

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE CIÊNCIAS RURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM GEOMÁTICA**

**AUTOMAÇÃO NO PROCESSO DE VETORIZAÇÃO  
DE ÁREAS DE INTERESSE E *BUFFER* VARIÁVEL  
SOBRE IMAGENS DE SATÉLITE**

**DISSERTAÇÃO DE MESTRADO**

**Éric Tadiello Beltrão**

**Santa Maria, RS, Brasil  
2012**

**AUTOMAÇÃO NO PROCESSO DE VETORIZAÇÃO  
DE ÁREAS DE INTERESSE E *BUFFER* VARIÁVEL SOBRE  
IMAGENS DE SATÉLITE**

por

**Éric Tadiello Beltrão**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Geomática, Área de Concentração em Tecnologia da Geoinformação, da Universidade Federal de Santa Maria (UFSM,RS), como requisito parcial para obtenção do grau de **Mestre em Geomática.**

**Orientador: Prof. Rudiney Soares Pereira**

**Santa Maria, RS, Brasil  
2012**

**Universidade Federal de Santa Maria  
Centro de Ciências Rurais  
Programa de Pós-Graduação em Geomática**

**A Comissão Examinadora, abaixo assinada,  
aprova a Dissertação de Mestrado**

**AUTOMAÇÃO NO PROCESSO DE VETORIZAÇÃO DE ÁREAS DE  
INTERESSE E *BUFFER* VARIÁVEL SOBRE IMAGENS DE SATÉLITE**

elaborada por  
**Éric Tadiello Beltrão**

como requisito parcial para obtenção do grau de  
**Mestre em Geomática**

**COMISSÃO EXAMINADORA:**

---

**Rudiney Soares Pereira, Dr.**  
(Presidente/Orientador)

---

**Gelson Lauro Dal Forno, Dr. (UFSM)**

---

**Daniel Boemo, Dr. (IF-Farroupilha)**

Santa Maria, 02 de agosto de 2012.

## **AGRADECIMENTOS**

Primeiramente, a Deus pela vida e saúde.

A minha esposa e meus familiares, pelo apoio, pela compreensão e a paciência proporcionada no decorrer do curso.

Ao professor Dr. Rudiney Soares Pereira, pela disponibilidade em orientar-me, apresentando opiniões, e discutindo sobre assuntos de grande valia para o desenvolvimento deste trabalho, pela sua paciência e sabedoria que fizeram com que o objetivo dessa dissertação fosse alcançado.

Aos amigos e colegas que me apoiaram de uma ou de outra forma no decorrer desta jornada. A Universidade Federal de Santa Maria (UFSM), funcionários e aos professores do programa de Pós Graduação em Geomática (PPGG) pelos conhecimentos construídos durante esta caminhada.

## EPÍGRAFE

“Nem tão longe que eu não possa ver;  
Nem tão perto que eu possa tocar;  
Nem tão longe que eu não possa crer que  
um dia chego lá;  
Nem tão perto que eu possa acreditar que o  
dia já chegou.”

(Humberto Gessinger)

## RESUMO

Dissertação de Mestrado  
Programa de Pós-Graduação em Geomática  
Universidade Federal de Santa Maria

### **AUTOMAÇÃO NO PROCESSO DE VETORIZAÇÃO DE ÁREAS DE INTERESSE E *BUFFER* VARIÁVEL SOBRE IMAGENS DE SATÉLITE**

Autor: Éric Tadiello Beltrão  
Orientador: Rudiney Soares Pereira  
Santa Maria, 02 de agosto de 2012

O constante aperfeiçoamento nas técnicas de desenvolvimento de aplicativos computacionais, associado ao avanço tecnológico dos dispositivos eletrônicos que suportam tais sistemas, vem contribuindo significativamente com as mais diversas áreas do conhecimento, agregando a estas, muitas vezes, maior precisão e agilidade em seus processos. Com base neste contexto, e voltando nossos olhares para as áreas de Sistema de Informações Geográficas (SIG), e Processamento Digital de Imagens, realizou-se o desenvolvimento de dois programas que servissem como proposta de inovação tecnológica no que diz respeito à vetorização de áreas de interesse sobre imagens de satélite, e criação de *buffer* variável sobre as mesmas, variação esta que segue as características de largura da área analisada. Ambos os programas foram desenvolvidos na linguagem de programação *Python* e tem como produto final um arquivo *shape* contendo a vetorização de todas as áreas encontradas em uma imagem de satélite por meio de classificação ou filtragem desta. Também se apresenta neste trabalho a manipulação dos arquivos gerados, a fim de nortear as ações necessárias para que se utilize, da melhor forma, o produto desses programas, desta maneira auxiliando o usuário dos sistemas computacionais de SIG referente ao desempenho e precisão do seu trabalho.

Palavras-Chave: *python*, automação, geoprocessamento, vetorização, sistema de informações geográficas.

## **ABSTRACT**

Master's Dissertation  
Post-graduate Course in Geomatics  
Universidade Federal de Santa Maria

### **VECTORIZATION PROCESS AUTOMATION IN AREAS DE INTEREST AND BUFFER VARIABLE ON SATELLITE IMAGES**

Author: Éric Tadiello Beltrão  
Advisor: Rudiney Soares Pereira  
Santa Maria, 02 august, 2012

The constant improvement in computer application development techniques, combined with the technological advancement of electronic devices that support such systems, comes with the most contributing significantly different areas of knowledge, adding to these, often, greater accuracy and agility in their processes. Based on this context, and returning our looks for areas of geographic information System (GIS), and Digital image processing, the development of two programs that serve as a proposal of technological innovation as regards the vectorization of areas of interest on satellite images, and creating the buffer variable on the same, this variation that follows the characteristics of width of the area considered. Both programs were developed in the Python programming language and has as final product a shape file containing the vectorization of all areas found in a satellite image by means of classification or filtering. Also presents itself in this work the manipulation of files generated, in order to guide the actions necessary to use, in the best way, the product of these programs, thus helping the user of GIS computer systems for the performance and accuracy of their work.

Keywords: *python*, automation, geoprocessing, vectorization, geographical information system.

## LISTA DE SIGLAS

APP - Área de Preservação Permanente

CAD - *Computer Aided Design*

GIS - *Geographic Information System*

GPL - *General Public License*

IDE - *Integrated Development Environment*

INPE - Instituto Nacional de Pesquisas Espaciais

LIS - *Land Information System*

NCGIA - *National Center for Geographic Information and Analysis*

PDI - Processamento Digital de Imagens

QGIS - Quantum GIS

SAR - *Synthetic Aperture Radar*

SIG - Sistema de Informações Geográficas



## LISTA DE TABELAS

Tabela 1 - Iniciativas e objetivos do NCGIA .....	19
Tabela 2 - Bandas do sensor TM e suas aplicações .....	41

## LISTA DE ILUSTRAÇÕES

Figura 1 - Hierarquia da informação.....	20
Figura 2 - Componentes do SIG.....	22
Figura 3 - Configuração de um SIG.....	23
Figura 4 - Estrutura de dados.....	24
Figura 5 - Equação de uma imagem.....	28
Figura 6 - Esquema de uma imagem digital.....	28
Figura 7 - Sensores e suas resoluções espaciais.....	30
Figura 8 - Vizinhança horizontal e vertical.....	31
Figura 9 - Vizinhança diagonal.....	31
Figura 10 - Vizinhança 6.....	32
Figura 11 - Vizinhança 18.....	32
Figura 12 - Vizinhança 26.....	33
Figura 13 - Processos de interação do código Python.....	36
Figura 14 - Suavização da imagem por meio do Filtro de Mediana.....	38
Figura 15 - Uso do filtro de Canny para reconhecimento de bordas.....	38
Figura 16 - Uso do filtro passa-faixa.....	39
Figura 17 - Recorte de imagem da banda 4 do sensor TM.....	42
Figura 18 - Plugin PyDev do IDE Eclipse, e funcionalidade conclusão de código.....	44
Figura 19 - Código contendo filtro para reconhecimento de bordas.....	46
Figura 20 - Resultado do filtro para realce de bordas.....	46
Figura 21 - Região ampliada da Figura 22.....	47
Figura 22 - Código do filtro passa-faixa.....	48
Figura 23 - Resultado do filtro passa-faixa.....	49
Figura 24 - Rastreamento de delimitação de áreas de interesse.....	51
Figura 25 - Mascara de análise.....	52
Figura 26 - Sequência lógica de teste.....	53
Figura 27 - Imagem criada para testes iniciais.....	54
Figura 28 - Polígono plotado sobre área de interesse.....	55
Figura 29 - Arquivo shape exibido pelo software Quantum GIS.....	60
Figura 30 - Imagem da funcionalidade selecionar feições.....	61
Figura 31 - Representação de lâminas d'água e buffer destas.....	62
Figura 32 - Demonstração do buffer referente ao curso d'água.....	63
Figura 33 - Imagem que representa curso de água.....	64
Figura 34 - Representação do buffer variável.....	65
Figura 35 - Buffer genérico sobre buffer variável.....	66

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	11
<b>1.1 Motivação</b> .....	12
<b>1.2 OBJETIVOS</b> .....	13
1.2.1 Objetivo Geral.....	13
1.2.2 Objetivos Específicos.....	14
<b>2 REVISÃO DE LITERATURA</b> .....	15
<b>2.1 Sistema de Informações Geográficas</b> .....	15
2.1.1 Breve histórico.....	17
2.1.2 Hierarquia da informação.....	19
2.1.3 Componentes de Informática.....	20
2.1.4 Módulos de um SIG.....	22
<b>2.2 Processamento digital de imagens</b> .....	25
2.2.1 Áreas de Aplicação.....	25
2.2.2 Representação da Imagem.....	27
2.2.3 Resolução Espacial.....	29
2.2.4 Relacionamento de Vizinhança.....	30
<b>2.3 Linguagem de programação Python</b> .....	33
2.3.1 Breve histórico.....	34
2.3.2 Compilação e interpretação.....	34
2.3.3 Frameworks.....	36
3.4 Filtros.....	37
<b>3 MATERIAL E MÉTODOS</b> .....	40
<b>3.1 Escolha do material de trabalho</b> .....	40
3.1.1 Imagem de teste.....	40
3.1.2 Linguagem de programação.....	43
3.1.3 Ambiente de desenvolvimento.....	43
<b>3.2 Filtro: Método para obtenção da área de interesse</b> .....	45
3.2.1 Identificando bordas.....	45
3.2.2 Filtro por frequência.....	48
<b>3.3 Desenvolvendo o software</b> .....	50
3.3.1 Máscara de Análise .....	51
3.3.2 Versão Mono e Multi Polígono.....	53
3.3.4 Ajuste de traçado.....	56
3.3.5 Buffer variável.....	56
<b>4 RESULTADOS E DISCUSSÕES</b> .....	58
<b>4.1 Otimização de tempo e precisão</b> .....	58
<b>4.2 Manipulando shapefile com QGIS</b> .....	59
4.2.1 Dividindo shapefiles.....	60
4.2.2 Criando delimitação APP.....	61
<b>5 CONCLUSÕES E RECOMENDAÇÕES</b> .....	67
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	68
<b>ANEXOS</b> .....	71

# 1 INTRODUÇÃO

A cada dia que passa, os sistemas computacionais vem ganhando maior notoriedade em nossas vidas, modificando a dinâmica empresarial, a relação entre instituições de ensino e seus acadêmicos e, até mesmo, na forma com que a sociedade interage.

Não é difícil encontrar sujeitos que sejam resistentes ao avanço tecnológico, com um discurso de que os computadores vieram para tirar o emprego dos indivíduos, pois onde se tinham instituições com dez atendentes ao público sem o computador, agora se tem três atendentes usando esse dispositivo e obtendo por vezes resultados ainda mais otimizados do que com o quadro anterior de funcionários.

Tendo em vista que se vive em um país democrático e com liberdade de expressão, não se pode tirar o direito de nenhum indivíduo de chegar a suas próprias conclusões sobre o assunto. O espantoso é que, por muitas vezes, não se percebe que, se por um lado o avanço tecnológico tirou empregos, por outro lado gerou novos empregos.

O advento dos sistemas computacionais, dispositivos móveis e sistemas de comunicação modernos, geraram uma demanda de mão de obra qualificada nestas áreas, lembrando que não se está falando apenas de computadores convencionais, mas sim de qualquer dispositivo que possua um sistema embarcado. Vendo por esse lado, pode-se discutir o mérito de quantas novas profissões, áreas de atuação e vagas de emprego a tecnologia da informação gerou, sendo estas, nas áreas de fabricação de *hardware* (componentes físicos), desenvolvimento de *software* (programas e sistemas computacionais), venda e a manutenção destes, sem contar com novos paradigmas que surgiram, como as empresas virtuais entre outros.

Mesmo reconhecendo os benefícios trazidos pelo avanço tecnológico para os dias atuais, não se pode alimentar a ilusão de que os computadores são a solução para os problemas da humanidade. Como diria Laurent Gasser (apud PORTAL REGIONAL, 2010), “Computadores não resolvem problemas, eles executam soluções”. Tendo essa ideia como base, deve-se visualizar esses equipamentos como uma ferramenta a

mais em nosso leque de possibilidades técnicas e formas distintas para revolução de problemas.

Segundo a professora francesa (MAUBORGNE, 2010), que tem seu foco na disciplina de estratégia, nem toda inovação é tida como boa, pois existe a necessidade de ser útil e de atender à demanda dos usuários. O mesmo pode-se aplicar no que diz respeito aos sistemas informatizados, já que, nem todos os lançamentos, ou novas funcionalidades propostas trarão reais vantagens a seus utilizadores, fazendo com que esses continuem com as soluções usadas anteriormente.

No decorrer deste trabalho, descreve-se sobre a utilização do Sistema de Informações Geográficas (SIG), do processamento digital de imagens, bem como, será demonstrada a utilização de dois protótipos (algo ainda em estágio de testes ou desenvolvimento) de *software* desenvolvidos como proposta de inovação tecnológica na área de SIG. Esses *softwares* foram desenvolvidos com o intuito de automatizar procedimentos que, até o momento, vem sendo realizados pelos usuários dos programas computacionais de SIG, mais especificamente relacionados à delimitação de áreas de interesse por meio de vetorização.

## **1.1 Motivação**

A Geomática vem se consolidando como área de atuação interdisciplinar e tem a multiusabilidade como uma de suas grandes armas, isso porque, a utilização de informações geoespaciais vem se difundindo com grande velocidade na sociedade atual.

Atualmente, torna-se indiscutível a popularização de serviços geoespaciais, tais como, os mapas online, onde se podem traçar rotas de viagem, localizar um endereço em específico, seja ele comercial, para lazer ou hospedagem. Outro ponto, que não se pode deixar de ressaltar, é que, com o avanço tecnológico, os dispositivos móveis começaram a ganhar funcionalidades espaciais, seja um sistema que se comunique com um GPS via *bluetooth* ou, um aparelho que já possua tal dispositivo embutido, o

que contribui para a popularização de serviços nessa área.

Durante o tempo decorrido no programa de pós-graduação para obtenção do título de mestre em Geomática, depara-se com diversas teorias, práticas profissionais e de pesquisa, que vão desde a coleta de dados e seu processamento, até a exibição de resultados. Resultados esses que podem demonstrar espacialmente e/ou quantitativamente, informações valiosas para diferentes áreas do conhecimento.

Dentre as atividades, tipicamente realizadas em um sistema computadorizado de SIG, está a representação do mundo real. Essa representação pode ser feita de duas formas: por imagens raster (termo alemão que se refere à grade ou dados matriciais), ou por vetorização (dados vetoriais como pontos, linhas e áreas). Dados vetoriais são representados por pontos, linhas e polígonos, que são, formas de demonstrar respectivamente uma determinada coordenada, um trajeto ou caminho e, por fim, a delimitação de uma área.

Dentre as tecnologias envolvidas em SIG, está o processamento digital de imagens. Diferentemente das técnicas que trabalham com dados vetoriais, essa tecnologia tem seu foco de atuação sobre dados raster, os quais contêm uma representação numérica de pixel a pixel da imagem gerada, seja ela monocromática ou colorida. A ideia de interação entre as metodologias de processamento digital de imagem e de vetorização, juntamente com o intuito de otimizar desempenho e precisão na delimitação de áreas de interesse, foram pontos cruciais de motivação para o desenvolvimento deste trabalho.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo Geral**

Desenvolver programas computacionais que venham a servir como proposta de inovação tecnológica, no que diz respeito, à automação do processo de vetorização das

áreas de interesse sobre imagens de satélite, bem como, demonstrar a viabilidade do desenvolvimento de *buffers* variáveis sobre as mesmas.

### 1.2.2 Objetivos Específicos

Para alcançar o objetivo geral seja alcançado, serão necessários os seguintes objetivos específicos:

- Desenvolver um programa computacional que vetorize, por meio de polígonos, áreas de interesse previamente identificadas sobre imagens de satélite;
- Desenvolver um programa computacional que gere um *buffer* variável sobre áreas de interesse, tendo a sua variação associada à largura da área analisada;
- Criar imagens para testes pontuais do programa que não são atendidos pela imagem de satélite.

## 2 REVISÃO DE LITERATURA

### 2.1 Sistema de Informações Geográficas

Em um primeiro momento, quando se fala em Sistema de Informações Geográficas (SIG), muitos esperam uma definição sobre o que realmente ele vem a ser. Segundo Miranda (2010), por mais que seu objetivo nunca tenha se modificado, a definição de SIG vem evoluindo ao longo do tempo e ressignificando seus conceitos acompanhando as inovações provenientes das áreas de pesquisa que contribuem para sua evolução, dentre as quais, pode-se citar, a informática e a geografia.

Cada uma das áreas de conhecimento e pesquisa contribui com a evolução do SIG e de sua definição, e acabam trazendo consigo suas próprias evoluções. Dentro da informática, podem-se enfatizar as ferramentas de banco de dados e de linguagem de programação, já, no que diz respeito à geografia, tem-se forte amparo relacionado à área de mapas e, ainda, outras linhas de pesquisa que enfatizam sistemas para tomada de decisão.

