

**TATIANA MONTES CELINSKI**

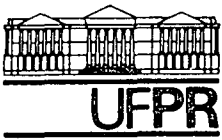
**MÉTODOS DE AGRUPAMENTO  
UMA ABORDAGEM COMPARATIVA COM APLICAÇÃO  
EM SEGMENTAÇÃO DE IMAGENS DE PROFUNDIDADE**

Dissertação apresentada como requisito parcial  
à obtenção de grau de Mestre. Curso de Pós-  
Graduação em Informática, Setor de Ciências  
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Olga R. P. Bellon

**CURITIBA**

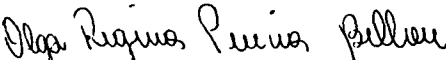
**1998**



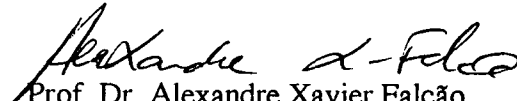
## PARECER


Nós, abaixo assinados, membros da Comissão Examinadora da defesa de Dissertação de Mestrado em Informática, da aluna Tatiana Montes Celinski, avaliamos o trabalho intitulado “**Métodos de Agrupamento: Uma Abordagem Comparativa com Aplicação em Segmentação de Imagens de Profundidade**”, cuja defesa foi realizada no dia 09 de novembro de 1998. Após a Avaliação, decidimos pela Aprovação da Candidata.

Curitiba, 09 de novembro de 1998

  
Prof.<sup>a</sup>. Dr.<sup>a</sup>. Olga Regina Pereira Bellon  
Presidente

  
Prof. Dr. Nelson Delfino D'Avila Mascarenhas

  
Prof. Dr. Alexandre Xavier Falcão

  
Prof. Dr. Alexandre Ibrahim Direne

Dedico este trabalho a meus pais, que destinaram muitos anos de suas vidas plantando amor e compreensão, na esperança de colher seus frutos. Dedico também a meu esposo e filhas, pelo apoio constante, sem o qual eu nada poderia.

## AGRADECIMENTOS

Ao Departamento de Informática da UFPR, seus professores e funcionários, por proporcionar todas as condições necessárias para o desenvolvimento do Curso de Mestrado.

De maneira especial aos professores André, Roberto, Picanço e Alexandre, que colaboraram de forma mais direta junto ao curso.

Ao Luciano e ao Daniel, alunos da iniciação científica, por sua contribuição.

Aos caros colegas de curso: Aldri, Nélis, Andrey e Elisa, e a todos os demais.

A minha orientadora Olga.

A Deus, sempre presente.

## SUMÁRIO

<b>LISTA DE ILUSTRAÇÕES .....</b>	<b>viii</b>
<b>LISTA DE TABELAS.....</b>	<b>xiii</b>
<b>LISTA DE QUADROS .....</b>	<b>xiv</b>
<b>RESUMO .....</b>	<b>xv</b>
<b>ABSTRACT .....</b>	<b>xvi</b>
<b>1 INTRODUÇÃO.....</b>	<b>1</b>
1.1 MOTIVAÇÃO .....	1
1.2 OBJETIVOS .....	5
1.3 CONTEÚDO .....	5
<b>2 ALGORITMOS E MÉTODOS DE AGRUPAMENTO .....</b>	<b>6</b>
2.1 MÉTODOS PARTICIONAIS .....	7
2.1.1 Funções Critério .....	8
2.1.1.1 Soma de Erro-quadrático .....	8
2.1.1.2 Caso <i>Hard</i> .....	10
2.1.1.3 Traço.....	11
2.1.1.4 Caso <i>Fuzzy</i> .....	13
2.1.2 Algoritmos Básicos.....	15
2.1.2.1 Algoritmo <i>K-means</i> .....	16
2.1.2.2 Algoritmo ISODATA .....	18

2.1.2.3 Algoritmo CLUSTER.....	20
2.1.2.4 Algoritmo <i>Fuzzy K-means</i> .....	21
2.2 MÉTODOS HIERÁRQUICOS .....	22
2.2.1 Métodos Aglomerativos.....	24
2.2.1.1 Algoritmos <i>Single-link</i> e <i>Complete-link</i> .....	27
2.2.2 Métodos Divisivos.....	30
2.2.3 Métodos Grafo-teóricos.....	33
2.2.3.1 Algoritmo <i>Single-link</i> .....	34
2.2.3.2 Algoritmo <i>Complete-link</i> .....	35
2.2.3.3 Árvore Geradora Mínima e o Algoritmo <i>Single-link</i> .....	37
<b>3 APLICAÇÕES .....</b>	<b>40</b>
3.1 BASEADAS EM MÉTODOS PARTICIONAIS .....	41
3.2 BASEADAS EM MÉTODOS HIERÁRQUICOS.....	79
3.3 DIAGRAMAS DE CLASSIFICAÇÃO.....	104
<b>4 EXPERIMENTOS .....</b>	<b>105</b>
4.1 ALGORITMOS IMPLEMENTADOS.....	106
4.2 RESULTADOS OBTIDOS .....	109
4.2.1 Primeira Fase.....	110
4.2.2 Refinamento da Segmentação .....	119
4.3 ABORDAGEM DESENVOLVIDA .....	124
<b>5 CONCLUSÃO .....</b>	<b>132</b>
5.1 CONTRIBUIÇÕES.....	132
5.2 TRABALHOS FUTUROS .....	134

<b>ANEXO 1 - MATRIZES ESPARSAS .....</b>	<b>136</b>
<b>ANEXO 2 - DISTÂNCIA MAHALANOBIS .....</b>	<b>139</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>140</b>

## LISTA DE ILUSTRAÇÕES

1 GRUPOS PRÓXIMOS, COM ALGUNS PONTOS INDEFINIDOS.....	13
2 FLUXOGRAMA SIMPLIFICADO DOS ALGORITMOS <i>K-MEANS</i> E <i>ISODATA</i> .....	17
3 <i>DENDROGRAM</i> DE LIMIAR .....	23
4 (A) GRUPOS BEM SEPARADOS, DE FORMA ESFÉRICA, GRUPOS DE FORMA UM POUCO ALONGADA, (C) GRUPOS INCIDENTES, (D) GRUPOS EM FORMA DE LINHA.....	26
5 <i>DENDROGRAMS</i> DE PROXIMIDADE .....	30
6 GRAFOS DE LIMIAR E <i>DENDROGRAMS</i> PARA OS AGRUPAMENTOS HIERÁRQUICOS <i>SINGLE-LINK</i> E <i>COMPLETE-LINK</i> .....	37
7 EXEMPLOS DE ALGORITMOS <i>SINGLE-LINK</i> AGLOMERATIVO E DIVISIVO BASEADOS NA ÁRVORE GERADORA MÍNIMA.....	39
8 ESTRUTURA DE UM AGRUPAMENTO KOHONEN .....	68
9 ESTRUTURA DE UMA PIRÂMIDE. O NÍVEL É DADO À DIREITA, E O TAMANHO DO NÍVEL EM <i>PIXELS</i> É DADO À ESQUERDA.....	80
10 UM EXEMPLO DE UMA ÁRVORE BINÁRIA CORRESPONDENDO A UM <i>HEAP</i> COM DEZ VALORES .....	91
11 IMAGENS (A) SINTÉTICA E (B) REAL DE AGPART2-1 .....	109
12 IMAGENS (A) SINTÉTICA E (B) REAL DE BIGWYE-1.....	110
13 IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	110



