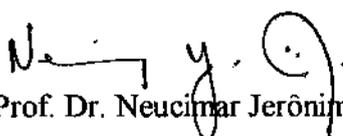


---

## Um Algoritmo Morfológico para Processamento de Fotomosaicos

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pelo Sr. Francisco Pinto Araújo Júnior e aprovada pela Comissão Julgadora.

Campinas, 20 de Março de 1996

  
Prof. Dr. Neucimar Jerônimo Leite

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para a obtenção do Título de Mestre em Ciência da Computação.



**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**

Araújo Junior, Francisco Pinto

Ar15a Um algoritmo morfológico para processamento de  
fotomosaicos / Francisco Pinto Araújo Júnior -- Campinas, [S.P.  
s.n.], 1996.

Orientador : Neucimar Jerônimo Leite

Dissertação (mestrado) - Universidade Estadual de Campinas,  
Instituto de Matemática, Estatística e Ciência da Computação.

I. Processamento de imagem. I. Leite, Neucimar Jerônimo. II.  
Universidade Estadual de Campinas. Instituto de Matemática,  
Estatística e Ciência da Computação. III. Título.

# Um Algoritmo Morfológico para Processamento de Fotomosaicos<sup>1</sup>

Francisco Pinto Araújo Júnior<sup>2</sup>

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
IMECC - UNICAMP

## Banca Examinadora :

- Neucimar Jerônimo Leite, Dr (orientador)<sup>3</sup>
- Jorge Stolfi, PhD<sup>3</sup>
- Roberto A. Lotufo, PhD<sup>4</sup>
- Paulo Lício de Geus, PhD (suplente)<sup>3</sup>

---

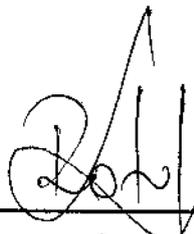
<sup>1</sup> Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação da UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

<sup>2</sup> O autor é Bacharel em Ciência da Computação, formado pela Universidade Federal do Ceará.

<sup>3</sup> Professor do Departamento de Ciência da Computação - IMECC - UNICAMP.

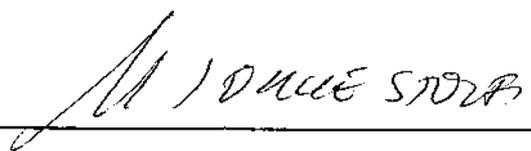
<sup>4</sup> Professor do Faculdade de Engenharia Elétrica - DCA - UNICAMP.

Tese de Mestrado defendida e aprovada em 20 de março de 1996  
pela Banca Examinadora composta pelos Profs. Drs.



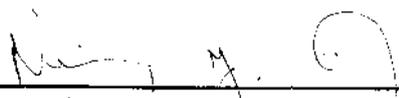
---

Prof (a). Dr (a). ROBERTO DE ALENCAR LOTUFO



---

Prof (a). Dr (a). JORGE STOLFI



---

Prof (a). Dr (a). NEUCIMAR JERÔNIMO LEITE

## Agradecimentos

Acima de tudo, agradeço à Deus pela inspiração e força no desenvolvimento deste trabalho.

Agradeço aos meus pais por sempre acreditarem na minha capacidade.

Agradeço a minha esposa pelo seu amor e dedicação.

Agradeço ao meu orientador, Prof. Neucimar J. Leite, pelo apoio constante no desenvolvimento deste trabalho.

Agradeço à Funceme pela oportunidade, em especial ao Manuel Pereira e Francisco Viana.

Agradeço ao Ricardo Sobral e José Matos pelo apoio técnico.

Agradeço ao CNPq pelo apoio financeiro.

## Resumo

### Um Algoritmo Morfológico para Processamento de Fotomosaicos<sup>1</sup>

Nós definimos, neste trabalho, um método morfológico para combinar imagens que se interceptam através de um processo denominado *photomosaicking*. Um fotomosaico é definido considerando a região de interseção, entre duas imagens, na qual uma linha de costura é detectada. O método descrito aqui utiliza a noção de linha de divisor de águas de uma função para extrair informações globais de uma imagem representando a homogeneidade entre as regiões de interseção. A partir destas informações globais, nós obtemos uma linha de costura conexa e irregular, e portanto mais realista do que aquelas definidas pelos métodos existentes.

---

<sup>1</sup> Este trabalho está inserido no contexto do projeto Geoprocessamento : Sistemas e Técnicas - GEOTEC, no âmbito do PROTEM - CC.

## Abstract

### A Morphological Algorithm for Photomosaicking

We define a morphological algorithm for combining two overlapping images into a single one by a process named photomosaicking. A photomosaic is defined considering an overlap region between two or more images, in which we detect a join line or seam. The method described here considers the notion of the watersheds of a function to extract global informations of an image denoting the homogeneity between two overlapping regions. By taking into account global informations of the overlap region, the watershed algorithm defines a seam which is connected, irregular and thus more realistic than the existing methods for photomosaicking.

# Conteúdo

<b>1. Introdução</b> .....	<b>1</b>
<b>2. O Problema de Criação de Fotomosaicos</b> .....	<b>4</b>
2.1 Introdução.....	4
2.2 Descrição do Problema.....	5
2.3 Alguns Métodos de Construção de Fotomosaicos.....	7
2.3.1 Método de Milgram.....	7
2.3.2 Método de Shiren .....	11
2.3.3 Linha de Costura Irregular .....	15
<b>3. Breve Introdução à Morfologia Matemática</b> .....	<b>16</b>
3.1 Introdução.....	16
3.2 Operações Morfológicas.....	16
3.3 Transformações Morfológicas.....	18
3.3.1 Transformações Clássicas .....	18
3.3.2 Transformações HMT.....	19
3.4 Operações Morfológicas de Base : Erosão e Dilatação .....	20
3.4.1 Definição .....	20
3.4.2 Extensões : Imagens em Níveis de Cinza .....	21
3.5 Abertura e Fechamento.....	23
3.5.1 Definição .....	23
3.6 Algumas Transformações Morfológicas .....	25
3.6.1 Afinamento e Espessamento.....	25
3.6.1.1 Definição .....	25
3.6.1.2 Extensões ao Caso Numérico .....	27
3.6.2 Espessamento Homotópico e Esqueleto por Zona de Influência .....	27
3.7 Transformações Geodésicas.....	29

---

3.7.1 Distâncias Geodésicas.....	29
3.8 Linha de Divisor de Águas.....	32
3.8.1 Definição.....	32
3.8.2 Implementação.....	33
<b>4. O Algoritmo.....</b>	<b>36</b>
4.1 Introdução.....	36
4.2 Descrição do Algoritmo.....	37
4.2.1 Definição da Linha de Costura Final.....	40
4.2.2 Eliminação de Caminhos Opcionais.....	42
4.2.3 Composição do Fotomosaico.....	43
4.3 Comentários Finais.....	45
4.3.1 Índice de Correlação.....	45
4.3.2 As Bordas da Imagem de Correlação.....	48
4.3.3 Informações Locais versus Informações Globais.....	49
<b>5. Conclusão.....</b>	<b>52</b>
<b>Apêndice A - Fontes de Erros.....</b>	<b>53</b>
A.1 Tipos de Distorções.....	53
A.2 Correção Radiométrica.....	58
A.3 Correção Geométrica.....	60
<b>Apêndice B - Listagens.....</b>	<b>62</b>
B.1 Definições de Tipos.....	62
B.2 Afinamento Numérico.....	63
B.3 Divisor de Águas.....	65
B.4 Seleção da Costura a partir do Divisor de Águas.....	70
<b>Bibliografia.....</b>	<b>75</b>
<b>Índice Remissivo.....</b>	<b>79</b>

## Lista de Figuras

Figura 2.1 : Imagens componentes de um fotomosaico .....	5
Figura 2.2 : Uma região de interseção e sua costura .....	6
Figura 2.3 : Perfil de linha das imagens que se interceptam .....	6
Figura 2.4 : Princípio do método de Milgram .....	9
Figura 2.5 : Fotomosaico construído a partir do método de Milgram .....	10
Figura 2.6 : Descontinuidade vertical em uma costura monodimensional.....	11
Figura 2.7 : Procura bidimensional dos pontos de costura.....	13
Figura 2.8 : Fotomosaico construído a partir do método de Shiren .....	14
Figura 3.1 : Exemplo de transformações clássicas.....	18
Figura 3.2 : Família de configurações de um elemento estruturante.....	20
Figura 3.3 : Elemento estruturante 4- e 8-conectado.....	20
Figura 3.4 : Exemplo de erosão e dilatação.....	21
Figura 3.5 : Exemplo de erosão e dilatação numérica.....	22
Figura 3.6 : Exemplo de abertura e fechamento .....	24
Figura 3.7 : Exemplo de abertura e fechamento numérico .....	24
Figura 3.8 : Famílias homotípicas $M$ e $L$ , em malha 4- e 8-conexas .....	26
Figura 3.9 : Exemplo de afinamento e espessamento.....	26
Figura 3.10 : Exemplo de esqueleto por zona de influência .....	28
Figura 3.11 : Famílias $E$ , em malha 4- e 8-conexas.....	28
Figura 3.12 : Definição de distância geodésica.....	29
Figura 3.13 : Exemplo de círculo geodésico .....	30
Figura 3.14 : Exemplo de dilatação e erosão geodésica.....	31
Figura 3.15 : Exemplo de SKIZ geodésico .....	32
Figura 3.16 : Definição intuitiva de LDA.....	33
Figura 3.17 : Processo iterativo de definição da LDA.....	34
Figura 3.18 : Processo de obtenção da LDA.....	35

---

Figura 4.1 : Parte superior do fotomosaico.....	38
Figura 4.2 : Parte inferior do fotomosaico.....	39
Figura 4.3 : LDA obtida da correlação entre as imagens da Figura 4.1 e Figura 4.2.....	39
Figura 4.4 : Dilatação geodésica do conjunto $\partial S$ .....	41
Figura 4.5 : Resultado do cálculo da linha ótima.....	43
Figura 4.6 : Fotomosaico morfológico.....	44
Figura 4.7 : LDA obtida de correlação 9x9.....	46
Figura 4.8 : Fotomosaico morfológico utilizando uma correlação 9x9.....	47
Figura 4.9 : Algumas posições entre as imagens de um mosaico.....	48
Figura 4.10 : Direção das LDA relativa às bordas da imagem de correlação.....	49
Figura 4.11 : Imagem de correlação com uma zona de baixa correlação.....	49
Figura 4.12 : Definição da costura baseada no método de Milgram.....	50
Figura 4.13 : Definição da costura baseada no método morfológico.....	51
Figura 5.1 : Tela do programa de photomosaicking.....	52
Figura A.1 : Varredura multiespectral.....	53
Figura A.2 : Distorções comuns em imagens de sensoriamento remoto.....	54
Figura A.3 : Distorções de atitude.....	55

## Lista de Equações

Equação 2.1 .....	8
Equação 2.2 .....	8
Equação 2.3 .....	12
Equação 2.4 .....	12
Equação 2.5 .....	12
Equação 2.6 .....	12
Equação 2.7 .....	12
Equação 2.8 .....	13
Equação 2.9 .....	13
Equação 2.10 .....	13
Equação 2.11 .....	13
Equação 3.1 .....	19
Equação 3.2 .....	20
Equação 3.3 .....	20
Equação 3.4 .....	21
Equação 3.5 .....	21
Equação 3.6 .....	21
Equação 3.7 .....	22
Equação 3.8 .....	22
Equação 3.9 .....	22
Equação 3.10 .....	22
Equação 3.11 .....	23
Equação 3.12 .....	23
Equação 3.13 .....	24
Equação 3.14 .....	24
Equação 3.15 .....	25

---

Equação 3.16 .....	26
Equação 3.17 .....	26
Equação 3.18 .....	27
Equação 3.19 .....	27
Equação 3.20 .....	27
Equação 3.21 .....	27
Equação 3.22 .....	28
Equação 3.23 .....	28
Equação 3.24 .....	30
Equação 3.25 .....	30
Equação 3.26 .....	31
Equação 3.27 .....	31
Equação 3.28 .....	31
Equação 3.29 .....	31
Equação 3.30 .....	33
Equação 3.31 .....	33
Equação 3.32 .....	33
Equação 4.1 .....	37
Equação 4.2 .....	40
Equação 4.3 .....	41
Equação 4.4 .....	45
Equação 4.5 .....	46
Equação 4.6 .....	46

# 1. Introdução

Através de imagens de sensoriamento remoto podemos obter um grande número de informações sobre os recursos do nosso planeta. Estas informações podem ser úteis para diversos propósitos, como mapeamento geológico [Abrams et al. (1985)]; estudo de cobertura vegetal [Stone et al. (1994)]; planejamento e gerenciamento agrícola [Thenkabail et al. (1994)]; distribuição e inspeção de empréstimos agrícolas com acompanhamento da colheita; detecção de áreas de queimadas; estimativa de produção agrícola e gerenciamento dos recursos hídricos disponíveis para necessidades agrícolas [Thiruvengadachari (1981)]; controle da qualidade da água armazenada em grandes reservatórios [Verdin (1985)], etc.

Freqüentemente, para obter as informações acima, nós necessitamos considerar o problema em que o estudo de uma região necessita estender-se por cenas vizinhas contidas em imagens diferentes. Neste caso, torna-se necessário unir as cenas das regiões vizinhas, adquiridas eventualmente por um ou mais sensores diferentes. O processo de composição destas imagens é chamado de *photomosaicking*. Através de *photomosaicking* nós podemos, ainda, reconstruir regiões afetadas por efeitos espúrios, como cobertura de nuvens e sombras [Hummer-Miller (1989)].

Um fotomosaico é definido considerando uma região de interseção entre duas ou mais imagens, na qual definimos a linha de junção ou costura. Esta costura indica onde uma imagem termina e a outra começa.

As imagens que compõem o fotomosaico deverão ser selecionadas de tal modo que aspectos como mudanças sazonais ou horárias, por exemplo, não comprometam a qualidade da imagem resultante. Estes aspectos podem gerar fotomosaicos onde regiões descontínuas, representadas por variações abruptas de pixels, sejam claramente identificadas.

Imagens de sensoriamento remoto são obtidas por sensores que captam uma determinada frequência do espectro eletromagnético. Estes sensores estão sujeitos a erros

ocasionados por falhas em seu dispositivo de captação. Além das falhas do próprio sensor, a plataforma de aquisição, onde se encontra o sensor, pode sofrer influência da imprecisão do seu mecanismo de navegação. Além disso, a atmosfera terrestre também pode afetar a qualidade e a precisão da informação adquirida.

Como consequência do conjunto sensor/plataforma/cena, os erros são agrupados em duas categorias. Os *erros geométricos* que alteram a relação entre o valor medido (ou captado) e a sua correta localização dentro da cena; e os *erros radiométricos* que alteram a relação entre a informação medida e seu valor real.

Além dessa classificação, podemos dividir os tipos de erros, baseado em sua fonte (ou origem), em 5 categorias [Bernstein (1976)]: *efeitos da plataforma* (ocasionados por variações de altitude, atitude, velocidade nominal da plataforma, etc); *efeitos da cena* (ocasionados pela rotação da Terra e por transformações de projeção); *efeitos do sensor* (ocasionados por falhas no detector, falhas no conversor analógico digital, etc); *efeitos da cena e do sensor* (ocasionados pela vista ou visada do sensor e pela perspectiva sob a qual o sensor capta a cena); e *efeitos atmosféricos* (ocasionados pelo *espalhamento* - difusão de parte da radiação incidente em diversas direções, e *absorção* - retenção e conversão de parte da energia radiante em outras formas de energia).

O procedimento desenvolvido para correção de imagens de satélites é baseado na mudança dos dados do sensor (intensidade) para corrigir erros radiométricos, e na relocação dos pontos da imagem, de modo a melhorar suas propriedades geométricas. Para corrigir a geometria, é necessário computar os coeficientes de um modelo matemático que descreve as distorções na imagem, associando-os a uma função de mapeamento que corrige este tipo de erro [Colwell (1983)] (o Apêndice A apresenta, a título de informação, um breve comentário adicional sobre fontes de erros de imagens de sensoriamento remoto).

Após esta etapa do processamento, os diferentes segmentos de uma imagem representando uma grande cena podem ser combinadas através de técnicas de

photomosaicking<sup>1</sup>. Neste trabalho nós descrevemos um método morfológico para a composição de fotomosaicos. Como veremos, a partir da noção de linha de divisor de águas de uma função, o método extrai informação global de uma imagem de correlação para detectar áreas homogêneas na qual a costura deve estar definida.

O aspecto estritamente local dos algoritmos de photomosaicking leva à definição de uma linha de costura regular, com um único ponto por linha. Nosso método define uma linha de costura irregular, e portanto, mais realista, cuja estrutura depende de características globais da imagem de interseção.

Este trabalho está organizado da seguinte forma. O capítulo 2 apresenta uma descrição do problema de photomosaicking e ilustra os métodos existentes. O capítulo 3 introduz alguns conceitos de Morfologia Matemática. O capítulo 4 descreve o algoritmo morfológico de construção de fotomosaicos. Finalmente, a conclusão é apresentada no capítulo 5.

---

<sup>1</sup> Antes disto, a informação geo-referenciada da imagem é considerada para estabelecer o real posicionamento entre as imagens que compõem o fotomosaico. Neste caso, essas imagens estão associadas a uma projeção que relaciona cada um dos seus pontos ao seu posicionamento geográfico. A partir destas informações pode-se definir facilmente um retângulo representando a região de interseção entre as duas imagens.

## 2. O Problema de Criação de Fotomosaicos

### 2.1 Introdução

A partir da década de 70, um grande número de aplicações começaram a necessitar, para fins diversos, da combinação ou “photomosaicking” de muitas cenas provenientes de satélites [Colwell (1983)]. Naquela época, o Departamento do Interior dos Estados Unidos era um dos principais interessados nesta operação. Um dos objetivos era a possibilidade de utilização de imagens obtidas a baixas altitudes, e portanto com uma maior quantidade de informação, como fonte de dados sobre vastas regiões. A utilidade dessas imagens de satélites seria ampliada se as informações pudessem ser obtidas a partir de cenas arbitrárias resultantes do photomosaicking de cenas que se recobrem, com um mínimo de costuras visíveis [Zobrist et al. (1983)].

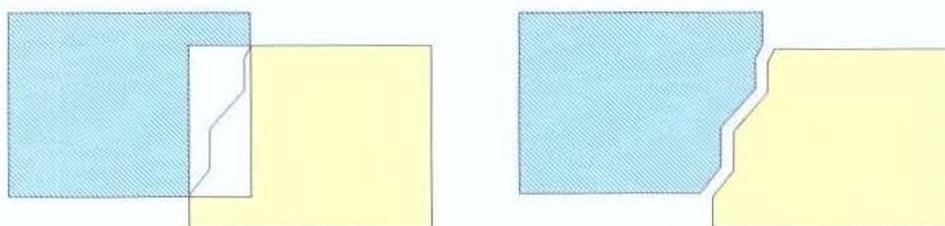
Em 1977, técnicas especializadas foram desenvolvidas para o photomosaicking de imagens do Landsat MSS no Jet Propulsion Laboratory (JPL) [Zobrist (1978)], [Zobrist-Bryant (1979)]. Estas técnicas compreendiam o uso de informações de controle obtidas de mapas, integração de informações geométricas de controle, e o uso de um sistema de informação baseado em imagens (IBIS - Image Based Information System) [Zobrist-Bryant (1979)].

Em 1978, grandes fotomosaicos digitais do Landsat foram preparados pela IBM [Bernstein (1974)] e no USGS (United States Geological Survey - a principal agência de mapeamento dos Estados Unidos) [Chavez (1977)]. Muitos desses esforços eram voltados principalmente para o processo de retificação das imagens [Moik (1980)]. As técnicas mais recentes de photomosaicking são baseadas, sobretudo, em critérios locais de otimização visando diminuir os efeitos indesejáveis das linhas de costura [Milgram (1975)], [Milgram (1977)], [Murai-Okuda (1980)], [Peleg (1981)], [Shiren et al. (1989)]. Alguns destes métodos podem ser associados à operação de transformação histogrâmica, e conseqüente alteração da informação, que amenizem estes efeitos [Milgram (1977)].

A seguir daremos uma descrição geral do problema de criação de fotomosaicos e abordaremos dois algoritmos bastante mencionados na literatura para ilustrar o princípio básico e as considerações estritamente locais utilizadas na sua definição.

## 2.2 Descrição do Problema

Excetuando-se o processo de registro, necessário em uma etapa inicial, a criação de fotomosaicos consiste basicamente da definição dos pontos de cada imagem que irão compor a linha de junção ou costura da imagem resultante. A linha de costura está geodesicamente localizada nas duas imagens adjacentes, em sua região de interseção. Tomemos como exemplo o fotomosaico horizontal mostrado na Figura 2.1, com uma imagem à esquerda e outra à direita. Os dados situados à esquerda do limite serão excluídos da imagem da direita. De forma semelhante, os dados situados à direita do limite serão excluídos da imagem da esquerda. Assim, as duas imagens irão se unir e formar o fotomosaico.



*Figura 2.1 : Imagens componentes de um fotomosaico.*

O problema neste caso consiste, principalmente, da definição da linha de costura entre as duas imagens. As imagens devem ser combinadas de tal modo que a imagem resultante não tenha arestas artificiais espúrias. Estas arestas são representadas por diferenças perceptíveis entre os pontos dentro da região de interseção [Milgram (1977)].

De um modo geral, os algoritmos para a construção de fotomosaicos são definidos através da localização de pontos cuja percepção visual, na transição para o ponto seguinte da costura, é a mínima possível, atenuando as discontinuidades geradas pelo photomosaicking. A Figura 2.2 mostra duas imagens e uma costura dentro da região de

interseção. A partir do perfil de linha das duas imagens (Figura 2.3), podemos dizer que os pontos da costura devem ser aqueles cuja intensidade é a mais próxima possível, entre os perfis de linha.

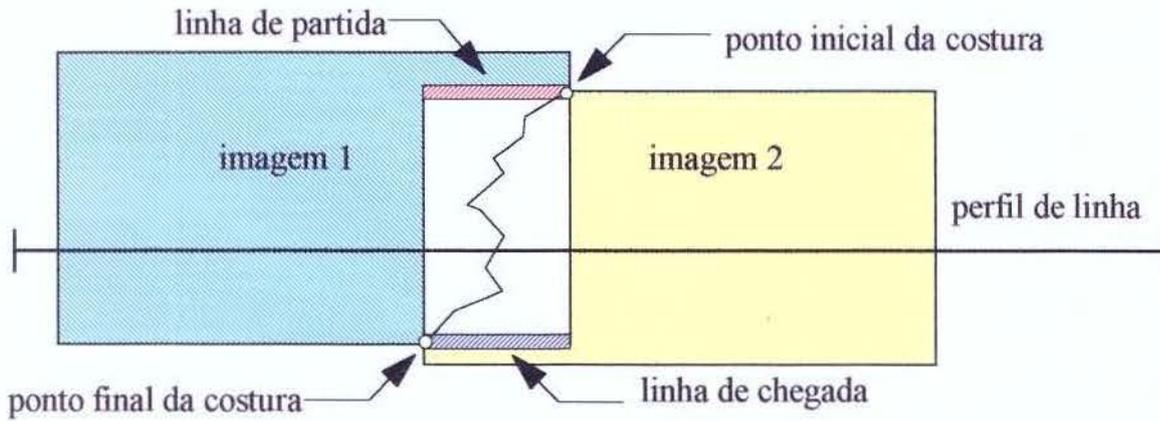


Figura 2.2 : Uma região de interseção e sua costura.