Sistema de Informações Geográficas é um sistema de computadores e periféricos, programas, dados, pessoas, organizações e instituições com o propósito de coletar, armazenar, analisar e disseminar informações sobre áreas da Terra (CHRISMAN, 1997).

A definição apresentada por Chrisman (1997) pode não agradar a alguns, pois envolve elementos como “pessoas”, o que afeta a própria definição de sistema. Outro ponto, que pode deixar muitos com uma ideia errada, é o fato de que se tem nessa definição os elementos periféricos e programas. Isso pode levar o leitor a uma interpretação que o SIG surgiu após o advento da computação, o que não é verdade.



Os Sistemas de Informações Geográficas já existiam bem antes do surgimento dos computadores e da utilização de seus sistemas computacionais. A evolução dos SIGs vem de séculos de produção de mapas e da compilação de registros geográficos. Pelo que as pesquisas históricas indicam, os primeiros a empregarem o conceito de registro de propriedade ou registro da terra foram os romanos, o que deixa claro que tais sistemas não são uma invenção do homem moderno e que existem muitos outros conceitos envolvidos para a compreensão dos mesmos (MIRANDA, 2010 apud BERNHARDSEN, 1999).

Realmente, chegar a uma definição definitiva do que vem a ser um SIG se torna uma tarefa muito difícil, isso porque, existem inúmeras áreas de interesse que contribuem para o tema, cada uma dando o seu ponto de vista e a sua própria definição. Para facilitar esse trabalho, pode-se utilizar a técnica de estudo individualizado de cada um dos termos envolvidos nessa expressão.

Dentre os termos envolvidos, segundo Grilo (1996 apud BERTALANFFY, 1952), "Sistema é um todo organizado e unido, composto por duas ou mais partes interdependentes, componentes ou subsistemas, e delimitado por fronteiras identificáveis do seu macrossistema ambiental". A compreensão dessa definição deixa claro que SIG não é um simples sistema computacional; este até pode fazer parte do SIG, mas o contrário não acontece.

Segundo Freire (2010, apud SHAUGHNESSY, 1976), "Um sistema é um conjunto de partes interdependentes que, conjuntamente, formam um todo unitário, efetuando uma dada função". Tanto a definição de Bertalanffy quanto a definição de Shaughnessy afirmam que o sistema é um conjunto de partes, e mais, trabalhando por um objetivo comum. Como ve-se até o momento, o termo sistema pode ser usado em diferentes áreas do conhecimento, como na geografia que trabalha com sistema solar, sistema cartográfico, entre outros, ou no caso da biologia, que estuda sistemas digestivos, circulatórios, cardíacos, cada um sendo o todo de diversas partes.

Após descrever sobre os conceitos de sistema, ver-se-á algumas definições sobre o termo informação. Segundo Rezende (2003, apud MACHADO, 2002), "informação é uma abstração informal, que está na mente de alguém, representando algo significativo para uma pessoa"; já para Setzer (2002), "a transformação de dados

em informações deve ser vista, simplificada, como um tipo de pré-processamento de um processo de elaboração".

Juntamente com as definições de autores renomados, podem-se identificar os pontos principais de interesse que leva ao entendimento sobre o real significado do termo SIG. Com base no que foi descrito no parágrafo anterior, pode-se afirmar que, há uma transformação de dados em informação, ou seja, o dado isoladamente não tem o devido significado para seu utilizador. O significado só é obtido quando realizado o processamento desses dados, que resulta na informação que poderá ser analisada e interpretada por seus usuários.

Finalmente, o termo geográfica implica diretamente no conhecimento da localização dos itens de dados desejados, da mesma forma que permite que os mesmos possam ser calculados em coordenadas geográficas (latitude, longitude). Segundo Vieira (2003), trabalhos de cunho geográfico tem "trato com o espaço", sendo esta uma das grandes distinções da geografia em relação às demais ciências sociais. Ainda, sobre o ponto de vista de Vieira (2003), torna-se de grande importância a valorização das técnicas de cartografia, pois é por meio delas que se torna possível a sistematização do conhecimento cartográfico ou espaço geográfico.

### 2.1.1 Breve histórico

O real crescimento na manipulação de informações geográficas se deu na década de 80, juntamente e decorrente da popularização dos computadores e de seu potencial de processamento e armazenamento de dados. Foi realmente estimulante, para a época, a possibilidade de automatizar os processos relacionados à localização espacial, apoiado em sistemas de referência ou localização espacial, coordenadas, o que para uma máquina se torna fácil de processar por se tratar de instruções matemáticas (MIRANDA, 2010).

De acordo com Miranda (2010), a essência do SIG sempre esteve diretamente relacionada a sua característica interdisciplinar, e seu amadurecimento dependente do

bom relacionamento entre as áreas afins por meio de suas pesquisas e descobertas.

Por muito tempo, a fonte principal de informações a respeito de Sistemas de Informações Geográficas foram as conferências de organizações governamentais, principalmente Americanas. Um dos pontos principais de interesse dessas instituições foram os avanços tecnológicos específicos como os algoritmos de manipulação de informações geográficas e aplicações de projetos (MIRANDA, 2010).

Com o objetivo de priorizar o foco sobre o assunto SIG, surgiram em meados dos anos 90, organizações internacionais e séries de conferências, das quais se pode citar a *Annual Conference and Exposition of Geographical Information System/Land Information System (GIS/LIS)*, que ocorreu entre 1988 e 1995. Nelas foram apresentados trabalhos pioneiros sobre gerência de dados espaciais, visualização, estrutura de dados espaciais, conversão de dados espaciais, modelagem ambiental, sistemas de suporte a decisão, sistemas espacialistas e abordagem orientadas por objetos, incertezas e propagação de erros em dados geográficos digitais, análise espacial, entre outras aplicações (MIRANDA, 2010).

Os países desenvolvidos lideraram a pesquisa espacial com o uso de computadores. Por exemplo, no Reino Unido, Lord Chorley, um membro da House of Lords, presidiu um Comitê de consulta em 1985 para aconselhar o governo sobre o manuseio futuro da informação geográfica[...]. Uma iniciativa de pesquisa, ainda no Reino Unido, nos anos 80, concretizou-se com a instituição do *Regional Research Laboratory (RRL)* refletindo a importância dada ao manuseio de dados do censo e geodemográficos (MIRANDA, 2010, p. 21).

Em 1988, três anos mais tarde a iniciativa britânica citada anteriormente, teve início nos Estados Unidos o *National Center for Geographic Information and Analysis (NCGIA)* com o intuito de formar uma base de pesquisa em SIG. Para isso, diversas iniciativas de pesquisas foram adotadas em diferentes subáreas e com períodos variáveis de duração.

Com base nas iniciativas de pesquisa do NCGIA, pode-se visualizar na Tabela 1 cada uma das iniciativas adotadas, bem como, o objetivo principal a ser atingido pela mesma.

Tabela 1 – Iniciativas e objetivos do NCGIA

Iniciativa	Objetivo
Precisão de bancos de dados espaciais	Métodos e técnicas para trabalhar com erros e incertezas em dados geográficos
Linguagens de relações espaciais	Teoria de relações especiais usando linguagem natural e matemática
Representações múltiplas	Organização de versões múltiplas de métricas e topologias do mesmo dado para acesso eficiente
O uso e valor da informação geográfica	Melhorar modelos de acompanhamento do uso da informação geográfica
Arquitetura de banco de dados espaciais complexos	Eficiência no processamento, armazenamento, manipulação e análise de banco de dados espaciais
Sistemas espaciais de suporte à decisão	Papel do SIG no processo de tomada de decisão
Visualização da qualidade do dado espacial	Focalizar meios eficientes de manipular e comunicar visualmente componentes de qualidade de dados
Formalizando o conhecimento cartográfico	Identificar e buscar necessidades de pesquisa na compilação, generalização e produção automática de mapas
Instituições que compartilham informação geográfica	Procurar facilitar o compartilhamento de informações geográficas entre instituições
A relação espaço-tempo no SIG	Estudo do espaço geográfico, tempo e mudanças relacionadas a objetos no espaço geográfico
Integração de sensoriamento remoto e SIG	Identificar impedimentos para uma integração total de SR e SIG
Interface para usuário de SIG	Métodos de interação homem-máquina e questões relacionadas no projeto e implementação de interfaces para SIG
Análise geográfica e SIG I: análise espacial	Análise estatística de dados geográficos
Múltiplas ações para SIG na pesquisa de mudanças globais nos EUA	Desenvolvimento de um conhecimento abrangente e quantitativo das mudanças globais com o uso de SIG e análise de dados espaciais
Tomada de decisão colaborativa espacial	Ampliar o quadro de conhecimento conceitual atual para sistemas espaciais de suporte à decisão

Fonte: Miranda (2010)

### 2.1.2 Hierarquia da informação

A organização das informações em sistemas SIG, ao contrário do que muitos pensam, não se faz apenas de informações espaciais, e mesmo entre as informações

espaciais, nem todas são geográficas. A figura 1 representa a hierarquia das informações no sistema estudado, onde se pode identificar a informação sendo dividida em espacial e não espacial, sendo que, dentro dessa última, pode-se encontrar os sistemas gerenciais.

Os dados espaciais se subdividem em geográficos, que compreendem o sistema de informação geográfica, e não geográficos, que compreendem os sistemas de desenho assistido por computador (DAC) ou CAD do termo inglês *computer aided design*.

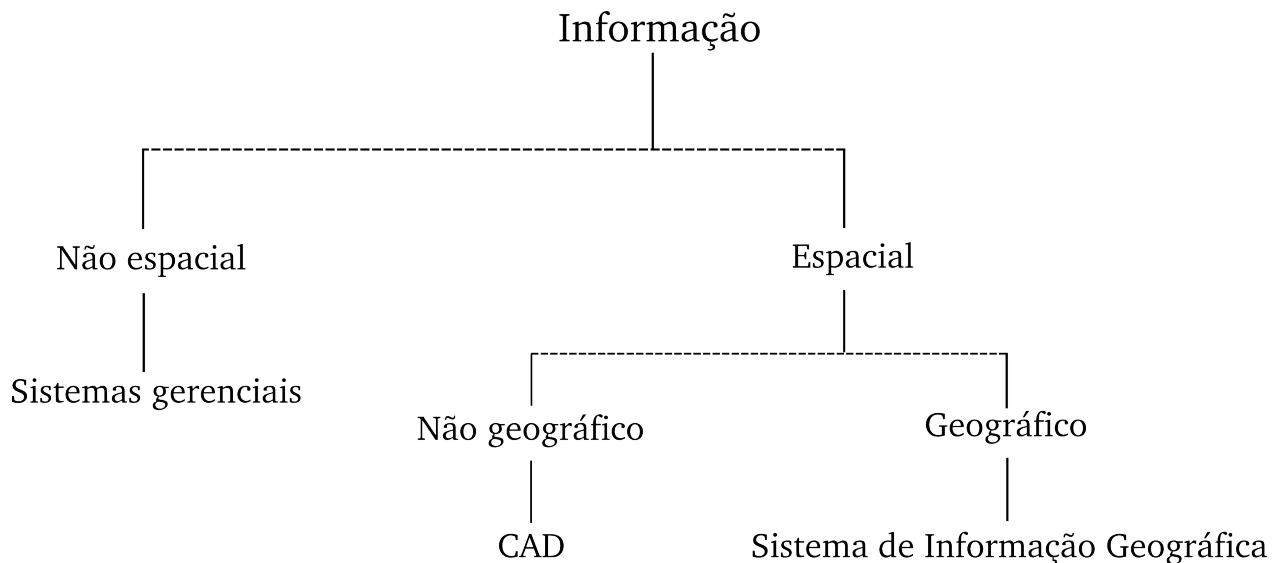


Figura 1 – Hierarquia da informação - Miranda (2010)

### 2.1.3 Componentes de Informática

Como já discutido anteriormente, os sistemas computacionais são ferramentas amplamente utilizadas no mundo SIG, da mesma forma que não se pode ficar com a visão simplista de que tais se limitem a sistemas computacionais. Nas definições descritas anteriormente, fica clara a importância da cartografia, geografia, matemática e tantas áreas distintas da ciência, e sem desmerecer nenhuma dessas, descrever-se-á

neste subcapítulo alguns componentes da informática que colaboram para o constante avanço na área do geoprocessamento (MIRANDA, 2010).

Como praticamente tudo que envolve a área de estudo da informática, pode-se realizar uma classificação em três tipos de componentes distintos. Em primeiro lugar tem-se os equipamentos definidos como de entrada, pode-se listar a mesa digitalizadora e o rastreador óptico ou *scanner*, ambos tem a função de converter mapas e dados analógicos (em papel) para o formato digital (legível por computador).

O equipamento responsável pelo processamento é o computador, segundo componente a ser descrito, e podendo este ser de diferentes plataformas. Entre os computadores atuais, podem-se definir três tipos distintos dependendo do fim para o qual se propõem. São eles: os computadores de grande porte ou também conhecidos como *mainframes*, os computadores de médio porte geralmente com múltiplos processadores, e os microcomputadores ou computadores pessoais que nos dias de hoje vêm com um potencial de processamento relativamente aceitável para atividades rotineiras em SIG.

No que diz respeito aos componentes de saída, tem-se dois tipos de representação, permanente ou efêmero. Entre os permanentes, cita-se os traçadores gráficos ou pelo seu termo em inglês *plotters* que, na atualidade, podem representar tanto dados matriciais quanto dados vetoriais, e manusear folhas até nos formatos A0 (841mm × 1189mm) para impressão de mapas, cartas topográficas entre outros materiais da área.

Outro elemento que faz parte dos componentes de saída que geram materiais permanentes é a impressora. Sendo um dos primeiros componentes utilizados pelo computador. No caso do SIG, é amplamente utilizada para impressão de relatórios, gráficos e estatísticas em geral.

Quando se fala em representações efêmeras, refere-se às informações que são apresentadas por um curto prazo de tempo, ou seja, as saídas de informações apresentadas por esses componentes não estarão sempre visíveis a seus utilizadores. Como exemplo, tem-se o monitor de computadores onde ocorre uma constante alteração nas informações apresentadas de acordo com sua manipulação pelo usuário (MIRANDA, 2010).

Pode-se identificar na representação da figura 2, a forma com que os componentes de informática se relacionam entre si, bem como o componente de processamento, representado pelo computador, faz o “meio de campo” entre os componentes de entrada (Digitalizadora, Rastreador) e os componentes de saída (Traçador Gráfico, Impressora).

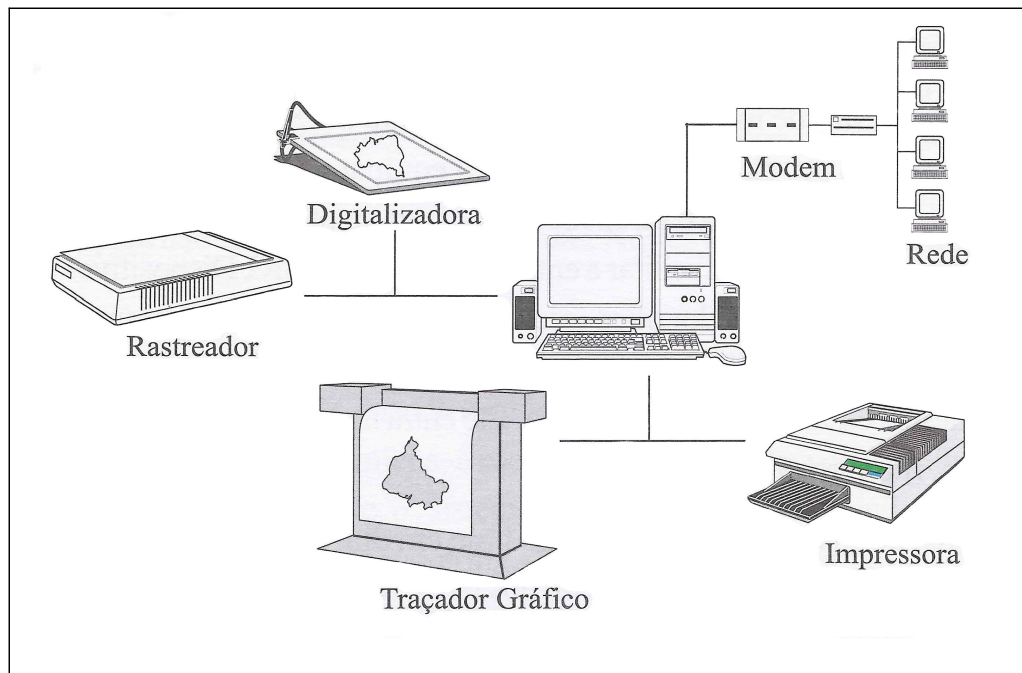


Figura 2 – Componentes do SIG – Miranda (2010)

Na figura 2, demonstrada anteriormente, tem-se também, a representação de uma rede de computadores que pode ser local ou remota. Essa ligação com redes de computadores é realmente muito comum. É por meio dela que diversos dispositivos e recursos são disponibilizados, seja para servir como dispositivo de entrada, processamento ou saída.

#### 2.1.4 Módulos de um SIG

As subdivisões e representações dos módulos de um SIG moderno variam muito e existem várias sugestões para o mesmo por motivos já descritos anteriormente. Conforme Miranda (2010), “optou-se por dividir um sistema de informações geográficas em quatro subsistemas: entrada, gerência dos dados, análise e saída”, o que, segundo o mesmo autor, não é um modelo estanque, podendo tomar formas e funcionalidades diferentes, o que não invalida o modelo para fins didáticos, facilitando assim, a compreensão do processo num geral.

Analisando neste momento a figura 3, pode-se identificar o subsistema de entrada de dados, constituído por funcionalidades que coletam e pré-processam os dados espaciais e não espaciais. O produto desse primeiro módulo pode ser mapas já existentes, observações de campo, imagens oriundas de satélites ou de sensores embarcados em aeronaves e, até mesmo, informações coletadas via sistema de posicionamento global, conhecido também como Global Positioning System (GPS).

