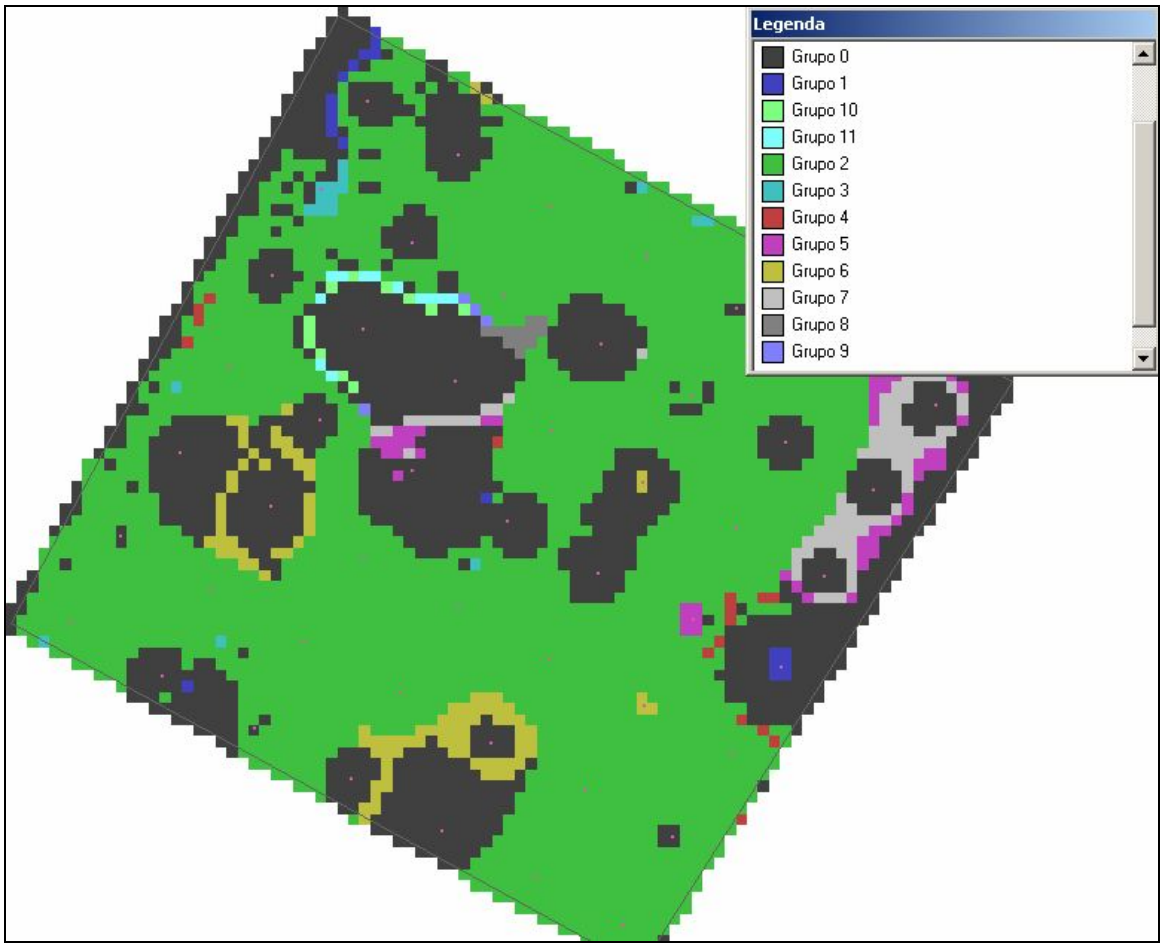

Figura 6.17 – Mapa georreferenciado utilizando SOMCode

Por meio da figura acima, é possível observar que um Mapa de Kohonen de dimensões maiores permite que sejam visualizados mais detalhes no mapa georreferenciado resultante, com a formação de um número maior de agrupamentos (11 nesse caso). No entanto, como no mapa da Figura 6.15, dois agrupamentos permaneceram como predominantes.

Assim, se desejarmos obter uma visão bastante aprofundada da região estudada, a fim de realizar uma tomada de decisão com maior precisão, devemos utilizar Mapas de Kohonen de maior dimensão. Por outro lado, se o resultado for bastante parecido utilizando-se diferentes dimensões, não influenciando no resultado final no contexto aplicado (no caso, Irrigação de Precisão), é mais conveniente utilizar-se o mapa com menores dimensões, que oferece um resultado satisfatório com um custo computacional reduzido.

O último teste realizado com a Rede SOM foi a utilização apenas das variáveis de

umidade do solo e da altitude interpolada. O Mapa de Kohonen, de dimensões 15 x 15, obtido após a execução do treinamento e mapeamento SOM é mostrado na figura 6.18.