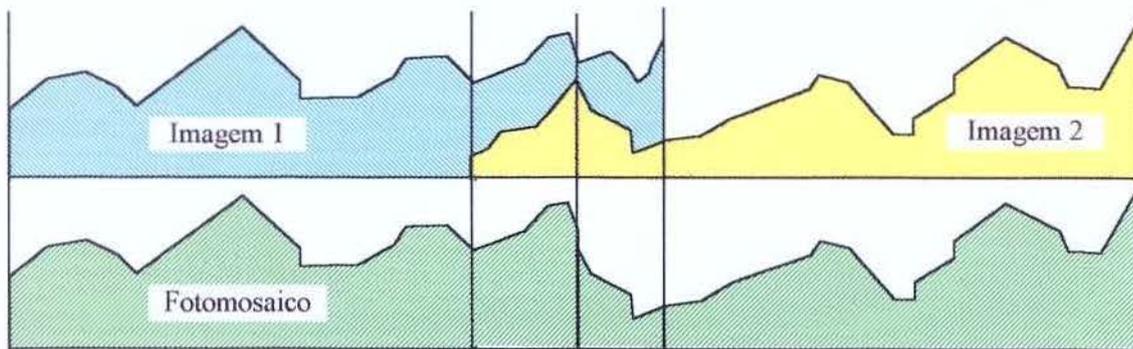


Figura 2.3 : Perfil de linha das imagens que se interceptam.

## 2.3 Alguns Métodos de Construção de Fotomosaicos

Os algoritmos existentes para criação de fotomosaicos [Milgram (1975)], [Milgram (1977)], [Murai-Okuda (1980)], [Peleg (1981)], [Shiren et al. (1989)] definem uma costura com somente um ponto por linha, para um fotomosaico horizontal, ou coluna, para um fotomosaico vertical (veja, mais adiante, a descrição de dois desses métodos). Ao considerar uma vizinhança mono-dimensional, os autores, em geral, definem uma costura baseada na diferença entre os pontos da suas regiões de interseção. Na definição de um fotomosaico horizontal (respectivamente, vertical) o método tem a desvantagem de introduzir descontinuidades na direção vertical (respectivamente, horizontal). Uma operação para localmente suavizar a aresta artificial resultante através de uma função rampa é discutida em [Milgram (1977)]. Obviamente, esta suavização pode representar alteração na informação original ao longo da vizinhança da costura.

Apresentamos a seguir, duas importantes abordagens para a construção de fotomosaicos.

### 2.3.1 Método de Milgram

Um método muito conhecido para construção de fotomosaicos é descrito em [Milgram (1975)], [Milgram (1977)]. A primeira versão do método de Milgram consiste basicamente de um ajuste histogrâmico e de uma busca de similaridades, linha à linha (para um fotomosaico horizontal), dos pontos que compõem a costura.

O ajuste histogrâmico é utilizado para compensar fatores sazonais e meteorológicos que tendem a tornar diferente o sinal captado. Este ajuste é realizado através de um deslocamento do histograma, de modo que estas curvas tenham suas médias coincidentes [Milgram (1975)]. Embora este seja um procedimento comumente utilizado para esses propósitos, o ajuste histogrâmico representa, naturalmente, a perda da informação original.

A similaridade é medida pela variação entre as duas imagens,  $D$  e  $E$ ,

respectivamente à direita e à esquerda (para um fotomosaico horizontal), representada pelo somatório de diferença absoluta,  $S_{x,y}$ , dos pontos  $(x, y)$  em um vizinhança  $u$ , ou seja:

$$S_{x,y} = \sum_{i=-\frac{u}{2}+1}^{\frac{u}{2}} |D_{x+i,y} - E_{x+i,y}| \quad \text{Eq. 2.1}$$

Assim, a posição  $x$  escolhida a cada linha  $y$ , é aquela que apresentar o menor valor de  $S_{x,y}$ . O valor da faixa de busca de  $x$  varia entre  $\frac{u}{2}$  e  $L - \frac{u}{2}$  ( $L$  é a largura da linha). Uma falha deste método é não garantir que o ponto final (Figura 2.2) será alcançado pela costura [Milgram (1977)], além da possibilidade de gerar grandes arestas artificiais entre dois pontos consecutivos da mesma.

Em uma nova versão deste método [Milgram (1977)], dado um ponto inicial para a linha de costura, os pontos seguintes são selecionados numa faixa  $\frac{u}{2}$  de pontos à esquerda e à direita da posição do último ponto escolhido para costura (Figura 2.4). O ponto selecionado deve ser aquele de maior similaridade. A similaridade agora é medida pela variação entre as duas imagens, representada pela diferença absoluta  $S_{x,y}$  entre cada ponto  $(x,y)$ , isto é :

$$S_{x,y} = |D_{x,y} - E_{x,y}| \quad \text{Eq. 2.2}$$

Além disso, associa-se à costura uma função custo  $C$ , definida como a soma de todos os valores de diferenças absolutas dos pontos da costura. Para isso uma imagem com os valores de diferenças absolutas será construída, e para cada ponto desta imagem será atribuído um valor que mede o custo da costura até aquele ponto. Um apontador para a posição do ponto de costura anterior deve ser definido. Dessa forma todos os pontos da imagem de diferenças absolutas pertencem a uma malha de costuras, onde cada ponto tem um custo associado a sua posição. A linha de costura do fotomosaico será aquela de menor custo, acumulado no final da malha.

A construção dessa malha é feita da seguinte forma. Os pontos de partida da

linha inicial (Figura 2.2) têm, inicialmente, custo zero e os demais pontos desta linha, custo infinito. Para as linhas  $y$ , subsequentes, o custo  $C_{x,y}$  de cada ponto  $(x, y)$  é a soma do custo do ponto precedente (na linha anterior) com o valor de diferença absoluta do próprio ponto. A seleção do melhor ponto precedente é realizada através da busca daquele ponto que tiver menor custo, numa faixa de vizinhança  $\frac{u}{2}$  à esquerda e à direita do ponto em questão. Como o custo é cumulativo, a seleção da linha de costura apropriada se dá pela escolha do ponto de chegada de menor custo (Figura 2.4).

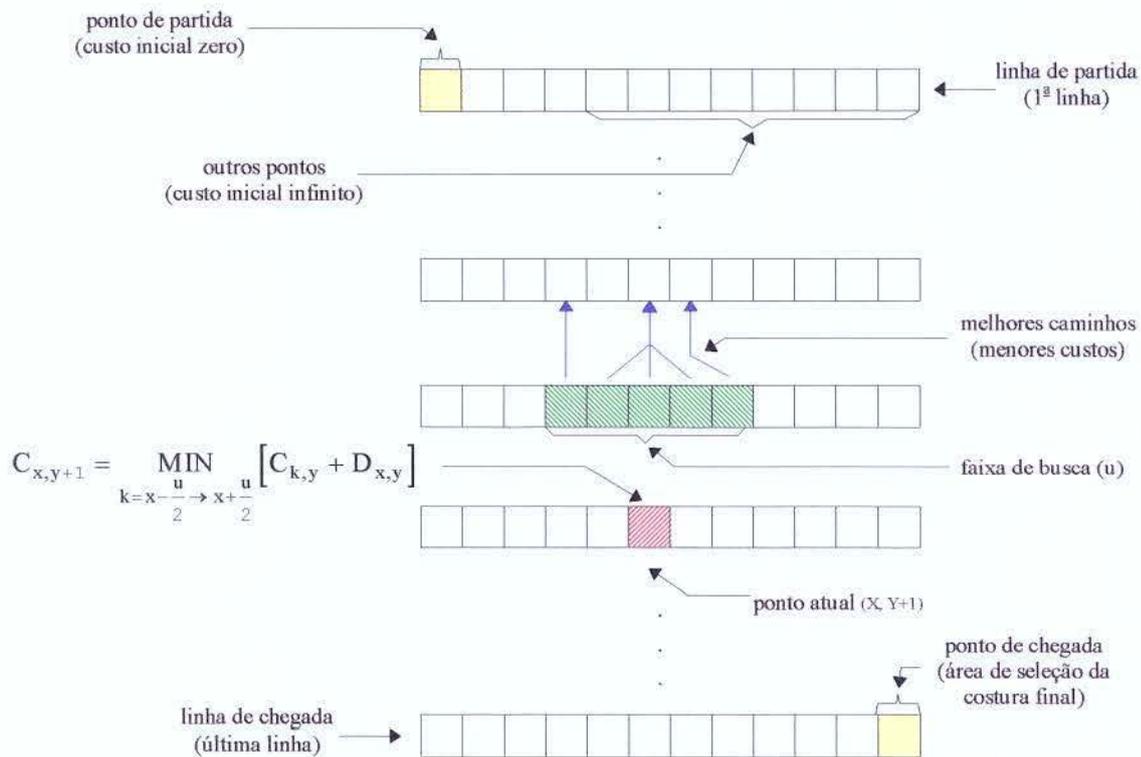
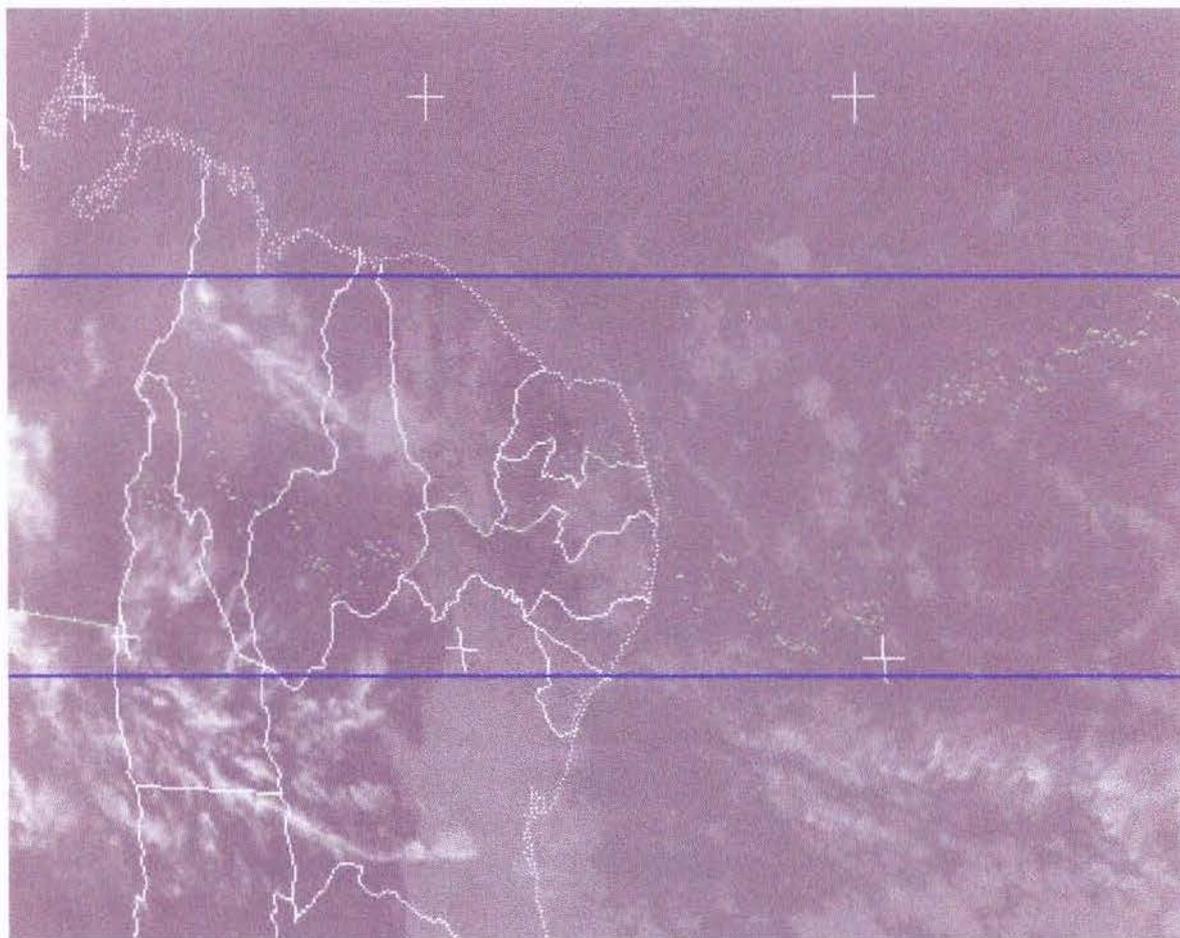


Figura 2.4 : Princípio do método de Milgram [Milgram (1977)].

Um exemplo de um fotomosaico vertical realizado com este método pode ser visto na Figura 2.5. Observe que devido à restrição espacial na busca dos pontos da costura, ela pode não acompanhar livremente os pontos de maior correlação. Aumentando-se a faixa de busca  $u$  pode-se melhorar o resultado ( $u = 10$ , no exemplo mostrado na Figura 2.5), ampliando-se, porém, os efeitos das arestas artificiais.

Como forma de amenizar os efeitos das arestas artificiais, o autor sugere uma suavização ao longo da normal a linha que une dois pontos consecutivos da costura, através de uma função rampa. Embora este procedimento reduza seus efeitos, ela torna-se desaconselhável quando a informação original não pode ser alterada.



*Figura 2.5 : Fotomosaico construído a partir do método de Milgram.*

### 2.3.2 Método de Shiren

O método de definição dos pontos de costura de Milgram mede as discontinuidades de forma monodimensional, minimizando as diferenças entre as imagens no sentido perpendicular ao sentido de busca dos pontos (Figura 2.6). Assim, num fotomosaico horizontal, onde a busca da linha de costura é realizada em direção vertical, a atenuação destas discontinuidades só ocorre na direção horizontal, entre o lado esquerdo e o lado direito da linha de costura. Esta discontinuidade evidencia o que se chama de “aresta artificial”, identificada por uma diferença abrupta ao longo da linha que une dois pontos consecutivos da costura (seta amarelas na Figura 2.6). Estas arestas artificiais podem ser minimizadas através da redução da faixa de busca  $u$ , como no método de Milgram. Por outro lado, esta restrição espacial diminui as chances de se obter uma linha de costura melhor.

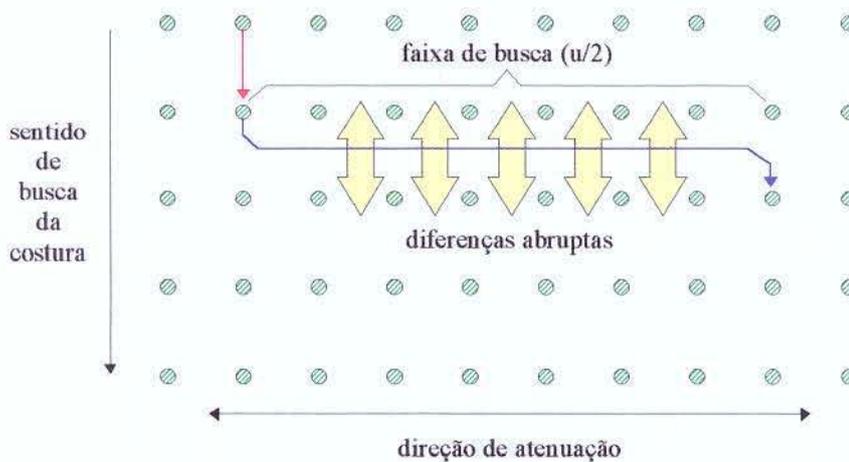


Figura 2.6 : Descontinuidade vertical em uma costura monodimensional.

O método de Shiren [Shiren et al (1989)] considera o espaço bidimensional, durante a busca dos pontos da costura do fotomosaico. Sejam  $V_{x,y}$  e  $H_{x,y}$  medições da aresta vertical e horizontal, respectivamente, isto é :

$$V_{x,y} = \sum_{i=-u/2}^{u/2} S_{x,y+i} \quad \text{Eq. 2.3}$$

e,

$$H_{x,y} = \sum_{i=-u/2}^{u/2} S_{x+i,y} \quad \text{Eq. 2.4}$$

O algoritmo consiste da busca seqüencial dos pontos de costura, linha à linha, de modo a minimizar estas arestas artificiais, considerando os valores  $V_{x,y}$  e  $H_{x,y}$  numa vizinhança  $u$  do último ponto de costura escolhido.

Neste caso, se um ponto de costura na linha  $y$  é  $(x, y)$ , então o ponto  $(x', y+1)$  de costura da linha  $y+1$  (no caso de um fotomosaico de costura vertical) deve ser procurado seqüencialmente nos dois lados deste ponto.

Assim, seja, para o lado direito,

$$V_{x,y}^d(k) = \text{MIN}(V_{x,y+1}, V_{x+1,y+1}, \dots, V_{x+k,y+1}) \quad \text{Eq. 2.5}$$

a menor medida vertical da aresta, para os pontos  $(x, y+1)$ ,  $(x+1, y+1)$ , ... ,  $(x+k, y+1)$ , na linha  $y+1$ , e seja

$$H_{x,y}^d(k) = \text{MAX}(H_{x,y+1}, H_{x+1,y+1}, \dots, H_{x+k,y+1}) \quad \text{Eq. 2.6}$$

a maior medida horizontal de aresta, à direita, para os pontos  $(x, y+1)$ ,  $(x+1, y+1)$ , ... ,  $(x+k, y+1)$ , na linha  $y+1$ . Assim, a procura para o lado direito é feita considerando

$k = 0, 1, 2, \dots, L-x-\frac{u}{2}$  ( $L$  é a largura da imagem de diferença absoluta), se a diferença

entre  $V_{x,y}^d(k)$  e  $H_{x,y}^d(k)$  é menor ou igual a um limiar  $T$  predefinido, isto é,

$$\frac{1}{u+1} \cdot |V_{x,y}^d - H_{x,y}^d| \leq T \quad \text{Eq. 2.7}$$

Caso contrário, encerra-se a busca pelo lado direito, recomeçando o procedimento pelo lado esquerdo. O valor de T considerado pode variar entre  $\frac{1}{32}$  e  $\frac{1}{64}$  do nível de cinza máximo presente na imagem.

De forma semelhante, seja

$$V_{x,y}^e(k) = \text{MIN}(V_{x-1,y+1}, V_{x-2,y+1}, \dots, V_{x-k,y+1}) \tag{Eq. 2.8}$$

e

$$H_{x,y}^e(k) = \text{MAX}(H_{x-1,y+1}, H_{x-2,y+1}, \dots, H_{x-k,y+1}) \tag{Eq. 2.9}$$

respectivamente, a menor e maior medida vertical e horizontal de aresta para os pontos  $(x-1, y+1), (x-2, y+1), \dots, (x-k, y+1)$ , na linha  $y+1$ , para  $k = 1, 2, \dots, \frac{u}{2}$ , se

$$\frac{1}{u+1} \cdot |V_{x,y}^e - H_{x,y}^e| \leq T \tag{Eq. 2.10}$$

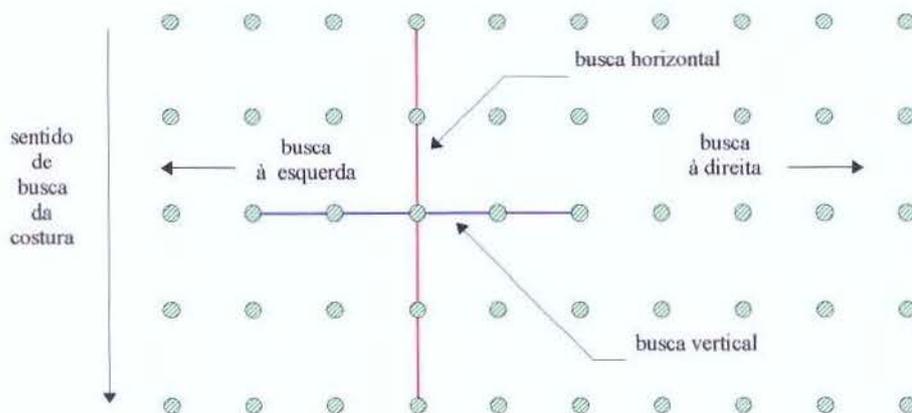
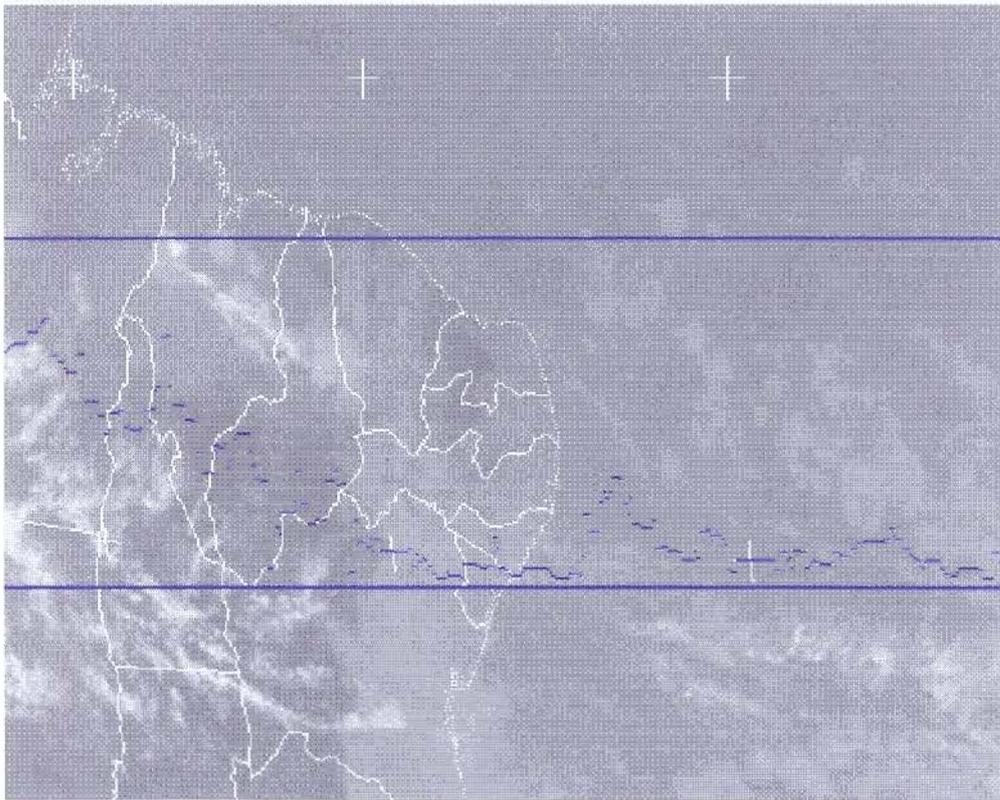


Figura 2.7: Procura bidimensional dos pontos de costura [Shiren et al (1989)].

Depois de terminar a busca nos dois sentidos, o melhor ponto de costura na linha  $y+1$  será o ponto  $(x', y+1)$  definido pela Eq. 2.11.

$$V_{x',y+1}(k) = \text{MIN}(V_{x,y+1}^d(k), V_{x,y+1}^e(k)) \quad \text{Eq. 2.11}$$

Com este método, não existe significativa distinção entre arestas verticais e horizontais se o limiar  $T$  é definido com um valor menor do que a sensibilidade do olho à diferenças de brilho de níveis de cinza adjacentes [Shiren et al (1989)]. Um exemplo de um fotomosaico vertical realizado com este método pode ser visto na Figura 2.8.



*Figura 2.8 : Fotomosaico construído a partir do método de Shiren.*

Mais uma vez observe que critérios locais de busca podem impor a passagem da linha de costura em regiões com variações abruptas de níveis de cinza.

### 2.3.3 Linha de Costura Irregular

Neste trabalho consideramos o problema da construção de fotomosaicos a partir de uma linha de costura irregular. Esta linha atravessa a região de interseção de forma desconhecida, podendo definir qualquer caminho na região de interseção e, conseqüentemente, ter mais de um ponto por linha ou coluna.

Como veremos mais adiante, esta costura mais natural é definida graças a uma operação morfológica de segmentação definida a partir da extração de características globais da imagem.

## 3. Breve Introdução à Morfologia Matemática

### 3.1 Introdução

Há muitas maneiras de abordar a descrição de fenômenos que se estendem no espaço [Serra (1981)]. Uma destas abordagens é considerá-los como objetos, isto é, como subconjuntos de seu espaço de definição. A análise destes fenômenos enquanto objetos pode ser resumida na observação de sua geometria e forma. A idéia básica é compará-los com objetos - *elementos estruturantes* - de forma conhecida. O método que deriva deste ponto de vista é chamado de Morfologia Matemática. Neste capítulo introduzimos algumas noções básicas de Morfologia Matemática. Para maiores detalhes consultar, por exemplo, [Serra (1981)], [Serra (1986)], [Sternberg (1986)], [Haralick et al. (1987)], [Banon-Barrera (1994)].