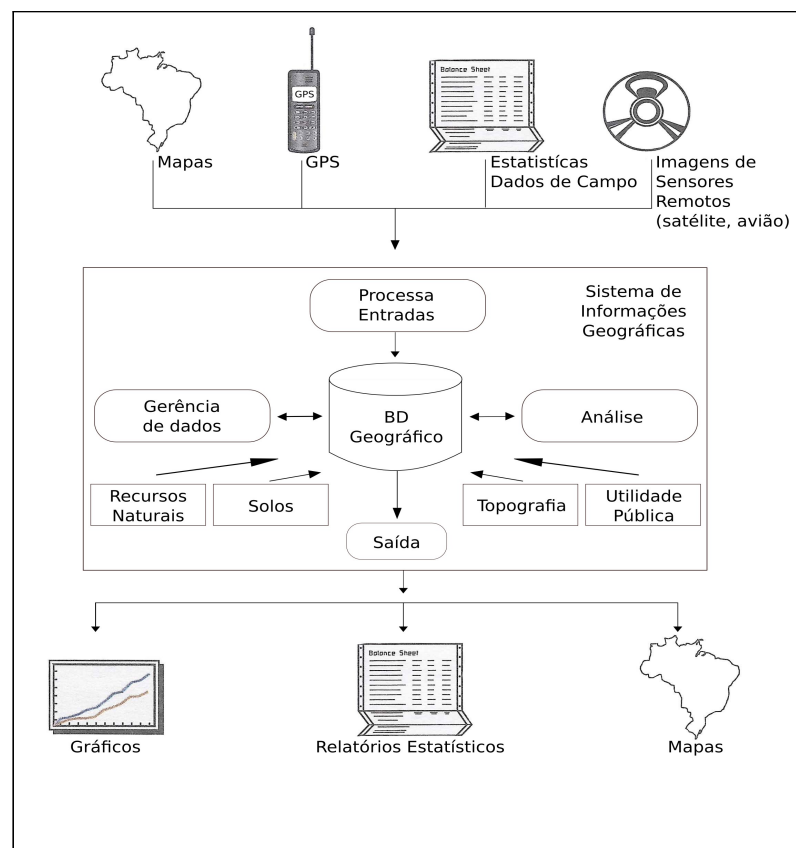


Figura 3 – Configuração de um SIG – Miranda (2010)



A funcionalidade do subsistema de gerência de dados tem como foco principal a organização desses dados, de forma que seja possível sua recuperação, atualização e edição.

Os elementos com que o subsistema de gerenciamento de dados trabalha são identificados como **base de dados geodemográficos**, que atua diretamente com uma coleção de mapas e informações associadas na forma digital. Na figura 4 tem-se uma ideia de como a estrutura de dados geográficos se constitui, com atributos de dados qualitativos e quantitativos, e dados espaciais que se especializam em geometria e se subdividem em três tipos, sendo eles pontos, linhas e áreas.

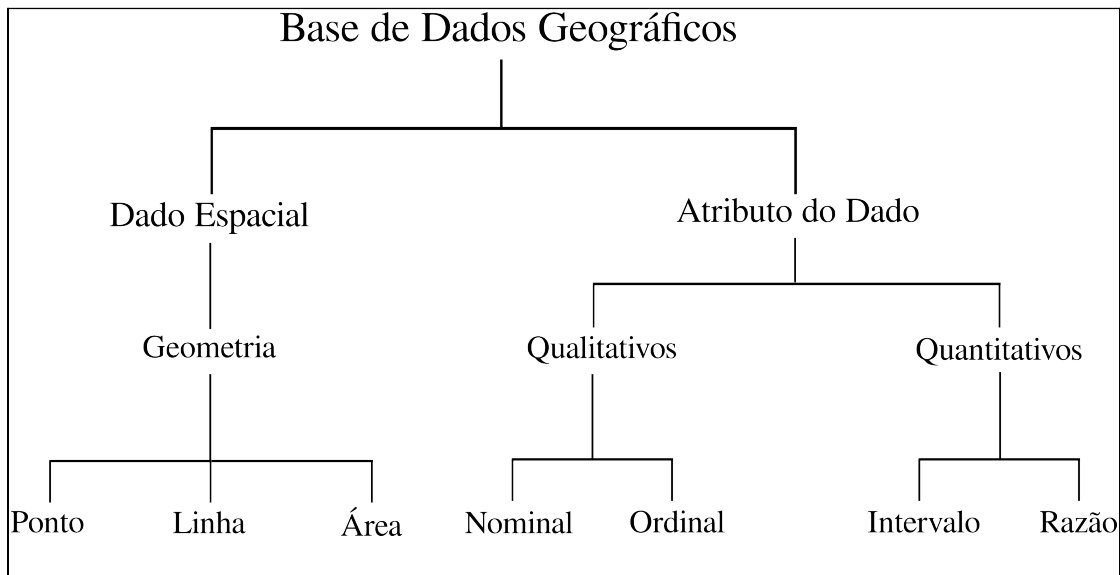


Figura 4 – Estrutura de dados – Miranda (2010)

Ainda sobre a perspectiva da figura 3 tem-se o subsistema de análise de dados, que é considerado o coração de um sistema SIG. Esse subsistema, segundo Miranda (2010), tem a função de “agregar e desagregar dados, estimar parâmetros e restrições e realizar funções de modelagem, entre outras tarefas”. Por esse motivo, são de responsabilidade desse módulo todas as ações responsáveis por análise de mapas

destinada à comparar e contrastar padrões de fenômenos relacionados à Terra, sempre procurando melhoria de métodos e desempenho.

Atualmente, os SIGs utilizam o potencial de processamento dos computadores para medir, comparar e descrever o conteúdo de bancos de dados, trabalhando com dados de diferentes tipos, podendo se combinar informações espaciais gerando uma nova perspectiva ou funcionalidade no que diz respeito as informações geográficas.

## **2.2 Processamento digital de imagens**

O desenvolvimento de sistemas autônomos que reproduzam as capacidades de sistema visual humano e que sejam capazes de reagir a estímulos visuais de forma adequada à área específica sob investigação ainda é um grande desafio. Uma das tarefas mais importantes em aplicações envolvendo um sistema de visão computacional é a análise ou a interpretação de imagens, cujo objetivo principal é obter uma descrição que contenha informação suficiente para distinguir entre diferentes objetos de interesse, de forma confiável e requerendo o mínimo de intervenção humana (PEDRINI, SCHWARTZ, 2008, p. 1).

Com base na citação e, em conformidade com as ideias dos autores Pedrini e Schwartz (2008), torna-se notório o estímulo para o desenvolvimento de novas técnicas e para a análise automática ou semiautomática de informações que podem ser extraídas de uma imagem ou cena com o auxílio de computadores e sistemas informatizados. Esse estímulo, segundo os mesmos autores, vem da capacidade humana de captar, processar e interpretar grande volume de dados de natureza visual.

### **2.2.1 Áreas de Aplicação**

Existem atualmente diversas áreas de interesse onde o processamento digital de imagens (PDI) pode se fazer presente. Associando essa afirmação ao avanço tecnológico atualmente aplicado na área da tecnologia digital, é o que tem permitido um número crescente de algoritmos, técnicas e aplicações para as mais diferentes

finalidades no que diz respeito a imagens digitais (PEDRINI, SCHWARTZ, 2008).

Dentre as áreas onde o processamento digital de imagens tem importante papel na resolução de problemas, ou no auxílio à tomada de decisão, pode-se citar Medicina, Biologia, Astronomia, Microscopia, Artes, Arqueologia e ainda na automação industrial, sensoriamento remoto, área militar, segurança e vigilância.

Um das áreas, citada no parágrafo anterior, e que possui grande suporte por parte do PDI, é a Medicina, onde se utiliza tal tecnologia para auxiliar nos diagnósticos médicos em exames que utilizam imagens como base de suas análises. Alguns dos exames que pode-se listar como exemplo seriam os que capturam imagens por raios X, tomografia computadorizada, ressonância magnética e ultrassonografia.

Em várias especialidades da Medicina, o uso das técnicas de análise e interpretação de imagens facilita na identificação de lesões e/ou regiões atingidas por câncer, assim proporcionando maior rapidez e precisão nos diagnósticos prestados a seus pacientes.

O uso das técnicas de PDI são direcionadas a verificação de falhas em circuitos impressos, separação de peças por meios automatizados, ou até mesmo no reconhecimento de defeitos encontrados no uso de soldas, tanto em automação industrial, montagem e inspeção de produtos.

Em sensoriamento remoto, a análise de fotografias aéreas ou imagens de satélite permite uma melhor compreensão da superfície terrestre, auxiliando o acompanhamento de áreas urbanas, a previsão de fenômenos como terremotos, erupções, inundações e furacões, monitoração de áreas atingidas por erosão, previsão de safras, determinação de áreas de desmatamento, análise de composição do solo, extração de feições cartográficas (por exemplo, estradas, rios, edificações, divisa de culturas), dentre outras atividades (PEDRINI, SCHWARTZ, 2008, p. 3).

Como identifica-se na citação feita na obra dos autores Pedrini e Schwartz (2008), juntamente com o que já foi descrito no capítulo um, esta dissertação tem sua fundamentação teórica e prática diretamente relacionada ao uso do processamento digital de imagens e sensoriamento remoto, que trata das técnicas de captura de informação acerca de um objeto sem ter com ele um contato direto, nesse caso as imagens de satélites.

A área militar foi uma das pioneiras na utilização das técnicas de processamento

digital de imagens, possuindo inúmeras aplicações para as mais diversas finalidades, tais como, identificação de alvos por imagens de satélites, navegação de veículos autônomos, entre outras (PEDRINI, SCHWARTZ, 2008).

### 2.2.2 Representação da Imagem

A representação e manipulação de uma imagem em computador requer a definição de um modelo matemático adequado da imagem. [...]

Um modelo físico para a intensidade de uma cena sob observação pode ser expresso em termos do produto entre dois componentes, a quantidade de luz incidente na cena e a quantidade de luz refletida pelos objetos presentes na cena. Esses componentes são chamados de *iluminância* e *reflectância* (PEDRINI, SCHWARTZ, 2008, p. 14).

Conforme as afirmações de Pedrini e Schwartz (2008), uma imagem é o produto da interação entre iluminância e a reflectância, e para realizar sua representação, se faz necessária a definição de um modelo matemático, que neste caso é representada pela função  $f(x,y)$  onde

$$f(x,y) = i(x,y) r(x,y) \quad (1)$$

sendo  $i(x,y)$  é uma determinação obtida pela fonte de energia (luz), enquanto  $r(x,y)$  é uma determinação obtida pelas características de reflectância dos objetos na cena.

Segundo os mesmos autores, o processo de digitalização envolve dois passos, sendo estes a amostragem e a quantização.

A amostragem consiste em discretizar o domínio de definição da imagem nas direções  $x$  e  $y$ , gerando uma matriz  $M \times N$  amostradas, respectivamente. A quantização consiste em escolher um numero inteiro  $L$  de níveis de cinza (em uma imagem monocromática) permitidos para cada ponto da imagem (PEDRINI, SCHWARTZ, 2008, p. 15).

A equação que representa uma imagem digital pode ser vista na figura 5, onde cada  $f(x,y)$  corresponde a um valor dentro do estabelecido pelo valor  $L$  descrito na citação realizada anteriormente, e também ao menor elemento da imagem, pela literatura definido como pixel (do inglês *picture element*).

$$f(x, y) \approx \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & & & \\ f(N-1, 0) & f(N-1, 1) & \dots & f(N-1, N-1) \end{bmatrix}$$

Figura 5 – Equação de uma imagem – Miranda (2006)

Como demonstra a figura 5, as imagens digitais são geralmente representadas por uma estrutura que contém um número variável de linhas e colunas, sendo esta variação de 0 a  $N - 1$ . Outra representação de imagens digitais amplamente utilizadas na área de PDI é, por meio de uma grade de linhas e colunas, formando assim uma matriz de pixels a qual pode-se analisar com o auxílio da figura 6. O pixel é constituído de três propriedades, um valor, uma posição e um tamanho.

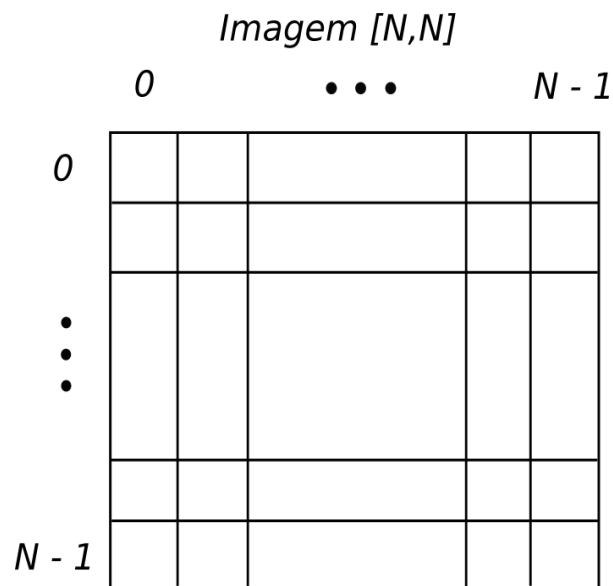


Figura 6 – Esquema de uma imagem digital – Miranda (2006)

### 2.2.3 Resolução Espacial

Em conformidade com as ideias de Pina e Santos (2000), resolução espacial “é uma medida da menor separação entre dois objetos, ou seja, representa a capacidade do sistema sensor de “enxergar” objetos na superfície terrestre: quanto menor o objeto passível de ser visto, maior é a resolução espacial.”

Sobre a perspectiva de Pedrini e Schwartz (2008), “a resolução espacial está associada à densidade de pixels da imagem [...] quanto maior a densidade de pixels em uma imagem, maior será a resolução da imagem”. Outro ponto importante para a compreensão do assunto, é que, uma imagem que contenha um grande número de pixels não necessariamente terá maior resolução do que outra que tenha menor número de pixels.

A resolução de uma imagem deve ser definida de acordo com as exigências de detalhamento necessário, a fim de propiciar a identificação dos alvos desejados. Em uma região de 400 cm<sup>2</sup>, se tivermos 20 amostras para o eixo x, e 20 amostras para o eixo y, cada pixel da imagem representará 1 cm x 1 cm. Se quer uma resolução maior, deve-se aplicar para essa mesma área uma amostragem maior, como por exemplo, 40 amostras para o eixo x, e 40 amostras para o eixo y, sendo assim cada pixel representaria 0.5 cm x 0.5 cm da imagem, dando um maior detalhamento sobre os objetos analisados.

Com base nas ilustrações da figura 7, pode-se identificar quatro sensores e suas respectivas resoluções espaciais. Note que na imagem da foto aérea, onde cada pixel representa 0.5 cm x 0.5 cm, tem-se o maior número de amostragens para uma mesma área, enquanto que, na imagem do Landsat 7, onde cada pixel representa 30 m x 30 m, se obteve o menor número de amostragens, fazendo com que esta tenha menor detalhamento do que a primeira.

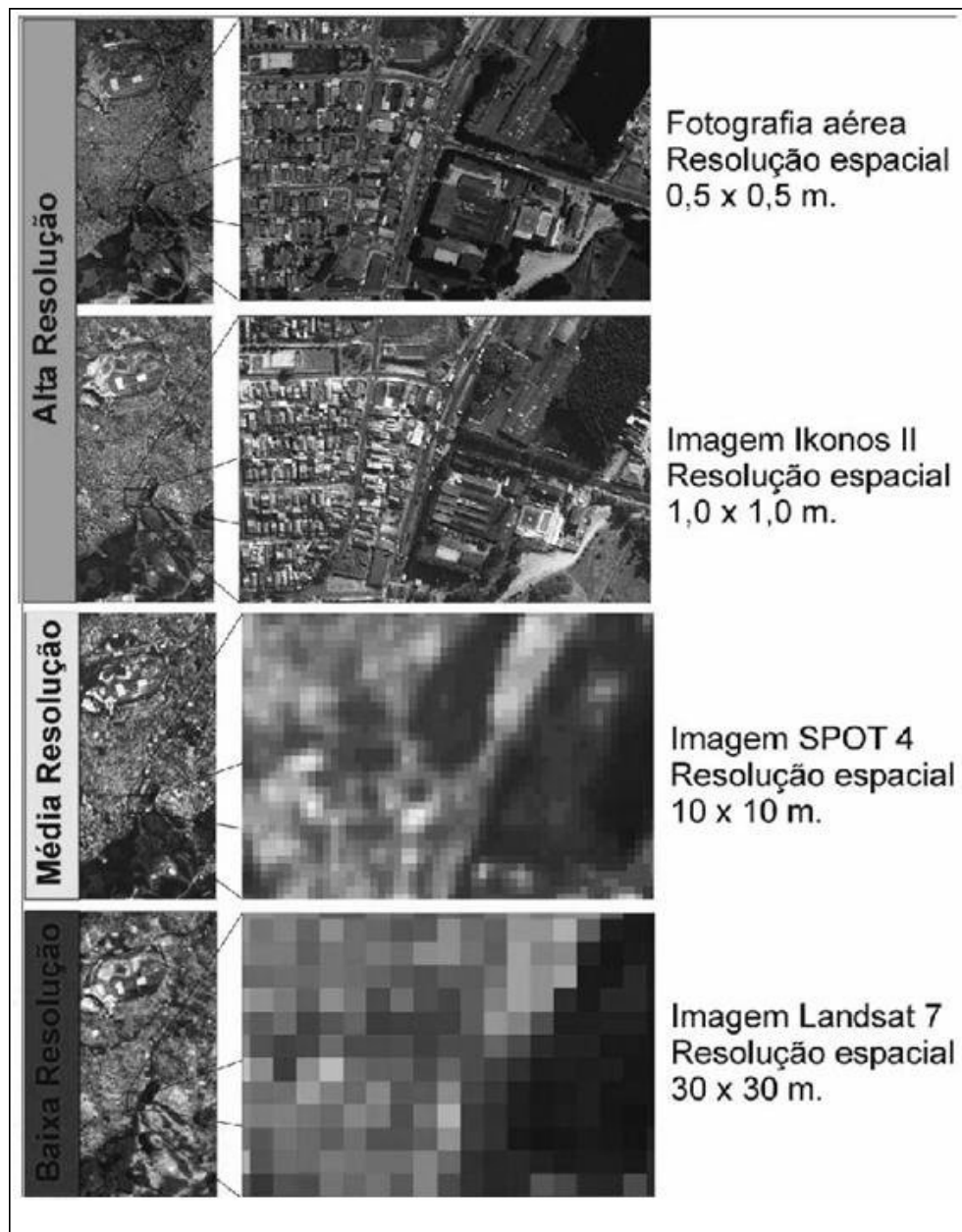


Figura 7 - Sensores e suas resoluções espaciais. Melo (2002)

#### 2.2.4 Relacionamento de Vizinhança

A vizinhança é um dos relacionamentos básicos entre os elementos de uma imagem. Cada pixel de localização (x,y) em uma imagem possui quatro vizinhos

horizontais e verticais, que se posicionam nas coordenadas  $(x + 1, y)$ ,  $(x - 1, y)$ ,  $(x, y + 1)$  e  $(x, y - 1)$  formando assim a chamada vizinhança – 4 de simbologia  $N_4(f)$  como demonstrado na figura 8 (PEDRINI, SCHWARTZ, 2008).