14	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS .....	111
15	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO CLUSTER: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	112
16	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO CLUSTER: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS .....	112
17	IMAGENS SINTÉTICAS DE AGPART2-1 SEGMENTADAS PELO ALGORITMO CLUSTER EM 5 ATÉ 12 GRUPOS.....	113
18	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO ISODATA: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	114
19	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO ISODATA: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS .....	114
20	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	115
21	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	115
22	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY CLUSTER</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	116
23	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY CLUSTER</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	116
24	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY ISODATA</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	117
25	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY ISODATA</i> : (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS.....	117

26	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 23 GRUPOS E (B) BIGWYE-1 COM 21 GRUPOS .....	120
27	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 16 GRUPOS E (B) BIGWYE-1 COM 33 GRUPOS FIGURA.....	120
28	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO CLUSTER EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 24 GRUPOS E (B) BIGWYE-1 COM 22 GRUPOS.....	121
29	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO CLUSTER EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 22 GRUPOS E (B) BIGWYE-1 COM 27 GRUPOS .....	121
30	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO ISODATA EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 16 GRUPOS E (B) BIGWYE-1 COM 21 GRUPOS .....	121
31	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO ISODATA EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 17 GRUPOS E (B) BIGWYE-1 COM 32 GRUPOS .....	122
32	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 26 GRUPOS E (B) BIGWYE-1 COM 39 GRUPOS.....	122
33	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 18 GRUPOS E (B) BIGWYE-1 COM 14 GRUPOS .....	122

34	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY CLUSTER</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 28 GRUPOS E (B) BIGWYE-1 COM 30 GRUPOS.....	123
35	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY CLUSTER</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 39 GRUPOS E (B) BIGWYE-1 COM 41 GRUPOS.....	123
36	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY ISODATA</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 31 GRUPOS E (B) BIGWYE-1 COM 34 GRUPOS.....	123
37	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY ISODATA</i> EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 23 GRUPOS E (B) BIGWYE-1 COM 42 GRUPOS.....	124
38	FASES DA DETEÇÃO DE BORDAS PARA A IMAGEM SINTÉTICA AGPART2-1	125
39	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS .....	126
40	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS .....	127
41	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS .....	127

42	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS .....	127
43	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 28 GRUPOS E (B) BIGWYE-1 COM 30 GRUPOS.....	128
44	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: AGPART2-1 COM 25 GRUPOS E (B) BIGWYE-1 COM 35 GRUPOS.....	128
45	IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 29 GRUPOS E (B) BIGWYE-1 COM 26 GRUPOS.....	128
46	IMAGENS REAIS SEGMENTADAS PELO ALGORITMO <i>FUZZY K-MEANS</i> EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 26 GRUPOS E (B) BIGWYE-1 COM 30 GRUPOS.....	129

## LISTA DE TABELAS

1	DISTÂNCIAS PARA OS GRUPOS “DIVISOR” E “RESTANTE” .....	32
2	DISTÂNCIAS PARA OS GRUPOS “DIVISOR” E “RESTANTE” .....	32
3	DISTÂNCIAS PARA OS GRUPOS “DIVISOR” E “RESTANTE” .....	33
4	TEMPO DE COMPUTAÇÃO E VALOR DO ERRO DOS ALGORITMOS.....	118
5	VALORES DE ERRO PARA O ALGORITMO <i>K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. SINTÉTICA AGPART2-1) .....	130
6	VALORES DE ERRO PARA O ALGORITMO <i>K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. SINTÉTICA BIGWYE-1).....	130
7	VALORES DE ERRO PARA O ALGORITMO <i>K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. REAL AGPART2-1) .....	130
8	VALORES DE ERRO PARA O ALGORITMO <i>K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. REAL BIGWYE-1).....	131
9	VALORES DE ERRO PARA O ALGORITMO <i>FUZZY K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. SINTÉTICA AGPART2-1) .....	131
10	VALORES DE ERRO PARA O ALGORITMO <i>FUZZY K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. SINTÉTICA BIGWYE-1).....	131
11	VALORES DE ERRO PARA O ALGORITMO <i>FUZZY K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. REAL AGPART2-1) .....	131
12	VALORES DE ERRO PARA O ALGORITMO <i>FUZZY K-MEANS</i> COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IM. REAL BIGWYE-1).....	132

## LISTA DE QUADROS

1 <i>SIMULATED ANNEALING</i> X PROBLEMA DO AGRUPAMENTO .....	76
--	----

## RESUMO

Este trabalho apresenta um estudo comparativo sobre métodos de agrupamento, incluindo diferentes abordagens para segmentação de conjuntos de dados que utilizam métodos de agrupamento. As abordagens são baseadas nas duas técnicas de agrupamento mais populares, agrupamento particional e agrupamento hierárquico, e vêm introduzir heurísticas que, de alguma forma, levem à obtenção de um melhor resultado para o agrupamento. A partir das implementações de diversos métodos de agrupamento, este trabalho introduz uma nova heurística para a segmentação de imagens de profundidade por agrupamento, com o objetivo de melhorar o resultado da segmentação obtendo uma partição melhor para o agrupamento. A abordagem desenvolvida é baseada na extração de informações, para inicialização dos algoritmos de agrupamento, a partir de uma pré-segmentação da imagem usando detecção de bordas.

## ABSTRACT

This research presents a comparative study on clustering methods including different approaches for the segmentation of data sets which use clustering methods. The approaches are based on the two more popular clustering techniques — partitional clustering and hierarchical clustering — and introduce heuristics which, to a certain extent, lead to the achievement of a better result in clustering. Beginning with the implementation of several clustering methods, this study introduces a new heuristic to segment range images for clustering, with the aim of improving the result of segmentation in order to obtain a better partition for the clustering. The approach is based on the extraction of information for initialising clustering algorithms, starting by applying edge detection to obtain a pre-segmented image.



## 1 INTRODUÇÃO

Este trabalho apresenta um estudo comparativo sobre métodos de agrupamento, incluindo diferentes abordagens para segmentação de conjuntos de dados que utilizam métodos de agrupamento. As abordagens são baseadas nas duas técnicas de agrupamento mais populares: (1) agrupamento particional, e (2) agrupamento hierárquico, e vêm introduzir heurísticas que, de alguma forma, levem à obtenção de um melhor resultado para o agrupamento.

A partir das implementações de diversos métodos de agrupamento, este trabalho introduz uma nova heurística para a segmentação de imagens de profundidade por agrupamento, com o objetivo de melhorar o resultado da segmentação obtendo uma partição melhor para o agrupamento. A abordagem desenvolvida é baseada na extração de informações, para inicialização dos algoritmos de agrupamento, a partir de uma pré-segmentação da imagem usando detecção de bordas.

### 1.1 MOTIVAÇÃO

Sistemas de visão computacional buscam recuperar informações úteis do mundo real a partir de imagens, criando um modelo para tal. Neste contexto, imagens de profundidade (*range images*), que possuem a vantagem de dar a informação 3D diretamente, vêm sendo bastante utilizadas nos últimos anos (BESL e JAIN, 1988).

Computacionalmente, a interpretação direta de imagens, representadas por grandes quantidades de valores organizados na forma de matrizes, é difícil. Portanto, a segmentação de imagens constitui um importante passo para a extração de inferências significativas de dados visuais para posterior interpretação da imagem (BESL e JAIN, 1988).

O processo de segmentação de uma imagem refere-se ao particionamento da mesma em segmentos que correspondam a regiões homogêneas, segundo um predicado definido, existindo uma correspondência entre regiões e objetos ou partes de objetos na cena. (COLEMAN e ANDREWS, 1979; JAIN, CASTURI e SCHUNCK, 1995; BELLON, 1997).

