

ALEKSANDER MADURO FRANÇA

**DESENVOLVIMENTO DE INTERFACE GRÁFICA EM IDL
PARA IMPLEMENTAÇÃO DE ALGORITMOS DE DETECÇÃO
DE MUDANÇAS EM IMAGENS MULTI-TEMPORAIS**

Monografia de graduação apresentada ao Departamento de
Ciência da Computação da Universidade Federal de Lavras
como parte das exigências do curso de Ciência da
Computação para a obtenção do título de Bacharel em Ciência
da Computação

LAVRAS
MINAS GERAIS - BRASIL
2007

ALEKSANDER MADURO FRANÇA

**DESENVOLVIMENTO DE INTERFACE GRÁFICA EM IDL
PARA IMPLEMENTAÇÃO DE ALGORITMOS DE DETECÇÃO
DE MUDANÇAS EM IMAGENS MULTI-TEMPORAIS**

Monografia de graduação apresentada ao Departamento de
Ciência da Computação da Universidade Federal de Lavras
como parte das exigências do curso de Ciência da
Computação para a obtenção do título de Bacharel em Ciência
da Computação

Área de Concentração:

Interface Gráfica

Orientador:

Prof. Luis Marcelo Tavares de Carvalho

LAVRAS
MINAS GERAIS - BRASIL
2007

Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca Central da UFLA

França, Aleksander Maduro

Desenvolvimento de Interface Gráfica em IDL para Implementação de Algoritmos de Detecção de Mudanças em Imagens Multi-temporais. / Aleksander Maduro França. Lavras – Minas Gerais, 2007. 48p: il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Sensoriamento Remoto. 2. IDL. 3. Interface Gráfica. 4. Detecção de Mudança. FRANÇA, A. M. II. Universidade Federal de Lavras. III. Título

ALEKSANDER MADURO FRANÇA

**DESENVOLVIMENTO DE INTERFACE GRÁFICA EM IDL
PARA IMPLEMENTAÇÃO DE ALGORITMOS DE DETECÇÃO
DE MUDANÇAS EM IMAGENS MULTI-TEMPORAIS**

Monografia de graduação apresentada ao Departamento de
Ciência da Computação da Universidade Federal de Lavras
como parte das exigências do curso de Ciência da
Computação para a obtenção do título de Bacharel em Ciência
da Computação

Aprovada em ____ de Julho de 2007

Prof. André Luiz Zambalde

Prof. André Vital Saúde

Prof. Luis Marcelo Tavares de Carvalho
(Orientador)

LAVRAS
MINAS GERAIS - BRASIL

*Dedico este trabalho a Deus e a meus
pais queridos: Saul e Silvani.*

AGRADECIMENTOS

A Deus por me proporcionar a obtenção deste título e por tudo que tenho conquistado. Ao meu pai que tanto se sacrificou, à minha mãe querida pelo constante apoio e preocupação, à Glorinha que esteve sempre ao meu lado. E aos amigos pelos momentos compartilhados.

DESENVOLVIMENTO DE INTERFACE GRÁFICA EM IDL PARA IMPLEMENTAÇÃO DE ALGORITMOS DE DETECÇÃO DE MUDANÇAS EM IMAGENS MULTI-TEMPORAIS

Resumo: A preocupação com o monitoramento das florestas tem ganhado força, e um dos principais motivos é o dilema atual do aquecimento global. Uma das maneiras de se monitorar uma grande cobertura de área é utilizando técnicas de sensoriamento remoto para coletar imagens multi-temporais de uma mesma região e analisá-las através de um algoritmo de detecção de mudanças. Neste trabalho foi desenvolvida uma interface gráfica utilizando a linguagem de programação IDL, e nesta interface gráfica foi adaptado um algoritmo de detecção de mudança implementado na linguagem IDL. A implementação da interface gráfica em IDL mostrou que a linguagem é de grande potencial para desenvolvimentos de interfaces gráficas e de rotinas na área de processamento de imagens digitais, e merece ser mais bem explorada.

Palavras-chave: sensoriamento remoto, idl, interface gráfica, detecção de mudança.

DEVELOPMENT OF GRAPHICAL INTERFACE IN IDL FOR IMPLEMENTATION OF CHANGE DETECTION ALGORITHM IN MULTITEMPORAL IMAGERY

Abstract: The concern with the monitoring of the forests has been increasing, and one of the main reasons is the global warming. One of the ways of monitoring a great area covering is by using remote sensing techniques to collect multi-temporary images of a same area, and analyzing them through an algorithm of change detection. In this work, a graphical user interface has been developed using the IDL programming language, and into this graphical interface, an algorithm of change detection written in IDL has been adapted. Development of the graphical interface in IDL showed that IDL programming language is of great potential for developments in the area of digital image processing, and it deserves to be better explored.

Keywords: remote sensing, idl, graphical interface, change detection.

SUMÁRIO

SUMÁRIO.....	ii
LISTA DE FIGURAS	iv
1. INTRODUÇÃO.....	1
1.1. Considerações Iniciais.....	1
1.2. Objetivo Geral.....	2
1.4. Organização do Trabalho	3
2. REFERENCIAL TEÓRICO.....	4
2.1. Sensoriamento Remoto	4
2.1.1. Sensores.....	5
2.1.2. Sensores Remotos.....	5
2.2. Reflectância.....	7
2.3. Imagem Digital.....	7
2.4. Detecção de Mudanças.....	10
2.4.1. Um Exemplo de Detecção de Mudança	11
2.4.2. Algoritmo de detecção de mudanças	12
2.5. Interface Homem Máquina	13
2.5.1. Usabilidade	15
2.5.2. GUI – Graphical User Interface	15
2.6. A linguagem IDL	16
2.7. O Software ENVI.....	17
3. METODOLOGIA.....	20
3.1. Tipo de Pesquisa	20
3.2. Procedimento.....	20
3.2.1. Estudo do Algoritmo de Detecção de Mudanças	21
3.2.2. Análise do ENVI e das Suas Aplicações.....	22
3.2.3. Estudo da Linguagem IDL	22
3.2.4. Estudo das Aplicações com Widgets em IDL	22
4. RESULTADOS E DISCUSSÃO	25
4.1. O Desenvolvimento da Interface.....	25
4.2. As Rotinas Criadas.....	25
4.3. A Rotina monografia.pro	26

4.4. A Rotina monografia_event	28
4.5. A Escolha das Imagens	28
4.6. Disposição das Imagens na Interface	30
4.7. Campo de Seleção das Bandas	30
4.8. A escolha do IDLDE para desenvolvimento.....	31
4.9. O controle dos widgets.....	32
4.10. Janela de escolha das imagens	32
4.11. Importâncias Apresentadas	32
5. CONCLUSÕES	34
5.1. Considerações Finais.....	34
5.2. Proposta para Trabalhos Futuros.....	34
6. REFERÊNCIAS BIBLIOGRÁFICAS	35

LISTA DE FIGURAS

Fonte: Figura 2.1: Espectro eletromagnético.....	4
Figura 2.3: Exemplo de convenção dos eixos para representação de imagens digitais.....	8
Figura 2.4: Seção Enlarguecida	9
Figura 2.5: Exemplo de um Histograma (esquerda) e sua imagem relacionada (direita) ...	10
Figura 2.6: Primeira e segunda imagem	12
Figura 2.7: Resultante à esquerda e com realce à direita.....	12
Figura 2.8: Subtração entre 2 bandas de uma imagem .dat	13
Figura 2.9: Diagrama do Projeto de Interfaces com o usuário.	14
Figura 2.10: Interface IDLDE	17
Figura 2.11: Interface do GUIBuilder	18
Figura 2.12: Interface do ENVI 4.3.....	18
Figura 2.13: ENVI Zoom	19
Figura 3.1: widget_button	23
Figura 3.2: widget_label.....	23
Figura 3.3: widget_draw.....	23
Figura 3.4: widget_text.....	24
Figura 3.5: widget_tree.....	24
Figura 4.1: Interface do Software Desenvolvido.....	25
Figura 4.2: A Base topo da aplicação	26
Figura 4.3: Exemplo da hierarquia das bases	26
Figura 4.4: Exemplo da Identação.....	27
Figura 4.5: Criação da estrutura info e do ponteiro pinfo	27
Figura 4.6: Parte do Código que recebe os valores do usuário.....	28
Figura 4.7: Parte do Código que mostra informações dos valores dos pixels das imagens	28
Figura 4.8: Botão de Seleção das Imagens mostrada à esquerda	29
Figura 4.9: Janela para escolher imagens	29
Figura 4.10: Imagem T1, Imagem T2 e a imagem resultante.....	30
Figura 4.11: Campo de escolha	31

1. INTRODUÇÃO

1.1. Considerações Iniciais

Recentemente, discussões relacionadas ao meio ambiente ganharam força, principalmente com a questão do aquecimento global, em que a temperatura média do planeta está aumentando drasticamente. As razões que contribuem para este aquecimento vão desde a poluição provocada pelas indústrias e por veículos automotivos, até o desmatamento das nossas florestas.

A melhor maneira de se estudar uma grande cobertura de área, como o Estado de Minas Gerais é a partir das técnicas de sensoriamento remoto, onde ocorre o mapeamento e o monitoramento necessário para tal cobertura de área. A técnica de sensoriamento remoto coleta, através das imagens digitais capturadas, informações importantes sobre várias variáveis, como o solo, a vegetação, relevo, água e erosão. Estas informações são tratadas e analisadas por profissionais da área de sensoriamento remoto, e o conhecimento adquirido é utilizado para solução dos problemas encontrados.

A técnica de coleta de imagens de sensoriamento remoto nos permite obter imagens de uma mesma área em tempos diferentes, mais conhecidas como imagens multi-temporais. As imagens obtidas são analisadas para se obter as detecções de mudanças ocorridas entre elas. A análise é feita a partir de um algoritmo de detecção de mudança. As mudanças detectadas e extraídas servem para atualizar uma base de dados espacial. Com uma base de dados de informações antigas e informações novas, há a possibilidade de se fazer comparações e concluir sobre as mudanças ocorridas em qualquer cobertura de área durante o intervalo de tempo em que as imagens foram coletadas.

Todo o processo proposto acima vem sendo desenvolvido anualmente em um projeto do Laboratório de Estudos em Manejo Florestal – LEMAF, situado no Departamento de Ciências Florestais da Universidade Federal de Lavras. O projeto utiliza a ferramenta de processamento de imagens ENVI para executar o algoritmo de detecção de mudanças, o qual foi desenvolvido na linguagem de programação IDL.

O IDL é um ambiente de trabalho completo para análise e visualização interativa de dados de qualquer natureza. Uma das principais vantagens do IDL para processamento de imagens é o fato de ele ser orientado a vetor, ou seja, processa dados vetoriais com

facilidade. As imagens de sensoriamento remoto são digitais, ou seja, possuem uma estrutura matricial bidimensional. A linguagem IDL possibilita uma maior flexibilidade quando se trabalha com imagens digitais. Além de possuir uma vasta biblioteca de funções específicas para tratamento de imagens.

O IDL possui ferramentas avançadas de representações gráfica, matemática e estatística. As funções podem ser criadas e executadas dentro do próprio ambiente de desenvolvimento. Pode-se usar uma interface de linha de comando a qual requer a inserção de comandos pelo teclado, o que torna a tarefa do usuário mais trabalhosa. Uma alternativa mais sucinta é construir interfaces gráficas para rodar uma seqüência de linhas de códigos. O IDL possui ferramentas específicas para construção de interfaces gráficas com controles de *widgets*. Os *widgets* nos possibilitam construir estruturas de interface com botões, direcionamentos de eventos, ícones e rótulos.