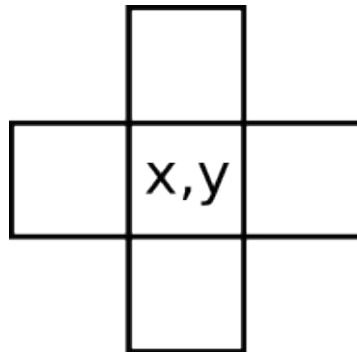


Figura 8 – Vizinhança horizontal e vertical – Miranda (2006)

Seguindo os princípios de vizinhança, também existem os quatro vizinhos diagonais do pixel, elementos esses posicionados nas coordenadas  $(x - 1, y - 1)$ ,  $(x - 1, y + 1)$ ,  $(x + 1, y - 1)$  e  $(x + 1, y + 1)$ , denominados pela literatura como  $N_d(f)$  e representados na figura 9.

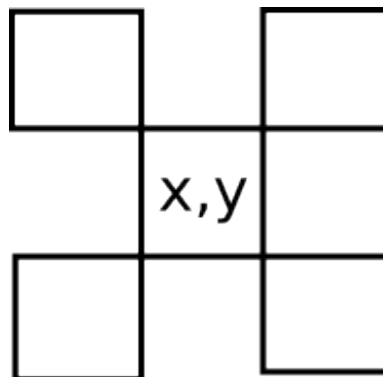


Figura 9 – Vizinhança diagonal – Miranda (2006)



Além das vizinhanças horizontais e verticais, e das vizinhanças diagonais, existe a chamada vizinhança-8, que nada mais representa do que a união entre as duas representações anteriores que é demonstrada pela nomenclatura a seguir.

$$N8(f) = N_4(f) \cup N_d(f) \quad (2)$$

O conceito de vizinhança também pode se aplicar às imagens tridimensionais, onde cada pixel, obrigatoriamente, é composto por três coordenadas, sendo estas x, y e z. O pixel de uma imagem tridimensional pode possuir vizinhança-6 como ilustrado na figura 10, vizinhança-18 como ilustrado na figura 11 e finalmente vizinhança-26 como ilustrado na figura 12.

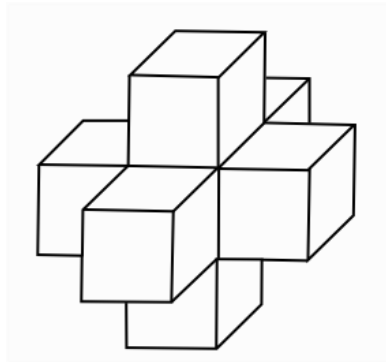


Figura 10 - Vizinhança 6 – Pedrini e Schwartz (2008)

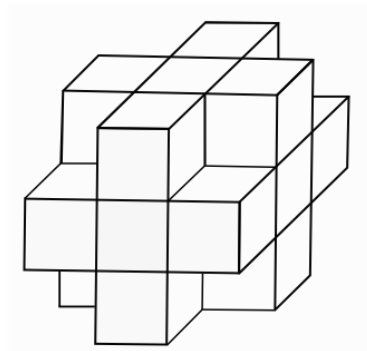


Figura 11 - Vizinhança 18 - Pedrini e Schwartz (2008)

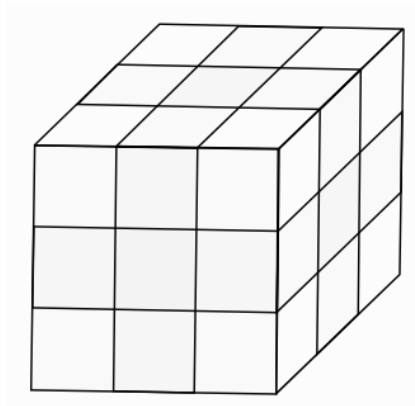


Figura 12 - Vizinhança 26 - Pedrini e Schwartz (2008)

### 2.3 Linguagem de programação Python

A linguagem de programação Python, vem ganhando cada dia mais espaço no mercado de desenvolvimento internacional de aplicativos computacionais. Algumas de suas características serão descritas nos próximos parágrafos a fim de se compreender suas especificidades e sua grande aceitação no cenário mundial.

Primeiramente, deve-se destacar sobre Python que se trata de um *software* de código aberto, permitindo assim a interação da comunidade de desenvolvedores com ela, reportando problemas, propondo melhorias, fazendo com que a mesma continue evoluindo a passos largos. Python é distribuído por meio de uma licença compatível com a *General Public License* (GPL), porém com maior flexibilidade, permitindo que seja utilizada, até mesmo, por produtos proprietários (BORGES, 2010).

Além de ser utilizado como linguagem principal no desenvolvimento de sistemas, o Python também é muito utilizado como linguagem script em vários softwares, permitindo automatizar tarefas e adicionar novas funcionalidades, entre eles: BrOffice.org, PostgreSQL, Blender, GIMP e Inkscape (BORGES, 2010, p.13).

Python é uma linguagem de programação considerada de altíssimo nível, orientada a objeto, de tipagem dinâmica, pelo fato de o tipo das variáveis serem inferidos em tempo de execução pelo interpretador, atribuindo a ela as operações

passíveis de execução e denominadas fortes, pois as operações são validadas antes da execução.

Considerando que não ocorrem mudanças automáticas de tipagem, como alteração de um valor numérico para texto, caso necessário, deve-se efetuar explicitamente a conversão do tipo de dado das variáveis antes das operações.

### 2.3.1 Breve histórico

Segundo Borges (2010), a linguagem Python foi desenvolvida no ano de 1990, a partir de uma outra linguagem da época chamada de ABC. Seu criador foi Guido Van Rossum, mais precisamente no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda. Seu foco principal é atuar em aplicações na área da Física e Engenharia.

Na atualidade a linguagem é amplamente aceita por empresas de alta tecnologia para aplicação nas mais diversas áreas. Entre essas empresas destacam-se.

- Google (aplicações Web).
- Yahoo (aplicações Web).
- Microsoft (IronPython: Python para .NET).
- Nokia (disponível para as linhas recentes de celulares e PDAs).
- Disney (animações 3D).

### 2.3.2 Compilação e interpretação

Para que ocorra a execução de um programa em Python, o mesmo precisa ter seu código fonte traduzido para o que é chamado de *bytecode*, que por sua vez, vem a ser um formato binário para o interpretador. Assim como em Java e em outras

linguagens interpretadas, o arquivo *bytecode* é multiplataforma e pode ser executado sem o código fonte (BORGES, 2010).

A grande vantagem de se ter um arquivo *bytecode* para um interpretador é a portabilidade que o mesmo proporciona à aplicação. Sendo esse um arquivo único para diferentes interpretadores, basta que se tenha o interpretador Python para os sistemas Linux, Windows, Mac, Solares, Android, Iphone, e se tem a possibilidade de executar a aplicação em qualquer um dos dispositivos, seja ele computador pessoal, servidor ou, até mesmo, dispositivo móvel.

Quando da execução de um código Python, o interpretador gera o *bytecode* e o armazena em disco, sendo que, para a próxima execução, não será necessário recompilar o código fonte, dessa forma proporcionando um ganho de desempenho na carga do programa.

Sempre que houver uma alteração no código fonte ou, por algum motivo, o *bytecode* não se encontrar mais em disco, o interpretador se encarrega de regerar o arquivo *bytecode* e o mantém em disco novamente.

O armazenamento em disco do *bytecode*, pode ser feito com a extensão “pyc” (normal), ou com a extensão “pyo” (otimizado). Outra funcionalidade, que pode ser implementada, é o empacotamento do *bytecode* juntamente com o interpretador o que facilita a distribuição de aplicações para máquinas que não possuam Python instalado.

Os empacotadores são utilitários que são usados para construir executáveis que englobam o *bytecode*, o interpretador e outras dependências, permitindo que o aplicativo rode em máquinas sem Python instalado, o que facilita a distribuição de programas (BORGES, 2010, p.13).

Entre os empacotadores para Python pode-se citar:

- Py2exe (apenas para Windows).
- cx\_Freeze (multiplataforma).

As interações descritas nos parágrafos anteriores são apresentadas na figura 13.

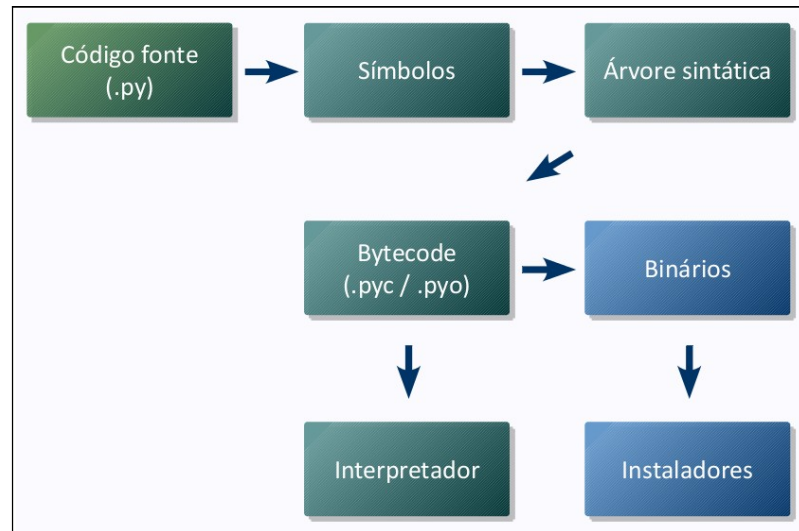


Figura 13 – Processos de interação do código Python – Borges (2010)

### 2.3.3 Frameworks

Outro ponto forte da linguagem de programação Python, é a expressiva quantidade de *frameworks* de desenvolvimento existente para a mesma. *Framework*, segundo Borges (2010), vem a ser um conjunto de componentes de software, como bibliotecas e utilitários em geral que foram desenvolvidas para serem utilizadas por outros sistemas.

Entre alguns dos *frameworks* existentes pode-se citar:

- Web: Django, TurboGears, Zope e web2py.
- Interface gráfica: wxPython, PyGTK e PyQt.
- Processamento científico: NumPy e SciPy.
- Processamento de imagens: PIL.
- 2D: Matplotlib e SVGFig.
- 3D: Visual Python, PyOpenGL e Python Ogre.
- Mapeamento objeto-relacional: SQLAlchemy e SQLAlchemyObject.

Dentre os *frameworks*, citados anteriormente, pode-se destacar a *Python Imaging Library* (PIL), que nada mais é do que, uma biblioteca com inúmeras funcionalidades voltadas ao processamento digital de imagens. Recordando o item sobre processamento digital de imagens, tem-se consciência de que uma imagem é constituída por uma estrutura matricial de representação de dados, sendo a biblioteca PIL (componente de *software*) a grande facilitadora nas manipulações necessárias no que diz respeito a imagens.

Com o auxílio da biblioteca PIL torna-se possível, em poucas linhas de código, a realização de manipulações como leitura, exibição, edição e escrita de imagens. Dentre outras funcionalidades disponíveis, pode-se listar: recortar, colar, mesclar, redimensionar, rotacionar, gerar histograma e método de representação da quantidade de pixels por grupo, o que permite uma melhor utilização das técnicas de contraste e realce de imagens (LUNDH, ELLIS, 2002).

### 3.4 Filtros

Segundo Pedrini e Schwartz (2008), a operação de filtragem de imagens pode ser realizada tanto no domínio espacial quanto no domínio de frequência. Os filtros são classificados em três categorias. Os filtros denominados de passa-baixas são indicados para suavização de imagens, ou seja, as frequências mais altas, que correspondem às mudanças bruscas no padrão de sinal da imagem ou ruídos, são amenizadas.

Dentre os filtros de suavização cita-se o Filtro de Média, o Filtro Gaussiano, o Filtro de Mediana sendo que este tem sua utilização representada na figura 14, os Filtros de Ordem de Moda e o Filtro de Média dos K Vizinhos Seleccionados.



Figura 14 - Suavização da imagem por meio do Filtro de Mediana - Fonte SOUZA (2012).

Os filtros denominados passa-alta são utilizados com um caráter inverso do filtro passa-baixas anteriormente explanado. Sua utilização está relacionada a enfatizar os ruídos ou transições entre áreas distintas da imagem, também conhecidas como bordas. Dentre os filtros de passa-alta ou também denominados de filtros de acentuação, pode-se citar: os Filtros Gradientes ou Derivada de Imagens, o Filtro e Operador de Sobel, o Filtro de Roberts, o Filtro de Prewitt, o Filtro Laplaciano, o Filtro LoG ou o Laplaciano do Gaussiano, o Filtro de Canny representado na figura 15 (A - imagem original, B - produto do filtro) e outros filtros direcionais (norte, noroeste, sudeste).

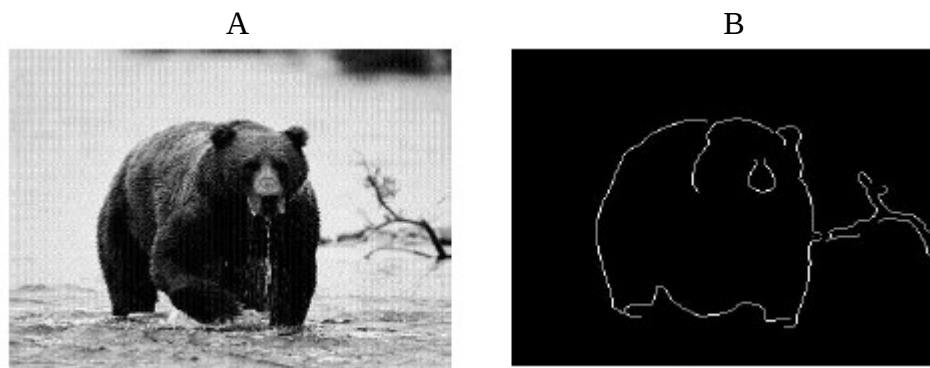


Figura 15 - Uso do filtro de Canny para reconhecimento de bordas - Fonte SOUZA (2012).

A terceira categoria, conforme a conceituação de Pedrini e Schwartz (2008), é representada pelo filtro passa-faixa. Esse filtro trabalha com o realce de um intervalo de frequência da imagem, geralmente rejeitando os demais valores fora da frequência selecionada como pode ser visto na figura 16 (A - imagem original, B - produto do filtro).

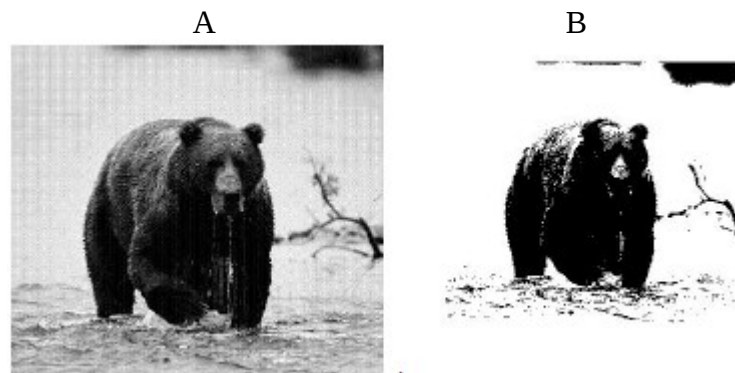


Figura 16 - Uso do filtro passa-faixa – Fonte SOUZA (2012)

Realizando uma análise mais detalhada na figura 16, verifica-se que todos os pixels em uma faixa mais alta do nível de cinza foram realçados para 255, sendo este o nível mais alto de sinal, já os demais assumiram valor 0 (zero), nível mais baixo de sinal, criando assim um efeito de contraste máximo, o que facilita a identificação de áreas de interesse na imagem.



## **3 MATERIAL E MÉTODOS**

Neste capítulo descrevemos sobre materiais, ferramentas, técnicas e tecnologias utilizadas no desenvolvimento deste trabalho. São detalhados os critérios na escolha e criação de imagens, utilização de IDE de Desenvolvimento, utilização de filtros e das técnicas de programação e processamento digital das imagens utilizadas.

### **3.1 Escolha do material de trabalho**

#### **3.1.1 Imagem de teste**

A imagem escolhida para fazer parte do desenvolvimento deste protótipo de inovação tecnológica, tinha que apresentar algumas características específicas ao propósito do mesmo. Tendo como foco a automação, no que diz respeito a vetorização de áreas de interesse, mais precisamente em corpos de água para auxiliar na delimitação de área de preservação permanente (APP).

Dentre as características necessárias, uma delas é possuir reflectância que facilite a identificação de corpo de água, o que segundo Figueiredo (2005) é encontrado na banda 4 do sensor TM do satélite Landsat, como pode ser visto na tabela 2, onde são apresentadas as bandas do sensor TM e suas aplicações.