As técnicas de segmentação para imagens de profundidade podem ser classificadas em duas categorias: baseadas em região e baseadas em borda. Abordagens baseadas em região buscam agrupar *pixels* em regiões de acordo com a similaridade das propriedades de superfície. Por outro lado, abordagens baseadas em borda buscam detectar mudanças abruptas de orientação e profundidade na superfície (JAIN, CASTURI e SCHUNCK, 1995).

Três técnicas baseadas em região bastante usadas são: crescimento de regiões (*region growing*) (BESL e JAIN, 1988), divisão-e-união (*split-and-merge*) (TAYLOR, SAVINI e REEVES, 1989) e agrupamento (*clustering*) (COLEMAN e ANDREWS, 1979). As técnicas de crescimento de regiões e divisão-e-união apresentam a necessidade de se definir valores de limiar rígidos, cuja obtenção automática é difícil. Já as técnicas de agrupamento são menos dependentes de limiares e, sob este aspecto, isto constitui uma vantagem (BELLON, 1997).

Na segmentação de imagens por agrupamento, os elementos da imagem (*pixels*) dentro de um grupo apresentam uma maior similaridade entre si. Em termos de imagem, cada *pixel* é representado por um vetor de características. Uma imagem de profundidade possui naturalmente três características que são as coordenadas ( $x, y, z$ ) de cada *pixel*. Entretanto, outras características podem ser extraídas a partir destas, como valores das normais, valores de

curvatura, entre outros. Logo, a seleção do conjunto característico representa um passo essencial na segmentação por agrupamento, ficando sujeita ao tipo de aplicação (COLEMAN e ANDREWS, 1979; KRISHNAPURAM e MUNSHI, 1991).

Por outro lado, a segmentação de imagens por agrupamento apresenta certas restrições, sendo necessária a sua associação a outras técnicas. Em BELLON (1997), propõe-se a combinação das técnicas de agrupamento e de detecção de bordas para segmentação/reconstrução de imagens de profundidade, permitindo uma estimativa coerente sobre o número de regiões na imagem e uma melhor conectividade entre as regiões.

Num contexto mais amplo de agrupamento, a análise de grupo (*cluster analysis*) constitui o estudo formal de métodos de agrupamento, ou classificação de objetos, com aplicações nas mais diversas áreas, como biologia, medicina, psicologia, além de outras, incluindo visão computacional (DUBES e JAIN, 1988).

Métodos de agrupamento (*clustering*) têm como objetivo a classificação de amostras ou objetos em um espaço multidimensional de acordo com suas características intrínsecas, resultando em classes ou grupos bem definidos (DUDA e HART, 1973; DUBES e JAIN, 1988; EVERITT, 1993). Conseqüentemente, tais métodos vêm sendo freqüentemente utilizados para a segmentação de imagens, com o objetivo de agrupar elementos da imagem que apresentam características similares (COLEMAN e ANDREWS, 1979; HOFFMAN e JAIN, 1987; BENSaid et al., 1996; SABATA, ARMAN e AGGARWAL, 1993; BARONTI et al., 1990; GROSKI e JAIN, 1986).

DUBES e JAIN (1976, 1980, 1988) apresentam estudos sobre métodos de agrupamento. Segundo eles, os vários métodos utilizados para agrupar dados estão baseados em diferentes critérios e diferentes metodologias. Em geral, estes métodos podem ser reunidos em duas categorias principais: (1) métodos particionais e, (2) métodos hierárquicos (incluindo

grafos teóricos). Por questão de complexidade computacional, métodos de agrupamento particionais são mais adequados a grandes conjuntos de dados e, portanto, mais largamente utilizados, especialmente em segmentação de imagens. Entretanto, um método nem sempre é genérico o suficiente para classificar qualquer conjunto de dados.

Os trabalhos de DUBES e JAIN (1976, 1980, 1988) dão uma boa base para usuários potenciais de métodos de agrupamento. Muitos outros trabalhos são encontrados na literatura, sempre com o objetivo de tentar resolver um ou mais problemas comuns a um determinado método de agrupamento (ZAHN, 1971; HUBERT, 1974; COLEMAN e ANDREWS, 1979; GROSKI e JAIN, 1986; HOFFMAN e JAIN, 1987; BARONTI et al., 1990; KRISHNAPURAM e MUNSHI, 1991; KURITA, 1991; SELIM e ALSULTAN, 1991; ZHANG e BOYLE, 1991; VENKATESWARLU e RAJU, 1992; HATTORI e TORII, 1993; SABATA, ARMAN e AGGARWAL, 1993; BABU e MURTY, 1994; KAMEL e SELIM, 1994; TSAO, BEZDEK e PAL, 1994; AL-SULTAN, 1995; BENSALD et al., 1996). A idéia comum é introduzir heurísticas para: (1) reduzir o esforço computacional, (2) determinar o número de grupos adequado ou, (3) para encontrar uma boa partição para o agrupamento.

Embora os métodos de agrupamento de um modo geral sejam bastante populares, não há na literatura trabalho mais recente e, ao mesmo tempo, aplicado à segmentação de imagens de profundidade, que reúna e descreva os diferentes métodos existentes, classificando-os e comparando-os sob diferentes aspectos.

## 1.2 OBJETIVOS

Este trabalho tem como objetivo apresentar um estudo comparativo sobre métodos de agrupamento, incluindo diversos algoritmos que realizam agrupamento. Este estudo também compreende diferentes abordagens para segmentação de conjuntos de dados, que associam métodos de agrupamento à outras regras ou métodos, para um melhor desempenho dos mesmos.

Outro objetivo deste trabalho é apresentar os resultados da segmentação de imagens de profundidade, a partir das implementações de diferentes algoritmos de agrupamento. Para isto, faz-se necessário a seleção das imagens para teste e, em consequência, a seleção do conjunto característico adequado.

Objetiva-se também, apresentar uma nova abordagem para melhoramento do desempenho de algoritmos de agrupamento na segmentação de imagens de profundidade. A abordagem desenvolvida é baseada na extração de informações, para inicialização dos algoritmos de agrupamento, a partir de uma pré-segmentação da imagem usando detecção de bordas.

## 1.3 CONTEÚDO

Este trabalho é organizado da seguinte maneira. O Capítulo 2 consiste de uma descrição teórica dos métodos e algoritmos de agrupamento. O Capítulo 3 discute diferentes abordagens para segmentação de imagens que aplicam esses métodos, e introduz diagramas de classificação dessas abordagens. O Capítulo 4 mostra resultados experimentais obtidos pela

implementação e comparação de diferentes algoritmos de agrupamento, aplicados à segmentação de imagens de profundidade, bem como resultados decorrentes da implementação da nova abordagem desenvolvida. Em seguida, no Capítulo 5, apresenta-se a conclusão do trabalho, seguida das referências bibliográficas.

## 2 ALGORITMOS E MÉTODOS DE AGRUPAMENTO

Segundo DUDA e HART (1973), o objetivo dos métodos de agrupamento é produzir uma descrição em termos de grupos de amostras, os quais possuem similaridades internas fortes, em um espaço  $n$ -dimensional. A similaridade entre amostras pode ser medida de várias maneiras, dependendo do tipo de dados envolvidos, como pode ser visto em EVERITT (1993). Outro ponto importante é o conjunto de características utilizado para descrever cada elemento do conjunto, o qual deve ser relevante para o tipo de dados e também para o tipo de aplicação sendo tratada.