### 3.2 Operações Morfológicas

As operações morfológicas, baseadas na teoria dos conjuntos, visam quantificar a descrição de uma estrutura. Esta operação pode ser um tanto quanto subjetiva, com muitas ambigüidades e propósitos. Não se pode associar esta quantificação tão somente a um fenômeno particular, como por exemplo, ao estudo e à classificação de um determinado tipo de vegetal ou solo. A atenção maior pode estar em aspectos espaciais do objeto tais como suas características, suas dimensões, suas orientações, suas relações com o meio, etc [Serra (1981)].

Inicialmente, e de uma maneira geral, nós temos apenas um conhecimento intuitivo sobre a estrutura espacial dos objetos. Morfológicamente, mensurar as características de um objeto nada mais é do que procurar ver neste objeto aquelas características conhecidas que desejamos encontrar. As operações que realizamos sobre este objeto de estudo são denominadas “*Operações Morfológicas*”. As características conhecidas que utilizamos num processo de comparação são os “*elementos estruturantes*”.

A comparação do objeto de estudo com os elementos estruturantes se dá através de operações que transformam o objeto segundo esta comparação. As informações

extraídas da imagem são, na verdade, medições do efeito destas operações sobre o objeto. Isto é, o resultado de uma ou várias transformações  $\phi(X)$  é seguido de uma medição  $W(\phi(X))$ , onde  $X$  representa um objeto e  $\phi$  uma transformação morfológica qualquer. O comportamento destas transformações nos dá uma idéia das reais características da imagem original.

### 3.3 Transformações Morfológicas

De um modo geral, as transformações morfológicas podem ser classificadas em dois tipos :

1. Transformações clássicas sobre conjuntos : União, Interseção, Complemento, Diferença, Diferença Simétrica.
2. Transformações em Tudo ou Nada (HMT - Hit or Miss Transformation).

#### 3.3.1 Transformações Clássicas

Seja  $f(x)$  uma imagem binária discreta, isto é,  $\{f(x) \in [0,1] : x \in Z^2\}$ , onde  $Z$  denota o conjunto dos números inteiros. Esta imagem pode ser representada por um conjunto  $X$  dado por  $X = \{x \in Z^2 : f(x) = 1\}$ .

Sejam duas imagens  $X$  e  $Y$ . O complemento  $X^c$  do conjunto  $X$  é  $X^c = \{x \in Z^2 : f(x) = 0\}$ . A união  $X \cup Y$  de  $X$  com  $Y$  é  $Z = \{a \in Z^2 : a \in X \text{ ou } a \in Y\}$ . A interseção  $X \cap Y$  de  $X$  com  $Y$  é  $Z = \{a \in Z^2 : a \in X \text{ e } a \in Y\}$ . A diferença  $X \setminus Y$  de  $X$  com  $Y$  é definida como  $X \cap Y^c$ . A diferença simétrica  $X - Y$  de  $X$  com  $Y$  é definida como  $(X \cup Y) \cap (X \cap Y)^c$  (Figura 3.1).

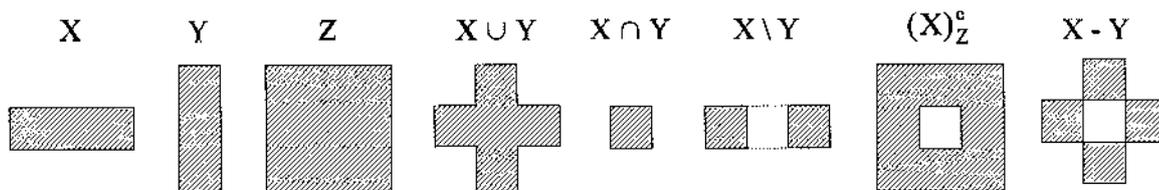


Figura 3.1. Exemplo de transformações clássicas

Denotamos, ainda, a translação  $X_u$  de um conjunto  $X$  por um vetor  $u$  por

$$X_u = \{ z \in Z^2 \mid z = x + u, x \in X \}$$

### 3.3.2 Transformações HMT

As transformações HMT resultam em um novo subconjunto caracterizando a estrutura de um objeto  $X$ . Esta informação estrutural é representada por transformações da própria imagem, e a sua medida é o resultado da quantização da transformação resultante.

Seja um elemento estruturante  $B_X = (B_X^1, B_X^2)$ , centrado em  $x \in X$  e constituído de *duas fases*,  $B_X^1$  e  $B_X^2$ , com mesma origem. A HMT geral ou transformação de vizinhança para este elemento é dada por:

$$\forall x \in Z^2, X \otimes B = \{ x \in X : B_X^1 \subset X, B_X^2 \subset X^c \} \quad \text{Eq. 3.1}$$

A partir de um elemento estruturante podemos definir uma família de configurações, representada por rotações deste em torno do seu elemento central. A Figura 3.2 apresenta um elemento estruturante  $B_X$ ,  $3 \times 3$ , e sua família de configurações (rotações de  $45^\circ$ ). O asterisco “\*” indica que o ponto pode pertencer tanto à fase  $B_1$  quanto à fase  $B_2$ .

Informalmente, a transformação HMT é realizada através de uma varredura de um elemento estruturante  $B$ , de forma conhecida, sobre uma cena  $X$ . Este elemento  $B$  é deslocado sobre  $X$  de tal forma que sua origem (previamente conhecida) passe por todos os pontos do espaço da cena. Para cada posição da origem  $B_X$  de  $B$ , em  $X$ , é feita uma pergunta relativa à união, interseção ou inclusão de  $B$  com  $X$ . O conjunto dos pontos correspondendo a respostas positivas constitui o novo conjunto da imagem transformada.

$$\begin{aligned}
 B_x^0 &= \begin{bmatrix} B_1 & B_1 & B_1 \\ * & B_1 & * \\ B_2 & B_2 & B_2 \end{bmatrix} & B_x^1 &= \begin{bmatrix} * & B_1 & B_1 \\ B_2 & B_1 & B_1 \\ B_2 & B_2 & * \end{bmatrix} & B_x^2 &= \begin{bmatrix} B_2 & * & B_1 \\ B_2 & B_1 & B_1 \\ B_2 & * & B_1 \end{bmatrix} & B_x^3 &= \begin{bmatrix} B_2 & B_2 & * \\ B_2 & B_1 & B_1 \\ * & B_1 & B_1 \end{bmatrix} \\
 B_x^4 &= \begin{bmatrix} B_2 & B_2 & B_2 \\ * & B_1 & * \\ B_1 & B_1 & B_1 \end{bmatrix} & B_x^5 &= \begin{bmatrix} * & B_2 & B_2 \\ B_1 & B_1 & B_2 \\ B_1 & B_1 & * \end{bmatrix} & B_x^6 &= \begin{bmatrix} B_1 & * & B_2 \\ B_1 & B_1 & B_2 \\ B_1 & * & B_2 \end{bmatrix} & B_x^7 &= \begin{bmatrix} B_1 & B_1 & * \\ B_1 & B_1 & B_2 \\ * & B_2 & B_2 \end{bmatrix}
 \end{aligned}$$

Figura 3.2 : Família de configurações de um elemento estruturante  $B_x$ .

### 3.4 Operações Morfológicas de Base : Erosão e Dilatação

#### 3.4.1 Definição

A erosão pode ser definida a partir da subtração vetorial dos elementos da imagem X com o elemento estruturante B [Haralick et al. (1987)], isto é :

$$X \ominus B = E^B (X) = \left\{ d \in Z^2 : d + b \in X \text{ para todo } b \in B \right\} \quad \text{Eq. 3.2}$$

A dilatação pode ser definida a partir da adição vetorial dos elementos dos conjuntos X e B :

$$X \oplus B = D^B (X) = \left\{ d \in Z^2 : d = x + b \text{ para todo } x \in X \text{ e } b \in B \right\} \quad \text{Eq. 3.3}$$

Para uma malha quadrada (caso mais comum de representação de uma imagem digitalizada), o elemento estruturante elementar pode ser 4- ou 8-conectado, como mostra a Figura 3.3.

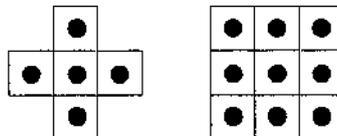


Figura 3.3. Elemento estruturante 4- e 8-conectado.

A erosão e a dilatação podem ser vistas como casos particulares da HMT. No caso da erosão, respostas positivas referentes à inclusão do elemento estruturante  $B$  (monofásico) em  $X$  formam o novo conjunto erodido de  $X$ . O conjunto dilatado de  $X$  corresponde a respostas positivas à interseção de  $X$  com  $B$  (Figura 3.4). Assim, a erosão e a dilatação podem ser definidas ainda por:

$$E^B(X) = \left\{ x \in Z^2 : B_x \subset X \right\} = \bigcap_{b \in B} X_b, \quad e \quad \text{Eq. 3.4}$$

$$D^{\overset{v}{B}}(X) = \left\{ x \in Z^2 : X \cap \overset{v}{B}_x \neq \emptyset \right\} = \bigcup_{b \in \overset{v}{B}} X_b \quad \text{Eq. 3.5}$$

onde  $\overset{v}{B}$  corresponde ao transposto de  $B$ , dado por :

$$\overset{v}{B} = \{ -b : b \in B \} \quad \text{Eq. 3.6}$$

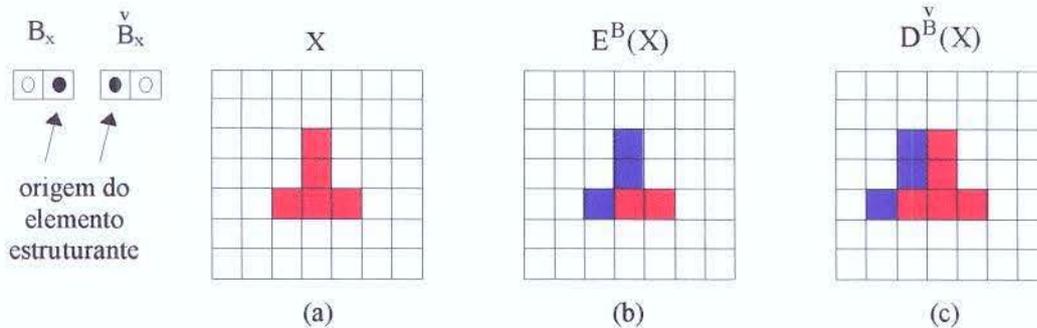


Figura 3.4 : Exemplo de erosão e dilatação de um conjunto  $X$ , com um elemento estruturante  $B$ ; em (b), pontos azuis representam a parte de  $X$  eliminada pela erosão; em (c), pontos azuis são adicionados a  $X$  pela dilatação.

### 3.4.2 Extensões : Imagens em Níveis de Cinza

A definição das operações acima pode ser estendida ao caso numérico, ou seja, às imagens em níveis de cinza (veja, por exemplo, [Haralick et al. (1987)]). Sejam  $f : F \rightarrow Z$  e  $g : G \rightarrow Z$ , definidas em  $Z^2$ , duas funções (superfícies) representando imagens em níveis

de cinza. Assim, a definição da erosão,  $f \ominus g$ , e da dilatação,  $f \oplus g$ , de  $f$  por  $g$ , é :

$$(f \ominus g)(x) = \text{MIN}_{z \in G, x+z \in F} \{f(x+z) - g(z)\}, \text{ e} \tag{Eq. 3.7}$$

$$(f \oplus g)(x) = \text{MAX}_{z \in G, x-z \in F} \{f(x-z) + g(z)\} \tag{Eq. 3.8}$$

Para um elemento estruturante planar ( $g(z) = 0$ ), a erosão e a dilatação de  $f$  por  $g$  pode ser representada simplesmente por (Figura 3.5) :

$$(f \ominus g)(x) = \text{MIN}_{z \in G, x+z \in F} \{f(x+z)\}, \text{ e} \tag{Eq. 3.9}$$

$$(f \oplus g)(x) = \text{MAX}_{z \in G, x-z \in F} \{f(x-z)\} \tag{Eq. 3.10}$$

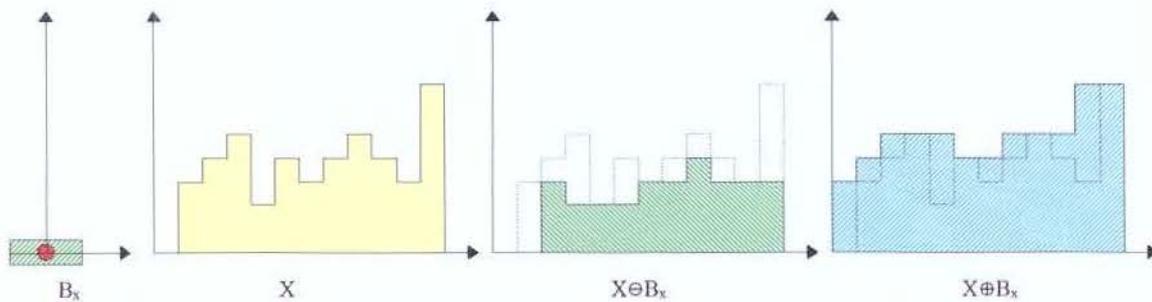


Figura 3.5 : Exemplo de erosão e dilatação numérica de  $X$  por  $B_x$  (planar).

Algumas propriedades da erosão e dilatação são :

- A erosão e a dilatação são invariantes a translação, isto é,  $X_H \ominus B = (X \ominus B)_H$  e  $X_H \oplus B = (X \oplus B)_H$ .
- A erosão e dilatação são operações duais, em relação a complementação, isto é,  $X \oplus \overset{V}{B} = (X^c \ominus B)^c$
- A erosão e a dilatação são crescentes, isto é, se  $X \subset Y$  então  $E^B(X) \subset E^B(Y)$  e  $D^B(X) \subset D^B(Y)$ .

- A erosão e a dilatação não são idempotentes, isto é,  $E^B(E^B(X)) \neq E^B(X)$ , e  $D^B(D^B(X)) \neq D^B(X)$ .
- A erosão e a dilatação não são homotópicas, isto é, não preservam o número de componentes conexas da imagem.

## 3.5 Abertura e Fechamento

### 3.5.1 Definição

As operações morfológicas mais complexas são definidas a partir de combinações das operações de erosão e dilatação. Exemplos bem conhecidos destas combinações são a abertura e o fechamento.

A abertura é uma operação morfológica que tem a propriedade de eliminar partes dos objetos cuja dimensão é inferior à dimensão do elemento estruturante (Figura 3.6). Simbolicamente:

$$O^B(X) = (X \ominus B) \oplus \overset{\vee}{B} = D^{\overset{\vee}{B}}(E^B(X)) \quad \text{Eq. 3.11}$$

A operação dual da abertura é o fechamento que tem a propriedade de unir objetos cuja distância é inferior à dimensão do elemento estruturante (Figura 3.6). Simbolicamente :

$$F^B(X) = (X \oplus \overset{\vee}{B}) \ominus B = E^B(D^{\overset{\vee}{B}}(X)) \quad \text{Eq. 3.12}$$

A Figura 3.6 ilustra as operações de abertura e fechamento.

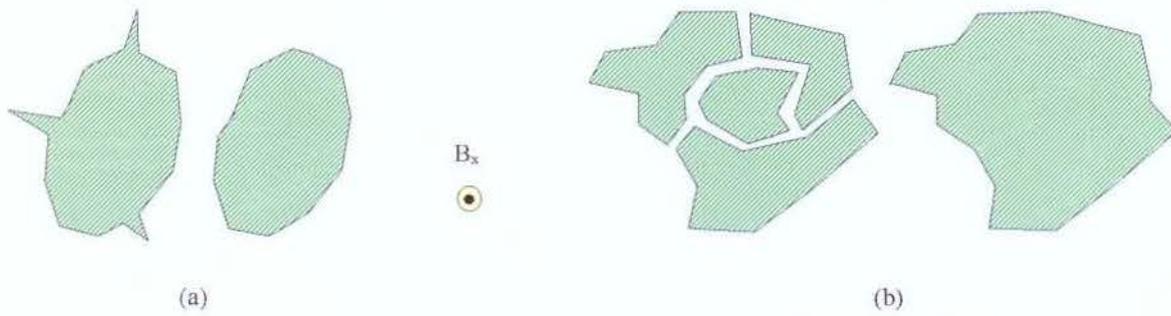


Figura 3.6 : Exemplo de abertura (a) e fechamento (b) com o elemento estruturante  $B_x$ .

Para o caso numérico, as operações de abertura e fechamento podem dadas por:

$$O^g f(x) = \text{MAX} \left\{ \text{MIN} \left\{ f(x+z) - g(z) \right\} \right\}, \text{ e} \quad \text{Eq. 3.13}$$

$z \in G, \quad x+z \in F$

$$F^g f(x) = \text{MIN} \left\{ \text{MAX} \left\{ f(x-z) + g(z) \right\} \right\} \quad \text{Eq. 3.14}$$

$z \in G, \quad x+z \in F$

A Figura 3.7 ilustra estas operações considerando um elemento estruturante planar.

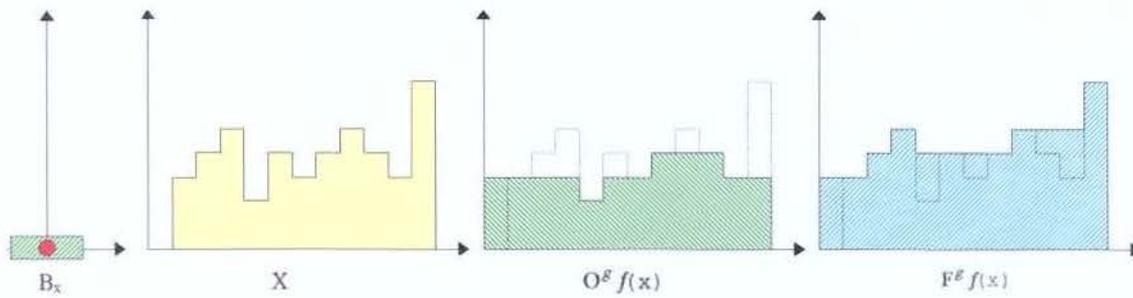


Figura 3.7 : Exemplo de abertura e fechamento numérico, de  $X$  por  $B_x$  planar.

## 3.6 Algumas Transformações Morfológicas

Ilustraremos, a seguir, algumas transformações morfológicas que serão consideradas, posteriormente, na definição do algoritmo de photomosaicking.

### 3.6.1 Afinamento e Espessamento

#### 3.6.1.1 Definição

O afinamento e o espessamento são operações homotópicas definidas a partir de uma transformação HMT geral (Eq. 3.1). A operação de afinamento representa uma aproximação para o caso discreto, do esqueleto definido no espaço Euclidiano. Esta operação consiste da supressão de pontos da imagem  $X$  de acordo com um elemento estruturante  $B$  indicando configurações homotópicas na vizinhança de um ponto (Figura 3.9 a, b). Simbolicamente :

$$X \ominus B = X \setminus (X \otimes B), \quad \text{Eq. 3.15}$$

onde  $B$  representa uma família de configurações homotópicas representando pontos da imagem que podem ser eliminados sem que a sua conexidade seja alterada. Na prática, diferentes famílias de elementos estruturantes homotópicos podem ser definidos. Um exemplo é apresentado na Figura 3.8. Observe que a mudança de estado do ponto central, nestas configurações, não altera localmente o número de componentes conexas da vizinhança.

$$\begin{matrix}
 L_4^0 = \begin{bmatrix} 0 & 0 & 0 \\ * & 1 & * \\ 1 & 1 & 1 \end{bmatrix} & L_4^1 = \begin{bmatrix} * & 0 & * \\ 1 & 1 & 0 \\ 1 & 1 & * \end{bmatrix} & L_8^0 = \begin{bmatrix} 0 & 0 & 0 \\ * & 1 & * \\ 1 & 1 & 1 \end{bmatrix} & L_8^1 = \begin{bmatrix} * & 0 & * \\ 1 & 1 & * \\ * & 1 & * \end{bmatrix} \\
 M_4^0 = \begin{bmatrix} * & 0 & * \\ * & 1 & * \\ 1 & 1 & 1 \end{bmatrix} & M_4^1 = \begin{bmatrix} * & 0 & * \\ 1 & 1 & * \\ 1 & 1 & * \end{bmatrix} & M_8^0 = \begin{bmatrix} * & 0 & * \\ 1 & 1 & 1 \\ * & 1 & * \end{bmatrix} & M_8^1 = \begin{bmatrix} * & 0 & 0 \\ 1 & 1 & * \\ * & 1 & * \end{bmatrix}
 \end{matrix}$$

Figura 3.8 : Famílias homotópicas M e L, em malha 4- e 8-conexas.

A estrutura básica deste esqueleto homotópico contém um pixel de espessura (Figura 3.9 b), onde cada pixel possui dois vizinhos, exceto nos casos de pontos extremos e pontos de interconexão.

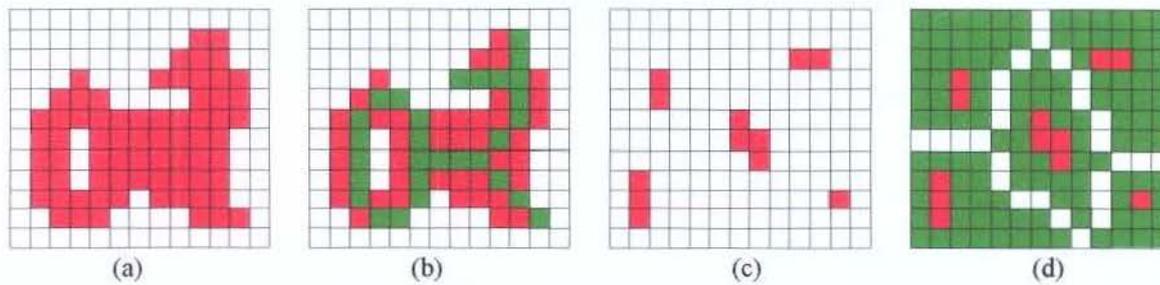


Figura 3.9 : Exemplo de afinamento e espessamento. Pontos verdes representam, respectivamente, o afinamento (b) e o espessamento (d) dois objetos originais (a) e (c).

O espessamento consiste da adição de pontos à imagem X de acordo com a configuração de um dado elemento estruturante B (Figura 3.9 c, d). Simbolicamente :

$$X \odot B = X \cup (X \otimes B) \tag{Eq. 3.16}$$

O afinamento de um conjunto X é dado pela aplicação, em paralelo, até a idempotência da seguinte seqüência de transformações :

$$(X \circ B)_\infty = (\dots (((X \circ B^0) \circ B^1) \circ B^2) \dots \circ B^i), \tag{Eq. 3.17}$$

onde  $B^i$  representa as diferentes configurações da família homotópica B (Figura 3.8, por

exemplo).

De maneira análoga, o espessamento de um conjunto  $X$  é dado pela aplicação, até a idempotência da seguinte seqüência de transformações :

$$(X \odot B^c)_\infty = (\dots (((X \odot B^{0^c}) \odot B^{1^c}) \odot B^{2^c}) \dots \odot B^{1^c}) \quad \text{Eq. 3.18}$$

### 3.6.1.2 Extensões ao Caso Numérico

A definição das operações acima pode ser estendida às imagens em níveis de cinza [Beucher (1983)]. Seja  $f : F \rightarrow Z$ , uma função representando uma imagem em nível de cinza, e  $B$  um elemento estruturante planar. A definição de afinamento,  $(f \circ B)(x)$ , e de espessamento,  $(f \odot B)(x)$ , de  $f$  por  $B$ , é:

$$(f \circ B)(x) = \begin{cases} \text{Se } \text{Max}\{f(x_{B_0})\} < f(x) \leq \text{Min}\{f(x_{B_1})\}, & \text{Max}\{f(x_{B_0})\} \\ f(x) & \end{cases}, \text{ e Eq. 3.19}$$

$$(f \odot B)(x) = \begin{cases} \text{Se } \text{Max}\{f(x_{B_0})\} < f(x) \leq \text{Min}\{f(x_{B_1})\}, & \text{Min}\{f(x_{B_1})\} \\ f(x) & \end{cases} \quad \text{Eq. 3.20}$$

onde  $x_{B_0}$  e  $x_{B_1}$  são as fases “0” e “1”, respectivamente, do elemento estruturante utilizado (Figura 3.8, por exemplo).