Tabela 2 - Bandas do sensor TM e suas aplicações

<i>BANDA</i>	<i>FAIXA ESPECTRAL <math>\mu m</math></i>	<i>PRINCIPAIS APLICAÇÕES</i>
1	0,45 – 0,52	- Mapeamento de águas costeiras - Diferenciação entre solo e vegetação - Diferenciação entre vegetações coníferas e decídua
2	0,52 – 0,60	- Reflectância da vegetação verde sadia
3	0,63 – 0,69	- Absorção da clorofila - Diferenciação de espécies vegetais
4	0,76 – 0,90	- Levantamento de biomassa - Delineamento de corpos d'água
5	1,55 – 1,75	- Medidas de umidade da vegetação - Diferenciação entre nuvens e neve
6	10,4 – 12,5	- Mapeamento de estresse térmico em plantas - Outros mapeamentos térmicos
7	2,08 – 2,35	- Mapeamento hidrotermal -

Fonte: FIGUEIREDO (2005)

Outro aspecto, não menos importante, na escolha da imagem que serviria de norteadora na construção desse protótipo, é possuir características reais e variáveis para que se pudesse investigar o maior número de possibilidades de interação desse *software* com um caso real.

Dentre os alvos desejados em uma imagem podemos listar:

- Lagos
- Lagoas
- Reservatórios de água artificiais ou não
- Rios ou cursos de água com diferentes larguras e se possível contendo ramificações.

As características, descritas anteriormente, foram encontradas na imagem denominada "LANDSAT\_5\_TM\_20090411\_223\_081\_L2\_BAND4" obtida no catálogo de

imagens do sítio do Instituto Nacional de Pesquisas Espaciais (INPE), datada de 11 de abril de 2009, sem cobertura de nuvem e pertencente à banda 4 do sensor TM como já mencionado anteriormente.

Para o desenvolvimento do que foi proposto, não se fez necessária a utilização da imagem por completo, optando-se por realizar o recorte de uma área localizada no quadrante superior esquerdo da imagem. Essa área contendo todas as características necessárias para o desenvolvimento da pesquisa, é representada na figura 17 de coordenadas de canto UTM XY superior esquerdo (687236, 6707269), inferior esquerdo (687236, 6694754), superior direito (705542, 6707269) e inferior direito (705542, 6694754).

Cabe frisar que não se está levando em consideração a resolução espacial da imagem, pois se sabe que as imagens Landsat são de resolução 30 x 30 m, o que impossibilitaria teste de largura para cálculo APP inferiores a essa distância. Porém, por se tratar de um protótipo, e esse possuir o dever de atuar em distintas resoluções, adotou-se uma variável de sistema responsável por controlar as funcionalidades do *software*, de acordo com sua resolução. A essa variável foi admitida diferentes resoluções, para assim se analisar o comportamento do sistema.



Figura 17 - Recorte de imagem da banda 4 do sensor TM – INPE (2011)

### 3.1.2 Linguagem de programação

A linguagem de programação adotada para o desenvolvimento desse sistema foi a linguagem Python. Por mais que o primeiro contato com ela tenha sido já durante o curso para obtenção do título de mestre em Geomática, sua curva de aprendizado se mostrou relativamente curta, além de proporcionar um excelente custo-benefício, o que pode ser confirmado pelo item 2.3 presente na revisão de literatura.

### 3.1.3 Ambiente de desenvolvimento

O ambiente de desenvolvimento utilizado, ou *Integrated Development Environment* (IDE) como é conhecido, foi o Eclipse SDK na sua versão 3.5.2, também denominado de GALILEO. O Eclipse por sua vez, consagra-se como uma IDE altamente flexível no que diz respeito a sua configurabilidade e expansibilidade.

Dentre características que colaboram para a ampla utilização dessa IDE, por desenvolvedores de todo o mundo, estão os pacotes de idioma que dão a possibilidade de personalização da mesma para distintas nacionalidades. Além disso, sua gratuidade, os inúmeros *plugins* que adicionam funcionalidades adicionais e sua alta capacidade de integração com *frameworks* de desenvolvimento fazem da IDE Eclipse uma opção sempre relevante (GUGIK, 2010).

Por mais que sua maior utilização seja para desenvolvimentos na área da tecnologia Java, os *plugins* existentes permitem que sejam desenvolvidos projetos nas mais distintas linguagens, entre as quais se destacam: C, C++, C#, PHP, Perl, Ruby, Python, Prolog, Lua, entre outras.

No caso do desenvolvimento de projetos em Python, por meio da IDE Eclipse, faz-se necessária a instalação do *plugin* denominado "PyDev", o qual adiciona funcionalidades que vêm facilitar a implementação de tais projetos. Dentre as funcionalidades adicionadas pelo *plugin* PyDev, encontram-se a análise de código que

pode apontar possíveis erros na escrita de comandos, variáveis não utilizadas ou não resolvidas, importações não utilizadas ou não resolvidas.

Outra funcionalidade, não menos importante, do *plugin* anteriormente citado é a conclusão de código; essa, por sua vez, permite que o desenvolvedor tenha à sua disposição uma caixa auxiliar contendo as opções de código para autocompletar de acordo com o que está disponível para certo comando ou variável. Essa funcionalidade, juntamente com a interface do Eclipse já configurado com o PyDev, pode ser vista na figura 18.

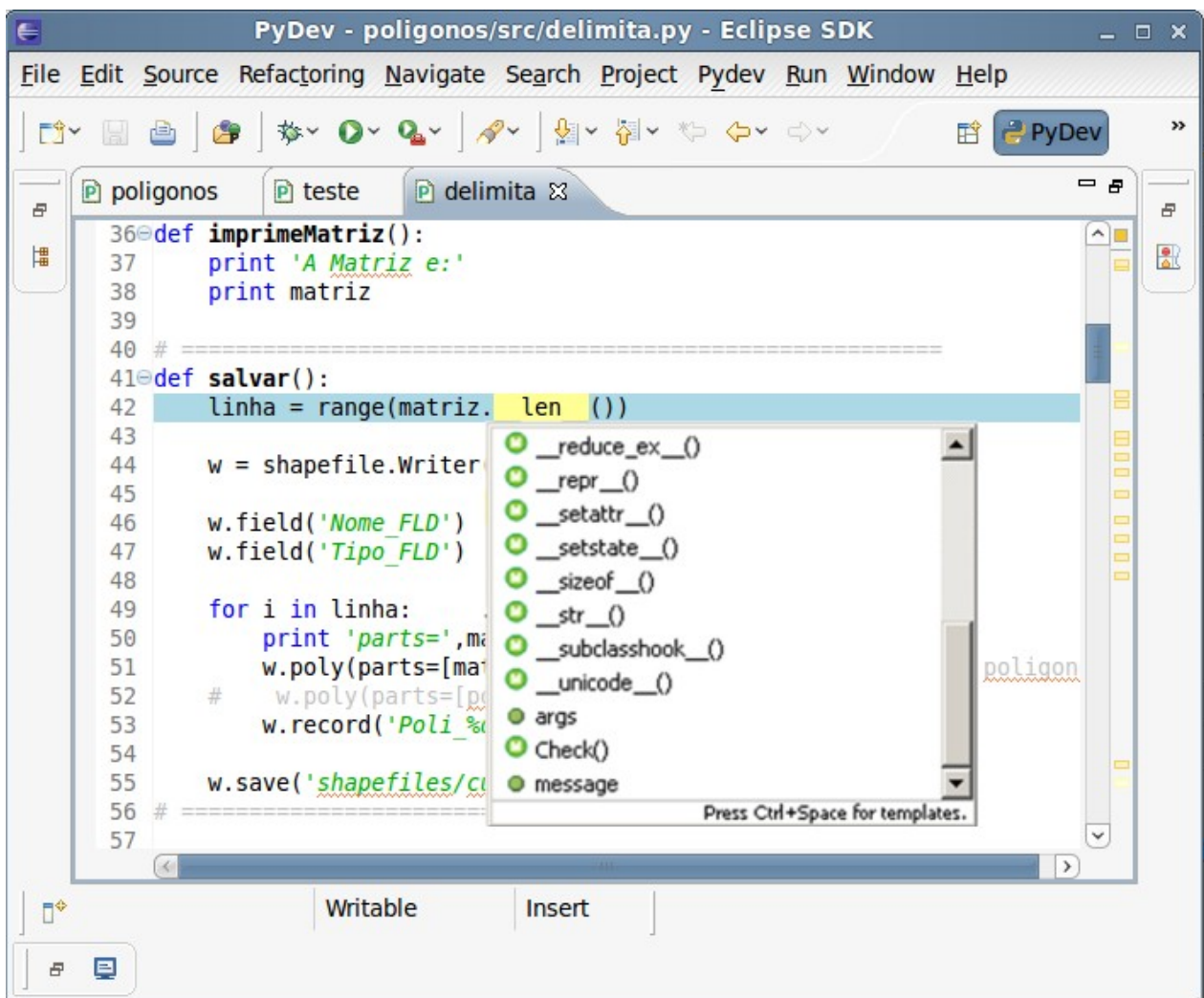


Figura 18 - *Plugin* PyDev do IDE Eclipse, e funcionalidade conclusão de código.

## 3.2 Filtro: Método para obtenção da área de interesse

Após a definição do material de trabalho e das ferramentas a serem utilizadas, vem a etapa de manipulação das mesmas. Com intuito de propiciar o desenvolvimento do software que venha a auxiliar na vetorização de cursos de água, para com isso automatizar as delimitações de APPs, procurou-se por meio das técnicas de filtro de imagens realçar o leito dos corpos d'água existentes na imagem.

A seguir serão descritas as técnicas de filtro utilizadas, por mais que nem todas tenham obtido o resultado esperado. Optou-se pela descrição da técnica utilizada, embora produto gerado não tenha sido efetivamente usado, para que assim se pudesse vislumbrar os caminhos seguidos até chegar aos resultados finais.

### 3.2.1 Identificando bordas

Tendo como objetivo principal a vetorização dos leitos das laminas d'água, o primeiro método de filtragem utilizado foi o de identificação de bordas. Essa funcionalidade é obtida em função da importação da biblioteca PIL, responsável pela adição de várias funcionalidades no que diz respeito ao processamento digital de imagens.

É mostrada na figura 19 um trecho de código em Python, similar ao usado para realizar a filtragem da imagem da figura 17, contendo importação de bibliotecas, leitura de imagem, processamento, e por fim seu salvamento.

```
1 import Image
2 import ImageOps
3 import ImageFilter
4
5 img = Image.open("entrada.png")
6
7 img = ImageOps.autocontrast(img)
8 img = ImageOps.equalize(img)
9
10 img = img.filter(ImageFilter.FIND_EDGES)
11
12 img.save('saida.png', 'PNG')
```

Figura 19 - Código contendo filtro para reconhecimento de bordas

O resultado do filtro para reconhecimento de bordas pode ser analisado na figura 20. Esse filtro faz parte da classe de filtros passa-alta que realçam os pixels que se encontram em posições de maior discrepância. Nesse caso, nas margens dos corpos d'água, o valor do pixels que representam o solo tem seu valor relativamente distinto do pixel que representa a água.

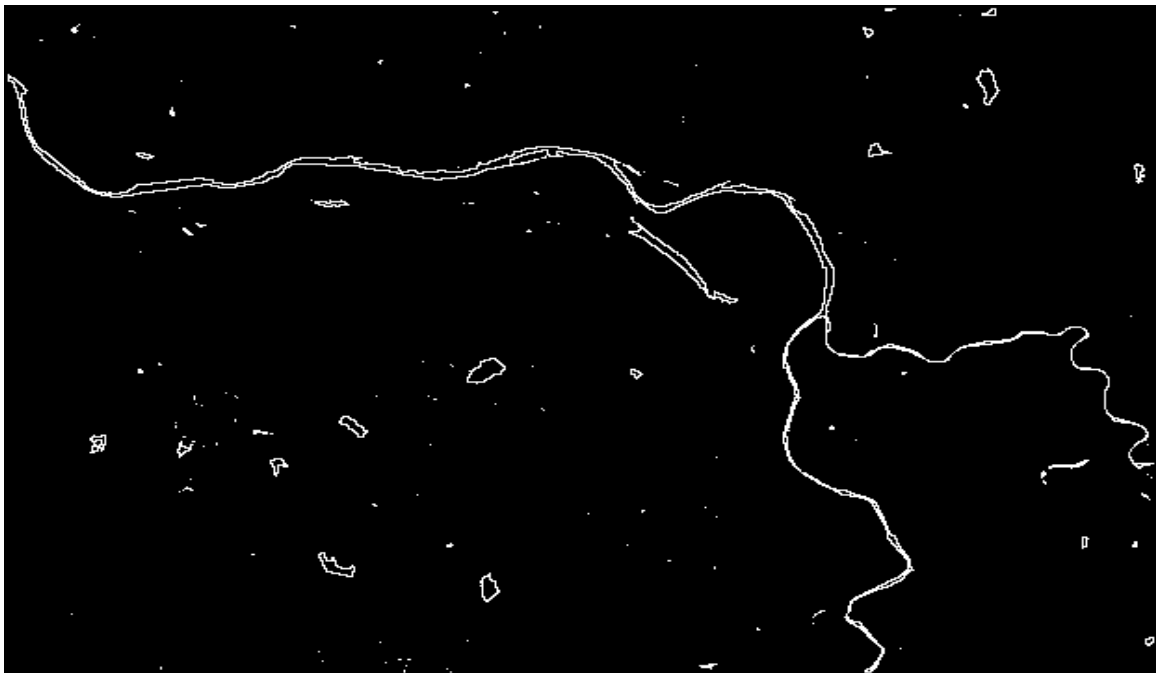


Figura 20 - Resultado do filtro para realce de bordas

Ainda que o resultado obtido pelo filtro de realce das bordas tenha atendido o

seu propósito, uma análise mais detalhada no produto gerado por esse mostrou aspectos que fizeram desta uma solução inviável, ou, no mínimo, de maior complexidade para serem solucionadas por um programa computacional.

Os aspectos relacionados ao parágrafo anterior podem ser analisados na figura 21, que apresenta a ampliação de uma região da figura 20 onde se podem identificar algumas de suas características. Nota-se que o traçado branco que atravessa a imagem da sua esquerda para a sua direita, tem início com uma sequência de dois ou três pixels agrupados, logo em seguida se dividem, se agrupam novamente, e por fim se dividem e se distanciam novamente até a lateral mais a direita da imagem.

O traçado, que foi mencionado anteriormente, representa um curso d'água, e esse, em sua variação de largura, é representado pelo traçado branco que se agrupa quando da pouca largura e se distancia quando do aumento dessa. Esse mesmo comportamento se repete quando estão presentes lagos, lagoas ou represas, que são representadas por polígonos que também podem ser visualizados na figura 21.



Figura 21 - Região ampliada da Figura 20

A ocorrência desses agrupamentos e desagrupamentos entre os traçados tornam o processo de vetorização mais complexo, pois quando estão agrupados há a necessidade de identificar quais pixels representam a borda e quais o interior, visto que,



tem-se a existência de agrupamentos com mais de dois pixels. Na segunda situação, onde há um distanciamento nos traçados, a complexidade é de identificar a parte interna e a parte externa do corpo d'água, sendo esse um requisito necessário para o sucesso no desenvolvimento do sistema proposto.

### 3.2.2 Filtro por frequência

Como proposta de solução para os problemas encontrados no produto gerado pelo filtro descrito no item anterior, adotou-se a técnica de filtragem que atua em uma faixa de frequência da imagem. Como já foi esclarecido na revisão de literatura, esse tipo de filtro se enquadra na categoria dos filtros passa-faixa, que atua realçando uma faixa de valores de interesse, e rejeitando as demais frequências fora dela.

A figura 22 apresenta um código similar ao usado para realizar a filtragem sobre a imagem mostrada na figura 17. Esse código apresenta a importação da biblioteca PIL representada pela linha 1. A abertura da imagem na linha 3 do código. A filtragem, propriamente dita, na linha 5, onde se adotou como faixa de interesse as frequências de imagens menores que 30, sendo que essas seriam elevadas ao valor 255 que representa tonalidade branca na imagem e, por fim, sua exibição e salvamento nas linhas seguintes.

```
1 import Image
2
3 imagem = Image.open("entrada.tif")
4
5 img = imagem.point(lambda i: i < 30 and 255)
6
7 img.show()
8
9 img.save("saida.png")
```

Figura 22 - Código do filtro passa-faixa

O resultado da utilização do filtro, anteriormente descrito, é exibido na figura 23, onde se podem identificar as principais diferenças desta em relação às técnicas passa-alta apresentadas na figura 20.

Uma das principais diferenças entre as duas técnicas é que o ponto de interesse para a realização do realce, que no caso do passa-alta é o nível de contraste entre os padrões de pixel. Já no caso do filtro passa-faixa, o ponto de interesse é uma faixa de valor da imagem, nesse caso os valores que representam o corpo d'água.

Com base no exposto anteriormente, fundamenta-se a adoção da técnica passa-faixa pelo fato de que esta representa o corpo d'água como pixels realçados, reduzindo assim a complexidade de se identificar interior e exterior das bordas de um corpo d'água, conclusão necessária para o sucesso da técnica de vetorização aqui proposta.