Métodos de agrupamento hierárquicos (DUBES e JAIN, 1976, 1980, 1988) e grafo-teóricos geram uma seqüência aninhada de partições e, geralmente, requerem somente a matriz de proximidades entre os objetos. Métodos de agrupamento particionais, como do tipo minimização de custo, geram uma única partição dos dados e requerem os dados na forma de uma matriz de amostras.

Devido às características dos métodos, cada qual tem, geralmente, seu próprio domínio de aplicação. De acordo com DUBES e JAIN (1976, 1980, 1988), métodos hierárquicos (incluindo grafo-teóricos) são mais populares em áreas nas quais é necessário construir taxonomias, como biologia, ciências sociais e do comportamento. Já os métodos particionais

são utilizados quando uma única partição é importante; devido à sua simplicidade e eficiência computacional, são adequados a grandes conjuntos de dados, sendo largamente utilizados, especialmente em segmentação de imagens.

DUBES e JAIN (1976, 1980, 1988) também fazem a distinção entre métodos e algoritmos de agrupamento. Um método de agrupamento é uma estratégia geral para a formação de grupos. Já um algoritmo é a implementação de uma estratégia que incorpora várias heurísticas. Neste capítulo, as bases teóricas dos métodos de agrupamento são descritas, bem como algoritmos que incorporam os métodos.

## 2.1 MÉTODOS PARTICIONAIS

Dadas  $n$  amostras em um espaço  $d$ -dimensional, o problema do agrupamento particional consiste em determinar uma partição das amostras em  $K$  grupos, de forma que amostras em um grupo são mais similares entre si do que em relação a amostras em grupos diferentes (DUDA e HART, 1973; DUBES e JAIN, 1980, 1988). Uma função critério, tal como a soma do erro-quadrático, é adotada para medir a qualidade de agrupamento, de uma partição dos dados. Assim, o problema consiste em encontrar a partição que extremiza a função critério.

## 2.1.1 Funções Critério

### 2.1.1.1 Soma de Erro-quadrático

A função critério mais simples e mais freqüentemente usada para agrupamento particional é o critério soma de erro-quadrático (DUBES e JAIN, 1976, 1980, 1988). Consideremos uma matriz de  $n$  amostras e  $d$  dimensões (características) a ser particionada em  $K$  grupos  $\{C_1, C_2, \dots, C_K\}$ , tal que o grupo  $C_K$  tem  $n_K$  amostras e cada amostra está em um único grupo:

$$\sum_{k=1}^K n_k = n. \quad (1)$$

O vetor médio ou centro do grupo  $C_K$  é definido como o centróide do grupo, ou:

$$\mathbf{m}^{(k)} = (1/n_k) \sum_{\mathbf{x} \in X_i} \mathbf{x}_i^{(k)}, \quad (2)$$

onde  $\mathbf{x}_i^{(k)}$  é a  $i$ -ésima amostra pertencente ao grupo  $C_K$ . O erro-quadrático para o grupo  $C_K$  é a soma das distâncias Euclidianas quadráticas entre cada amostra em  $C_K$  e seu centro de grupo  $\mathbf{m}^{(k)}$ :

$$e_k^2 = \sum_{i=1}^{n_k} (\mathbf{x}_i^{(k)} - \mathbf{m}^{(k)})^T (\mathbf{x}_i^{(k)} - \mathbf{m}^{(k)}). \quad (3)$$



O erro-quadrático para o agrupamento total contendo  $K$  grupos é a soma dos erros-quadráticos dos grupos:

$$E_K^2 = \sum_{k=1}^K e_k^2. \quad (4)$$

O objetivo de um método de agrupamento com função critério soma do erro-quadrático é encontrar uma partição contendo  $K$  grupos que minimize  $E_K^2$ , onde este erro representa os desvios das amostras em relação aos centróides. Em outras palavras, as amostras são vistas como uma coleção de  $K$  nuvens de forma esférica, e o uso do critério soma do erro-quadrático tenta fazer as  $K$  nuvens tão compactas internamente e bem separadas entre si quanto possível (DUBES e JAIN, 1988).

Desta forma, o critério soma de erro-quadrático é bem adequado para problemas nos quais os grupos têm essa forma. Em problemas nos quais ocorrem grandes diferenças no número de amostras em grupos distintos, é possível que uma partição que divide um grupo grande é favorecida sobre uma que mantém a integridade dos grupos simplesmente porque a redução pequena no erro-quadrático encontrado é multiplicada por muitos termos na soma. Esta situação pode ocorrer com frequência devido à presença de pontos fora de contexto (*outliers*) ou ruído, prejudicando a interpretação e a avaliação dos resultados do agrupamento. Assim, a seleção de uma função critério adequada deve ser considerada (DUDA e HART, 1973).

O uso da distância Euclidiana supõe que os valores característicos não são correlacionados entre si. Em muitas aplicações, esta suposição não é justificada, e a distância Euclidiana pode ser uma medida inadequada. Uma alternativa é o uso da distância Mahalanobis

quadrática (ver Apêndice 1), que normaliza os valores característicos (DUBES e JAIN, 1980, 1988).

### 2.1.1.2 Caso *Hard*

Outra forma de definir o problema do agrupamento em termos de minimização da soma das distâncias quadráticas intra-grupo é dada a seguir (DUBES e JAIN, 1988). Seja  $x_{ij}$  a  $j$ -ésima característica da  $i$ -ésima amostra,  $i=1, \dots, n$ ;  $j=1, \dots, d$ . Seja  $y_{ik}=1$  se a  $i$ -ésima amostra pertence ao  $k$ -ésimo grupo, e  $y_{ik}=0$  se a  $i$ -ésima amostra não pertence ao  $k$ -ésimo grupo,  $k=1, \dots, K$ . O centróide do  $k$ -ésimo grupo,  $\mathbf{z}_k$ , é  $\mathbf{z}_k = (z_{k1}, \dots, z_{kd})$ , onde:

$$z_{kj} = \frac{\sum_{i=1}^n (y_{ik} x_{ij})}{\sum_{i=1}^n (y_{ik})}. \quad (5)$$

Uma amostra pertence a um determinado grupo quando tem a menor distância quadrática para o respectivo centróide. A variação intra-grupo total  $S_T$  é:

$$S_T = \sum_{i=1}^n \sum_{k=1}^K y_{ik} \sum_{j=1}^d (x_{ij} - z_{kj})^2. \quad (6)$$

Assim, dada uma matriz  $X$  de  $n$  amostras, e o número de grupos  $K$ , o problema consiste em encontrar a matriz  $Y$ ,  $K \times n$ , que minimiza  $S_T$ , com:

$$\sum_{k=1}^K y_{ik} = 1 \quad \text{e} \quad \sum_{i=1}^n y_{ik} > 0, \quad (7) \text{ e } (8)$$

onde o termo  $y_{ik}$  denota o relacionamento da  $i$ -ésima amostra com o  $k$ -ésimo grupo. A matriz  $Y$  é composta de 0's e 1's, com exatamente um 1 em cada coluna e ao menos um 1 em cada linha.

Este conceito de agrupamento é conhecido como agrupamento rígido ou *hard*, semelhante ao agrupamento *fuzzy*, que é visto mais adiante (DUBES e JAIN, 1988; HATHAWAY e BEZDEK, 1994; TSAO, BEZDEK e PAL, 1994; BENSAID et al., 1996). Como exemplo, a matriz de relacionamentos seguinte representa o agrupamento das amostras  $\{x_1, \dots, x_5\}$  nos três grupos  $\{x_1, x_5\}$ ,  $\{x_2, x_3\}$  e  $\{x_4\}$ .