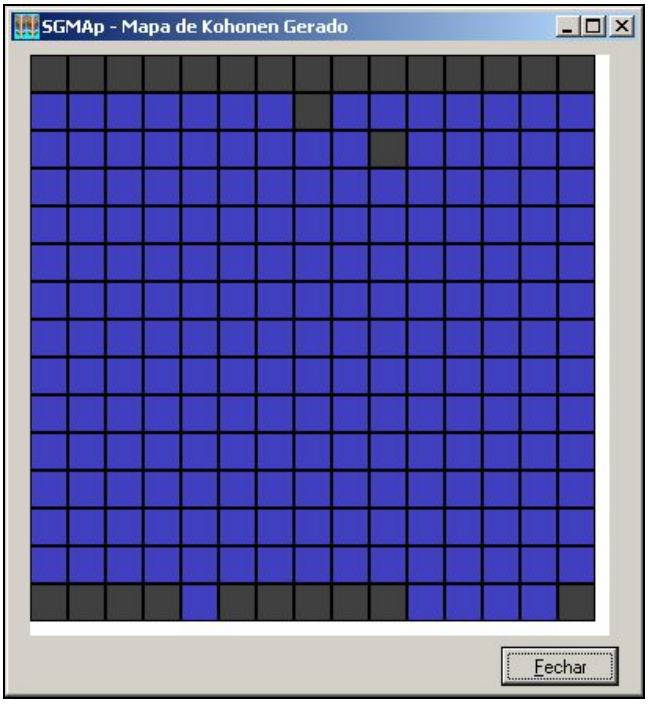

Figura 6.18 – Mapa de Kohonen gerado (15 x 15)

Por meio desse mapa, pode-se observar a formação de apenas 2 agrupamentos. O mapa georreferenciado da Figura 6.19 mostra o resultado final obtido.

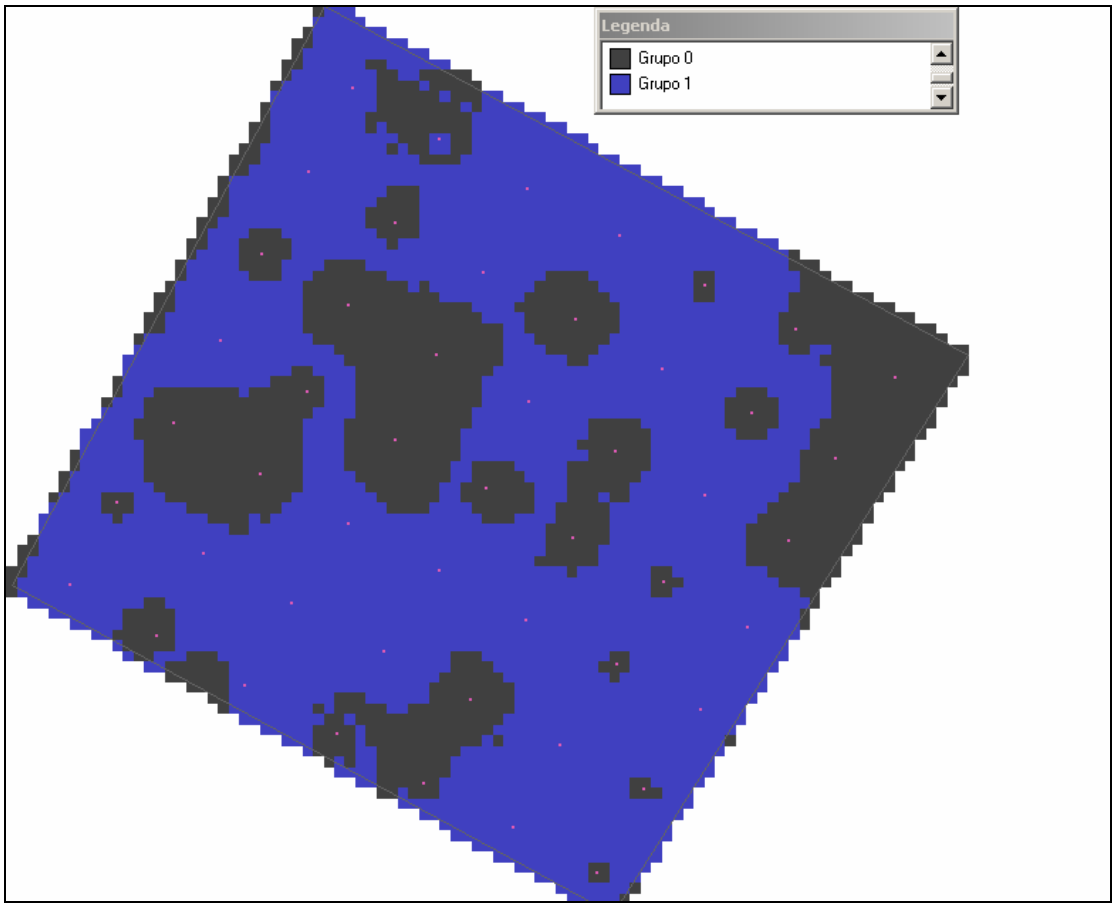

Figura 6.19 – Mapa georreferenciado utilizando umidade do solo e altitude

Por meio desse mapa pode-se observar, em comparação com os mapas de umidade do solo gerados, que os dados de altitude foram úteis para limitar a tomada de decisão em apenas dois agrupamentos distintos.

