



CCPgEE/CCT-UFPB

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
CENTRO DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE FEDERAL DA PARAÍBA

DISSERTAÇÃO DE MESTRADO
IMPLEMENTAÇÃO DE UM SISTEMA CONTEXTUAL DE
CLASSIFICAÇÃO DE IMAGENS ORBITAIS
COMPATÍVEL COM O SISTEMA SITIM

RONEI MARCOS DE MORAES
1992

Implementação de um Sistema Contextual de
Classificação de Imagens Orbitais
compatível com o sistema SITIM

Ronei Marcos de Moraes

Implementação de um Sistema Contextual de
Classificação de Imagens Orbitais
compatível com o sistema SITIM

Dissertação apresentada ao Curso de
MESTRADO EM ENGENHARIA ELÉTRICA, da
Universidade Federal da Paraíba, em
cumprimento às exigências para obtenção
do Grau de Mestre.

ÁREA DE CONCENTRAÇÃO : PROCESSAMENTO DE IMAGENS

João Marques de Carvalho, PhD.

(Orientador)

CAMPINA GRANDE

JANEIRO - 1992

1. Bibliografia
2. Classificação de imagens
3. Sistema de imagens
4. Sistema de imagens



M827i

Moraes, Ronei Marcos de

Implementacao de um sistema contextual de classificacao de imagens orbitais compativel com o sistema SITIM / Ronei Marcos de Moraes.- Campina Grande, 1992.

100 f. : il.

Dissertacao (Mestrado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Engenharia Eletrica 2. Classificacao de Imagens - Sistema Contextual 3. Processamento de Imagens 4. Sistema SITIM 5. Dissertacao I. Carvalho, Joao Marques de, PhD. II. Universidade Federal da Paraiba - Campina Grande (PB) III. Título

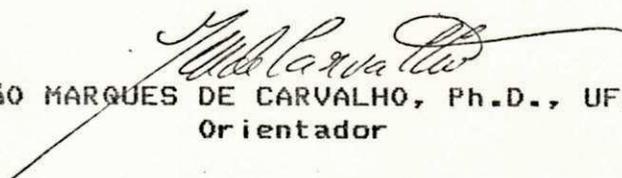
CDU 621.3(043)

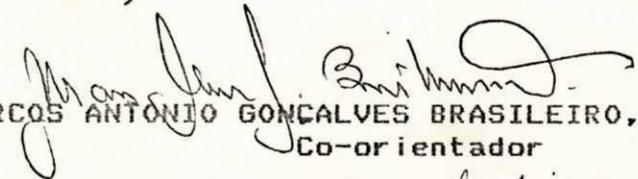
À minha família pelo apoio e
paciência irrestritos durante a
realização deste trabalho.

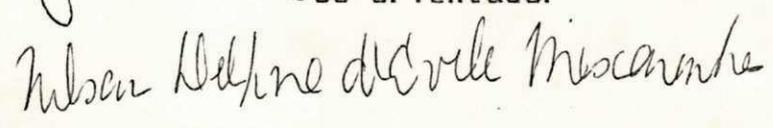
IMPLEMENTAÇÃO DE UM SISTEMA CONTEXTUAL DE CLASSIFICAÇÃO
DE IMAGENS ORBITAIS COMPATIVEL COM O SISTEMA SITIM

RONEI MARCOS DE MORAES

DISSERTAÇÃO APROVADA EM 27.01.92


JOÃO MARQUES DE CARVALHO, Ph.D., UFPB
Orientador


MARCOS ANTONIO GONCALVES BRASILEIRO, D.Sc., UFPB
Co-orientador


NELSON DELFINO D'AVILA MASCARENHAS, Ph.D., INPE
Componente da Banca


JUAN CARLOS CEBALOS, Dr., UFPB
Componente da Banca

CAMPINA GRANDE - PB
JANEIRO - 1992

Agradecimentos

A Capes e CNPq pelo apoio financeiro, sem o qual seria impossibilitada a realização deste empreendimento.

A João Marques de Carvalho e Marcos A. G. Brasileiro pelas orientações a mim ministradas.

Ao professor e pesquisador do INPE, Nelson F.D.A. Mascarenhas, pela cooperação e assistência, encurtando caminhos.

Ao Departamento de Engenharia Elétrica - UFPb pela confiança em mim depositada.

Ao Departamento de Estatística - UFPb pelo apoio e credibilidade irrestritos a este empreendimento.

Ao Departamento de Processamento de Imagens - INPE e ao CEPAGRI - UNICAMP pela cessão dos seus equipamentos e infraestrutura necessários a esta empreitada.

Aos amigos: Alfredo Jackson mestrando em Informática UFPb, Fernando Augusto e Ana Paula, pesquisadores do INPE, Laércio e Vanildes da Engespaço pela grande ajuda nos seus esclarecimentos na área de informática.

À amiga Claudia Regina Oliveira de Paiva Lima, do DE/UFPb, pelo apoio e incentivo nas horas difíceis.

À amiga Rosana da Silva pelo enorme apoio.

A Virgínia Ragoni M. Correia, pesquisadora do INPE, pelo apoio e esclarecimentos preciosos, sem os quais seria impossível a realização deste trabalho.

S U M Á R I O

Os sistemas contextuais de classificação são conhecidos na literatura especializada pela sua grande precisão e lentidão devido ao fato de possuírem um grande número de cálculos a serem feitos a cada iteração. Por este motivo, raras são as implementações práticas ou comerciais deste tipo de algoritmo em todo mundo. A nível de Brasil, o único sistema de processamento de imagens comercial nacional, também não conta com nenhuma implementação de um método contextual de classificação.

Com o intuito de sanar esta falta de um método de classificação de maior precisão no sistema SITIM desenvolvido pelo INPE e comercializado pela ENGESPACO de São José dos Campos - São Paulo e, verificar a veracidade da fama do método contextual de precisão e de ser um modelo muito lento em termos de processamento, foi estudado, aperfeiçoado e implementado um método de classificação bayesiano contextual.

São apresentados os resultados desta implementação e o sistema é testado em comparação com um método de classificação estatística de máxima verossimilhança já implementado no sistema SITIM.

A B S T R A C T

Systems for contextual classification of images are known in the specialized literature for their good accuracy of results and slowness of computation, due to the fact that a large number of computations is needed at each interaction. For this reason, very few attempts have been made to produce practical or comercial implementations for this class of algorithms. In Brazil, the only national image processing systems comercially avaiable does not offer a method of contextual image classification to their users.

With the intention to fill this lack of a high precision classification method at the SITIM system - developed by INPE and comercialized by ENGESPACO of São José dos Campos - São Paulo and, to verify the feasibility of a practical implementation of contextual algorithms, a bayesian classification method was studied, improved and implemented in this work.

This implementation results are presented and the system is tested in compariscn with a statistical method of classification by maximum likelihood, already implemented at the SITIM system.

ÍNDICE

Introdução	1
I.1.Motivação do trabalho	1
I.2.Organização do trabalho	2
CAPÍTULO 1. Introdução ao Sensoriamento Remoto	4
1.1.Introdução	4
1.2.As Imagens	5
1.2.1.Problemas na Detecção	7
1.3.O Básico do Processamento Digital de Imagens (P.D.I.) em Sensoriamento Remoto	9
1.3.1.Recursos do P.D.I.	10
1.3.1.a.Restauração	11
1.3.1.b.Realce de Contraste	11
1.3.1.c.Incorporação de Informações	12
1.3.2.Tratamento de Imagens	12
1.3.3.Classificação	16
1.3.3.a.Método Supervisionado Clássico	18
1.3.3.b.Método Supervisionado Contextual	19
1.3.3.c.Métodos de Relaxação	23
1.3.3.c.Estimativa de Parâmetros - Treinamento do Classificador	24
1.4.Interpretação Visual de Imagens	26
1.5.Testes e Controle	29

CAPÍTULO 2. O Método Contextual	30
2.1.Introdução	30
2.2.O Modelo Contextual p,q,r	33
2.2.1.O Modelo Owen & Switzer Generalizado	37
2.2.2.Estimativa de Parâmetros	42
2.2.2.a.Densidades das classes	42
2.2.2.b.Probabilidades à priori e Parâmetros	
Contextuais (caso de quatro vizinhos)	44
2.2.3.O limiar (1-e)	47
 CAPÍTULO 3. Delineamento e Implementação do Sistema	 53
3.1.Introdução	53
3.2.Descrição do Sistema	53
3.3.Implementação	61
 CAPÍTULO 4. Testes e Avaliação de Desempenho	 72
4.1.Introdução	72
4.2.Testes	72
4.3.Avaliação de desempenho	77
4.3.1.O Método da Máxima Verossimilhança (MAXVER)	78
4.3.2.Método do Paralelepípedo	81
4.3.3.Método do Fatiamento de Histograma	81
4.3.4.Método do Fatiamento de Temperaturas	81
4.3.5.Método da Distância Euclidiana	82
4.4.Testes comparativos com o MAXVER	83

CAPÍTULO 5. Conclusões Finais e Sugestões	89
5.1. Introdução	89
5.2. Conclusões	89
5.3. Sugestões para trabalhos futuros	90

APÊNDICE A - LISTAGEM PROGRAMA MPQR

APÊNDICE B - LISTAGEM PROGRAMA FUNCAOE

APÊNDICE C - LISTAGEM PROGRAMA MXVCON

APÊNDICE D - LISTAGEM SUBROTINA CONTEXT

ÍNDICE BIBLIOGRÁFICO

ÍNDICE ANALÍTICO

ÍNDICE DE EQUAÇÕES

CAPÍTULO 1. Introdução ao Sensoriamento Remoto	4
(1.1) Critério de entropia	14
(1.2) Critério de entropia para X distribuído segundo uma normal multivariada	14
(1.3) Critério de Componentes Principais aplicado a entropia	14
(1.4) Princípio de Máxima Verossimilhança	15
(1.5) Equação de $L'_{ij}(\tilde{X})$	15
(1.6) Divergência.....	15
(1.7) Divergência média.....	16
(1.8) Distância J-M.....	16
(1.9) Risco Médio.....	18
(1.10) Minimização do risco médio.....	19
(1.11) Probabilidade a posteriori da classe k.....	19
(1.12) Regra bayesiana de classificação clássica.....	19
(1.13) Probabilidade de classificação correta.....	19
(1.14) Espaço das Decisões.....	20
(1.15) Vetor característico.....	20
(1.16) Regra de classificação Bayesiana contextual.....	20
(1.17a) Probabilidade a posteriori da classe k.....	21
(1.17b) Fator de correção contextual.....	21

CAPÍTULO 2. O Método Contextual.....	30
(2.1) Vizinhaça Δ_i composta por cinco pixels em cruz.....	34
(2.2) Configuração particular de classes em cruz.....	34
(2.3) Probabilidades $g(\cdot \setminus k)$	34
(2.4) Densidade conjunta dos vetores característicos em X_i , dado as classes na cruz e X_i	35
(2.5) Função-Perda $L(C_i, \hat{C}_i)$	35
(2.6) Função-Perda média.....	35
(2.7) Perda esperada.....	35
(2.8) Regra de Bayes ótima de classificação contextual....	36
(2.9) Probabilidade $P(k \setminus \Delta_i)$	36
(2.10) Probabilidade $P(k \setminus \Delta_i)$ reescrita.....	36
(2.11) Densidade h condicionalmente independente dado as classes.....	37
(2.12) Fator de correção contextual.....	37
(2.13) Obtenção da função g para os padrões X, L e T	39
(2.14) Função g para os padrões X, L e T	39
(2.15) Procedimento de classificação contextual.....	40
(2.16) Obtenção da Função de correção contextual.....	41
(2.17) Função de correção contextual resumida.....	41
(2.18) Função de correção contextual expandida.....	41
(2.19) Função a e função b	41
(2.20) Densidade Normal multivariada.....	42
(2.21) Expressão analítica da densidade normal multivariada	42
(2.22) Modelos possíveis para a densidade	

normal multivariada a ser adotada.....	43
(2.23) Estimador para o vetor de média μ_k	43
(2.24) Estimador para a matriz de covariâncias, caso (iii)	43
(2.25) Estimador para a matriz de covariâncias, caso (i)..	44
(2.26) Função Indicadora.....	45
(2.27) Probabilidade à priori.....	45
(2.28) Estimador de M_x/M	46
(2.29) Estimador de M_l/M	46
(2.30) Estimador de M_I/M	46
(2.31) Estimadores de p, q, r	46
(2.32) Corte efetivo $(1-e_k)$	50
CAPÍTULO 3. Delineamento e Implementação do Sistema.....	53
(3.1) Probabilidade a priori da classe k	54
(3.2) Teste iterativo da estimativa de p, q, r	55
(3.3) Probabilidade a posteriori da classe k	57
(3.4) Probabilidade a posteriori do MAXVER.....	65
(3.5) Densidade Normal n -variada.....	65
(3.6) Equação da Posteriori do MAXVER.....	65
(3.7) Posteriori pela Máxima Verossimilhança.....	65
(3.8) Posteriori (4.3) reescrita.....	66
(3.9) Distância de Mahalanobis.....	66
(3.10) Distância redefinida pela posteriori.....	66
(3.11) Distância Euclidiana.....	66
(3.12) Primeira transformação no espaço dos atributos.....	66
(3.13) Segunda transformação no espaço dos atributos.....	66

(3.14) Distância Euclidiana com média centrada na origem..	67
(3.15) Distância (3.9) na forma final no MAXVER.....	67
(3.16) Probabilidade a posteriori contextual.....	68
(3.17) Adição do logaritmo à equação (3.15).....	68
(3.18) Equação (3.16) reescrita.....	69
(3.19) Definição de cologaritmo de x.....	69
(3.20) Equação (3.17) a menos do último membro.....	69
(3.21) Equação de classificação final.....	69
CAPÍTULO 4. Testes e Avaliação de Desempenho.....	72
(4.1) Maximização de probabilidade no MAXVER.....	79
(4.2) Estimador do vetor de médias.....	79
(4.3) Estimador da matriz de covariâncias.....	79
(4.4) Matriz de Transformação.....	80
(4.5) Matriz de Autocorrelação.....	80
(4.6) Critério da Distância Euclidiana.....	82
CAPÍTULO 5 Conclusões e Sugestões.....	89

não há equações neste capítulo

ÍNDICE DE FIGURAS

CAPÍTULO 1. Introdução ao Sensoriamento Remoto.....	4
1.1 Espectro eletromagnético de energia.....	5
1.2 Obtenção da imagem pelo satélite MSS LANDSAT.....	6
1.3 Obtenção de um pixel.....	7
1.4 Assinaturas espectrais de uma área com solo e vegetação combinadas a 50%, não se contando os efeitos atmosféricos.....	8
1.5 Formação da imagem.....	9
1.6 Fluxograma do P.D.I. em Sensoriamento Remoto.....	10
1.7 Definição da razão de verossimilhança no ponto.....	15
1.8 Modelo Discriminante de Fisher.....	17
CAPÍTULO 2. O Método Contextual.....	30
2.1 Vizinhança Δ_1 composta por 5 pixels em cruz.....	34
2.2 Padrões X , L e T	38
2.3 Correspondência e localização dos pixels na cruz.....	45
2.4 Corte efetivo da distribuição dado pela transformação no plano.....	49
2.5 (a) Região de corte $\alpha\%$ multivariado.....	49
(b) Curvas de nível para região rejeitada, caso multivariado.....	49
2.6 Corte efetivo da distribuição dado pela transformação no espaço.....	50

2.7 Fluxograma do Método Contextual p,q,r implementado....	52
CAPÍTULO 3. Delineamento e Implementação do Sistema.....	53
3.1 Modificação na estrutura do fluxograma para a estimativa iterativa de p,q,r.	56
3.2 Planos de imagem e a cruz para cada plano.....	59
3.3 Exemplificação de "buffer" rotativo.....	60
3.4 Fluxograma do sistema contextual implementado.....	62
3.5 Transformações no espaço dos atributos.....	67
CAPÍTULO 4. Testes e Avaliação de Desempenho	72
4.1 Imagem utilizada	73
4.2 Imagem da coleta de amostras homogêneas	74
4.3 Imagem classificada pelo MXVCON no SITIM 150	76
4.4 Imagem classificada pelo MXVCON no SITIM 340	77
4.5 Imagem classificada pelo MAXVER no SITIM 150	83
4.6 Imagem classificada pelo MAXVER no SITIM 340	85
4.7 Diferença entre as imagens geradas pelo MXVCON e pelo MAXVER	87
CAPÍTULO 5. Conclusões e Sugestões.....	89
não há figuras neste capítulo	

ÍNDICE DE TABELAS

CAPÍTULO 1. Introdução ao Sensoriamento Remoto	4
(1.1) Áreas de aplicação do Sensoriamento Remoto	4
(1.2) Métodos de classificação e metodologia de treinamento	25
CAPÍTULO 2. O Método Contextual	30
não há tabelas neste capítulo	
CAPÍTULO 3. Delineamento e Implementação do Sistema	53
(3.1) Sumário das rotinas do Método Contextual implementado	71
CAPÍTULO 4. Testes e Avaliação de Desempenho	72
(4.1) Número de amostras e pontos amostrados	74
(4.2) Matriz de Classificação	75
(4.3) Quadro remissivo de desempenho	77
(4.4) Área classificada pelo MAXVER - SITIM 150	84
(4.5) Área classificada pelo MXVCON - SITIM 150	84
(4.6) Área classificada pelo MAXVER - SITIM 340	85
(4.7) Área classificada pelo MXVCON - SITIM 340	86
(4.8) Tempos de processamento por sistema, dos programas MAXVER e MXVCON	86

CAPÍTULO 5. Conclusões e Sugestões 89

não há tabelas neste capítulo

Introdução

I.1. Motivação do trabalho

As técnicas de classificação de imagens são utilizadas desde longa data, quando o homem pôde digitalizar uma imagem e armazená-la sob a forma numérica. A partir de então, uma grande variedade de técnicas foram propostas com o objetivo de cada vez melhor reconhecer padrões e classificá-los em imagens digitalizadas. Desta gama de técnicas, um grande número dedicou-se à classificação de imagens multiespectrais de Sensoriamento Remoto (referências de número [1-34]). Nesta área, as técnicas estatísticas de classificação ganharam destaque pela sua precisão e confiabilidade [1], [3], [4], [6-20], [22], [23], [25-31], [33] e [34].

Este trabalho estuda um grupo restrito de classificadores estatísticos denominados *classificadores contextuais* [3], [7-13], [15-21], [31] e [33]. Seu nome deve-se ao fato de que para classificarem um ponto da imagem, levam em consideração o contexto onde este ponto está inserido. Este tipo de classificador surgiu em fins da década de 60 e por questões de lentidão pelo número de operações aritméticas envolvido, perdeu-se o interesse em sua utilização apesar da sua precisão superior. Atualmente, os avanços de "hardware" e "software" estão pondo fim a esta problemática do tempo de processamento e o reaparecimento deste classificador, de

uma forma viável está próxima. Acreditando nesta premissa, esta dissertação apresenta uma implementação viável e otimizada do classificador contextual bayesiano de Owen and Switzer [10], aprimorado por um trabalho de Hjort [17] para imagens multiespectrais de Sensoriamento Remoto.

O sistema foi desenvolvido compatível com a família SITIM - Sistema de Tratamento de Imagens - (baseado em microcomputador da família PC) de fabricação nacional e com grande difusão entre as instituições de pesquisa do país. O sistema utiliza o caso de quatro vizinhos - 4-NN - em forma de cruz, somado ao ponto em questão.

I.2. Organização do trabalho

Este trabalho está dividido em cinco capítulos, a saber: capítulo 1 - Introdução ao Sensoriamento Remoto; capítulo 2 - O Método Contextual; capítulo 3 - Delineamento e Implementação do sistema; capítulo 4 - Testes e Avaliação de Desempenho e capítulo 5 - Conclusões Finais e Sugestões.

No capítulo 1, é dada uma visão geral do Sensoriamento Remoto, passo a passo. Foi entendido que com esta detalhada introdução ao Sensoriamento Remoto, o restante do trabalho ficaria de mais fácil compreensão. Uma visão bibliográfica do assunto é realizada e o método de classificação contextual, de forma genérica, é introduzido e situado com respeito ao Sensoriamento Remoto.

O capítulo 2 aprofunda, com rigor estatístico conveniente, o tópic método contextual e analisa pormenorizadamente o método implementado.

No capítulo 3, é mostrado detalhadamente as modificações introduzidas visando aperfeiçoar e reduzir o número de cálculos do método para torná-lo viável.

No capítulo 4, os testes do sistema implementado são apresentados e também uma breve análise comparativa com o sistema de Máxima Verossimilhança, já implementado no sistema SITIM, por ser um método de estrutura estatística próxima ao sistema implementado.

O capítulo 5 finaliza a dissertação apresentando as conclusões finais e deixa sugestões para trabalhos futuros na área.

CAPÍTULO 1

Introdução ao Sensoriamento Remoto

1.1. Introdução

Dá-se o nome de Sensoriamento Remoto a um conjunto de técnicas que permitem a exploração de um objeto sem contato físico. Este "objeto" pode ser uma área geográfica, regiões do corpo humano - em aplicações médicas - entre outros. A exploração é feita através de sensores, como câmaras fotográficas, de televisão, etc.

Na tabela 1.1 são sumarizadas várias áreas de aplicação do Sensoriamento Remoto:

Á r e a	A p l i c a ç ã o
Meteorologia	Previsão do tempo a um determinado prazo, Mapeamento climático, etc.
Geologia	Procura de jazidas minerais (grande utilização), estudos de aproveitamento de solos, etc.
Agricultura	Previsão de safras, estudos de contaminação por pragas, etc.
Militar	Espionagem, teleguiagem de mísseis, controle de tráfego aéreo e marítimo, etc.
Indústria	Inventário e projeções de recursos hídricos pesca e salinas.
Ecologia	Pesquisas sobre o equilíbrio ecológico no planeta.
Demografia	Inventário e planejamento para controle do aumento demográfico, cidades, etc.

Tabela 1.1 - Algumas áreas de aplicação do Sensoriamento Remoto.

1.2. As Imagens

As imagens são registros de cenas focalizadas por sensores com resposta em frequência perfeitamente definidas do espectro. No entanto, os sensores não conseguem captar todo o espectro e registram faixas de frequência. Analogamente, o olho humano capta uma faixa muito estreita do espectro, figura 1.1, na maioria das vezes mais estreita que os sensores. Várias técnicas codificam as frequências do espectro não-visível de modo a tornar possível a visualização de uma imagem.

CURVAS DE RADIAÇÃO DO CORPO NEGRO E RADIAÇÃO SOLAR

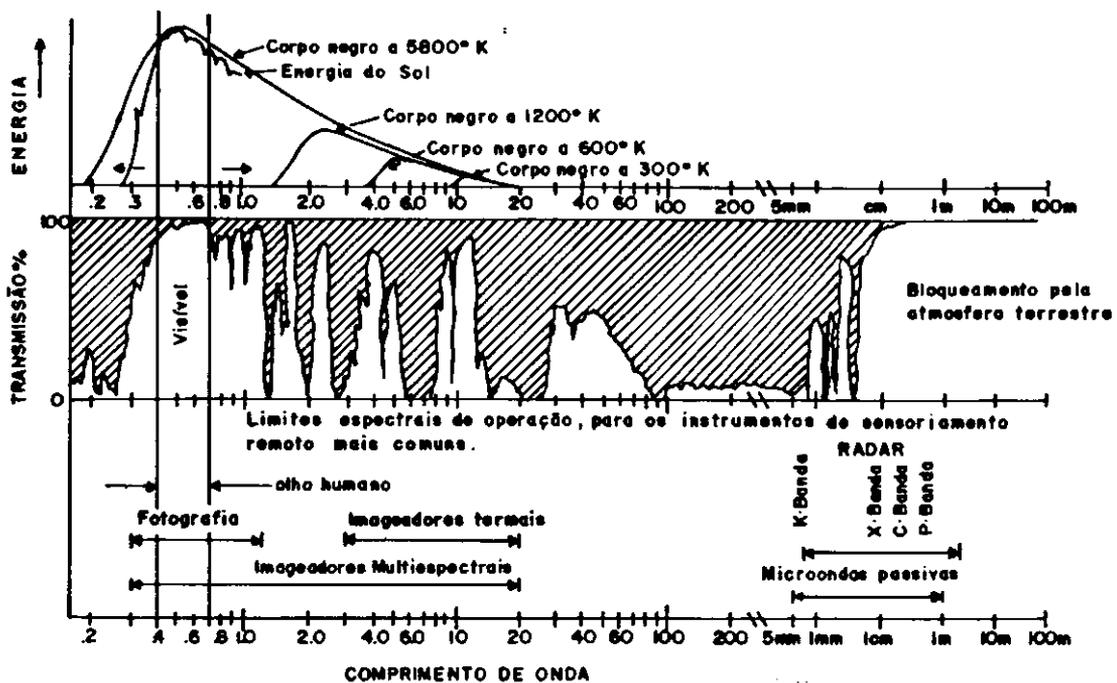


Figura 1.1 - Espectro eletromagnético.

Dois tipos de sensores podem ser utilizados na obtenção de uma imagem; são eles: ativos - transmitem pulsos de energia e medem a

energia refletida desses pulsos pelos objetos; passivos - medem radiações refletidas ou emitidas pelos objetos. Neste trabalho de dissertação, usar-se-á imagens geradas por satélites de sensoriamento remoto do tipo comercial, ou seja, munidos de sensores do tipo passivo.

A caráter de ilustração da obtenção de uma imagem por sensores passivos, será mostrada a forma de coleta dos dados dos satélites LANDSAT MSS 1 e 2 [35]. O satélite desloca-se no sentido Norte-Sul, enquanto um espelho a bordo imageia a superfície terrestre de Oeste para Leste em 33 milissegundos. São tomadas 3300 amostras em intervalos regulares de 9,95 microssegundos, ao longo de uma varredura de 185,2 km - figura 1.2. Nesta varredura, são imageadas seis linhas de dados por banda espectral, a cada vez.

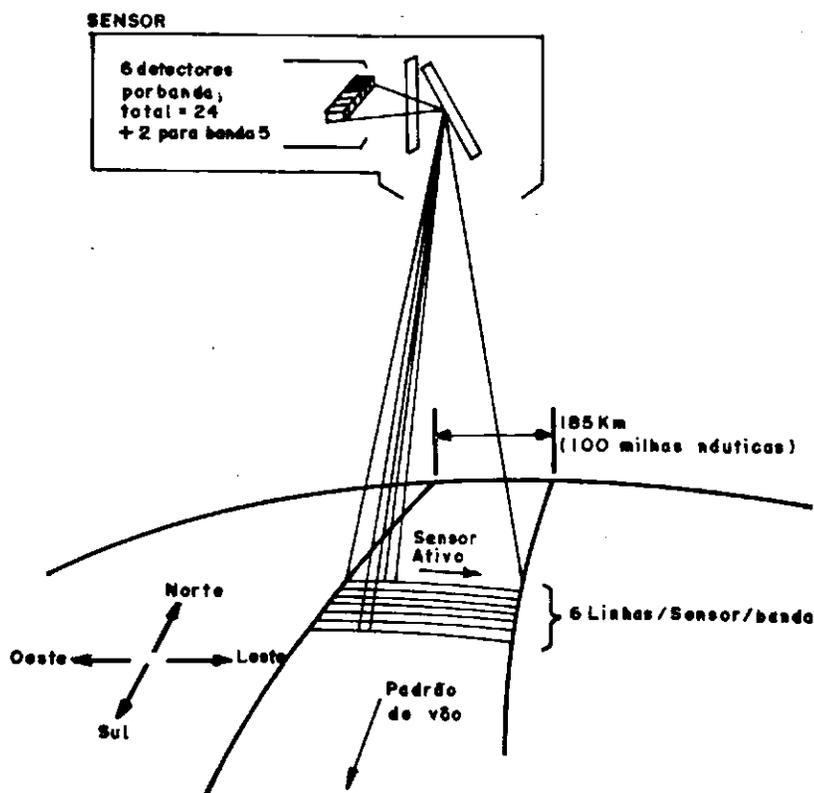


Figura 1.2 - Obtenção da imagem pelo LANDSAT.

O Campo de Visão Instantânea - VIC - cobre uma área de 79m x 79m, ou seja uma área de 6241 m² [36]. Entre cada amostragem, os sensores movem-se 56m, o que resulta um elemento de imagem ou "pixel" correspondendo a uma área de 56m x 79m, ou 4424 m². A esta redução de área dá-se o nome de *Recobrimento* - figura 1.3. Apesar dos dados serem tomados de uma área de 6241 m², para evitar distorções, os dados são arranjados como se tivessem sido tomados de uma área de 4424 m².

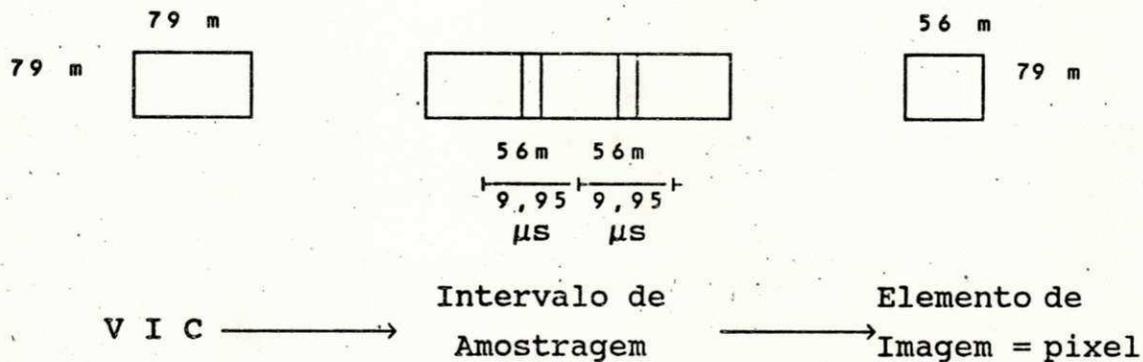


Figura 1.3 - Obtenção de um pixel.

1.2.1. Problemas na Detecção

Existem vários problemas na detecção de um objeto pelos sensores. Por exemplo, se o objeto focado tiver um tamanho menor que o VIC, ele não será detectado, a menos que o seu valor de brilho - VB, ou índice de luminosidade - se sobreponha ao da sua circunvizinhança. Se o objeto tiver um tamanho exatamente igual ao do VIC, também não será detectado, a menos que o VIC se sobreponha exatamente sobre ele e o seu VB seja maior que o da circunvizinhança.

Outra questão a ser considerada é o que se chama "confusão espectral". Um objeto qualquer tem uma curva espectral definida, porém esta curva se altera quando o objeto é imageado na presença

de outro - figura 1.4.

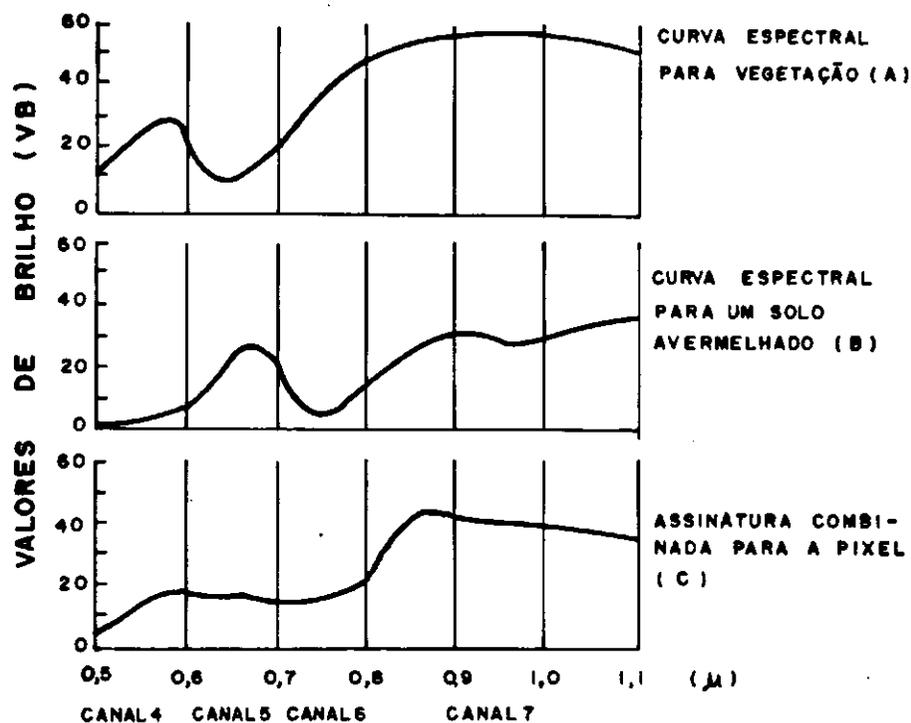


Figura 1.4 - "Confusão" espectral.