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

### 2.1.1.3 Traço

Um critério de agrupamento erro-quadrático pode ser obtido da decomposição da matriz esparsa  $S$  nas matrizes esparsas  $S_W$  (matriz esparsa *within* intra-grupo) e  $S_B$  (matriz esparsa *between* inter-grupo) (ver Apêndice 2) (DUDA e HART, 1973; DUBES e JAIN, 1988):

$$S_T = S_W + S_B. \quad (9)$$

A matriz  $S$  é fixa, isto é, não depende de como as amostras estão particionadas em grupos, dependendo somente do conjunto total de amostras. Entretanto, as matrizes intra-grupo e inter-grupo dependem do particionamento. Na partição ideal, a matriz intra-grupo deve ser mínima, e a matriz inter-grupo a máxima. Assim, um critério de agrupamento é obtido a partir do traço das matrizes intra e inter-grupo, sendo que o traço representa uma medida escalar das mesmas:

$$\text{tr}(S_W) = \sum_{k=1}^K \text{tr}(S^{(k)}) = \sum_{k=1}^K \sum_{i=1}^{n_k} (\mathbf{x}_i^{(k)} - \mathbf{m}^{(k)})^T (\mathbf{x}_i^{(k)} - \mathbf{m}^{(k)}) = E_K^2, \quad (10)$$

o que é idêntico ao critério erro-quadrático.

O  $\text{tr}(S^{(k)})$  é a soma das variâncias ao longo das direções características para o grupo  $k$  e mede a compactação do grupo  $C_k$ . Minimizar  $\text{tr}(S_W)$  é correspondente a maximizar  $\text{tr}(S_B)$  porque:

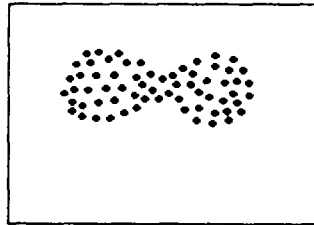
$$\text{tr}(S) = \text{tr}(S_W) + \text{tr}(S_B) \quad (11)$$

e  $\text{tr}(S)$  é o mesmo para qualquer partição.

#### 2.1.1.4 Caso *Fuzzy*

Todos os critérios erro-quadráticos vistos até aqui levam ao particionamento de um conjunto de amostras em grupos disjuntos, o que é bem adequado quando os grupos são compactos e bem separados, não havendo incerteza quanto à atribuição de uma amostra a um grupo (DUBES e JAIN, 1980, 1988). Portanto, apesar de úteis para muitos problemas, não são universalmente aplicáveis.

FIGURA 1 - GRUPOS PRÓXIMOS  
COM ALGUNS PONTOS INDEFINIDOS



Quando os grupos são próximos ou há sobreposição de grupos (Figura 1), é possível definir uma função critério para o agrupamento *fuzzy*, de forma que uma amostra pertence a um grupo com um certo grau de pertinência ou relacionamento. Uma função critério para agrupamento *fuzzy* é dada a seguir.

Seja  $x_{ij}$  a  $j$ -ésima característica da  $i$ -ésima amostra,  $i=1, \dots, n; j=1, \dots, d$ . O centróide do  $k$ -ésimo grupo,  $\mathbf{z}_k$ , é  $\mathbf{z}_k = (z_{k1}, \dots, z_{kd})$ , onde:

$$z_{kj} = \frac{\sum_{i=1}^n (y_{ik}^m x_{ij})}{\sum_{i=1}^n (y_{ik}^m)}. \quad (12)$$

Formalmente, a função critério correspondente é:

$$S_T = \sum_{i=1}^n \sum_{k=1}^K y_{ik}^m \sum_{j=1}^d (x_{ij} - z_{kj})^2, \quad (13)$$

onde  $m \geq 1$  é o expoente peso para relacionamento *fuzzy*.

Logo, dada uma matriz  $X$  de amostras, e o número de grupos  $K$ , o problema consiste em encontrar a matriz  $Y$ ,  $n \times K$ , que minimiza  $S_T$ . No agrupamento *fuzzy*, o grau de relacionamento corresponde a um valor no intervalo  $[0,1]$ , sendo válidas as condições dadas pelas Equações (7) e (8).

O grau de pertinência da  $i$ -ésima amostra para o  $k$ -ésimo grupo, nesse caso, é dado por:

$$y_{ik} = 1 / \sum_{j=1}^K \left\{ \left[ \sum_{l=1}^d (x_{il} - z_{kl}) \right] / \left[ \sum_{l=1}^d (x_{il} - z_{jl}) \right] \right\}^{2(m-1)}. \quad (14)$$

Como exemplo, a matriz de relacionamentos seguinte é uma partição *fuzzy* do conjunto de amostras  $\{x_1, \dots, x_5\}$ , pela qual a amostra  $x_1$  é fortemente relacionada ao grupo 1, assim como a amostra  $x_3$  é fortemente relacionada ao grupo 2. Já a amostra  $x_4$  é moderadamente relacionada aos grupos 2 e 3, e a amostra  $x_5$  é inequivocamente relacionada ao grupo 1.

$$Y = \begin{bmatrix} 0,95 & 0,11 & 0,05 & 0,02 & 1 \\ 0,03 & 0,78 & 0,81 & 0,47 & 0 \\ 0,02 & 0,11 & 0,14 & 0,51 & 0 \end{bmatrix}$$

Portanto, numa partição *fuzzy*, o grau de relacionamento é relaxado para um valor no intervalo  $[0,1]$ . Quando o expoente  $m \rightarrow 1$ , o grau de relacionamento  $y_{ik} \rightarrow 0$  ou 1. Assim, quando  $m=1$ , tem-se um agrupamento rígido (*hard*), idêntico ao caso anteriormente descrito, cuja função critério é definida pela Eq. (6). Fica claro, então, que uma partição *hard* é um caso particular de partição *fuzzy*, mas não o contrário.

O critério para agrupamento *fuzzy* pode ser considerado do tipo erro-quadrático, levando-se em conta que a saída de um agrupamento *fuzzy* não somente inclui uma partição dos dados como também informação adicional na forma de valores de relacionamento. A principal vantagem desta abordagem (*fuzzy*) é o fato de se obter mais informação sobre a estrutura dos grupos, mostrando o grau pelo qual cada amostra se ajusta a um determinado grupo, permitindo a diferenciação entre amostras que se ajustam completamente a um dos grupos daquelas que são ambíguas com relação a certos grupos (HATHAWAY e BEZDEK, 1994). Entretanto, da mesma forma que os demais critérios abordados anteriormente, o critério para agrupamento *fuzzy* leva a uma partição na qual os grupos se apresentam com populações mais ou menos equalizadas, não resolvendo adequadamente situações nas quais os grupos apresentam diferenças significativas quanto ao número de amostras.

### 2.1.2 Algoritmos Básicos

Nesta seção, apresentam-se os algoritmos *K-means*, ISODATA, CLUSTER e *fuzzy K-means*. Todos eles correspondem a diferentes implementações de métodos baseados na minimização de uma função critério.