## 3.6.2 Espessamento Homotópico e Esqueleto por Zona de Influência

Seja um conjunto  $X$ , definido em  $R^2$ , constituído de partículas individuais. A cada partícula  $X_i$  podemos associar uma zona de influência  $Y_i$ , tal que todos os pontos  $y$  de  $Y_i$  estejam mais próximos de  $X_i$  do que de  $X_j$ ,  $i \neq j$ , isto é

$$Y_i = \bigcup \left[ y : D(y, X_i) < D(y, X_j), \forall i \neq j \right] \quad \text{Eq. 3.21}$$

O esqueleto por zona de influência (SKIZ), representado pelo conjunto dos pontos que não pertencem a nenhuma zona de influência, divide o espaço no número de

partículas de  $X$ , e é um subconjunto do esqueleto de  $X^c$ . O esqueleto de  $X^c$  pode ser obtido através de um espessamento homotópico de  $X$ , por exemplo :

$$S_q(X^c) = ((X \odot L^c)_\infty)^c \tag{Eq. 3.22}$$

Este esqueleto (Eq. 3.22) difere da definição de SKIZ pela presença de ramos parasitas contendo pontos extremos (linhas tracejadas na Figura 3.10). Estes ramos podem ser suprimidos através da família  $E$  de elementos estruturantes (Figura 3.11). A família  $E$  é o conjunto das 4 rotações de  $90^\circ$  de  $E_4$  ou de  $E_8^0$  seguido de  $E_8^1$ , dependendo da conectividade adotada. Assim, o SKIZ pode ser definido como :

$$SKIZ(X) = (((X \odot L^c)_\infty \odot E^c)_\infty)^c \tag{Eq. 3.23}$$

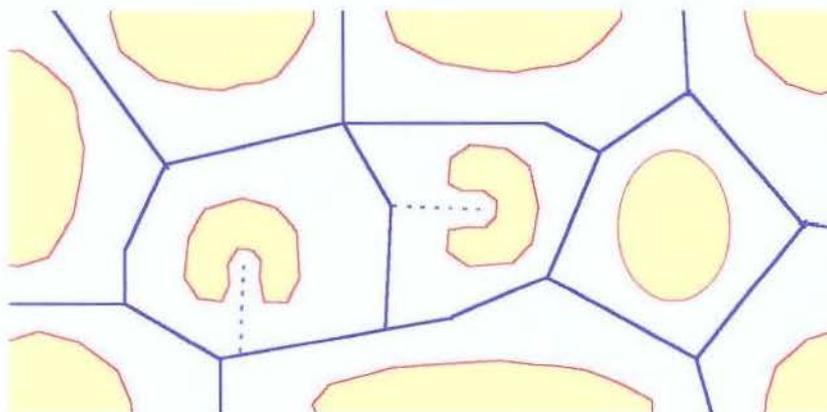


Figura 3.10 : Exemplo de esqueleto por zona de influência.

$$E_4 = \begin{bmatrix} * & 0 & * \\ 0 & 1 & 0 \\ * & * & * \end{bmatrix} \quad E_8^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ * & 1 & * \end{bmatrix} \quad E_8^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & * & * \end{bmatrix}$$

Figura 3.11 : Famílias  $E$ , em malha 4- e 8-conexas.

## 3.7 Transformações Geodésicas

Como já mencionamos, a finalidade da análise quantitativa de imagens é prover medições (métricas ou topológicas) de imagens [Lantuejoul-Maisonneuve (1984)]. Desde que, na prática, estas imagens não são passíveis de uma medição direta, a análise quantitativa de imagens geralmente é realizada através de uma transformação (no sentido de manter somente as informações necessárias à medição subsequente), e de uma quantificação dessa transformação. A imagem de estudo é composta, geralmente, de muitas partículas, das quais desejamos extrair informações individualmente. O objetivo aqui é poder analisar cada uma destas partículas individualmente. Uma maneira de se restringir o domínio das operações morfológicas de base é através do conceito de transformações geodésicas [Lantuejoul-Maisonneuve (1984)].

### 3.7.1 Distâncias Geodésicas

Seja a imagem apresentada na Figura 3.12 e os pontos  $x_1$ ,  $x_2$  e  $x_3$  das suas componentes. Definimos a distância geodésica  $D_X(x_1, x_2)$  entre  $x_1$  e  $x_2$  como o arco incluído em  $X$ , e que corresponde ao menor percurso entre  $x_1$  e  $x_2$ . Caso este arco não exista (a distância geodésica entre  $x_1$  e  $x_3$ , por exemplo), esta distância é considerada infinita.

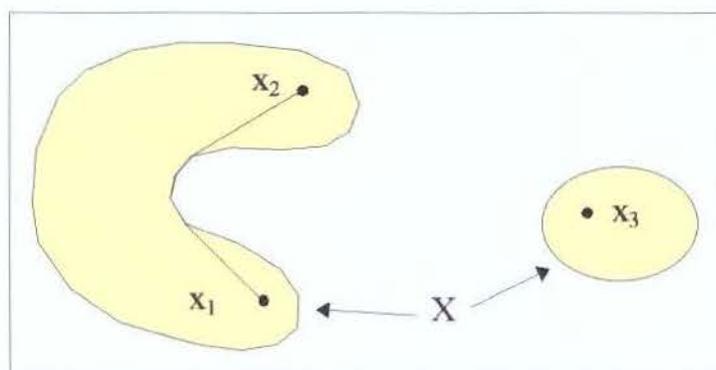


Figura 3.12 : Definição de distância geodésica.

Seja o conjunto  $B_X(x, R) = \{ y \in X : D_X(x, y) \leq R \}$  de todos os pontos de  $X$

cuja distância geodésica até  $x$  é menor ou igual a  $R$ . Este conjunto é denominado de círculo geodésico de raio  $R$  e centro  $x$  (Figura 3.13). Observe a diferença entre o círculo geodésico, centrado em  $x$ , e o equivalente Euclidiano.

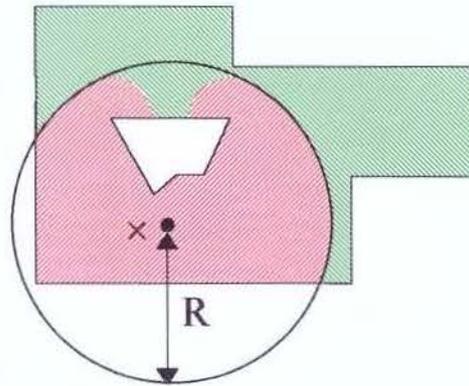


Figura 3.13 : Exemplo de círculo geodésico; a porção vermelha representa o círculo geodésico de raio  $R$ , para o ponto  $x$ ; o círculo preto representa a versão Euclidiana.

Usando esta métrica, a distância geodésica entre um ponto  $x$  ( $x \in X$ ) e um subconjunto  $Y$  ( $Y \subset X$ ),  $D_X(x, Y)$ , é definida como a menor distância geodésica entre  $x$  e qualquer ponto  $y$  de  $Y$ :

$$D_X(x, Y) = \min_{y \in Y} D_X(x, y) \quad \text{Eq. 3.24}$$

Assim, podemos definir todas as transformações morfológicas clássicas (erosão, dilatação, esqueletização, etc) no espaço métrico geodésico. A Figura 3.14 ilustra a dilatação e a erosão geodésicas. Por exemplo, se  $Y \subset X$ , os pontos  $x$  de  $X$ , tal que  $B_X(x, R)$  intercepta  $Y$ , constitui o conjunto  $R$ -dilatado de  $Y$  em  $X$ , denotado por:

$$D_X^R(Y) = \{ x \in X : B_X(x, R) \cap Y \neq \emptyset \} \quad \text{Eq. 3.25}$$

$Y$  é, geralmente, denominado de *marcador* da função  $X$ .

Na prática, a dilatação geodésica de tamanho  $N$ , em  $Z^2$ , é feita reiterando-se  $N$  vezes uma dilatação geodésica de dimensão 1 (com um elemento estruturante elementar  $B$ ), restrita ao espaço  $X$ , isto é :

$$D_x^1(Y) = (Y \oplus B) \cap X, \tag{Eq. 3.26}$$

e assim

$$D_x^N(Y) = \underbrace{D_x^1(D_x^1(\dots D_x^1(Y)\dots))}_{N \text{ vezes}} \tag{Eq. 3.27}$$

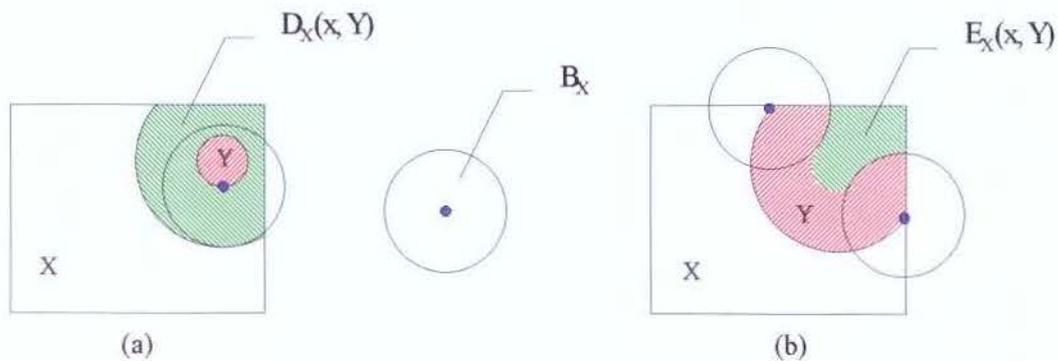


Figura 3.14 : Exemplo de dilatação (a) e erosão (b) geodésicas de Y em X.

De forma semelhante, podemos definir o conjunto R-erodido de Y em X como os pontos x de X de modo que  $E_x^R(Y)$  esteja totalmente incluído em Y :

$$E_x^R(Y) = \{ x \in X : B_x(x, R) \subseteq Y \} \tag{Eq. 3.28}$$

Da mesma forma, o esqueleto geodésico por zona de influência, segundo X,  $SKIZ_x(Y)$ , é definido como (Figura 3.15):

$$SKIZ_x(Y) = [((((Y \odot L^c) \cap X)_\infty \odot E^c) \cap X)_\infty]^c \tag{Eq. 3.29}$$

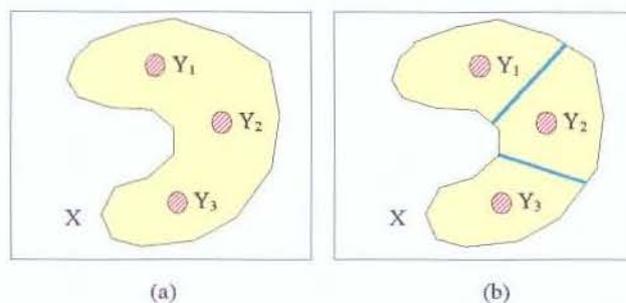


Figura 3.15 : Exemplo de SKIZ geodésico; em (a),  $Y_1$ ,  $Y_2$ , e  $Y_3$  são componentes de  $X$ ; em (b), o SKIZ geodésico de  $X$  é mostrado em azul.

## 3.8 Linha de Divisor de Águas

### 3.8.1 Definição

O conceito de linha de divisor de águas (LDA) de uma função é muito útil em segmentação de imagens [Beucher (1982)], [Beucher (1983)]. Seja  $f : (x, y) \in Z^2 \mapsto Z$  uma imagem em níveis de cinza. A representação de  $f$  pode ser vista como uma superfície topográfica, na qual certas informações, tais como o mínimo de uma função e o seu divisor de águas, são de particular interesse.

Uma abordagem intuitiva para o divisor de águas dessa função pode ser a seguinte [Beucher (1982)]. Imagine a nossa superfície topográfica como sendo porosa, e, suponha que a imergimos, pouco a pouco, em uma piscina de tal modo que o nível da água continue o mesmo em toda sua extensão. Sempre que a água proveniente de dois vales da superfície se encontram, um dique é construído, evitando a união dessas águas. Ao final, o conjunto dos diques, representando as fronteiras dos mínimos da superfície, constitui a LDA da mesma (Figura 3.16).

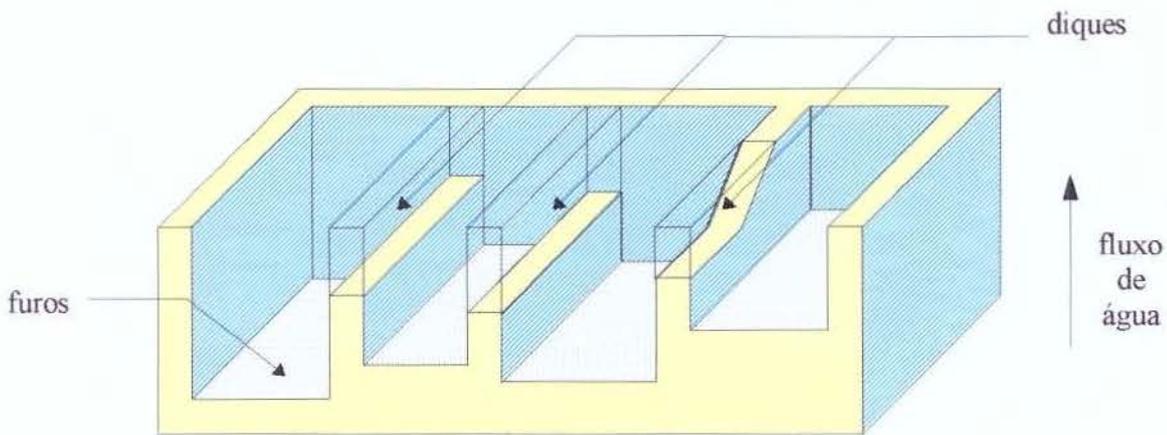


Figura 3.16 : Definição intuitiva de LDA.

### 3.8.2 Implementação

Morfologicamente, a LDA pode ser definida da seguinte forma [Beucher (1982)].

Seja  $\mathfrak{T}$  o espaço da nossa imagem  $f$ , e

$$X_i = \{ x \in \mathfrak{T}, f(x) \leq i \} \tag{Eq. 3.30}$$

$X_i$  é o conjunto de todos os pontos do espaço tal que  $f(x)$  é menor ou igual a  $i$ . Seja  $Z$  o conjunto de todas as LDA da imagem  $f$ , e  $Z_j$  o subconjunto de  $Z$  cujos pontos se encontram a uma altura  $j$ . Supondo que  $Z_k$  é conhecido, para  $k \leq j - 1$ , e sendo  $X_{j-1}$  o limiar da função  $f$  no nível  $j - 1$ , então  $X_{j-1} - Z_{j-1}$  representa o conjunto dos pontos cuja altura é menor do que  $j$ , e que pertencem somente a um “vale” da superfície (Figura 3.17). Neste caso, podemos ver que os pontos de  $X_j$  igualmente distantes de duas componentes conectadas pertencem aos divisores de água no nível  $j$ , ou seja :

$$Z_j = SKIZ_{X_j} ( X_{j-1} - Z_{j-1} ) \tag{Eq. 3.31}$$

e

$$Z = \bigcup_j Z_j \tag{Eq. 3.32}$$

Esta definição provê um método iterativo de construção da LDA de uma função

(Figura 3.17).

Freqüentemente, a computação de LDA é complexa, e o processo de reconstrução, nível à nível, utilizando limiares de uma função, é bastante demorado. Felizmente, a definição dessas LDA está relacionada com a operação de afinamento numérico, vista anteriormente (Eq. 3.19).

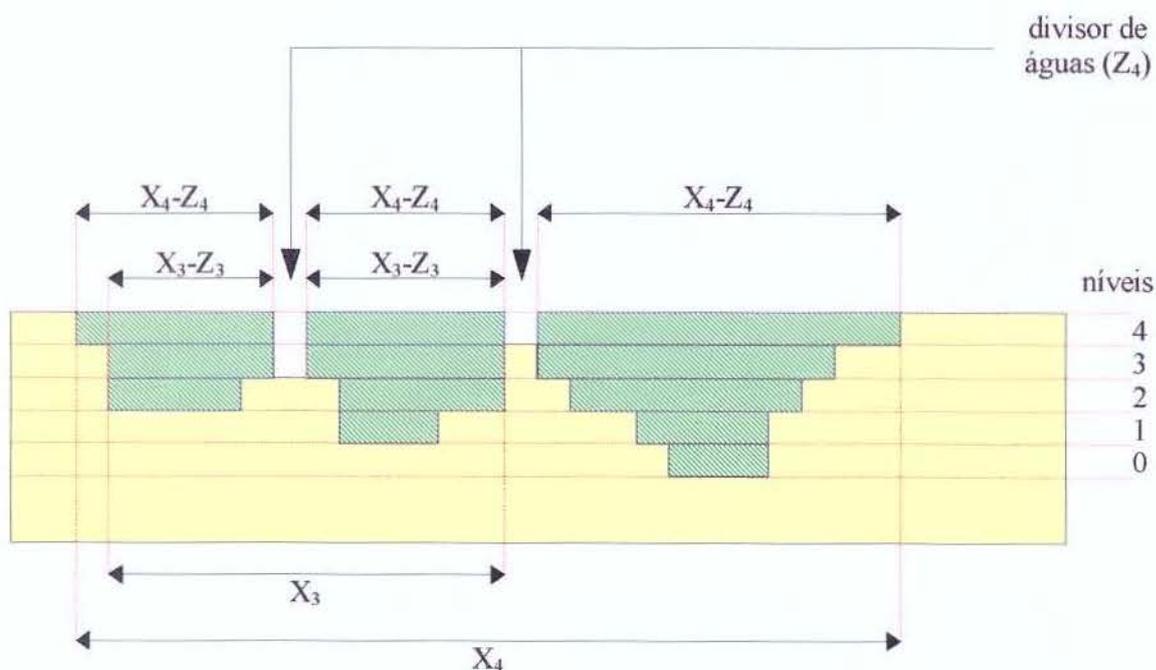


Figura 3.17 : Processo iterativo de definição da LDA.

De fato, executando-se um afinamento com as famílias homotópicas da Figura 3.8, por exemplo, pode-se mostrar que as LDA de uma função não mais são do que os arcos fechados resultantes desse afinamento [Beucher (1983)]. A Figura 3.18 dá um exemplo de uma imagem original e a imagem de LDA associada.

Os diferentes níveis de implementação do algoritmo de LDA, através deste método, podem ser vistos no Apêndice B.

251	252	252	247	247	252	250	249	254	252
250	251	247	248	248	252	251	253	252	251
251	252	247	248	247	250	252	252	251	255
250	253	246	247	247	248	252	252	253	252
255	252	246	246	246	248	249	251	255	250
254	249	246	247	247	250	249	255	250	249
251	252	246	247	247	251	253	254	251	249
246	254	247	247	247	252	252	251	252	248
249	252	247	248	248	252	251	251	253	247
249	249	251	252	251	251	251	251	247	249

(a)

251	252	252	247	247	252	250	249	254	252
250	251	246	248	248	252	250	253	252	251
251	252	246	248	246	246	252	251	251	255
250	253	246	246	246	246	246	252	253	252
255	246	246	246	246	246	246	246	255	250
254	246	246	246	246	250	246	255	249	249
251	252	246	246	246	246	253	254	249	249
246	254	246	246	246	246	252	251	252	248
249	252	246	246	246	252	251	251	253	247
249	249	251	252	251	251	251	251	247	249

(b)

Figura 3.18 : Processo de obtenção da LDA (b) da imagem em (a) através de afinamento numérico. Os retângulos azuis são os pontos que não fazem parte da LDA (círculos vermelhos).

## 4. O Algoritmo

### 4.1 Introdução

De um modo geral, a linha de costura que procuramos definir deve ter as seguintes características :

1. Ela deve ser conexa, isto é, definir um caminho 4- ou 8-conexo.
2. Cada ponto da linha deve representar, individualmente, um ponto de maior homogeneidade, com relação aos seus vizinhos não pertencentes a linha.
3. A linha deve atravessar toda a região de interseção.
4. O conjunto dos pontos que formam a linha deve passar no eixo medial de regiões homogêneas, visando evitar regiões próximas a altos gradientes onde o olho humano é mais sensível.
5. Ela deve ser irregular (linha não monotônica), podendo definir, de maneira mais realista, qualquer caminho na região de interseção.

Os métodos mencionados anteriormente definem linhas monotônicas (um pixel por linha ou coluna) e são baseados em informações estritamente locais. Além disso, a restrição espacial, associada à busca dos pontos da costura, tem como consequência o surgimento de arestas artificiais que invalidam o fotomosaico final [Milgram (1977)].

O método morfológico que descrevemos, a seguir, permite-nos definir um fotomosaico obedecendo aos critérios acima, a partir de informações globais da imagem. Através destas informações podemos garantir, por exemplo, os critérios (1), (4) e (5) não considerados nos métodos anteriores.

## 4.2 Descrição do Algoritmo

O método descrito aqui extrai informações globais de uma imagem de correlação relativa à região de interseção do fotomosaico. É esta característica global que permite a definição de um fotomosaico baseado numa linha de costura irregular.

Inicialmente, nós consideramos que a imagem de correlação corresponde a uma operação pontual representada pela diferença absoluta dos pixels nas duas regiões de interseção (comentários sobre índices de correlação serão feitos posteriormente). Desta forma, nós definimos uma imagem cujos baixos níveis de cinza representam fortes candidatos à linha de costura.