Com um sistema de detecção de mudanças que utiliza imagens digitais coletadas de sensoriamento remoto e um algoritmo de detecção de mudanças desenvolvido em IDL percebeu-se que a utilização de uma interface gráfica seria de grande contribuição para o tratamento das detecções de mudanças, uma vez que a usabilidade da mesma seja atingida. Quando se projeta uma interface gráfica para resolver um problema particular, somente serão desenvolvidas funcionalidades com objetivo de resolver tal problema. Isto torna a execução mais eficiente.

1.2. Objetivo Geral

Este trabalho tem como objetivo desenvolver uma interface gráfica de interação com o usuário, com o intuito de tornar o uso do sistema de detecção de mudanças mais simples, amigável e ao mesmo tempo eficiente, utilizando funcionalidades específicas que não são confusas ao usuário.

Aproveitar o recurso da linguagem IDL onde a programação, além de ser orientada a arrays para o processamento de imagens digitais, ela também é orientada a eventos, o que possibilita a construção da interface gráfica que interaja com o usuário.

Criar uma interface gráfica amigável para uso em análises de detecção de mudanças entre duas imagens de sensoriamento remoto. Torná-la disponível para ser utilizada no Departamento de Ciências Florestais da Universidade Federal de Lavras – MG.

Disponibilizar a interface de modo que, a partir da sua primeira versão, nela possam ser acrescentadas mais funcionalidades para tratamento e processamento de imagens digitais em trabalhos futuros.

1.4. Organização do Trabalho

O presente capítulo compõe a introdução do trabalho, com seus objetivos e relevância.

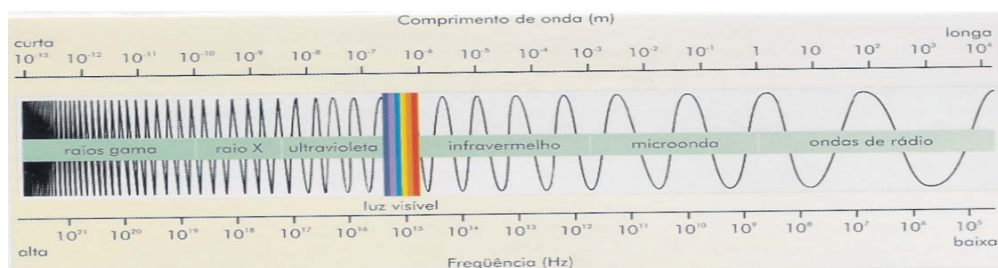
No Capítulo 2, são descritos os principais conceitos relacionados ao sensoriamento remoto, como a reflectância é e interpretada nas imagens de sensoriamento remoto através dos valores dos pixels, assim como os conceitos de detecção de mudança em imagens digitais. A partir daí foi relatada a idéia básica de interface homem-máquina, com a questão da usabilidade, e qual a importância de se construir uma interface gráfica para o usuário, o GUI – *Graphical User Interface*. Foram descritas também as ferramentas de desenvolvimento ENVI e o IDL. No Capítulo três é apresentada a metodologia usada para o desenvolvimento deste trabalho. No Capítulo quatro são apresentados os resultados e discussões do desenvolvimento do algoritmo. Por fim, são feitas as considerações finais no Capítulo 5.

2. REFERENCIAL TEÓRICO

2.1. Sensoriamento Remoto

Carvalho (2005) define uma imagem de sensoriamento remoto como sendo a representação de uma descrição de uma região na superfície terrestre em um ponto específico no tempo. Segundo Moreira (2001), sensoriamento remoto é a tecnologia que permite obter imagens e outros tipos de dados da superfície terrestre, através da captação do registro da energia refletida ou emitida pela superfície. Conforme Smith (2006), a ciência do sensoriamento remoto inclui observações espaciais, aéreas e de satélites das superfícies e atmosferas dos planetas no nosso sistema solar. Um bom exemplo de sensoriamento é o sistema visual humano, onde a retina exemplifica o sensor remoto, e ela registra imagens para serem processadas pelo cérebro. De acordo com Buiten (1996), o termo “sensoriamento remoto” surgiu da instrumentação, técnica e métodos utilizados para coleta de dados sobre a superfície da terra através da captura de imagens ou retratos de parte desta superfície.

A distribuição da radiação eletromagnética, por regiões, segundo o comprimento de onda e a frequência, recebe o nome de espectro eletromagnético. Para Florenzano (2002), o espectro abrange desde curtos comprimentos de onda, como os raios cósmicos e os raios gama, de alta frequência, até longos comprimentos de onda como as ondas de rádio e TV, de baixa frequência. O sensor pode atuar em diversas faixas do espectro eletromagnético (Figura 2.1), chamadas de bandas espectrais. Em cada banda, uma determinada característica do ambiente é mais bem visualizada.



Fonte: **Figura 2.1:** Espectro eletromagnético.

Souza (2004)

2.1.1. Sensores

Segundo Carvalho (2005), sensores são equipamentos utilizados para coletar dados brutos, geralmente na forma de grandezas físicas como temperatura, umidade, comprimento de onda e frequência. Um sensor é geralmente definido como um dispositivo que recebe e responde a um estímulo ou um sinal. Como o sinal é uma forma de energia, os sensores podem ser classificados de acordo com o tipo de energia que detectam. Por exemplo: sensores de luz, de temperatura, de calor, de radiação. Muitos destes equipamentos coletam os dados por meio de contato direto com o objeto estudado. Um termômetro, por exemplo, é utilizado para fazer a medição da temperatura corporal em contato direto com a pele (sensores de temperatura).

Os sensores são classificados como ativos quando produzem uma fonte de energia própria, são chamados de passivos quando dependem de uma fonte de iluminação externa para medirem a radiação refletida e/ou emitida pelos alvos. Conforme Florenzano (2002), a principal vantagem dos sensores ativos em relação aos passivos é que eles podem obter imagens tanto durante o dia como à noite e em qualquer condição meteorológica. Uma outra forma de classificar os sensores refere-se ao princípio de funcionamento dos mesmos. Assim, eles podem ser de varredura e de não varredura. Conforme citado por Novo (1989), os sistemas sensores de não varredura registram a radiação refletida de uma área da superfície da Terra em sua totalidade num dado instante. Nos sensores de varredura, a imagem da cena é formada pela aquisição sequencial de imagens elementares do terreno, ou elemento de resolução, também chamado pixel.

A Figura 2.2 da próxima página representa o satélite *Landsat 7*, que possui o sensor TM (*Tematic Map*). O Landsat é um satélite americano, passivo e de varredura, criado especificamente para a agricultura, e possui um bom custo/benefício.

2.1.2. Sensores Remotos

Sensores remotos são equipamentos capazes de medir as grandezas físicas sem estar em contato direto com o objeto sensoriado. Conforme Carvalho (2005), Dados de sensoriamento remoto são gerados por aparelhos que medem alguma grandeza física de interesse, após esta ter interagido com a superfície da Terra e com a atmosfera.

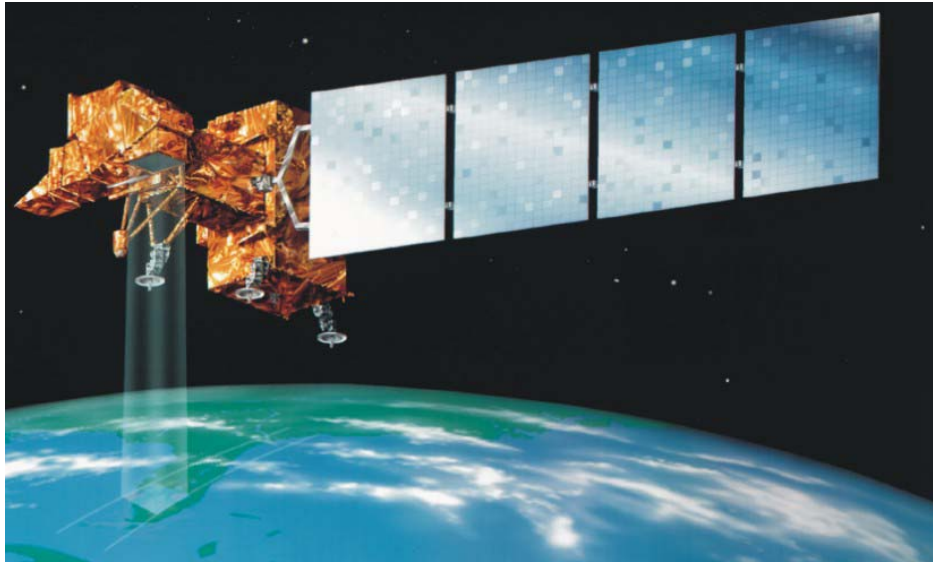


Figura 2.2: O satélite TM-Landsat 7

Fonte: Souza (2004)

Existem diversas definições das propriedades básicas de um sensor, levando-as a diferentes valores. Segundo Crosta (1993), estas propriedades podem ser resumidas como se segue, onde algumas são exclusivas para satélites:

- **Resolução Espectral:** conceito inerente às imagens multi-espectrais de sensoriamento remoto. É definida pelo número de bandas espectrais de um sistema sensor e pela largura do intervalo de comprimento de onda coberto por cada banda. Quanto maior o número de bandas e menor a largura do intervalo, maior é a resolução espectral de um sensor.

- **Resolução Radiométrica:** É dada pelos níveis de cinza, usados para expressar os dados coletados pelo sensor.

- **Resolução Espacial:** Refere-se ao tamanho da área representada por cada pixel da imagem.

- **Resolução Temporal:** Dada pelo intervalo de tempo entre duas tomadas da imagem.

Uma melhor resolução espacial melhora a delimitação das áreas porque diminui o número de informações por pixel na imagem, aumentando a definição da mesma. Uma melhor resolução espectral permite analisar a reflectância de uma mesma cena em várias faixas do espectro eletromagnético.

2.2. Reflectância

Florenzano (2002) cita que os objetos da superfície terrestre, como a vegetação, a água e o solo, refletem, absorvem e transmitem radiação eletromagnética em proporções que variam com o comprimento de onda, de acordo com as suas características biológicas, físicas e químicas, e cada objeto ou classe de objetos tem uma reflectância própria. Devido a essas variações é possível distinguir os objetos da superfície terrestre nas imagens de sensores remotos. A representação dos objetos nessas imagens varia da cor branca, quando refletem muita energia, à cor preta, quando refletem pouca energia.

A radiação solar que chega à superfície da Terra ao atingir a planta, interage com a mesma e resulta em três frações, a saber: uma parte dessa radiação é absorvida pelos pigmentos contidos na folha, outra parte é refletida pelas folhas, e uma terceira parte sofre o processo de transmissão, através das camadas de folhas que compõem a copa e através das camadas que constituem a folha. Conforme Moreira (2001), dos três componentes resultantes do fracionamento da radiação solar incidente, ao interagir com a planta, a energia refletida pela vegetação tem sido a mais utilizada pelos sensores porque é nesta faixa do espectro eletromagnético, que se dispõe da maior quantidade de sensores orbitais e sub-orbitais capazes de registrar informações da superfície terrestre.

Para Moreira (2001), quando se pretende utilizar o sensoriamento remoto para obter informações a respeito da cobertura vegetal de uma região, é preciso levar em consideração a interação da energia com a comunidade de plantas. O dossel vegetativo é o conjunto de todas as copas da vegetação, numa determinada área, independente da espécie. Quando a cobertura do solo pelas plantas é baixa, o dossel é dito incompleto e, neste caso, a reflectância do solo interfere na da planta.

Conforme citado por Valeriano (1988), outro aspecto a ser considerado no estudo da interação do dossel vegetativo com a radiação eletromagnética é a camada de folhas do dossel. A superposição de folhas provoca modificações na reflectância da cena.