6.5 Conclusões

O trabalho realizado possibilitou verificar que existe realmente uma tendência de agrupamento de regiões que possuem características semelhantes com relação ao comportamento de variáveis do solo e climatológicas.

Quando obtemos mapas utilizando apenas uma variável, como a umidade do solo,

esses agrupamentos se tornam bastante evidentes, principalmente com a visualização de seqüências de mapas durante um determinado período, pois nessas seqüências as diferenças entre a configuração dos mapas inicial e final são pequenas.

Já com mapas utilizando varias variáveis, agrupados por meio da execução do algoritmo de Kohonen, pode ser observado que a localização dos pontos não influencia no resultado final, principalmente quando os pontos de coleta de dados estão muito próximos uns dos outros. A utilização de valores de altitude interpolados também não influenciou no resultado final se levarmos em conta a área utilizada para o estudo e os mapas constituídos por três variáveis, pois a variação de altitude no local também é pequena. No entanto, a utilização dos dados de altitude em conjunto apenas com os dados de umidade do solo possibilitou a obtenção de apenas duas regiões com comportamentos distintos. Em resumo, pode ser observado que existe uma diferença grande na formação de agrupamentos, quando comparamos mapas utilizando apenas uma variável com mapas utilizando várias variáveis.

Finalmente, pôde ser constatado na análise dos resultados, que o aumento das dimensões do Mapa de Kohonen utilizado proporciona um aumento no nível de detalhes exibidos no mapa georreferenciado resultante. Entretanto, se esse nível de detalhes não influenciar na tomada de decisão, deve ser utilizado um mapa com dimensões menores, visando sempre à obtenção de resultados satisfatórios com o menor esforço computacional possível.

Com relação às ferramentas utilizadas para a geração dos mapas, ficou constatado que a aRT ainda precisa de melhorias para possibilitar a obtenção de resultados satisfatórios, visto que a mesma ainda é muito dependente dos algoritmos de interpolação disponíveis no ambiente R, que são de difícil utilização e compreensão do seu funcionamento.

Quanto à SOMCode, os algoritmos de treinamento e mapeamento da Rede de Kohonen se mostraram bastante eficientes, fornecendo resultados bastantes confiáveis. A

integração com a biblioteca TerraLib foi de fundamental importância para suportar a leitura e persistência dos dados.

A biblioteca TerraLib foi de fundamental importância no desenvolvimento do SGMAp, possibilitando o rápido desenvolvimento das interfaces de visualização, geração de agrupamentos e legendas e interpolação. Os algoritmos de interpolação forneceram resultados satisfatórios, e, se futuras implementações contemplarem os algoritmos geoestatísticos mais importantes executados pelo ambiente R, a mesma se tornará uma biblioteca mais completa para o desenvolvimento de Sistemas de Informações Geográficas. No entanto, alguns pontos negativos na arquitetura da TerraLib são passíveis de revisão, como algoritmos para geração de mapas com persistência direta em banco de dados, ou seja, sem armazenamento em memória principal, tornando a execução dos mesmos mais lenta e impossibilitando acesso concorrente.

Com relação aos objetivos esperados com esse trabalho, o mesmo mostrou que ferramentas como o SGMAp, geradas a partir de bibliotecas já consolidadas como a TerraLib, podem se tornar importantes para análise e tomada de decisão em agricultura de precisão, proporcionando, por exemplo, a economia de água e energia durante o período de irrigação.

7. Referências Bibliográficas

AMBIENTE R. **The R Project for Statistical Computing**. Disponível em: <http://www.r-project.org>. Acesso em: jan. 2008.

ART. **aRT – R TerraLib API**. Disponível em: <http://leg.ufpr.br/aRT>. Acesso em: jan. 2008.

BATCHELOR, B; WHIGHAM, K; DEWITT, J. ,et.al. **Precision Agriculture: Introducion to Precision Agriculture.** Iowa Cooperative Extension, 1997. 4p. Disponível em < http://www.extension.iastate.edu/Publications/PM1703.pdf>. Acesso em jun. 2008.

BLAKE, C.L.; MERZ, C.J. **UCI repository of machine learning databases**, Department of Information and Computer Science, University of California at Irvine, CA, 1998. Disponível em: <http://www.ics.uci.edu/~mlearn/MLRepository>.