Esta imagem de correlação pode ser vista como uma superfície topográfica contendo picos e vales, isto é, pontos cuja altura denota regiões de alta ou baixa correlação. Assim, nós podemos utilizar a noção de função LDA para segmentar a imagem de correlação, encontrando os pontos que pertencem aos vales (regiões de alta correlação) desta imagem. Aqui, o algoritmo de LDA deve ser aplicado ao negativo da imagem de correlação. Se  $m$  é o nível de cinza maximal desta imagem, o seu negativo ou complemento é dado por :

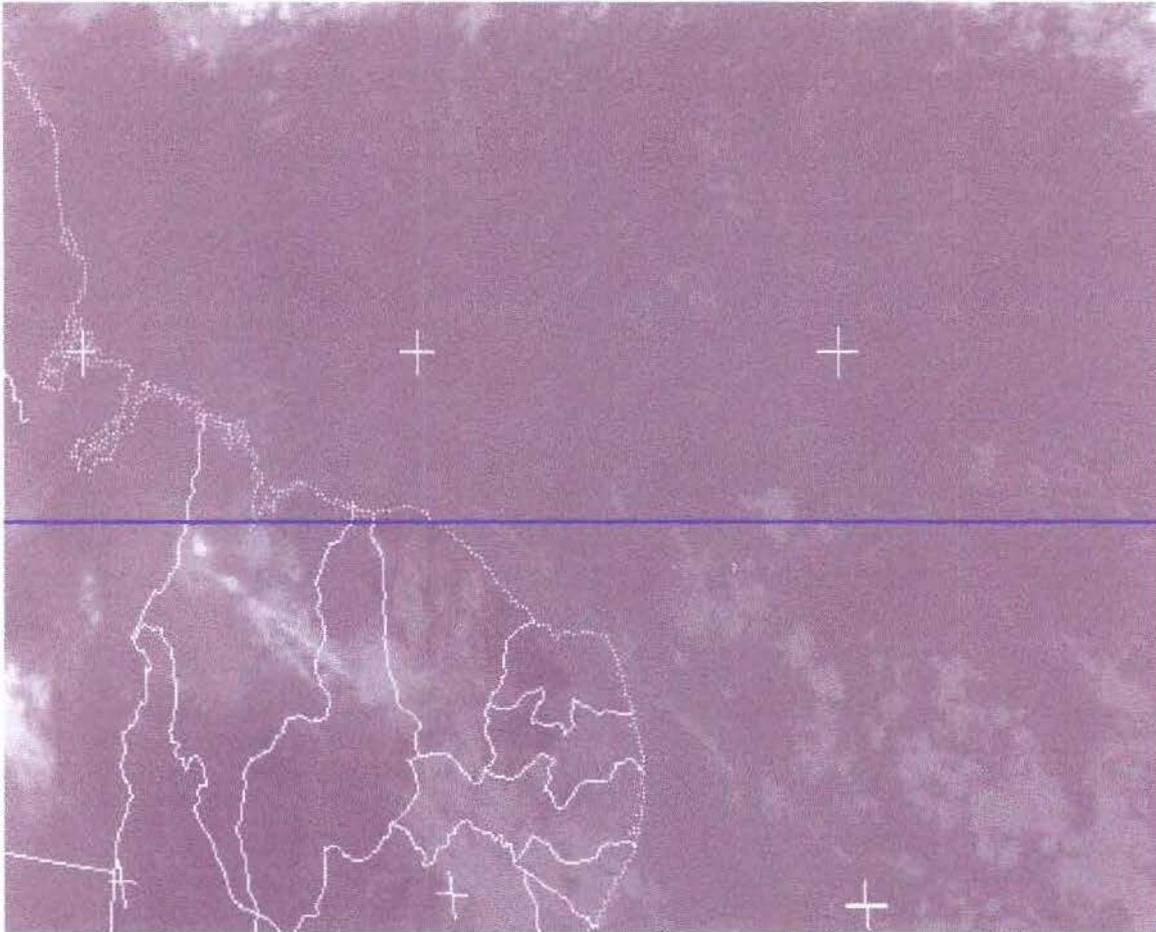
$$g' = m - g \quad \text{Eq. 4.1}$$

O algoritmo de photomosaicking consiste basicamente das duas etapas seguintes:

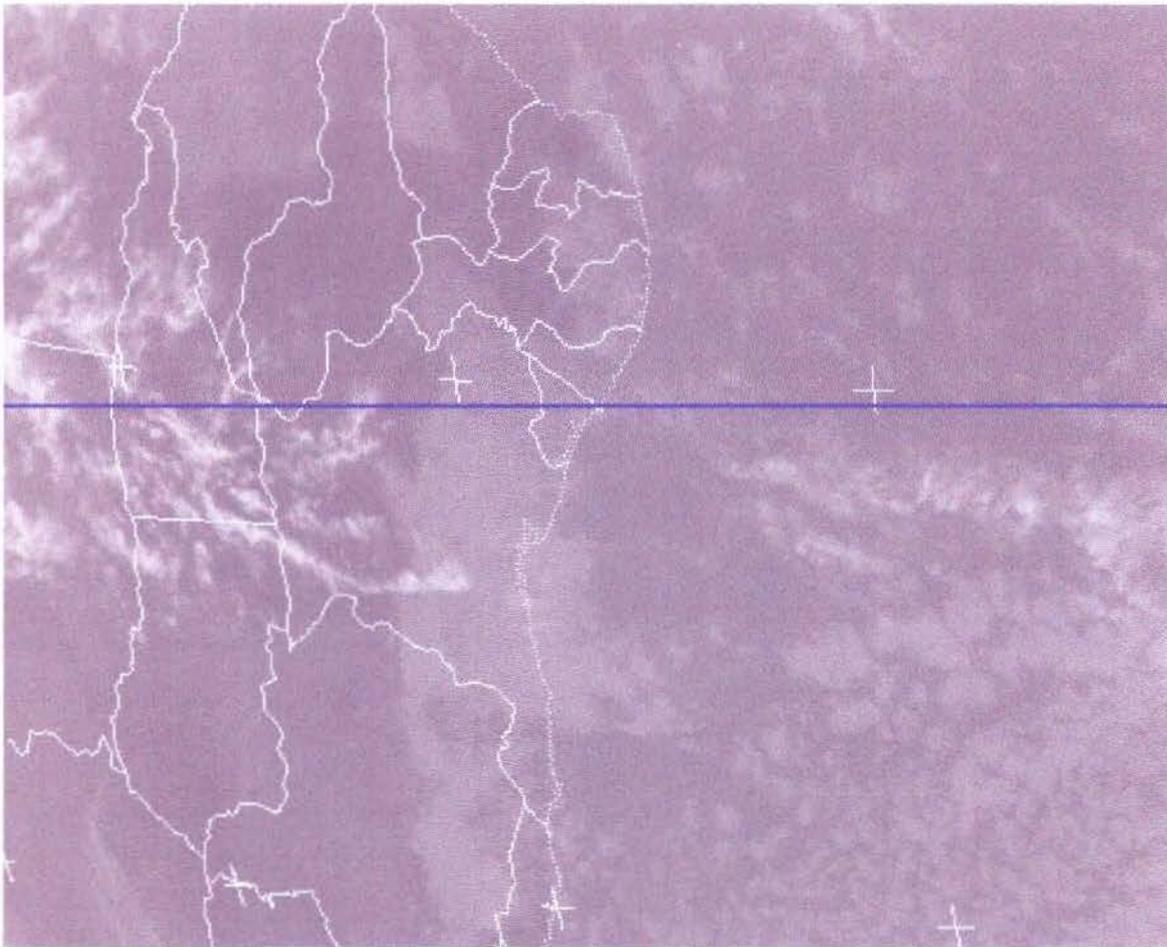
1. Definição das LDA da imagem de correlação.
2. Definição de uma imagem bifásica indicando os pontos relativos a cada lado do fotomosaico final.

As LDA podem ser definidas de acordo com o algoritmo apresentado no seção 3.8. A Figura 4.3 mostra um exemplo de uma imagem de LDA obtida a partir da imagem de

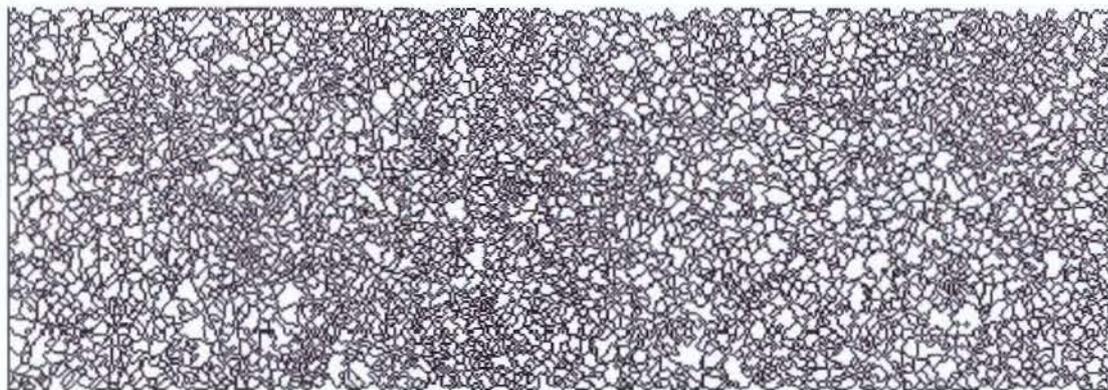
correlação entre as imagens da Figura 4.1 e Figura 4.2 (para um fotomosaico vertical).



*Figura 4.1 : Parte superior do fotomosaico.*



*Figura 4.2 : Parte inferior do fotomosaico.*



*Figura 4.3 : LDA obtida da correlação entre as imagens da Figura 4.1 e Figura 4.2.*

### 4.2.1 Definição da Linha de Costura Final

Como podemos ver na Figura 4.3, a segmentação por LDA produz uma imagem  $L$  supersegmentada, com muitos caminhos opcionais atravessando a região de interseção. O passo seguinte do algoritmo morfológico consiste da eliminação desses caminhos redundantes e da definição de uma imagem bifásica na qual cada fase está relacionada com um lado do fotomosaico.

Morfologicamente, esta imagem bifásica pode ser definida através das operações geodésicas discutidas na seção 3.7. A abordagem consiste em dilatar um marcador, a partir da borda esquerda (superior), da imagem segmentada, e outro marcador, a partir da sua borda direita (inferior), para um fotomosaico horizontal (vertical). Após a dilatação geodésica desses marcadores, as LDA que não se encontram entre duas fases diferentes podem ser eliminadas. A operação termina quando uma linha separando estas duas fases é encontrada.

Seja, para um fotomosaico vertical,  $\partial S$  e  $\partial I$  o conjunto dos marcadores relativos às bordas superiores e inferiores de uma imagem binária  $L$ . Esta imagem indica os pontos da imagem de correlação pertencentes à LDA. A este conjunto nós associamos etiquetas  $S$  e  $I$ , respectivamente.

A dilatação geodésica (Eq. 3.27) dos marcadores  $\partial S$  e  $\partial I$ , com relação à imagem  $L$ , pode ser dada por:

$$G = D_{L^c}^{\infty}(\partial S) \cup D_{L^c}^{\infty}(\partial I) \quad \text{Eq. 4.2}$$

Informalmente,  $D^{\infty}(\dots)$  significa execução da dilatação geodésica até estabilização (raio geodésico infinito). A imagem bifásica  $p$  contendo as etiquetas  $S$  e  $I$  é tal que :

$$p(x,y) = \begin{cases} S, & \text{se } [(x,y) \in G] \subset D_{L^c}^\infty(\partial S) \\ I, & \text{se } [(x,y) \in G] \subset D_{L^c}^\infty(\partial I) \\ 0, & \text{caso contrário} \end{cases} \quad \text{Eq. 4.3}$$

isto é, os pixels da imagem  $p$ , na posição  $(x, y)$ , indicam os pontos da imagem bifásica tendo valor  $S$  (se o ponto correspondente em  $G$  faz parte do conjunto dilatado  $\partial S$ ) ou  $D$  (se o ponto correspondente em  $G$  faz parte do conjunto dilatado  $\partial I$ ). A Figura 4.4 ilustra a dilatação geodésica do conjunto  $\partial S$  em  $L^c$ . Neste caso, consideramos que os pontos da *forma* (pontos pretos) têm valor 1 e são 8-conectados, e os pontos do fundo (pontos brancos) têm valor 0 e são 4-conectados. O elemento estruturante  $B$  usado na dilatação geodésica é 4-conectado (Figura 3.3).

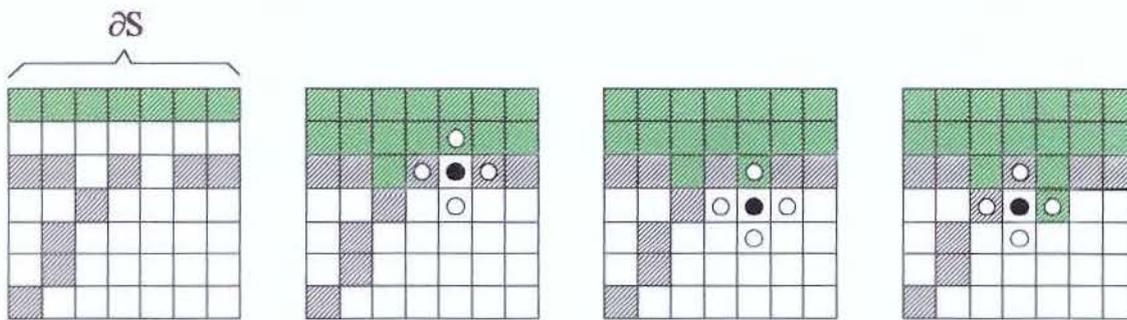


Figura 4.4 : Dilatação geodésica do conjunto  $\partial S$  por um elemento estruturante 4-conectado.

Depois deste passo, os pontos  $(G \oplus B) \setminus G$  da imagem  $L$  que não se encontram entre duas regiões etiquetadas com  $S$  e  $I$ , são eliminados, e, assim, o processo de dilatação, denotado pelas Equações 4.2 e 4.3, pode continuar. Estas operações se repetem até que a imagem  $p$  contenha apenas duas fases,  $S$  e  $I$ . A linha separando estas duas fases constitui a costura do fotomosaico.

Informalmente, a definição o processo de dilatação consiste simplesmente de uma varredura na imagem segmentada (imagem com as LDA) efetuando a propagação das

etiquetas S e I entre os pontos que não pertencem as linhas de costura <sup>1</sup>.

## 4.2.2 Eliminação de Caminhos Opcionais

Para a eliminação de caminhos opcionais da imagem supersegmentada, L, nós podemos considerar a altura das suas LDA <sup>2</sup>. Uma maneira simples de abordar esta etapa é eliminar, iterativamente, LDA com pontos elevados, correspondendo a baixas correlações.

A extensão do método morfológico para construção de fotomosaicos pode ser descrita da seguinte forma (para um mosaico vertical):

1. Definir uma imagem binária L representando os pontos de LDA da imagem de correlação (do negativo desta).
2. Propagar, em paralelo, as etiquetas S (a partir da borda superior) e I (a partir da borda inferior), até que uma LDA seja encontrada.
3. Repetir até que somente uma LDA separando as fases marcadas pelas etiquetas S e I seja encontrada:
  - Identificar a fase adjacente a pontos de LDA de menor correlação.
  - Eliminar aqueles pontos de menor correlação da LDA adjacente à respectiva fase.
  - Continuar propagação de etiquetas.

No caso específico em que as correlações mínimas forem iguais, nas duas fases,

---

<sup>1</sup> Numa abordagem seqüencial, este passo de propagação pode ser acelerado por algoritmos de preenchimento 4-conectado (veja, por exemplo, [Foley et al (1983)]).

<sup>2</sup> Observe que a modificação homotópica empregada em alguns casos para diminuir o efeito da supersegmentação da imagem de LDA não pode ser utilizada convenientemente aqui. Este método, associado geralmente à detecção de contornos de imagens numéricas, substitui os mínimos de uma função  $f$  por marcadores previamente definidos (no caso da detecção de contornos, estes marcadores são associados ao núcleo das diferentes componentes da imagem). Em seguida, uma operação de reconstrução define uma função  $f'$  cujo cálculo das LDA resulta numa imagem menos segmentada [Dougherty (1993)]

um critério a ser considerado é eliminar aquela LDA com maior número de pontos de correlação minimal (uma LDA será eliminada arbitrariamente caso o número desses pontos seja igual, nas duas fases).

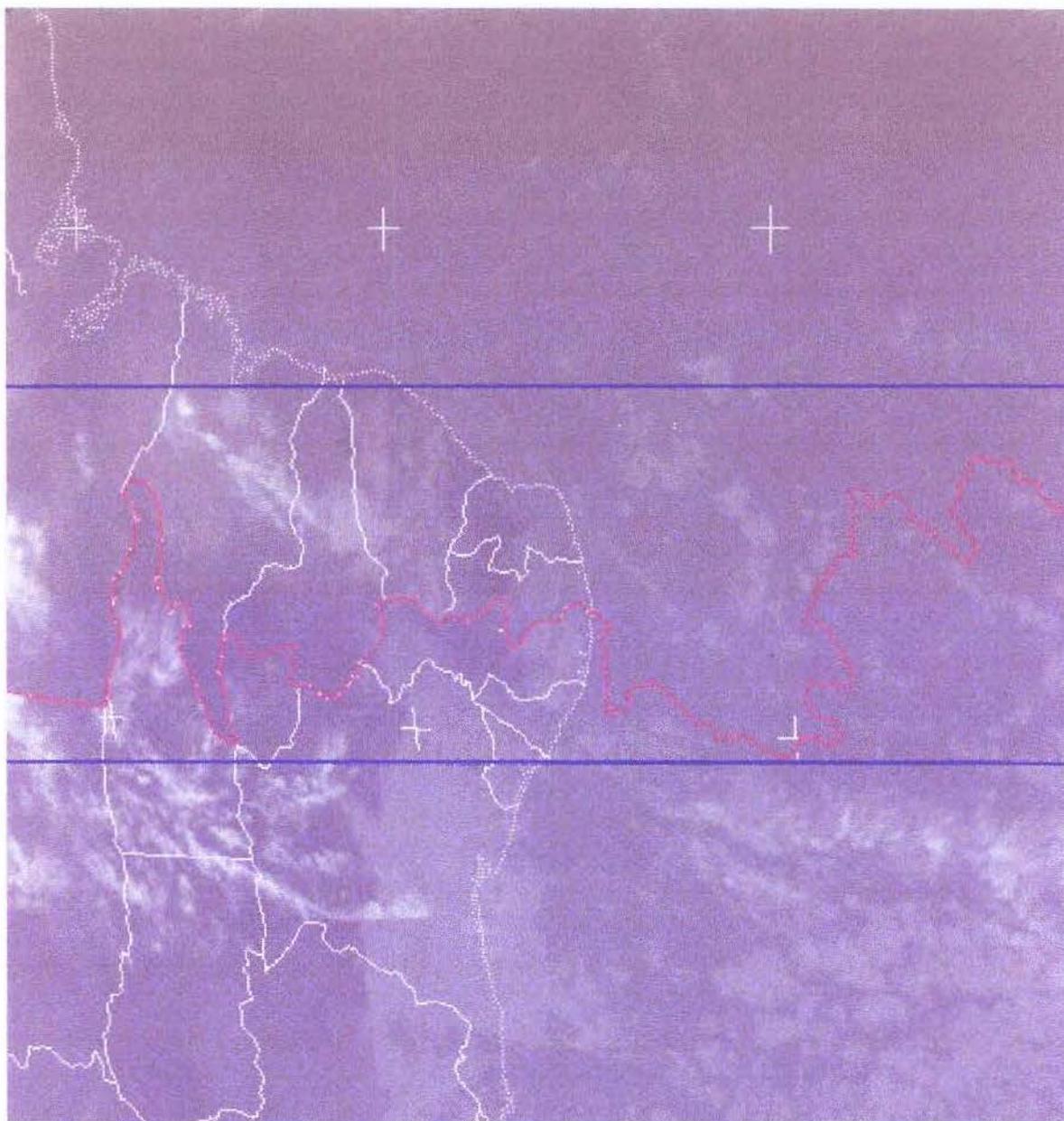
### 4.2.3 Composição do Fotomosaico

Como vimos, a imagem resultante deste processo consiste de duas fases, correspondendo às etiquetas S e I, separadas por uma linha de costura (Figura 4.5). A composição do fotomosaico é feita, facilmente, comparando-se o valor destas etiquetas. Assim, os pontos do fotomosaico, representando o conjunto dilatado superior ( $\partial S$ ) irão definir a parte superior do fotomosaico. Da mesma forma, os pontos representando o conjunto dilatado inferior ( $\partial I$ ) irão definir a parte inferior do fotomosaico. Os pontos da linha de costura podem ser preenchidos arbitrariamente com os pontos de uma das duas imagens.



Figura 4.5 : Resultado do cálculo da linha ótima.

A Figura 4.6 mostra um exemplo de fotomosaico morfológico, a partir das imagens da Figura 4.1 e Figura 4.2 (a linha azul delimita a região de interseção). Observe que a linha de costura encontrada passa sobre a grade de divisão política. Este resultado representa, de fato, um comportamento natural da linha, já que a mesma está presente nas duas imagens, representando, portanto, pontos de altíssima correlação.



*Figura 4.6 : Fotomosaico morfológico composto das imagens da Figura 4.1 e Figura 4.2.*

## 4.3 Comentários Finais

### 4.3.1 Índice de Correlação

Como vimos, o índice de correlação é o valor associado aos pixels da região de interseção, definindo uma nova imagem que orienta na obtenção da linha de costura. No nosso caso, a correlação entre as imagens deve medir, pontualmente, a semelhança entre seus níveis de cinza, de tal modo que a costura passe por pontos de maior correlação, gerando um mínimo de costuras visíveis.

O índice de correlação, IC, adotado anteriormente (diferenças absolutas pontuais) é um caso específico do índice de similaridade utilizado na identificação de padrões coincidentes, sendo bastante sensível a diferenças aditivas de amplitude [Gonzalez-Woods (1992)]. Esta última característica é importante, aqui, já que estamos, naturalmente, preocupados com as variações dos níveis de cinza dos pixels na região de interseção. A forma geral deste índice é dada por :

$$IC(x,y) = \sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} |I_1(x+i,y+j) - I_2(x+i,y+j)| \quad \text{Eq. 4.4}$$

Para o índice de similaridade, o valor da faixa de comparação  $u$  influencia no grau de atenuação das descontinuidades da linha de costura. Por exemplo, um valor de  $u$  igual a 0, como considerado anteriormente, pode gerar casos de descontinuidades abruptas na vizinhança da costura. Valores muito grandes de  $u$  podem evitar este problema, mas podem diminuir a sensibilidade do índice de correlação e, conseqüentemente, a atenuação das descontinuidades.

Outros índices de correlação tais como o Índice de Similaridade por Média Normalizada, dado por :

$$IC(x,y) = \sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} \left| (I_1(x+i,y+j) - \bar{I}_1(x,y)) - (I_2(x+i,y+j) - \bar{I}_2(x,y)) \right| \quad \text{Eq. 4.5}$$

e o Índice de Covariância :

$$IC(x,y) = \frac{\sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} (I_1(x+i,y+j) - \bar{I}_1(x,y))(I_2(x+i,y+j) - \bar{I}_2(x,y))}{\sqrt{\sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} (I_1(x+i,y+j) - \bar{I}_1(x,y))^2} \sqrt{\sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} (I_2(x+i,y+j) - \bar{I}_2(x,y))^2}} \quad \text{Eq. 4.6}$$

onde  $\bar{I}_1(x,y) = \frac{1}{u^2} \sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} I_1(x+i,y+j)$  e  $\bar{I}_2(x,y) = \frac{1}{u^2} \sum_{i=-\frac{u}{2}}^{\frac{u}{2}} \sum_{j=-\frac{u}{2}}^{\frac{u}{2}} I_2(x+i,y+j)$ , são menos

sensíveis a mudanças aditivas de amplitude [Colwell (1983)], [Gonzalez-Woods (1992)].

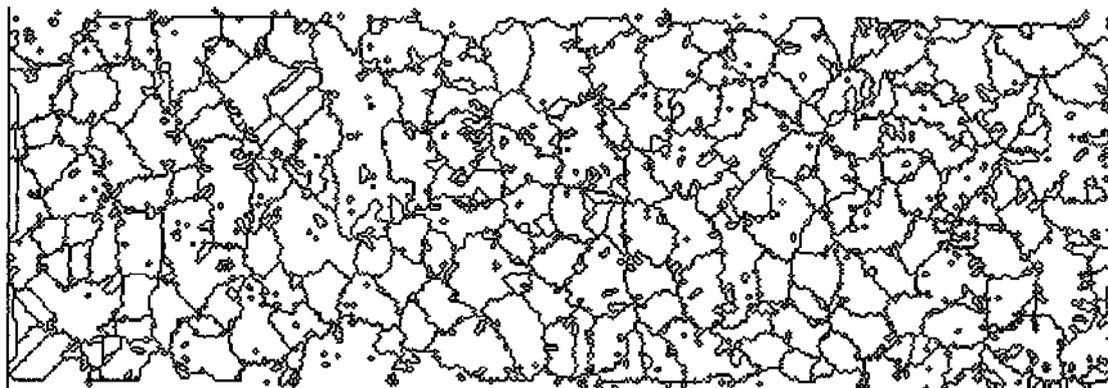


Figura 4.7 : LDA obtida da correlação 9x9 (Eq. 4.4) entre as imagens da Figura 4.1 e Figura 4.2.

A Figura 4.8 ilustra um fotomosaico a partir de um índice de correlação (Eq. 4.4) definido numa vizinhança 9x9. Observe que por gerar áreas de homogeneidade mais esparsas (compare as LDA da Figura 4.3 com a Figura 4.7), a costura final não passa necessariamente pela grade de divisão política, como na imagem da Figura 4.6 (diferenças absolutas pontuais,  $u = 0$ ), diminuindo o efeito de descontinuidades em alguns pontos da

linha de costura dessa imagem (“quebra” de nuvens).