2.3. Imagem Digital

A imagem digital é um dos produtos do sensoriamento remoto. Ela traz consigo informações coletadas pelo sensor no momento de sua passagem por determinada área. De acordo com Schneider (2001), as imagens geradas por sensores orbitais são imagens

chamadas matriciais, pois possuem uma natureza discreta, ou seja, são formadas de elementos independentes, dispostos na forma de uma matriz. Cada elemento dessa matriz tem sua localização definida em um sistema de coordenadas do tipo linha e coluna, representados respectivamente por x e y . O nome dado a esses elementos é pixel, derivado do inglês *picture element*. A Figura 2.3 é uma representação de uma imagem digital no plano cartesiano.



Figura 2.3: Exemplo de convenção dos eixos para representação de imagens digitais

Fonte: Adaptado de Gonzalez (2000)

Para um mesmo sensor remoto, cada pixel representa sempre uma área com as mesmas dimensões na superfície da Terra. Cada pixel possui também um atributo numérico z , que indica o nível de cinza desse elemento, conhecido como NC o qual varia do preto ao branco. Crosta (1993) ressalta que o nível de cinza de um pixel corresponde sempre à média da intensidade da energia refletida ou emitida pelos diferentes materiais presentes nesse pixel. O quanto de área que cada pixel representa depende da resolução espacial do sensor. A Figura 2.4, como exemplo, tem uma seção alarguecida onde se pode enxergar os quadrados (pixels). A intensidade do brilho varia entre 0 e 255, sendo que o valor zero é a porção do preto (escuro), e o valor 255 é a porção do branco (claro).

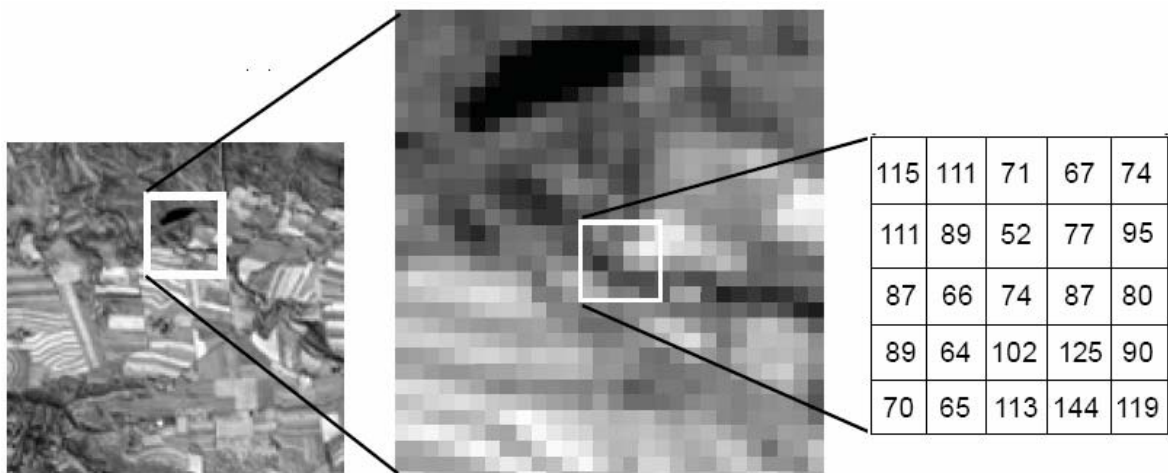


Figura 2.4: Seção Enlarguecida

Fonte: Adaptado de Smith (2006)

O histograma é uma das formas mais comuns de se representar a distribuição dos NCs de uma imagem. Conforme citado por IDL (2006), o histograma fornece a informação sobre quantos pixels na imagem possuem cada valor possível de níveis de cinza (que, no caso das imagens de oito bits, variam de 0 a 255) ou, de forma equivalente, são também conhecidos como distribuição de intensidades.

A Figura 2.5 mostra um exemplo de um histograma e sua imagem relacionada. O valor mais comum desta imagem é 180, que compõe o background da imagem. Embora o fundo (background) da imagem aparenta ser bem uniforme, ele contém muitas variações pequenas.

Segundo Larsen (2003), a coloração de uma imagem de satélite é denominada composição colorida falsa-cor. Essa é a forma mais tradicional de se combinar três bandas espectrais de satélite para a formação de uma composição colorida e sem dúvida, a mais usada. Pode-se usar quaisquer três bandas para se produzir uma composição colorida. Contudo, uma cuidadosa seleção das mesmas faz-se necessária, para que a combinação destas contenha a informação espectral realmente desejada. Isso obviamente se complica conforme aumenta o número de bandas, o que é uma tendência sempre crescente em sensores remotos. Em seguida, deve ser buscada uma alocação de cores que traduza essa informação para o olho humano da melhor forma possível.

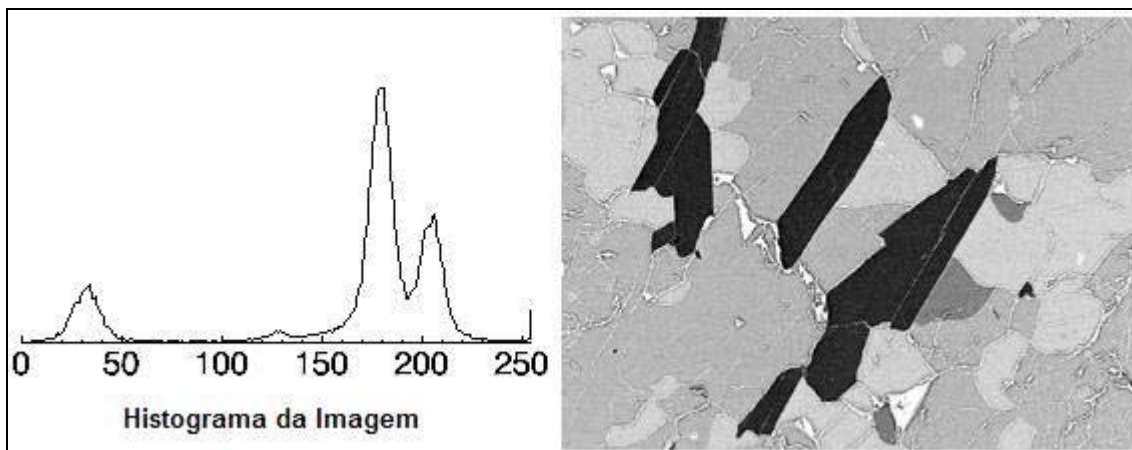


Figura 2.5: Exemplo de um Histograma (esquerda) e sua imagem relacionada (direita)

Fonte: Adaptado de ILD (2006)

Pode-se obter qualquer cor através da mistura do verde, vermelho e azul. Por esse motivo, computacionalmente, uma cor é representada pela quantidade de vermelho, verde e azul que a compõe. Schneider (2001) cita que um. Um exemplo de composição falsa-cor, a qual mais se aproxima da realidade, com as imagens do Landsat é a seguinte: a terceira banda no azul, a quarta banda no vermelho e a quinta banda no verde.

De acordo com Florenzano (2002), as imagens de sensores remotos, como fonte de dados da superfície terrestre, são cada vez mais utilizadas para a elaboração de diferentes mapas. Enquanto os mapas contêm informações, as imagens obtidas de sensores remotos contêm dados brutos, que só se tornam informação após a sua interpretação.

2.4. Detecção de Mudanças

Conforme citado por Janssen (1993), sensoriamento remoto está sendo amplamente usado em conjunto com SIG – Sistemas de Informação Geográficas. Bases de dados SIG são usadas para aprimorar a extração de informações relevantes de imagens de sensoriamento remoto, enquanto que dados de sensoriamento remoto fornecem imagens periódicas de características geométricas e temáticas dos objetos na terra, aperfeiçoando nossa habilidade de detectar mudanças e atualizar as bases de dados do sistema de informação geográfica. Com as imagens, em tempos distintos, é possível detectar as mudanças ocorridas de acordo com as devidas diferenças de refletâncias entre elas e fazer análises.

Martin (1989) citou em seu trabalho que sensoriamento remoto de satélite oferece um método potencialmente poderoso de monitorar mudanças em imagens em altas resoluções temporais e baixo custo. De acordo com Deer (1995), detecção de mudança é o processo de identificação de diferenças no estado de um objeto ou fenômeno observando-o em tempos diferentes. A premissa básica de se usar dados de sensoriamento remoto para detecção de mudanças é que as mudanças no objeto de interesse resultarão em mudanças nos valores de radiação ou textura local, os quais devem ser tratados separadamente de mudanças causadas por outros fatores, como diferenças nas condições atmosféricas, iluminação, ângulo de visão e umidade do solo. É fundamental que as mudanças de interesse sejam separáveis das mudanças esperadas ou eventos que não são interessantes.

De acordo com Carvalho (2001), as dificuldades com a detecção de mudanças numa cobertura de área são complicadas se comparadas com o mapeamento de cobertura da área, forçando assim limites para a automação.

2.4.1. Um Exemplo de Detecção de Mudança

Conforme Wickware (1981), a detecção de mudanças envolve uma simples comparação visual entre duas imagens. Além disso, muitos dos estudos têm requerido uma abordagem quantitativa mais detalhada para as análises destas mudanças.

De acordo com Deer (1995), existem várias técnicas comuns de detecção de mudança em uso no sensoriamento remoto. O exemplo mais simples é o de diferenciação de imagens. Uma imagem é subtraída da outra, pelo seu valor de brilho do pixel, também conhecido como Intensidade Radiométrica. O valor absoluto é retornado. Quando não existe diferença, ou seja, quando não houve nenhuma detecção de mudanças, o valor do pixel é zero e a representação do pixel é escura. Quando existe algum tipo de detecção de mudança, o valor do pixel é diferente de zero.

Um exemplo simples, mas muito utilizado na área médica seria uma imagem do cérebro de um humano tirada em duas situações distintas. Fazendo a diferença entre as duas imagens obtém-se uma imagem resultante. As Figuras 2.6 e 2.7 abaixo exemplificam este processo.