BOLFE, E.L; GOMES, J.B.V**. Geoestatísitca como subsídio à implantação de Agricultura de Precisão**. Ruralsoft.com Soluções em tecnologia para pecuária. Disponível em: <http://www.agronline.com.br/artigos/artigo.php?id=210>. Aceso em: mar. 2008.

BORLAND. **Borland Software Corporation**. Disponível em: <http://www.borland.com>. Acesso em: abr. 2008

CÂMARA, G. **Modelos, linguagens e arquiteturas para banco de dados geográficos**. 1995. Tese de Doutorado em Geoprocessamento – Instituto Nacional de Pesquisas Espaciais, 1995.

CÂMARA, G.; SOUZA, R.C.M.; PEDROSA, B.; VINHAS, L.; MONTEIRO, A.M.; PAIVA, J.A.; CARVALHO, M.T.; GATTASS, M. **TerraLib: technology in support of GIS innovation.** In: II Simpósio Brasileiro em Geoinformatica, GeoInfo2000, 2000, São Paulo.

CASANOVA, M.; CÂMARA, G.; DAVIS, C.; VINHAS, L.; QUEIROZ, G. R. **Bancos de Dados Geográficos**. Curitiba: Editora Mundo Geo, 2005, ed.1, v. 1. 504 p.

COELHO, E.C. **Influência da densidade amostral e do tipo de interpolador em mapas temáticos.** 2006. 205 f. Dissertação (Mestrado em Engenharia Agrícola) –

Centro de Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste do Paraná,Cascavel-PR, 2006.

CRESSIE, N.A. **Statistics for special data.** New York: John Wiley, 1991. 900 p.

DAVIS, C.; CÂMARA, G. Arquitetura de sistemas de informação geográfica. In: CÂMARA, G.; DAVIS, C.; MONTEIRO, A.M.V. **Introdução à Ciência da Geoinformação.** São José dos Campos: INPE, 2001, cap.3.

ESMAT, A.K; SOURELL, H; KRAUSE, R. **Precision Irrigation: new strategy irrigation water management.** In: Conference on International Agricultural Research for Development. Deutscher Tropentag, 2002. Witzenhausen, 2002.

FACCAMP. **Orientação a Objetos – Resumo das Principais Características do Conceito de Orientação a Objetos**. Disponível em: <http://www.faccamp.br/~andre/download/eng_oo_res.pdf>. Acesso em abr. 2008.

FARIAS, P.R.S.; NOCITI, L.A.S; BARBOSA, J.C.; PERECIN, D. Agricultura de Precisão: mapeamento da produtividade em pomares cítricos usando geoestatística. **Revista Brasileira de Fruticultura**, v. 25, n.2, p.235-241, 2003.

FOOK, K. D**. Integração da estatística espacial em ambientes GIS**. 2005, 40p. Monografia de Qualificação (Pós-Graduação em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, 2005.

GENTIL, L.V; FERREIRA, S.M. Agricultura de precisão: prepare-se para o futuro, mas com os pés no chão. **Revista A Granja**, Porto Alegre, n 610, 1999, p.12-17.

GUIMARAES, C.C. **Fundamentos de Banco de Dados – Modelagem, Projeto e Linguagem SQL**. Campinas: Unicamp, 2006, ed. 1, v.1.

HADDAD, M. **A Origem da Orientação a Objetos**. Disponível em: <http://www.matheus.eti.br/node/7>. Acesso em: abr. 2008.

HARA, L.T. **Técnicas de apresentação de dados em geoprocessamento.** 1997, 127p. Dissertação de Mestrado (Pós-Graduação em Sensoriamento Remoto) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, 1997.

HERBARIO. **Site Herbário – Artigos e papers relacionados ao meio ambiente,**

agricultura e preservação ambiental. Disponível em <http://www.herbario.com>. Acesso em jul. 2008

ISAAKS, E.H; SRIVASTAVA; R.M. **Applied Geostatistics.** 1., ed. Oxford: Oxford Universit Press, 1989, 561 p.

JOHANN, M.O., 2004. **Curso de Introdução à Programação C++. Universidade Federal do Rio Grande do Sul**, Instituto de Informática, Centro de Empreendimentos em Informática. Disponível em: <http://www.inf.ufrgs.br/~johann/cpp2004>. Acesso em: fev. 2008.

JOHANNSEN, C.J. 1996. **Overview of precision farming**. Proceedings of Information Ag Conference. Champaing, IL, p.53-54, jul. 1996.

KOHONEN, T. **Self-organizing maps**. EUA: Springer, 2001, 501 p.