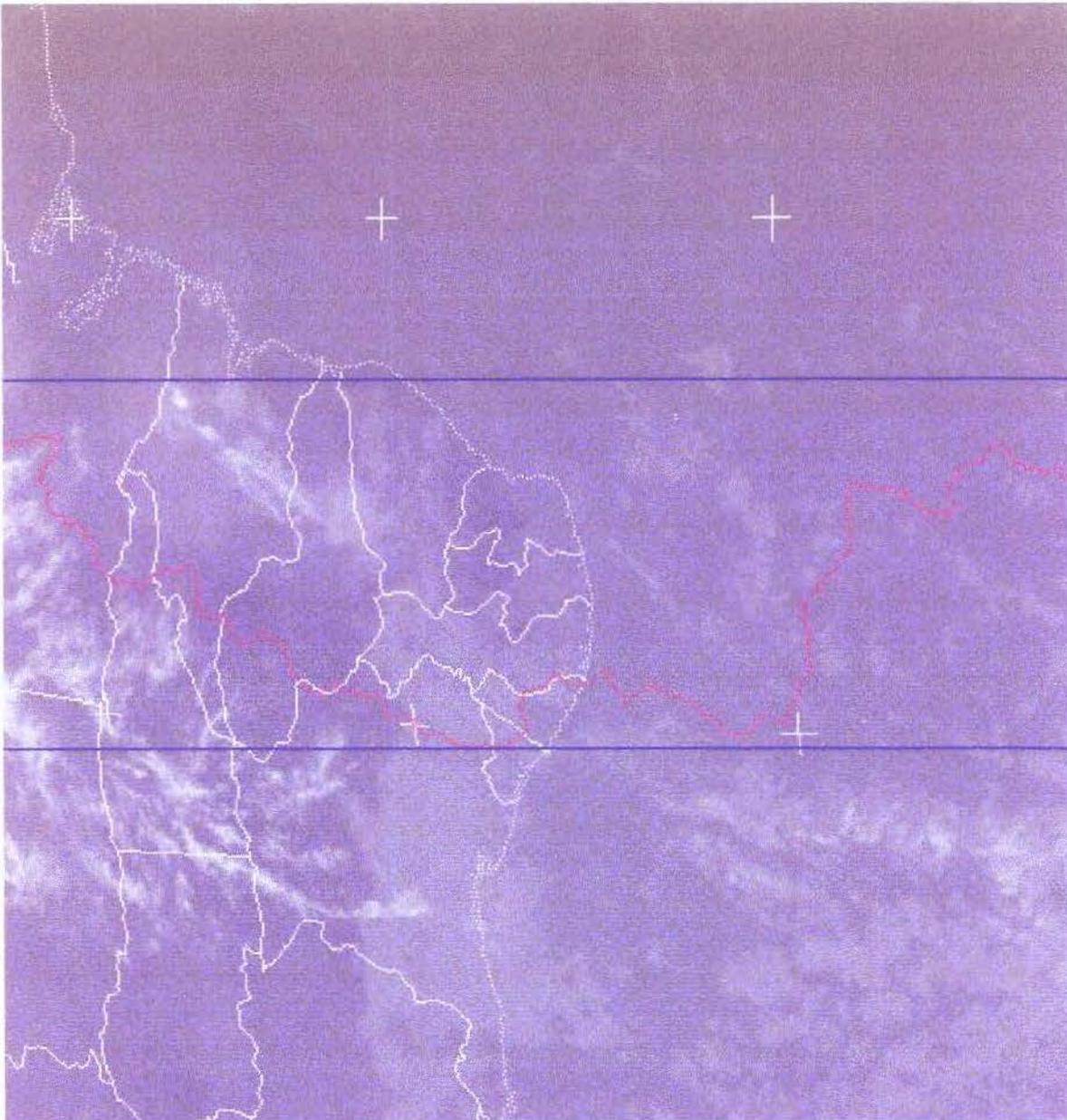


Figura 4.8 : Fotomosaico morfológico composto das imagens da Figura 4.1 e Figura 4.2, utilizando uma correlação  $9 \times 9$  (Eq. 4.4).

### 4.3.2 As Bordas da Imagem de Correlação

A fim de obtermos a disposição correta da linha de costura na região de interseção, os pontos das bordas da imagem de correlação precisam estar corretamente inicializados, antes do cálculo das LDA. A Figura 4.9 mostra algumas posições entre duas imagens de um fotomosaico.

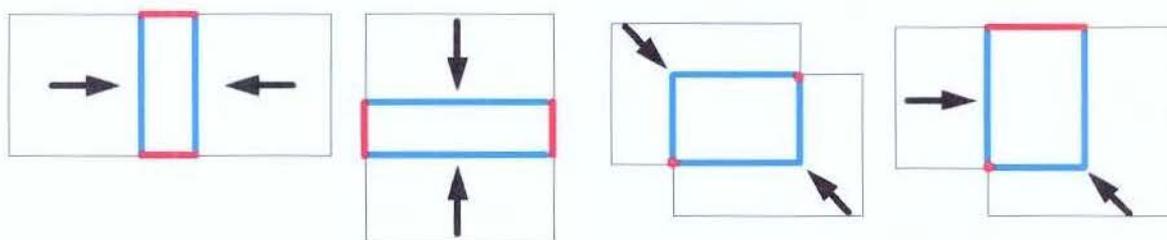


Figura 4.9 : Algumas posições entre as imagens de um mosaico. Em vermelho estão marcadas os pontos extremos das linhas de costuras. Em azul estão os pontos das zonas de influência das imagens sobre a região de interseção. As setas indicam o sentido desta influência.

As linhas vermelhas representam os pontos de interseção do retângulo envolvendo a região onde será definida a linha de costura. Estes pontos devem fazer parte do conjunto dos pontos extremos desta linha e, conseqüentemente, devem ser inicializados com zero (alta correlação). Por sua vez, os demais pontos do retângulo, orientando a direção da linha de costura, devem ser inicializados com um valor máximo  $N^{-1}$  (baixa correlação). Estes valores garantem a existência de LDA atravessando a região, de um extremo a outro dos pontos de interseção do retângulo. A Figura 4.10 ilustra este detalhe.

<sup>1</sup> Este valor  $N$  deve ser maior do que qualquer outro no interior da imagem.

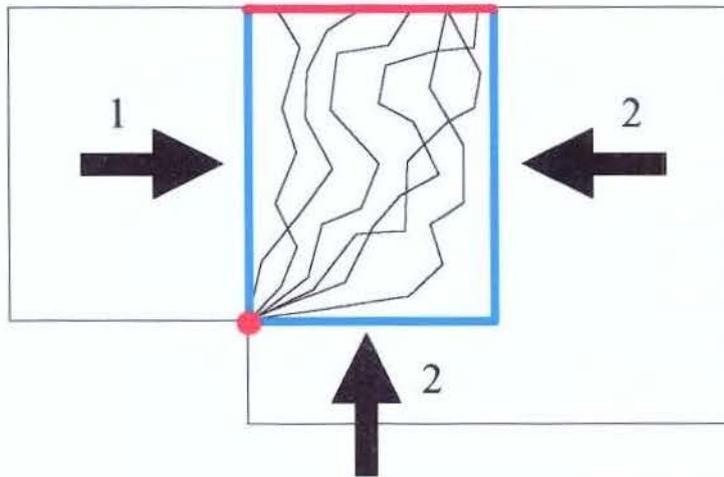


Figura 4.10 : Direção das LDA relativa às bordas da imagem de correlação.

### 4.3.3 Informações Locais versus Informações Globais

Para evidenciar a diferença entre o método proposto aqui e aqueles que levam em conta apenas informações locais dos pontos, vamos considerar a imagem da Figura 4.11. Esta imagem corresponde às diferenças absolutas entre os pontos das imagens que se sobrepõem, num fotomosaico vertical. O polígono na figura indica uma região de baixa correlação.

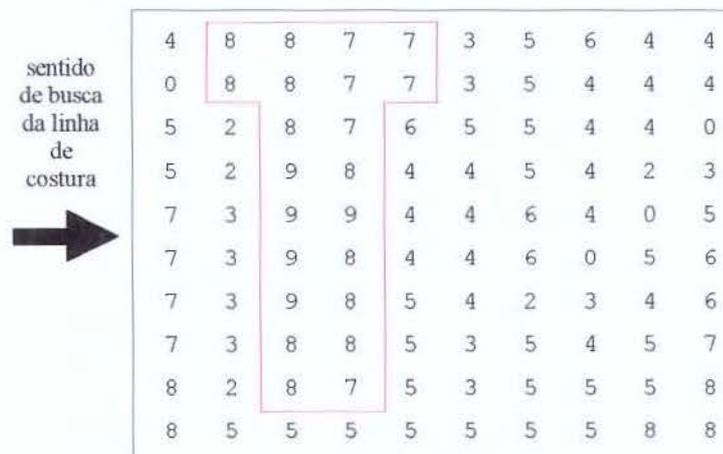


Figura 4.11 : Imagem de correlação com uma zona de baixa correlação.

O algoritmo de Milgram [Milgram (1977)] é um bom exemplo para ilustrar esta diferença. Como vimos na seção 2.3, ele cria uma costura com um ponto por coluna, definido como o menor valor numa vizinhança do ponto, de tamanho  $\omega$ . Para minimizar a descontinuidade na direção vertical, o método considera que após a detecção de um ponto  $(x,y)$ , o próximo ponto, na posição  $(x+1,k)$ , deva ser definido sobre  $\omega+1$  pontos, onde  $-\frac{\omega}{2} \leq k \leq \frac{\omega}{2} + 1$ .

A Figura 4.12 mostra a costura obtida (linha de menor custo) de acordo com o algoritmo de Milgram ( $\omega$  igual a 4).

Observe que neste caso alguns pontos de baixa correlação (pontos marcados por retângulos) pertencem à costura. Obviamente, podemos evitar este problema através do aumento da vizinhança monodimensional  $\omega$ . Na prática, este aumento implica na introdução de descontinuidade na direção horizontal (setas).

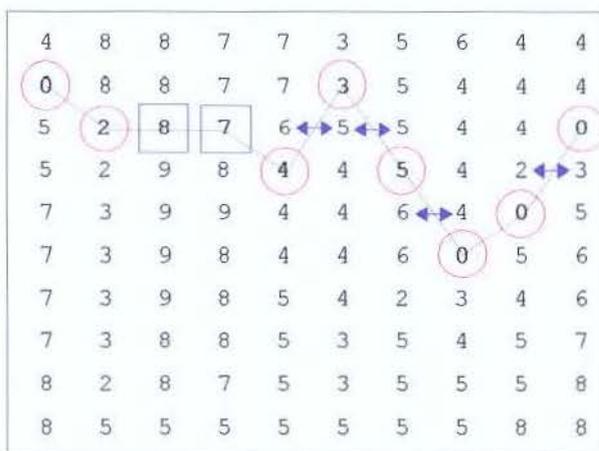


Figura 4.12 : Definição da costura baseada no método de Milgram.

A costura detectada através do método morfológico pode ser vista na Figura 4.13. Observe que a linha não atravessa regiões de baixa correlação. Outra diferença é que a linha de costura é conexa. O critério de conexidade é garantido pela propriedade homotópica do algoritmo de linha de divisor de águas.

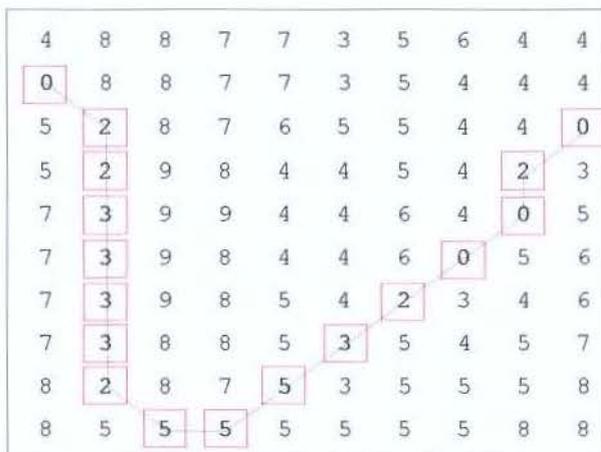


Figura 4.13 : Definição da costura baseada no método morfológico.

## 5. Conclusão

Nós definimos, neste trabalho, um algoritmo para construção de fotomosaicos baseado numa operação morfológica que vem sendo muito utilizada em problemas de segmentação de imagens. A partir da extração de informações globais de uma imagem, a noção de linha de divisor de águas de uma função permite-nos definir uma costura conexa, irregular e, portanto, mais realista do que aquelas obtidas por outros métodos de photomosaicking. Além disto, devido à natureza paralela das operações morfológicas consideradas aqui, o método pode ser facilmente descrito de acordo com o modelo SIMD (Single Instruction Multiple Data) de computação.

A Figura 5.1 apresenta um ambiente onde foram implementados os métodos mencionados neste trabalho. O programa é portátil e o algoritmo está sendo atualmente implementado no sistema geográfico de informação SPRING, do INPE (Instituto Nacional de Pesquisas Espaciais).

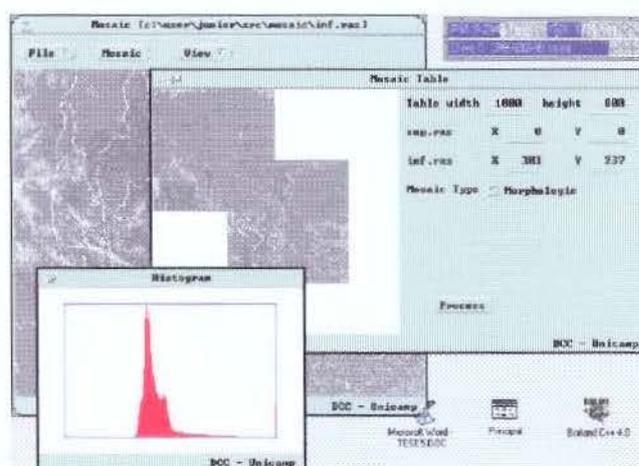


Figura 5.1 : Tela do programa de photomosaicking.

Uma extensão natural deste trabalho diz respeito aos métodos de composição de fotomosaicos para imagens coloridas e considerações de métodos globais para a detecção da linha ótima da imagem de LDA (baseada num custo topográfico dos caminhos, por exemplo).

## Apêndice A - Fontes de Erros

### A.1 Tipos de Distorções

Uma grande variedade de distorções em imagens têm sua origem no sensor ou na natureza da cena vista pelo sensor. O sistema sensor consiste do mecanismo detector de energia ou radiação mais a plataforma de aquisição no qual está montado. A geração desses erros é involuntária e está relacionada, principalmente, com imprecisões do mecanismo detector e de navegação da plataforma.

Para o caso do Landsat [Colwell (1983)], por exemplo, a cena adquirida por seus sensores é captada ponto à ponto, linha à linha. Cada linha é montada através de uma varredura transversal à trajetória da plataforma de aquisição (Figura A.1). À incrementos do seu deslocamento, no sentido de sua trajetória, as linhas são captadas consecutivamente para formar o conjunto de linhas da imagem.

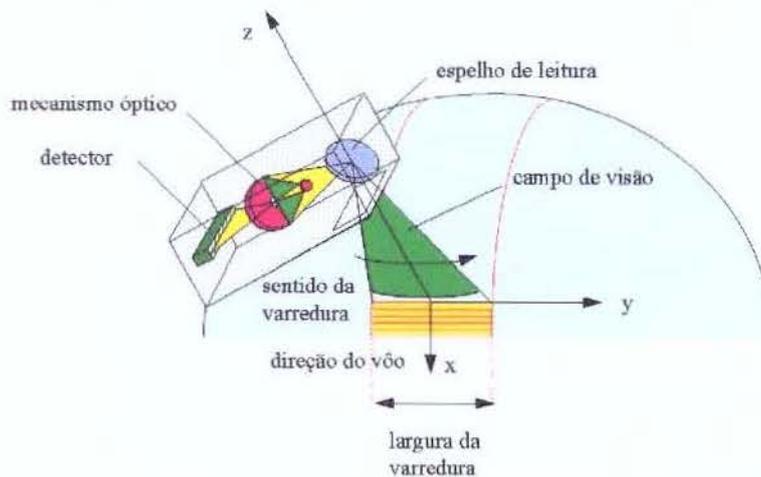


Figura A.1 : Varredura multiespectral.

Como consequência do conjunto sensor/plataforma/cena, os erros são agrupados em duas categorias. Os *erros geométricos* que alteram a relação entre o valor medido (ou captado) e a sua correta localização dentro da cena, e os *erros radiométricos* que alteram a relação entre a informação medida e seu valor real. Além dessa classificação,

podemos dividir os tipos de erros, baseado em sua fonte (ou origem), em 5 categorias [Bernstein (1976)], relacionadas a seguir.

### 1. Efeitos da Plataforma

- **Altitude** - Uma falha no sistema de navegação que leve a desvios na altitude nominal da plataforma produzirão distorções de escala nos dados captados pelo sensor. Por exemplo, se a altitude nominal da plataforma diminuir os dados irão apresentar uma elevação na escala, devido a uma aproximação da plataforma em direção à cena.

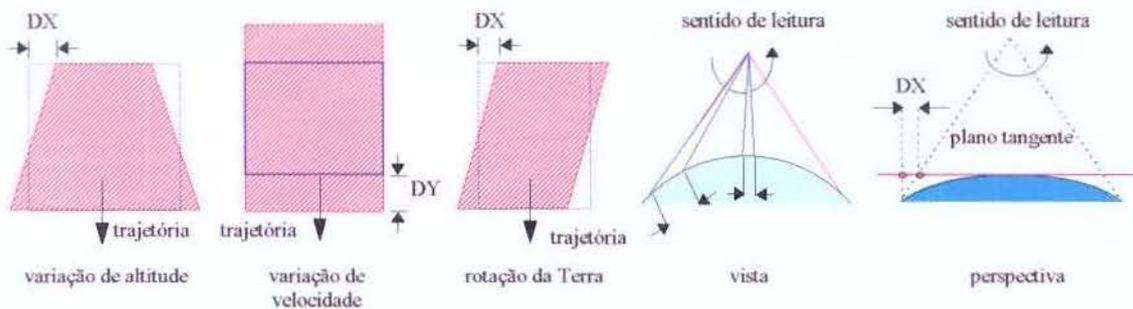


Figura A.2 : Distorções comuns em imagens de sensoriamento remoto.

- **Altitude** - O sistema de coordenadas da plataforma é composto de 3 eixos : vertical, trajetória e transversal. O eixo *vertical* é coincidente com a normal à superfície da terra. O eixo *trajetória* é paralelo ao vetor velocidade da plataforma. E o eixo *transversal* é perpendicular aos outros dois. Os desvios de atitude são provocados por rotações destes eixos, e geram deformações geométricas na imagem. Uma rotação no eixo transversal irá gerar uma *variação de altura*, e conseqüentemente irá deslocar o local de varredura do sensor de sua posição original, atrasando-a ou adiantando-a. Uma rotação no eixo vertical irá provocar uma *guinada* na plataforma, cujo efeito é uma rotação do local de varredura em torno deste eixo. A rotação no eixo trajetória irá provocar um *balanço* que deslocará lateralmente o local de varredura (veja Figura A.3).

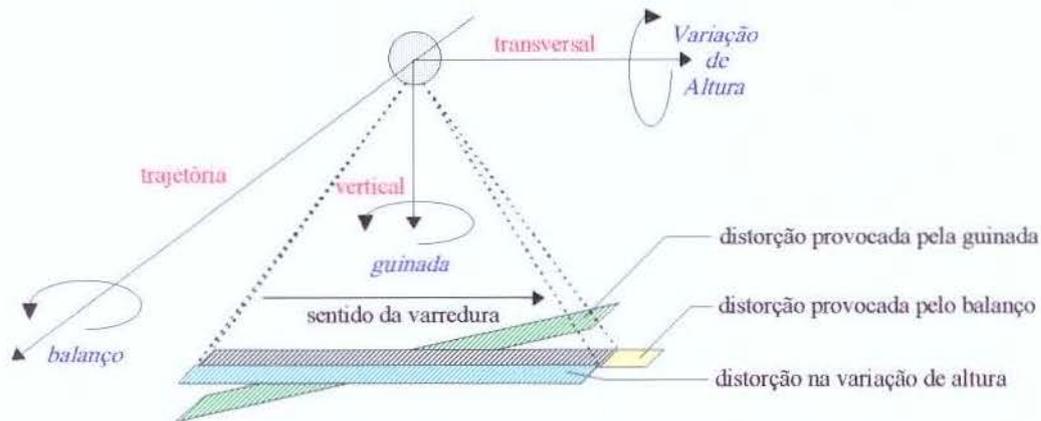


Figura A.3 : Distorções de atitude.

- **Velocidade** - A velocidade nominal da plataforma deve ser mantida sob controle para que a resolução vertical (no sentido da trajetória) permaneça dentro do seu valor nominal. Uma variação na velocidade nominal irá alterar a quantidade de terra coberta pela passagem da plataforma ao longo de sua trajetória. Dessa forma, a resolução vertical terá seu valor distorcido e proporcional a variação de velocidade.

## 2. Efeitos da Cena

- **Rotação da Terra** - A cena captada pelo sensor é uma amostra da região observada cuja resolução é proporcional à altitude da plataforma, ao tamanho do campo de visão do sensor e à velocidade da plataforma. Estes dois últimos definem a resolução horizontal e vertical, respectivamente. Porém, a cena sob o sensor está em constante deslocamento ocasionado pelo movimento de rotação da Terra. Assim, há um gradual deslocamento, no sentido leste-oeste, na cena captada pelo sensor. Isto causa uma distorção geométrica ao longo da leitura da linha.
- **Projeção** - A imagem captada pelo sensor é armazenada, inicialmente, no formato (ou projeção) no qual a plataforma enxerga a Terra. Porém, sua utilização somente é importante se for possível compará-la com os mapas disponíveis. Para isso, é necessário que a imagem sofra transformações que a levem para o

formato (ou projeção) no qual os mapas são construídos. Embora essas transformações não constituam propriamente erros geométricos, elas operam com os dados de forma semelhante as transformações que corrigem as distorções nos dados. E portanto estão sujeitas a erros da mesma ordem de grandeza.

### 3. Efeitos do Sensor

- **Falha de Detector** - Os sensores utilizados para captar imagens usam detectores que transformam o sinal de radiância emitida pela cena em uma voltagem ou número digital [Colwell (1983)]. Múltiplos detectores são utilizados em um sistema sensor, e devido a variações na saída individual de um detector, existirá diferentes saídas para o mesmo nível de radiância. O sinal captado pode sofrer influências aditivas (deslocamento) ou multiplicativas (ganho), diferenciando o sinal de saída do seu valor real (falha radiométrica).
- **Falha no Conversor Analógico/Digital** - O conversor analógico-digital, responsável por transformar o nível de radiância (sinal analógico) em uma quantização particular (sinal digital), tem um finito número de bits para realizar esta quantização. A seleção de um conversor com um número suficiente de bits (8 bits ou mais) é ideal para a minimização dos efeitos que uma quantização de ruídos pode provocar (ruídos confundindo-se com os valores medidos). Se o conversor analógico/digital tem um erro de deslocamento ou ganho (erro radiométrico), ele pode também degradar os dados de maneira similar ao erro de deslocamento e ganho do detector [Colwell (1983)].

### 4. Efeitos do Sensor e da Cena

- **Vista** - A área de uma cena vista pelo sensor é deformada para os pontos pertencentes aos extremos da linha, a medida que se afastam da Nadir. Esta deformação é proporcional à tangente do ângulo de leitura (ângulo entre o ponto enxergado na superfície da Terra e a normal passando pela Nadir), em vez

do próprio ângulo, e assim, uma amostra dos dados é feita a intervalos regulares de tempo, e isto produz uma deformação ao longo da linha.

- **Perspectiva** - Para muitas aplicações de sensoriamento remoto, a imagem gerada pelo sensor representa a projeção dos pontos da Terra em um plano tangente passando pela Nadir, com todas as linhas de projeção sendo normal ao plano. Os dados do sensor, entretanto, representam projeções perspectivas, isto é, projeções cujas linhas se encontram em um ponto acima do plano tangente. Em alguns tipos de sensores, como é o caso do Landsat, isto só produz distorções ao longo da linha [Bernstein (1976)].

#### 5. Efeitos Atmosféricos

- Os efeitos atmosféricos não podem ser considerados propriamente como erros. Todavia, consideramos freqüentemente os efeitos atmosféricos de *espalhamento* (difusão de uma parte da radiação incidente em todas as direções) e *absorção* (retenção e conversão de parte da energia radiante em outras formas de energia) como sendo uma fonte de erro que pode minimizar nossa habilidade de extrair informações úteis de dados de sensoriamento remoto [Jensen (1986)]. A radiação solar não é muito afetada em sua trajetória através do vácuo espacial. Entretanto, quando interage com a atmosfera terrestre ela é seletivamente espalhada e absorvida. O efeito da atmosfera no brilho espectral cria problemas quando tentamos comparar as medições de respostas espectrais feitas no local com medições do sensor, por exemplo, para uma mesma área geográfica. Isto afeta a exatidão da classificação. A compensação desses efeitos é muito difícil, mas, se ela pode ser feita, aumenta a exatidão dessas informações remotamente obtidas [Bernstein (1976)].