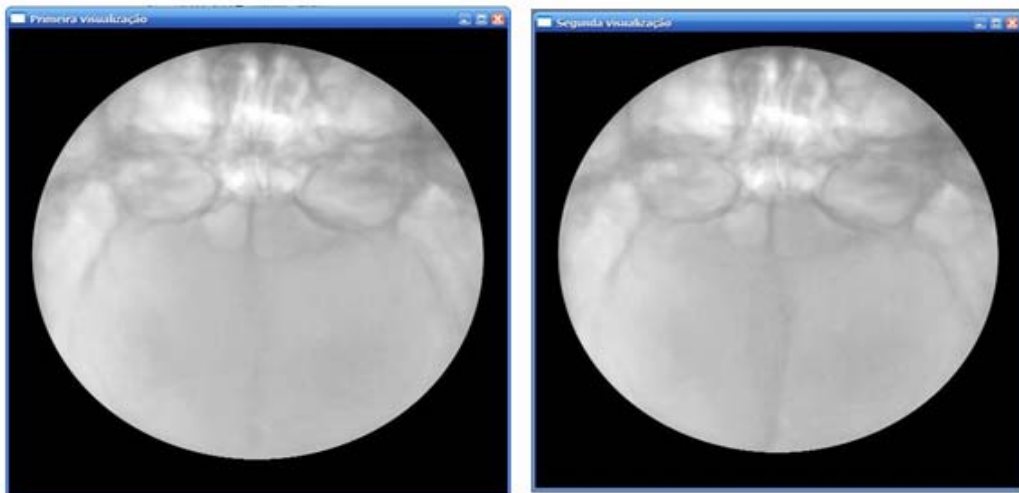


Figura 2.6: Primeira e segunda imagem

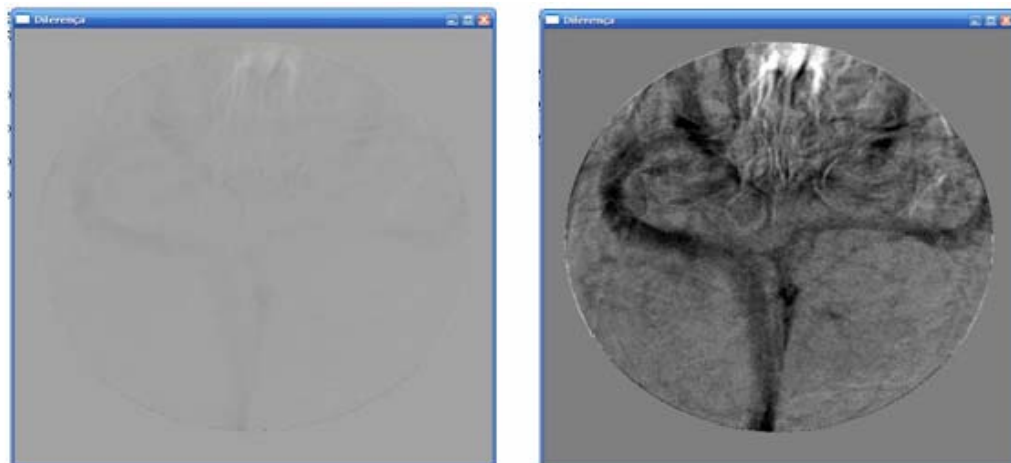


Figura 2.7: Resultante à esquerda e com realce à direita

A Figura 6 contém as imagens multi-temporais do cérebro. A Figura 7 retorna uma imagem trabalhada pelo algoritmo de detecção, onde são retornados somente os pixels de valores maiores do que 10 assim como os pixels de valores menores do que -10.

2.4.2. Algoritmo de detecção de mudanças

Segundo Muchoney e Haack (1994), vários algoritmos de detecção de mudanças têm sido desenvolvidos, sendo que a técnica da razão de bandas tem se destacado devido à sua simplicidade de implementação e eficiência. Vogelmann e Rock (1986), estudando danos causados em florestas de *Pinus* nas montanhas de Vermont, USA, encontraram que a

razão de bandas permitiu quantificar os níveis destes danos. Hoffer e Lee (1990) usaram três técnicas de detecção de mudanças: razão entre bandas, diferença entre bandas e análise por componentes principais.

Os autores reportam que as três técnicas permitiram a discriminação de diferentes grupos de cobertura florestal, não havendo diferenças significativas entre as técnicas testadas. Carvalho (2001) utilizou em sua tese o algoritmo básico onde a imagem resultante origina de uma simples operação matemática de subtração pixel a pixel entre as imagens interessadas com o uso da linguagem IDL. Uma operação matemática simples feita em IDL está sendo exemplificada na Figura 2.8. Este é o mesmo algoritmo que gerou as imagens representadas pelas Figuras 2.6 e 2.7 utilizando a subtração.

```
PRO File_input_mod
;abrir a imagem para dentro do idl
openr,unit,'c:\RSI\idl63\examples\data\cereb.dat',/GET_LUN

;define o tamanho do array
img1 = bytarr(512, 512)
img2 = bytarr(512, 512)

READU,unit,img1
READU,unit,img2

;comando para encerrar comunicação
FREE_LUN,unit

;cria uma janela do tamanho ideal para visualização
WINDOW,xsize=512,ysize=512,TITLE='Primeira visualização'

;seleciona um quadro específico para visualização
TVSCL,img1[*,*]

;cria uma janela do tamanho ideal para visualização
WINDOW,xsize=512,ysize=512,/free,TITLE='Segunda visualização'

;seleciona um quadro específico para visualização
TVSCL,img2[*,*]

WINDOW,xsize=512,ysize=512,/free,TITLE='Diferença'

diferenca = fix(img2) - img1

TVSCL,diferenca

END
```

Figura 2.8: Subtração entre 2 bandas de uma imagem .dat

Fonte: IDL (2006)

2.5. Interface Homem Máquina

De acordo com Hix & Hartson (1993), a interface homem-máquina, conhecida também com IHM, é o que acontece quando um usuário humano e um sistema computacional se juntam para realizar uma tarefa. O estudo desta interação homem-

computador tenta responder qual a melhor maneira de fazer esta interação funcionar. O desafio na maioria dos trabalhos em IHM é de um jeito ou de outro fornecer ao usuário o maior nível de usabilidade.

Cunha (2004) cita que cada usuário tem sua própria concepção de Interfaces entre o homem e uma máquina qualquer. A experiência particular de interação com sistemas é o principal fator que caracteriza o termo. De acordo com Mandel (1997), citado por Cunha (2004), um indivíduo pode acreditar que a interface é “o que ele vê na tela”. A interface gráfica computacional funciona como um agente de interação que torna possível a comunicação entre o homem e a máquina.

Conforme Bonsiepe (1997), interface gráfica computacional pode ser mais bem definida como uma especificação dos objetos que o usuário vê no monitor e as regras básicas para interagir com estes objetos. A Figura 2.9 busca representar o projeto de interfaces gráficas com o usuário.

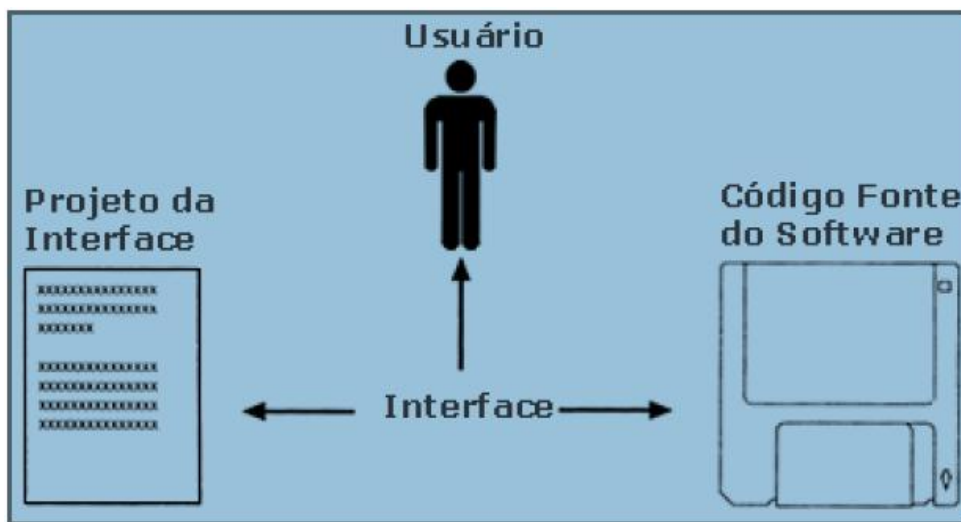


Figura 2.9: Diagrama do Projeto de Interfaces com o usuário.

Fonte: (Bonsiepe, 1997)

Conforme Cunha (2004), é através da interface que o usuário se comunica com o sistema por meio de um plano físico, perceptivo e cognitivo. Desta forma, a interface com o usuário tornou-se um conceito geral para projetistas e pesquisadores, passando a ser definida como algo que se pode mapear, projetar, implementar e unir à funcionalidade do sistema no seu desenvolvimento.

2.5.1. Usabilidade

Do ponto de vista técnico, Cybis (1994) citam que usabilidade é a combinação das seguintes características para uma interface de qualidade:

- Fácil de aprender
- Maior velocidade em que o usuário realiza as tarefas
- Baixa taxa de erro do usuário
- Satisfação subjetiva do usuário
- Tempo excedente de retenção do usuário

Cunha (2004) complementa que a aprovação integral das características de um sistema depende de dois aspectos fundamentais: o social e o prático. Um jogo é considerado como inaceitável pela sociedade se apresentar um nível muito elevado de violência para usuários infanto-juvenis. Além da aceitação social, existem também as questões correspondentes à aceitação prática. Um *site* fornecedor de serviços gratuitos na Internet (ex: e-mails, listas de discussão, páginas pessoais) também terá pouca aceitação se durante o cadastro for exigido um número elevado de informações do usuário.

Hix e Hartson (1993) ressaltam que a usabilidade é relacionada à eficácia e à eficiência da interface com o usuário, e também à reação do usuário com aquela interface.

2.5.2. GUI – Graphical User Interface

Interface gráfica (GUI, do inglês *Graphical User Interface*) é um mecanismo de interação homem-computador. De acordo com Answer (2006), GUI é uma interface de usuário baseada em gráficos que incorpora janelas móveis, ícones e um mouse. A habilidade de se redimensionar uma aplicação em janelas, mudar cores e estilo das fontes são suas principais vantagens, embora a sua criação seja mais detalhada e complicada.

De acordo com Answer (2006), as GUIs têm se tornado o meio padrão de se interagir com o computador. Elas substituem uma alternativa de interação homem-computador, que são as linhas de comando. Os principais GUIs são as interfaces Windows e MAC, juntamente com o MOTIF para Unix e o Gnome e o KDE para Linux.

A ferramenta utilizada para o desenvolvimento da interface gráfica proposta é a linguagem de programação IDL, que será citada no próximo tópico deste referencial.

2.6. A linguagem IDL

O IDL – Linguagem de Dados Interativo (do inglês Interactive Data Language) é uma linguagem de programação muito poderosa e estruturada, baseada em vetores ou *arrays*, oferecendo uma grande variedade de rotinas gráficas, controles amigáveis de interface, além de possibilitar a adição de novas rotinas, resultando num poderoso instrumento no desenvolvimento de visualizações interativas aplicadas ao sensoriamento remoto e ao SIG. IDL (2006) cita que programar em IDL é uma alternativa de ganho de tempo se comparado com a programação em FORTRAN ou C. Tarefas que gastam semanas para serem implementadas nestas linguagens tradicionais são implementadas em algumas horas com o IDL. Dados podem ser explorados iterativamente usando comandos IDL ou então várias rotinas de aplicação podem ser criadas no ambiente de desenvolvimento IDL Developer Environment - IDLDE (Figura 2.10).

O IDL possui uma ferramenta interativa que é usada para criação de interface gráfica. Ela é chamada de GUIBuilder. Ele gera o código fonte que define a interface e contém eventos que são chamadas pelos botões. O GUIBuilder só oferece suporte para o Windows, embora o código que ele gera é portátil em qualquer plataforma que o suportada.

O GUIBuilder possui várias ferramentas que simplificam o desenvolvimento das aplicações. Estas ferramentas permitem criar botões que compõem a interface. Estes botões são denominados *widgets* (Figura 2.11).