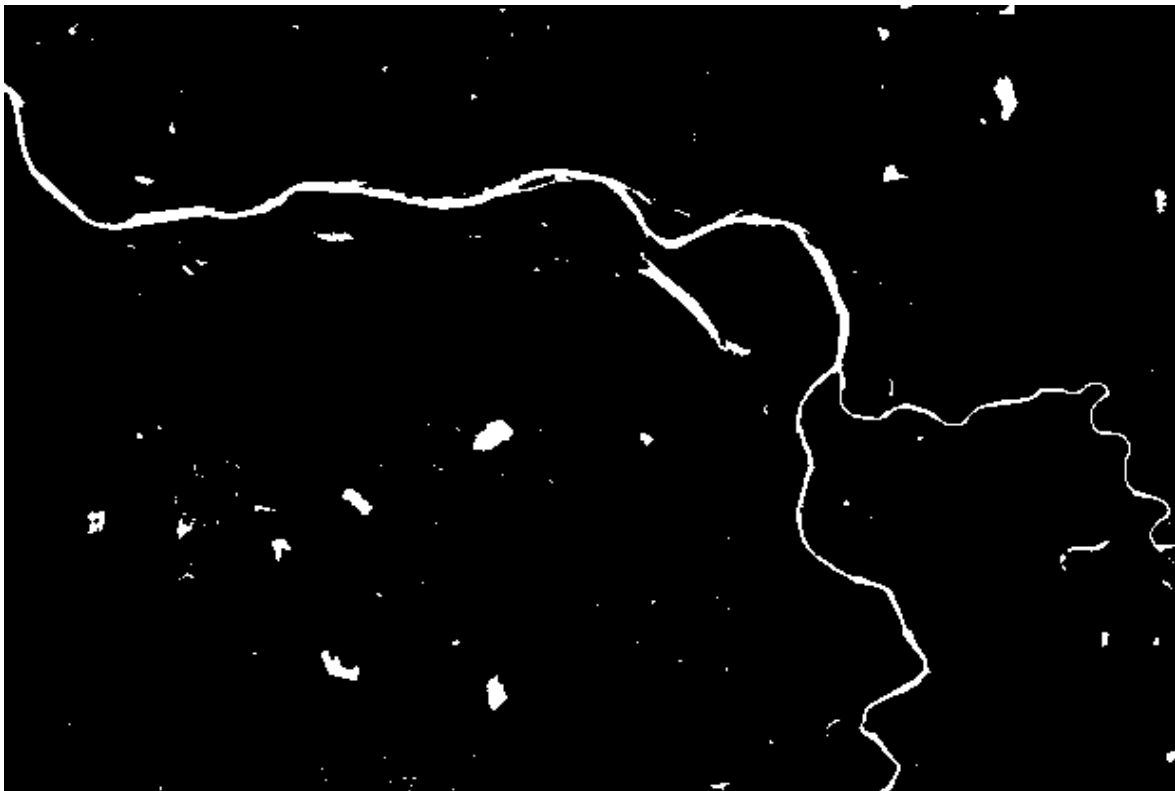


Figura 23 - Resultado do filtro passa-faixa

### 3.3 Desenvolvendo o *software*

O desenvolvimento do *software* como já mencionado anteriormente, adotou a linguagem de programação Python, tendo suas funcionalidades como ferramentas para resolução dos problemas que esse protótipo se propõe a resolver, tendo suas funcionalidades diagramadas no Anexo A.

Torna-se indispensável para a compreensão deste trabalho, que a localização espacial das áreas de interesse é um dos primeiros objetivos após estas estar devidamente identificadas na imagem. A técnica, utilizada para esse fim, foi uma varredura no produto gerado pelo filtro passa-faixa, sendo encaradas como ponto de interesse as áreas de pixel com valor de 255, este obtido após a filtragem, e representando áreas de corpo d'água.

A varredura realizada levou em consideração as características matriciais que são encontradas, tanto na imagem original, quanto no produto gerado pela sua filtragem. Dessa forma, a análise desempenhada em busca das áreas de interesse ocorre rastreando-se a imagem em um sentido da esquerda para a direita, analisando-se todas as colunas da primeira linha, e no sentido de cima para baixo, pois o processo realizado na primeira linha acaba se repetindo nas linhas consecutivas até que chegue ao fim da imagem.

Ao ser encontrado um pixel de interesse, por meio da varredura descrita anteriormente, o *software* faz chamada de uma funcionalidade secundária denominada de “delimita()”. Funcionalidade conhecida entre desenvolvedores de diversas linguagens de programação como “método”, e é usada para limitar especificidades do desenvolvimento, tornando-o de certa forma mais flexível, pois podem ter sua chamada a qualquer momento e de qualquer parte do código.

No ato da chamada do método “delimita()”, este deve receber as coordenadas (x, y) do primeiro pixel de interesse encontrado, com base no qual o método tem por objetivo registrar as bordas da área de interesse a que esse pixel pertence, sendo realizado no sentido horário até que se encontre o pixel inicial.

A figura 24 demonstra graficamente a interação das funcionalidades do *software*

de vetorização no que diz respeito ao rastreamento e delimitação das áreas de interesse descritas nos últimos parágrafos.

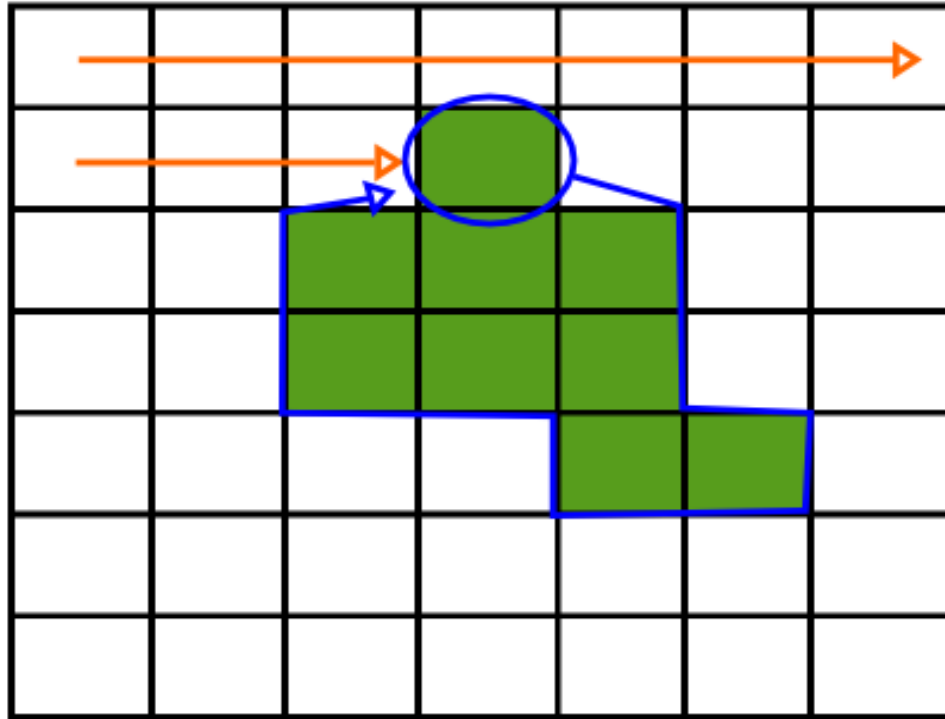


Figura 24 - Rastreamento de delimitação de áreas de interesse.

### 3.3.1 Máscara de Análise

Com base nas teorias para filtragem de imagens, adotou-se a utilização de uma máscara para que fosse possível realizar a delimitação da borda nas áreas de interesse encontradas. Os renomados filtros citados no item 3.4 se utilizam de máscaras de filtragem, que podem ser representadas por matrizes de diversas dimensões como, por exemplo, 3 x 3 ou 5 x 5, não sendo obrigatoriamente possuírem dimensões quadradas.

Geralmente, a utilização de máscaras é para fins de filtragem nas imagens *raster*, gerando posteriormente um produto igualmente *raster*, isso com base no cálculo dos valores de pixels presentes na imagem original e de valores presentes na máscara

que variam de acordo com seu propósito.

No desenvolvimento desse programa, utilizam-se uma máscara lógica no lugar de uma máscara numérica, sendo que essa adotou o conceito de vizinhança-8, tendo em vista que se tem possibilidade de adotar 8 direções diferentes a partir do pixel de interesse. A figura 25 representa as direções possíveis de serem adotadas como destino para a delimitação para das áreas analisadas.

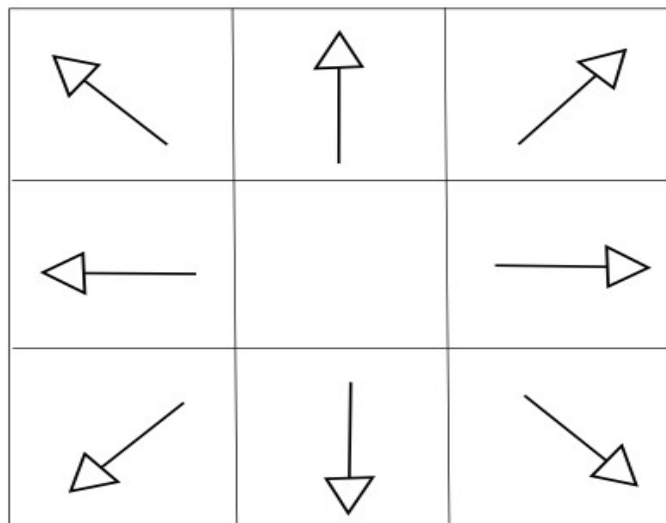


Figura 25 - Máscara de análise

A fim de se chegar a uma conclusão sobre a direção de pixel dentre as 8 possíveis a se tomar, construiu-se uma sequência lógica de testes, tendo o intuito de percorrer a vizinhança em nível oito do pixel de interesse no sentido anti-horário, com início no pixel superior direito e fim no pixel intermediário direito, possibilitando, dessa forma, o rastreamento e reconhecimento do próximo pixel de relevante a ser visitado.

A representação da sequência lógica de testes adotada é representada pela figura 26, sendo o símbolo central a representação de um dos pixels presentes na borda da área a ser vetorizada.

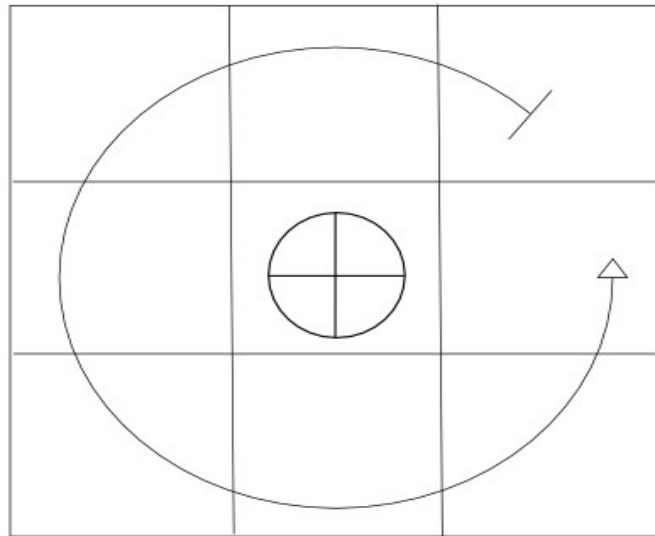


Figura 26 - Sequência lógica de teste

### 3.3.2 Versão Mono e Multi Polígono

Em um primeiro momento, após a definição da máscara a ser usada e da técnica de investigação que identificaria as bordas da imagem, criou-se uma imagem de menor complexidade para que fossem realizados os primeiros testes. Essa imagem pode ser vista na figura 27. Nessa mesma etapa, adotou-se a utilização de estrutura de dados que armazenassem os pontos pertencentes a borda da imagem.

A estrutura utilizada para armazenamento das coordenadas de interesse tem sua denominação como lista encadeada. Essa estrutura vem dos conceitos da computação e tem como principais características a flexibilidade de tipos de dados que podem armazenar e o fato de ser uma estrutura dinâmica, o que permite que essa estrutura ocupe apenas o espaço necessário em memória e, ainda, possa se expandir quando preciso.

As características citadas anteriormente fazem dessa uma estrutura ideal para armazenamentos de coordenadas de interesse, já que não se tem uma prévia de quantos pontos seriam. Sendo assim, tem-se a possibilidade de criar uma lista vazia e, sempre que necessário, adicionar as coordenadas encontradas até a conclusão dos

objetivos.

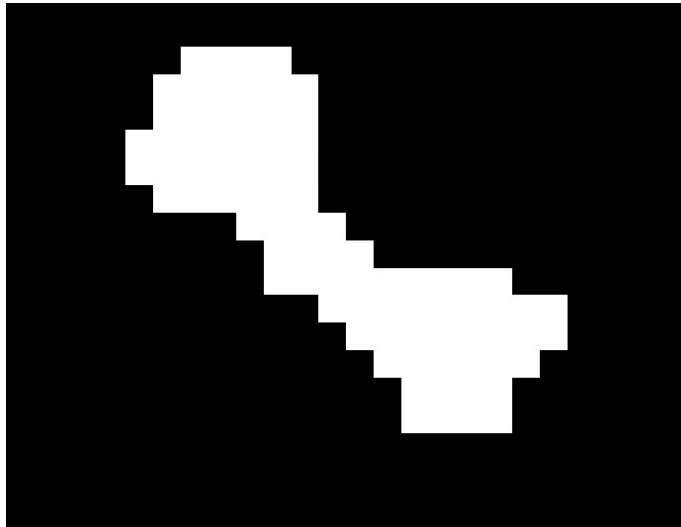


Figura 27 - Imagem criada para testes iniciais.

Após atingir o objetivo de armazenamento de todos os pontos que formam o polígono referente à imagem apresentada na figura 27, fez-se uso de uma classe Python denominada *shapefile* na sua versão 1.1.3. Com o auxílio dessa classe, foi possível a criação do arquivo *shape* que posteriormente é mostrada sobre a imagem de teste na figura 28.

Faz-se necessário esclarecer que, nos primeiros testes realizados, adotou-se por padrão armazenar como coordenada de referência, apenas a que representa o canto superior esquerdo de cada pixel. Sendo assim, obviamente, a área de interesse não foi totalmente coberta pelo polígono gerado por meio do protótipo de vetorização.

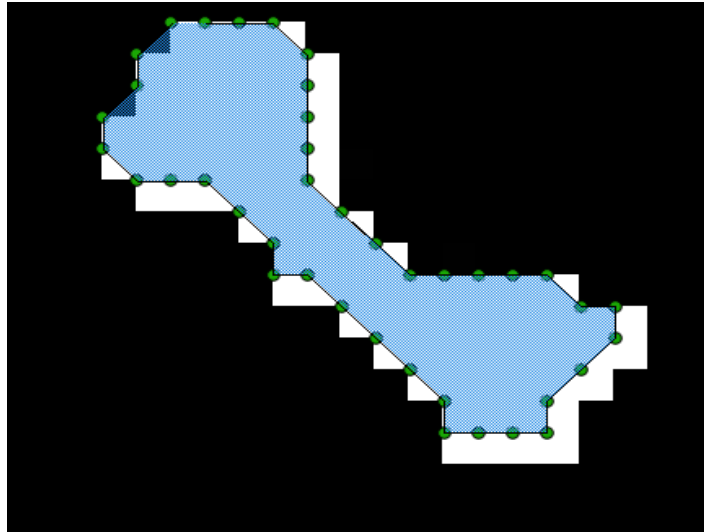


Figura 28 - Polígono plotado sobre área de interesse.

Como descrito anteriormente, em um primeiro momento, faz-se necessário realizar testes e imagem contendo apenas uma área de interesse, ocasionando a existência de apenas um polígono no arquivo *shape* criado. Como se sabe que, em casos reais, uma imagem de satélite pode registrar cenas com milhares de áreas de interesse, nesse caso proporcionando a existência de iguais milhares de polígonos para limitá-los de forma vetorial.

Com o intuito de proporcionar a evolução do protótipo, no sentido de atender às necessidades já expostas, construiu-se uma segunda imagem semelhante à figura 27, com o diferencial de que, nessa segunda, tem-se a existência de duas áreas de interesse.

A mudança proposta nessa segunda interação do protótipo não trouxe apenas mais um desafio, mas vários. Em um primeiro momento, tem-se que ter em mente que a mudança proposta não traz apenas a possibilidade da existência de dois polígonos, mas de inúmeros polígonos.

O fato de não se ter a certeza de quantas áreas de interesse poderiam-se encontrar em uma imagem, remete, novamente, às vantagens de se utilizar uma estrutura de armazenamento de dados dinâmica. Nesse caso a lista encadeada que já se descreveu no item 3.3.2.



Da mesma forma, como não se sabia quantos pontos um polígono poderia ter, não se sabe quantas áreas de interesse uma imagem pode possuir. Sendo assim, criou-se uma lista encadeada para armazenar listas encadeadas, tornando a manipulação, tanto de coordenadas, quanto de polígonos, uma ação variável, mas controlada, já que, essa estrutura retorna seus atributos sempre que necessário.

Um segundo ponto de complexidade, adicionado pelos múltiplos polígonos, é o fato de que, a cada área de interesse encontrada teria-se que chamar o método de delimitação. Isso não poderia acontecer quando a área de interesse já fosse uma das delimitadas, sendo que o método de varredura pode encontrar várias linhas e colunas da mesma área de interesse. Como solução para o problema exposto, adotou-se a utilização de uma verificação que identifica se o pixel de interesse encontrado já faz parte de uma das áreas anteriormente delimitadas.

#### 3.3.4 Ajuste de traçado

O ajuste de traçado constituiu-se como uma das etapas fundamentais no desenvolvimento desse protótipo. Nela realizaram-se alterações que tornaram possível a cobertura total das áreas de interesse pelo polígono gerado automaticamente pelo *software*. O traçado dos polígonos depende das coordenadas de imagem adicionadas ao arquivo *shape*, e, para que se tivesse o resultado esperado, foi necessária a verificação de direção dos pixels de interesse, pois, dependendo disso as coordenadas a serem adicionadas seriam em pontos diferentes do pixel já visitado.

Vale destacar que, nessa etapa do desenvolvimento, já se trabalha o programa sobre o produto do filtrado que se executou sobre a imagem real de satélite, representada pela figura 23. Condições adversas as já conhecidas foram encontradas e, para essas, foram desenvolvidas as soluções necessárias.