KOHONEN, T; HYNNIEN, J; KANGAS, J; LAAKOSEN, J. **SOM PAK: The Self-Organizing Map Program Package**. Helsinki: University of Technology, 1995. (Relatório Técnico, A31).

KRIGE, D.G. 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. **Journal of the Chemical**, Metallurgical and Mining Society of South Africa, dezembro de 1951. Pág. 119-139.

MATSURA, E. E. ; TESTEZLAF, R.; CARDOSO, J.L**. A Importância da irrigação no desenvolvimento do Agronegócio**. In: 1ª Reunión de Coordinación de la Red Riegos del Programa CYTED, 2005, Lima. 1ª Reunión de Coordinación de la Red Riegos del Programa CYTED, 2005.

MIZRAHI, V.V. **Treinamento em Linguagem C++ - Módulo 1**. São Paulo: Makron Books do Brasil Editora Ltda, 1995, ed.1, 300 p.

MOLIN, J.P. Agricultura de Precisão: situação atual e perspectivas. In: FANCELLI, A.L.; DOURADO-NETO, D. **Milho: Estratégias de Manejo Para Alta Produtividade.** Piracicaba:ESALQ/USP, 2003, v.1, p. 89-98.

MOLIN, J.P. **Fatores restritivos à adoção da agriculutura de precisão.** In: Congresso Brasileiro de Soja, 2002, Foz do Iguaçu-PR. II Congresso Brasileiro de Soja. Londrina-PR: Embrapa Soja, 2002, v.180, p. 221-229

MOLIN, J.P. **Tendências da agricultura de precisão no Brasil.** In: Congresso Brasileiro de Agricultura de Precisão. Piracicaba-SP, ESALQ/USP, 2007. Anais, 2004, p1-10.

NAMHEE, K. **Optimal biological carbon sequestration region considered with water availability in North Carolina**. 2008, 46p. Dissertação de Mestrado (Pós-graduação em Administração Ambiental) – Nicholas School of Environment and Earth Sciences of Duke University, Durham, NC, EUA.

NASA. **Nasa WMS**. Disponível em: <http://oneart.jpl.nasa.gov.br?wms.cgi>. Acesso em: abr. 2008.

NATIONAL RESEARCH COUNCIL. **Water conservation, reuse and recycling: proceedings of an Iranian-American workshop** . EUA: National Academies Press, 2005, 279 p.

PIERCE, F.J.; NOWAK, P. **Aspects of Precision Agriculture**. Advances in Agronomy, San Diego, CA, EUA, 1999, vol.67, p. 1-85.

POSTGRESQL. **PostgreSQL, The World's Most Advanced Open Source Database**. Disponível em: <http://www.postgresql.org>. Acesso em abr. 2008.

QUANTUMGIS. **QGIS Community Home Page**. Disponível em: <http://www.qgis.org>.Acesso em: abr. 2008.

SCHIMALSKI, M.B.; LOCH, C.; SIEVERS, R. **Metodologia para obtenção do mapa de produtividade para o milho.** In: Congresso Brasileiro de Cadastro Técnico Multifinalitário, UFSC. Florianópolis-SC. Anais, 2004, v. CD-ROM.

SHARMA, T. **Spatial Interploation Techiniques in GIS.** 1999. Disponível em: <http://www.geog.ubc.ca/courses/geog516/notes/interpolation.html>. Acesso em 04/2008.

SILVA, M. A.; MONTEIRO, A.M.V., CAMARA, G. **SOMCode : Design patterns and generic programming in the implementation of Self-Organizing Maps**. In: III Simpósio Regional de Geoprocessamento e Sensoriamento Remoto, 2006, Aracaju-SE. Anais do III Simpósio Regional de Geoprocessamento e Sensoriamento Remoto. Aracaju-SE: Embrapa Tabuleiros Costeiros, 2006.

SOMCODE. **Biblioteca SOMCode**. Disponível em <http://somcode.cpatc.embrapa.br>. Acesso em jan. 2008.

SPERANZA, E.A.; LOPES, W.C; TORRE-NETO, A. **Ambiente computacional para gerenciamento e controle inteligente da irrigação.** In: Congresso Brasileiro de Engenharia Agrícola, 2007, Bonito-MS. Anais, 2007, v. CD-ROM.

S-PLUS. **Statistical analysis software**. Disponível em: <http://www.insightful.com>. Acesso em: abr. 2008.

SRINIVASAN, A. **Handbook of Precision Agriculture**. Philadelphia, PA, EUA: Haworth Press, 2006, 683 p.

TERRALIB. **Biblioteca TerraLib**. Disponível em: <http://www.terralib.org>. Acesso em: mar. 2008.

TERRAVIEW, 2008. **Aplicativo TerraView.** Disponível em: <http://www.dpi.inpe.br/terraview/index.php>. Acesso em jan. 2008.