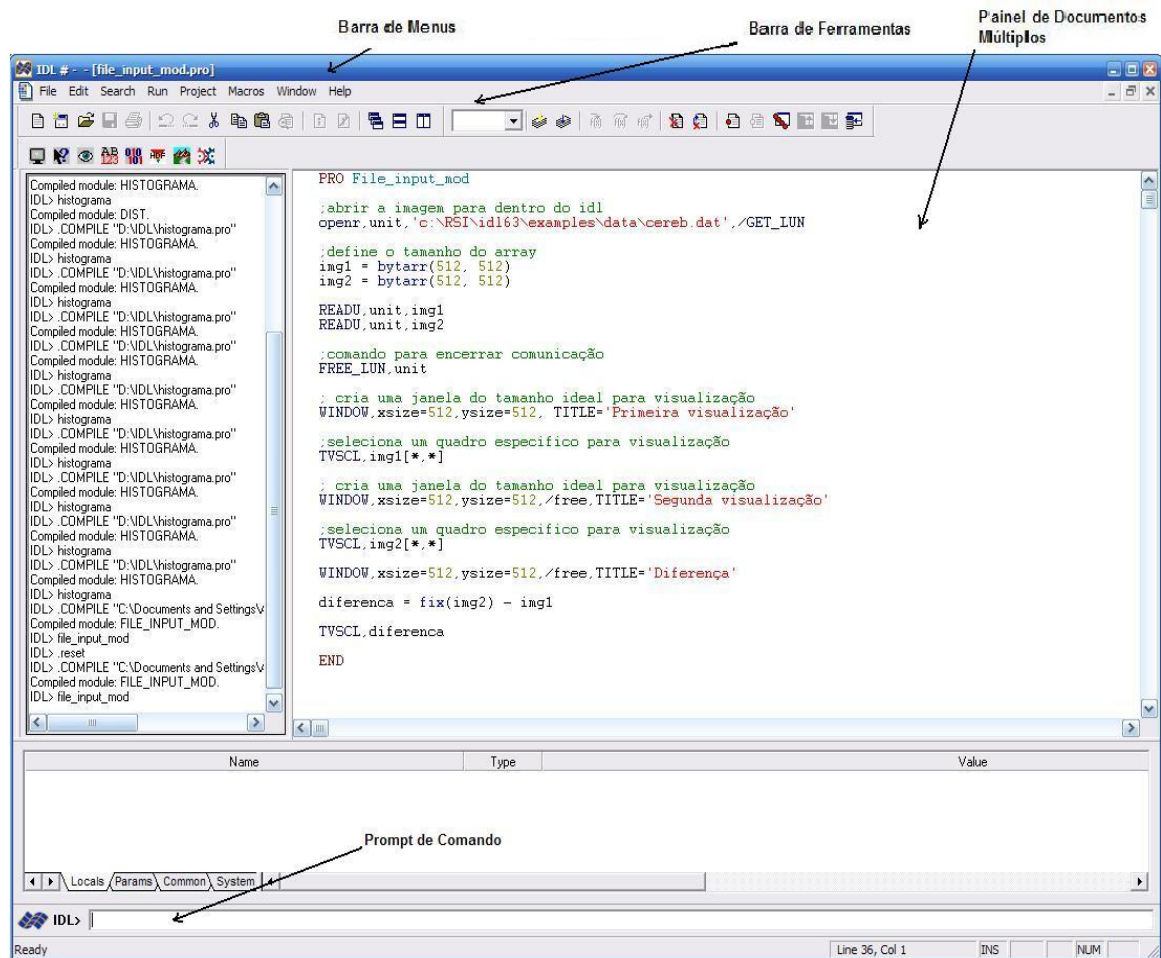


Figura 2.10: Interface IDLDE

Fonte: IDL (2006)

2.7. O Software ENVI

O ENVI é um software utilizado para visualizar, analisar, e apresentar qualquer tipo de imagem. Ele é totalmente escrito em IDL. Algoritmos escritos em IDL podem ser integrados no menu do ENVI (interface). O pacote completo de processamento de imagens do ENVI inclui ferramentas espectrais, correções geométricas, análise de terreno, análise de radar, suportes para sistemas de informações geográficas (SIG) para raster e vetor e muito mais. Tudo isso com grande facilidade de uso. A Figura 2.12 mostra a interface do ENVI 4.3.

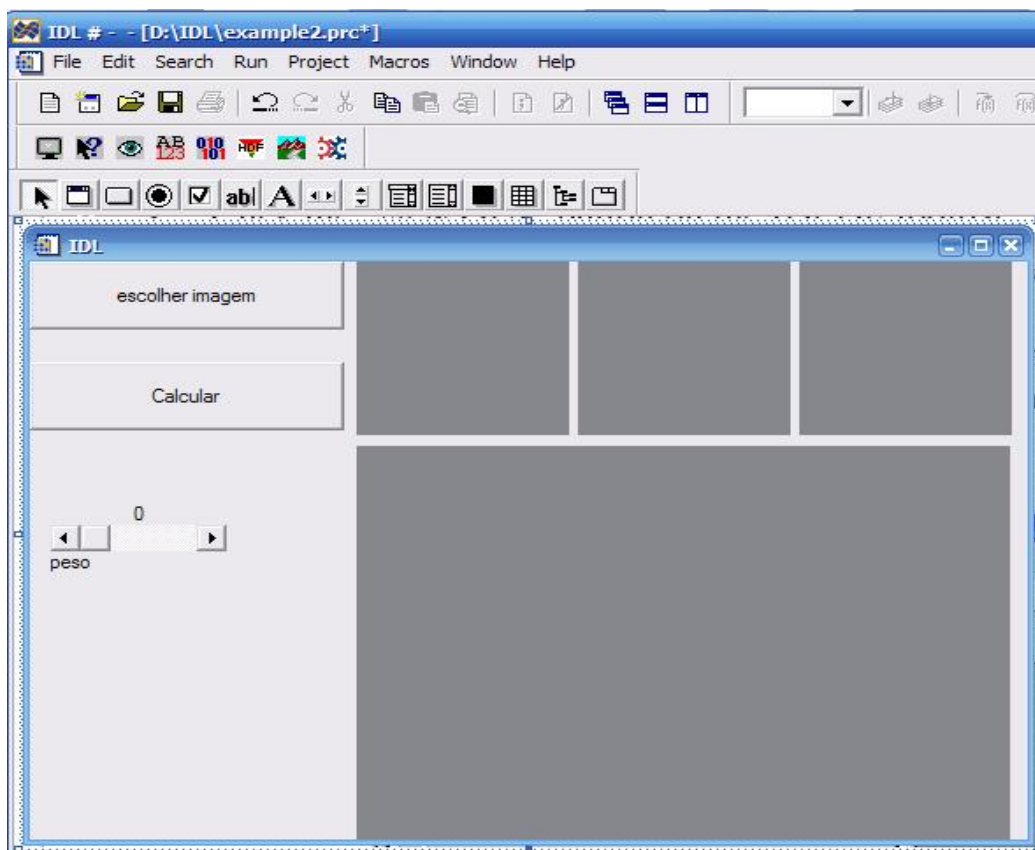


Figura 2.11: Interface do GUIBuilder

Fonte: IDL (2006)



Figura 2.12: Interface do ENVI 4.3

Fonte: ENVI (2006)

O ENVI possui uma funcionalidade muito poderosa para visualização que é o ENVI Zoom. A interface permite acesso rápido a ferramentas comuns de visualização, como contraste, brilho e transparência. Pode-se trabalhar com múltiplas camadas de dados ao mesmo tempo em uma mesma janela e usar o gerenciador de dados e de camadas para rastrear os conjuntos de dados múltiplos (Figura 2.13).

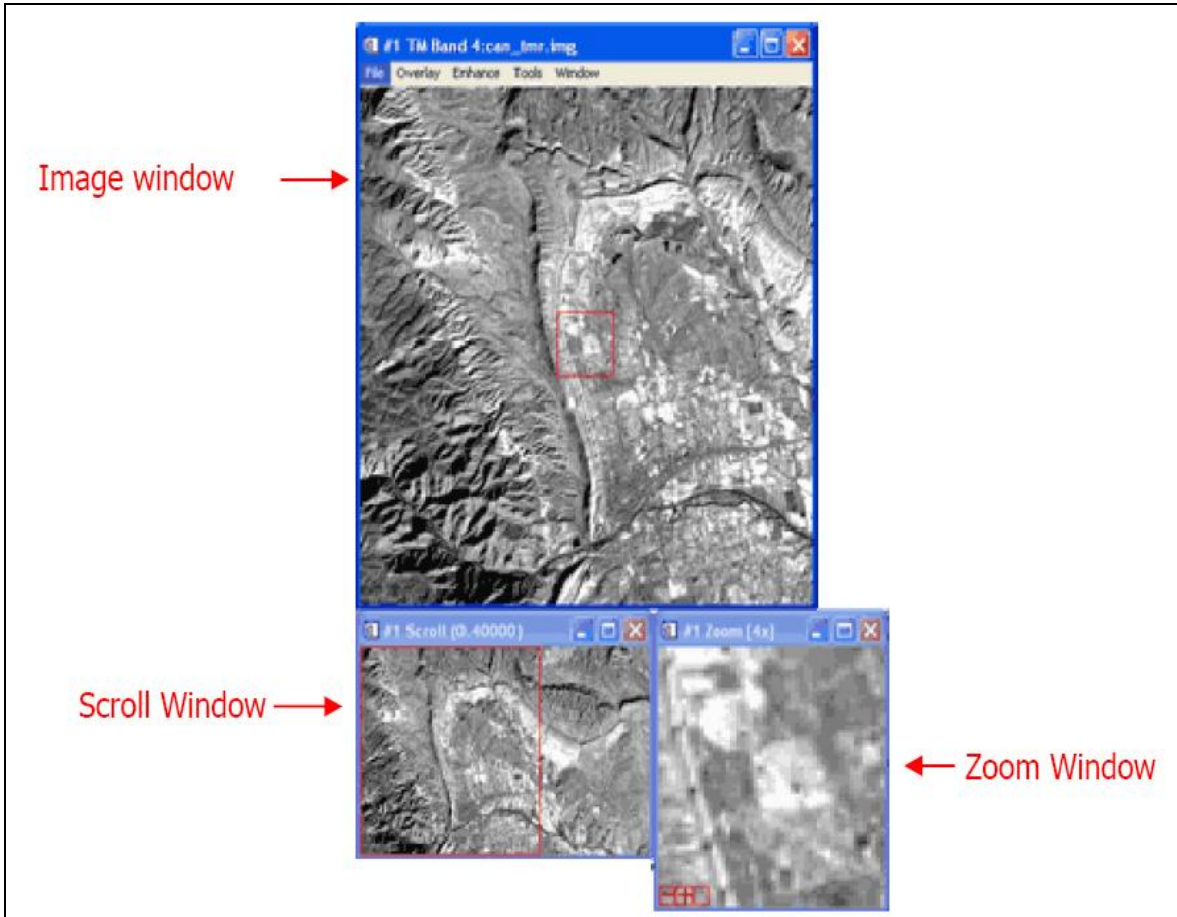


Figura 2.13: ENVI Zoom

Fonte: ENVI (2006)

3. METODOLOGIA

3.1. Tipo de Pesquisa

De acordo com Thiollent (1997), a metodologia é a maneira ou o método que deve ser usado para se conduzir uma pesquisa. Cohen (1971) disseram que o método científico é a técnica mais segura inventada pelo homem para controlar o fluxo das coisas e estabelecer crenças estáveis.

Quanto ao objetivo geral da pesquisa, a mesma foi classificada como pesquisa exploratória. A pesquisa exploratória tem como objetivo proporcionar maior familiaridade com o problema, tornando-o assim mais explícito e possibilitando a construção de hipóteses. Pode-se dizer que esse método de pesquisa tem como objetivo principal o aprimoramento de idéias ou a descoberta de intuições.

Quanto à natureza da pesquisa, ela pode ser classificada como tecnológica, uma vez que o desenvolvimento, utilizando pesquisa exploratória, objetivou a criação de um produto que é a interface gráfica de detecção de mudanças. Conforme Jung (2004), a pesquisa tecnológica utiliza conhecimentos básicos, tecnologias existentes, conhecimentos tecnológicos e tem como objetivo um novo produto ou processo.