Os sensores do tipo passivo sentem muito o efeito atmosférico de duas formas: dispersão e absorção. Enquanto a dispersão acrescenta luminosidade, a absorção retira. A atmosfera dispersa o comprimento de onda verde [0,5 μ m;0,6 μ m] em média sete vezes mais que o comprimento de onda infravermelho [0,7 μ m;1,1 μ m]. A absorção afeta principalmente a faixa infravermelha do espectro. É possível fazer correções para a dispersão, mas é muito difícil fazê-lo para a absorção. Um outro problema grave é a qualidade da luz refletida que depende da radiação incidente. A assinatura espectral muda muito se o dia tem nuvens ou não. Outra questão é que dois objetos distintos podem ter diferenças espectrais desprezíveis na zona

multiespectral, ou seja, em várias bandas espectrais. A poluição pode distorcer a radiação dentro das chamadas "janelas atmosféricas" - intervalo do espectro que não é absorvido.

Assim, é importante ressaltar que a escolha correta dos canais a serem observados pode contornar a maioria destes problemas.

1.3. O Básico do Processamento Digital de Imagens (P.D.I.) em Sensoriamento Remoto

Didaticamente, pode-se pensar na imagem como uma matriz de pequenos quadrados, onde cada elemento - pixel - contém um valor de sombreamento proporcional a reflectância da imagem na respectiva posição.

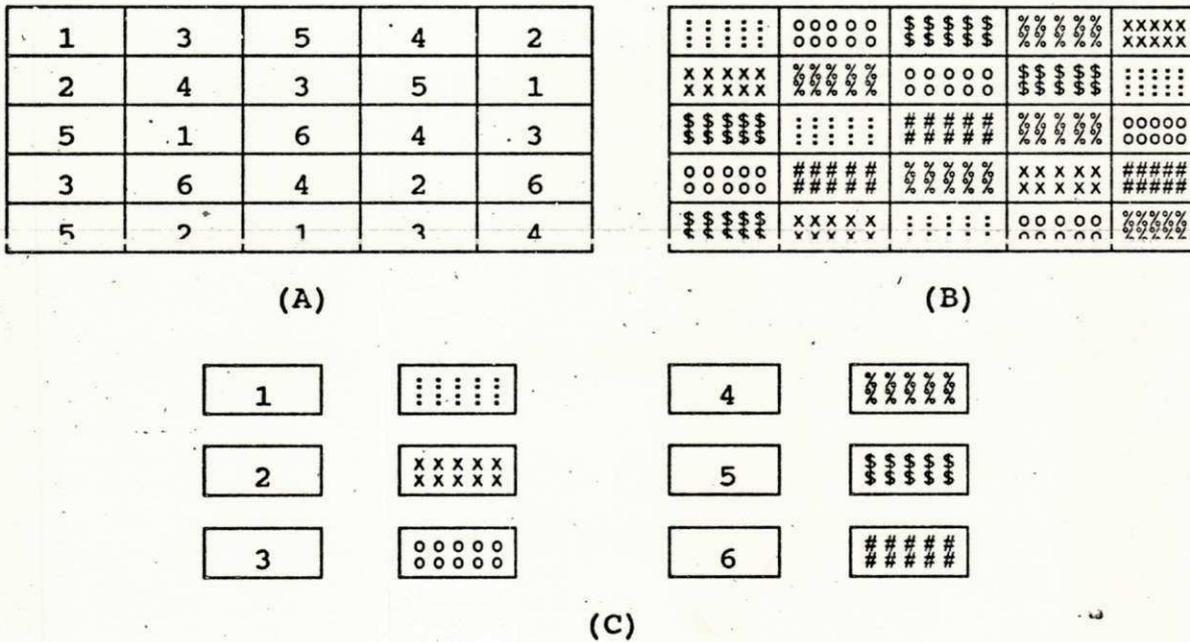


Figura 1.5 - Codificação (C) da matriz (A) resulta em uma imagem (B).

Sinteticamente, na geração da imagem, para cada valor atribui-se uma cor (quando o equipamento é colorido), ou níveis de cinza (quando o equipamento é preto e branco). Pode-se pensar no

campo de P.D.I., segundo o fluxograma abaixo:

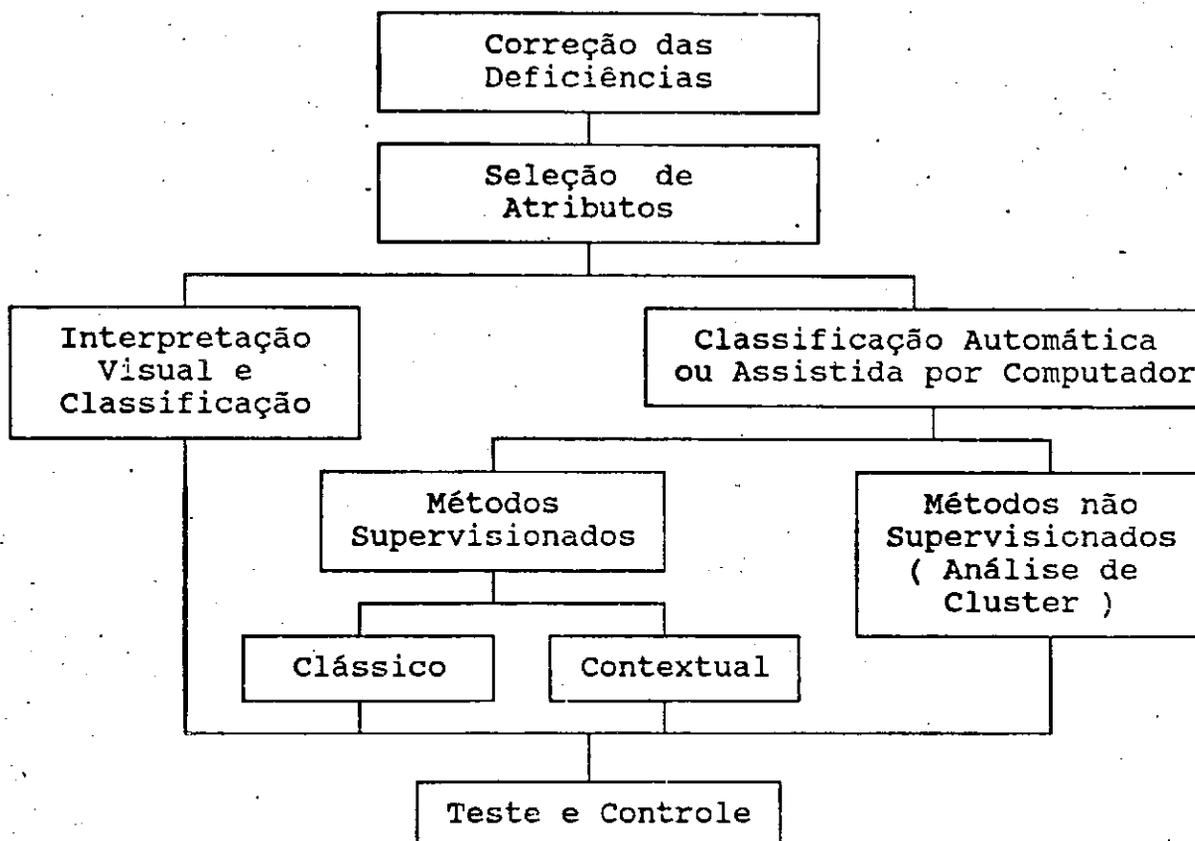


Figura 1.6 - Fluxograma do P.D.I. em Sensoriamento Remoto

De acordo com este fluxograma, será feita a seguir uma análise sucinta de cada um dos seus pontos, dando ao leitor além de uma visão global do processo, alguns detalhes do mesmo.

1.3.1. Recursos do P.D.I.

Existem recursos para modificar ou restaurar uma imagem dentro da idéia de correção de deficiências. Esses recursos são de grande valia ao usuário. A seguir será dada uma breve introdução a estes recursos, o que objetiva cada um deles e a necessidade da sua utilização.

1.3.1.a. Restauração e Eliminação do Ruído

A restauração da imagem ou eliminação do "ruído", pode ser necessária para as imagens obtidas próximas ao limite de resolução dos seus sensores, como em Astronomia, Microscopia Eletrônica, Sensoriamento Remoto Militar, etc. Nestes casos, onde a resolução é bem alta, ou sob condições adversas, como movimento rápido relativo objeto-sensor; ou turbulência, pode-se notar na imagem distorções, borramentos e outros tipos de imperfeições. A restauração, como o próprio nome já diz, consiste no uso de um conjunto de complexas técnicas obtidas a partir de modelos estatísticos, matemáticos ou físicos, que constituem funções de transferência para os pontos-problema, ou até toda a imagem, num caso mais complexo.

Comumente, em casos onde a imagem captada está contaminada por ruído aleatório, são utilizados processos de filtragem na imagem. Isso faz com que o "ruído" presente na imagem seja removido. A implementação destes filtros pode ser feita por "software" [37] - que também tece várias comparações entre diversos filtros - ou por "hardware" [38] - que mostra em detalhes a implementação de um filtro por gradiente de Roberts.

1.3.1.b. Realce de Contraste

Basicamente, a necessidade de se realçar contrastes em uma imagem deve-se ao fato da visão humana poder distinguir um número finito de níveis de sombreamento, que é muito inferior ao que um monitor de computador pode oferecer. Assim, esta limitação pode causar muita confusão à vista humana quando os níveis de

sombreamento são muito próximos. As metodologias aplicadas permitem modificar as tonalidades de sombreamento - no caso de imagens em preto e branco - para que sejam melhor identificadas. Pode-se destacar os *filtros lineares, não lineares* e técnicas de *equalização de histograma* [39]. Um recurso que por vezes é utilizado, dependendo do equipamento disponível, é o da *falsa cor*. Este recurso bem simples, permite colorir a imagem com tonalidades diferentes, independentes das suas cores reais - daí o seu nome - de acordo com o nível de sombreamento determinado. Isso aumenta em muito a capacidade de discernimento da vista humana, já que para níveis de cinza é da ordem de centenas e para cores, a capacidade de distinção chega a alguns milhares.

1.3.1.c. Incorporação de Informações

O uso de informações adicionais sobre a topografia do terreno, apontou a necessidade da criação dos chamados Modelos Digitais de Terreno - M.D.T. - que consiste em alocar em determinados pixels, informações sobre a topografia do terreno naquele ponto. Suas principais aplicações são: construções de imagens em perspectiva, estudos de luminosidade, avaliações de projetos e até mesmo estudo e previsão de catástrofes.

1.3.2. Tratamento de Imagens

Geralmente, não se possui o equipamento adequado para trabalhar com a imagem da forma que se deseja, visto que são necessários grandes esforços computacionais para um estudo deste tipo. Quando esta situação ocorre, se faz necessária uma

"compressão" dos dados. Sendo $X_{1 \times n}$, o vetor de atributos (espaço de todas as medidas possíveis feitas pelos sensores do satélite), comprimir esses dados, significa reduzir a dimensão n deste vetor a níveis aceitáveis pelo usuário e pelo equipamento. Deve-se realçar, no entanto, a necessidade de procurar sempre a maior sofisticação de técnicas e equipamento o quanto possível. Com a facilidade de acesso a diferentes tipos de tecnologias de "software" e "hardware" e com a diminuição dos seus custos, estes tipos de técnicas para selecionar atributos poderão vir a se tornarem desnecessárias no decorrer dos próximos anos. Porém, como ainda são necessárias em muitos casos, far-se-á aqui uma breve descrição destes tipos de técnicas.

Os dois conjuntos de técnicas mais empregadas utilizam métodos estatísticos. O primeiro conjunto é constituído de transformações nos espaços dos atributos e o segundo utiliza medidas de distância.

No primeiro caso, o objetivo é reduzir a dimensão preservando ao máximo a representação de padrões e/ou a discriminação entre as classes caracterizadas pelos atributos. São utilizados ou o critério do Erro Médio Quadrático (Componentes Principais) ou o critério da Entropia.

Resumidamente, a técnica de Componentes Principais [40] considera a distribuição multivariada global da mistura de classes. Obtém-se a matriz de covariâncias $\Sigma_{n \times n}$ do vetor de atributos. A matriz de transformação $T_{p \times n}$ será tal que as linhas da matriz T são auto-vetores correspondentes aos maiores auto-valores de Σ . Assim, as componentes do vetor $Y = T X$, são

não-correlacionadas e Y terá dimensão $p < n$. A escolha do valor numérico de p é dada ao usuário que geralmente dispõe de uma avaliação sobre a representação do vetor Y para vários valores de p .

O critério da Entropia [41], procura reduzir a dimensão preservando o máximo de informação possível.

$$H(X) = - E \left[\log (f_x (X)) \right] \quad (1.1)$$

Por exemplo, se X tiver distribuição Normal, ou Gaussiana, com matriz de covariância Σ , então:

$$H(X) = \frac{1}{2} \log |\Sigma| + \frac{n}{2} \log (n \pi e) \quad (1.2)$$

onde π é o número pi, e e é o número de Neper.

Aplicando Componentes Principais :

$$H(Y) = \frac{1}{2} \sum_{i=1}^p \log \lambda_i + \frac{p}{2} \log (2 \pi e) \quad (1.3)$$

onde p é a dimensão do vetor Y e λ_i são os auto-valores da matriz T .

No segundo caso, ou seja, medidas de distância, visa-se a redução da dimensão procurando minimizar a probabilidade de erro. Na realidade, os critérios fornecem limites inferiores e superiores para esta probabilidade. São muito utilizados os critérios da Divergência e a Distância J-M.

A Divergência [41] toma as classes W_i e W_j , com suas respectivas densidades de probabilidades condicionais : $f(X \setminus W_i)$ e $f(X \setminus W_j)$. Seu princípio consiste em tomar a razão de

verossimilhança $L_{ij}(\underline{X})$ - figura 1.7 - como:

$$L_{ij}(\underline{X}) = \frac{f(\underline{X} \setminus W_i)}{f(\underline{X} \setminus W_j)} \quad (1.4)$$

e,

$$L'_{ij}(\underline{X}) = \log_e \left(L_{ij}(\underline{X}) \right) \quad (1.5)$$

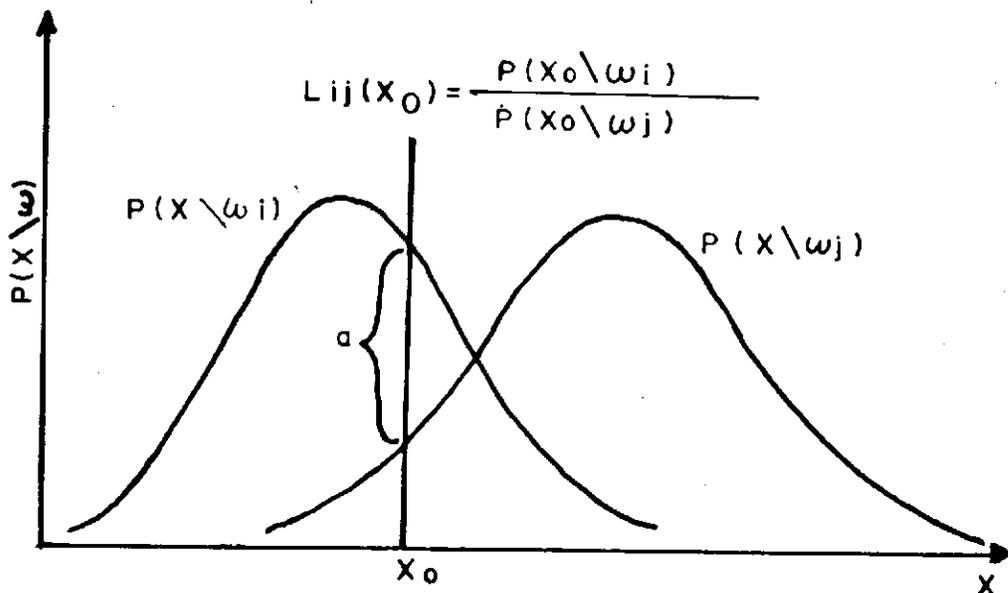


Figura 1.7 - Definição da razão de verossimilhança no ponto.

A Divergência é definida por :

$$D_{ij} = E \left(L'_{ij}(\underline{X}) \setminus W_i \right) + E \left(L'_{ji}(\underline{X}) \setminus W_j \right) \quad (1.6)$$

onde :

$$E \left(L'_{ij}(\underline{X}) \setminus W_i \right) = \int_{\underline{X}} L'_{ij}(\underline{X}) f(\underline{X} \setminus W_i) d\underline{X} \text{ e,}$$

$$E \left(L'_{ji}(\underline{X}) \setminus W_j \right) = \int_{\underline{X}} L'_{ji}(\underline{X}) f(\underline{X} \setminus W_j) d\underline{X}$$

Um procedimento usual para computar a Divergência entre M classes, $M > 2$, é computar a divergência média entre todos os pares de classes e, selecionar o subconjunto de N atributos, para o qual a divergência média definida por :

$$D_M = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=1}^M D_{ij} \quad (1.7)$$

é máxima.

No entanto há problemas, pois essa forma não é ótima. Um único termo da somatória pode elevar muito o resultado final. Uma recomendação seria analisar cada divergência entre as classes i e j , num processo de seleção.

Para evitar os problemas da Divergência, pode-se utilizar a Distância Jeffries-Matusita [35]. Ela é uma medida da diferença média entre funções densidades de duas classes, e é definida formalmente por:

$$J_{ij} = \left\{ \int_{\tilde{X}} \left(\sqrt{f(\tilde{X} \setminus W_j)} - \sqrt{f(\tilde{X} \setminus W_i)} \right)^2 d\tilde{X} \right\}^{1/2} \quad (1.8)$$

1.3.3. Classificação

Não se entrará aqui nos méritos das formas atualmente utilizadas para classificação de imagens orbitais, mas dos fundamentos destes processos. Lembrando o fluxograma da figura 1.6, vê-se as diversas formas de classificação. Todos estes processos tem por base a Teoria Estatística de Decisão, idealizada por Fisher [40] e [42] - figura 1.8 - na década de 30.

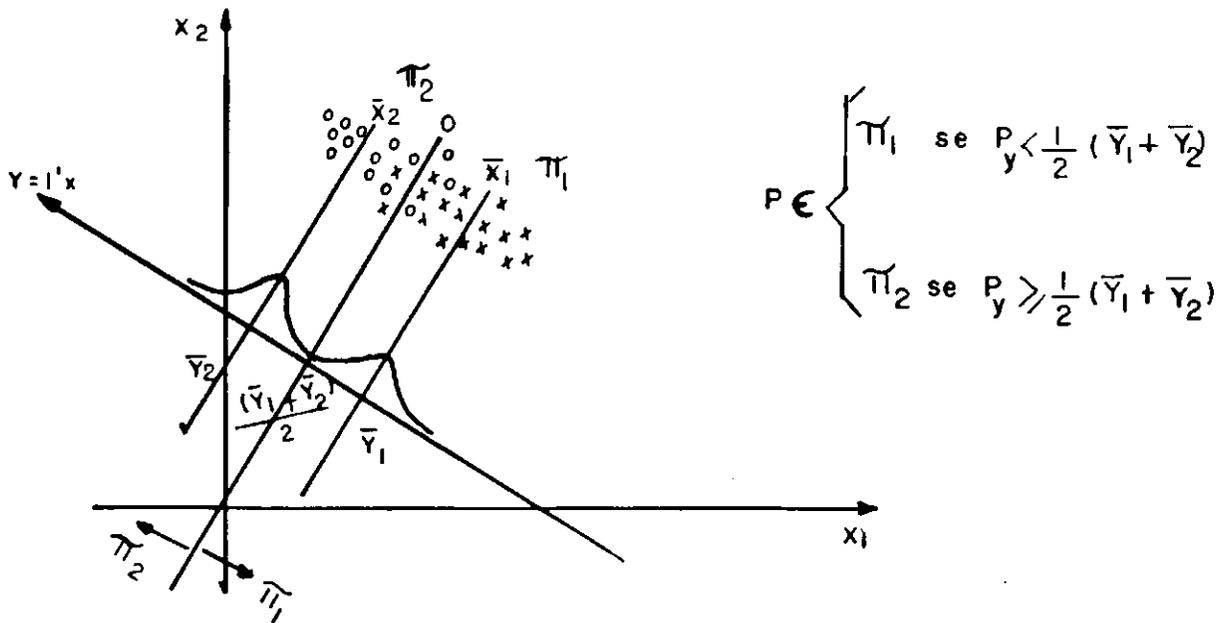


Figura 1.8 - Princípio discriminante de Fisher.

O modelo sugerido de discriminação entre duas classes, poderia também alocar novos objetos aos grupos. Observa-se, no entanto, que mesmo para um caso muito simples, alguns pontos serão classificados de maneira errônea. Na realidade, para toda e qualquer classificação, sempre haverá o risco de se classificar um objeto em uma classe a qual, na verdade, ele não pertence.

Os métodos de classificação dividem-se em dois grupos principais: os não-supervisionados e os supervisionados. Este último, por sua vez, também se subdivide em dois: Métodos Clássicos e Métodos Contextuais. Os não-supervisionados dividem as classes automaticamente e em sua maioria baseiam-se no princípio de Fisher. Alguns métodos podem ser baseados também em funções discriminantes matemáticas, como por exemplo a Teoria dos Grafos e relações nebulosas [5]; abordagens estatísticas assintóticas [1], ou ainda modelos de pertinência nebulosa [43]. Os supervisionados

são mais precisos por levarem em consideração informações fornecidas pelo usuário, a verdade terrestre. Baseiam-se, além de funções discriminantes, na teoria de Bayes e requerem alguns conhecimentos prévios das classes.

1.3.3.a. Método Supervisionado Clássico

Nos métodos clássicos, cada "pixel" é classificado tomando-se em consideração as informações contidas no "pixel". Já nos contextuais, levam-se em conta também as informações dos "pixels" da vizinhança, para a classificação.

Apenas a título de ilustração, será mostrado aqui um método de cada. Será dada maior atenção aos métodos supervisionados, por serem próximos à área de interesse deste trabalho de dissertação. Como exemplo do método supervisionado clássico, será mostrado o Método do Custo Mínimo de Classificação Errônea (ECM) [41]. Apesar de não ser um recurso ótimo em termos computacionais, é bastante didático. Sendo o vetor de atributos \underline{x} composto por variáveis aleatórias x_i , com densidades $f_i(\underline{X})$, $i = 1, 2, \dots, N$ que serão atribuídas às populações π_j (partição do espaço dos atributos), com probabilidades a priori $P(\pi_j)$, $j = 1, 2, \dots, M$; $M < N$ e associadas às densidades $f(\underline{X} \setminus \pi_j)$. O custo de se classificar um objeto é dado por $C(\pi_k \setminus \pi_j)$, com probabilidade $P(\pi_k \setminus \pi_j)$.

Assim, o risco médio de se decidir pela população π_j , é dada por :

$$R.M.(\pi_j) = \sum_{\substack{l=1 \\ l \neq j}}^M P(\pi_l \setminus \pi_j) C(\pi_l \setminus \pi_j) \quad (1.9)$$

Portanto, o procedimento ótimo de decisão a ser tomado, é o que minimize o risco médio, ou seja :

$$ECM (\pi_j) = \sum_{j=1}^M P (\pi_j) \left[\sum_{\substack{l=1 \\ l \neq j}}^M P (\pi_l \mid \pi_j) C (\pi_l \mid \pi_j) \right] \quad (1.10)$$

Se acaso os custos dos erros de classificação errônea forem todos iguais, a minimização da equação (1.9) equivale a maximizar a probabilidade a posteriori $P (\pi_k \mid X)$. Por Bayes:

$$P (\pi_k \mid X) = \frac{f (X \mid \pi_k) P (\pi_k)}{f_k (X)} \quad (1.11)$$

Como o denominador é independente, têm-se a regra de decisão de alocar o "pixel" x_j na população π_k se :

$$P (\pi_k) f (X \mid \pi_k) > P (\pi_i) f (X \mid \pi_i) ; i \neq k \quad (1.12)$$

Têm-se ainda que o cálculo da probabilidade de classificação correta é dada por :

$$P (C.C.) = \sum_{i=1}^M \int_{R_i} P (\pi_i) f (X \mid \pi_i) dX \quad (1.13)$$

onde R_i é a região atribuída à população π_i .

1.3.3.b. Método Supervisionado Contextual

Os métodos contextuais, fundamentalmente, introduzem um fator de correção denominado *fator de correção contextual* [15]. Resumidamente, temos K classes possíveis para classificação, com probabilidades a priori p_1, p_2, \dots, p_k , com densidade da classe k

dada por $f_k (x)$.

Reserva-se também a possibilidade de se classificar um "pixel" de forma duvidosa. Assim, o espaço das decisões será:

$$\Omega = \{ 1, 2, \dots, K, D \} \quad (1.14)$$

Para facilitar a compreensão, assume-se que a perda incorrida por todos os erros de classificação seja um, e que a perda por se estar em dúvida seja $e \in [0, 1-1/K]$. Nos sistemas, geralmente, essa função-perda é implementada com valores fixos, ou "default" para cada classe. Esses valores também podem ser alterados, privilegiando certa(s) classe(s) de acordo com a conveniência do usuário, fazendo diferenças mais realistas na classificação.

Num método como este, pode-se fazer com que a classificação de cada "pixel" dependa até de todos os outros "pixels" da imagem. Didaticamente, consideram-se apenas um "pixel" e sua vizinhança. Para o "pixel" i , temos a sua vizinhança denotada por V_i , e o seu vetor característico por D_i . Se V_i consiste em n "pixels" em adição ao "pixel" central, o vetor característico será denotado por :

$$D_i = \{ x_i, x_{i1}, \dots, x_{in} \} \quad (1.15)$$

onde as variáveis aleatórias x também podem vir a ser vetores aleatórios X , quando no caso de imagens multiespectrais..

A regra de classificação Bayesiana pode ser :

$$C_i = \begin{cases} D, & \text{se } P_i (k \setminus D_i) < 1 - e, \text{ para todo } k = 1, \dots, K \\ m, & \text{se } P_i (m \setminus D_i) = \max_{k \leq K} P_i (k \setminus D_i) \geq 1 - e \end{cases} \quad (1.16)$$

onde $P_i (k \setminus D_i)$ é a probabilidade de C_i seja igual a k dado D_i , ou

seja a probabilidade a posteriori da classe k , dado os vetores característicos em V_i . Assim, sem entrar em muitos detalhes, tem-se a expressão :

$$P_i (k \setminus D_i) = \frac{p_k f_k (x_i) R_k (x_{i1}, \dots, x_{in})}{P_i (D_i)} \quad (1.17a)$$

onde :

$$R_k (x_{i1}, \dots, x_{in}) = \sum_{k_1} \dots \sum_{k_n} h (x_{i1}, \dots, x_{in} \setminus x_i, k, k_1, \dots, k_n) g (k_1, \dots, k_n \setminus k) \quad (1.17b)$$

e onde a função h descreve a densidade de probabilidade conjunta dos vetores característicos D_i , dado que os "pixels" em V_i pertencem à classe k ("pixel" central), k_1, k_2, \dots, k_n . O segundo fator, função g , é a probabilidade desta configuração em particular de classes, dado que o "pixel" central pertence à classe k . A somatória é o total das possíveis configurações.

A expressão (1.17) pode parecer de difícil resolução à primeira vista, principalmente pelo grande número de termos envolvidos, mas não o é na prática. O denominador, como no método clássico é independente de K , o que facilita, bastando maximizar o numerador. Nos métodos clássicos maximiza-se $p_k f_k (x_i)$, podendo-se dizer que R_k é aquele fator de correção contextual mencionado anteriormente. O exemplo abaixo ilustra o mecanismo :

Exemplo 1.1.: Seja $V_i =$ "pixel" $i + 4$ "pixels-vizinhança".

A função g é:

$$g(k_1, k_2, k_3, k_4 \setminus k) = PR \left(\begin{array}{c} \boxed{k_1} \\ \boxed{k_3} \quad \boxed{} \quad \boxed{k_2} \\ \boxed{k_4} \end{array} \right) \begin{array}{l} \text{pixel central} \\ \text{é da classe} \\ k \end{array}$$

A função h é :

$$h(x_{i1}, x_{i2}, x_{i3}, x_{i4} \setminus x, k, k_1, k_2, k_3, k_4) = PR \left(\begin{array}{c} \boxed{x_{i1}} \\ \boxed{x_{i3}} \quad \boxed{} \quad \boxed{x_{i2}} \\ \boxed{x_{i4}} \end{array} \right) \begin{array}{c} \boxed{k_1} \\ \boxed{k_3} \quad \boxed{x_i} \quad \boxed{k_2} \\ \boxed{k_4} \end{array}$$

sendo que x_i pertence a classe k , conforme mencionado anteriormente. #

Até aqui foram apresentados apenas a nível introdutório os sistemas de classificação. Será visto no próximo capítulo, em mais detalhes, o método contextual. No entanto, para mais detalhes de diversos métodos de classificação estatística, pode-se citar Baridó [44] que em sua dissertação de mestrado tece comentários sobre vários métodos - paramétricos e não-paramétricos - e os compara a partir de simulações. O próprio método contextual tem a sua versão não-paramétrica que pode ser visto em Baridó [44] e James [45] que também fornece além do algoritmo, um programa simples em linguagem BASIC para o caso de um vizinho (1-NN). Também em James [45], pode-se ver no capítulo 10 uma explanação introdutória sobre o reconhecimento de padrões e classificação

auxiliado por inteligência artificial que deverá ser a tendência para os próximos anos.

1.3.3.c. Métodos de Relaxação

Os métodos de relaxação que são utilizados como dispositivos de classificação contextual são baseados em uma filosofia diferente da apresentada em 1.3.3.b. Inicialmente, estimam-se os "graus de pertinência" de um "pixel" a cada uma das K classes possíveis. Normalmente, utiliza-se para o "pixel" i , as K probabilidades a posteriori $P(k \mid x_i)$, $k = 1, \dots, K$. Neste ponto, observa-se os vizinhos deste "pixel" para checar se os "graus de pertinência" destes estão em razoável correspondência com aqueles do "pixel" i . Esta avaliação é baseada em alguns conhecimentos a priori dos padrões nos quais as classes ocorrem. Classifica-se assim o "pixel" i numa classe concordante com os seus vizinhos, a partir das informações das probabilidades a posteriori - "graus de pertinência" - e probabilidades a priori obtidas de informações da própria imagem [15].

O procedimento pode ser ilustrado a partir do seguinte exemplo:

Exemplo 1.2.: Dentre todas as classes disponíveis para um dado "pixel" i , a que obteve maior "grau de pertinência" foi a classe "Campo de Feijão". No entanto, os "pixels" da sua vizinhança obtiveram maior grau para a classe "Bosque de Eucaliptos" e as informações a priori fornecidas dão conta que um campo de feijão em meio a um bosque de eucaliptos é de probabilidade muito

pequena. Num caso como este, deseja-se ajustar as estimativas iniciais para o "pixel" i e sua vizinhança. Inicia-se assim um procedimento iterativo visando ajustar as probabilidades a posteriori para o "pixel" i e seus vizinhos. Faz-se necessário observar que as probabilidades a posteriori são obtidas a partir de um procedimento *não-contextual*. #

Este procedimento iterativo pode ser formulado de muitas maneiras diferentes, existindo portanto, vários tipos de métodos de relaxação [15-16] e [46].