THE U.S. HOUSE COMMITTEE IN AGRICULTURE. **Agriculture glossary, 2000**. Disponível em: <http://agriculture.house.gov/info/glossary/p.htm>. Acesso em: abr. 2008.

TORRE-NETO, A.; FERRAREZI, R.A ; RAZERA, D. E. ; SPERANZA, E. A. ; LOPES, W C ; LIMA, T.P.F.S; RABELLO, L. M. ; VAZ, C.M.P. **Wireless sensor network for variable rate irrigation in citrus**. In: FRUIT, NUT AND VEGETABLE PRODUCTION ENGINEERING SYMPOSIUM, 7th - INFORMATION & TECHNOLOGY FOR SUSTAINABLE FRUIT & VEGETABLE PRODUCTION. Montpellier, França, 2005. Anais, 2005. v. CD ROM.

TORRE-NETO, A.; SCHÜELLER, J.K.; HAMAN, D.Z**. Automated system for variable rate microsprinkler irrigation in citrus: A demonstration unit**. In: Proc. Third European Conference on Precision Agriculture. Montpellier, 2001. p.725-730.

TORRE-NETO, A; RODRIGUES, E. L. L.; FERRAREZI, R. A.; SPERANZA, E. A.; OLIVEIRA, A. C. N. **Rede de sensores e atuadores sem fio para irrigação com taxa variável**. In: VIII Simpósio Brasileiro de Automação Inteligente. Florianópolis-SC, 2007. Anais, 2007. v. CD ROM.

TSCHIEDEL, M; FERREIRA, M.F. Introdução à Agricultura de Precisão. **Revista**

Ciência Rural, v.32, n.1, 2002.

VESANTO, J; HIMBERG, J; ALHONIEMI,E.; PARHANKANGAS, J**.** Self-Organizing Map in Matlab: The SOM Toolbox. **Proceeding of the Matlab DSP Conference**, Espoo, Filand, novembro de 1999, p. 35–40.

VIEIRA, S.R. Geoestatística de estudos de variabilidade espacial do solo. In: NOVAIS, R.F.; ALVAREZ, V.H.; SCHAEFER, C.E.G.R.(Ed.). **Tópicos em Ciência do Solo.** Viçosa: Sociedade Brasileira de Ciência do Solo, 2000, v.1, p. 1-54.

APÊNDICE A – Tratamento do “Efeito de Borda”

As Figuras A1 e A2 abaixo mostram, para efeito de ilustração, as diferenças entre mapas de umidade do solo referentes às médias diárias em uma data específica, com a utilização e não utilização do tratamento do “efeito de borda”, respectivamente.

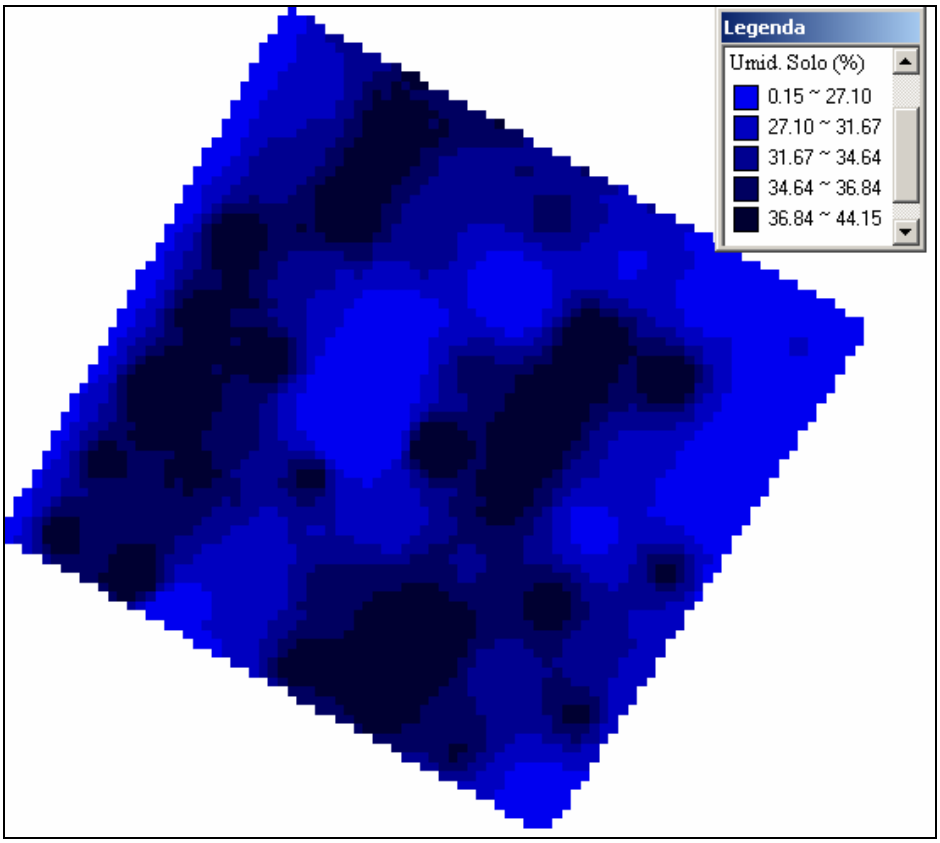

Figura A1 – Mapa de umidade do solo com o tratamento do “efeito de borda”