3.2. Procedimento

O presente trabalho foi realizado no laboratório de geoprocessamento, situado no LEMAF – Laboratório de Estudos em Manejo Florestal, no Departamento de Ciências Florestais da Universidade Federal de Lavras. O período de realização foi entre novembro de 2006 e julho de 2007. O desenvolvimento contou com acompanhamento técnico do orientador responsável e apoio dos alunos estagiários do próprio laboratório que estão já ligados com pesquisas e estudos na área de processamento de imagens e detecção de mudanças. Foram utilizados o software ENVI e a linguagem de programação IDL, juntamente com o seu interpretador, sendo o ENVI na sua versão 4.3 e o IDL na sua versão 6.3. Estes softwares foram instalados em um microcomputador de processador Pentium 3.0, com 1Gb de memória RAM.

A metodologia proposta neste trabalho está estruturada em cinco etapas, descritas em ordem cronológica, as quais foram: O estudo do algoritmo de detecção de mudanças; análise do ENVI e das suas aplicações; estudo da linguagem IDL; e estudo das aplicações com *widgets* em IDL.

3.2.1. Estudo do Algoritmo de Detecção de Mudanças

O estudo do algoritmo de detecção de mudanças foi efetuado para se obter um maior entendimento e também para se determinar a maneira como as funcionalidades poderiam ser implementadas na interface gráfica. Percebeu-se que este algoritmo utiliza várias funções prontas da biblioteca do ENVI. As funções da biblioteca do ENVI são escritas em IDL, mas os seus códigos-fonte são salvos em formato binário, impossibilitando a visualização e a análise dos mesmos.

A utilização de funções do ENVI em uma aplicação construída em pela linguagem IDL requer o uso das bibliotecas do ENVI para a aplicação ser executada. Para isto acontecer, o ENVI deve estar sempre rodando junto com a aplicação, ou seja, o funcionamento do algoritmo de detecção de mudanças dependente do software ENVI em execução. O ideal é fazer o algoritmo ser executado dentro de uma interface gráfica, tornando a mesma independente do ENVI. O algoritmo conta com três passos principais. O Layer stacking, a normalização e a imagem diferença.

- **Layer stacking**

As imagens multi-temporais podem não estar alinhadas, ou seja, não correspondem exatamente a uma mesma área de cobertura. O *layer stacking* reconhece somente a união entre as duas imagens e o restante é descartado da análise. A nova variável vai armazenar as bandas da união das duas imagens. Por exemplo, duas imagens de Landsat com seis bandas passam pelo algoritmo e o resultado é um arquivo com 12 bandas, se existir união.

- **Normalização**

Esta parte do algoritmo normaliza as duas imagens para correção de fatores que influenciam o valor de reflectância dos objetos: atmosfera, topografia, geometria de iluminação e de observação.

- **Imagem diferença**

A diferença entre as bandas de uma imagem e outra é feita nesta parte do algoritmo, resultando na imagem diferença, pronta para análises de detecção.

3.2.2. Análise do ENVI e das Suas Aplicações

O estilo de interface e as formas de aplicações do ENVI foram analisados para se manter um padrão baseado no ENVI e a interface gráfica desenvolvida. Desta forma, um usuário que já é familiarizado com o ENVI teria maior facilidade no uso do software.

3.2.3. Estudo da Linguagem IDL

Como o algoritmo de detecção de mudanças já foi escrito em IDL, a decisão mais aceitável foi escolher o próprio IDL como linguagem de desenvolvimento da interface gráfica. Houve-se então a necessidade de pesquisar sobre a esta linguagem.

3.2.4. Estudo das Aplicações com Widgets em IDL

Foi analisada a viabilidade de utilização do ambiente de desenvolvimento GUIBuilder e do IDLDE. O ambiente escolhido para este trabalho foi o IDLDE. O fundamento do desenvolvimento de uma interface gráfica são os *widgets*. Esta foi a principal parte do desenvolvimento da interface gráfica. Para facilitar o entendimento sobre os *widgets*, a **Tabela 1** abaixo traz a relação dos *widgets* utilizados na interface.

Tabela 3.1: Lista dos *Widgets* utilizados

WIDGET_BASE	Cria o <i>widget</i> base (container para os outros <i>widgets</i>)
WIDGET_BUTTON	Cria botões de <i>widgets</i> .
WIDGET_DRAW	Cria um <i>widget</i> para inserir imagens ou desenhos.
WIDGET_LABEL	Cria <i>widgets</i> de rótulos.
WIDGET_TEXT	Cria <i>widgets</i> de texto.
WIDGET_TREE	Cria <i>widgets</i> no formato de árvore.
WIDGET_CONTROL	Realiza, gerencia, e destrói os <i>widgets</i> .

- **WIDGET_BASE**: Usado para criar bases de widgets. De acordo com IDL (2006), ele serve de container para os outros widgets, inclusive também para as outras bases.

```
base1 = WIDGET_BASE( )
base2 = WIDGET_BASE(base1)
base3 = WIDGET_BASE(base2)
```

- **WIDGET_BUTTON**: cria um botão que é ativado ao passar o cursor do mouse sobre o botão e pressionar o botão do mouse (Figura 3.1).



Figura 3.1: widget_button

- **WIDGET_LABEL**: cria áreas estáticas de texto. Eles são utilizados em rótulos pequenos (Figura 3.2).



Figura 3.2: widget_label

- **WIDGET_DRAW**: são áreas retangulares usadas para tratar janelas gráficas (Figura 3.3).



Figura 3.3: widget_draw

- **WIDGET_TEXT**: cria áreas de textos onde podem ser inseridas mais de uma linha de texto e os textos podem ou não ser editáveis (Figura 3.4).



Figura 3.4: widget_text

- **WIDGET_TREE**: é usado para criar e popular um widget do estilo árvore. ela apresenta uma hierarquia onde pode organizar uma grande variedade de estrutura de dados e informações (Figura 3.5).

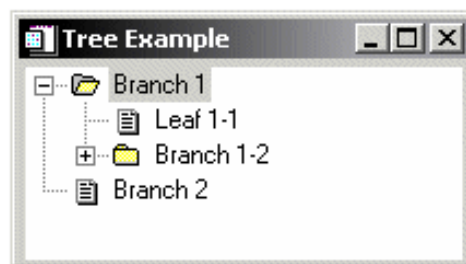


Figura 3.5: widget_tree

- **WIDGET_CONTROL**: é utilizado para controle dos widgets. Realiza, gerencia, e destrói os *widgets*. Sem o *widget_control*, os *widgets* não são executados.
- **XMANAGER**: Controla o processamento dos eventos até que todos os *widgets* sejam destruídos.

4. RESULTADOS E DISCUSSÃO

4.1. O Desenvolvimento da Interface

Para facilitar o entendimento do método de desenvolvimento, foi decidido nomear o software de detecção de mudança desenvolvido neste trabalho de SDM.

O desenvolvimento inicial foi realizado com a construção da interface gráfica pura, utilizando *widgets*. A interface contém botões para inserir as imagens T1 e T2, alguns botões e controle de bandas para opção de tipos de visualização e três janelas de visualização das imagens T1, T2 e Resultante.

A imagem T1 é uma imagem extraída no tempo T1, a imagem T2 é uma imagem extraída do mesmo local, mas no tempo T2 e a imagem resultante é a imagem onde se obtém o resultado dos processos. A Figura 4.1 representa o modelo de sua interface inicial.

No caso particular do Laboratório de Estudos em Manejo Florestal, imagens geo-referenciadas de sensoriamento remoto do Estado de Minas Gerais são utilizadas, com diferenças de 2 anos entre a imagem T1 e a imagem T2.

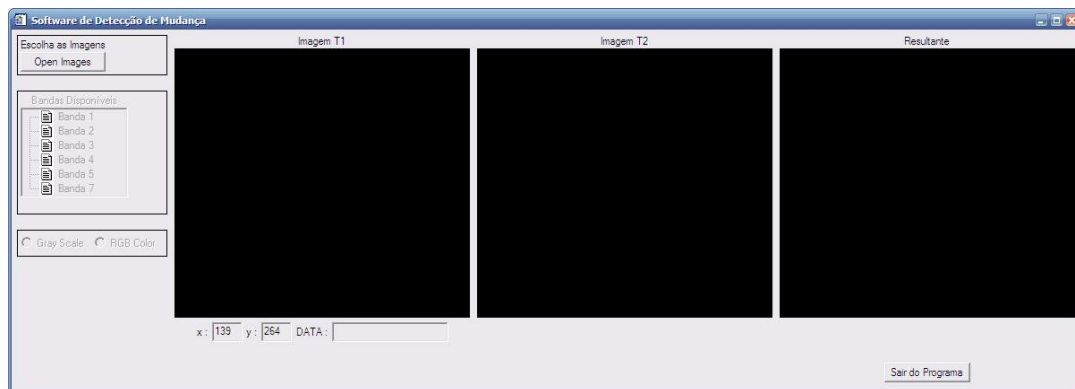


Figura 4.1: Interface do Software Desenvolvido

4.2. As Rotinas Criadas

Foram criadas duas rotinas. Na rotina principal foi implementada a interface gráfica do SDM. A outra rotina é a de manipulação dos eventos que são criados quando o usuário interage com a interface. Esta segunda rotina recebe os eventos do usuário como

parâmetro. A extensão dos arquivos de rotinas do IDL é .pro. As rotinas criadas foram “monografia.pro” e “monografia_event.pro”.

4.3. A Rotina *monografia.pro*

A primeira coisa a se iniciar na criação de uma interface gráfica no IDL é a criação da sua base líder. A Figura 4.2 exemplifica o código fonte da base inicial.

```
PRO MONOGRAFIA

; Inicialização de
; algumas Variáveis

device,decomposed=0
tamx = 330
tamy = 300
opcao = ''

; TOP-LEVEL BASE
; Base principal:Esta é a base principal
; Conhecida como EVENT.TOP
base = WIDGET_BASE(TITLE= 'Software de Detecção de Mudança', $
                  /ROW, YOFFSET=53)
```

Figura 4.2: A Base topo da aplicação

Em cima da base topo, também conhecida por *top level base*, foram criadas todas as outras bases, com seus respectivos botões e eventos. Pense nas bases como se fossem camadas, onde a base líder é a primeira das camadas onde vai sustentar as outras bases.

A Figura 4.3 exemplifica a hierarquia nas bases, utilizando as quatro bases utilizadas no desenvolvimento no segundo nível e mais quatro bases do widget wbase1 no terceiro nível.

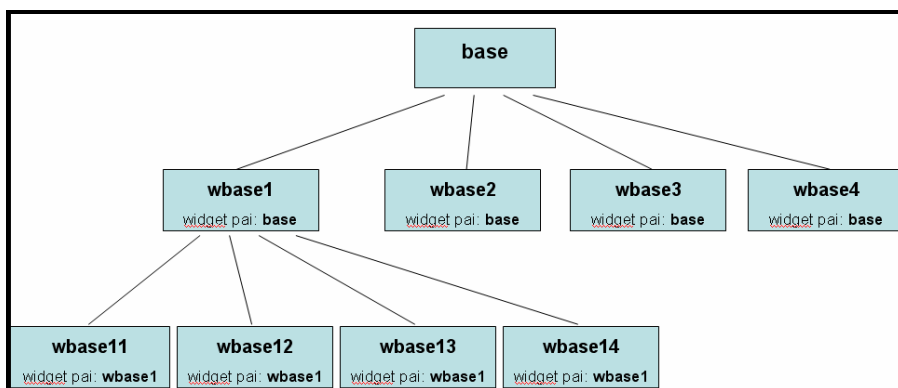


Figura 4.3: Exemplo da hierarquia das bases

Para maior entendimento do algoritmo, o código-fonte foi indentado na forma de árvore, onde as sub-bases ficam sempre mais à direita. Os nós são os WIDGET_BASEs e as folhas são os outros tipos de WIDGETS. Um exemplo da indentação é mostrado na Figura 4.4 abaixo

```
wbase14 = WIDGET_BASE(wbase1, /FRAME)
  grayscale = WIDGET_BASE(wbase14, /COL)
    selected = WIDGET_LABEL(grayscale, VALUE='Selected Band')
    sel_band = WIDGET_TEXT(grayscale, XSIZE=20, VALUE='teste grayscale')

  rgb = WIDGET_BASE(wbase14, /COL)
    rgb1 = WIDGET_BASE(rgb, /ROW)
      dummy = WIDGET_LABEL(rgb1, Value= ' R : ')
```

Figura 4.4: Exemplo da Identação

Depois de ter efetuado todo o processo de criação dos widgets, é preciso utilizar o WIDGET_CONTROL para realizar os *widgets* criados. O próximo passo foi a construção da rotina de manipulação de eventos.