1.3.3.d. Estimativa de Parâmetros - Treinamento do Classificador

Os métodos de classificação supervisionados necessitam de estimativas para os parâmetros das funções discriminantes. Apesar de ser possível em alguns casos, determinar tais funções com base em considerações teóricas, com o conhecimento prévio a respeito da imagem, ou eventualmente pela intuição, o mais comum é o uso de informações retiradas de áreas de treinamento na própria imagem, fotografias aéreas, mapas, ou investigação direta no campo. Esses conhecimentos, denominados *verdades terrestres*, são na realidade, "pixels" sobre os quais se conhece suas verdadeiras classes - daí o nome. Com base nisto, o classificador é "treinado" a reconhecer "pixels" de estrutura semelhante. A estimativa da densidade de probabilidade para o vetor de atributos, $f_k(x)$, $k = 1, \dots, K$ e os parâmetros das classes, como probabilidades a priori, probabilidades condicionais, etc, requerem diferentes tipos de conhecimento da verdade terrestre. Por exemplo, para estimar as

densidades $f_k(x)$, necessita-se de áreas razoavelmente homogêneas de treinamento para cada classe. No entanto, para as probabilidades à priori $\pi(k)$, as estimativas não podem advir de tais áreas de treinamento, necessitando de uma sistemática diferente, como por exemplo, uma inspeção visual da cena ou pontos amostrados aleatoriamente. No caso de classificadores contextuais, as funções discriminantes necessitam da verdade terrestre dos pontos e de um número determinado de vizinhos, de acordo com o modelo, exigindo uma forma de amostragem específica para estes pontos.

A tabela 1.2, adaptada de Saebo et alii [15], mostra alguns métodos de classificação, a densidade multivariada comumente utilizada, os parâmetros de cada modelo a serem estimados e possíveis caminhos para fazê-los:

Metodo	$\bar{I}_k(x)$	Parâmetro de classe	Procedimento para fornecer estimativas
CLÁSSICO			
- Máxima Verossimilhança	μ_k, Σ_k	—	Áreas de treinamento homogêneas
- Bayes	μ_k, Σ_k	$\pi(k)$	+ pontos amostrados
CONTEXTUAL			
- Owen & Switzer	μ_k, Σ_k	$\pi(k), p, q, r$	Áreas de treinamento homogêneas + pontos amostrados com seus vizinhos.
- Relaxação	μ_k, Σ_k	$\pi(k), \pi(l \setminus k)$	Áreas de treinamento homogêneas + pontos amostrados com seus vizinhos.

Tabela 1.2 - Alguns métodos de classificação e metodologia de treinamento.

Uma explanação detalhada sobre os métodos pode ser encontrada em [15-20] e [46] e, mais especificamente sobre o método contextual de Owen & Switzer, no próximo capítulo. Sobre o Método da Máxima Verossimilhança, podem ser encontrado detalhes em [35].

1.4. Interpretação Visual da Imagem

Neste tópico, será dada uma breve introdução às metodologias de análise para quem não dispõe de nenhum dos recursos citados anteriormente. Trata-se da Interpretação Visual da Imagem (I.V.I.) [47]. A imagem é então adquirida já classificada, porém sem um fim específico. Normalmente, quando se utiliza um processo de classificação, a tendência é a procura de uma melhor qualidade na área específica de interesse. Este pode ser o grande ponto desfavorável a interpretação visual. Este tipo de material é fornecido pelo Instituto de Pesquisas Espaciais - INPE - como sendo uma grande foto em papel, podendo ser esta colorida ou preto e branco, de determinada época de interesse, mono ou multiespectral, etc; ou seja, há várias opções de modo que o usuário deve procurar a que melhor se adequa à sua necessidade. Tendo a imagem em mãos, o passo seguinte é o de reconhecer o que existe na imagem e extrair dela o que se interessa.

A interpretação visual requer do usuário conhecimentos prévios de razoável monta. Por exemplo, há a necessidade do conhecimento prévio da época adequada para a obtenção da imagem, tipo de produto (colorido, ou P/B, já que nem sempre o colorido é que traz mais informações), as bandas a serem escolhidas (cada banda reflete uma determinada faixa do espectro). Para trabalhos com

monitoramento ambiental, por exemplo, pode-se encontrar boas indicações em Tucker [48], as escalas (que devem ser ao máximo compatíveis com mapas, cartogramas, etc; disponíveis sobre a região), entre outros. Dependendo do tipo de interesse, a IVI requer mais conhecimentos da área "in loco", o que nem sempre é possível, ou associações com fotos aéreas, entre outros.

Os fotointerpretadores podem ser classificados em três grupos: primeiro, os que possuem conhecimento de campo; segundo, os que não possuem este conhecimento e, por fim, os que não possuem este conhecimento, mas que possuem fontes de informações adicionais. Os primeiros conseguem interpretar de forma mais fácil que os demais. Os do terceiro grupo podem ter dificuldades em cruzar as informações de que dispõem com a imagem. Já os do segundo grupo, dificilmente conseguirão fazê-lo sem o uso de *chaves de interpretação* - conhecimentos utilizados que são estruturados como se fossem formulações para se resolver um problema. É de grande valia que o uso das chaves de interpretação, quando necessárias (na maioria das vezes), seja feita de forma invariante, ou seja, não possa mudar de um fotointerpretador para outro. Para isso, cria-se legendas que vão funcionar como critérios de separação das diversas classes. Além de fixar a interpretação, as legendas auxiliam uma leitura mais rápida.

Feita a interpretação, parte-se para extrair da imagem as informações que se procura desde o início. Supondo um exemplo agrícola de estimativa de áreas plantadas, pode-se agir de duas formas. Na primeira, usa-se uma malha de pontos proporcionais à escala para se estimar a área, como um método grosseiro de

integração. Logicamente que os resultados serão induzidos a muitos erros, às vezes até podem comprometer todo um trabalho. Uma solução seria trabalhar com uma malha de pontos bastante densa, o que ainda assim proporciona um risco de proporções razoáveis se o terreno em questão tiver um relevo acidentado. Uma segunda forma, já bem mais sofisticada, requer um computador em associação a uma mesa digitalizadora. A mesa transfere contornos da região ao computador e em associação a um mapa da região, incorpora também informações sobre o relevo. A seguir, um simples processo de integração numérica fornece a estimativa da área com erros bastante aceitáveis. Esta forma, porém é bastante onerosa, mas permite que se leve em consideração a tridimensionalidade. Com a tecnologia atual - a utilização de "scanners" pode transformar o trabalho ainda mais preciso, pois evita o contato manual - a transferência da imagem é feita por um processo digital de leitura da imagem. A grande desvantagem destes dois últimos processos é a necessidade de "software" específico, nem sempre disponível com facilidade para este tipo de aplicação.

Em algumas aplicações da área agrícola, quando possível em termos de escala, pode-se utilizar dados amostrais do espaçamento médio no plantio na área de interesse para a estimativa da área plantada [49]. No entanto, a distância do pesquisador pode impossibilitar este procedimento. Geralmente, para se obter uma escala que permita tal visualização, deve-se recorrer ao uso de aeronaves, o que pode tornar inviável economicamente um projeto desta natureza. Porém, conseguindo-se agrupar recursos desta magnitude, os resultados deverão ser bastante confiáveis,

podendo-se precisar de forma estatística o erro envolvido na operação.

1.5. Testes e Controle

Qualquer que seja o tipo de classificação, seja visual ou assistida por computador com qualquer algoritmo que se possa utilizar, sempre irão aparecer dúvidas com respeito à classificação obtida. Para sanar este problema, o procedimento usual é visitar o local. No entanto, este tipo de procedimento nem sempre é possível ou, em certos casos, nem mesmo viável. Uma forma de solução para o caso é a conjugação de imagens, tanto no aspecto espectral - imagens multiespectrais - quanto no aspecto temporal - imagens obtidas em diferentes épocas do mesmo local. Por exemplo, no caso agrícola da seção anterior, fotos obtidas em diferentes épocas mostrarão taxas de cobertura de solo diferenciadas, permitindo a diferenciação de culturas.

Na maioria das vezes os erros cometidos na classificação são inversamente proporcionais aos recursos - humanos, financeiros e de equipamento - alocados para o cumprimento da tarefa. Por estes motivos, é sempre recomendável o máximo aproveitamento e utilização de recursos e a maior sofisticação possível. É claro, que isto depende da finalidade com que se faz um processamento de uma imagem, já que não vale a pena investir na melhoria da classificação de uma região que não a de interesse, e da proporção de erro aceitável para o cumprimento da análise.

CAPÍTULO 2

O Método Contextual

2.1. Introdução

Foi visto no capítulo anterior, na seção 1.3.3.b, uma pequena introdução ao método supervisionado contextual de classificação. Neste capítulo, ele será visto de uma forma analítica mais aprofundada e, será imposto um rigor matemático e estatístico convenientes.

Inicialmente, como uma visão panorâmica, será feito uma revisão histórica de alguns métodos que foram propostos por diversos autores para a seguir discutir mais profundamente o modelo escolhido para implementação descrita nesta dissertação que deverá ser compatível com o sistema SITIM 150 do INPE/Engespaço - São José dos Campos - SP.

Os métodos contextuais em geral, baseiam-se no mesmo princípio, como descrito no capítulo anterior. Exemplos disto, que podem ser citados: Welch e Salter [3] que remonta à época do surgimento dos primeiros métodos contextuais propostos, em 1971. Landgrebe em 1980 [6], propôs um algoritmo espectral-espacial com a denominação ECHO. No entanto, já se sabia que o número de vizinhos era crítico e o aumento do número de vizinhos aumentava sobremaneira o tempo de processamento. Devido a esses problemas, Swain et alli [7] propuseram em 1980, o uso de um sistema de multiprocessadores em paralelo com o "hardware" principal visando aumentar a velocidade de processamento. Swain et alli [8], 1981,

propuseram um novo algoritmo a partir da teoria de decisão estatística composta. Tilton e Swain [9] estudaram em 1981 com profundidade a metodologia contextual e propõem um algoritmo aproximadamente contextual, tentando resolver o problema do tempo de processamento e um outro algoritmo híbrido. Posteriormente Tilton et alli [11] propuseram em 1982, um estimador alternativo para a função contextual visando aumentar a sua eficiência. Wharton [12] em 1982, desenvolveu um algoritmo - CONAN - para classificação de dados terrestres de alta resolução. Owen [14] procurando resolver o problema de velocidade propôs em 1984 uma restrição ao número de classes em cada vizinhança. Hjort e Mohn [16] publicaram uma comparação entre vários métodos contextuais. Saebo et alli [15] publicam em 1985 um relatório técnico sobre a implementação de métodos contextuais. Neste relatório, pode-se ver nos apêndices, os trabalhos de Hjort e Mohn [16], Hjort [17], Hjort et alli [18], Hjort e Mohn [19] e Hjort [20], sobre modelos, estimadores e ajustes das regras contextuais. Ripley [50] em um trabalho introdutório ao Sensoriamento Remoto publicado em 1986, cita uma pequena bibliografia contextual. No mesmo ano, Bezdek [21] publicou um algoritmo contextual utilizando abordagem nebulosa - "fuzzy". Recentemente, depois de uma certa ausência na literatura especializada, aparecem os trabalhos de Klein e Press [31] e [33], de 1989 e 1990 respectivamente, propondo um método contextual com estrutura de correlação estatística espacial e vizinhos não classificados, os quais podem ou não pertencer às áreas de treinamento.

De qualquer forma, o grande problema quanto ao processamento

dos métodos contextuais de classificação, ainda não foi resolvido. O fator de correção contextual - apresentado na equação (1.17b) - com suas somatórias, proporcionais ao total de possíveis configurações, ainda está presente. Como mencionado acima, vários trabalhos se concentraram na resolução, ou tentativa de reduzir o tempo computacional gasto. Na equação (1.17b) o número de termos é K^α , onde α é o número de vizinhos. Hjort [17] em 1985 com a sua metodologia baseada em um trabalho anterior de Owen and Switzer [10], reduzia o número para $8K-7$. No entanto, essas reduções ainda não eram satisfatórias para a época, principalmente pelo equipamento utilizado. Atualmente, com o rápido avanço de "hardware" e novas linguagens de programação, mais rápidas e poderosas, o problema de redução da quantidade de cálculos envolvida para aumentar a velocidade deverá perder a importância.

Após esta pequena introdução, passaremos a discutir com maior profundidade o sistema cuja implementação é apresentada neste trabalho. Inicialmente, a escolha do algoritmo de Hjort [17] se deu devido à significativa redução do número de termos conseguida em seu artigo e também porque dentro de certos limites, este método é o mais aceitável, estatisticamente falando. Isto se deve ao fato de que a restrição a duas classes nas cruzes, empiricamente falando, tem fundamento real. Dificilmente encontram-se imagens onde este número em cada cruz supera três classes. Para o caso específico de três classes, no entanto, não é propriamente tão difícil assim. Porém, se este caso fosse também considerado, obviamente a redução não seria tão significativa. Em seu artigo, Hjort [17] denomina-o *Modelo p,q,r*, como sendo um

modelo de probabilidade geométrica simples. No seu artigo é considerado o caso de quatro e oito vizinhos. Aqui será considerado o caso de quatro vizinhos apenas e será este o modelo a ser implementado.

2.2 O Modelo Contextual p, q, r .

Para melhor compreensão e arranjo do texto, será relatado o método em sua totalidade, mesmo sob pena de ser repetitivo em algumas partes com respeito ao capítulo anterior. A notação será seguida de [17], facilitando consultas posteriores.

Como visto no capítulo anterior, na imagem a ser classificada, considera-se a existência de K classes. Por mera formalidade, define-se a imagem como um "array" bidimensional de \mathcal{N} "pixels" multivariados, onde $\mathcal{N} = \mathcal{N}_1 \times \mathcal{N}_2$, sendo \mathcal{N}_1 o número de linhas e \mathcal{N}_2 o número de colunas. No espaço de decisão Ω , \hat{C}_i a procurada classe do "pixel" i , $\hat{C}_i \in \{1, 2, \dots, K, D\}$; onde D é a reserva de se classificar um "pixel" de forma duvidosa, ou seja, não se pode classificar o "pixel" i em nenhuma das K classes disponíveis. No caso de imagens multiespectrais, o "pixel" i é na realidade um vetor X d -dimensional composto pelas medidas espectrais dos diversos canais adquiridos pelo usuário. Por exemplo, no caso de uma imagem SPOT, o usuário pode adquirir até quatro bandas da mesma imagem. Neste caso, na composição da imagem, cada "pixel" seria um vetor de dimensão quatro. Os vetores X de uma classe k , seguem uma densidade de probabilidade multivariada $f_k(X)$.

Considera-se agora que a vizinhança assumida para o "pixel" i

seja composta pelos seus quatro vizinhos imediatamente ao norte, ao sul, à leste e à oeste, como na figura 2.1, formando uma cruz.

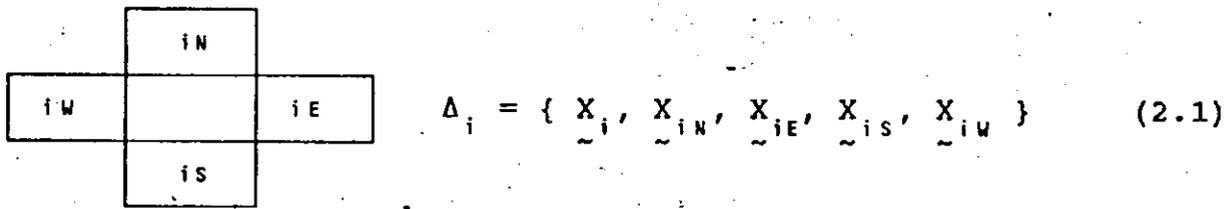


Figura 2.1 - A vizinhança Δ_i composta por cinco pixels em cruz.

A probabilidade desta configuração particular de classes em cruz é dada por :

$$g(a, b, c, d \setminus k) = \Pr \{ C_{iN} = a, C_{iE} = b, C_{iS} = c, C_{iW} = d \setminus C_i = k \} \quad (2.2)$$

No entanto, é óbvio que geometricamente falando, os vizinhos podem ser permutadas e geram novas probabilidades $g(\cdot \setminus k)$, sendo elas:

$$\begin{array}{cccc} g(a, b, d, c \setminus k) & g(b, a, c, d \setminus k) & g(c, a, b, d \setminus k) & g(d, a, b, c \setminus k) \\ g(a, c, b, d \setminus k) & g(b, a, d, c \setminus k) & g(c, a, d, b \setminus k) & g(d, a, c, b \setminus k) \\ g(a, c, d, b \setminus k) & g(b, c, a, d \setminus k) & g(c, b, a, d \setminus k) & g(d, b, a, c \setminus k) \\ g(a, d, b, c \setminus k) & g(b, c, d, a \setminus k) & g(c, b, d, a \setminus k) & g(d, b, c, a \setminus k) \\ g(a, d, c, b \setminus k) & g(b, d, c, a \setminus k) & g(c, d, b, a \setminus k) & g(d, c, a, b \setminus k) \\ & g(b, d, a, c \setminus k) & g(c, d, a, b \setminus k) & g(d, c, b, a \setminus k) \end{array} \quad (2.3)$$

o que perfaz um total de 24 possíveis configurações, ou P^4 - permutação de quatro classes distintas. Soma-se a estas, as possíveis combinações com uma, duas e três possíveis classes distintas, se assumimos $K = 4$. É necessário ainda, levar em conta que se o número total de classes for maior que quatro, têm-se ainda as possíveis combinações das K classes, com cada caso apresentado.

Deve-se ainda computar a densidade conjunta dos vetores

característicos em Δ_i dado as classes na cruz e X_i , ou seja:

$$f(\Delta_i \setminus k, a, b, c, d) = f_k(X_i) \quad h(\Delta_i \setminus k, a, b, c, d, X_i) \quad (2.4)$$

A proposta Bayesiana para classificação recomenda minimizar o risco de classificação errônea [17]. Seja $L(C_i, \hat{C}_i)$ a função-perda por se decidir classificar o "pixel" i em uma classe a qual ele não pertença:

$$L(C_i, \hat{C}_i) = \begin{cases} 0, & \text{se } \hat{C}_i = C_i & \text{(decisão correta)} \\ 1, & \text{se } \hat{C}_i \neq C_i \text{ e } \hat{C}_i \in \{1, \dots, K\} & \text{(decisão errada)} \\ e, & \text{se } \hat{C}_i = D & \text{(dúvida)} \end{cases} \quad (2.5)$$

onde $e \in [0, 1 - (1/k)]$.

A perda média é dada por:

$$L_M(C_i, \hat{C}_i) = \frac{1}{\pi} \sum_{i=1}^{\pi} L(C_i, \hat{C}_i) \quad (2.6)$$

que é na realidade a soma da fração de classificação errônea com a fração de classificação duvidosa. Portanto, a perda esperada é uma função destas duas frações, ou seja:

$$\begin{aligned} E[L_M(C_i, \hat{C}_i)] &= \hat{e}(e) + e \hat{d}(e) \quad (2.7) \\ &= E\left[\frac{1}{\pi} \sum_{i=1}^{\pi} L(C_i, \hat{C}_i) \right] \end{aligned}$$

onde \hat{e} e \hat{d} são as taxas de erro e dúvida respectivamente.

A regra ótima de Bayes de classificação manda minimizar a perda esperada $E[L_M(C_i, \hat{C}_i)]$. Sem entrar em detalhes¹, pode-se

1.

Nota: Pode-se ver em Welch and Salter [1], uma metodologia análoga para se obter um classificador contextual.

mostrar que a regra geral de classificação com opção de rejeição encontrada será :

$$\hat{C}_i = \begin{cases} k, & \text{se } P(k \setminus \Delta_i) = \max_{m \leq K} P_i(m \setminus \Delta_i) \geq 1-e \\ D, & \text{se } P(k \setminus \Delta_i) < 1-e, \forall k \in [1, K]. \end{cases} \quad (2.8)$$

onde $P(k \setminus \Delta_i) = \Pr \{ C_i = k \setminus \Delta_i \}$ (2.9)

Segundo a escola Bayesiana [52] e [53], a probabilidade $P(k \setminus \Delta_i)$ é uma probabilidade a posteriori. Esta probabilidade deve depender dos parâmetros k e Δ_i e da probabilidade "inicial" para o fator k que é a probabilidade a priori de k ; $\pi(k)$. A probabilidade a priori - $\pi(k)$ - é interpretada como a quantidade de "pixels" da classe k em uma cena imaginária muito maior que a cena em questão, ou o que sobra da cena em questão, depois de removidos os conjuntos de treinamento. Então, pode-se reescrever a equação (2.9) como:

$$P(k \setminus \Delta_i) = \frac{1}{f(\Delta_i)} \pi(k) \sum_{a, b, c, d} g(a, b, c, d \setminus k) f(\Delta_i \setminus k, a, b, c, d) \quad (2.10)$$

$$= \frac{1}{f(\Delta_i)} \pi(k) f(X_i) \sum_{a, b, c, d} g(a, b, c, d \setminus k) h(\Delta_i \setminus k, a, b, c, d, X_i)$$

onde $f(\Delta_i)$ é a densidade marginal de Δ_i , fazendo $\sum_{k=1}^K P(k \setminus \Delta_i) = 1$.

Como pode-se notar, sem grande esforço, a implementação e o custo computacional da expressão (2.10), principalmente pela soma quádrupla de K classes para cada "pixel", pode tornar essa abordagem impraticável. O que se tem feito é tentar uma modelagem

das probabilidades g e h , de modo a tentar reduzir e simplificar a expressão (2.10). Para h , vários autores, como Welch and Salter [3], Swain, Vanderman and Tilton [8] e Owen [14] assumiram os vetores X como sendo condicionalmente independentes dado as classes:

$$\begin{aligned} f(\Delta_i \setminus k, a, b, c, d) &= f_k(X_i) f_a(X_{iN}) f_b(X_{iE}) f_c(X_{iS}) f_d(X_{iW}) ; \\ h(\Delta_i \setminus k, a, b, c, d) &= f_a(X_{iN}) f_b(X_{iE}) f_c(X_{iS}) f_d(X_{iW}) \end{aligned} \quad (2.11)$$

Os autores acima usaram densidades de classes multivariadas Gaussianas, ou Normais, com matrizes de covariâncias iguais [3], proporcionais [14] ou diferentes [8].

Para g , Owen [14] assumiu que a natureza empregou um particular modelo de Campo de Poisson descritos por Switzer [51], quando dividiu o terreno observado em classes e formulou g em termos de um parâmetro que denominou β . Owen and Switzer [10] usaram o modelo de probabilidade geométrica para g , mantendo a premissa do Campo de Poisson. Em Hjort and Mohn [16] e Hjort [17] é dada uma generalização do modelo de Owen and Switzer [10]. Este modelo será o estudado na próxima seção.

2.2.1. O modelo Owen & Switzer generalizado

O ponto de partida é tentar reduzir o número de termos, K^4 , na soma:

$$R_K(\Delta_i) = \sum_{a, b, c, d} h(\Delta_i \setminus k, a, b, c, d, X_i) g(a, b, c, d \setminus k) \quad (2.12)$$

Esta soma é dada sobre todas as possíveis configurações na

cruz. Partindo da premissa que existam situações nas quais os "pixels" são em área pequenos se comparados com a extensão das classes [14], nestes casos se observaria que os "pixels" na cruz seriam de uma mesma classe; ou em alguns casos, de duas classes; ou ainda em um número reduzido de casos de três, bastante reduzido de quatro classes e muito raramente de cinco classes. Assim, a premissa que será adotada é que o número de classes presentes na cruz é no máximo duas classes. Portanto, somente para estes casos, assumiremos que a probabilidade seja maior que zero (eventualmente poderá ser zero também para alguns destes casos). Adotando esta sistemática, o número de termos sofrerá uma redução para $8K-7$, o que é bastante significativo.

A idéia inicial é a premissa de que a natureza dividiu a cena em regiões. Essas regiões são grandes se comparadas ao tamanho dos "pixels". Assumindo ainda que a maioria das cruzes de "pixels" possuem pontos pertencentes no máximo a duas classes, na figura 2.2 são ilustrados os três tipos possíveis de cruzes com probabilidade maior (ou eventualmente igual) a zero:

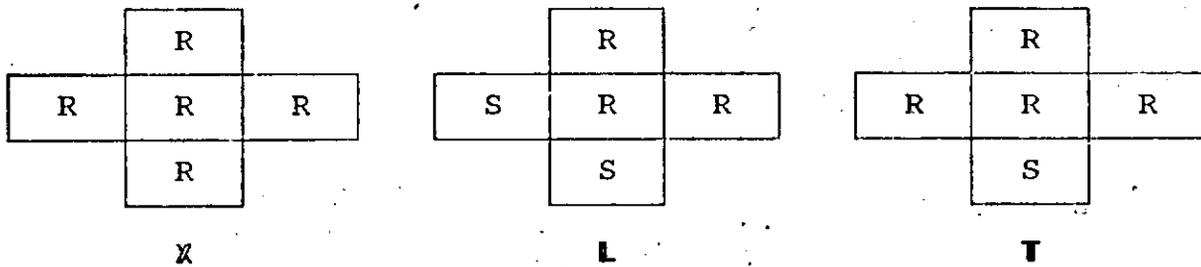


Figura 2.2 - Padrões X, L e T.

Pode-se observar que para X existe somente um padrão - o padrão da figura 2.2 - e para L e T, quatro padrões possíveis pela

rotação dos padrões R e S. Assim, sendo p a probabilidade do padrão X , $q/4$ a probabilidade de cada padrão L (q é a probabilidade do conjunto de padrões L) e $r/4$ de cada padrão T (raciocínio análogo a q), temos que $p + q + r = 1$, pela restrição do espaço de todos os padrões possíveis.

A seguir, a natureza rotula $1, \dots, K$ regiões, com probabilidades a priori $\pi(1), \dots, \pi(K)$, independentes para cada região. Assim, pode ocorrer que duas regiões vizinhas estejam numa mesma classe. Considerando os padrões X, L e T , tem-se que [17]:

$$\left. \begin{aligned} \pi(k) g(k, k, k, k \setminus k) &= p \pi(k) + \frac{1}{4} q \pi(k)^2 + \dots + \frac{1}{4} r \pi(k)^2 \\ &= \pi(k) \{ p + (q + r) \pi(k) \} \\ \pi(k) g(k, k, m, m \setminus k) &= \frac{1}{4} q \pi(k) \pi(m) \\ \pi(k) g(k, k, m, k \setminus k) &= \frac{1}{4} r \pi(k) \pi(m) \end{aligned} \right\} \quad (2.13)$$

para $m, k = 1, 2, \dots, K; m \neq k$.

Então:

$$\left. \begin{aligned} g(k, k, k, k \setminus k) &= p + (q + r) \pi(k) \\ g(k, k, m, m \setminus k) &= g(m, k, k, m \setminus k) = g(m, m, k, k \setminus k) \\ &= g(k, m, m, k \setminus k) = \frac{1}{4} q \pi(m) \\ g(k, k, m, k \setminus k) &= g(k, k, k, m \setminus k) = g(m, k, k, k \setminus k) \\ &= g(k, k, m, k \setminus k) = \frac{1}{4} r \pi(m) \end{aligned} \right\} \quad (2.14)$$

para $m, k = 1, 2, \dots, K; m \neq k$. Somente esses $8K-7$ termos são positivos.

Esta especificação para g e a especificação (2.11) para h

determinam um procedimento de classificação contextual. De acordo com as expressões (2.8) e (2.10), a regra manda encontrar a classe k que tenha o maior valor de

$$\pi(k) f_k(X_i) R_k(\Delta_i) \quad (2.15)$$

onde $R_k(\Delta_i)$ é descrito pela equação (2.12) e é interpretado como um fator de correção espacial de um procedimento não-contextual.

Pela especificação (2.11), $R_k(\Delta_i)$ pode ser reescrito como:

$$\begin{aligned} R_k(\Delta_i) &= \sum_{a,b,c,d} g(a,b,c,d \setminus k) f_a(X_{iN}) f_b(X_{iE}) f_c(X_{iS}) f_d(X_{iW}) \\ &= \{p + (q+r) \pi(k)\} f_k(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) f_k(X_{iW}) \\ &\quad + \frac{1}{4} q \sum_{m \neq k} \pi(m) f_k(X_{iN}) f_k(X_{iE}) f_m(X_{iS}) f_m(X_{iW}) \\ &\quad + \frac{1}{4} q \sum_{m \neq k} \pi(m) f_m(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) f_m(X_{iW}) \\ &\quad + \frac{1}{4} q \sum_{m \neq k} \pi(m) f_m(X_{iN}) f_m(X_{iE}) f_k(X_{iS}) f_k(X_{iW}) \\ &\quad + \frac{1}{4} q \sum_{m \neq k} \pi(m) f_k(X_{iN}) f_m(X_{iE}) f_m(X_{iS}) f_k(X_{iW}) \\ &\quad + \frac{1}{4} r \sum_{m \neq k} \pi(m) f_k(X_{iN}) f_k(X_{iE}) f_m(X_{iS}) f_k(X_{iW}) \\ &\quad + \frac{1}{4} r \sum_{m \neq k} \pi(m) f_k(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) f_m(X_{iW}) \end{aligned}$$

$$\begin{aligned}
 & + \frac{1}{4} r \sum_{m \neq k} \pi(m) f_m(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) f_k(X_{iW}) \\
 & + \frac{1}{4} r \sum_{m \neq k} \pi(m) f_k(X_{iN}) f_m(X_{iE}) f_k(X_{iS}) f_k(X_{iW})
 \end{aligned} \tag{2.16}$$

Depois de manipulações algébricas, [17], chega-se a:

$$R_k(\Delta_i) = p A_k(\Delta_i) + q B_k(\Delta_i) + r C_k(\Delta_i) \tag{2.17}$$

onde :

$$A_k(\Delta_i) = f_k(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) f_k(X_{iW})$$

$$\begin{aligned}
 B_k(\Delta_i) = \frac{1}{4} \left[& f_k(X_{iN}) f_k(X_{iE}) b(X_{iS}, X_{iW}) \right. \\
 & + f_k(X_{iS}) f_k(X_{iW}) b(X_{iN}, X_{iE}) + \\
 & + f_k(X_{iE}) f_k(X_{iS}) b(X_{iW}, X_{iN}) + \\
 & \left. + f_k(X_{iW}) f_k(X_{iN}) b(X_{iE}, X_{iS}) \right]
 \end{aligned}$$

$$\begin{aligned}
 C_k(\Delta_i) = \frac{1}{4} \left[& f_k(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) a(X_{iW}) + \right. \\
 & + f_k(X_{iN}) f_k(X_{iE}) a(X_{iS}) f_k(X_{iW}) + \\
 & + f_k(X_{iN}) a(X_{iE}) f_k(X_{iS}) f_k(X_{iW}) + \\
 & \left. + a(X_{iN}) f_k(X_{iE}) f_k(X_{iS}) f_k(X_{iW}) \right]
 \end{aligned} \tag{2.18}$$

e onde :

$$\begin{aligned}
 a(x) &= \sum_{m=1}^k \pi(m) f_m(x) \\
 b(x,y) &= \sum_{m=1}^k \pi(m) f_m(x) f_m(y)
 \end{aligned} \tag{2.19}$$

Obviamente, o modelo ainda não está completo. Há a necessidade

de se estimar os parâmetros p , q , r , $\pi(k)$ e $f_k(x)$. E é exatamente isto que será visto na seção a seguir.

2.2.2. Estimativa de Parâmetros

Como já foi mencionado, o modelo que será objetivo central deste estudo será o que comporta quatro vizinhos em cruz. Inicialmente, será mostrado uma forma de obtenção das densidades de probabilidade de classe $f_k(X)$. Depois a estimativa das probabilidades a priori $\pi(k)$ e dos parâmetros p, q, r . Essa divisão se dá porque como já foi visto na tabela 1.2, são necessários dois tipos de amostragem distintos para a estimativa dos parâmetros, dependendo da sua necessidade.

2.2.2.a. Densidades das classes

Normalmente, implicitamente ou explicitamente, as densidades individuais das classes são consideradas normais multivariadas da forma:

$$f_k(X) \sim N_d(\mu_k, \Sigma_k)(X) \quad (2.20)$$

com densidade dada por:

$$f(X) = \frac{1}{(2\pi)^{d/2}} |\Sigma_k|^{-1/2} \exp\left[-\frac{1}{2} (X-\mu_k)' \Sigma_k^{-1} (X-\mu_k)\right] \quad (2.21)$$

onde d é a dimensão do vetor observado X e π é o número "pi".