### 2.1.2.1 Algoritmo *K-means*

O algoritmo *K-means* é um dos algoritmos de agrupamento particional mais utilizados, e pode ser definido da seguinte forma (DUBES e JAIN, 1980, 1988; ZHANG e BOYLE, 1991; VENKATESWARLU e RAJU, 1992):

- Passo 1. Estabelece os centros de grupo iniciais, arbitrariamente;
- Passo 2. Distribui os elementos entre os grupos utilizando a distância Euclidiana quadrática mínima, isto é, atribuindo um elemento ao grupo cujo centro é mais próximo;
- Passo 3. Calcula centros de grupo dos grupos existentes, pela média de todos os componentes de cada grupo;
- Passo 4. Os passos 2 e 3 devem se repetir até que não haja mais trocas nos centros de grupos ou que se atinja o número máximo de iterações definido.

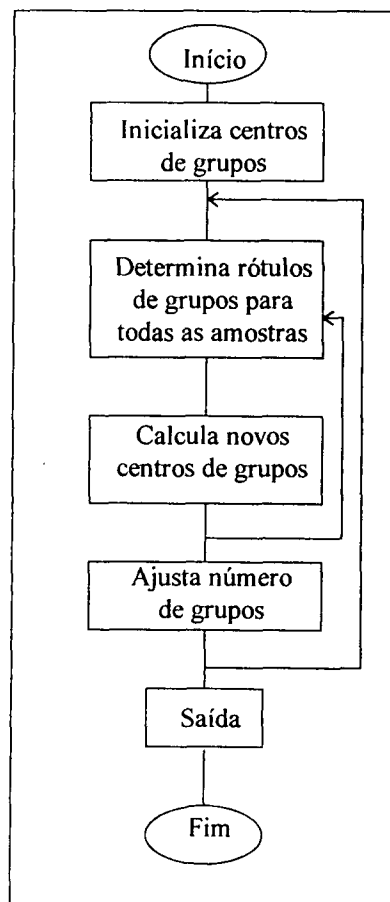
Dubes e Jain apresentam um fluxograma simplificado (Figura 2) para os algoritmos *K-means* e ISODATA, sendo que o algoritmo *K-means* não faz o ajuste do número de grupos, limitando-se ao laço interno do fluxograma (DUBES e JAIN, 1976).

Uma vantagem deste algoritmo em relação a outros é a pouca dependência de limiares fornecidos pelo usuário, sendo necessários apenas o número de grupos, que é fixo, e o número de iterações e/ou outras variáveis que possam estabelecer a parada do algoritmo. Porém, uma boa estimativa para o número de grupos nem sempre ocorre. Além disso, diferentes inicializações podem levar a diferentes partições finais, porque algoritmos baseados na minimização do erro-quadrático convergem para um mínimo local, o que pode levar a uma solução inadequada para o agrupamento. SELIM e ISMAIL (1984) mostram que o algoritmo *K-means* converge em um número finito de iterações, e apresentam condições sob as quais a solução obtida é um mínimo local.



Desde que o conjunto de amostras seja finito, há um número finito de partições possíveis. Assim, na teoria, o problema do agrupamento pode ser resolvido sempre por enumeração exaustiva. Entretanto, na prática é inviável, pois há aproximadamente  $c^n/c!$  maneiras de particionar um conjunto de  $n$  elementos em  $c$  grupos. Como exemplo, uma busca exaustiva para o melhor conjunto de 5 grupos em 100 elementos requereria mais que  $10^{67}$  particionamentos (DUDA e HART, 1973).

FIGURA 2 - FLUXOGRAMA SIMPLIFICADO DOS ALGORITMOS *K-MEANS* E ISODATA



FONTE: DUBES e JAIN (1976)

Uma maneira de superar esta desvantagem é executar o algoritmo diversas vezes com diferentes inicializações, para aumentar as chances de obtenção do mínimo global (DUBES e JAIN, 1980, 1988). Outra abordagem freqüentemente usada na busca por partições ótimas é a otimização iterativa, na qual a idéia básica é encontrar alguma partição inicial razoável e mover amostras de um grupo para outro se o movimento melhora o valor da função critério (DUDA e HART, 1973; ZHANG e BOYLE, 1991). Da mesma forma, tal abordagem garante otimização local mas não global, embora aumente as chances de se encontrar o mínimo global.

#### 2.1.2.2 Algoritmo ISODATA

O algoritmo ISODATA é o mais famoso dos algoritmos de agrupamento particionais e é obtido pela adição dos seguintes passos no algoritmo *K-means* (DUDA e HART, 1973; DUBES E JAIN, 1976, 1980, 1988):

Passo 5. Remove grupos contendo menos que  $N$  amostras, e trata as amostras envolvidas como fora de contexto (ruído);

Passo 6. Une e/ou divide grupos restantes. Se não é necessário união ou divisão, pára;

Passo 7. Calcula novos centros de grupos e repete passos 2 a 6.

O ISODATA utiliza heurísticas para criar novos grupos enquanto tenta encontrar o número de grupos requisitado pelo usuário. A capacidade de divisão e união de grupos existentes permite ao algoritmo realizar uma melhor seleção dentre uma variedade de agrupamentos.

Dois parâmetros,  $T_{11}$  e  $T_{12}$ , fornecidos pelo usuário controlam a divisão e união de grupos. O primeiro controla a admissão para divisão e união da seguinte maneira: define  $K'_1$  como a parte inteira de  $(1 + T_{11}) K_1$ , onde  $K_1$  é o número de grupos desejado e fornecido pelo usuário. Se o número corrente de grupos excede  $K'_1$ , o subprograma de união é ativado. Se não, o subprograma de divisão é ativado. O segundo parâmetro,  $T_{12}$ , ajusta limiares internos nos subprogramas de união e divisão. O limiar de divisão é  $T_{12}$  vezes o erro médio, enquanto o limiar de união é  $T_{12}$  vezes a distância média entre centros de grupos.

O processo de inicialização no algoritmo ISODATA é mais elaborado: o primeiro dos centros de grupo iniciais é o centróide de todos os elementos. Todos os elementos são testados e aquele com maior distância em relação ao centróide constitui um novo centro de grupo inicial. Tal procedimento dá uma distribuição mais uniforme dos centros de grupos iniciais. Este algoritmo também necessita de parâmetros que devem ser fornecidos pelo usuário. No entanto, tem como vantagem a capacidade de recuperar partições iniciais inadequadas, pela flexibilidade quanto à seleção do número de grupos, que não é necessariamente fixo, e impedir que amostras fora de contexto possam distorcer a forma dos grupos.

A Figura 2 exibe um fluxograma simplificado dos algoritmos *K-means* e ISODATA, no qual o laço externo mostra o ajuste do número de grupos que ocorre no algoritmo ISODATA (DUBES e JAIN, 1976).

O algoritmo ISODATA, da mesma forma que o algoritmo *K-means*, converge para uma solução mínima local. Com relação a este fato, as mesmas considerações feitas para o algoritmo *K-means* valem para o algoritmo ISODATA.

### 2.1.2.3 Algoritmo CLUSTER

Também baseado na minimização de erro-quadrático, o algoritmo CLUSTER (DUBES e JAIN, 1976, 1988) gera uma seqüência não hierárquica dos melhores agrupamentos contendo 1, 2, ...,  $K$  grupos.

Este algoritmo envolve duas fases as quais são repetidas até que um passo através de ambas não diminui o erro-quadrático dos agrupamentos. A primeira fase cria uma seqüência de agrupamentos contendo 2, 3, ...,  $K$  grupos, onde  $K$  é especificado pelo usuário. Os dois centros de grupo iniciais são o centróide de todas as amostras e a amostra mais distante deste centróide, a qual é removida do agrupamento existente e identificada como o terceiro centro de grupo. O algoritmo *K-means* é então executado para a convergência deste novo agrupamento.