Para poder usar as variáveis novamente em outros eventos, ou seja, para fazer as variáveis tornarem globais criou-se uma estrutura, e nesta estrutura foram inseridas as variáveis. Além disto, foi criado um ponteiro para apontar para esta estrutura sempre que possível. A parte do código da Figura 4.5 mostra a parte da criação da estrutura e dos ponteiros criados.

```
info = { base: base, $
  wbase1: wbase1, $
  wbase11: wbase11, $
  wbase12: wbase12, $
  wbase13: wbase13, $
  wbase14: wbase14, $
  wOpenImages: wOpenImages, $
  wImg1: wImg1, $
  wImg2: wImg2, $
  wResult: wResult, $
  wXpos: wXpos, $
  wYpos: wYpos, $
  wZpos: wZpos, $
  ping1: PTR_NEW(), $
  ping2: PTR_NEW(), $
  preresult: PTR_NEW()}

;Ponteiro criado para manipular as variáveis globais
;Entre um evento e outro
pinfo = PTR_NEW(info, /NO_COPY)
```

Figura 4.5: Criação da estrutura info e do ponteiro pinfo

4.4. A Rotina *monografia_event*

Quando o usuário executar alguma interação com o SDM os eventos são levados para esta rotina. Lá eles serão manipulados. O código abaixo (Figura 4.6) mostra a parte onde os parâmetros são lançados.

```
PRO MONOGRAFIA_EVENT, event
; This procedure is the event handler for a simple set of 'action' buttons.
; When a widget is touched, put its User Value into 'eventval':
WIDGET_CONTROL, event.id, GET_UVALUE = eventval
; This procedure call the top event and puts its User Value into 'pinfo'
WIDGET_CONTROL, event.top, GET_UVALUE= pinfo
```

Figura 4.6: Parte do Código que recebe os valores do usuário

Após a entrada dos parâmetros, foi usada a estrutura CASE que controla os tipos de eventos. A Figura 4.7 exemplifica as estruturas case no IDL.

```
; This CASE statement branches based upon the value of 'eventval':
CASE eventval OF

'draw' : Begin
; Controles da mostragem da localização dos pixels.
; Mostra o valor x e y do pixel no passar do mouse
widget_control, (*pinfo).wxpos, SET_VALUE=STRCOMPRESS(event.x,/REMOVE_ALL)
widget_control, (*pinfo).wypos, SET_VALUE=STRCOMPRESS(event.y,/REMOVE_ALL)
```

Figura 4.7: Parte do Código que mostra informações dos valores dos pixels das imagens

4.5. A Escolha das Imagens

Foi criado um botão para a seleção das imagens selecionadas. Ao clicar no botão “Abrir Imagens”, abre-se uma janela para selecionar a imagem. A Figura 4.8 mostra a disponibilidade inicial do SDM e a Figura 4.9 mostra a janela que abre para a seleção das imagens.

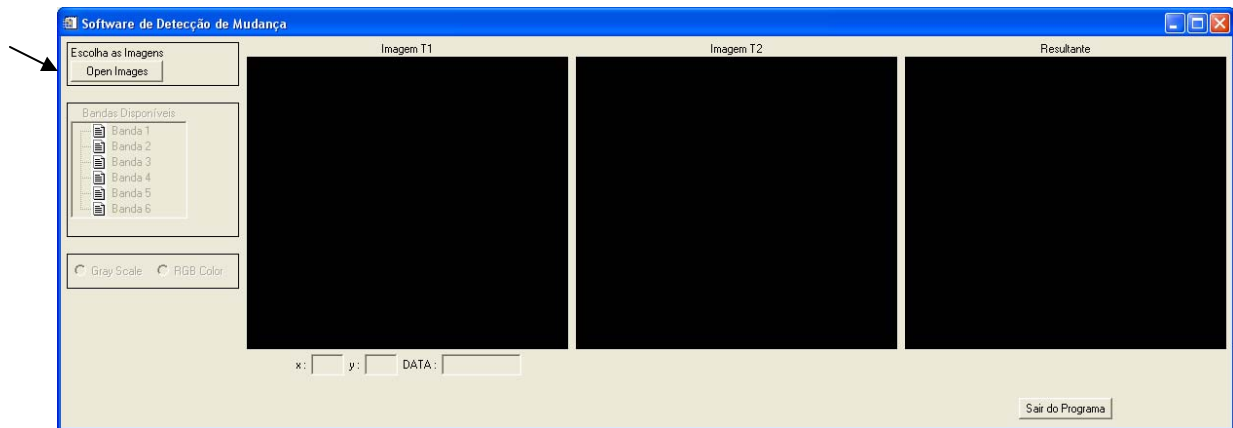


Figura 4.8: Botão de Seleção das Imagens mostrada à esquerda

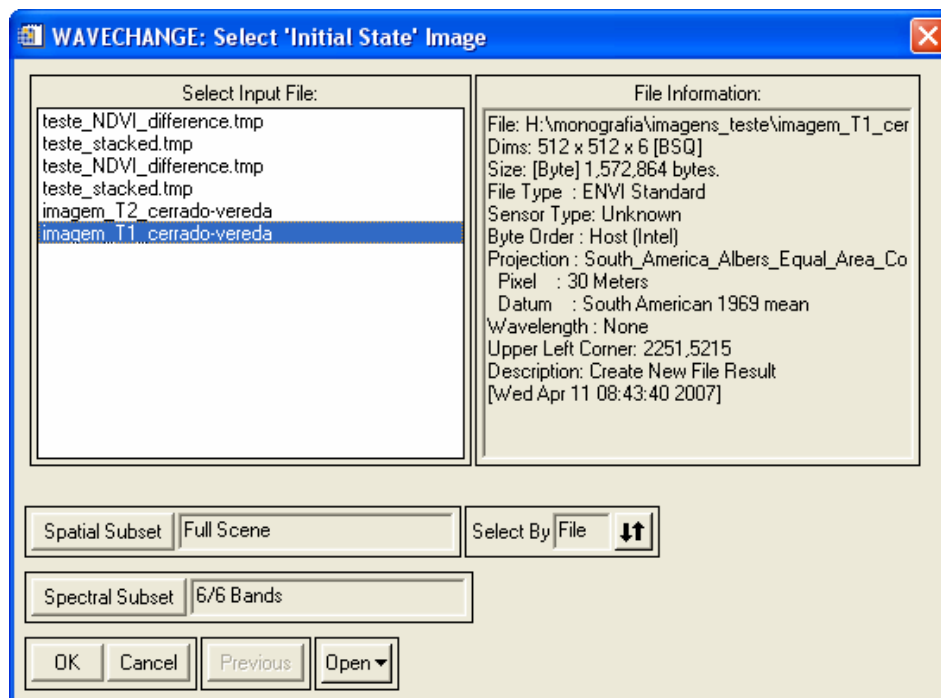


Figura 4.9: Janela para escolher imagens

Nesta janela existem duas colunas. A coluna da esquerda é denominada “Select Input File”, e ela é usada para armazenar o caminho das imagens que vão abrindo. A coluna da direita, denominada “File Information”, mostra toda informação necessária relacionada à imagem selecionada.

4.6. Disposição das Imagens na Interface

As imagens selecionadas são carregadas e passam pelo algoritmo de detecção de mudanças. A terceira imagem é o resultado da detecção de mudanças. (Figura 4.10).

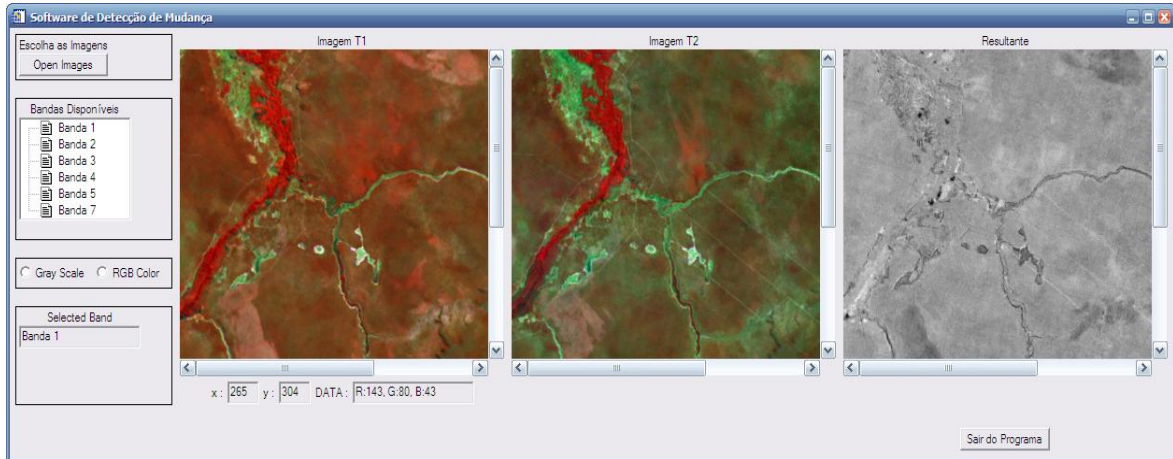


Figura 4.10: Imagem T1, Imagem T2 e a imagem resultante.

Foi criado um campo abaixo do display da Imagem T1 que fornece ao usuário os valores das coordenadas x e y dos pixels, assim como o valor de cada pixel. Ao passar o mouse pelas imagens os valores vão se modificando pixel a pixel.

4.7. Campo de Seleção das Bandas

Após a seleção das imagens T1 e T2, os campos da escolha do tipo de imagem e das bandas é habilitado juntamente com o campo de escolha entre imagens em escala de cinza, rotulado como “Grayscale” (Figura 4.11a) e imagens RGB, como “RGB Collor” (Figura 4.11b). O usuário deve escolher entre estas duas opções. Caso seja optado pela escala de cinza, somente uma banda do espectro eletromagnético deve ser selecionada. Se a opção escolhida for RGB Color, deve-se escolher uma banda para o vermelho, uma banda para o verde e outra banda para o azul. As imagens mostradas na Figura 4.10 colocam a quarta banda na faixa do vermelho, a quinta banda na faixa do verde e a terceira banda na faixa do azul.