No entanto, esta regra pode ter exceções, principalmente porque em alguns casos algumas classes podem apresentar "sub-classes" implícitas. Isto pode ser percebido no histograma da classe, indicado pela presença de várias modas, ou seja, pode ser

visto várias curvas no histograma. Nestes casos, pode-se também modelar $f_k(X)$ como uma mistura de Gaussianas.

Usualmente, dispõe-se de pelo menos um conjunto de treinamento para cada classe. Então considera-se a existência de vetores observados $X_{\sim 1}^{(k)}, \dots, X_{\sim n_k}^{(k)}$ de "pixels" dos quais são conhecidas as verdadeiras classes $C_i \in \{1, 2, \dots, K\}$. Através destes conjuntos de treinamento, geralmente escolhidos em áreas homogêneas situadas seguramente dentro das classes apropriadas, serão obtidas as densidades $f_k(X)$.

Dependendo do tamanho das amostras e conhecimentos prévios sobre as classes, pode-se trabalhar sobre pelo menos três possíveis modelos:

- (i) $f_k(X) \sim N_d(\mu_k, \Sigma)(X)$
- (ii) $f_k(X) \sim N_d(\mu_k, \lambda_k \Sigma)(X)$ (2.22)
- (iii) $f_k(X) \sim N_d(\mu_k, \Sigma_k)(X)$

No último caso, (iii), a estimativa é de uma normal d-dimensional com média μ_k e matriz de covariâncias Σ_k para cada classe k. Pode-se estimar μ_k e Σ_k , pelos estimadores:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} X_j^{(k)} \quad (2.23)$$

$$\hat{\Sigma}_k = \frac{1}{n_k - 1} (X_j^{(k)} - \hat{\mu}_k)(X_j^{(k)} - \hat{\mu}_k)' \quad (2.24)$$

No caso (i), a obtenção de $\hat{\mu}_k$ é dada em (2.23) e a matriz de covariâncias Σ é estimada por:

$$\hat{\Sigma} = \sum_{k=1}^K \left(\frac{n_k}{n} \right) \hat{\Sigma}_k \quad (2.25)$$

onde $\hat{\Sigma}_k$ é estimado por (2.24) e $n = \sum_{k=1}^K n_k$.

Já no caso (ii), pode-se usar um dos muitos procedimentos iterativos para a obtenção de soluções de máxima verossimilhança $\tilde{\lambda}_k$ e $\tilde{\Sigma}$. Em Owen [14] pode-se ver no apêndice 2, um procedimento computacional para o cálculo das matrizes de covariâncias proporcionais.

2.2.2.b. Probabilidades a priori e parâmetros contextuais (caso de quatro vizinhos)

As probabilidades a priori $\pi(k)$ são usualmente tomadas com distribuição uniforme do tipo $\pi(k) = 1/K$, ou adaptadas de experiências anteriores. Porém, estatisticamente falando, é bem melhor que a estimativa de $\pi(k)$ seja a melhor possível, visto que há uma influência direta da probabilidade a priori $\pi(k)$ sobre a probabilidade a posteriori $P(k \mid \Delta_i)$, como visto em (2.10). Por isso, serão usados aqui estimadores de $\pi(k)$ com base em amostras tomadas na própria imagem.

Considera-se que a *verdade terrestre* provém de M amostras de cruces de "pixels". Idealmente, essas M cruces deveriam se constituir em uma amostra aleatória, mas na maioria das vezes isso não é possível, devido principalmente às distâncias entre o pesquisador e suas amostras. Essas cruces não podem vir das amostras de treinamento das densidades $f_k(X)$, devido à homogeneidade.

Seja C_{ij} a classe de um "pixel" arbitrário j na cruz de número i . Convenientemente, serão expressos os estimadores com a ajuda da função indicadora:

$$I(C_{ij}, k) = \begin{cases} 1, & \text{se } C_{ij} = k \\ 0, & \text{caso contrário} \end{cases} \quad (2.26)$$

onde $i = 1, \dots, n$; $j = 1, 2, 3, 4, 5$ correspondente aos "pixels" da cruz conforme mostra a figura 2.3:

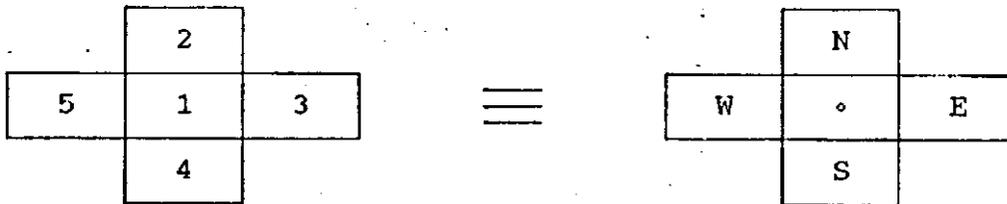


Figura 2.3 - Correspondência e localização dos "pixels" na cruz.

O estimador natural para a probabilidade a priori da classe k , $\pi(k)$ é :

$$\hat{\pi}(k) = \frac{1}{5M} \sum_{i=1}^M \sum_{j=1}^5 I(C_{ij}, k) \quad (2.27)$$

que nada mais é que a contagem das ocorrências da classe k nas M cruces amostradas dividido pelo número total de "pixels" que é $5M$, resultando em uma proporção dos "pixels" da classe k na amostra tomada.

Para a estimativa de p , q , r , das M cruces amostradas, tomam-se M_x cruces como as que tem cinco "pixels" de alguma classe k . Analogamente, M_l cruces para as que tem três "pixels" de uma classe k e dois de uma classe m qualquer e, M_r cruces para as que tem quatro "pixels" de uma classe k e um de uma classe m , como na figura 2.2. As demais cruces que porventura sejam amostradas,

devem ser descartadas. Redefine-se então M como sendo: $M = M_x + M_L + M_I$. Estima-se a razão M_x / M por:

$$\begin{aligned} \left(\frac{\hat{M}_x}{M} \right) &= \pi(k) g(k, k, k, k \setminus k) = \sum_{k=1}^K \pi(k) \{ p + (1 - p) \pi(k) \} \\ &= p + (1 - p) w \end{aligned} \quad (2.28)$$

onde $w = \sum_{k=1}^K \pi(k)^2$.

As estimativas para M_L / M e M_I / M são respectivamente:

$$\left(\frac{\hat{M}_L}{M} \right) = \sum_{k=1}^K \sum_{\substack{m=1 \\ m \neq k}}^K \pi(k) q \pi(m) = q (1 - w) \quad (2.29)$$

$$\left(\frac{\hat{M}_I}{M} \right) = \sum_{k=1}^K \sum_{\substack{m=1 \\ m \neq k}}^K \pi(k) r \pi(m) = r (1 - w) \quad (2.30)$$

Assim sendo, obtêm-se os estimadores para p, q, r :

$$\hat{q} = \frac{M_L / M}{1 - \hat{w}} ; \quad \hat{r} = \frac{M_I / M}{1 - \hat{w}} ;$$

$$\hat{p} = 1 - \hat{q} - \hat{r} = \frac{M_x / M - \hat{w}}{1 - \hat{w}} ; \quad (2.31)$$

onde $\hat{w} = \sum_{k=1}^K \hat{\pi}(k)^2$; estimador de w .

Segundo Hjort [17], a forma dos estimadores acima não depende da forma de amostragem dos M "pixels", mas suas propriedades sim.

O método descrito acima requer a posse de um conjunto representativo de "pixels" externo aos conjuntos de treinamento das classes, conforme já foi mencionado. Isto porém, em certos casos, é uma tarefa difícil e algumas vezes praticamente impossível.

Hjort [17] dá um procedimento iterativo alternativo para contornar este problema, usando vetores não classificados e equações de máxima verossimilhança para estimar p, q, r . Obviamente, será menos preciso que a forma anterior. Outra alternativa, comumente usada na prática é utilizar uma pré-classificação, ou seja, fazer uma classificação ponto-a-ponto da imagem por algum método de classificação, geralmente supervisionado e a seguir, refinar a solução usando um método contextual, pela sua maior precisão.

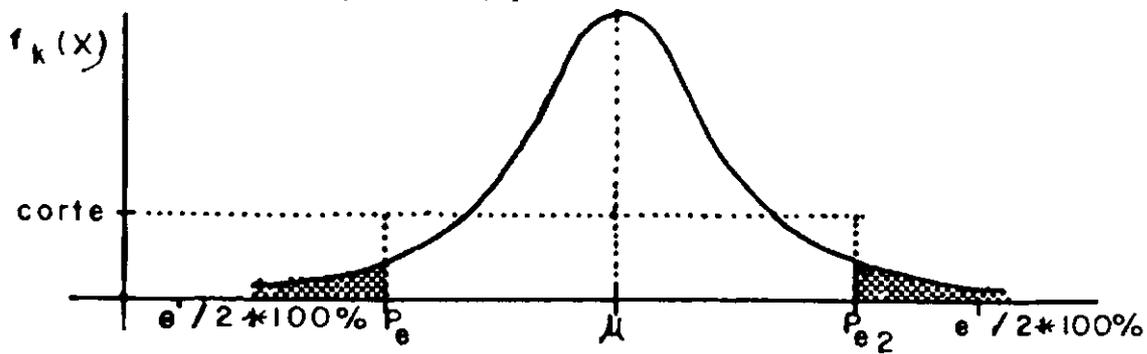
2.2.3. O limiar (1-e)

O limiar (1-e) contido na expressão (2.8), pode ser escolhido pela conveniência do usuário, podendo ser inclusive um vetor com valores distintos, ponderando diferentemente as classes em questão. Pode-se ressaltar também que para deixar o usuário mais à vontade com o sistema, a premissa que a soma dos valores deve resultar na unidade foi abolida, podendo o valor de entrada ser qualquer valor no intervalo $(0,1]$. Isto altera a filosofia do limiar de rejeição contido na expressão (2.8). Neste caso, como (1-e) será escolhido pelo usuário e um valor alto pode significar uma ou mais classes sem pontos classificados em seu nome, será utilizado um processo que ponderará seu valor para que este se

ajuste a cada caso.

Pela própria expressão (2.15) pode-se notar o que foi mencionado. A expressão contém três fatores: $\pi(k)$, $f_k(X_i)$ e $R_k(\Delta_i)$. O primeiro deverá ser um valor na ordem de 10^{-1} , ou no máximo 10^{-2} . O segundo deverá ser um valor de ordem menor ou igual a 10^{-1} , podendo atingir valores muito pequenos. O terceiro, como se trata de várias multiplicações de densidades $f_k(X_i)$, deverá atingir valores de no mínimo 10^{-4} , o que pode ser comprovado pela expressão (2.16). Portanto, os valores da expressão (2.15) deverão ser na ordem de no mínimo 10^{-6} . Com isto, um valor de corte e de ordem maior que esta significa nenhum ponto classificado na classe dada, pelo simples fato que numericamente é impossível existir um valor nestas condições. Portanto, deve-se fazer uma transformação, para que e seja um corte efetivo para a classe em questão.

Sendo então um corte $e'=(1-e)$ definido pelo usuário, geometricamente ele será interpretado como uma fatia (no caso unidimensional) da distribuição a ser descartada. Por isto, se for tomada a probabilidade de $e'/2$ na Normal padrão $[0,1]$, multiplicada ao seu desvio-padrão e subtraída sua média estimada na amostragem de áreas homogêneas, resulta em um novo ponto p_e cuja densidade equivalerá à área da distribuição a ser descartada. Estatisticamente falando, tomar a densidade no ponto p_e significa descartar as caudas da distribuição normal na faixa de probabilidade $\alpha = (1 - e'/2) * 100\%$, restando o corpo central da densidade como pode ser visto na figura 2.4, no caso unidimensional:



onde $p_e = \mu - Z(e') * \sigma$; e $p_{e2} = \mu + Z(e') * \sigma$

 zona de exclusão.

Figura 2.4 - Corte efetivo da distribuição dado pela transformação.

No caso n-dimensional, esta estratégia significa eliminar o volume proporcionado pela cauda no espaço, restando o volume do corpo central - Figura 2.5(a) e (b).

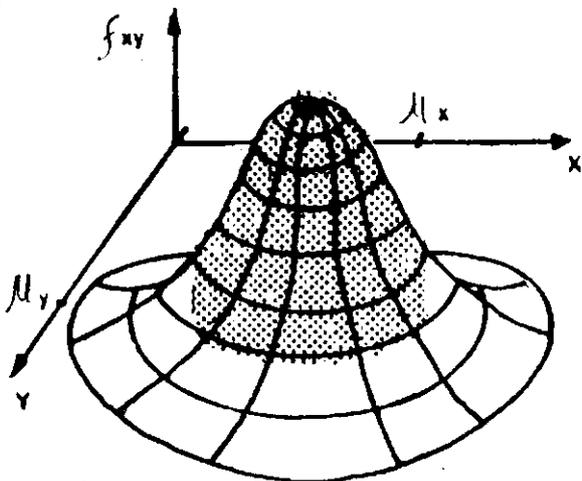


Figura 2.5 (a) Região do "sino" é a região admitida % no caso multivariado.

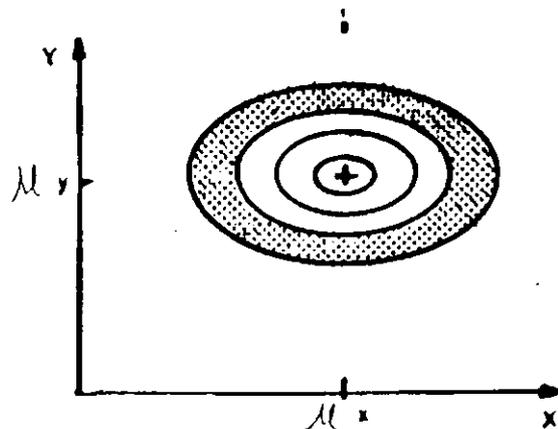


Figura 2.5 (b) Vista da região rejeitada nas curvas de níveis; caso multivariado

No entanto, para isto seria necessário o cálculo de integrais de volume n-dimensionais. Para se evitar isto, considerou-se a seguinte estratégia: se for tomada cada densidade unidimensional

ao invés da multivariada, o volume a ser considerado será de no mínimo $\alpha\%$, já que o volume se tornará um cubo e não mais um elipsóide - Figura 2.6.

Em um caso mais geral, se houver um limiar para cada classe, teremos $(1-e_k)$. Neste caso, a operação será:

$$\text{corte}_k = f_k \left[\hat{\mu}_k \pm Z(e_k') * \hat{\Sigma}_{kk}^{1/2} \right] \quad (2.32)$$

onde $\hat{\mu}_k$ é o vetor de médias da classe k e $\hat{\Sigma}_{kk}$ é a matriz de covariâncias, tomada apenas a sua diagonal. Assim, o corte_k será o valor a ser excedido na densidade multivariada para que o algoritmo possa classificar o "pixel" i em alguma das classes disponíveis - Figura 2.5(a).

O corte_k dado por (2.32) substituirá $(1-e)$ na expressão (2.8), sendo fixado valores "default", para o caso do usuário que não quiser ponderar cada classe de forma independente, ou cujo interesse pese de forma uniforme sobre as classes em questão. A visualização fica a encargo da figura 2.6:

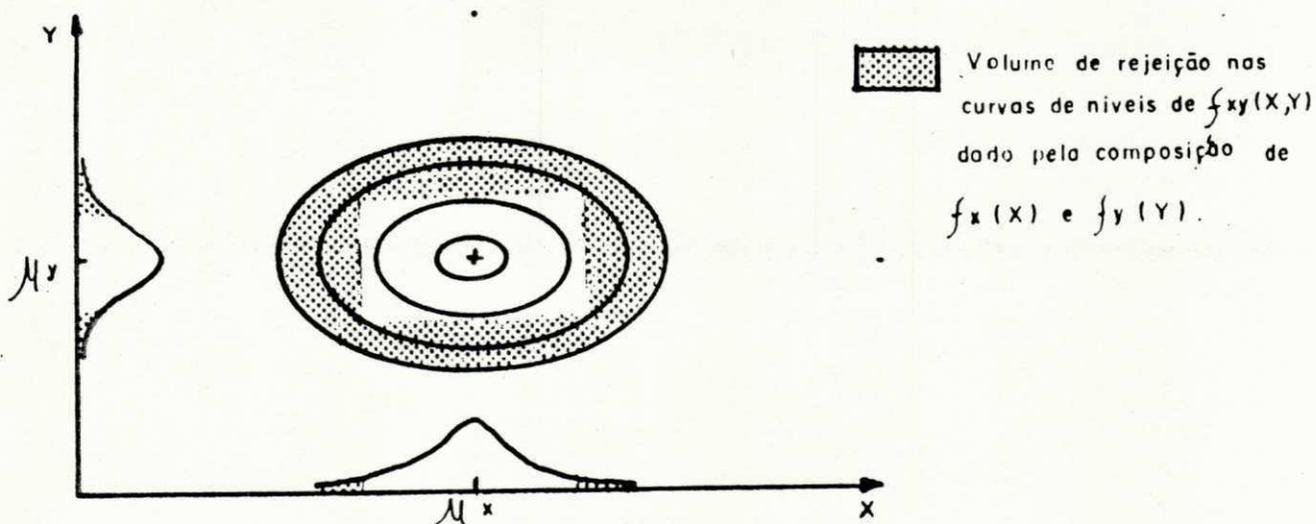
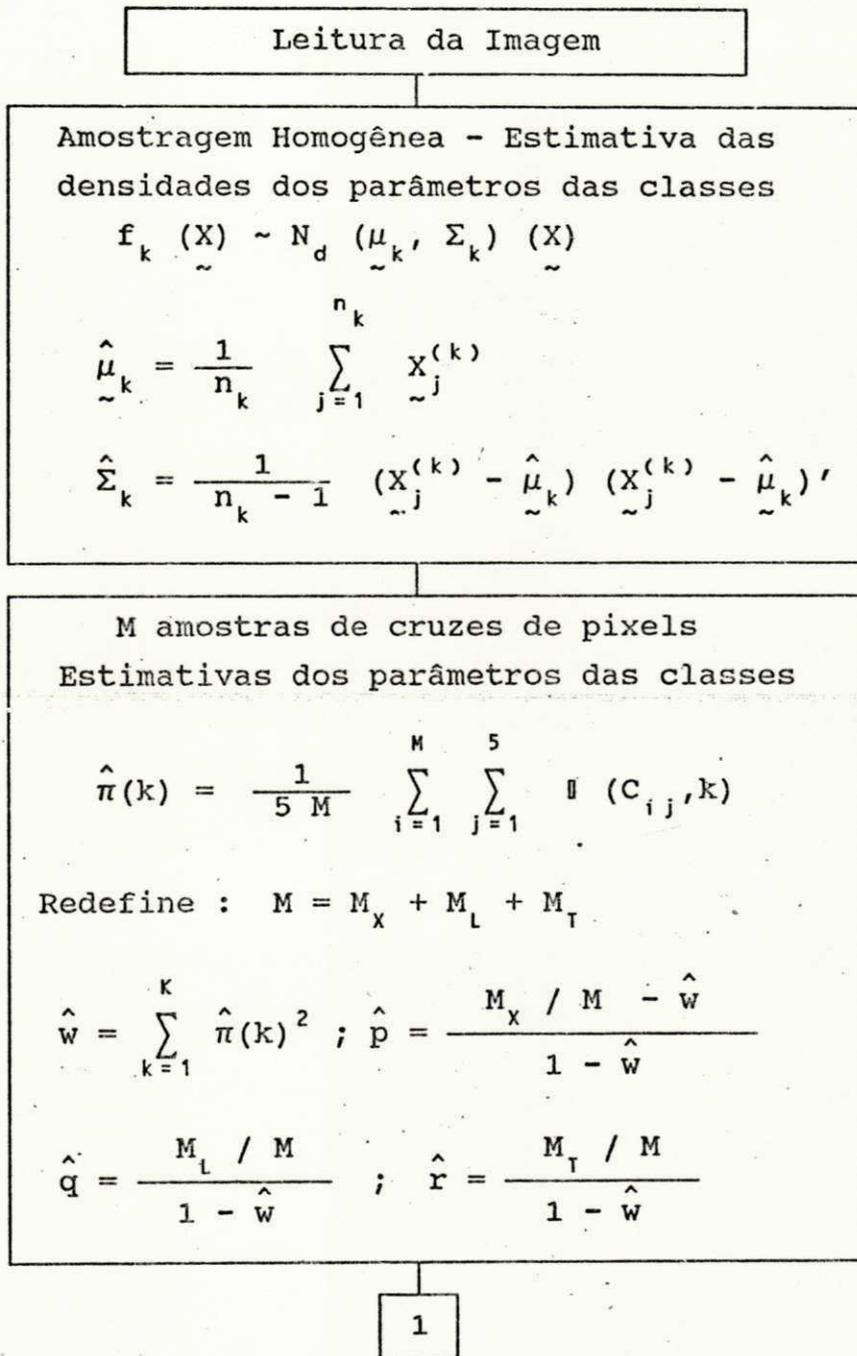


Figura 2.6-Corte efetivo da distribuição dado pela transformação no espaço.

Para finalizar este capítulo, mostra-se a seguir um fluxograma remissivo do método que será implementado, como uma forma de síntese.



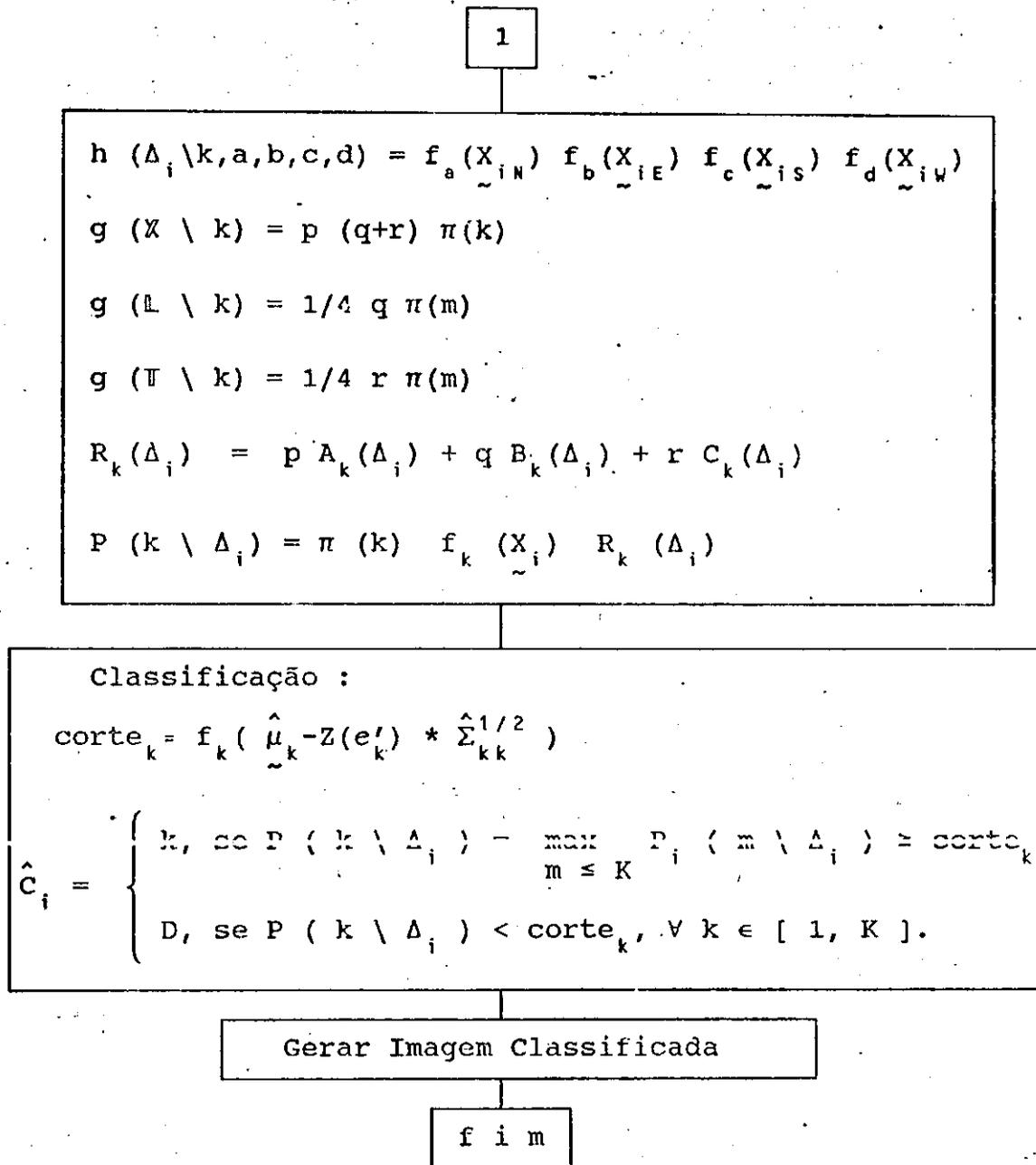


Figura 2.7 - Fluxograma do Método Contextual p,q,r a ser implementado.

CAPÍTULO 3

Delineamento e Implementação do Sistema

3.1. Introdução

Conforme já foi mencionado anteriormente, objetivou-se com este trabalho implementar um sistema de classificação contextual compatível com o SITIM-150 desenvolvido pelo INPE e produzido pela Engespaço de São José dos Campos - S.P.

Com a preocupação de torná-lo compatível com as versões futuras do sistema SITIM, ou torná-lo facilmente compatível, a linguagem de programação escolhida foi a linguagem C. A estrutura básica do programa é a que pode ser vista na figura 2.7 do capítulo 2. Conforme a necessidade, adiante, o fluxograma da figura 2.7 será refinado, mostrando os passos de forma mais detalhada. Convém adiantar que algumas rotinas deste programa usarão informações fornecidas por programas já implementados no sistema SITIM. A leitura de imagens será feita pelos procedimentos do sistema, assim como a aquisição das amostras homogêneas, facilitando a adaptação do usuário já acostumado ao sistema, à utilização do novo módulo sem grandes alterações na forma de trabalho.

3.2. Descrição do sistema

O fluxograma da figura 2.7 mostra o sistema isolado. No entanto, para a adaptação ao SITIM, algumas rotinas que já fazem parte do sistema foram utilizadas, reduzindo assim a redundância e

permitindo uma adaptação mais rápida do usuário ao novo módulo.

Por estes motivos, foram aproveitadas as rotinas de leitura, e as de aquisição de amostras homogêneas do classificador de Máxima Verossimilhança - MAXVER. As características desta rotina não serão alteradas, podendo o sistema classificar, como antes, até 20 classes diferentes e adquirir até 100 amostras - ver SITIM [54].

A aquisição das M amostras de cruces de pixels para a estimativa dos parâmetros das classes foi implementada. Esta rotina é capaz de amostrar cruces de pixels diretamente sobre a imagem, podendo segundo as premissas do modelo, ter até duas classes distintas em cada cruz. O sistema deixa a cargo do usuário a definição da classe, devendo esta obrigatoriamente estar incluída entre as classes amostradas anteriormente com as rotinas MAXVER. A estimativa das probabilidades a priori segue a equação (2.27), rerepresentada e renumerada abaixo:

$$\hat{\pi}(k) = \frac{1}{5M} \sum_{i=1}^M \sum_{j=1}^5 \Pi(C_{ij}, k) \quad (3.1)$$

Se acaso o usuário não tomar amostras para todas as classes contidas no arquivo MAXVER, o programa avisará o usuário da falta das classes não amostradas. Isto se deve ao fato de que as probabilidades a priori influem de modo significativo na classificação.

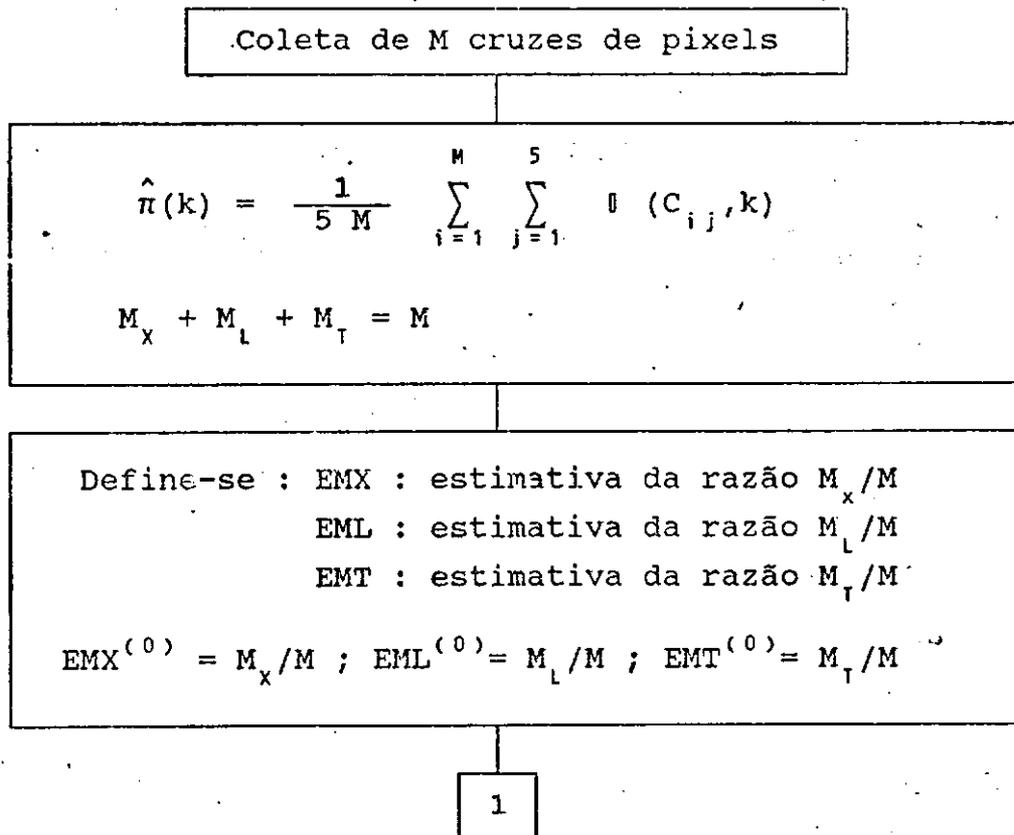
Procedendo desta forma, o sistema não necessitará de redefinir M, como a soma $M = M_x + M_L + M_T$, ou seja, não desprezará amostras, tornando o trabalho do usuário inútil na coleta das mesmas.