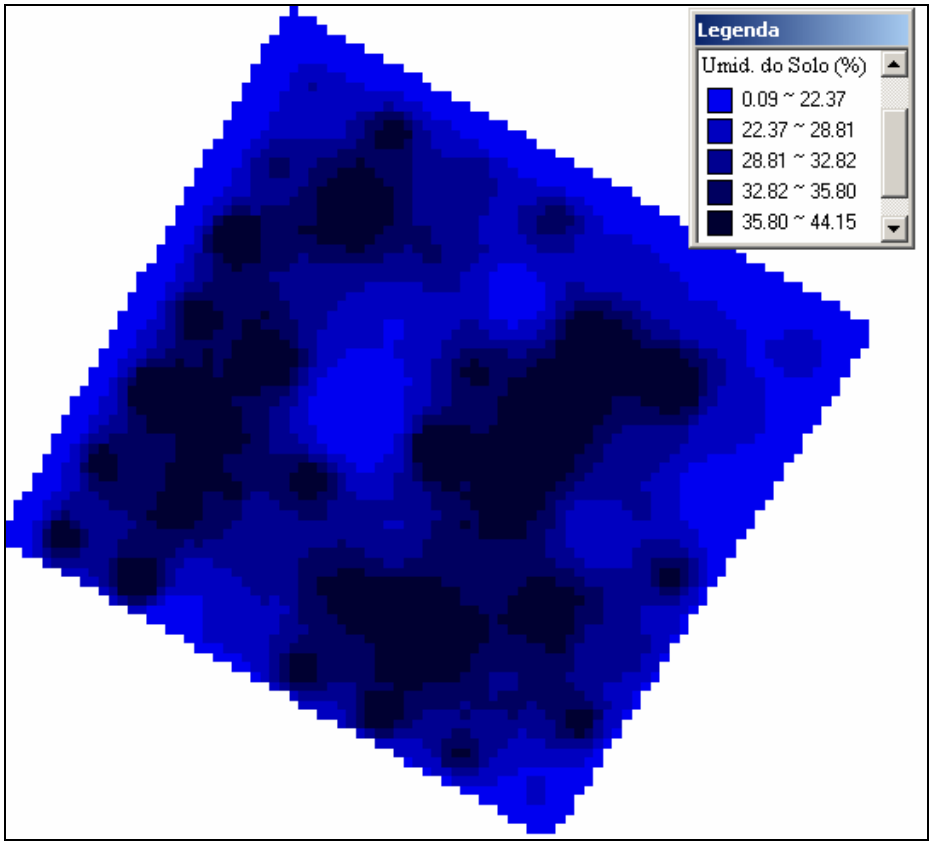

Figura A2 – Mapa de umidade do solo sem o tratamento do “efeito de borda”

APÊNDICE B – Trechos de código fonte em C++

• Conexão com o banco de dados PostGIS utilizando objetos da TerraLib:

```
// Instanciar objeto de conexão
TePostGIS conn_terralib = new TePostGIS();

// Conectar, com parâmetros: host, usuário, senha, banco e porta
conn_terralib->connect("localhost","postgres","postgres","irrigap",5432)
```