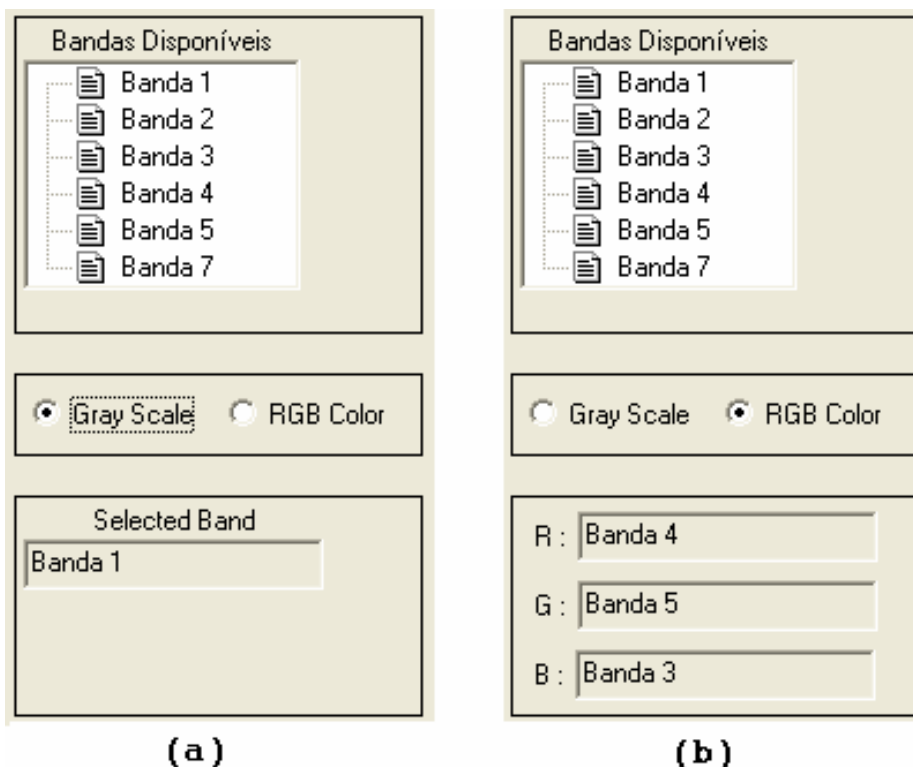


Figura 4.11: Campo de escolha

4.8. A escolha do IDLDE para desenvolvimento

A decisão do IDLDE para desenvolvimento partiu da necessidade de se organizar melhor com relação ao código-fonte do programa. O GUIBuilder é mais fácil de se construir a interface pura, mas na hora de referenciar os botões para suas devidas rotinas, o desenvolvimento fica confuso. Nos testes feitos ele criou automaticamente três arquivos diferentes de código, tornando-o ainda mais confuso. Resolveu-se então utilizar o IDLDE, onde criam-se *widjets* diretamente pela linha de comando, e somente um arquivo de código fonte é necessário ser criado, ao invés de três arquivos de código fonte. Outra vantagem é que a estética do código-fonte fica na responsabilidade do desenvolvedor.

4.9. O controle dos widgets

Um dos desafios foi o controle dos *widgets* criados. Para implementações onde janelas são criadas a partir da interação do usuário, a manipulação dos widgets deve ser feita com muita atenção porque a linguagem é um pouco confusa neste aspecto.

A modelagem deste trabalho foi baseada em apenas uma interface, onde todos os widgets foram criados em uma só base topo. Isto tornou a manipulação dos eventos mais fácil.

4.10. Janela de escolha das imagens

A janela já mostrada na Figura 4.10 é uma função do ENVI. Ela torna fácil a visualização das imagens selecionadas. A vantagem desta janela é a facilidade de manipulação e de visualização dos dados das imagens. Após a primeira seleção da imagem, a mesma fica armazenada no campo “*Select input file*” para futuras seleções. A desvantagem de se utilizar esta janela é que sempre na primeira seleção das imagens a raiz leva para uma pasta do ENVI. Isto só pode ser modificado se fosse implementada uma função em IDL que a substitua.

4.11. Importâncias Apresentadas

A interface desenvolvida possui as funcionalidades necessárias para um resultado visual da mudança entre as duas imagens multi-temporais. Do ponto de vista da GUI, ela atende à definição de uma Interface Gráfica que é, segundo IDL (2006), uma aplicação de *widgets* que permite ao usuário manipular o fluxo de eventos dentro de um programa sem o contato com o código-fonte.

Pequenas e médias empresas que atuam na área de processamento de imagens digitais podem desenvolver tais interfaces. A vantagem seria a possibilidade de se ter uma ferramenta personalizada para um problema específico. Com isso a empresa se livra de efetuar a compra de ferramentas poderosas, que possuem preços elevados. O único requisito que um computador necessita para rodar uma rotina em IDL é que uma máquina virtual do IDL seja instalada.

Uma interface gráfica escrita em IDL necessita de uma modelagem prévia, quando se vai efetuar no ambiente de IDLDE – *Developer Environment*, porque isto exige uma grande quantidade de variáveis, e a melhor forma de se ter uma organização destas variáveis é efetuando nomenclaturas seqüenciais.

5. CONCLUSÕES

5.1. Considerações Finais

O objetivo deste trabalho foi o desenvolvimento de uma interface gráfica para a execução do algoritmo de detecção de mudanças do projeto do inventario utilizando a linguagem IDL.

Tratando-se da linguagem de programação em questão, observou-se que além de ser uma linguagem poderosa na manipulação de dados vetoriais, no caso as imagens digitais, ela também possui um grande potencial no desenvolvimento de interfaces gráficas personalizadas para manipulação de imagens, e vale a pena ser verificada.

A opção de estilo de desenvolvimento utilizando o IDLDE foi satisfatória, obtendo assim maior organização com relação ao código.

O desenvolvimento não implementou todas as funções em IDL puro. Algumas funções, como a seleção da imagem, foram implementadas com funções do ENVI.

Em relação à interface gráfica, pode-se observar que, apesar de simples, atingiu ao objetivo de visualização das imagens e de dados das imagens. Além disto, a interface permite a inserção de outras funcionalidades em trabalhos futuros.

5.2. Proposta para Trabalhos Futuros

Adicionar mais funcionalidades ao SDM que são consideradas importantes, como funções de Zoom para melhor detalhamento dos pixels das imagens. Uma outra funcionalidade considerada importante seria um link para as três imagens de modo que os mesmos movimentos feitos em uma imagem sejam refletidos para todas as outras. Isto facilita visualização de mesmos pontos nas imagens.

Com relação à linguagem de desenvolvimento, um trabalho futuro seria implementar todas as funções em IDL puro, sem nenhuma ligação com o ENVI.

Melhorar o algoritmo na parte de seleção de imagens para que todo e qualquer tipo de imagens possam ser analisados na interface gráfica.

6. REFERÊNCIAS BIBLIOGRÁFICAS

ANSWER, W.C.B.; **GUI: Definition and Much More.** [online] Disponível em <http://www.answers.com/topic/gui>, 2006

BONSIEPE, G.U.I. **Design: do Material ao Digital.** Florianópolis: SEBRAE/SC, tradução de Cláudio Dutra, 1997.

BUITEN, H. J.; CLEVERS. J.G.P.W. **Land Observation by Remote Sensing: Theory and Applications.** 642p. 1996.

CARVALHO, L.M.T. **Mapping and Monitoring Forest Remnants: a Multi-scale Analysis of spatio-Temporal Data.** Tese Wageningen University, 2001

CARVALHO, L.M.T. **Sistemas de Informações Geográficas e Sensoriamento Remoto dos Recursos Florestais.** 109 p.: il. – Curso de Pós-Graduação “Lato Sensu” (Especialização) a Distância: Manejo de Florestas Plantadas. UFLA/FAEPE, 2005

COHEN, M.; NEGEL. E. **Introducción a la Lógica y al Método Científico.** Buenos Aires: Amorrutu, 1971

CROSTA, A. P. **Processamento Digital de Imagens de Sensoriamento Remoto.** Campinas/SP: IG/UNICAMP, 1993.

CUNHA, L. H. T., **Modelagem e Desenvolvimento de um Software para a Diretoria de Recursos Humanos da Universidade Federal de Lavras, MG, UFLA.** Monografia de conclusão de curso. DCC/UFLA. 2004.

CYBIS, W.A. **A Identificação de Objetos de Interfaces Homem-Computador e seus Atributos Ergonômicos.** Tese de Doutorado, Engenharia de Produção. UFSC, Florianópolis: 1994.

DEER, P. **Digital Change Detection Techniques in Remote Sensing**. Information Technology Division Electronics and Surveillance Research Laboratory, 1995

DOBSON, J. E. **Automated geography**. The Professional Geographer, Washington, v. 35, n.2, p.135-143, 1983.

FLORENZANO, T. G. **Imagens de Satélite para Estudos Ambientais**. São Paulo/SP: Oficina de Textos, 2002.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. São Paulo. Editora Edgard Blucher, 2000.

HAMË T.; HEILER, I.; MIGUEL-AYANS, J. S. **An Unsupervised Change Detection and Recognition System for Forestry**. International Journal of Remote Sensing 19: 1079-1099. 1998

HIX, D.; HARTSON, H. R. **Developing User Interfaces**. Canadá. Editora Wiley. 1993.

HOFFER, R.M.; LEE, K.S. **Change Detection of Forest Cover Using Satellite Radar Data**. In: Forest Service Remote Sensing Applications Conference, 3, Tucson, Arizona, 1990. **Proceedings**...Bethesda: American Society of Photogrammetry and Remote Sensing, 1990. p. 349-355.

IDL User's Guide. **Research Systems, Inc.**, 1200pp, 2006.

JANSSEN L.L.F. **Methodology ofr Updating Terrain Object Data from Remot Sensing Data: The Application of Landsat TM Data with Respect to Agricultural Fields**. Tese de Doutorado. Wageningen Agricultural University, Wageningen, 1993.

JUNG C F. "Metodologia para Pesquisa & Desenvolvimento". Ed. Axcel Books do Brasil Editora, 2004.

LARSEN, S. **Tempo é Dinheiro**. Revista InfoGeo, ano 4, número 25. 2003.

MARTIN, L. R. G. **Accuracy Assessment of Landsat-Based Visual Change Detection Methods Applied to the Rural-Urban Fringe.** Photogrammetric Engineering and Remote Sensing, 55, 2, pp. 209-215. 1989

MOREIRA, M. A. **Fundamentos do Sensoriamento Remoto e Metodologias de Aplicação.** São José dos Campos/SP: Instituto Nacional de Pesquisas Espaciais (INPE), 2001.

MUCHONEY, D.M.; HAACK, B.N. **Change detection for monitoring forest defoliation.** In: International Journal of Remote Sensing, London v.60, n.10, p.1243-51, 1994.

NILSEN, J. **Usability Engineering.** California: Academic Press, 1993.

NOVO, E.M.L.M., **Sensoriamento Remoto: Princípios e Aplicações.** São Paulo, ED. Edgard Blucher, p308. 1989.

RSI, **Image Processing in IDL**, IDL version 6.3, April 2006 Edition, 2006

SCHNEIDER, B. **Apostila de Computação Gráfica - Com ênfase em síntese de imagens** Lavras/MG: UFLA, 2001 [online] Agosto/2003 <http://www.comp.ufla.br/~bruno/aulas/cg/apostila/apostila-cg.pdf>.

SMITH, R. B. **Introduction to Remote Sensing of Environment (RSE) TNTmips.** Microimages. 2006. [Online] Disponível em: www.microimages.com

SOUZA, V. C. O. **Portal Vertical do GeoSolos: Geoprocessamento e Caracterização de Agroecossistemas Cafeeiros.** Monografia de conclusão de curso. DCC/UFLA. Pág18. 2004

THIOLLENT, M. **Pesquisa-Ação nas Organizações,** São Paulo, Atlas. 1997

VALERIANO, D.M. **Interações da Radiação Solar com a Vegetação.** São José dos Campos/SP: Instituto Nacional de Pesquisas Espaciais (INPE), 1988.

VOGELMANN, J.E.; ROCK, B.N. **Assessing forest decline in Coniferous forests of vermont using NS-001 Thematic Mapper simulator data.** In: International Journal of Remote Sensing, London, 17, p.1303-1321, 1986.

WICKWARE, G.M. **Change detection in the Peace-Athabasca Delta Using Digital Landsat data.** In: Remote Sensing of Environment, New York, n.11, p.9-25, 1981.