A primeira passagem pela fase 1 dá um conjunto de  $K$  agrupamentos. A fase 2 cria, então, outro conjunto de agrupamentos pela união dos grupos existentes, dois de cada vez, para ver se um agrupamento melhor é encontrado. Depois de cada passo pelas fases 1 e 2, os erros-quadráticos dos agrupamentos tendo o mesmo número de grupos são comparados com aqueles agrupamentos anteriores ao passo. Se qualquer um dos erros-quadráticos é menor do que o anterior, outro passo pelas fases 1 e 2 é iniciado. Isto se repete até que os erros-quadráticos não mais diminuam.

Com exceção do número máximo de grupos, o algoritmo CLUSTER não necessita de parâmetros específicos, o que é sempre uma vantagem. A saída deste algoritmo não é apenas uma partição dos dados, mas uma seqüência de partições. Tal como os algoritmos *K-means* e ISODATA, as partições geradas são soluções locais.

Nenhuma heurística é apresentada, dirigida ao algoritmo CLUSTER, para determinar o grupo a ser dividido e o novo centróide para os agrupamentos contendo 3, 4, ...,  $K$  grupos, quando da divisão de grupos na fase 1, bem como para selecionar os dois grupos para união na fase 2. Embora não dirigidas ao algoritmo CLUSTER, as heurísticas para a união e divisão de grupos apresentadas por ZHANG e BOYLE (1991) são bem apropriadas. Para a divisão, é selecionado o grupo com maior erro-quadrático médio. A amostra mais distante do centro do grupo selecionado constitui um novo centróide. Para a união, é determinado o par de grupos mais próximos. O grupo resultante da união destes dois grupos selecionados tem como centróide o vetor obtido pela média ponderada dos centróides dos grupos selecionados.

#### 2.1.2.4 Algoritmo *Fuzzy K-means*

O algoritmo de agrupamento *fuzzy K-means* (FCM) caracteriza-se por atribuir a cada amostra um grau de relacionamento para cada grupo (BABU e MURTY, 1994; HATHAWAY e BEZDEK, 1994; KAMEL e SELIM, 1994; TSAO, BEZDEK e PAL, 1994; BENSALID et al., 1996). A descrição do algoritmo FCM, que utiliza a função critério definida pela Eq. (13), é mostrada a seguir:

Passo 1. Fixa o número de grupos  $K$  e escolhe os centros de grupos iniciais arbitrariamente; estabelece o expoente  $1 < m < \infty$  e o número máximo de iterações;

Passo 2. Calcula a matriz de relacionamentos  $Y$  pela Eq. (14);

$$\text{Se } \sum_{l=1}^d (x_{il} - z_{jd}) = 0, \text{ então } y_{ij}=1 \text{ e } y_{ik}=0, \text{ para } k \neq j. \quad (15)$$

Passo 3. Calcula o vetor de centros  $Z$  pela Eq. (12);

Passo 4. Se o vetor de centros estabiliza ou se atinge o número máximo de iterações, pára, senão volta ao passo 1.

A principal vantagem deste algoritmo é o fato de se obter mais informação sobre a estrutura dos grupos, mostrando o grau pelo qual cada amostra se ajusta a um determinado grupo, o que é útil quando os grupos não se apresentam bem separados. Porém, a partição resultante é uma solução mínima local. A escolha do número adequado de grupos também constitui uma dificuldade para o algoritmo *fuzzy K-means*.

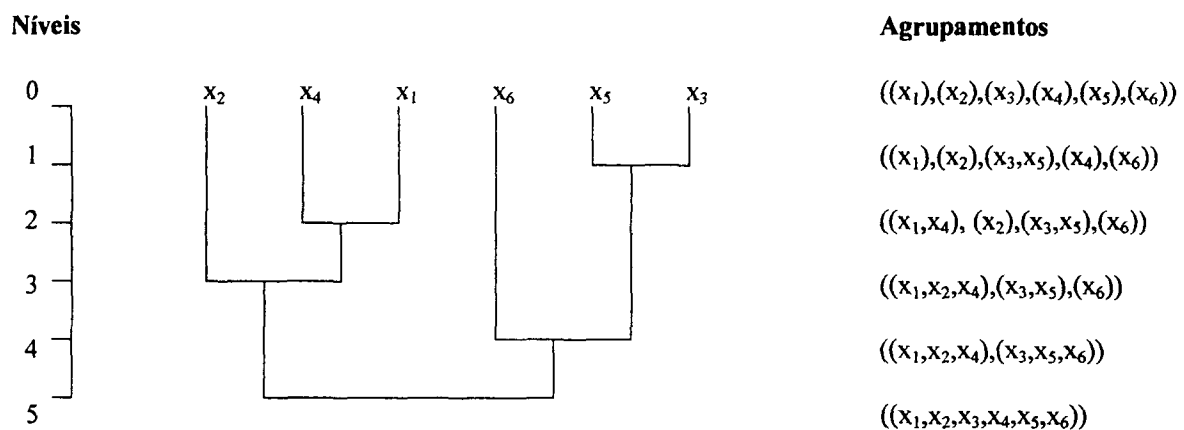
## 2.2 MÉTODOS HIERÁRQUICOS

Um método de agrupamento hierárquico é um procedimento para a criação de uma seqüência aninhada de partições dos dados a partir de uma matriz de proximidades, organizando a matriz de proximidades como um tipo de árvore evolutiva. Numa matriz de proximidades, as linhas e colunas correspondem às amostras e as entradas são as distâncias ou dissimilaridades entre as amostras. Pode-se iniciar com cada amostra constituindo um grupo, e realizar a união de pares de grupos mais próximos em um processo iterativo até que todas as amostras constituam um único grupo. Esta é uma abordagem aglomerativa. Já na abordagem divisiva, todas as amostras começam em um único grupo, que é dividido em dois subgrupos, cada qual dividido novamente, e assim sucessivamente até que cada amostra constitua um grupo (DUDA e HART, 1973; DUBES e JAIN, 1976, 1980, 1988; EVERITT, 1993).

Para cada agrupamento hierárquico corresponde uma árvore, chamada *dendrogram*<sup>1</sup>, que mostra como as amostras estão agrupadas. Cortando horizontalmente um *dendrogram* tem-se uma partição. Quando o *dendrogram* mostra uma escala que registra a ordem pela qual os agrupamentos acontecem, é chamado *dendrogram* de limiar. Quando o *dendrogram* vem acompanhado de uma escala proporcional, que registra o valor de dissimilaridade em cada nível de agrupamento, é chamado *dendrogram* de proximidade (DUBES e JAIN, 1980, 1988).

A Figura 3 mostra um *dendrogram* de limiar para um problema hipotético envolvendo 6 amostras. O nível 1 mostra as 6 amostras como grupos isolados. No nível 2, as amostras  $x_3$  e  $x_5$  são agrupadas formando um grupo, e permanecem juntas em todos os níveis subsequentes. No último nível, tem-se um único grupo formado por todas as amostras.

FIGURA 3 - DENDROGRAM DE LIMIAR



Essa visão gráfica da estrutura hierárquica, através do *dendrogram*, é conveniente e de fácil compreensão, o que é, em muitos problemas, uma vantagem. Na maioria das vezes, entretanto, o investigador não está interessado em completar a hierarquia, mas somente em umas poucas partições obtidas da hierarquia. Desta forma, partições são obtidas cortando-se o

<sup>1</sup> O termo *dendrogram* refere-se a uma representação em forma de árvore.