• Implementação dos algoritmos de interpolação da biblioteca TerraLib utilizados:

```
// Vizinho mais próximo
double nearestNeighbor(const kdKey& key)
{
	vector<kdDataItem> report;
	vector<double> sqrDists;

	fillNNVector(report, 1);

	kd_.nearestNeighborSearch(key, report, sqrDists, 1);

	if(sqrDists[0] >= TeMAXFLOAT)
		return -TeMAXFLOAT;
	else
		return report[0].value();
}


// Média dos vizihos mais próximos
double avgNearestNeighbor(const kdKey& key, const unsigned int&
numberOfNeighbors)
{
	vector<kdDataItem> report;
	vector<double> sqrDists;

	fillNNVector(report, numberOfNeighbors);

	kd_.nearestNeighborSearch(key,	report,	sqrDists,
numberOfNeighbors);

	double numElements = numberOfNeighbors;

	double sum = 0.0;

	for(unsigned int i = 0; i < numberOfNeighbors; ++i)
	{
		if(sqrDists[i] >= TeMAXFLOAT)
		{
			--numElements;
			continue;
		}
```

```
                        if(sqrDists[i] == 0.0)
                              return report[i].value();

                        sum += report[i].value();
                  }

                  if(numElements > 0.0)
                return sum / numElements;
                  else
                        return -TeMAXFLOAT;
            }


            // Inverso da Distância elevado a uma potência
            double  distWeightAvgNearestNeighbor(const  kdKey&  key,  const
unsigned int& numberOfNeighbors, const int& powValue)
            {
                  vector<kdDataItem> report;
                  vector<double> sqrDists;

                  fillNNVector(report, numberOfNeighbors);

                  kd_.nearestNeighborSearch(key,      report,      sqrDists,
numberOfNeighbors);

                  double num = 0.0;
                  double den = 0.0;

                  if(powValue == 1.0)
                  {
                        for(unsigned int i = 0; i < numberOfNeighbors; ++i)
                        {
                              if(sqrDists[i] >= TeMAXFLOAT)
                                    continue;

                              if(sqrDists[i] == 0.0)
                                    return report[i].value();

                              double wi = 1.0 / (sqrt(sqrDists[i]));

                              num += (wi * report[i].value());
                              den += wi;
                        }
                  }
                  else if(powValue == 2.0)
                  {
                        for(unsigned int i = 0; i < numberOfNeighbors; ++i)
                        {
                              if(sqrDists[i] >= TeMAXFLOAT)
                                    continue;

                              if(sqrDists[i] == 0.0)
                                    return report[i].value();

                              double wi = 1.0 / (sqrDists[i]);

                              num += (wi * report[i].value());
                              den += wi;
                        }
                  }
```

```
                else
                {
                        for(unsigned int i = 0; i < numberOfNeighbors; ++i)
                        {
                                if(sqrDists[i] >= TeMAXFLOAT)
                                        continue;

                                if(sqrDists[i] == 0.0)
                                        return report[i].value();

                                double  wi  =  1.0  /  pow(sqrt(sqrDists[i]),
powValue);

                                num += (wi * report[i].value());
                                den += wi;
                        }
                }

                if(den != 0.0)
              return num / den;
                else
                        return -TeMAXFLOAT;
            }
```

• Implementação do treinamento em lote da RedeSOM pela SOMCode :

```
int BatchLearning::Learning ( NetParams& net ) {

      int i, k, j, i1, i2, bmu, t;
      Value_Type h, r, htot;
      Value_Type  d                 = net.getData()->getDimension();
      int         m                       = net.getMapcode()->getMapSize();
      int         n                       =    net.getData()->getDataSize();

      int         dimension         = net.getMapcode()->getDimension();
      int         lattice                 = net.getMapcode()->getLattice();
      int*  dimensions        = net.getMapcode()->getDimensions();
      int         num_iterations    = net.getNumIterations();
      int         neighboor         = net.getMapcode()->getNeighborType();
      TMatrix entries               = net.getData()->getEntries();
      SOMData * data                = net.getData();
      TMatrix delta                 = net.getDelta();

      TMatrix codebook        = net.getMapcode()->getCodebook();
      TVector vn                    = create_vector(1, m);
      TMatrix S                     = create_matrix( 1, m, 1, (int)d );
      Mapcode * mapcode       = net.getMapcode();

      TBMU  * tmp_bmu = new TBMU;


      r = net.getInitNeighbor();

      for (t=1; t<= num_iterations; t++ ) {
            setActualIteraction( t );

            tmp_bmu = MakeBMU( data, mapcode );
```

```
            // Inicia o vetor de Voronoi e as variáveis SumData
            for (i=1; i<= m; i++) { vn[i] = 0; for (k=1; k<=d; k++) S[i][k]
= 0.0;}

            for (j=1; j<= n; j++ ) {
                  bmu = tmp_bmu[0][j-1][0].i;
                  vn[bmu]++;
                  for (k=1; k<=d; k++) S[bmu][k] += entries[j][k];
            };

            // Preenche mapcode com zeros
            for (i=1; i<=m; i++) for (k=1; k<=d; k++) codebook[i][k] = 0.0;


            for (i1=1; i1<=m; i1++) {
                  htot = 0;
                  for (i2=1; i2 <= m; i2++ ) {

                        h  =  getTopology()->H(  delta[i1][i2],  r,  neighboor
);
                        for   (   k=1;   k<=d;   k++)   codebook[i1][k]   +=
h*S[i2][k];
                        htot += h*vn[i2];
                  }
                  for (k=1; k<= d; k++)
                        if (htot != 0 ) codebook[i1][k] /= htot;
            }
            mapcode->setCodebook( codebook );
            r = radius( r, num_iterations, t);
      }
      net.getMapcode()->setCodebook( codebook );

      free_vector( vn, 1, m );
      free_matrix( S, 1, m, 1, (int)d );
      return 0;

};
```