## A.2 Correção Radiométrica

Os sistemas de sensores requerem calibração e correção dos dados para obtenção de dados confiáveis [Jensen (1986)]. A importância da correção radiométrica certamente aumentará a qualidade das pesquisas de monitoramento de mudanças ambientais globais que usam imagens de satélite [Ahern et al (1988)]. A calibração/correção radiométrica nominal dos dados é o primeiro tipo de operação de correção efetuado nos dados. Caso esta correção falhe na tentativa de compensar os erros radiométricos, uma calibração suplementar deverá ser operada [Colwell (1983)].

Sensores remotos usam detectores que convertem a radiância captada em valores de voltagem ou brilho [Jensen (1986)]. Esses instrumentos detectores são analisados com dados dos testes de pré-lançamento para determinar as características operacionais dos instrumentos, como sua razão sinal/ruído, estabilidade, linearidade de resposta, e sensibilidade (saída em unidades digitais por unidade de radiância) [NOAA (1988)]. Entretanto, estas características de pré-lançamento não são as mesmas quando o equipamento está em órbita. Uma razão para isso é que o ambiente termal varia com a posição na órbita, fazendo a sensibilidade variar durante o percurso da órbita. Além disso, os componentes dos instrumentos envelhecem bastante no intervalo de tempo entre os testes de pré-lançamento e o lançamento, e continuam envelhecendo durante o tempo no qual os instrumentos ainda permanecem em órbita [NOAA (1988)]. Por isso é necessário efetuar calibrações frequentemente em seus dados.

Os dados de calibração irão produzir pontos na curva de entrada e saída de cada detector, que são utilizadas para fazer a calibração/correção nominal. Observações solares ocasionais são utilizadas no processo de correção para detectar possíveis mudanças nos sensores. Dessa forma, mudanças graduais no sinal de cada detector podem ser detectadas e medidas [Colwell (1983)]. Estes dados suplementares de calibração podem ser utilizados para construir, para cada detector, uma tabela que especifica a intensidade radiométrica corrigida para cada valor de saída do detector [Colwell (1983)].

Estas técnicas nominais de correção nem sempre fazem completamente a compensação para diferenças na saída dos vários detectores [Colwell (1983)]. Quando isto ocorre, os efeitos são algumas vezes suficientemente grandes para produzir “listras” (ou “faixas”) horizontais visíveis nas imagens, ou perda parcial ou completa de linhas [Jensen (1986)]. Uma calibração suplementar pode ser freqüentemente utilizada para reduzir estes efeitos abaixo do nível de detecção visual.

Quando um dos detectores falham completamente, a informação é perdida e em seu lugar serão colocados zeros, que aparecerão como linhas pretas. Porém não há como recuperar um dado que nunca foi adquirido. Entretanto é possível melhorar o aspecto visual pela introdução de valores estimados para as linhas pedidas, baseado em sua vizinhança.

Algumas vezes, um detector pode falhar parcialmente, e ficar fora de ajuste (provê resultados de leituras 2 vezes maior que os outros detectores, por exemplo). Quando isso ocorre, a imagem irá aparecer com listras ou faixas horizontais. O dado é válido, mas deve ser corrigido (recuperado) para ter o mesmo contraste que os outros detectores. A correção consiste de medir as diferenças na curva de resposta do detector, em diferentes bandas espectrais, sobre áreas de radiância uniforme (superfícies de água, por exemplo) para diferentes níveis de intensidades e modificar as tabelas nominais de correção radiométrica para produzir curvas de respostas idênticas para os detectores em cada banda [Colwell (1983)].

Deve-se enfatizar que a calibração suplementar produz apenas um efeito cosmético. Ao contrário das técnicas de calibração nominal, a calibração suplementar não tenta fazer a saída do detector correta. Ela tenta somente fazê-las iguais [Colwell (1983)].

## A.3 Correção Geométrica

Muitas abordagens de correção de distorções geométricas consistem basicamente de três passos, ou seja, a localização de pontos de controle, a determinação do modelo de distorção, e o processo de correção propriamente dito [Colwell (1983)].

- **Localização de Pontos de Controle** - Para se efetuar a correção das distorções geométricas em uma imagem, é necessário que se localize nesta imagem os pontos de controle. Estes pontos estão geralmente associados a acidentes geográficos, lagos, estradas, etc, cuja localização geográfica é conhecida. Esta localização é necessária pois a posição geográfica destes pontos de controle são parâmetros do processo de modelagem das distorções, descrito a seguir.
- **Modelagem das Distorções Geométricas** - Os modelos de descrição das distorções são divididos em duas classes : aqueles que podem ser computados através de tabelas ou parâmetros da plataforma de aquisição e do sensor, e aqueles que não podem ser computados se um número mínimo de pontos de controle não existir [Colwell (1983)].

Na primeira classe, encontram-se muitos erros conhecidos previamente, e que estão relacionados com a natureza de captação das imagens, como por exemplo, a distorção panorâmica, a inclinação causada pela rotação da Terra, a projeção, etc. Estes erros são também chamados de erros sistemáticos.

Os efeitos de erros de atitude e de posicionamento, são considerados erros não sistemáticos, e são eliminados (ou reduzidos) usando os pontos de controle. Estes pontos de controle são utilizados para modelar as variações causadas pelo balanço, guinada, e mudanças de altura [Colwell (1983)].

A modelagem das distorções geométricas demandam operações específicas para cada tipo de erro geométrico, e vão sendo sucessivamente aplicadas para compor um modelo geral das distorções geométricas. As equações utilizadas

---

para realizar estas correções podem ser vistas, para maiores detalhes, em [Colwell (1983)].

- **Processo de Correção** - O processo de correção se resume a efetuar a transformação dos dados de entrada em um novo conjunto de saída, através de um mapeamento de espaços. A saída da função de mapeamento entre espaços é uma tabela de coordenadas, de linhas e colunas, no espaço de entrada correspondendo as coordenadas da grade fixa de interpolação no espaço de saída. As coordenadas do espaço de entrada correspondendo a uma dada coordenada no espaço de saída pode então ser achada por uma interpolação bilinear no espaço de entrada.



## B.2 Afinamento Numérico

```
// Verifica se um ponto e' passivel de afinamento numerico em sua vizinhanca
// 8-conexa. Retorna "YES" se positivo. Este afinamento utiliza a familia "M".
//
// M 1(0)   M 2(1)   M 3(2)   M 4(3)   M 5(4)   M 6(5)   M 7(6)   M 8(7)
// -----
// X 0 X   X 0 0   X 1 X   X 1 X   X 1 X   X 1 X   X 1 X   0 X X
// 1 1 1   1 1 X   1 1 0   1 1 0   1 1 1   X 1 1   0 1 1   0 1 1
// X 1 X   X 1 X   X 1 X   X X 0   X 0 X   0 0 X   X 1 X   X 1 X
//
//           M 2'(8)           M 4'(9)           M 6'(10)           M 8'(11)
//           -----
//           0 0 X           X 1 X           X 1 X           X X 0
//           X 1 1           0 1 1           1 1 X           1 1 0
//           X 1 X           0 X X           X 0 0           X 1 X
//
int GreyThinning(DATA v0, // valor do elemento a ser verificado
                DATA v1, // Vizinho superior
                DATA v2, // Vizinho superior direito
                DATA v3, // Vizinho direito
                DATA v4, // Vizinho inferior direito
                DATA v5, // Vizinho inferior
                DATA v6, // Vizinho inferior esquerdo
                DATA v7, // Vizinho esquerdo
                DATA v8, // Vizinho superior esquerdo
                DATA member, // Membro da familia "M" a ser observada
                DATA *value) { // Valor afinado, se for V0 for "afinavel"

    DATA min, max;

    switch (member) {
        case 0:
            max=v1;
            min=MMIN(v3,v5,v7);
            break;
        case 1:
            max=MAX(v1,v2);
            min=MIN(v5,v7);
            break;
        case 2:
            max=v3;
            min=MMIN(v5,v7,v1);
            break;
        case 3:
            max=MAX(v3,v4);
            min=MIN(v7,v1);
            break;
        case 4:
            max=v5;
            min=MMIN(v7,v1,v3);
            break;
        case 5:
            max=MAX(v5,v6);
            min=MIN(v1,v3);
            break;
        case 6:
            max=v7;
            min=MMIN(v1,v3,v5);
            break;
        case 7:
            max=MAX(v7,v8);
            min=MIN(v3,v5);
            break;
        case 8:
    
```

```

        max=MAX(v8,v1);
        min=MIN(v3,v5);
        break;
    case 9:
        max=MAX(v6,v7);
        min=MIN(v1,v3);
        break;
    case 10:
        max=MAX(v4,v5);
        min=MIN(v7,v1);
        break;
    case 11:
        max=MAX(v2,v3);
        min=MIN(v5,v7);
        break;
    }
    if ( max<v0 && v0<=min ) {
        *value=max;
        return 1;
    }
    return 0;
}

// Calcula o afinamento numerico de uma imagem
void GreyThinning(DATA POINTER image, // Buffer da imagem a ser afinada
                 LONGINT w,         // Numero de colunas do buffer
                 LONGINT h) {       // Numero de linhas do buffer

    // Buffer auxiliar
    DATA POINTER buffer,
    // Ponteiros moveis para o buffer e sua vizinhanca 8-conexa
    POINTER pb,
    POINTER pb1, POINTER pb2, POINTER pb3, POINTER pb4,
    POINTER pb5, POINTER pb6, POINTER pb7, POINTER pb8,
    // Ponteiro movel para a imagem
    POINTER pi;
    // Membro da familia a ser utilizada
    DATA m,
    // Valor afinado de retorno
    v;
    // Coluna e linha durante varredura
    LONGINT x, y;
    // Controle do laço (durante todas as passadas)
    int fall,
    // Controle de substituicao (durante a ultima passada)
    fthis;

    // Tentar alocar memoria
    if ( (buffer=MemoryAlloc(w*h))! ) return;

    // Fazer copia da imagem em buffer
    for (pi=image,pb=buffer,y=0; y<h; y++)
        for (x=0; x<w; x++,pb++,pi++) *pb=*pi;

    do { // Laço de afinamento
        for (fall=TRUE,m=0; m<12; m++) {
            for (fthis=TRUE,y=1; y<h-1; y++) {
                pi=image+y*w+1;
                pb=buffer+y*w+1;
                pb1=pb-w; pb2=p1+1; pb3=p2+w; pb4=p3+w;
                pb5=p4-1; pb6=p5-1; pb7=p6-w; pb8=p7-w;
                for (x=1;x<w-1;x++) {
                    // Verifica se o ponto "pb" e'passivel de afinamento
                    if ( GreyThinning(*pb,*pb1,*pb2,*pb3,*pb4,*pb5,*pb6,*pb7,*pb8,m,&v) ){
                        // Atualizar o ponto com o valor afinado
                        *pi=v;
                        // Sinalizar esta passada : um ponto foi afinado
                        fall=fthis=FALSE;
                    }
                }
            }
        }
    } while (fall);
}

```

```

    }
    pi++;pb++;pb1++;pb2++;pb3++;pb4++;pb5++;pb6++;pb7++;pb8++;
  }
}
// Se ocorreu o afinamento de algum ponto, durante todas as passadas
if ( !fthis )
  // Colocar esta passada sobre "image" novamente em "aux".
  for (pi=image,pb=buffer,y=0; y<h; y++)
    for (x=0; x<w; x++,pi++,pb++)
      *pb=*pi;
}
} while ( !fall );

// Liberar memoria auxiliar
MemoryFree(buffer);
}

```

## B.3 Divisor de Águas

```

// Calcula o negativo de uma imagem.
void Negative(DATA POINTER pout, // Buffer de saida (pode ser igual a "in"
             DATA POINTER pin, // Buffer de entrada
             LONGINT w,        // Numero de colunas do buffer
             LONGINT h) {      // Numero de linhas do buffer

  for (LONGINT y=0; y<h; y++)
    for (LONGINT x=0; x<w; x++,pin++,pout++) *pout=(DATA) (255-(int)(*pin));
}

// Verifica se um ponto ("v0") e' um ponto de linha de divisor de aguas(LDA),
// dado sua vizinhanca 4-conexa. Retorna "YES" se for ponto de LDA.
int IsWatershed(DATA v0, // Ponto a ser verificado
               DATA v1, // Vizinho superior
               DATA v3, // Vizinho direito
               DATA v5, // Vizinho inferior
               DATA v7) { // vizinho esquerdo

  // Valor maximo na vizinhanca 4-conexa de "v0"
  DATA max4=MAX(v0,MAX(v1,MAX(v3,MAX(v5, v7))));
  // Valor minimo na vizinhanca 4-conexa de "v0"
  DATA min4=MIN(v0,MIN(v1,MIN(v3,MIN(v5, v7))));
  // Verifica se "v0" e' o maximo local. Isto e', se "v0" for igual a
  // max4, mas sem que seja tambem o minimo local(min4).
  if ( max4!=min4 && v0==max4 ) return YES;

  // Quantidade de vezes que "v0" e'maior que seus vizinhos
  DATA t=0;
  if ( v1<v0 ) t++;
  if ( v3<v0 ) t++;
  if ( v5<v0 ) t++;
  if ( v7<v0 ) t++;
  // Se nao for no minimo maior do que dois vizinhos
  if ( t<=1 ) return NO;

  // Valor maximo na vizinhanca 4-conexa horizontal(`a esquerda e `a direita) de "v0"
  DATA maxh=MAX(v0,MAX(v3,v7));
  // Valor minimo na vizinhanca 4-conexa horizontal(`a esquerda e `a direita) de "v0"
  DATA minh=MIN(v0,MIN(v3,v7));
  // Verifica se "v0" e' maximo local horizontalmente
  if ( maxh!=minh && v0==maxh ) return YES;

  // Valor maximo na vizinhanca 4-conexa vertical (acima e abaixo) de "v0"
  DATA maxv=MAX(v0,MAX(v1,v5));
  // Valor minimo na vizinhanca 4-conexa vertical (acima e abaixo) de "v0"
  DATA minv=MIN(v0,MIN(v1,v5));
  // Verifica se "v0" e' maximo local horizontalmente

```

```

    if ( maxv!=minv && v0==maxv ) return YES;

    return NO;
}
// Esta funcao verifica se algum ponto de LDA foi esquecido pela funcao
// "IsWatershed" declarada acima. Neste caso, o criterio e' a conexidade
// do ponto em relacao a um conjunto de LDA ja definidas. Isto e', pontos
// que sejam maiores que no minimo 2 vizinhos e, que sejam "conexoes",
// "bifurcacoes" ou "cruzamentos" sao, agora, tomados como pertencentes ao
// conjunto LDA. IMPORTANTE : Assume-se, aqui, que os pontos a serem
// verificados nao sao pontos de borda.
int IsWatershedConex(DATA POINTER pi, // Buffer numerico
                    DATA POINTER pw, // Buffer com LDA
                    LONGINT w) { // Numero de colunas do buffer

// Quantidade de vezes que "v0" e'maior que seus vizinhos
DATA t=0;
if ( *(pi-w)< *pi ) t++;
if ( *(pi+1)< *pi ) t++;
if ( *(pi+w)< *pi ) t++;
if ( *(pi-1)< *pi ) t++;
// Se nao for no minimo maior do que dois vizinhos
if ( t<=1 ) return NO;

// Calcular o numero de conexidade de Rutovitz :
// 0 -> ponto interno      -> nao pode pertence a LDA.
// 1 -> ponto de borda     -> não pode pertence a LDA.
// 2 -> ponto de conexao   -> pode pertencer a LDA.
// 3 -> ponto de bifurcacao -> pode pertencer a LDA.
// 4 -> ponto de cruzamento -> pode pertencer a LDA.
DATA rutovitz=(ABS(*(pw-w)  -*(pw-w+1)) + // pb1 - pb2
              ABS(*(pw-w+1)-*(pw+1))  + // pb2 - pb3
              ABS(*(pw+1)  -*(pw+1+w)) + // pb3 - pb4
              ABS(*(pw+1+w)-*(pw+w))  + // pb4 - pb5
              ABS(*(pw+w)  -*(pw+w-1)) + // pb5 -pb6
              ABS(*(pw+w-1)-*(pw-1))  + // pb6 -pb7
              ABS(*(pw-1)  -*(pw-1-w)) + // pb7 -pb8
              ABS(*(pw-1-w)-*(pw-w))  ); // pb8 -pb1

// Se for no minimo conexao e' ponto de LDA
if ( rutovitz>1 ) return YES;
return NO;
}

// Esta funcao verifica se todos os vizinhos de um ponto sao da mesma cor.
// Se verdadeiro, entao esta cor unica e' retornada. A cor deve estar acima
// de um valor minimo (minimum).
SHORTINT
UniqueColor(LONGINT minimum, // valor minimo
            LONGINT v1,      // valor do vizinho superior
            LONGINT v2,      // valor do vizinho superior direito
            LONGINT v3,      // valor do vizinho direito
            LONGINT v4,      // valor do vizinho inferior direito
            LONGINT v5,      // valor do vizinho inferior
            LONGINT v6,      // valor do vizinho inferior esquerdo
            LONGINT v7,      // valor do vizinho esquerdo
            LONGINT v8,      // valor do vizinho supeior esquerdo
            LONGINT *unicolor) { // valor da unica cor na vizinhanca

LONGINT v=0;
if ( v1>=minimum ) { v=v1; }
if ( v2>=minimum ) { if ( v==0 ) { v=v2; } else if ( v2!=v ) return 0; }
if ( v3>=minimum ) { if ( v==0 ) { v=v3; } else if ( v3!=v ) return 0; }
if ( v4>=minimum ) { if ( v==0 ) { v=v4; } else if ( v4!=v ) return 0; }
if ( v5>=minimum ) { if ( v==0 ) { v=v5; } else if ( v5!=v ) return 0; }
if ( v6>=minimum ) { if ( v==0 ) { v=v6; } else if ( v6!=v ) return 0; }
if ( v7>=minimum ) { if ( v==0 ) { v=v7; } else if ( v7!=v ) return 0; }
if ( v8>=minimum ) { if ( v==0 ) { v=v8; } else if ( v8!=v ) return 0; }
if ( v==0 ) return NO;
*unicolor=v;
}

```

```

    return YES;
}

// Esta funcao faz um melhoramento na imagem de divisor de aguas.
// Ela verifica quais ponto da imagem afinada numericamente sao realmente
// pontos de divisores de aguas. Para fazer isto, todos os vales sao etiquetados
// com um valor unico, e todos os pontos da imagem afinada que estao entre dois
// pontos da mesma etiqueta sao removidos.
void
Improvemem(DATA POINTER water,      // Image afinada
           LONGINT w, LONGINT h) { // Largura e altura da imagem afinada

    LONGINT POINTER buffer,
            POINTER pb,
            POINTER pb1, POINTER pb2, POINTER pb3, POINTER pb4,
            POINTER pb5, POINTER pb6, POINTER pb7, POINTER pb8;

    DATA POINTER pw;
    LONGINT x,y;

    if ( (buffer=(LONGINT POINTER)MemoryAlloc(w*h*sizeof(LONGINT)))==NULL ) return;

    // Copiar water em buffer
    pb=buffer;
    pw=water;
    for (y=0; y<h; y++)
        for (x=0; x<w; x++,pw++,pb++) *pb=(DATA)*pw;

    // Iniciar etiquetagem
    LONGINT label=10;
    pb=buffer;
    for (y=0; y<h; y++)
        for (x=0; x<w; x++,pb++)
            if ( *pb==0 ) { // valley
                // Esta funcao esta descrita em [Foley et al (1993)]
                KFloodFill2(buffer,x,y,w,h,*pb,label);
                label++;
            }

    LONGINT color;

    // Processo de eliminacao : se nao mais do que uma etiqueta esta na
    // vizinhanca 8-conexa de um ponto, e na vizinhanca 4-conexa deste ponto
    // existirem etiquetas em simetria (em cima e embaixo, ou a esquerda e a
    // direita), entao o ponto deve ser removido.
    for (y=1; y<h-1; y++) {
        pb=buffer+y*w+1;
        pb1=pb-w; pb2=pb1+1;
        pb3=pb2+w; pb4=pb3+w;
        pb5=pb4-1; pb6=pb5-1;
        pb7=pb6-w; pb8=pb7-w;
        for (x=1; x<w-1; x++,pb++,pb1++,pb2++,pb3++,pb4++,pb5++,pb6++,pb7++,pb8++)
            if ( *pb==1 ) // E' uma linha afinada
                // nao mais do que uma etiqueta esta na vizinhanca 8-conexa
                if ( UniqueColor(10,*pb1,*pb2,*pb3,*pb4,*pb5,*pb6,*pb7,*pb8, &color) )
                    if ( (*pb1==color&&*pb5==color)||(*pb3==color&&*pb7==color) )
                        *pb=color;
    }

    // Remake tha watershed image
    pw=water;
    pb=buffer;
    for (y=0; y<h; y++)
        for (x=0; x<w; x++,pb++,pw++)
            if ( *pb==1 ) *pw=1; else *pw=0;

    MemoryFree((DATA POINTER)buffer);
}

```

```

// Esta funcao verifica se um ponto e' ponto de extremo de objetos binarios
// (familia "E"). E'utilizada aqui para extrair pontos extremos das LDA.
// Retorna "YES" se for ponto extremo.
//
//      E 0      E 1      E 2      E 3      E 4      E 5      E 6      E 7
// ----      ----      ----      ----      ----      ----      ----      ----
// 0 0 0      0 0 0      0 0 X      0 0 X      X 1 X      X X 0      X 0 0      0 0 0
// 0 1 0      0 1 0      0 1 1      0 1 X      0 1 0      0 1 0      1 1 0      X 1 0
// X 1 X      0 X X      0 0 X      0 0 0      0 0 0      0 0 0      X 0 0      X 0 0
//
//              E 1'              E 3'              E 5'              E 7'
//              ----              ----              ----              ----
//              0 0 0              X 0 0              0 X X              0 0 0
//              0 1 0              X 1 0              0 1 0              0 1 X
//              X X 0              0 0 0              0 0 0              0 0 X
//
int IsExtreme(DATA v0,          // valor do elemento a ser verificado
              DATA v1,          // Vizinho superior
              DATA v2,          // Vizinho superior direito
              DATA v3,          // Vizinho direito
              DATA v4,          // Vizinho inferior direito
              DATA v5,          // Vizinho inferior
              DATA v6,          // Vizinho inferior esquerdo
              DATA v7,          // Vizinho esquerdo
              DATA v8,          // Vizinho superior esquerdo
              DATA member) { // Membro da familia "M" a ser observada

if ( v0 != 1 )
    return 0;
switch (member) {
case 0:
    if ( v1==0 && v2==0 && v3==0 && v5==1 && v7==0 && v8==0 )
        return YES;
    break;
case 1:
    if ( v1==0 && v2==0 && v3==0 && v6==0 && v7==0 && v8==0 )
        return YES;
    break;
case 2:
    if ( v1==0 && v3==1 && v5==0 && v6==0 && v7==0 && v8==0 )
        return YES;
    break;
case 3:
    if ( v1==0 && v4==0 && v5==0 && v6==0 && v7==0 && v8==0 )
        return YES;
    break;
case 4:
    if ( v1==1 && v3==0 && v4==0 && v5==0 && v6==0 && v7==0 )
        return YES;
    break;
case 5:
    if ( v2==0 && v3==0 && v4==0 && v5==0 && v6==0 && v7==0 )
        return YES;
    break;
case 6:
    if ( v1==0 && v2==0 && v3==0 && v4==0 && v5==0 && v7==1 )
        return YES;
    break;
case 7:
    if ( v1==0 && v2==0 && v3==0 && v4==0 && v5==0 && v8==0 )
        return YES;
    break;
case 8:
    if ( v1==0 && v2==0 && v3==0 && v4==0 && v7==0 && v8==0 )
        return YES;
    break;
case 9:
    if ( v1==0 && v2==0 && v3==0 && v4==0 && v5==0 && v6==0 )