A primeira estimativa das razões M_x / M , M_L / M e M_T / M é tomada como a própria razão das amostras, gerando assim a primeira

estimativa para p, q e r. A seguir, inicia-se o processo iterativo para o refinamento da estimativa, dado pelas equações (2.28), (2.29) e (2.30) para as razões M_x / M , M_L / M e M_T / M e pelas equações em (2.31) para p, q e r. O processo será interrompido quando o teste iterativo

$$\frac{|\hat{p}^{(i)} - \hat{p}^{(i-1)}| + |\hat{q}^{(i)} - \hat{q}^{(i-1)}| + |\hat{r}^{(i)} - \hat{r}^{(i-1)}|}{3} < \delta \quad (3.2)$$

for satisfeito. Foi adotado o valor de $\delta \leq 0.00001$ ou 1×10^{-5} como chave de saída. Este trecho do fluxograma pode ser visualizado na figura 3.1:



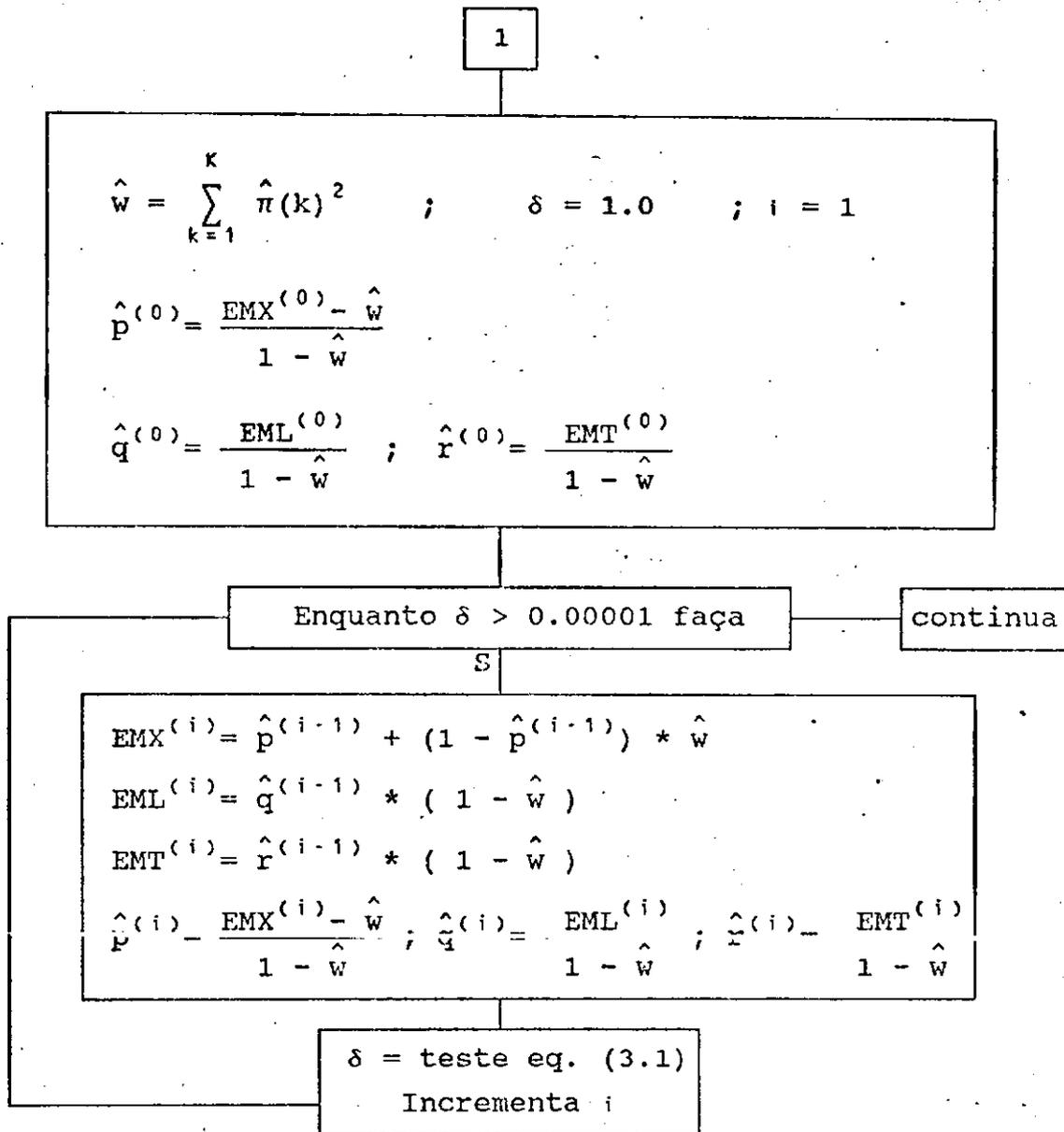


Figura 3.1 - Modificação na estrutura do fluxograma para a estimativa iterativa de p,q,r.

Pela forma adotada, a função $h(\cdot|\cdot)$ na equação (2.11) é tida como condicionalmente independente, e toma valores diretamente nas funções estimadas pela rotina do MAXVER.

A função $g(\cdot|\cdot)$ dada pela equação (2.14) terá seus valores

armazenados após o levantamento das M. cruzes e estimativa dos parâmetros p, q e r.

Finalmente, o fator de correção contextual $R_k(\Delta_i)$, dado pelas equações (2.17), (2.18) e (2.19) é calculado agrupando os resultados anteriores, já na forma dada por estas equações que já engloba as funções $g(\cdot \setminus \cdot)$ e $h(\cdot \setminus \cdot)$. Por este motivo, os cálculos das funções $g(\cdot \setminus \cdot)$ e $h(\cdot \setminus \cdot)$ estão implícitos no cálculo de $R_k(\Delta_i)$.

A forma de cálculo para a probabilidade a posteriori

$$P(k \setminus \Delta_i) = \pi(k) f_k(\tilde{X}_i) R_k(\Delta_i) \quad (3.3)$$

é usada para todas as K classes. Deste cálculo, será tomado o máximo valor, e comparado com o valor de corte k , calculado segundo a expressão (2.32), para a tomada da decisão segundo (2.8). É colocado também em aberto a opção de se alterar os valores de corte k para cada classe - no máximo de 20 para guardar compatibilidade com o MAXVER já implementado - para adequar o sistema a diferentes formas de utilização e permitindo maior flexibilidade e iteração com o usuário. A imagem gerada ao final da classificação, pode ser visualizada no canal gráfico ou exportada para disco, utilizando-se de rotinas já disponíveis.

O módulo segue a estrutura SITIM, permitindo ao usuário trabalhar de forma independente com cada programa. Por isto, o fluxograma será "quebrado" em várias partes. A aquisição das amostras homogêneas fica por conta das rotinas do MAXVER. As amostras em cruz, estão em uma estrutura denominada MPQR, que basicamente resume-se ao fluxograma da figura 3.1. Para a leitura dos pixels individualmente, foi feita uma adaptação da rotina

MIXAQU, que faz parte agora da rotina MPQR. O arquivo de saída CONTEXTO.CON contém informações sobre o número total de classes, as probabilidades a priori $\pi(k)$ e naturalmente p, q, r . Outra rotina denominado CONTEXT contém as funções $h(\cdot \setminus \cdot)$, $g(\cdot \setminus \cdot)$ e $R_k(\cdot)$, lendo as saídas dos arquivos MAXVER e MPQR, para calcular as referidas funções. A rotina FUNCAOE faz a ligação com o usuário com respeito ao corte k , calculando os valores "default" a partir do valor de K fornecido por MPQR e permitindo alterações, se necessário for. Esta rotina gera um arquivo de saída E.CON com os valores do corte k para leitura posterior.

A rotina seguinte, é o programa classificador MXVCON que se encarrega de calcular a posteriori e classificar cada pixel, utilizando o módulo CONTEXT como subrotina e ler a função e' do arquivo E.CON, fazendo as transformações necessárias para a sua utilização na classificação como um limiar de corte efetivo. Usa subrotinas para calcular $f_k(X)$, e as funções $a(x)$ e $b(x,y)$ dadas pela expressão (2.19). Ela também se encarrega de colocar de forma adequada a imagem para que as rotinas já residentes no SITIM possam se encarregar da visualização ou armazenamento da mesma.

Este programa é derivado do programa MAXVER, já disponível no sistema SITIM, com as adaptações necessárias para que o sistema não imponha a premissa estatística de classes com probabilidades iguais - isto é modificado pela rotina MPQR - e também levando em conta a correção contextual pelo fator $R_k(\Delta_i)$. Pelo fato de trabalhar com vizinhos e com vários planos de imagem ou bandas - Figura 3.2 - sua forma de leitura da imagem também foi alterado, utilizando-se a técnica de "buffer" rotativo. No entanto, por

questões de requisitos de memória, o número de bandas que o usuário poderá adquirir da imagem em uso é limitado a três e acomodadas em planos de memória que são denominados canais. O sistema utiliza um quarto canal exclusivamente como área de saída, onde será gerada a imagem classificada.

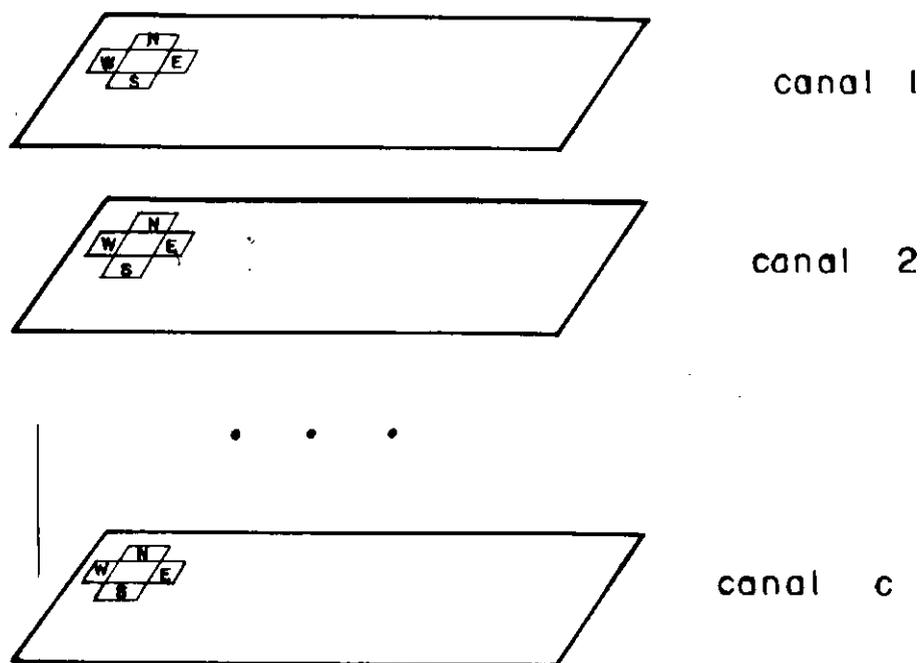
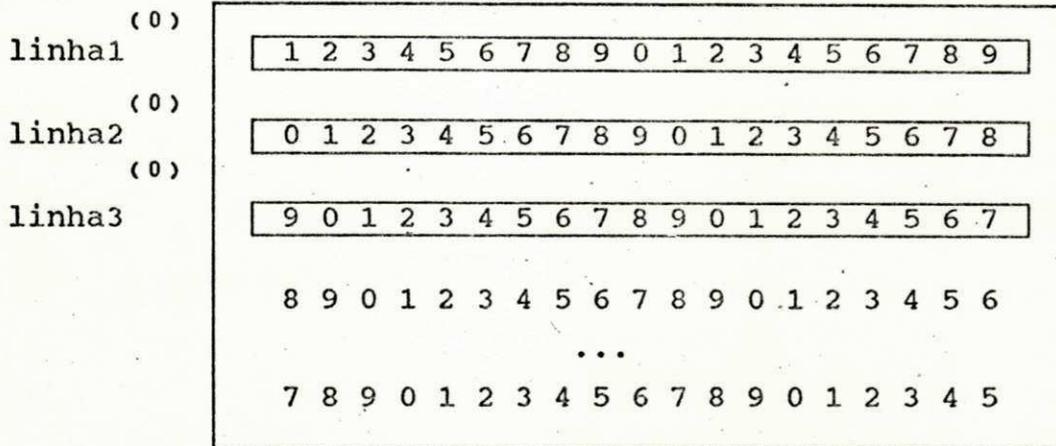


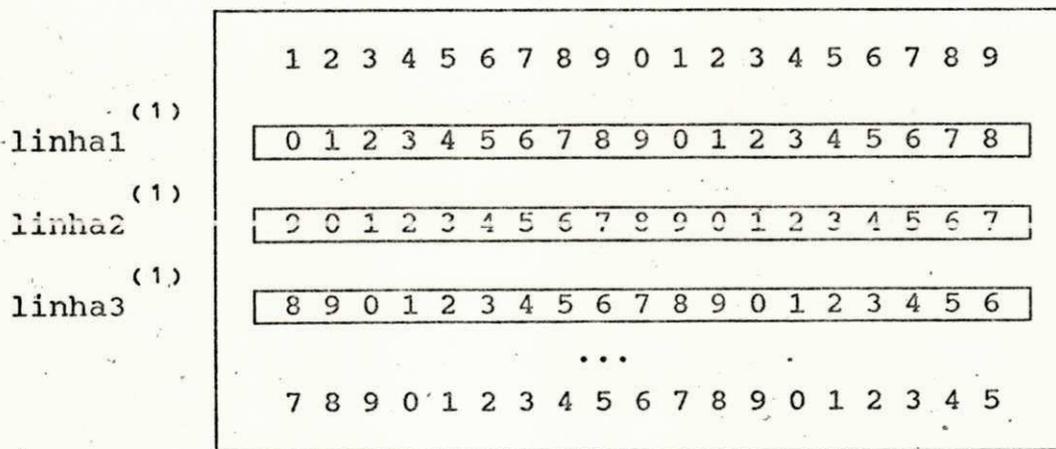
Figura 3.2 - Os vários planos ou bandas da imagem multiespectral e a cruz para cada plano ou banda.

Resumidamente, a técnica do "buffer" rotativo compõe-se de manter para cálculos um número fixo de linhas e a cada nova iteração, desprezar-se a primeira linha, rearranjando o "buffer" de forma que a nova linha a ser lida seja incorporada no final. Isto é mostrado pela figura (3.3), para o caso específico de três linhas para os cálculos das cruzes, como o que é utilizado pelo sistema contextual implementado:



fase (0) :o programa lê as três primeiras linhas para os cálculos em cruz para classificação.

Figura3.2a - Fase inicial do "buffer" rotativo.



fase (1) :após os calculos, a linha1⁽⁰⁾ deve ser retirada do "buffer" mas pode-se conservar as linhas 2⁽⁰⁾ e 3⁽⁰⁾ que serão necessárias aos cálculos da fase (1), agilizando o algoritmo.

O procedimento se repete até o final do arquivo.

Figura3.2b - "Buffer" rotativo em atividade de leitura.

3.3. Implementação

Será dada a seguir uma visão geral do sistema final, de modo que o leitor, da mesma forma que um usuário, possa visualizar este bloco do sistema. Isto também pode ajudar na utilização posterior das rotinas, já que é fundamental o seu uso na sequência correta, sob pena de um trabalho inutilizado, da mesma forma que as rotinas, por exemplo, do MAXVER. Enfim, estas próximas seções podem ser utilizadas como um manual para detalhamento e utilização deste sistema.

Para melhor compreensão e visualização, será mostrado o fluxo de operação do sistema implementado. Para o leitor familiarizado com a utilização do SITIM 150, a descrição será de fácil interpretação, o mesmo não podendo ser dito para os que não o conhecem. Para estes, recomenda-se uma leitura do manual do mesmo antes de prosseguir.

O sistema foi implementado sob sistema operacional MS-DOS versão 4.01 da Microsoft e linguagem de programação C, também da Microsoft, versão 5.01. Como já descrito, o aproveitamento das rotinas já incorporadas ao sistema SITIM, foi feito com objetivo de que o usuário final se sentisse mais a vontade com o novo sistema e também porque o seu aproveitamento criaria um sistema mais aperfeiçoado, pelo fato de muitas de suas rotinas já serem otimizadas e exaustivamente testadas.

A rotina implementada modifica o MAXVER no ponto da maximização, aumentando a potência do teste com a rotina de $R_k(\Delta_i)$ - contido na rotina CONTEXT - devendo ser feita previamente a modificação dos parâmetros de comparação corte_k vindos da rotina

FUNCAOE. Quebrando uma premissa estatística do MAXVER original¹, a inclusão das probabilidades a priori diferenciadas para as classes modifica os cálculos aperfeiçoando estatisticamente o sistema. Também modifica-se a coleta de amostras, com a adição do programa MIXAQU reformulado para a coleta de cruces e renomeado para MPQR. O MPQR é encarregado também do cálculo das probabilidades a priori e dos parâmetros p,q,r. Estas modificações originariam o fluxograma mostrado na figura 3.4:

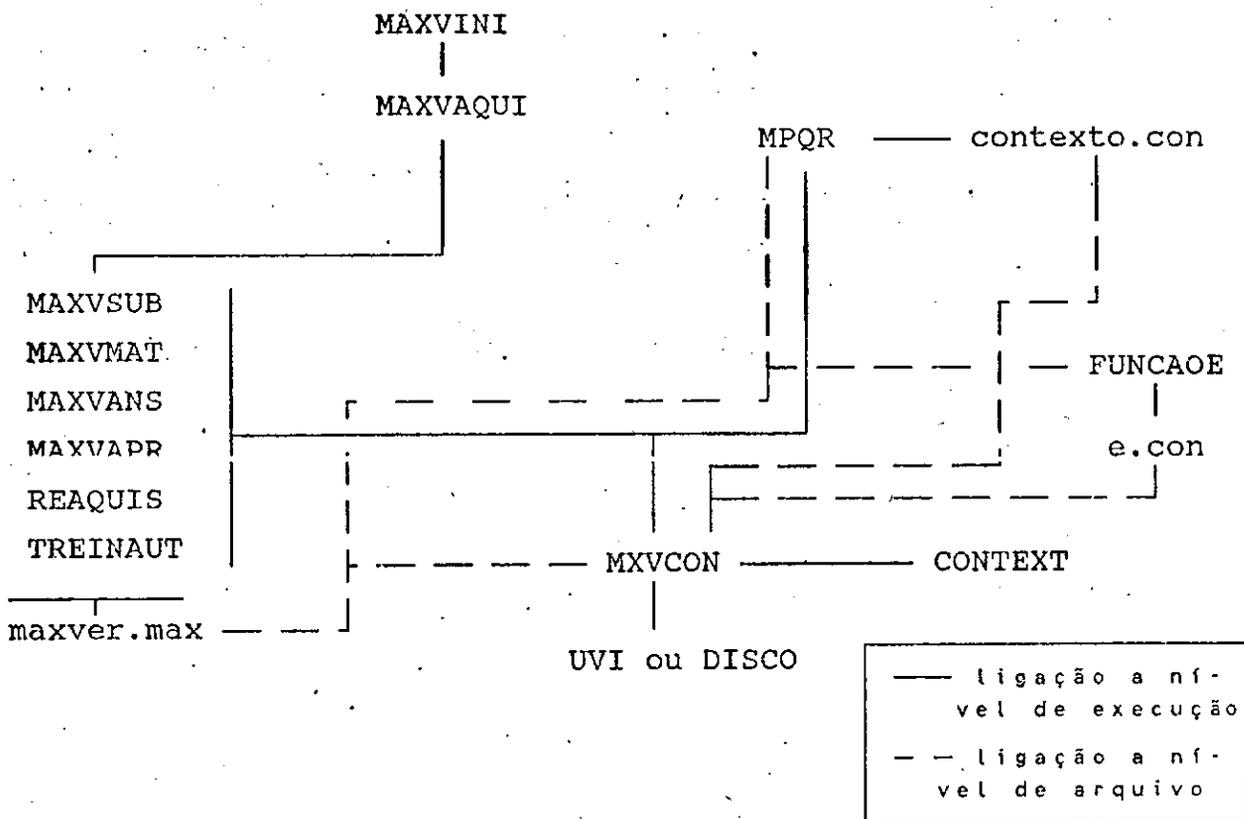


Figura 3.4 - Fluxograma do sistema contextual implementado.

onde : - MPQR é MIXAQU modificado para leitura das cruces e

¹ Nota: No MAXVER original, há a premissa estatística de que as probabilidades das classes sejam iguais.

cálculos dos parâmetros $\pi(k), p, q, r$. Comunica-se com `maxver.max` para aferição do número total de classes e nome das mesmas.

- `MXVCON` é `MAXVCLA` modificado incluindo $R_k(\Delta_i)$ e chamadas a subrotina `CONTEXT` e aos arquivos `e.con` e `contexto.con`.

- `FUNCAOE` corresponde-se com `maxver.max` para aferição do número total de classes e os nomes das mesmas.

O módulo segue a estrutura `SITIM`, permitindo ao usuário trabalhar com cada programa de forma independente. A aquisição das amostras homogêneas fica por conta da rotina `MAXVAQUI` do sistema `MAXVER`. Isto faz com que o usuário deva inicializar o sistema da mesma forma que inicializa o `MAXVER`, ou seja, com o `MAXVINI`. No fluxograma, a aparência de hierarquia sugerida é proposital. A sequência a ser utilizada inicia-se com o `MAXVINI`. Ele inicializará um arquivo de codi-nome `maxver.max`. Este arquivo poderá ser alterado com todas as rotinas disponíveis no `MAXVER`, conforme mostra o fluxo. Quando este arquivo não puder mais ser alterado, ou seja, finalizar o trabalho pontual, chama-se a rotina `MPQR`. Esta rotina, que incorpora o `MIXAQU`, faz o levantamento contextual, checando sempre a existência de parâmetros correspondentes no arquivo `maxver.max`, daí a restrição hierárquica - não será possível incluir uma classe no contexto que não esteja presente nas amostras homogêneas, o que acarretaria em falta de parâmetros no final.

O programa `MPQR` basicamente resume-se ao fluxograma da figura

3.1., tendo como arquivo de saída um arquivo de nome `contexto.con` com informações sobre o número de classes, probabilidades a priori de cada uma e os parâmetros p, q, r .

Seguindo a hierarquia, executa-se a rotina `FUNCAOE` para atualização dos parâmetros no arquivo `e.con`. Há a opção do usuário desbalancear o vetor para sua melhor conveniência, podendo impor uma restrição maior para uma ou mais classes em detrimento das demais. O usuário deverá observar que as componentes deste vetor devem estar compreendidas no intervalo $(0,1]$. Caso contrário, o programa emitirá uma mensagem alertando-o sobre o fato e pedindo correções.

O programa classificador `MXVCON` é a modificação do programa `MAXVCLA` com a inclusão de $R_k(\circ)$ e $\pi(k)$, chegando-se ao cálculo da probabilidade a posteriori - equação (2.15) - e posteriormente da equação (2.8), com a qual classifica o "pixel". Há de se observar que a primeira e última linhas da imagem, da mesma forma que a primeira e última colunas não serão classificadas por não se terem informações sobre seus vizinhos, uma vez que definida a área de classificação, o sistema passa a ter informações apenas desta nova área, excluindo o restante, se houver (no caso de se trabalhar com toda a imagem da tela). A forma de leitura da imagem também foi alterada, já que há a necessidade de uma leitura simultânea de três linhas para os cálculos das cruzes. Por isto foi incorporada uma rotina de "buffer" rotativo, que é na verdade a leitura de três linhas, fazendo-se as operações necessárias sobre elas, devendo-se para o próximo ciclo de operações eliminar-se a primeira linha, ajustar a segunda como a primeira e terceira como

a segunda. A nova terceira linha do "buffer" será a quarta linha da imagem a ser lida e, assim por diante, como mostra a figura 3.2. O MXVCON deve ler também informações dos arquivos *maxver.max* e *e.con*, utilizando a rotina CONTEXT como subrotina auxiliar nos cálculos do contexto.

Também foi feita uma modificação na rotina original do MAXVER para o cálculo da densidade $f_k(X)$. Nos cálculos do MAXVER, Velasco [55], a densidade a ser calculada

$$P[W_j \setminus X] = \frac{P[X \setminus W_j] P[W_j]}{P[X]} \quad (3.4)$$

segue uma distribuição normal multivariada, segundo a forma:

$$P[W_j \setminus X] = \frac{1}{(2\pi)^{n/2} |C_j|^{1/2}} \exp \left(-\frac{1}{2} (X - M_j)^t C_j^{-1} (X - M_j) \right) \quad (3.5)$$

Para otimização dos cálculos, considerou-se que $P(X)$ seja igual para todo X e que $P(W_j)$ seja equiprovável para todas as classes, fazendo com que a função de verossimilhança recaia, quando da maximização, para a seguinte equação:

$$P[W_j \setminus X] = P[X \setminus W_j] \quad (3.6)$$

Pela Máxima Verossimilhança, tem-se que:

$$P[W_j \setminus X] = -\frac{1}{2} (X - M_j)^t C_j^{-1} (X - M_j) - \frac{n}{2} \log 2\pi - \frac{1}{2} \log |C_j| \quad (3.7)$$

Como $\frac{n}{2} \log 2\pi$ é constante e considerando-se o sinal, basta minimizar a expressão :

$$P[W_j \mid X_j] = (\underline{X} - \underline{M}_j)^t C_j^{-1} (\underline{X} - \underline{M}_j) + \log |C_j| \quad (3.8)$$

Considerando-se que a *distância de Mahalanobis* ao quadrado é definida como:

$$r_j^2 = (\underline{X} - \underline{M}_j)^t C_j^{-1} (\underline{X} - \underline{M}_j) \quad (3.9)$$

então deve-se minimizar : $r_j^2 + \log |C_j| = d_j^2(\underline{X})$ (3.10)

Por álgebra linear, se a matriz C_j é matriz identidade, as elipses originadas pelo cálculo de r_j^2 se reduzem à distância Euclidiana do ponto X à origem ao quadrado, ou seja:

$$r^2 = (\underline{X}_1 - \underline{M}_1)^2 + (\underline{X}_2 - \underline{M}_2)^2 + \dots + (\underline{X}_n - \underline{M}_n)^2 \quad (3.11)$$

No caso do MAXVER, tomou-se ainda duas transformações, de modo a centralizar a nuvem de pontos. Inicialmente, fez-se

$$\underline{X}' = \underline{X} - \underline{M}_j \quad (3.12)$$

e depois $\underline{X}'' = T_j \underline{X}'$ (3.13)

onde T_j transforma a matriz C_j em matriz identidade. Com estas transformações, o cálculo realizado no MAXVER, para r^2 , passa a ser somente

$$r_j^2 = \left(\tilde{x}_1'' \right)^2 + \left(\tilde{x}_2'' \right)^2 + \dots + \left(\tilde{x}_n'' \right)^2 \quad (3.14)$$

A figura 3.5 ilustra o processo:

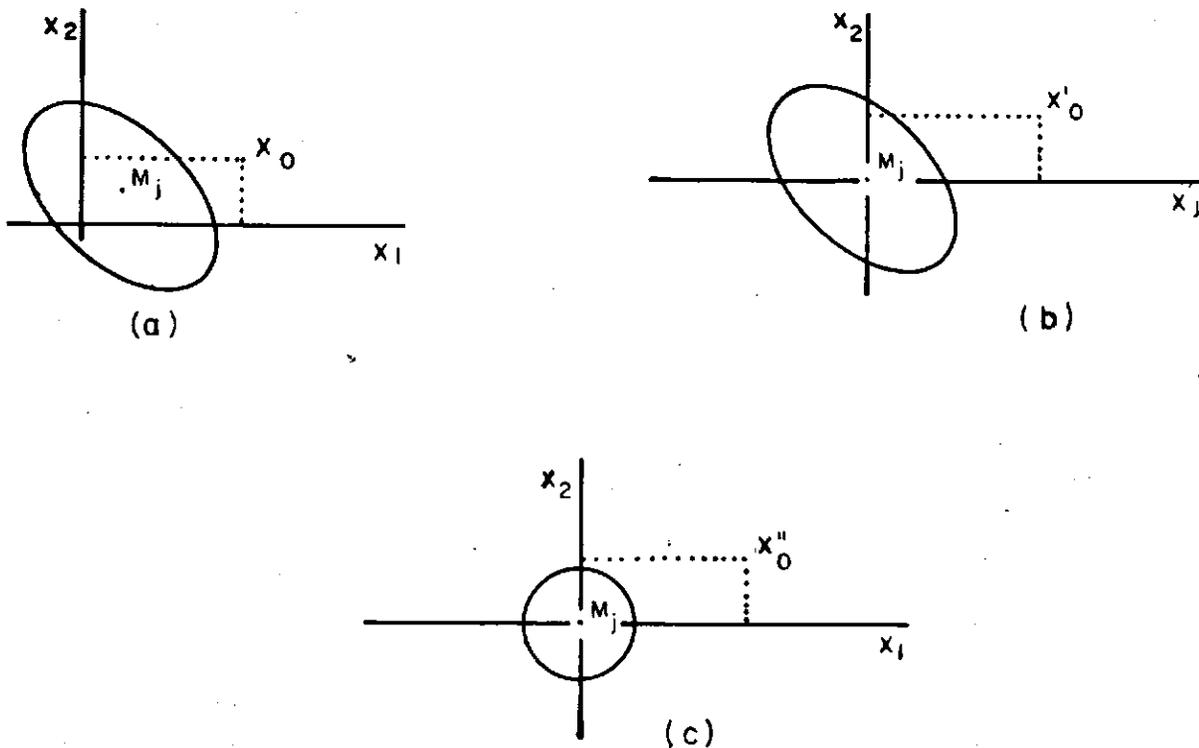


Figura 3.5 - Transformações no espaço dos atributos

Com estas simplificações, o cálculo de (3.10) fica muito mais fácil e rápido. A forma da expressão (3.14) resulta em uma vantagem adicional, já que a classe a ser designada para o ponto X , é a que tiver o menor valor, abaixo de um limiar de rejeição L escolhido pelo usuário. A expressão reescrita combinando (3.10) e (3.14) fica da forma:

$$d_j^2(\underline{X}) = \underline{X}_1^2 + \underline{X}_2^2 + \dots + \underline{X}_n^2 + \log |C_j| \quad (3.15)$$

Assim, na medida em que as partes da soma forem sendo calculadas e se porventura, exceder o limiar ou o valor já obtido para uma outra classe, o cálculo é abandonado e o algoritmo inicia os cálculos de uma nova classe, já que esta classe não será atribuída.

Pelo motivo acima, houve alterações na composição dos cálculos do MAXVCLA, pois para o modelo contextual, esta parte da rotina não pode ser adaptada, pois as premissas iniciais não podem ser assumidas. Assim, no MXVCON será necessário o cálculo da $f_k(\underline{X}_i)$, devendo ser adaptada uma outra biblioteca em adição ao MAXVLIB2. Esta rotina externa é denominada CONTEXT.

Portanto, o aperfeiçoamento no sistema contextual, visto o acima exposto, é a adoção de logaritmo na expressão

$$P(k \setminus \Delta_i) = \pi(k) f_k(\underline{X}_i) R_k(\Delta_i) \quad (3.16)$$

$$\Rightarrow \ln P(k \setminus \Delta_i) = \ln [\pi(k)] + \ln [f_k(\underline{X}_i)] + \ln [R_k(\Delta_i)] \quad (3.17)$$

Como cada termo é o logaritmo neperiano de um número compreendido no intervalo [0,1], isto significa que estes termos serão negativos. Assim, a expressão (3.17) pode ser reescrita como:

$$\text{coln } [P(k \setminus \Delta_i)] = \text{coln } [\pi(k)] + \text{coln } [f_k(\underline{X}_i)] +$$

$$+ \text{coln} [R_k (\Delta_i)] \quad (3.18)$$

onde coln é o cologaritmo neperiano da expressão envolvida, ou seja,

$$\text{coln} (x) = - \ln (x) \quad (3.19)$$

O que o $\text{coln} (x)$ faz é transformar a maximização em minimização. Se a soma:

$$\text{coln} [P(k \setminus \Delta_i)] = \text{coln} [\pi(k)] + \text{coln} [f_k (X_i)] \quad (3.20)$$

for maior que $\text{coln} (\text{corte}_k)$, antes do cômputo de $R_k(\Delta_i)$, ou ultrapassar uma outra classe antes deste mesmo cômputo, o cálculo para esta classe k é abandonado, buscando-se uma nova classe, se houver. Isto otimiza os cálculos, já que o maior tempo computacional gasto pela rotina será no cálculo de $R_k(\Delta_i)$.

Classificando assim, o ponto na classe k onde

$$\hat{C}_i = \begin{cases} k, & \text{se } \text{coln}[P(k \setminus \Delta_i)] = \min_{m \leq K} \left(\text{coln} P_i (m \setminus \Delta_i) \right) \leq \text{coln}(\text{corte}_k) \\ D, & \text{se } \text{coln}[P(k \setminus \Delta_i)] > \text{coln} (\text{corte}_k), k \in [1, K]. \end{cases} \quad (3.21)$$

o que modifica a forma da equação (2.8).

O arquivo de subrotinas CONTEXT conterà as seguintes subrotinas: CLASSIFICA_IMAGEM_CONT(), CONTEXT() e DENSID().

A subrotina CLASSIFICA_IMAGEM_CONT() substitui a rotina *classifica_imagem()* da biblioteca MAXVLIB2 e recebe de MXVCON os parâmetros de *contexto.con* e *e.con*, lê a imagem através do

"buffer" rotativo e classifica a imagem, produzindo a saída classificada no canal 3 da Unidade Visualizadora de Imagem - UVI. Utiliza como suas subrotinas CONTEXT() e DENSID().

A subrotina CONTEXT() recebe os parâmetros de contexto con e os valores da cruz de pixels enviados por CLASSIFICA_IMAGEM_CONT(). Usa DENSID() para calcular as densidades $f_k(X)$, e no seu corpo calcular as funções $A_k(\Delta_i)$, $B_k(\Delta_i)$ e $C_k(\Delta_i)$ - equação (2.18) - $a(x)$ e $b(x,y)$ - equação (2.19) - para chegar ao fator de correção $R_k(\circ)$ para cada caso.