#### 3.3.5 *Buffer* variável

Para que se concebesse a proposta de um *buffer* variável, adotou-se a metodologia de desenvolvimento de um novo programa computacional. Esse novo programa, por sua vez, utilizou-se de todas as técnicas e funcionalidades existentes no primeiro *software*, sendo implementadas algumas modificações e adicionando funcionalidades a fim de atender o seu novo desafio, sendo que sua estrutura de funcionamento pode ser analisada no diagrama contido no Anexo B.

Como princípios adotados no desenvolvimento desse segundo programa, tem-se a definição das larguras dos cursos d'água e da distância de *buffer* adotada para cada um dos casos.

Para se realizar o cálculo das larguras do corpo d'água, adotou-se a metodologia de se identificar a direção a seguir sobre a borda da área de interesse. Com base nessa direção, calculou-se a largura em linha reta em um ângulo de  $-90^\circ$  da direção identificada, já visto que a direção de rastreamento da borda do corpo d'água ocorre no sentido horário.

Após a identificação da largura do corpo d'água no ponto analisado, identifica-se a distância de *buffer* adequado para a mesma, e se projeta o ponto do *buffer* a um ângulo de  $90^\circ$  da direção da borda, assim sendo, para fora da área de interesse que se está analisando.

Cabe frisar que a proposta de um *buffer* variável tem seu intuito principal no uso sobre cursos d'água, sabendo-se das demandas na área de geoprocessamento no uso de tais funcionalidades a fim de um ganho de desempenho, e aprimoramento do processo operacional na delimitação de APPs sobre cursos d'água.

Com o uso desta funcionalidade propõem-se uma inovadora forma de cálculo para as Áreas de Preservação Permanente, onde se pode entrar com a legislação vigente e suas determinações referentes as distâncias de preservação para cada largura dos cursos d'água. Sendo isso feito, o programa computacional realiza automaticamente a delimitação de acordo com as características do corpo d'água analisado.

## 4 RESULTADOS E DISCUSSÕES

Neste capítulo, discutir-se-á sobre os resultados obtidos e sobre o ponto de vista quantitativo e qualitativo. Em um primeiro momento, descreve-se sobre a relação tempo e precisão na vetorização realizada por usuário de sistema SIG, e da vetorização automatizada por meio do protótipo desenvolvido.

Além da relação tempo, também se considera a relação precisão como referencial de suma importância para o sucesso na geração de um produto final de qualidade.

### 4.1 Otimização de tempo e precisão

Tendo em vista os testes realizados com o *software* desenvolvido sobre a imagem representada pela figura 23, que é o produto da filtragem da imagem representada pela figura 17, chega-se a excelentes resultados no que diz respeito a tempo de vetorização e precisão da mesma.

No que se refere ao tempo de execução do programa, primeiramente, tem-se que entender melhor as características dos materiais envolvidos. A imagem utilizada possuía dimensões de 617 pixels de largura por 424 pixels de altura, totalizando 261.608 (duzentos e sessenta e um mil seiscentos e oito) pixels a serem analisados. A configuração básica do computador utilizado: processador Centrino Core™2 Duo 2.0, 3 GB de memória RAM 667, e com 3 M de cache.

Com base nas características, citadas anteriormente, obteve-se o tempo de aproximadamente 9 segundos para a realização da análise da imagem e criação do arquivo *shape* por parte do protótipo desenvolvido. Esse arquivo *shape* constituiu-se de 131 polígonos representando corpos d'água, sendo que, para a representação de cada um deles, foram necessárias 4 ou mais coordenadas da imagem, tendo em vista que esse deveria representar a conversão do dado *raster* para o dado vetorial,

contabilizando num total de 7.249 (sete mil duzentos e quarenta e nove) coordenadas.

Segundo INPE (2012), “Quanto maior for a precisão na digitalização do contorno do rio, maior será a precisão da delimitação da APP, por consequência, maior coerência com a medida no terreno”. Essa afirmação nos leva a refletir sobre as diferenças entre uma digitalização dirigida por um ser humano, ou por meio do uso de um sistema informatizado.

Analisando manuais de uso como o do próprio INPE (2012), onde se representa a forma de digitalização de rios por meio do *software* Spring, identifica-se que, dificilmente se consegue representar de forma fiel as bordas das áreas de interesse.

A digitalização de áreas de interesse é de suma importância para a qualidade do produto final em SIG, mas, se aplicado em grandes áreas, torna-se relativamente exaustivo e demorado para qualquer profissional da área. Sendo assim, faz-se compreensível a incansável procura por aprimoramento de sistemas existentes, e implementação de novos produtos e módulos de automação nos processos de SIG.

Nos testes realizados, o protótipo desenvolvido demonstrou digitalizar áreas de interesse com 100% de cobertura destas, seguindo fielmente a borda dos pixels pertencentes às mesmas. Nesse caso, o grande diferencial de qualidade, vem da classificação adotada para reconhecimento das áreas de interesse; quanto melhor a classificação, melhor a digitalização.

## **4.2 Manipulando *shapefile* com QGIS**

A manipulação do arquivo *shape* gerado automaticamente pode ser efetuado por diversos sistemas computacionais de SIG. Contudo durante os testes realizados utilizou-se o *software* Quantum GIS 1.4.0. Com auxílio desse sistema demonstramos o arquivo gerado com seus 131 polígonos como pode ser visto na figura 29.

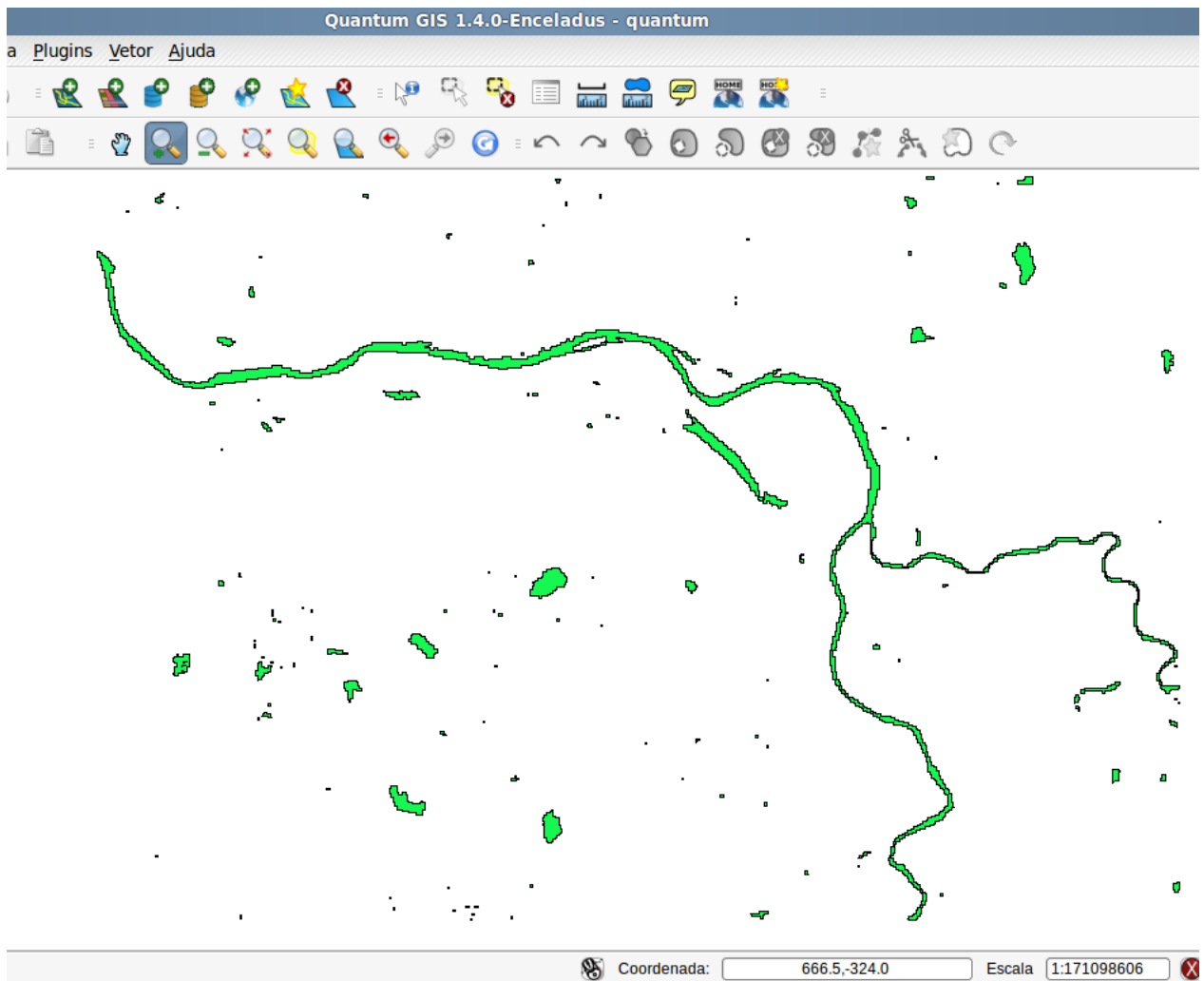


Figura 29 - Arquivo *shape* exibido pelo *software* Quantum GIS

#### 4.2.1 Dividindo shapefiles

Com o intuito de mostrar a delimitação de APP nas margens de corpo d'água com base no arquivo que foi gerado automaticamente, teve-se a necessidade de dividir as feições (polígonos) existentes, isso porque as regras que se aplicam a rios e cursos de água, não são as mesmas aplicadas a lagos, lagoas e reservatórios naturais ou artificiais.

Para esse fim, utiliza-se a funcionalidade selecionar feições, presente na barra de ferramentas do *software*, e apresentado pela figura 30 (também circulado na figura

31). Por meio dessa funcionalidade, foi possível a seleção de todas as feições e retirar a seleção da única feição que representa o curso de água.

Contendo apenas os lagos, lagoas e reservatórios de água selecionadas, fez-se a utilização de uma segunda funcionalidade do *software* Quantum GIS que se denomina Salvar seleção como *Shapefile*, que pode ser encontrada no menu Camada e possibilita a criação de um novo arquivo *shape* contendo apenas as feições previamente selecionadas.

Da mesma forma, pode-se criar um arquivo *shape* contendo apenas as feições que representem cursos de água, podendo ser generalistas quanto à questão das larguras desses ou, até mesmo, realizarem-se divisões no que tange à existência de arquivos divididos pela largura dos rios aos quais estão relacionados.

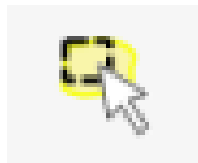


Figura 30 - Imagem da funcionalidade selecionar feições

#### 4.2.2 Criando delimitação APP

Para criar uma delimitação de APP com base na divisão do arquivo original de digitalização descrita no capítulo anterior, basta que se realize a importação do novo arquivo *shape* contendo as feições desejadas para que se faça uso da funcionalidade, presente nos mais diversos sistemas de SIG, denominada como *Buffer*. No caso do Quantum GIS, localizado na sequência de menu “Vetor / Ferramentas de Geoprocessamento / *Buffer(s)*”.

O *buffer* é utilizado quando se deseja criar uma delimitação vetorial de distância, editável pelo usuário, sobre um dado também vetorial. A figura 31 demonstra a utilização do arquivo *shape* que contém as feições de lagos, lagoas e reservatórios,

sendo que sobre essas foi gerado um *buffer* contendo uma distância padrão das feições citadas anteriormente.

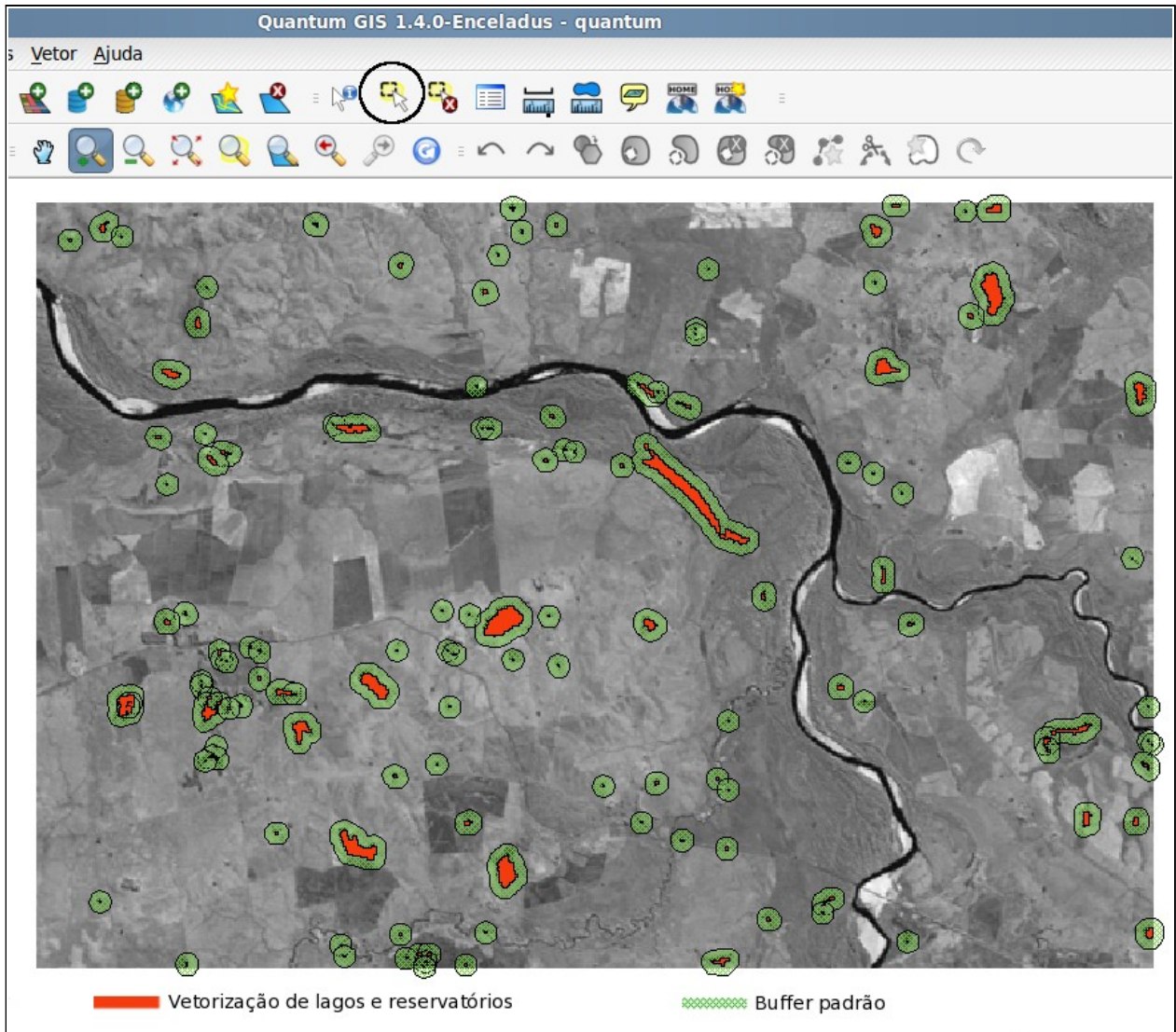


Figura 31 - Representação de lâminas d'água e *buffer* destas.

Da mesma forma que se criou a representação das lâminas d'água e seus *buffers* de proteção para os lagos, lagoas e reservatórios d'água, na figura 32 representa-se a mesma área contendo a exibição do arquivo *shape* do curso d'água, e gerou-se o *buffer* de proteção para o mesmo. Salienta-se que a exibição das feições de

forma separada tem intuito meramente didático, não impedindo que se manipulem ambos os arquivos ao mesmo tempo.

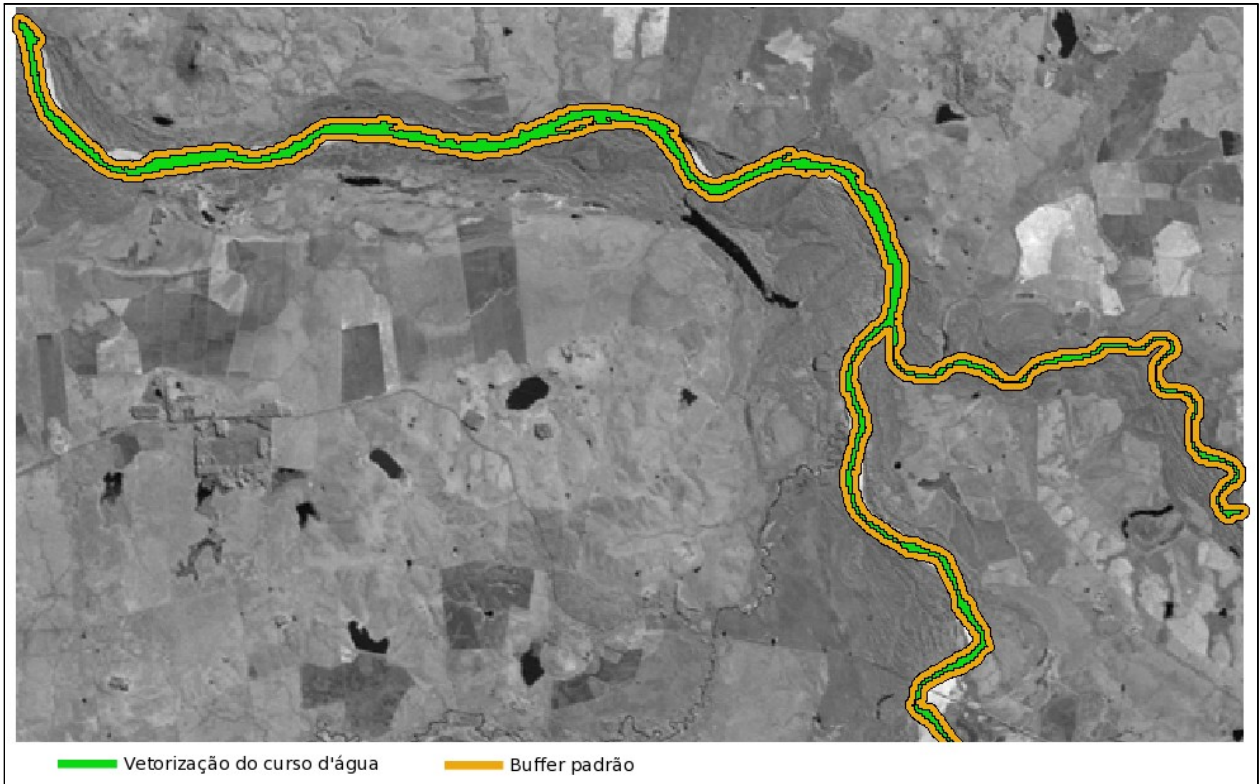


Figura 32 - Demonstração do *buffer* referente ao curso d'água

Sabendo-se que, como um dos objetivos deste trabalho, tinha-se a proposta de implementar um *buffer* variável de acordo com a largura dos cursos d'água, a figura 33 exibe a representação de um curso d'água com diferentes larguras, principalmente em dois pontos da imagem onde ganham notoriedade com bem mais do que um pixel de largura.