```

```

        return YES;
    break;
case 10:
    if ( v3==0 && v4==0 && v5==0 && v6==0 && v7==0 && v8==0 )
        return YES;
    break;
case 11:
    if ( v1==0 && v2==0 && v5==0 && v6==0 && v7==0 && v8==0 )
        return YES;
    break;
}
return 0;
}

// Calcula a imagem de LDA de "image". Esta funcao e' baseada no algoritmo
// descrito no artigo : "EXTREMA OF GREY-TONE FUNCTION AND MATHEMATICAL MORPHOLOGY",
// de SERGE BEUCHER, 1983 IEEE.
// Esta funcao retorna um ponteiro para um buffer que contem a imagem LDA, na
// mesma dimensao de "image".
DATA POINTER WaterShed(DATA POINTER image, // Buffer da imagem numerica
                        LONGINT w,        // Numero de colunas do buffer
                        LONGINT h) {      // Numero de linhas do buffer

    // Ponteiro para o buffer da imagem afinada
    DATA POINTER thin, POINTER pt,
    // Ponteiro para o buffer da imagem LDA
    POINTER water, POINTER pw,

    // Ponteiro para buffer auxiliar
    POINTER buffer,

    // Ponteiros moveis para o buffer auxiliar
    POINTER pb,
    POINTER pb1,POINTER pb2,POINTER pb3,POINTER pb4,
    POINTER pb5,POINTER pb6,POINTER pb7,POINTER pb8;

    // Membro da familia
    DATA m,

    // Valores da vizinhanca de um ponto da imagem afinada
    v1, v3, v5, v7;

    // Coluna e linha durante a varredura
    LONGINT x, y;

    // Controle do laço (durante todas as passadas)
    int fall,

    // Controle de substitulcao (durante a ultima passada)
    fthis;

    // Alocar memoria para o buffer da imagem afinada
    if ( (thin=MemoryAlloc(w*h))==NULL )
        return NULL;

    // Calcular o negativo de "image"e colocar em "thin"
    Negative(thin,image,w,h);

    // Calcular o afinamento numerico de "thin"
    GreyThinning(thin,w,h);

    // Alocar memoria para buffer auxiliar
    if ( {buffer=MemoryAlloc(w*h))==NULL ) {
        MemoryFree(thin);
        return NULL;
    }

    // Alocar memoria para a imagem de LDA

```

```

if ( (water=MemoryAlloc(w*h))==NULL ) {
    MemoryFree(thin);
    MemoryFree(buffer);
    return NULL;
}

// Selecionar os pontos da imagem afinada que deveram pertencer a imagem LDA
for (pt=thin,pw=water,y=0; y<h; y++) {
    for (x=0; x<w; x++,pt++,pw++) {

        // Analizar pontos de borda
        if ( x==0 ) v7=0; else v7=*(pt-1);
        if ( x==w-1 ) v3=0; else v3=*(pt+1);
        if ( y==0 ) v1=0; else v1=*(pt-w);
        if ( y==h-1 ) v5=0; else v5=*(pt+w);

        // Verificar se "pt" é ponto de LDA
        if ( IsWatershed(*pt,v1,v3,v5,v7) )
            *pw=1;
        else
            *pw=0;
    }
}

// Selecionar pontos extra para pertencer a imagem de LDA.
for (y=1; y<h-1; y++)
    for (pt=thin+w*y+1,pw=water+y*w+1,x=1; x<w-1; x++,pt++,pw++)
        if ( IsWatershedConex(pt,pw,w) )
            *pw=1;

// Melhorar a imagem de LDA.
Improvement(water,w,h);

// Liberar memoria da imagem afinada
MemoryFree(thin);

// Fazer copia da imagem de LDA no buffer auxiliar "buffer"
for (pw = water, pb = buffer, y = 0; y < h; y++)
    for (x = 0; x < w; x++, pw++, pb++)
        *pb = *pw;

// Eliminacao de pontos extremos
do {
    for (fall=TRUE,m=0; m<12; m++) {
        for (fthis=TRUE,y=1; y<h-1; y++) {
            pw=water+y*w+1;
            pb=buffer+y*w+1;
            pb1=pb-w;
            pb2=p1+1;
            pb3=p2+w;
            pb4=p3+w;
            pb5=p4-1;
            pb6=p5-1;
            pb7=p6-w;
            pb8=p7-w;
            for (x=1; x<w-1; x++) {
                // Se for ponto extremo
                if ( IsExtreme(*pb,*pb1,*pb2,*pb3,*pb4,*pb5,*pb6,*pb7,*pb8,m) ) {
                    // Sinalizar alteracao
                    fall=fthis=NO;

                    // Eliminar ponto
                    *pw=0;
                }
                pw++;
                pb++;
                pb1++;
                pb2++;
            }
        }
    }
}

```

```

        pb3++;
        pb4++;
        pb5++;
        pb6++;
        pb7++;
        pb8++;
    }
}
// Se houve alteracao nesta passada
if ( !fthis )
    // Fazer copiar de "water" para "buffer"
    for (pw=water,pb=buffer,y=0; y<h; y++)
        for (x = 0; x < w; x++, pw++, pb++)
            *pb=*pw;
}
} while ( !fall );

// Liberar memoria de buffer
MemoryFree(buffer);

// Retornar o buffer da imagem LDA
return water;
}

```

## B.4 Seleção da Costura a partir do Divisor de Águas

```

// Verifica se um pontos e' adjacente a cor "c".
// Isto e', se algum ponto de sua vizinhanca (v1, v3, v5, v7) e'
// da cor "c".
inline int IsNeighborhood(DATA c, // Valor do ponto a ser verificado
                          DATA v1, // Valor do vizinho superior
                          DATA v3, // Valor do vizinho direito
                          DATA v5, // Valor do vizinho inferior
                          DATA v7) { // Valor do vizinho esquerdo

    if ( v1==c || v3==c || v5==c || v7==c ) return 1;
    return 0;
}

// Verifica se um pontos e' adjacente a cor "c".
// Isto e', se algum ponto de sua vizinhanca (v1, v3, v5, v7) e'
// da cor "c". Nao pode haver a presenca de "nc" na vizinhanca.
inline int IsAdjacent(DATA c, // Valor do ponto a ser verificado
                     DATA v1, // Valor do vizinho superior
                     DATA v3, // Valor do vizinho direito
                     DATA v5, // Valor do vizinho inferior
                     DATA v7) { // Valor do vizinho esquerdo
                     DATA nc) { // Cor da adjacencia oposta

    if ( v1==nc || v3==nc || v5==nc || v7==nc )
        return NO;

    if ( (v1==0 && v5==c) || (v1==c && v5==0) )
        return YES;
    if ( (v3==0 && v7==c) || (v3==c && v7==0) )
        return YES;

    return NO;
}

// Verifica se o ponto em questao e' um ponto de LDA quebrada.
// Isto e', se a vizinhanca apresenta as configuracoes abaixo.
// Nao pode haver a presenca de "nc" (outra etiqueta) ou zero
// (buraco) na vizinhanca.
inline int IsBroken(DATA c, // Valor do ponto a ser verificado (ponto de LDA)
                   DATA vb1, // Valor do vizinho superior de "c"

```

```

        DATA vb3, // Valor do vizinho direito de "c"
        DATA vb5, // Valor do vizinho inferior de "c"
        DATA vb7, // Valor do vizinho esquerdo de "c"
        DATA vi,  // Valor numerico do ponto "c" de LDA
        DATA vi1, // Valor do vizinho superior de "vi"
        DATA vi3, // Valor do vizinho direito de "vi"
        DATA vi5, // Valor do vizinho inferior de "vi"
        DATA vi7, // Valor do vizinho esquerdo de "vi"
        DATA nc) { // Cor da adjacencia oposta

if ( vb1==nc || vb3==nc || vb5==nc || vb7==nc ) return NO;
if ( vb1==0 || vb3==0 || vb5==0 || vb7==0 ) return NO;

if ( vb1==1 || vb3==1 || vb5==1 || vb7==1 ) {
// Caso especial : evitar que pontos menores sejam eliminados

if ( vb1==1 && vb5==c && vi>vi1 )
return YES; // "i" e "i1" sao eliminaveis

if ( vb1==c && vb5==1 && vi>vi5 )
return YES; // "i" e "i5" sao eliminaveis

if ( vb3==1 && vb7==c && vi>vi3 )
return YES; // "i" e "i3" sao eliminaveis

if ( vb3==c && vb7==1 && vi>vi7 )
return YES; // "i" e "i7" sao eliminaveis
}
else {
if ( vb1==c && vb5==c )
return YES;
if ( vb3==c && vb7==c )
return YES;
}
return 0;
}

// Eliminar uma linha de costura
void DelSeam(DATA POINTER imi, // Imagem de intersecao
            DATA POINTER imb, // Imagem LDA
            LONGINT w, // Numero de colunas da imagem
            LONGINT h, // Numero de linhas da imagem
            SHORTINT *go) { // Controle de estabilizacao do algoritmo

// Ponteiro movel para imagem de intersecao
DATA POINTER pi, POINTER pi1, POINTER pi3, POINTER pi5, POINTER pi7,

// Ponteiro movel para imagem de intersecao
POINTER pb, POINTER pb1, POINTER pb3, POINTER pb5, POINTER pb7;

// Coluna e linha durante a varredura
LONGINT x, y;

// Valor maximo de pontos de LDA
DATA max,

// Label da "forma" e "fundo"
l1, l2;

// Valor maximo dos lados adjacentes a LABEL1 e LABEL2, respectivamente.
SHORTINT max1,max2;

// Histogramas dos pontos de LDA adjacentes a LABEL1 e LABEL2, respectivamente.
DATA hist1[256], hist2[256];

// Iniciar histograma
for (int i=0;i<256;i++) {
hist1[i]=0;

```

```

    hist2[1]=0;
}
max1=max2=-1;

// Procurar o maximo nas fases das etiquetas LABEL1 e LABEL2
for (y=1; y<h-1; y++) {
    pi=imi+y*w+1;
    pb=imb+y*w+1;
    pb1=pb-w;
    pb3=pb+1;
    pb5=pb+w;
    pb7=pb-1;
    for (x=1; x<w-1; x++,pi++,pb++,pb1++,pb3++,pb5++,pb7++)
        // E' ponto de LDA
        if ( *pb==1 )
            if ( IsAdjacente(LABEL1,*pb1,*pb3,*pb5,*pb7,LABEL2) ) {
                max1=MAX(max1,(int)*pi);
                hist1[*pi]++;
            }
            else if ( IsAdjacente(LABEL2,*pb1,*pb3,*pb5,*pb7,LABEL1) ) {
                max2=MAX(max2,(int)*pi);
                hist2[*pi]++;
            }
        }
}

// Se nao existe nenhum ponto adjacente as etiquetas LABEL1 ou LABEL2 (final)
if ( max1==-1 && max2==-1 ) return;

// Se a fase de LABEL1 e' maxima
else if ( max1>max2 ) {
    max=(DATA)max1;
    l1=LABEL1;
    l2=LABEL2;
    *go=NO;
}

// Se a fase de LABEL2 e' maxima
else if ( max2>max1 ) {
    max=(DATA)max2;
    l1=LABEL2;
    l2=LABEL1;
    *go=NO;
}

// Se a fase 1 apresenta mais maximos que a fase 2 (opcao arbitraria em
// caso de empate)
else if ( hist1[max1]>=hist2[max2] ) {
    max=(DATA)max1;
    l1=LABEL1;
    l2=LABEL2;
    *go=NO;
}
else {
    max=(DATA)max2;
    l1=LABEL2;
    l2=LABEL1;
    *go=NO;
}

// Marcar os pontos a serem eliminados
for (y=1; y<h-1; y++) {
    pi=imi+y*w+1;
    pb=imb+y*w+1;
    pb1=pb-w;
    pb3=pb+1;
    pb5=pb+w;
    pb7=pb-1;
    for (x=1; x<w-1; x++,pi++,pb++,pb1++,pb3++,pb5++,pb7++)
        if ( *pb==1 && *pi>=max )
            if ( IsAdjacente(l1,*pb1,*pb3,*pb5,*pb7,l2) )

```

```

        *pb=10; // Indicador de eliminacao
    }

    // Eliminar e efetuar dilatacao com funcao de preenchimento
    for (y=1; y<h-1; y++) {
        pb=imb+y*w+1;
        pb1=pb-w;
        pb3=pb+1;
        pb5=pb+w;
        pb7=pb-1;
        for (x=1; x<w-1; x++,pb++,pb1++,pb3++,pb5++,pb7++)
            if ( *pb==10 )
                if ( IsNeighbor(l1,*pb1,*pb3,*pb5,*pb7) )
                    KFloodFill(imb,x,y,w,h,*pb=0,l1);
                else
                    *pb=11;
    }

    // Eliminar pontos residuais de LDA quebradas no passo de eliminacao/dilatacao
    // acima
    for (y=1; y<h-1; y++) {
        pi=imi+y*w+1;
        pi1=pi-w;
        pi3=pi+1;
        pi5=pi+w;
        pi7=pi-1;
        pb=imb+y*w+1;
        pb1=pb-w;
        pb3=pb+1;
        pb5=pb+w;
        pb7=pb-1;
        for (x=1; x<w-1; x++,pi++,pi1++,pi3++,pi5++,pi7++,pb++,pb1++,pb3++,pb5++,pb7++)
            if ( *pb==1 )
                if ( IsBroken(l1,*pb1,*pb3,*pb5,*pb7,*pi,*pi1,*pi3,*pi5,*pi7,l2) )
                    *pb=11;
    }
}

// Selecionar a melhor linha de costura.
void SelectSeam(DATA POINTER imi, // Intersection image
               DATA POINTER imb, // Watershed image
               LONGINT w, // Numero de colunas de "image"
               LONGINT h) { // Numero de linhas de "image"

    SHORTINT go; // controle de finalizacao
    // Seleccionar a linha
    do {
        go=YES;
        // Eliminar linhas
        DelSeam(imi,imb,w,h,&go);
    } while ( !go );
}

```

## Bibliografia

- [Abrams et al. (1985)] M. Abrams, A. Blusson, V. Carrere, T. Nguyen, Y Rabu, *Image Processing Applications for Geologic Mapping*, IBM Journal Res. Devel., Vol 29, March 1985, pp. 177-187.
- [Ahern et al (1988)] F. J. Ahern, R. J. Brown, J. Cihlar, R. Gauthier, J. Murphy, R. A. Neville, P. M. Teillet, *Radiometric Correction of Visible and Infrared Remote Sensing Data at Canada Centre for Remote Sensing*, Remote Sensing Yearbook 1988/89, Taylor & Francis, 1988.
- [Banon-Barrera (1994)] G. J. F. Banon, J. Barrera, *Bases da Morfologia Matemática para Análise de Imagens Binárias*, IX Escola de Computação, 24 a 31 de julho, 1994, Recife, PE, Brasil.
- [Beucher (1982)] Serge Beucher, *Watershed of Functions and Picture Segmentation*, Fontainebleau, 1982, pp. 1928-1931.
- [Beucher (1983)] Serge Beucher, *Extrema of Grey-Tone Functions and Mathematical Morphology*, Fontainebleau, 1983.
- [Bernstein (1974)] Ralph Bernstein, *Final Report-Feasibility of Generating Mosaics Directly from ERTS-1 Digital Data*, IBM Report No. FSC 74-0140. April 30, 1974.
- [Bernstein (1976)] Ralph Bernstein, *Digital Image Processing of Earth Observation Sensor Data*, IBM Journal of Research and Development, January 1976, pp. 40-57
- [Chavez (1977)] Pat Chavez, *Digital Mosaic of the State of Nevada*, USGS EROS Data Center PAO No. E45168CT, 1977.
- [Colwell (1983)] Robert N. Colwell, *Manual of Remote Sensing, Vol. 1, 2ª Edição*, American Society of Photogrammetry, 1983.
- [Dougherty (1993)] E. R. Dougherty, *Mathematical Morphology in Image Processing*, Marcel Dekker, New York, 1993.

- 
- [Foley et al. (1993)] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes, *Computer Graphics - Principles and Practice*, 2nd Edition, Addison-Wesley, 1993.
- [Gonzalez-Woods (1992)] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.
- [Haralick et al. (1987)] Robert M. Haralick, Stanley R. Sternberg, Xinhua Zhuang, *Image Analysis Using Mathematical Morphology*, IEEE Transactions on Pattern Analysis and Machine Intelligence, July 1987, pp. 532-550.
- [Hummer-Miller (1989)] Susanne Hummer-Miller, *A Digital Mosaicking Algorithm Allowing for an Irregular Join Line*, PE&RS, January 1989, pp. 43-47
- [Jensen (1986)] John R. Jensen, *Introductory Digital Image Processing - A Remote Sensing Perspective*, Prentice-Hall, 1986.
- [Lantuejoul-Maisonneuve (1984)] C. Lantuejoul, F. Maisonneuve, *Geodesic Methods in Quantitative Image Analysis*, Pattern Recognition, Vol. 17 No. 2, 1984, pp. 177-187.
- [Milgram (1975)] David L. Milgram, *Computer Methods for Creating Photomosaics*, IEEE Trans. Comp., November 1975, pp. 1113-1119.
- [Milgram (1977)] David L. Milgram, *Adaptive Techniques for Photomosaicking*, IEEE Trans. Comp., November 1977, pp. 1175-1180.
- [Moik (1980)] Johannes G. Moik, *Digital Processing of Remotely Sensed Images*, NASA SP-431. U.S. Government Printing Office, Washington, D.C., 1980.
- [Murai-Okuda (1980)] Shunji Murai, Tsutomu Okuda, *Digital Mosaic of Color Aerial Photographs*, 14th Congress, International Society for Photogrammetry, Hamberg, 1980 pp. 570-578.
- [NASA (1979)] NASA, *Reference Publication 1039, Synthetic Aperture Radar/Landsat MSS Image Registration*, NASA Report RP-1039.

- 
- [Niblack (1981)] Wayne Niblack, *The Control Point Library Building System*, Photogrammetry Engineering and Remote Sensing, December 1981, pp. 1709-1715.
- [NOAA (1988)] U.S. Department of Commerce, National Oceanic and Atmospheric Administration, National Environment Satellite, Data and Information Service, *Data Extraction and Calibration of Tiros-N/NOAA Radiometers - NOAA Technical Memorandum NESS 107 - Rev. 1*, October 1988.
- [Peleg (1981)] Shmuel Peleg, *Elimination of Seams from Photomosaics*, IEEE Conference on Pattern Recognition and Image Processing, 1981, pp. 426-429.
- [Serra (1981)] Jean Serra, *Image Analysis and Mathematical Morphology*, Vol. 1, Academic Press, 1981.
- [Serra (1986)] Jean Serra, *Introduction to Mathematical Morphology*, Computer Vision, Graphics, and Machine Vision, 1986, pp. 283-305.
- [Shiren et al. (1989)] Yang Shiren, Li Li and Gao Peng, *Two-Dimensional Seam-Point Searching in Digital Image Mosaicking*, PE&RS, January 1989, pp. 49-53.
- [Sternberg (1986)] Stanley R. Sternberg, *Grayscale Morphology*, Computer Vision, Graphics, and Machine Vision, 1986, pp. 333-355.
- [Stone et al. (1994)] Thomas A. Stone, Peter Schlesing, Richard A. Houghton, George M. Woodwell, *A Map of the Vegetation of South America Based on Satellite Imagery*, PE&RS, May 1994, pp. 541-551.
- [Thenkabail et al. (1994)] Prasad S. Thenkabail, Andrew D. Ward, John G. Lyon, Carolyn J. Merry, *Thematic Mapper Vegetation Indices for Determining Soybean and Corn Growth Parameters*, PE&RS, April 1994, pp. 437-442.
- [Thiruvengadachari (1981)] S. Thiruvengadachari, *Satellite Sensing of Irrigation Patterns in Semiarid Areas : An Indian Study*, PE&RS, October 1981, pp. 1493-1499.

- 
- [Verdin (1985)] James P. Verdin, *Monitoring Water Quality Conditions in a Large Western Reservoir with Landsat Imagery*, PE&RS, March 1985, pp. 343-353.
- [Zobrist (1978)] Albert L. Zobrist, *Multiple-Frame, Full Resolution Landsat Mosaicking to Standard Map Projection*, Proceedings American Society of Photogrammetry, Fall Technical Meeting, Albuquerque, NM, 1978, pp. 608-616.
- [Zobrist-Bryant (1978)] Albert L. Zobrist, Nevin A. Bryant, *Elements of an Image Based Information System*, Policy Analysis and Information Systems, Vol. 1, No 2, 1978, pp. 71-90.
- [Zobrist-Bryant (1979)] Albert L. Zobrist, Nevin A. Bryant, *Map Characteristics of Landsat Mosaics*, Proceedings American Society of Photogrammetry Annual Convention, Washington, D.C., 1979, pp. 260-273.
- [Zobrist et al. (1983)] Albert L. Zobrist, Nevin A. Bryant, Ronald G. McLeod, Technology for Large Digital Mosaics of Landsat Data, PE&RS, September 1983, pp. 1325-1335.

# Índice Remissivo

## A

abertura	
definição .....	23
exemplo .....	24
afinamento .....	35
definição .....	25
exemplo .....	26
ajuste histográfico .....	7
arestas artificiais .....	5; 8; 10; 12; 36

## C

círculo geodésico .....	30
configurações homotópicas .....	25
correção geométrica .....	60
correção radiométrica .....	58
costura monodimensional .....	11

## D

definição da linha de costura .....	5
descontinuidades .....	5; 7; 11; 45
deteção de contornos .....	42
dilatação	
definição .....	20
exemplo .....	21; 22; 31
extensões .....	21
geodésica .....	30; 40
propriedades .....	22
vista como HMT .....	21
distância geodésica .....	29

## E

efeitos atmosféricos .....	2; 57
absorção .....	57
espalhamento .....	57
efeitos da cena .....	2; 55
projeção .....	55
rotação da Terra .....	55
efeitos da plataforma .....	2; 54
altitude .....	54
atitude .....	54
velocidade .....	55
efeitos do sensor .....	2; 56
falha de detector .....	56
falha no conversor analógico/digital .....	56

efeitos do sensor e da cena .....	2; 56
perspectiva .....	57
vista .....	57
elemento estruturante .....	16
família E .....	28
família L .....	26
família M .....	26
planar .....	22
erosão .....	20
definição .....	20
exemplo .....	21; 22; 31
extensões .....	21
geodésica .....	31
propriedades .....	22
vista como HMT .....	21
erros geométricos .....	2; 53
erros não sistemáticos .....	60
erros radiométricos .....	2; 53
erros sistemáticos .....	60
espessamento	
definição .....	25
exemplo .....	26
extensões .....	27
esqueleto .....	26; 28
espessamento .....	28
geodésico .....	31
homotópico .....	28
por zona de influência .....	28
<b>F</b>	
família de configurações .....	19; 20
fases de um elemento estruturante .....	19
fechamento	
definição .....	23
exemplo .....	24
fontes de erros .....	53
função custo .....	8
função rampa .....	7; 10
<b>H</b>	
HMT .....	19; 21; 25
definição .....	19
<b>I</b>	
idempotência .....	26
índice de correlação .....	45
índice de covariância .....	46

índice de similaridade .....	45
índice de similaridade por média normalizada .....	45
informação geo-referenciada .....	3
<b>L</b>	
LDA .....	48
definição .....	32
definição intuitiva .....	32
exemplo .....	34; 35
implementação por afinamento .....	34
obtenção de linha de costura .....	37
linha de divisor de águas de uma função .....	3
localização de pontos de controle .....	60
<b>M</b>	
mapeamento de espaços .....	61
marcador .....	30, 40
modelagem das distorções geométricas .....	60
modificação homotópica .....	42
<b>O</b>	
operações morfológicas .....	16; 23
<b>P</b>	
percepção visual .....	5
photomosaicking .....	1; 3; 4; 25; 37
processo de correção geométrica .....	60
propagação .....	41; 42
<b>S</b>	
sensoriamento remoto .....	1
similaridade .....	8
<b>SKIZ</b>	
definição .....	27
exemplo .....	32
supersegmentação .....	40; 42
<b>T</b>	
técnicas nominais de correção radiométrica .....	59
técnicas suplementares de correção radiométrica .....	59
transformações clássicas .....	18
exemplo .....	18
transformações geodésicas .....	29
transformações morfológicas .....	18; 25
<b>V</b>	
vizinhança mono-dimensional .....	7