A subrotina DENSID() faz o cálculo das densidades normais multivariadas, segundo a equação (3.5) e recebe os parâmetros de CLASSIFICA_IMAGEM_CONT() para realizar a tarefa, devolvendo o valor numérico da densidade pedida. São mantidos os cálculos da distância de Mahalanobis com as devidas transformações vistas acima para distância Euclidiana, como uma forma de minimizar o tempo computacional gasto.

Estas transformações em nada afetarão o resultado final [55], já que são apenas um deslocamento da nuvem de pontos na primeira e uma projeção na segunda transformação.

Esta ordenação acima imposta se deve ao fato de que ao rodar o programa em módulos separados, a não atualização de um destes parâmetros pode por a perder todo um trabalho bem feito.

Como sumário, a tabela 3.1 orienta sobre a ordem das rotinas a serem chamadas, a finalidade e se caso fôr, os arquivos de dados atualizados por cada uma delas:

número de ordem	arquivo de programa	Descrição de execução
1	MAXVINI	Inicialização de todo o sistema, originando o arquivo <u>maxver.max</u>
2	MAXVAQUI	Adquire amostras de áreas homogêneas podendo ser combinado com qualquer rotina do MAXVER original
3	MPQR	Adquire amostras em cruz e calcula as probabilidades a priori e os parâmetros p,q,r, originando o arquivo <u>contexto.con</u>
4	FUNCAOE	Obtém os limiares necessários aos cálculos de classificação. Pode ser alterado pelo usuário, mas as componentes do vetor devem estar obrigatoriamente no intervalo (0,1). Origina o arquivo <u>e.con</u>
5	MXVCON	Classifica os pixels por um algoritmo contextual, lendo parâmetros dos arquivos <u>e.con</u> e <u>contexto.con</u> e utilizando como subrotina o programa CONTEXT para o cálculo do fator de correção R.

Tabela 3.1. Sumário das rotinas do Método Contextual implementado.

CAPÍTULO 4

Testes e Avaliação de Desempenho

4.1. Introdução

Neste capítulo, será feito uma análise dos testes do sistema contextual implementado e avaliar-se-á o seu desempenho em comparação com o sistema MAXVER que já é parte integrante do sistema SITIM, do qual o sistema contextual herdou várias de suas características.

O "hardware" utilizado é composto por um micro-computador compatível PC/AT-386-SX - com "clock" de 25MHz, memória principal de 4Mb, com co-processador aritmético 80387 e monitor DATANAV. A esta máquina, são acrescentadas quatro placas dotadas de alimentação e endereçamento independentes para trabalho específico com imagens, compondo o sistema SITIM-150. A implementação do sistema data de 1987.

Ultrapassando os objetivos iniciais desta dissertação, também foi desenvolvida uma versão para o sistema SITIM-340. Nesta versão o "hardware" é composto por um micro compatível PC/AT-286 - com "clock" de 20MHz, memória principal de 1Mb, também com co-processador aritmético 80287 e monitor NEC Multisync 3D - modificado pela inclusão de uma placa apenas, no próprio barramento do computador para o trabalho específico com imagens.

4.2. Testes

Foi obtida junto ao INPE-São José dos Campos uma imagem

LANDSAT-TM da Represa Balbina no estado do Amazonas, em meio a floresta amazônica com localização aproximada de 1.5° de latitude Sul e 61° de longitude Oeste, pré-processada, pronta para a classificação.

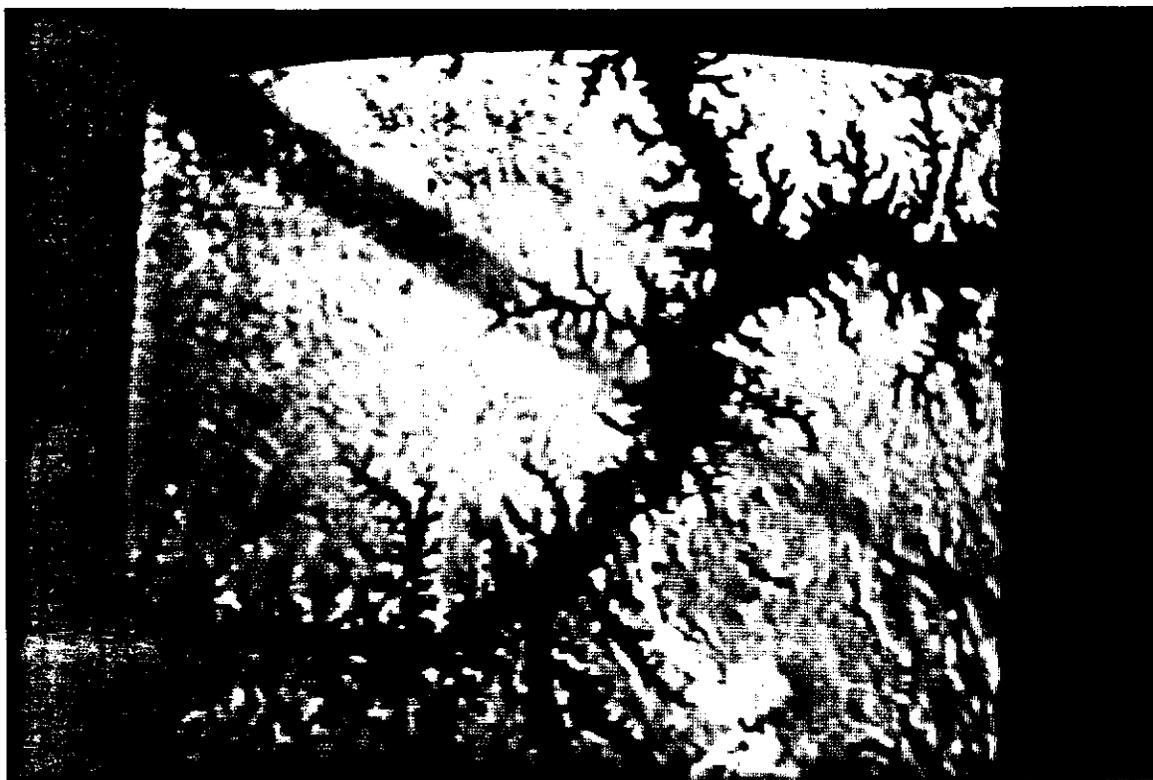


Figura 4.1. Imagem utilizada.

Foram tomadas amostras homogêneas para subsidiar a classificação, através do programa MAXVAQUI que faz parte das rotinas do sistema MAXVER. Pode-se visualizar a coleta destas amostras através da figura 4.2:

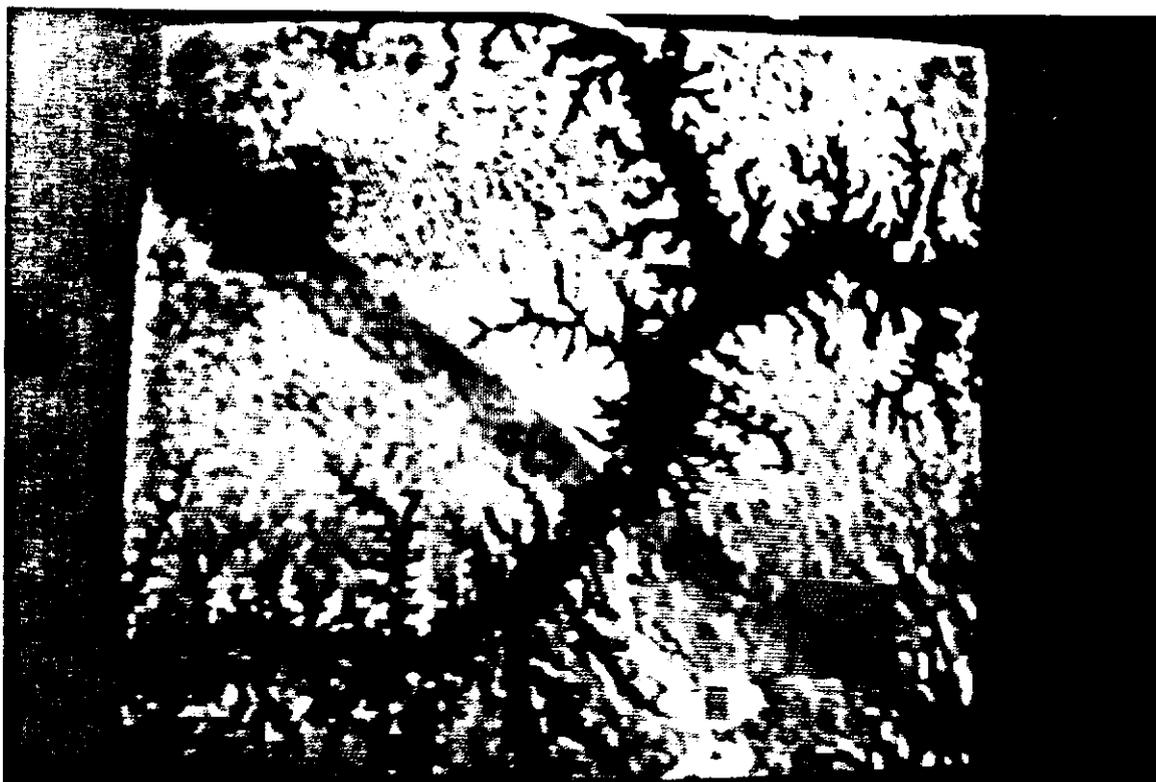


Figura 4.2. Imagem da coleta das amostras homogêneas

Na tabela 4.1, observa-se o número de pontos amostrados para cada classe e o número de amostras tomadas.

Classe	Número de Amostras	Número de Pontos Amostrados
1- Floresta	2	24.576
2- Rio	4	12.520
3- Solo	2	1.362

Tabela 4.1. Número de amostras e pontos amostrados por classe.

A Matriz de Classificação obtida a partir das amostras

homogêneas colhidas através do MAXVAQUI - rotina do MAXVER - é mostrada na tabela 4.2, com limiar L dado por 7.80:

Classificação	Nada	1	2	3
1- Floresta	6,20	93,80	0,00	0,00
2- Rio	7,50	0,10	92,40	0,00
3- Solo	4,80	1,30	0,00	93,80
Desempenho Médio = 93,34				
Abstenção Média = 6,59				
Confusão Média = 0,08				

Tabela 4.2. Matriz de Classificação.

Inicialmente, foi executado o programa em um sistema SITIM-150 do Centro de Pesquisas da Agricultura - CEPAGRI da UNICAMP. O processamento se deu nas coordenadas do cursor ((198,182);(397,381)) contendo uma matriz de 200x200 pontos, totalizando 40.000 pontos. Há de se observar que para o sistema contextual implementado, uma moldura de largura de um pixel da imagem não será classificada em virtude de não se poder utilizar seus vizinhos. Desta forma, a classificação real será de uma matriz de 198x198 pontos, totalizando 39.204 pontos.

O tempo de execução do classificador MXVCON, contado a partir da tomada da área a ser classificada foi de 24:49:6 minutos no sistema SITIM 150 com a configuração descrita acima. Isto resultou em um tempo de processamento por "pixel" de 0,038 segundo o que significa um número de 26,32 "pixels" por segundo.

Na figura 4.3, observa-se a imagem classificada gerada pelo classificador MXVCON.

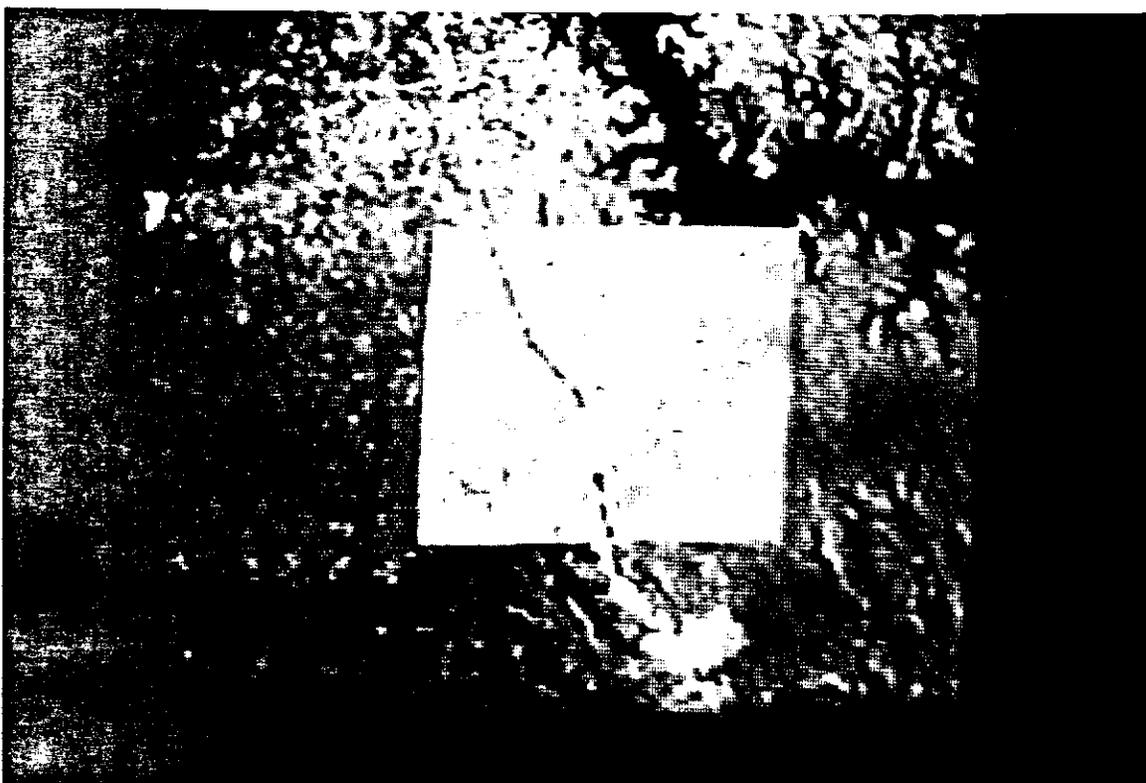


Figura 4.3. Imagem classificada pelo MXVCON no SITIM 150.

Na versão para o sistema SITIM-340, testada em um sistema do Departamento de Processamento de Imagens do INPE, o MXVCON processou a imagem de coordenadas do cursor ((396,268);(755,579)) com 360x312 "pixels". Como visto acima, isto resulta em uma janela de 358x310 "pixels", ou seja, 110980 pontos, levando um tempo de 2:17:58:3 horas. Isto significa um "pixel" classificado a cada 0,075 segundo ou um número de 13,41 "pixels" classificados por segundo.

Na figura 4.4, pode-se observar a imagem classificada pelo

MXVCON no sistema SITIM 340.

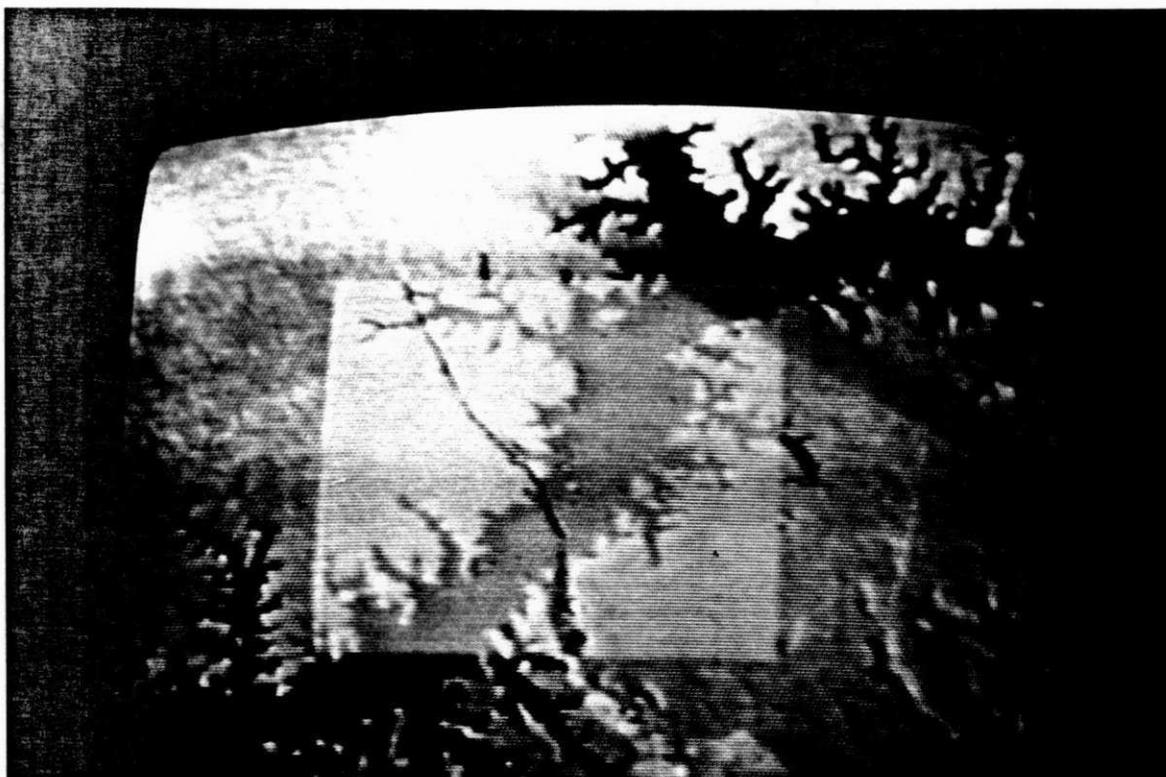


Figura 4.4. Imagem classificada pelo MXVCON no SITIM 340.

Na tabela 4.3, pode-se visualizar um quadro com o resumo das informações sobre o desempenho de cada sistema.

Sistema	Janela	Tempo Total	Tempo por "pixel"
150	39.204	24:49:6 min	0.038 seg
340	110.313	2:17:58:3 hor	0.075 seg

Tabela 4.3. Quadro remissivo de desempenho.

4.3. Avaliação de desempenho

Os sistemas de classificação do SITIM 150 não são passíveis de

serem comparados em um mesmo nível com o método implementado, dadas às premissas diferentes pelas quais operam. Por isto, será feita neste capítulo uma breve avaliação comparativa apenas entre o método de máxima verossimilhança (MAXVER) e o método contextual (MXVCON) implementado. A razão desta única comparação deve-se ao fato de que suas premissas não são consideravelmente diferentes, já que se tratam de dois classificadores com estrutura bayesiana.

Apresentar-se-á a seguir uma breve descrição de cada um dos métodos implantados no SITIM, os quais são listados abaixo:

- 1) Método da Máxima Verossimilhança.
- 2) Método do Paralelepípedo.
- 3) Método do Fatiamento de Histograma.
- 4) Método de Fatiamento de Temperaturas.
- 5) Método da Distância Euclidiana.

O método (1), já foi discutido de forma preliminar no capítulo 1, porém, com ênfase em alguns aspectos específicos à forma de implantação no sistema. Os demais não foram abordados anteriormente e isto será feito a seguir. Deve-se ressaltar que o método (5) encontra-se disponível apenas na versão 340 do sistema SITIM.

4.3.1. O Método da Máxima Verossimilhança (MAXVER)

No SITIM, a abordagem estatística para o Método da Máxima Verossimilhança é ligeiramente diferente da tratada inicialmente no capítulo 1, devido a uma certa simplificação do método, por parte dos que o implementaram. Tradicionalmente, assume-se que o

ponto tem densidade condicional de probabilidade $P[X \sim \setminus W(j)]$, onde X é o vetor de características naquele ponto e $W(j)$ é a j -ésima classe. Assim deve-se maximizar a probabilidade $P[W(j) \sim \setminus X]$, ou seja:

$$P[W(j) \sim \setminus X] = \frac{P[X \sim \setminus W(j)] P[W(j)]}{P[X]} \quad (4.1)$$

Na programação MAXVER do SITIM, houve uma simplificação; o cálculo de $P[W(j) \sim \setminus X]$ pôde ser evitado usando-se uma premissa que todos os $P[X]$ e $P[W(j)]$ são igualmente prováveis. Neste caso, $P[X \sim \setminus W(j)]$ é a chamada Verossimilhança e supõe-se distribuir-se segundo uma normal multivariada, com a média e a matriz de covariâncias calculadas como em (2.23) para a média e semelhante a (2.24) para a matriz de covariâncias, como segue:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} X_j^{(k)} \quad (4.2)$$

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} (X_j^{(k)} - \hat{\mu}_k) (X_j^{(k)} - \hat{\mu}_k)^t \quad (4.3)$$

onde n_k é o número de pontos da amostra.

A estimativa dos parâmetros é feita anteriormente pelo módulo MAXVAQUI, oferecendo-se também a opção de descarte de amostras, análise de cada amostra adquirida, matriz de classificação, visualização das amostras na unidade de visualização (UVI), treinamento automático do classificador (treinamento não-supervisionado, por k -médias), seleção de atributos por Jeffries-Matusita e por fim, apresentação na UVI ou em disco da

imagem classificada, que não necessariamente é a imagem toda.

A classificação é feita com base nos cálculos apresentados no capítulo 3, e toma-se a decisão de classificar o ponto em uma classe quando $d_j^2(X)$ é o menor entre todas as classes e menor que o limiar L fornecido. A matriz de covariâncias no sistema MAXVER não é armazenada, mas sim a matriz de autocorrelação. Quando necessário, é feita a transformação:

$$\Sigma = A - MM^t \quad (4.4)$$

onde : $A = \frac{1}{n_k} \sum_{i=1}^N X_i X_i^t$, (4.5)

é a matriz de autocorrelação.

Todo este sistema de classificação está disponível no SÍTIM nas rotinas MAXVER, composta por MAXVINI - cria um arquivo de trabalho, ou o atualiza no caso de um novo trabalho; MAXVAQUI - aquisição de amostras; MAXVSUB - subtração de amostras, visando melhorar a classificação; MAXVMAT - matriz de classificação; MAXVANS - mostra qual a porcentagem dos pontos de cada amostra classificados em cada uma das classes adquiridas; MAXVAPR - localiza na UVI todas as amostras adquiridas; REAQUIS - readquire novas bandas, caso isto seja necessário; TREINAUT - treinamento automático; MAXVCLA - classificação por máxima verossimilhança; MAXVMOS - mostra o resultado de uma classificação feita em disco; SELATR - seleciona atributos usando a distância J-M; MAXVLST - lista na impressora os parâmetros do arquivo maxver inicializado com MAXVINI.

4.3.2. Método do Paralelepípedo

Podendo ser chamado isoladamente pelo nome SINGLE, este sistema determina os pixels que tenham níveis de cinza compreendidos em um intervalo especificado para cada canal; intervalo este obtido em uma fase de treinamento prévia, podendo o usuário modificá-las a seu contento.

O programa exige como entrada a imagem na UVI e tem como saídas a imagem classificada no canal gráfico e um relatório de resultados no terminal.

4.3.3. Método do Fatiamento de Histograma

Este método simples e intuitivo, consiste em "fatiar" o histograma em intervalos e atribuir a estas fatias cores na UVI. No máximo podem ser estipuladas 14 fatias e este fatiamento pode ser feito de três maneiras:

- a) Equidistribuição : as fatias são escolhidas de modo que cada uma delas contenha o mesmo número de pixels.
- b) Normal : O histograma é dividido em intervalos iguais.
- c) Manual : O fatiamento é feito pelo usuário.

O programa tem como entrada a imagem na UVI e como saídas a imagem classificada no canal gráfico, um arquivo de tabela no "ramdisk" e um relatório de resultados no terminal. Pode ser chamado isoladamente por FATIA.

4.3.4. Método de Fatiamento de Temperaturas

Este programam visa a classificação pelo método do fatiamento de Histograma, no modo *normal*, utilizando temperaturas. Pode ser

criadas até 7 fatias e consiste em dividir o intervalo entre dois valores de níveis de cinza em fatias iguais a serem associadas a cores no canal gráfico da UVI.

O programa, que pode ser chamado por FATTEMP, tem como entrada a imagem na UVI e como saída a imagem classificada no canal gráfico. Entretanto, só utiliza imagens geradas pelos canais infra-termais dos satélites.

As aplicações principais para este programa estão na área de meteorologia, onde é possível distinguir, por exemplo, massas de ar frias ou quentes sobre o continente ou oceano.

4.3.5. Método da Distância Euclidiana

Utiliza o critério do centro da classe mais próxima medido pela distância euclidiana para classificar um pixel multiespectral em uma classe dada.

$$\underline{X} \in C_i \text{ se } \|\underline{X} - \underline{M}_i\| < \|\underline{X} - \underline{M}_j\|, \forall j \neq i; \quad (4.6)$$

onde: C_i é a classe i

\underline{X} é o vetor de atributos

~

\underline{M}_i é o vetor de médias da classe.

Este sistema é compatível com o MAXVER e exige entradas dos arquivos maxver e de contexto maxver, tendo como saídas os arquivos maxver atualizado e imagem classificada. É chamado diretamente pelo nome CLASEUC e permite a iteração com todos os arquivos MAXVER citados no item anterior

4.4. Teste comparativo com o MAXVER

Os resultados que foram obtidos pela utilização do classificador foram os seguintes: Número de amostras e pontos amostrados foram os mesmos descritos em 4.2, haja visto que foram tomadas pelo MAXVAQUI. Desta forma, a matriz de classificação também será a mesma descrita 4.2. Para se ter uma base melhor para tecer comentários e avaliações, a janela de classificação também foi a mesma com 200x200 "pixels", gerando a imagem da figura 4.5.

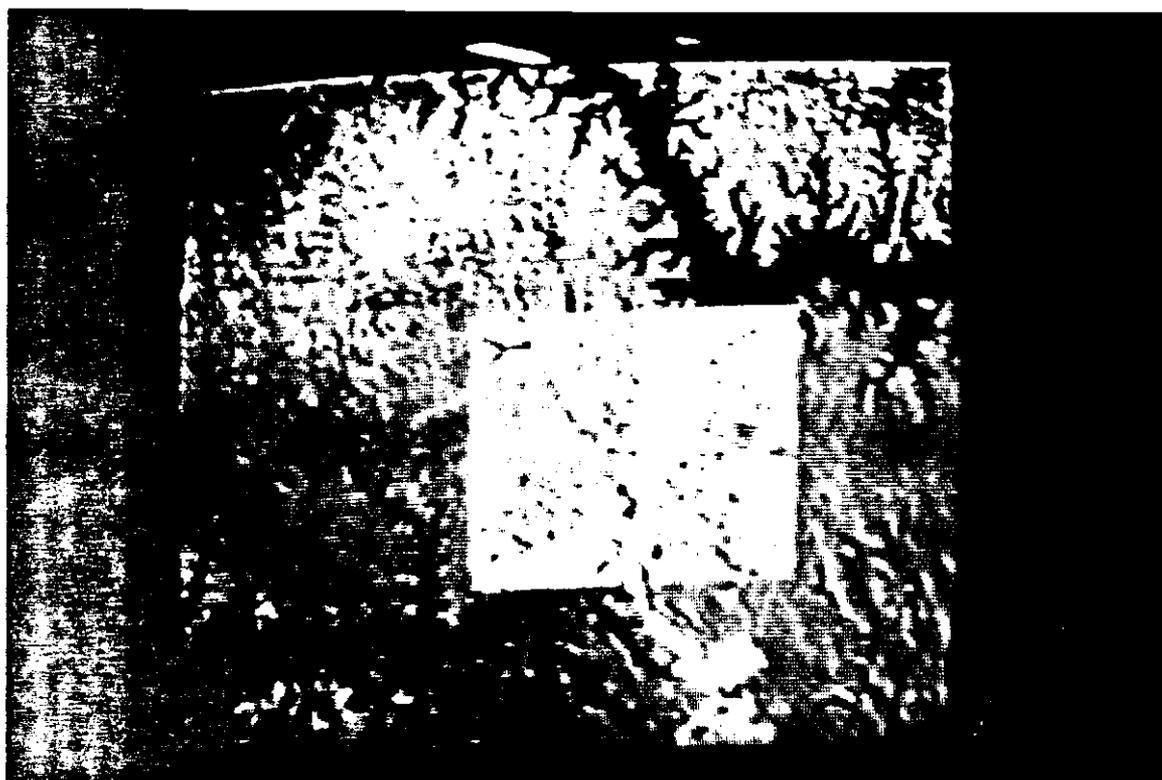


Figura 4.5. Imagem classificada pelo MAXVER no SITIM 150.

Para esta janela de classificação, o MAXVER no sistema SITIM-150 com base AT-386 e monitor DATANAV, levou 22:10 segundos, gerando a seguinte tabela de classificação, com limiar de 7.80:

Classe	Área
Floresta	63.30 km ²
Rio	30.80 km ²
Solo	0.50 km ²
Área não classificada	13.40 km ²

Tabela 4.4 - Área classificada pelo MAXVER - sistema SITIM 150.

Faz-se necessário notar que assim como o sistema contextual MXVCON, o limiar de rejeição L do MAXVER influi no resultado final. Este limiar L = 7,80, é "default" do sistema.

A seguir, a tabela 4.5 gerada pelo método contextual:

Classe	Área
Floresta	67.5837km ²
Rio	35.7507km ²
Solo	1.2231km ²
Área não classificada	1.2933km ²

Tabela 4.5 - Área classificada pelo MXVCON - sistema SITIM 150.

Pode-se notar pelas tabelas (4.4) e (4.5) e também pela imagem gerada por ambos os classificadores que há uma indiscutível superioridade do método contextual sobre o método de máxima verossimilhança. Numericamente, pode-se notar a área não classificada por cada método, a proporção não classificada chega a ser 10,36 vezes menor no método contextual. Observando as imagens geradas, principalmente na área de solo, nota-se o rendimento bastante superior do método contextual. Nota-se que na classe rio, há um trecho de rio que não foi reconhecido pelo MAXVER.

No sistema SITIM 340 com base AT-285 e monitor de vídeo NEC-3D, classificando a mesma janela de 359x311 "pixels", o tempo

gasto pelo MAXVER foi de 2:39:3 minutos. A tabela 4.6 foi gerada por esta classificação:

Classe	Área
Floresta	56.4678km ²
Rio	31.5756km ²
Solo	0.5292km ²
Área não classificada	12.5154km ²

Tabela 4.6 - Área classificada pelo MAXVER - sistema SITIM 340.

Esta classificação gerada pelo MAXVER, pode ser visualizada na figura 4.6.



Figura 4.6. Imagem classificada pelo MAXVER no SITIM 340.

Com o método contextual, obteve-se a seguinte tabela:

Classe	Área
Floresta	60.8616km ²
Rio	36.2232km ²
Solo	1.3779km ²
Área não classificada	1.3779km ²

Tabela 4.7 - Área classificada pelo MXVCON - sistema SITIM 340.

Uma rápida comparação entre os desempenhos dos métodos em cada sistema SITIM, pode ser vista na tabela 4.8.

Sistema	Tempo de Processamento	
	MAXVER	MXVCON
SITIM 150	22:10 seg.	24:49:6 min.
SITIM 340	2:39:3 min.	2:17:58:3 hor.

Tabela 4.8 - Tempos de processamentos por sistema, dos programas MAXVER e MXVCON

Nota-se que pelo método contextual, o tempo gasto foi de 2:17:58:3 horas. Este tempo muito superior, no entanto, tem uma explicação; observa-se que a área classificada é aproximadamente a mesma nos dois casos e há uma diferença significativa do número de pontos calculados. A diferença é a resolução do monitor. O monitor NEC-3D tem uma resolução muito superior a do DATANAV. Colocando mais pontos por polegada na tela, e visto que a classificação tanto do MAXVER quanto do CONTEXTUAL leem a imagem da tela, explica-se por que o tempo de classificação foi tão diferente.

No sistema SITIM 340, também foi possível fazer a diferença entre as imagens geradas pela classificação do MXVCON e do MAXVER,

mostrada pela figura 4.7.



Figura 4.7. Diferença entre as imagens geradas pelo MXVCON (em cima à esquerda) e MAXVER (em cima à direita), resultando a imagem abaixo à esquerda.