A figura citada, no parágrafo anterior, foi criada com o objetivo específico de servir como base de testes para o desenvolvimento do programa que gera o *buffer* variável. Além de ter como característica a variação de largura, projetou-se uma bifurcação no curso da área de interesse, para que fosse possível analisar o comportamento do programa em situações como essa.





Figura 33 - Imagem que representa curso de água

Como produto do processamento realizado sobre a imagem de teste, chega-se ao que está sendo representado pela figura 34. Nela pode-se identificar o aumento do *buffer* adotado que ocorre, respectivamente, sobre as áreas onde se tem uma maior largura facilmente identificável.

Os cálculos adotados foram baseados na resolução espacial da imagem, e sobre as especificações de delimitação referente a corpo d'água da lei nº 12.651, de 25 de maio de 2012. Cabe salientar que, tanto a resolução espacial, quanto as distâncias a serem adotadas pelo *buffer*, são atributos variáveis podendo ser modificados de acordo com as necessidades do usuário.

Como padrões de distâncias adotadas para o desenvolvimento do *buffer* variável, segue abaixo especificações retiradas no Artigo 4º da lei citada no parágrafo

anterior (BRASIL, 2012).

- a) 30 (trinta) metros: cursos d'água de até 10 (dez) metros de largura;
- b) 50 (cinquenta) metros: cursos d'água de 10 (dez) a 50 (cinquenta) metros de largura;
- c) 100 (cem) metros: cursos d'água de 50 (cinquenta) a 200 (duzentos) metros de largura;
- d) 200 (duzentos) metros: cursos d'água de 200 (duzentos) a 600 (seiscentos) metros de largura;
- e) 500 (quinhentos) metros: cursos d'água superiores a 600 (seiscentos) metros;

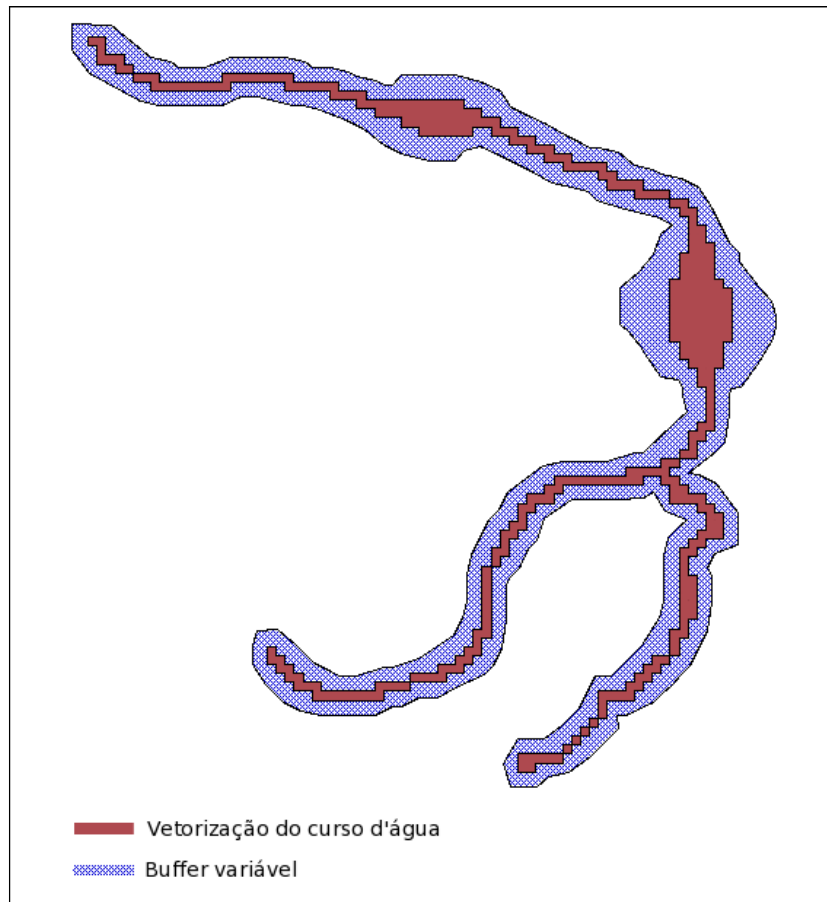


Figura 34 - Representação do *buffer* variável

A principal diferença entre o *buffer* variável que é proposto e um padrão construído a partir da média de largura do curso d'água é apresentada na figura 35. Nessa representação, identifica-se a variação da distância de delimitação adotada pelo

*buffer*, isto sem nenhuma interação do usuário durante o processo de construção do mesmo.

Sobre o *buffer* variável, plotou-se um *buffer* padrão utilizando da funcionalidade do Quantum GIS para esse fim, tornando possível uma melhor compreensão do funcionamento da técnica padrão e da técnica aqui proposta. Para se chegar a um resultado semelhante por meio do formato padrão dos *buffers*, seria necessário o fatiamento da digitalização original referente ao corpo d'água pela sua largura, e posteriormente a criação individual do *buffer* destes.

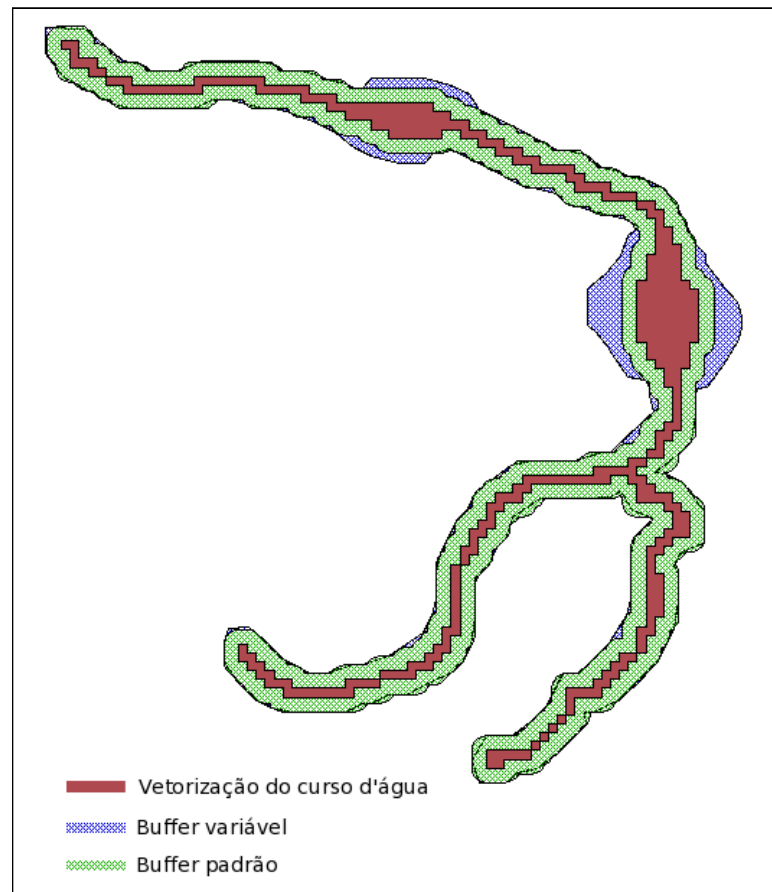


Figura 35 - *Buffer* genérico sobre *buffer* variável

## 5 CONCLUSÕES E RECOMENDAÇÕES

Ao término deste trabalho, constata-se que tanto o protótipo de vetorização de áreas de interesse, quanto o protótipo para criação de *buffer* variável sobre corpos d'água tiveram sua eficiência comprovada. Os testes realizados demonstraram um ganho significativo no que diz respeito ao desempenho e a precisão do produto gerado por parte dos protótipos desenvolvidos.

Espera-se que os protótipos desenvolvidos durante este trabalho sirvam de inspiração e até mesmo de base para que se dê continuidade na “caminhada” rumo à inovação e automação na área de SIG, tanto no contexto nacional quanto internacional.

Acreditamos que somente com ênfase na pesquisa, e por meio da análise e valorização do olhar crítico sobre as técnicas aplicadas na solução de problemas, é que se torna possível o constante aprimoramento humano e tecnológico.

A partir das viabilidades expressas neste trabalho, torna-se possível lançar olhares sobre novos horizontes. Os resultados obtidos apontaram para a possibilidade de se implementarem trabalhos futuros com o objetivo de tornar as funcionalidades apresentadas até o momento em produtos de *software*, seja em uma ferramenta autônoma, ou como módulos (partes) de programas já existentes.

## REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Lei n. 12.651, de 25 de maio de 2012. Dispõe sobre a proteção da vegetação nativa. **Casa Civil**. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_Ato2011-2014/2012/Lei/L12651.htm](http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2012/Lei/L12651.htm)>. Acesso em: 11 fev. 2012.

BORGES, Luiz Eduardo. **Python para Desenvolvedores**. Rio de Janeiro: Edição do Autor, 2010.

CHRISMAN, N., *Exploring Geographic Information Systems*. New York, John Wiley&Sons, 1997. Disponível em: <http://www.dpi.inpe.br/cursos/ser300/referencias.html> Acesso em: 08 jun. 2012.

FIGUEIREDO, Divino. **Conceitos Básicos de Sensoriamento Remoto**. 2005. Disponível em: [http://www.conab.gov.br/conabweb/download/SIGABRASIL/manuais/conceitos\\_sm.pdf](http://www.conab.gov.br/conabweb/download/SIGABRASIL/manuais/conceitos_sm.pdf). Acesso em: 08 jun. 2012.

FREIRE, Dermival Valim. **Gestão de Marketing**. 2010. Disponível em: <http://www.fag.edu.br/professores/freire/Cont%E1beis/Mat%20Apoio%2001%20Gest%E3o%20de%20Marketing-Ctb404.doc>. Acesso em: 06 mai. 2012.

GRILO, Rui Manuel Boletto. **Teoria da Gestão e da Complexidade**. 1996. Disponível no endereço: <http://www.manuelgrilo.com/rui/complexidade/trabalho.pdf>. Acesso em: 05 mai. 2012.

GUGIK, Gabriel. **SDK focado na construção de plataformas de desenvolvimento e frameworks para aplicativos**. 2010. Disponível em: <http://www.baixaki.com.br/linux/download/eclipse.htm>. Acesso em: 05 jun. 2012.

INPE. **Catálogo de Imagens do Instituto Nacional de Pesquisa Espaciais**. Disponível em: <http://www.dgi.inpe.br/CDSR/>. Acesso em: 15 mar. 2011.

INPE. **Tutorial – Mapa de distância**. Disponível em: [http://www.dsr.inpe.br/vcsr/files/14-mapa\\_distancias.pdf](http://www.dsr.inpe.br/vcsr/files/14-mapa_distancias.pdf). Acesso em: 25 fev. 2012.

LUNDH, Fredrik. ELLIS, Matthew. **Python Imaging Library Overview**. 2002. PIL 1.1.3 Manual. Disponível em: <http://www.pythonware.com/media/data/pil-handbook.pdf>. Acesso em: 06 mai. 2012.

MAUBORGNE, R. **[Entrevista disponibilizada em 31 de agosto de 2010, a Internet]**. Disponível em: <http://exame.abril.com.br/negocios/gestao/noticias/nem-toda-inovacao-boa-diz-renee-mauborgne-592813>>. Acesso em: 12 jun. 2012.

MELO, D. H. **Uso de dados Ikonos II na análise urbana: testes operacionais na zona leste de São Paulo**. São José dos Campos: INPE, 2002.

MIRANDA, José Iguelmar. **Fundamentos de Sistemas de Informações Geográficas**. Brasília, DF: Embrapa Informação Tecnológica, 2010.

MIRANDA, José Iguelmar. **Processamento de imagens digitais: prática usando Java™**. Campinas: Embrapa Informática Agropecuária, 2006.

PEDRINI, Hélio. SCHWARTZ, William Robson. **Análise de Imagens Digitais: princípios, algoritmos e aplicações**. São Paulo: Editora Thomson Learning, 2008.

PINA, Maria de Fátima. SANTOS, Simone M. **Conceitos básicos de Sistemas de Informação Geográfica e Cartografia aplicados à saúde**. Brasília: OPAS, 2000.

REZENDE, Cátia Gontijo. **Conceitos e Perspectivas em Sistemas de Informação e de Apoio a Tomada de Decisão**. 2003. Disponível em: <http://www1.serpro.gov.br/publicacoes/tematec/PUBTEM68.htm>. Acesso em: 06 mai. 2012.

SETZER, Valdemar W. **Dado, Informação, Conhecimento e Competência**. 2002. Disponível em: <http://www.ime.usp.br/~vwsetzer>. Acesso em: 06 mai 2012.

SOUZA, C. F. et al. **Visão Computacional**. In: EBAH rede social para o compartilhamento acadêmico. Taquaritinga: SP, 2010. Disponível em: <http://www.ebah.com.br/content/ABAAABeUIAB/visao-computacional>>. Acesso em: 07 jun. 2012.

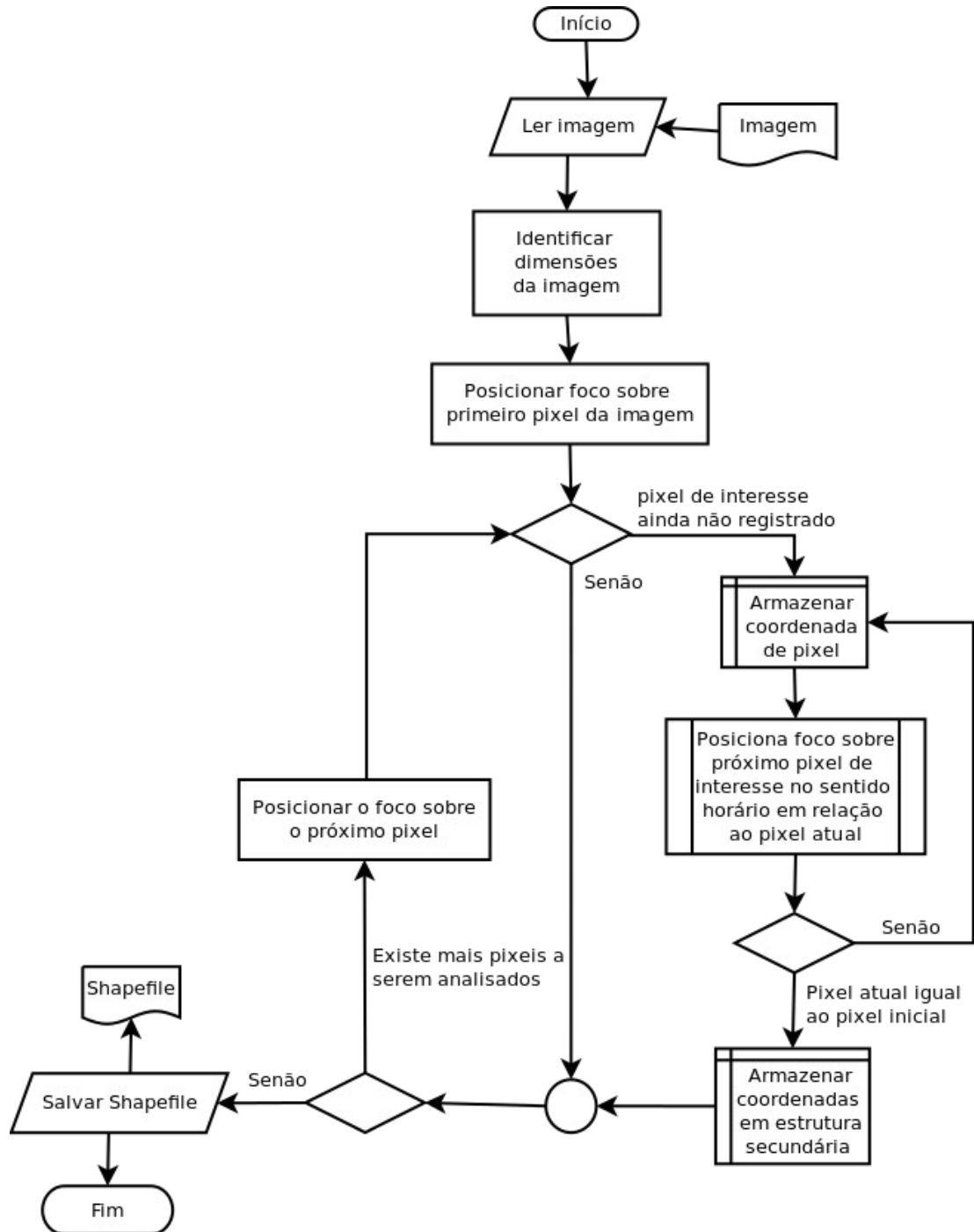
VIEIRA, Noemia R. **O espaço geográfico em questão: Uma experiência de**

**renovação teórico-metodológica no ensino de geografia.** 2003. Disponível em:  
<http://www2.fct.unesp.br/nera/publicacoes/ProdNoemia3.pdf>. Acesso em: 28 mar. 2012.

## **ANEXOS**



## Anexo A – Diagrama do algoritmo de vetorização



## Anexo B - Diagrama do algoritmo de *buffer* variável