Ficou claro que o MAXVER não conseguiu reconhecer as bordas entre as classes *rio* e *floresta* e deixou muitos pontos a serem classificados na área de *floresta*. A respeito da classe *solo*, o desempenho do classificador contextual MXVCON foi indiscutivelmente superior. Notou-se também que o MAXVER deixou de classificar uma área na borda das classes *rio* e *solo*. Nesta mesma área, o MXVCON também não se saiu bem quanto ao reconhecimento, mas ainda assim o seu rendimento foi superior. Pôde ser notado

também, alguns pontos na classe rio, que não foram classificados, formando uma espécie de "linha" ao longo do seu leito.

Pôde-se observar também que comparando as tabelas 4.4 e 4.5 com as tabelas 4.6 e 4.7, houve uma alteração das áreas classificadas a nível geral. Isto se deve ao fato da resolução maior do monitor de vídeo NEC-3D, o que fez com que a escala fosse alterada, permitindo maior precisão na estimativa de área. Possivelmente, também devido às diferenças entre os monitores, a nível visual, pôde-se observar uma melhora na precisão dos dois algoritmos no sistema SITIM 340, o que pode ser notado verificando os pares de figuras: 4.3 e 4.4; 4.5 e 4.6.

Pelos fatos vistos acima, pode-se concluir que existe validade na aplicação do método contextual, principalmente quando se procura uma maior precisão nos resultados obtidos, como por exemplo, pesquisas envolvendo áreas, como área plantada ou desmatada entre outras. Esta validade é ainda maior em termos de tempo de processamento se o tipo de monitor utilizado não for de resolução muito alta. É notório que a área a ser classificada também influi no tempo, mas quando se procura melhor qualidade na classificação, sem grandes restrições quanto ao tempo de processamento da imagem, sem dúvida o método contextual é realmente muito superior.

CAPÍTULO 5

Conclusões Finais e Sugestões

5.1. Introdução

Finalizando esta dissertação, são expostas as conclusões finais deste trabalho a nível geral e sobre a performance do sistema final.

Devido ao caráter desta dissertação ser a implantação de um método contextual compatível com o sistema SITIM, pode-se agora, ao término da jornada visualizar caminhos que talvez pudessem ser trilhados por trabalhos futuros. Este último capítulo visa deixar as conclusões deste e sugestões para futuros trabalhos na área, evitando assim diversos contratempos defrontados na confecção deste.

5.2. Conclusões

Apesar da lentidão inerente ao método contextual implantado, sua utilização é desejável sempre que se necessita de uma certeza maior de classificação. Por um outro fator, não inerente ao método, mas acoplado a ele, o usuário tem a facilidade neste sistema de definir os limiares conforme sua vontade, o que significa poder ponderar as diversas classes exercendo uma maior supervisão sobre o trabalho do algoritmo para uma ou mais classes específicas de interesse. Pela facilidade gerada por já se ter excedido o objetivo inicial de um sistema compatível com o SITIM 150 e já ser disponível a versão para o SITIM 340, a morosidade

pode até ser desprezada em uma classificação de apenas parte de uma imagem, dependendo, é claro, das dimensões da mesma e da máquina utilizada.

Pelas simulações em diversos tipos de máquina, pode-se concluir que as otimizações na programação do algoritmo surtiram efeito e a questão da morosidade nos cálculos da função de correção contextual $R_k(\Delta_i)$ é somente uma questão de avanço de "hardware" para suas proporções serem reduzidas.

5.3. Sugestões para trabalhos futuros

Com a finalização deste trabalho, imediatamente surgem idéias de como prolongá-lo, extende-lo ou aperfeiçoá-lo. Por isto, serão deixadas aqui sugestões para futuros trabalhos.

As rotinas resultaram de um trabalho de exaustiva busca na compreensão de um sistema, o SITIM-150, o qual já pode ser considerado em muitos aspectos ultrapassado. Há uma versão mais recente, o SITIM-340, feito também para a família PC que também já possui a sua versão das rotinas. No entanto, atualmente, encontra-se em fase de desenvolvimento no INPE em São José dos Campos - S.P., uma versão totalmente remodelada, programada orientada a objetos, para estações de trabalho com arquitetura RISC e sistema operacional tipo UNIX. Esta versão, seguramente mais poderosa, poderia ser alvo da adaptação deste sistema, talvez com o mesmo grau de mistura com o MAXVER.

Outra sugestão, a nível de processamento de imagens, seria a comparação mais elaborada entre os sistemas de classificação para várias imagens diferentes, ou até mesmo protótipos de imagens com

determinadas variáveis propositalmente distorcidas, como alta variância dentro das classes, bordas marcantes, etc.

Pode-se dar como sugestão também, uma tentativa de aperfeiçoamento deste sistema, otimizando-se suas rotinas, modificando-as até mesmo reescrevendo-as de forma a melhorar sua performance. Por exemplo poderia ser introduzido o uso de vetores não-classificados na estimativa dos parâmetros p, q, r , como descrito em Hjort [17].

Uma outra sugestão seria a avaliação de várias imagens, visando o entendimento da relação custo-benefício deste sistema buscando compreender até que ponto vale o uso de um algoritmo contextual em detrimento de um algoritmo clássico e vice-versa, ou seja, avaliar a melhoria com o classificador contextual sobre o classificador clássico.

Pode-se sugerir ainda um estudo de forma a viabilizar uma estrutura "fuzzy" contextual do tipo p, q, r tendo como base o trabalho de Bezdeck [21], ou ainda aperfeiçoar a estrutura de correlação das funções g e h , partindo dos trabalhos de Klein & Press [31] e [33].

Bibliografia

- [1] Cooper, D.B. and Freeman, J.H.; "On the Asymptotic Improvement in the Outcome of Supervised Learning Provided by Additional Nonsupervised Learning". IEEE Transactions on Computers, vol.19, n.11, november, 1970.
- [2] Sammon Jr, J.W.; "Interactive Pattern Analysis and Classification". IEEE Transactions on Computers, vol.19, no.7, 1970.
- [3] Welch, J.R. and Salter, K.G.; "A context algorithm for pattern recognition and image interpretation ". IEEE Transactions on Syst., Man and Cybernetics, vol.SMC-1, n.1, January, 1971, pp 24-30.
- [4] Eigen, D.J., Fromm, F.R. and Northouse, R.A.; "Cluster Analysis Based on Dimensional Information with Applications to Feature Selection and Classification". IEEE Transactions on Systems, Man and Cybernetics, vol.4, no.3, 1974.
- [5] Dunn, J.C.; "A Graph Theoretic Analysis of Pattern Classification via Tamura's Fuzzy Relations". IEEE Trans. Syst. Man. Cybern., vol SMC-4, 1974, pg 310-313

- [6] Landgrebe, D.A.; "The development of a spectral-spatial classifier for Earth observational data" : Pattern Recognition, vol.12, 1980, pp 165-175.
- [7] Swain, P. H.; Siegel, H. J. and Smith, B. W.; "Contextual Classification of Multispectral Remote Sensing Data using a Multiprocessor System." : IEEE Transactions on Geoscience and Remote Sensing, vol.GE-18, n.2, April, 1980, pp 197-203.
- [8] Swain, P. H.; Vanderman, S. B. and Tilton, J. C.; "Contextual Classification of Multispectral Image Data." : Pattern Recognition, vol.13, n.6, 1981, pp 429-441.
- [9] Tilton, J.C. and Swain, P.H.; Incorporating Spatial Context into Spatial Classification of Multidimensional Image Data. Technical Report of LARS - Purdue University, LARS - 072981, August, 1981, 92 pp.
- [10] Owen, A. and Switzer, P.; A neighbourhood based classifier for LANDSAT Data : Technical Report, Dep. of Statistics, Stanford University, 1982.
- [11] Tilton, J.C., Vanderman, S.B. and Swain, P.H.; "Estimation of context for statistical classification of multispectral image data." : IEEE Transactions on Geoscience and Remote Sensing, vol.GE-20, n.4, October, 1982, pp 445-452.

- [12] Whorton, S.W.; "A contextual classification method for recognizing land use patterns in high resolution remotely sensed data." : Pattern Recognition, vol.15, n.4, 1982, pp 317-324.
- [13] Owen, A.; " A neighbourhood-based classifier for LANDSAT Data.": The Canadian Journal of Statistics, vol.12, n.3, 1984, pp 191-200.
- [14] Mardia, K.S.; "Spatial Discrimination and Classifications Maps". Communications in Statistics - Theory and Methods, vol.13, no.18, 1984.
- [15] Saebo, H.V.; Braten, K.; Hjort, N.L.; Llewellyn, B. and Mohn, E.; Contextual Classification of Remotely Sensed Data : Statistical Methods and Development of a System - Report n.768 - Norwegian Computing Center - Oslo - 1985.
- [16] Hjort, N.L. and Mohn, E.; "A comparison of some contextual methods in remote sensing classification." : in Saebo, H.V. et al.; Contextual Classification of Remotely Sensed Data: Statistical Methods and Development of a System; Technical Report of Norwegian Computing Center, n.768, April, 1985, Oslo, Enclousure 2
- [17] Hjort, N.L.; "Neighbourhood Based Classification of Remotely Sensed Data based on Geometric Probability Models." : in

Saebo, H.V. et al; Contextual Classification of Remotely Sensed Data: Statistical Methods and Development of a System; Technical Report of Norwegian Computing Center, n.768, April, 1985, Oslo, Enclosure 4.

[18] Hjort, N.L.; Mohn, E. and Storvik, G.; "Contextual Classification of Remotely Sensed Data, Based on an Autocorrelation Model." : in Saebo H.V. et al; Contextual Classification of Remotely Sensed Data: Statistical Methods and Development of a System; Technical Report of Norwegian Computing Center, n.768, April, 1985, Oslo, Enclosure 5.

[19] Hjort, N.L. and Mohn, E.; "Notes on adjustments to contextual classification rules when some neighbours are missing." : in Saebo H.V. et al; Contextual Classification of Remotely Sensed Data: Statistical Methods and Development of a System; Technical Report of Norwegian Computing Center, n.768, April, 1985, Oslo, Enclosure 6.

[20] Hjort, N. L.; "Estimating parameters in neighbourhood based classifiers for remotely sensed data, using unclassified vectors." : in Saebo H.V. et al; Contextual Classification of Remotely Sensed Data: Statistical Methods and Development of a System; Technical Report of Norwegian Computing Center, n.768, April, 1985, Oslo, Enclosure 7.

[21] Bezdek, J.C.; Chuah, S.K. and Leep, D.; "Generalized

- k-Nearest Neighbor Rules. Fuzzy Sets and Syst, v.18, n.2, 1986, pg 237-256.
- [22] Therrien, C.W., Quatieri, T.F. and Dudgeon, D.E.; "Statistical Model-Based Algorithms for Image Analysis". Proceedings of the IEEE, vol.74, no.4, 1986.
- [23] Besag, J.; "On the Statistical Analysis of Dirty Pictures". JRSS, vol.48, no.5, 1986.
- [24] Govindan, V.K. and Shivaprasad, A.P.; "A Pattern Adaptive Thinning Algorithm. Pattern Recognition, vol.20, no.6, 1987.
- [25] Skidmore, A.K. and Turner, B.J.; "Forest Mapping Accuracies are Improved a Supervised Nonparametric Classifier with SPOT Data". Photogrammetric Engineering and Remote Sensing, vol.54, no.10, 1988.
- [26] Kent, J.T. and Mardia, K.V.; "Spatial Classification Using Fuzzy Membership Models". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.10, n.5, September, 1988, pp 659-671.
- [27] Gidas, B.; "A Renormalization Group Approach to Image Processing Problems". IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.11, no.2, 1989.

- [28] Sanjie, J. and Nagle, D.T.; "Pattern Recognition in the Ultrasonic Imaging of Reverberant Multilayered Structures". IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control, vol.36, no.1, 1989.
- [29] Guebert, M.D. and Gardner, T.W.; "Unsupervised SPOT Classification and Infiltration Rates on Surface Mined Watersheds, Central Pennsylvania". Photogrammetric Engineering and Remote Sensing, vol.55, no.10, 1989.
- [30] Lowell, K.E. "A Probabilist Modification of the Decision Rule in the Skidmore/Turner Supervised Nonparametric Classifier. Photogrammetric Engineering and Remote Sensing, vol.55, no.6, 1989.
- [31] Klein, R. and Press, S.J.; "Contextual Bayesian Classification of Remotely Sensed Data." : Commun. Statist. - Theory and Methods, vol.18, n.9, 1989, pp 3177-3202.
- [32] Cola, L.D.; "Fractal Analysis of a Classified Landsat Scene". Photogrammetric Engineering and Remote Sensing, vol.55, no.6, 1989, pp 601-610.
- [33] Klein, R. and Press, S.J.; "Contextual Bayesian Classification of Remotely Sensed Data when Training Data is a part of the Scene." . Revista Brasileira de Probabilidade e Estatística, vol.4, 1990, pp 43-67.

- [34] Jones, L. K. and Byrne, C.L.; "General Entropy Criteria for Inverse Problems, with Applications to Data Compression, Pattern Classification and Cluster Analysis". IEEE Transactions on Information Theory, vol.36, no.1, 1990.
- [35] Swain, P.H. and Davis, S.M.; Remote Sensing - The Quantitative Approach - McGraw Hill International Book Company - USA - 1978.
- [36] Garcia, G.J.; Sensoriamento Remoto: Princípios e Interpretação de Imagens - Editora Nobel - Brasil - 1982.
- [37] Barros, M.A.; Filtros Espaciais: Implementação, estudo comparativo e aplicação em Sensoriamento Remoto. Dissertação de Mestrado - Departamento de Engenharia Elétrica - UFPb - Campina Grande, 1990, 156 pág.
- [38] Barros, L.A.; Um circuito integrado para detecção de bordas com o Operador de Roberts. Dissertação de Mestrado - Departamento de Engenharia Elétrica - UFPb - Campina Grande, 1990, 156 pág.
- [39] Ciampi, M.G.; "Nociones Generales de Processamiento Digital de Imágenes". Revista Selper, março, 1986.
- [40] Johnson, R.A. and Wichern, D.W.; Applied Multivariate Statistical Analysis. Prentice-Hall Inc. - USA - 1982.

- [41] Mascarenhas, N.D.A. e Velasco, F.D.R.; Processamento Digital de Imagens - IV Escola Brasileiro-Argentina de Informática - Termas de Rio Hondo, Argentina - janeiro de 1989.
- [42] Fisher, R.A.; "The use of multiple measurements in taxonomic problems". Ann. Eugen. v.7, 1936, pg 178-188.
- [43] Kent, J.T. and Mardia, K.V.; "Spatial Classification Using Fuzzy Membership Models". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.10, n.5, September, 1988, pp 659-671.
- [44] Baridó, M.P.F.; Uma comparação entre diferentes métodos multivariados de classificação. Dissertação de Mestrado - Instituto de Matemática - UFRJ - Rio de Janeiro, 1989, 189 pág.
- [45] James, M.; Classification Algorithms - John Wiley & Sons Inc. - USA - 1985.
- [46] Fu, K.S. and Yu, T.S.; Statistical Pattern Classification using Contextual Information - Research Studies Press, 1980.
- [47] Duarte, V.; "Metodologia de Interpretação Visual de Dados Orbitais". Publicação interna INPE (no prelo).
- [48] Tucker, C.J.; "A comparison of satellite sensor bands for vegetation monitoring". Photogrammetric Engineering and

- Remote Sensing, vol.44, n.11, november, 1978.978.
- [49] Rudorff, B.F.T.; Palme, U.W.; Duarte, V.; Krug, T.; Tardin, A.T.; Batista, G.T.; Medeiros, J.S. e Godoy Jr., M.; Sensoriamento Remoto na Detecção de áreas plantadas e perdas na produção agrícola no município de Irecê - BA. Publicação interna - INPE - abril, 1986
- [50] Ripley, B.D.; "Statistics, images and pattern recognition." : The Canadian Journal of Statistics, vol.14, n.2, 1986, pp 83-111.
- [51] Switzer, P.; "A random set process in the plane with a Markov property." : Ann. Math. Statist., vol.36, 1965, pp 1859-1863.
- [52] Pereira, C.A.B. e Viana, M.A.G.; Elementos de Inferência Bayesiana. V Simpósio Brasileiro de Probabilidade e Estatística, São Paulo, 1982.
- [53] Maritz, J.S. and Lwin, T.; Empirical Bayes Methods. Chapman and Hall Ltd, Great Britain, 1989.
- [54] SITIM - Manual do Usuário
- [55] Velasco, F.D.R.; Prado, L.O.C. e Souza, R.C.M.; Sistema MAXVER: Manual do usuário. Publicação interna INPE - 1315 - NTI/110, julho 1978.

```

/*****
Nome      : mpqr - adaptado de mixaqu      VI-00
Autor     : Ronei Marcos de Moraes      ago-91
Resumo    : adquire parametros contextuais
           da cruz de pixels.
Entradas  : estrutura maxver
           nome do arquivo maxver
Saídas    : arquivo mpqr atualizado
Subrotinas : mixger,dialogo,gerente,
           adquire_amostras,cont

VI-00      : Ronei Marcos de Moraes - UFPA

```

```

/*****/

```

```

#include <string.h>
#include <stdio.h>
#include "maxver.h"
#include "atr.h"
#include "dia.h"
#include "info.h"

int      contadorf,
         num_clas_dist,
         contador_classe[21],
         mx,m1,mt;

char     nomec[12];          /* nome da componente */

main ()
{
    struct atributos      atr[NBAN_DISCO],          /* atributos do disco */
                          atr_tela[NBAN_TELA];     /* atributos da tela */
    struct arquivo_maxver max;                     /* arquivo maxver */
    struct contexto_maxver cont;                  /* arquivo de contexto */
    struct informacoes   info;                   /* arquivo de informacoes */

    int      i, j, k,
             janela[4],
             canal,
             var,
             nao_achou,
             tela,
             canr,cang,canb,
             nbandas,
             nb,
             m;

    char     nome_arquivo_max[8],
             modo[2],
             help[10];

    float    amos_ant,
             w,emx,eml,emt,
             delta,
             pi[21],
             p,q,r,
             p0,q0,r0;

    FILE     *mpqr;

    uvi_inicio();

    dia_titulo(" MPQR - Aquisicao de Amostras ");
    mx=0;m1=0;mt=0;

    /*

```

Leitura dos arquivos : contexto, maxver , atributos e informacoes

```
*/
perg1:
/* leitura do arquivo de contexto */
if(ler_arq_cont(&cont) == ERRO_CTE)
{
    dia_erro(" ERRO NA LEITURA DO CONTEXTO");
    goto fim_programa;
}

/* leitura do arquivo maxver */
if (ler_arq_max(cont.nome_arquivo_max,&max) == ERRO_MNE)
{
    dia_erro("ERRO NA LEITURA DO ARG. MAXVER");
    goto fim_programa;
}

dia_poe_string(4,18,"Nome da imagem : ",max.nome_imagem);

/*
Le atributos da imagem no disco
*/
switch(ler_atributos('d',max.nome_imagem,atr,&nbandas))
{
    case ERRO_INA : dia_erro("Imagem nao existe no disco");
                  goto fim_programa;
    default      : break;
}

/*
Modo de aquisicao das amostras
*/
perg2:
dia_limpa(5,18);
switch(dia_peg_string(5,18," T - Terminal, C - Cursor, ", "Modo: ",modo))
{
    case FIM      : goto fim_programa;
    case VOLTA   : goto perg1;
    case VAZIO    : dia_erro("Modo deve ser fornecido");
                  goto perg2;
    case NORMAL  : if (strcmpi(modo,"T")==0)
                  {
                      dia_poe_texto(10,18,"Modo Terminal nao adquire amostras",32,45,1,0);
                      tela=0;
                      goto fim_programa;
                  }
                  else
                  {
                      if (strcmpi(modo,"C")==0)
                      {
                          /* le atributos da imagem na tela */
                          switch(ler_atributos('t',"tela",atr_tela,&nb))
                          {
                              case ERRO_INA : dia_erro("IMAGEM NAO EXISTE NA TELA");
                                              goto fim_programa;
                              default      : break;
                          }

                          le_info(&info);

                          janela[0] = info.voo_x + 250;
                          janela[2] = info.voo_y + 250;
                          janela[1] = janela[3] = 9;
                          uvi_le_mux(&canr,&canb,&canb);
                          canal=canr;
                          tela=1;
                      }
                  }
}
else
{
```

```

        dia_erro("Opcao Invalida");
        goto perg2;
    }
}

/*
Monta tela para aquisicao e mostra parametros do arquivo contextual
Adquire amostras
*/
adquire:
    for (;;)
    {
        dia_limpa(8,18);
        switch(dia_pegar_string(22,18,"12 chars,,CR satisfeito","Pega amostras ? : ",nonec,12))
        {
            case FIM : goto fim_programa;
            case VAZIO : goto fora;
            case NORMAL : nao_achou = 1 ;

                if (tela)
                {
                    adquire_amostra(&max,atr,atr_tela,nbandas,janela,canal,var);
                    break;
                }

                else goto fora;
        }
    }
fora:

/* calculos de pi[k] e de p, q, r iterativos */

m = mx + ml + mt;
w = 0;
for (k=1; k <= max.num_classes; k++)
{
    pi[k] = (1.0/(5.0*m))*contador_classe[k];
    w = w + pi[k]*pi[k];
}

for (k=1; k <= max.num_classes; k++)
if (pi[k] == 0.0)
{
    memcpy(nomec,max.nome_classe[k],sizeof(max.nome_classe[k]));
    dia_poe_string(8,18,"Falta amostra para classe ",max.nome_classe[k]);
    dia_erro ("Vai ocorrer ERRO !!!");
    goto adquire;
}

emx = 1.0*mx/m; eml = 1.0*ml/m; emt = 1.0*mt/m;
p = (emx - w) / (1.0-w);
q = eml / (1.0-w);
r = emt / (1.0-w);
delta = 1.0;
while (delta < 0.00001)
{
    p0 = p; q0 = q; r0 = r;
    emx = p + (1-p) * w;
    eml = q * (1-w);
    emt = r * (1-w);
    p = (emx - w) / (1-w);
    q = eml / (1-w);
    r = emt / (1-w);
}

```

```

    delta = (abs (p-p0) + abs (q-q0) + abs (r-r0))/3.0;
}

/* passagem dos parametros numero de classes, pi[k] e */
/* dos parametros p,q,r para o arquivo MPQR.CON      */

mpqr = fopen ("CONTEXTO.CON","wt");
for (k = 1; k <= max.num_classes; k++)
    fprintf (mpqr, "%10.8f\n",pi[k]);
fprintf (mpqr, "%10.8f\n", p);
fprintf (mpqr, "%10.8f\n", q);
fprintf (mpqr, "%10.8f\n", r);
fclose (mpqr);

fim_programa:
    dia_cursor (23,1);
}

/*****
    acquire_amostra

    Autor      : Moacir Godoy Junior          mar-98
                Adriana Abrahao

                adaptacao (ago-91) :
                Ronei Marcos de Moraes (UFFb)

    Resumo     : acquire valores de refletancia do disco

    Entradas   : estrutura maxver

    Sairas     : arquivo mpqr (contexto) atualizada
*****/
acquire_amostra(max, atr, atr_tela,nbandas,janela,canal,var)

struct arquivo_maxver #max;
struct atributos      atr[];
struct atributos      atr_tela[];

int                   nbandas,
                      canal,
                      var,
                      janela[];

{

    int                i,j,k,status,          /* variaveis auxiliares      */
                      linha, coluna,        /* variaveis auxiliares      */
                      ib[TOT_BAND],         /* indice da banda adquirida */
                      coord_amostra[4];     /* coordenadas da amostra no disco */

    unsigned char      buffer[6];           /* bufer de imagem           */

    int                janelad[4],
                      nao_achou,
                      c[5];

    char               string;             /* escolha do usuario        */

/*
    Acquire posicao da amostra
*/
proti:
    dia_limpa(14,1);
    dia_poe_texto(23,31,"POSICIONE O CURSOR",33,40,5,1);
    status = uvi_cruz(canal,janela);

```

```

dia_limpa (23,1);
switch (status)
{
    case FIM : return(FIM);
    case VOLTA : return(VOLTA);
}

```

reentra_c:

```

dia_limpa(15,18);
switch(dia_peg_string(15,18,"12 chars,,CR satisfeito","Classe do pixel central : ",nomec,12))
{
    case FIM : goto fim;
    case VAZIO : goto fim;
    case NORMAL : nao_achou = 1 ;

        for (i=1; i<= max->num_classes; i++)
            if (strcmpi(nomec,max->nome_classe[i]) ==0)
            {
                nao_achou = 0 ;
                c[0] = i ;
            }
            if (nao_achou)
            {
                dia_erro("Classe nao existe");
                goto reentra_c;
            }
            break;
}

```

reentra_n:

```

dia_limpa(16,18);
switch(dia_peg_string(16,18,"12 chars,,CR satisfeito","Classe do pixel norte : ",nomec,12))
{
    case FIM : goto fim;
    case VAZIO : goto fim;
    case NORMAL : nao_achou = 1 ;

        for (i=1;i<= max->num_classes;i++)
            if (strcmpi(nomec,max->nome_classe[i]) ==0)
            {
                nao_achou = 0 ;
                c[1] = i ;
            }
            if (nao_achou)
            {
                dia_erro("Classe nao existe");
                goto reentra_n;
            }
            break;
}

```

reentra_s:

```

dia_limpa(17,18);
switch(dia_peg_string(17,18,"12 chars,,CR satisfeito","Classe do pixel sul : ",nomec,12))
{
    case FIM : goto fim;
    case VAZIO : goto fim;
    case NORMAL : nao_achou = 1 ;

        for (i=1;i<= max->num_classes;i++)
            if (strcmpi(nomec,max->nome_classe[i]) ==0)
            {
                nao_achou = 0 ;
                c[2] = i ;
            }
            if (nao_achou)
            {

```

```

        goto reentra_s;
    }
    break;
}

reentra_o:
dia_limpa(18,18);
switch(dia_peg_string(18,18,"12 chars,,CR satisfeito","Classe do pixel oeste : ",nomec,12))
{
    case FIM : goto fim;
    case VAZIO : goto fim;
    case NORMAL : nao_achou = 1 ;

        for (i=1;i<= max->num_classes;i++)
            if (strcmpi(nomec,max->nome_classe[i]) ==0)
            {
                nao_achou = 0 ;
                c[3] = i ;
            }
        if (nao_achou)
        {
            dia_erro("Classe nao existe");
            goto reentra_o;
        }
    break;
}

```

```

reentra_l:
dia_limpa(19,18);
switch(dia_peg_string(19,18,"12 chars,,CR satisfeito","Classe do pixel leste : ",nomec,12))
{
    case FIM : goto fim;
    case VAZIO : goto fim;
    case NORMAL : nao_achou = 1 ;

        for (i=1;i<= max->num_classes;i++)
            if (strcmpi(nomec,max->nome_classe[i]) ==0)
            {
                nao_achou = 0 ;
                c[4] = i ;
            }
        if (nao_achou)
        {
            dia_erro("Classe nao existe");
            goto reentra_l;
        }
    break;
}

```

/* Controles para validacao da amostra */

```
conta (c);
```

```

if (num_clas_dist <= 2)
{
    if (contadorf == 5) mx++;
    if ((contadorf == 4) || (contadorf == 1)) mt++;
    if ((contadorf == 3) || (contadorf == 2)) ml++;
    for (i=0;i<4;i++)
        contador_classe [c[i]]++;
    dia_erro(" Amostra OK !!! Teclre <ENTER> ...");
}
else
{
    dia_erro("Nao pode haver mais que duas classes na amostra !");
    goto reentra_c;
}

```

```
fim: return(ERRO_NORMAL);
```

```
}
```

```
/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
```

```
SUBROTINA      : conta (.)
```

```
Autor         : Ronei Marcos de Moraes          ago-91
```

```
Resumo        : conta o numero de classes distintas na amostra
```

```
Entradas      : vetos de amostra
```

```
Saidas        : numero de classes distintas  
                contador de tipos iguais
```

```
/*XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*/
```

```
conta(c)
```

```
int c[5];
```

```
{
```

```
int      aux,i,j,          /* variaveis auxiliares          */  
troca,    /* controle de trocas efetuadas      */  
maximo,   /* maximo valor da amostra       */  
cont_class, /* contador auxiliar de classes  */  
contador[5]; /* contador amostral da classe  */
```

```
/* Ordena os tipos */
```

```
maximo = 0;  
for (i=0; i<4; i++)  
{  
  for (j=0; j<4; j++)  
  {  
    if (c[j] > c[j+1])  
    {  
      aux = c[j];  
      c[j] = c[j+1];  
      c[j+1] = aux;  
    }  
  }  
}
```

```
for (i=0; i<=4; i++)  
{  
  contador[i] = 1;  
}
```

```
/* Calcula o numero de classes distintas */
```

```
maximo = c[0];  
cont_class = 1;  
troca = 0;  
  
for (j=1; j<=4; j++)  
{  
  if (maximo != c[j])  
  {  
    maximo = c[j];  
    cont_class++;  
    troca++;  
  }  
  else  
    contador[troca]++;  
}
```

```
/* Passa valores de cont_class e contador[0] para as  
   variaveis globais num_clas_dist e contadorf */
```

```
num_clas_dist = cont_class;
```

```

#####
Nome      : funcao          VI-00
Autor     : Ronei Marcos de Moraes   out-91
Resumo    : formacao da tabela dos limiares
           por valores "default" = minimo da
           tabela normal de mxvcon.
           Ha a opcao de modificacao conforme
           conveniencia do usuario, para valores
           no intervalo [0,1], sendo e[i]=0,
           entendido como valor livre.

Entradas  : estrutura maxver
           nome do arquivo maxver

Saidas   : arquivo E.CON

VI-00    : Ronei Marcos de Moraes - UFPb

```

```
#####
```

```

#include <stdio.h>
#include "maxver.h"
#include "dia.h"

/* programa principal */

double          e[21];
struct arquivo_maxver maxver;      /* arquivo maxver      */
struct contexto_maxver cont;       /* arquivo de contexto */
int             nc,                /* numero de classes   */
              l,c;                /* LINHA E COLUNA      */
float           aux;

main ()
{
    FILE          *fuco;          /* arquivo de saida funcao_e */
    char          character;
    int           i,numero;

    /*          uvi_inicio(); */          /*SITIM_340*/

    dia_titulo(" Programa FUNCAO_ERRO");

    /* leitura do arquivo de contexto */
    if(ler_arq_cont(&cont) == ERRO_CTE)
    {
        dia_erro(" ERRO NA LEITURA DO CONTEXTO");
        goto final;
    }

    /* leitura do arquivo maxver */
    if (ler_arq_max(cont.nome_arquivo_max,&maxver) == ERRO_MNE)
    {
        dia_erro("ERRO NA LEITURA DO ARQ. MAXVER");
        goto final;
    }

    dia_poe_string(4,18,"Nome da imagem : ",maxver.nome_imagem);

    /* escreve classes e valores default para funcao_e como o
       minimo da tabela normal de mxvcon.c          */

    nc = maxver.num_classes;
    for (i = 1; i <= nc; i++)
        e[i]= 0.0001;

    soma_erro();
}

```

```
/* opcao para usuario de alteracao ou nao */
```

```
pergi:
```

```
switch(dia_peg_string(l+=2,18," a - Altera, CR - termina","Alteracoes: ",&caracter,1))
{
  case FIM      : goto fim;
  case VOLTA   : goto fim;
  case VAZIO   : goto fim;
  case NORMAL  :
  {
    if (caracter != 'a')
    {
      dia_erro ("Resposta invalida");
      l-=2;
      dia_limpa(l,18);
      goto pergi;
    }
    else
      goto tabela;
  }
}
```

```
tabela : /* altera a tabela para valores escolhidos pelo usuario */
```

```
dia_limpa(5,18);
switch(dia_peg_int(10,18," num inteiro,CR 0","Numero da classe a ser alterada :",&numero))
{
  case FIM      : goto fim;
  case VOLTA   : goto fim;
  case VAZIO   : {
      dia_erro("Classe inexistente");
      goto tabela;
    }
  /* case NORMAL : break; */
}
```

```
l=11;
```

```
if ((numero > nc) || (numero <= 0))
```

```
{
  dia_erro("Valor de entrada ilegal. Reentre !");
  goto tabela;
}
```

```
else
```

```
{
  dia_poe_texto(l+=2,18," Edite o valor :",44,40,0,0);
  dia_cursor(l++,18);
  printf ("%3i %-13s %10.8f \n",numero,maxver.nome_classe[numero],e[numero]);
  dia_poe_texto(l++,18," Trocar por :",44,40,0,0);
  dia_cursor(l++,18);
  printf ("%13s : ",maxver.nome_classe[numero]);
  scanf ("%f",&aux);
  e[numero] = aux;
}
```

```
switch(dia_peg_string(l+=2,18," s - troca, CR - nao","Outras Trocas ? ",&caracter,1))
```

```
{
  case FIM      : goto final1;
  case VOLTA   : goto final1;
  case VAZIO   : goto final1;
  case NORMAL  :
  {
    if (caracter != 's')
    {
      dia_erro ("Resposta invalida");
      l-=2;
      dia_limpa(l,18);
      goto pergi;
    }
    else
      goto tabela;
  }
}
```

```

    }
}

final: /* checa se os valores de e[i] estao entre [0,1] */
for (i = 1; i <= nc; i++)
{
    if (e[i]>=0 & e[i]<=1);
    else
    {
        dia_erro(" Valor de e[i] ilegal !!! Corrija ! ");
        goto tabela;
    }
}

soma_erro();

fim: /* finalizando o programa */
dia_pos_texto (23,18,"FINALIZANDO SESSAO.",31,40,0,0);
fuce = fopen ("E.CON","w");

for (i = 1; i <= nc; i++)
    fprintf(fuce,"%10.8f\n",e[i]);

fclose (fuce);
final: /* fim do programa */

printf("");
}

```

```

/*****
SUBROTINA      : soma_erro ()

Autor          : Ronei Marcos de Moraes      ago-91

Resumo         : soma e imprime os valores dos erros
Entradas       : vetor de erro (nao explicita)
Saídas         : imprime o vetor de erro e soma
*****/

```

```

soma_erro ()
{
    float soma;
    int i;

    soma = 0.;
    dia_limpa(5,18);
    l=6;
    dia_cursor(1,18);
    printf("Num.   Classe   limiar");
    l++;

    for (i=1; i<= nc; i++)
    {
        soma += e[i];
        dia_cursor(l++,18);
        if (l>17)
        {
            dia_erro("Tecla <ENTER> para continuar !");
            dia_limpa(5,18);
            l=6;
            dia_cursor(1,18);
            printf("Num.   Classe   limiar");
            dia_cursor(l++,18);
        }
    }
}

```

```
printf ("%3i %-14s %10.8f \n",i,maxver.nome_classe[i],e[i]);  
]  
dia_cursor(i+=2,18);  
printf (" SOMA = %10.8f\n",soma);  
return (soma);  
  
/* fim de soma_erra */
```

```
}
```



```

/*-----
leitura dos arquivos : contexto , maxver , informacoes , configuracao
contextual e funcao.
-----*/
    uvi_inicio();                                     /*LMN3*/

    dia_titulo (" Classificacao CONTEXTUAL");

    /* leitura do arquivo de contexto */
    if(ler_arq_cont(&cont) == ERRO_CTE)
    {
        dia_erro(" ERRO NA LEITURA DO CONTEXTO");
        goto final;
    }

    /* leitura do arquivo maxver */
    if (ler_arq_max(cont.nome_arquivo_max,&max) == ERRO_MNE)
    {
        dia_erro("ERRO NA LEITURA DO ARG. MAXVER");
        goto final;
    }

/* */ /* leitura do arquivo contextual */
    if ((contex = fopen("contexto.con","r")) == NULL)
    {
        dia_erro("ERRO NA LEITURA DO ARG. CONTEXTUAL");
        goto final;
    }
    else
    {
        for (i = 1; i <= max.num_classes; i++)
            fscanf(contex,"%f", &pik[i]);
        fscanf(contex,"%f", &p);
        fscanf(contex,"%f", &q);
        fscanf(contex,"%f", &r);
    }

/* */ /* leitura do arquivo funcao_e */
    if ((fuce = fopen("e.con","r")) == NULL)
    {
        dia_erro("ERRO NA LEITURA DO ARG. FUNCAO_E");
        goto final;
    }
    else
    {
        for (i = 1; i <= max.num_classes; i++)
            fscanf(fuce,"%f", &e[i]);
    }

    /* escreve no terminal o nome da imagem */
    dia_poe_string(3,18,"Nome da imagem      : ",max.nome_imagem);

    /* leitura do arquivo de informacoes */
    le_info(&info);

/*-----
                                Classificacao na tela ou no disco ?
-----*/

pergi:
    dia_limpa (4,18);
    tela = 1;
    switch(dia_peg_string(4,18,"CR          tela,d          disco","Local          : ",
                        &string,1))
    {

```

```

case FIM : goto final;
case VOLTA : goto perg1;
case VAZIO :
{
    dia_poe_string (4,18,"Local          : ", "t");
    break;
}
default :
{
    switch(string)
    {
        case 't' : break;
        case 'd' :
        {
            tela = 0;
            break;
        }
        default :
        {
            dia_erro ("Resposta invalida");
            goto perg1;
        }
    }
    break;
}
}
}

```

```

/*-----
      leitura dos atributos
-----*/

```

```

/* se a classificacao e na tela */

```

```

if (tela == 1)

```

```

{

```

```

    switch(le_atributos('t',"tela",atr_tela,&nbandas_tela))

```

```

    {

```

```

        case ERRO_NORMAL : break;

```

```

        case ERRO_INA :

```

```

        {

```

```

            dia_erro ("Imagem nao existe na tela");

```

```

            goto perg1;

```

```

        }

```

```

        default :

```

```

        {

```

```

            dia_erro ("Erro na abertura do arquivo de atributos na tela");

```

```

            goto final;

```

```

        }

```

```

    }

```

```

    nbandas = nbandas_tela;

```

```

    memcpy(&atr[0],&atr_tela[0],sizeof(atr_tela));

```

```

}

```

```

else

```

```

/* classificacao no disco */

```

```

{

```

```

    switch(le_atributos('d',max.nome_imagem,atr,&nbandas))

```

```

    {

```

```

        case ERRO_NORMAL : break;

```

```

        case ERRO_INA :

```

```

        {

```

```

            dia_erro("Imagem nao existe no disco");

```

```

            goto perg1;

```

```

        }

```

```

        default :

```

```

        {

```

```

            dia_erro ("Erro na abertura do arquivo de atributos no disco");

```

```

            goto final;

```

```

        }

```

```

    }

```

```

switch(le_atributos('t',"tela",atr_tela,&nbandas_tela))
{
    case ERRO_NORMAL : break;
    case ERRO_INA :
    {
        dia_erro("Imagem nao existe na tela");
        goto final;
    }
    default :
    {
        dia_erro ("Erro na abertura do arquivo de atributos na tela");
        goto final;
    }
}
}

/*-----
verifica consistencia dos atributos
-----*/

if (cheque_tela_disco (atr,atr_tela,&max,tela) != ERRO_NORMAL)
{
    dia_erro("Erro ao checar consistencia das imagens");
    goto final;
}

/*-----
verifica consistencia das bandas
-----*/

if(verifica_bandas(&max, atr, indice, nbandas, tela) == ERRO_CON)
{
    dia_erro("Bandas adquiridas nao sao consistentes c/ os atributos");
    goto final;
}

/*-----
monta tabela grafica padrao (classificada na tela)
-----*/

if (tela == 1)
{
    monta_tab_graf (&info.lut_graf[0][0],&info.lut_graf[1][0],&info.lut_graf[2][0]);
    esc_info (&info);
}

/*-----
monta tela de classificacao
-----*/

perq2 :

switch (monta_tela_classif(&max, tela))
{
    case ERRO_NORMAL : break;
    case VOLTA : goto perq1;
    case FIM : goto final;
}

/*-----
escolher coordenadas na tela
-----*/

/* definicao das coordenadas iniciais do cursor */
janela[0] = info.voo_x + ((info.num_col_vis / 2) - 100. ) / info.zoom; /* LMN3 */

```

```
janela[2] = info.voo_y + ((info.num_lin_vis / 2) - 100. ) / info.zoom; /* LMN3 */
janela[1] = janela[3] = 200. / info.zoom; /* LMN3 */
```

```
uvi_le_mux(&canr,&canb,&canb);
```

```
perq3:
```

```
dia_limpa(23,1);
dia_poe_texto(23,31,"Escolha area a ser classificada",33,40,5,1);
status = uvi_cursor(canr, janela);
dia_limpa(23,31);
switch (status)
{
    case FIM : goto final;
    case VOLTA : goto perq2;
}
```

```
/* le coordenadas para classificacao */
```

```
if (tela == 0)
    calcula_coord_disco(janela,atr_tela,coord_classif);
else
{
    memcpy(coord_classif,janela,sizeof(coord_classif));
    uvi_limpa (3); /* limpa canal 3 */
}
```

```
/*-----*/
```

```
escreve atributos da imagem de saida
```

```
Esta imagem sera escrita no canal 3 da tela se a classificacao
for na tela. Caso contrario, sera escrita na primeira banda
disponivel no disco
```

```
-----*/
```

```
if(esc_atr_saida(atr,atr_tela,&ats,&nbandas,&nbandas_tela,
                coord_classif,janela,tela) == ERRO_EAS)
```

```
    goto final;
dia_limpa (23,1);
```

```
/*-----*/
```

```
classifica imagem
```

```
-----*/
```

```
if(tela == 1)
```

```
{
    /* monta associacao cor-tema */
    memset (&info.cor_tema[0],0,sizeof(info.cor_tema));
    for (i = 0; i <= max.num_classes; i++)
    {
        if (max.tema_cor[i] != 0) info.cor_tema[max.tema_cor[i]] = i;
    }
}
```

```
/* atualiza estrutura informacoes */
esc_info (&info);
```

```
uvi_lut_comb();
```

```
}
```

```
/* */ status=classifica_imagem_cont(atr,&ats,&max,indice,e,pik,p,q,r,coord_classif);
```

```
if (status == ERRO_CLA)
```

```
{
    dia_limpa (23,1);
    dia_erro("Erro na classificacao");
    goto final;
}
```

```
/*-----*/
```

```
atualiza no disco arquivo maxver (classif. no disco)
```

```
-----*/
```

```

if (tela == 0)
{
    if (escreve_arq_max (cont.nome_arquivo_max,&max) != ERRO_NORMAL)
    {
        dia_erro ("Erro de escrita no arquivo maxver");
        goto final;
    }
}

/*-----
    escreve no disco/ramdisk arquivo tabela
-----*/

memset (tab,0,sizeof(tab));
fator = ats.tam_pix_h * ats.tam_pix_v / 1000000.;
for (i = 1; i <= max.num_classes; i++)
{
    tab[i].tab_area = (float)max.num_classif_classe[i] * fator;
    tab[i].tab_cor = max.tema_cor[i];
    strcpy (tab[i].tab_rotulo,max.nome_classe[i]);
}

dia_limpa(12,18);
if (tela == 1)
    status = escreve_tab ('t',"tela",ats.banda,tab,max.num_classes+1);
else
    status = escreve_tab ('d',max.nome_imagem,ats.banda,tab,max.num_classes+1);

if (status != TAB_NORMAL)
{
    dia_erro ("Erro de escrita do arquivo tabela");
    goto final;
}

/*-----
    escreve no disco arquivo lut (classificacao no disco)
-----*/

if (tela == 0)
{
    memset (lut_ima,0,sizeof(lut_ima));
    for (i = 0; i < 3; i++)
        memcpy (&lut_ima[i][0],&info.lut_graf[i][0],128);

    escreve_arq_lut (max.nome_imagem,ats.banda,lut_ima);
}

/*-----
    mostra area classificada
-----*/

mostra_area_classif (&ats,&max);                                     /*FAM101*/

final:
    dia_limpa (23,1);
    dia_cursor (23,0);
    exit (1);
}

```

```
#####
```

```
ARQUIVO DE SUBROTINA : CONTEXT V 1.00
```

```
FUNCAO : calculo do fator contextual em cruz  
de 5 pontos.
```

```
SUBROTINAS : classifica_imagem_cont,  
context  
densid.
```

```
V 1.00 : Ronei Marcos de Moraes - UFFb, nov_91.
```

```
#####
```

```
#include <stdio.h>  
#include <memory.h>  
#include "maxver.h"  
#include "atr.h"  
#include "dia.h"  
#include <math.h>  
#include <config.h>
```

```
/* tabela normal de pontos mais provaveis */
```

```
float normal [2][32] = { 1.000, 0.900, 0.750, 0.600, 0.500, 0.400, 0.333, 0.250,  
0.200, 0.166, 0.143, 0.125, 0.111, 0.100, 0.090, 0.083,  
0.076, 0.071, 0.066, 0.062, 0.058, 0.055, 0.052, 0.050,  
0.040, 0.030, 0.020, 0.010, 0.005, 0.001, 0.0005, 0.0001,  
  
0.680, 0.760, 0.890, 1.040, 1.150, 1.290, 1.390, 1.540,  
1.650, 1.730, 1.800, 1.860, 1.920, 1.960, 2.000, 2.040,  
2.070, 2.100, 2.130, 2.150, 2.180, 2.200, 2.220, 2.240,  
2.340, 2.440, 2.580, 2.820, 3.020, 3.520, 3.670, 4.050  
};
```

```
int nband;  
char buffer[3][BUF_IMA*3]; /* bufer de classificacao */  
double probab, /* valor da probabilidade do pto */  
lim[21], /* limiar da transformacao */  
rk;  
float p,q,r, /* parametros classif. contextual */  
e[21], /* vetor erro */  
pik[21], /* vetor de probs a priori */  
norte[TOT_BAND], /* valor dos pixels do contexto */  
sul[TOT_BAND],  
centro[TOT_BAND],  
leste[TOT_BAND],  
oeste[TOT_BAND];
```

```
#####
```

```
SUBROTINA : classifica_imagem_cont  
(adaptado de maxvlib2).
```

```
Subrotinas : densid,context.
```

```
V 1.01 : Ronei Marcos de Moraes - UFFb, set_91.
```

```
#####
```

```
classifica_imagem_cont(atr,ats,max,indice,ei,piki,pp,qq,rr,janela)
```

```
struct atributos atr[];  
struct atributos *ats;  
struct arquivo_maxver *max;  
float pp,qq,rr, /* parametros classif. contextual */  
ei[], /* vetor erro */  
piki[]; /* vetor de probs a priori */  
  
int indice[],  
janela[];
```

```

char          bufout[BUF_IMA],      /* bufer de saida          */
              #p1[TOT_BAND],       /* ponteiro para linha1    */
              #p2[TOT_BAND],       /* ponteiro para linha2    */
              #p3[TOT_BAND];       /* ponteiro para linha3    */

int           classe,              /* nro. da classe          */
              i, j, k, jcia,        /* variaveis auxiliares    */
              ks, ci, l, ind,      /* variaveis auxiliares    */
              fat,                 /*                          */
              linha,               /* ponteiro para a linha   */
              coluna,              /* ponteiro para a coluna  */
              canal,               /* nro. da banda           */
              status,              /* variavel auxiliar       */
              classe1,             /* variavel auxiliar       */
              kmini;

float         mat_covar[3*TAM_MATR], /* matriz de covariancias */
              ze,                  /* variavel auxiliar       */
              rptoj,               /* distancia do pto a media */
              rsoma,               /* resultado intermediario */
              rfli,                /* variavel auxiliar       */
              poster,              /* probabilidade a posteriori */
              mini;                /* valor minimo            */

/* zera numero de pontos classificados de cada classe */
for (i = 0; i <= max->num_classes; i++)
    max->num_classif_classe[i] = 0;

nband = max->num_ban_aquis;
for (i = 1; i <= max->num_classes; i++)
{
    pik [i] = piki[i];
    e[i] = ei[i];
}

p = pp; q = qq; r = rr;

/* montagem da matriz de covariancias */
for(classe=1; classe <= max->num_classes; classe++)
{
    k = (classe-1)*TAM_MATR;
    ind = TOT_BAND*(classe-1);
    for(i=0; i <max->num_ban_aquis; i++)
        for(j=0; j<=i; j++)
            mat_covar[k++] = max->media_classe[ind + i]*max->media_classe[ind+j];

    k = TAM_MATR*(classe-1);
    fat = (max->num_ban_aquis * (max->num_ban_aquis + 1))/2 ;
    for (i=0; i < fat; i++)
        mat_covar[k++] = max->mat_auto[k] - mat_covar[k];
}

/* transformacao do erro aceitavel */
for (i = 1; i <= max->num_classes; i++)
{
    long comp1 = 100000 * e[i];
    ze = 0.0;

    for (j=0; j<32; j++)
    {
        long comp2 = 100000 * normal [0][j];
        if (comp2 <= comp1)
        {
            if (comp1 <= comp2)
            {
                ze = normal [1][j];
            }
        }
    }
}

```

```

        break;
    }
    else
    {
        ze = normal [i][j-1];
        break;
    }
}
}

/* */ if (ze==0) ze= 5.00; /* valor extremo para ze */
k = (1-1)*TOT_BAND;
ks = TAM_MATR*(i-1);
for (j=0; j<nband; j++)
    centro[j] = max->media_classe[k+j] - ze * sqrt(mat_covar[ks+(j*(j+3)/2)]);
densid(max,&i,centro);
rk = exp(4*log10(probab)*log(10));
lim[i] = probab * pik[i] * rk;
}

/* abre arquivos de imagens para cada banda adquirida */
for(canal = 0; canal < nband; canal++)
    if(abre_imagem(&atr[indice[canal]]) != ERRO_NORMAL) return(ERRO_CLA);

/* abre arquivo para imagem de saida */
if(abre_imagem(ats) != ERRO_NORMAL) return(ERRO_CLA);

/* classifica imagem */
for (jcla = 1; jcla <= max->num_classe; jcla++) /* 1a. classe a ser classif. */
    if (max->classes_classif[jcla] == 1) break;

/* controle de processamento */
linha = janela[3]-1;
coluna = janela[1]-2;
dia_poe_int (12,18,"Numero de linhas : ",linha);
dia_poe_int (14,18,"Numero de colunas : ",coluna);
dia_poe_texto (16,18,"Processando !!!",32,45,1,0);

/* le linhas da imagem */
/* inicializando buffer rotativo */
for(canal = 0; canal < nband; canal++)
{
    if(le_linha(&atr[indice[canal]],0,&buffer[0][BUF_IMA#canal],janela)!= ERRO_NORMAL)
        return(ERRO_CLA);
    p1[canal] = &buffer[0][BUF_IMA#canal];

    if(le_linha(&atr[indice[canal]],1,&buffer[1][BUF_IMA#canal],janela)!= ERRO_NORMAL)
        return(ERRO_CLA);
    p2[canal] = &buffer[1][BUF_IMA#canal];
}

linha = 1; coluna = 1;
dia_poe_int (18,18,"Linha central : ",linha);
dia_poe_int (20,18,"Coluna central : ",coluna);

for(linha = 2; linha < janela[3]; linha++)
{
    if (kbhit())
        while ((status=getch())!=0) continue;
    if (status==ESC)
    {
        printf("\x1b[0m");
        fecha_imagem(ats);
        for(canal = 0; canal < nband; fecha_imagem(&atr[indice[canal++]])); /*VDR01*/
        return(status);
    }
}
/*VDR01*/

```

```

for (canal = 0; canal < nband; canal++)
{
  if(!le_linha(&atr[indice[canal]],linha,&buffer[2][BUF_IMA#canal],janela)!= ERRO_NORMAL)
    return(ERRO_CLA);
  p3[canal] = &buffer[2][BUF_IMA#canal];
}

for (coluna=1; coluna<=janela[1]-2; coluna++)
{
  for (canal = 0; canal < nband; canal++)
  {
    norte[canal] = *(p1[canal] + coluna);
    sui[canal] = *(p3[canal] + coluna);
    oeste[canal] = *(p2[canal] + coluna - 1);
    centro[canal]= *(p2[canal] + coluna);
    leste[canal] = *(p2[canal] + coluna + 1);
  }

  k=1;
  context(max,k);
  poster = fabs(log(rk));
  poster += fabs(log(pik[k]));

  densid(max,&k,centro);
  poster += fabs(log(probab));

  mini = poster;
  kmini = 1;

  for (k=2; k<= max->num_classes; k++)
  {
    densid(max,&k,centro);
    poster = fabs(log(pik[k])) + fabs(log(probab));
    if (poster < mini)
    {
      context(max,k);
      poster += fabs(log(rk));

      if (poster < mini)
      {
        mini = poster;
        kmini = k;
      }
    }
  }
  if (mini > fabs(log(lim[kmini]))) kmini = 0;
  bufout [coluna] = max->tema_cor[kmini];
  max->num_classif_classe[kmini] +=1;

  /* controle de processamento atualizado */
  dia_poe_int (18,18,"Linha central : ",linha-1);
  dia_poe_int (20,18,"Coluna central : ",coluna);
}

/* loop das colunas */

/* mecanismo do buffer rotativo */
for (canal = 0; canal < nband; canal++)
{
  if(!le_linha(&atr[indice[canal]],linha-1,&buffer[0][BUF_IMA#canal],janela)!= ERRO_NORMAL)
    return(ERRO_CLA);
  p1[canal] = &buffer[0][BUF_IMA#canal];

  if(!le_linha(&atr[indice[canal]],linha,&buffer[1][BUF_IMA#canal],janela)!= ERRO_NORMAL)
    return(ERRO_CLA);
  p2[canal] = &buffer[1][BUF_IMA#canal];
}

```

```
/* p3 eh lido no retorno ao inicio do for de linhas */
```

```
escreve_linha(ats,linha-1,bufout,janela);
```

```
/* loop das linhas */
```

```
/* fecha arquivo das imagens */
```

```
printf ("\x1b[0m"); /* desliga atributo */
```

```
/*VDR01*/
```

```
fecha_imagem(ats);
```

```
for(canal = 0; canal < nband; fecha_imagem(&atr[indice[canal++]]));
```

```
return(ERRO_NORMAL);
```

```
#####
```

```
SUBROTINA : context
```

```
Subrotinas : densid
```

```
V 1.00 : Ronei Marcos de Moraes - UFPb, ago_91.
```

```
#####
```

```
context (max,k)
```

```
struct arquivo_maxver #max;
```

```
int #k;
```

```
{
```

```
double ak,bk,ck, /* equacoes descritas em (2.17) */
```

```
a,b, /* equacoes descritas em (2.19) */
```

```
dkn,dks,dkl,dko, /* variaveis auxiliares */
```

```
bso,bon,bn1,bls,
```

```
bso2,bon2,bn12,bls2,
```

```
an,as,al,ao;
```

```
int m;
```

```
densid(max,&k,norte);
```

```
dkn = probab;
```

```
densid(max,&k,sul);
```

```
dks = probab;
```

```
densid(max,&k,leste);
```

```
dkl = probab;
```

```
densid(max,&k,oeste);
```

```
dko = probab;
```

```
if (p != 0.0)
```

```
ak = dkn * dks * dkl * dko;
```

```
else
```

```
ak=0.0;
```

```
if ((q != 0.0) || (r != 0.0))
```

```
{
```

```
an=0; as=0; al=0; ao=0;
```

```
bso=bso2=0; bon=bon2=0; bn1=bn12=0; bls=bls2=0;
```

```
for (m = 1; m <= max->num_classes; m++)
```

```
{
```

```
densid(max,&m,norte);
```

```
bn12 += pik[m] * probab;
```

```
bon2 += probab;
```

```
an += pik[m] * probab;
```

```
densid(max,&m,sul);
```

```
bso2 += pik[m] * probab;
```

```

    bls2 += probab;
    as += pik[m] * probab;

    densid(max,&m,leste);
    bls2 *= pik[m] * probab;
    bn12 *= probab;
    al += pik[m] * probab;

    densid(max,&m, oeste);
    bon2 *= pik[m] * probab;
    bso2 *= probab;
    ao += pik[m] * probab;

    bso += bso2;
    bon += bon2;
    bn1 += bn12;
    bls += bls2;
}

bk = 0.25 * (dkn * dkl * bso + dkl * dke * bon + dks * dko * bn1 + dko * dkn * bls);
ck = 0.25 * (dkn * dkl * dks * ao + dkn * dkl * as * dko + dkn * al * dks * dko + an * dkl * dke * dko);
}
else
{
    bk=0.0;
    ck=0.0;
}

rk = p * ak + q * bk + r * ck;

    /* prevencao contra estouro de ponto flutuante */
    if (rk < 1E-100) rk = 1E-50;
}

```

```

/*****
SUBROTINA      : densid

V 1.00        : Ronei Marcos de Moraes - UFPb, ago_91.
*****/

```

```
densid (max,clas,ponto)
```

```

struct arquivo_maxver #max;
int #clas;
float ponto[];

```

```
{
```

```

int
    i,j,          /* variaveis auxiliares */
    ks,l;        /* variaveis auxiliares */

```

```

double
    pi,          /* valor de pi */
    aux;        /* variavel auxiliar */

```

```

float
    rptoj,      /* dist. pto a media */
    rsoma;     /* resultado intermediario */

```

```

pi = 3.141592654;
ks = (#clas-1)*TOT_BAND;
l = (#clas-1)*TAM_MATR;

```

```

rpto = 0.0;
for(i = 0; i < nband; i++)
{
  rsoma = 0.0;
  i += 1;
  for(j = 0; j <= i; j++)
    rsoma += max->mat_transf[1+j]*(ponto[i] - max->media_classe[ks+j]);

  rpto += rsoma*rsoma;
}

aux = exp((1.0*nband/2.0)*log(2.0*pi));
aux = aux * sqrt(max->determ[*clas]);
probab = 1/aux * exp (-0.5*rpto);

/* prevencao contra estouro de ponto flutuante */
if (probab < 1E-100) probab = 1E-50;
}

```

ÍNDICE BIBLIOGRÁFICO

Baridó	capítulo 1
Barros, L.A.	capítulo 1
Barros, M.A.	capítulo 1
Besag	introdução
Bezdek, Chuah & Leep	introdução, capítulo 2 e capítulo 5
Ciampi	capítulo 1
Cola	introdução
Cooper & Freeman	introdução e capítulo 1
Duarte	capítulo 1
Dunn	introdução e capítulo 1
Eigen, Fromm & Northouse	introdução
Fisher	capítulo 1
Fu & Yu	capítulo 1
Garcia	capítulo 1
Gidas	introdução
Govidan & Shivaprasad	introdução
Guebert & Gardner	introdução
Hjort, & Mohn	introdução e capítulo 2
Hjort	introdução, capítulo 2, e capítulo 5
Hjort, Mohn, & Storvik	introdução e capítulo 2
James	capítulo 1

Johnson	capítulo 1
Jones & Byrne	introdução
Kent & Mardia	introdução e capítulo 1
Klein & Press	introdução, capítulo 2 e capítulo 5
Landgrebe	introdução e capítulo 2
Lowell	introdução
Manual do Sitim - Programação	capítulo 3
Manual do Sitim - Usuário	capítulo 3, capítulo 4 e capítulo 5
Mardia	introdução
Mascarenhas e Velasco	capítulo 1
Owen	introdução e capítulo 2
Owen & Switzer	introdução e capítulo 2
Ripley	capítulo 2
Rudorff et al	capítulo 1
Saebo et al	introdução, capítulo 1 e capítulo 2
Sammon Jr	introdução
Saniie & Nagle	introdução
Skidmore	introdução
Swain & Davis	capítulo 1
Swain, Siegel, & Smith	introdução e capítulo 2
Swain, Vanderman, & Tilton	introdução e capítulo 2
Switzer	capítulo 2
Therrien, Quatieri & Dudgeon	introdução
Tilton & Swain	introdução e capítulo 2

Tilton, Vanderman & Swain

introdução e capítulo 2

Tucker

capítulo 1

Welch & Salter

introdução e capítulo 2

Whorton

introdução e capítulo 2

ÍNDICE ANALÍTICO

A	
Absorção	8
Amostragem aleatória	44
Amostras homogêneas	43, 48
de cruces	44
de treinamento	43, 44
Arquivos de saída de FUNCAOE	63
MAXVER	62, 63, 64
MPQR	63
Assinatura Espectral	8
Atenuação de contraste	11
B	
Bandas do satélite	26
Bayes	19, 20, 35
"Buffer" rotativo	59, 60, 64
C	
C, linguagem	53
Campo de Visão Instantânea	7
Canais	33, 59
Chaves de interpretação visual	27
Classes	13, 17
Classificação	
não-supervisionada	17
supervisionada	17, 18, 19
visual	26
Componentes Principais	13
CONAN	31
Confusão espectral	7, 8
CONTEXT	58, 61, 64, 68
contexto.con	63
Contraste, alteração	11
realce	11

Cor, falsa	12
Corte _k	50, 57, 58
Critério da Distância Euclidiana de Entropia	78, 82 14
Cruzes de pixels	22, 23, 38, 45
D	
Densidade de Probabilidade Normal ou Gaussiana	14, 37, 43, 48, 65
Disperção	8
Distância Euclidiana J-M de Mahalanobis	66, 78, 82 16 66
Distribuição Espectral	= assinat. espectral
Divergência média	14, 15 16
E	
e.con	58, 64, 65
ECHO	30
Entropia, critério	14
Equalização do Histograma	12
Espectro	5
Espacial, resolução	11
Espaço das Decisões	21, 33
Estimadores de contexto de densidades de probabilidade a priori	25, 44 24, 42 44
Euclidiana, distância	66
F	
Falsa cor	12
Fatiamento de Histograma de Temperatura	78, 81 78, 81
Fator de correção contextual	22, 40, 57, 58, 70
Filtros lineares não-lineares	12 12
Fisher	16

Fotointerpretação	= interpr. visual
FUNCAOE	58, 61, 62, 63
Função e	21, 47
g	22, 34, 39, 56, 57
h	22, 23, 37, 56, 57
indicadora	45
R_k	22, 37, 40, 41, 61, 90
"Fuzzy"	17, 31, 91
G	
g, função	22, 34, 39, 56, 57
graus de pertinência	23, 24
H	
h, função	22, 23, 37, 56, 57
Histograma	42, 43
Equalização de	12
Método do fatiamento	78, 81
I	
Imagens	5
classificação	1, 16
composição	9
incorporação de informações	12
interpretação visual	26
filtragem	11
multiespectrais	29
obtenção	6
processamento digital	10
realce	11
restauração	11
tratamento	12
J	
Janelas atmosféricas	9
Jeffries-Matusita, distância	16
L	
Limiares	= função e

Linguagem C	53
M	
Mahalanobis, distância	66
Matriz de Classificação	74, 75
Covariâncias	14, 37, 67
Transformação	67
Máxima Verossimilhança	26, 66, 78
MAXVCLA	62, 63, 80
MAXVER	54, 56, 57, 58, 61, 62, 64, 72, 78, 80, 82, 83, 87
Método não-Supervisionado	17
Supervisionado	17, 18, 19
Aproximadamente Contextual	31
Clássico	18
Contextual	19, 20
Híbrido	31
Relaxação	23, 24
MIXAQU	58, 62, 63
Modelo Discriminante de Fisher	16
Owen & Switzer	32
p,q,r	32, 33, 91
MPQR	57, 58, 65
MXVCON	58, 65, 67, 75, 76 84, 87
P	
Padrão X, L, T	38
Paralelepípedo, Método	78, 80
Parâmetro	24, 25
estimativa	24, 25, 42
"pixel"	7, 9
Pré-classificação	47
Probabilidade à priori	18, 24, 36, 39
à posteriori	19
de classificação	19
de erro	14

Processamento digital de imagens	9, 10
Processamento paralelo	30
R	
Realce de contraste	11
Recobrimento	7
Regiões homogêneas	43, 48
Remoto, Sensoriamento	9
Resolução Espacial	11
Risco Médio	18
Rotativo, "buffer"	59, 60, 64
Ruído	11
S	
Satélite LANDSAT	6
SPOT	33
Sensores ativos	5
passivos	6, 8
multiespectrais	6
Sensoriamento remoto	4
SITIM	53, 57, 72, 89, 90
Sombreamento	11
T	
Temperatura, método do fatiamento	78, 81
Teoria estatística de decisão	16
Transformação, matriz de	66
Treinamento, amostras	43, 44
do classificador	43, 44, 46
U	
UVI	79, 81
V	
Valor de Brilho	7
Verdade terrestre	18, 24, 44
Vetor característico	20
VIC	= Campo de Visao Instant
Vizinhança	20, 21, 34