*dendrogram* ou selecionando algumas das soluções na seqüência aninhada de agrupamentos que compõem a hierarquia. Em muitas aplicações faz-se necessário utilizar alguma heurística para determinar qual a partição que produz o melhor ajuste dos dados, ou que decide qual o número apropriado de grupos (EVERITT, 1993).

Uma desvantagem dos métodos hierárquicos, tanto divisivos quanto aglomerativos, é que uma partição inadequada não pode ser reparada num estágio posterior. Isto porque, nos métodos aglomerativos, amostras unidas num determinado nível permanecem unidas nos níveis subseqüentes. Já nos métodos divisivos, amostras separadas em determinado nível permanecem separadas em níveis posteriores (EVERITT, 1993).

### 2.2.1 Métodos Aglomerativos

Um procedimento de agrupamento hierárquico aglomerativo produz uma série de partições dos dados,  $P_n, P_{n-1}, \dots, P_1$ . A primeira,  $P_n$ , consiste de  $n$  grupos, cada qual contendo um único indivíduo (EVERITT, 1993). A operação básica do método aglomerativo é a seguinte:

Passo 1. Inicialização. Grupos  $C_1, C_2, \dots, C_n$  cada qual contendo um único indivíduo.

Passo 2. Encontra o par de grupos distintos mais próximos,  $C_i$  e  $C_j$ , une  $C_i$  e  $C_j$ , elimina  $C_j$  e decrementa o número de grupos de 1. Se o número de grupos é igual a 1, pára, senão volta ao passo 2.

Em cada estágio, indivíduos ou grupos, os quais estão mais próximos, são unidos. Há diferentes maneiras de definir distância entre um indivíduo e um grupo contendo vários



indivíduos, ou entre dois grupos de indivíduos. Entre outras, as seguintes medidas podem ser usadas para medir a distância entre dois grupos (DUDA e HART, 1973):

$$d_{\min}(X_i, Y_j) = \min_{x \in X_i, x' \in Y_j} \|x - x'\| \quad (16)$$

$$d_{\max}(X_i, Y_j) = \max_{x \in X_i, x' \in Y_j} \|x - x'\| \quad (17)$$

$$d_{\text{avg}}(X_i, Y_j) = (1/n_i n_j) \sum_{x \in X_i} \sum_{x' \in Y_j} \|x - x'\| \quad (18)$$

$$d_{\text{mean}}(X_i, Y_j) = \|m_i - m_j\| \quad (19)$$

Todas estas medidas normalmente produzem os mesmos resultados quando os grupos são compactos e bem separados. Entretanto, se os grupos são próximos uns dos outros ou se suas formas não são basicamente hiperesféricas, resultados diferentes podem ser obtidos.

Bastante conhecido, o método de agrupamento hierárquico aglomerativo *single-link* tem como característica determinante a distância ( $d_{\min}$ ), isto é, a distância entre grupos é definida pelo par de indivíduos mais próximos, onde somente pares consistindo de um indivíduo de cada grupo são considerados. Também popular, o método de agrupamento hierárquico aglomerativo *complete-link* é o oposto ao *single-link*, pois a distância entre grupos é definida como o par mais distante ( $d_{\max}$ ) de indivíduos, um de cada grupo (DUBES e JAIN, 1988; EVERITT, 1993).

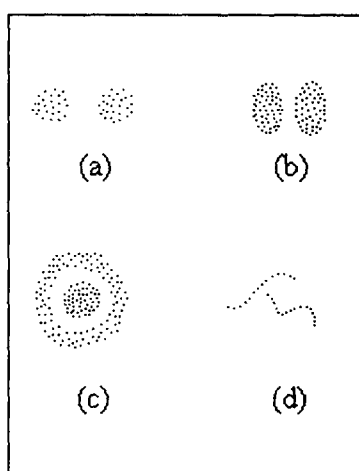
As medidas de distância máxima e mínima representam dois extremos, tendendo a ser sensíveis a ruído. Portanto, podem não apresentar bons resultados quando os grupos são alongados, ou apresentam formas e tamanhos diferentes (Figura 4 (b), (c), (d)). O método

*single-link* dá bons resultados quando os grupos são compactos e bem separados (Figura 4 (a)).

Da mesma forma, o método *complete-link* leva a bons resultados quando os grupos são compactos e de tamanho mais ou menos equalizado, devido à sua tendência de impor uma estrutura deste tipo. Caso contrário, os resultados podem não corresponder à estrutura de grupo em estudo (DUDA e HART, 1973).

DUDA e HART (1973) sugerem substituir o critério “encontrar o par de grupos distintos mais próximos” do método hierárquico aglomerativo básico por “encontrar o par de grupos distintos cuja união melhora uma função critério tanto quanto possível”. Um exemplo de função critério que pode ser utilizada é a soma do erro-quadrático. Desta forma, fica assegurada uma melhor escolha de grupos a cada iteração.

FIGURA 4 - (A) GRUPOS BEM SEPARADOS, DE FORMA ESFÉRICA, (B) GRUPOS DE FORMA UM POUCO ALONGADA, (C) GRUPOS INCIDENTES, (D) GRUPOS EM FORMA DE LINHA



FONTE: DUBES e JAIN (1980)

A seguir são descritos algoritmos para os métodos *single-link* e *complete-link*, de acordo com JOHNSON (1967).

### 2.2.1.1 Algoritmos *Single-link* e *Complete-link*

Partindo da matriz de dissimilaridades  $D=[d(i,j)]$ , este algoritmo produz uma seqüência aninhada de agrupamentos e um valor ou nível para cada agrupamento (JOHNSON, 1967; DUBES e JAIN, 1980, 1988).

Passo 1. Inicialização. Atribui cada amostra a um único grupo, obtendo-se os grupos  $C_0=(1), (2), \dots, (n)$ . O nível inicial é  $L_0=0$ . O índice  $h$  é o número do agrupamento. Faz  $h=0$ .

Passo 2. Encontra o menor valor em  $D[d(i,j)]$ , que corresponde ao par de grupos mais próximos, de acordo com:

$$d[(i),(j)]=\min\{d[(i),(j)]\},$$

considerando todos os pares de grupos no agrupamento corrente.

Passo 3. Faz  $h=h+1$ . Une os grupos  $(i)$  e  $(j)$  em um novo grupo, estabelecendo o agrupamento  $C_h$  com  $n-h$  grupos. Faz  $L_h=d[(i),(j)]$ .

Passo 4. Atualiza a matriz de proximidades  $D$ , suprimindo a linha e a coluna correspondentes aos grupos  $(i)$  e  $(j)$ , e adicionando uma linha e coluna correspondentes ao novo grupo formado. A proximidade entre o novo grupo, denotado por  $(i,j)$ , e grupo antigo  $(k)$ , para o algoritmo *single-link*, é:

$$d[(k),(i,j)]=\min\{d[(k),(i)],d[(k),(j)]\}.$$

Para o método *complete-link*, é:

$$d[(k),(i,j)]=\max\{d[(k),(i)],d[(k),(j)]\}.$$

Passo 5. Se  $h=n-1$ , ou todas as amostras estão em um só grupo, pára. Senão, volta ao passo 2.

O exemplo seguinte ilustra o algoritmo descrito acima (DUBES e JAIN, 1988). Seja a matriz de distâncias:

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
 1 & \left[ \begin{array}{ccccc}
 0,0 & 2,3 & 3,4 & 1,2 & 3,7 \\
 & 0,0 & 2,6 & 1,8 & 4,6 \\
 & & 0,0 & 4,2 & 0,7 \\
 & & & 0,0 & 4,4 \\
 & & & & 0,0
 \end{array} \right]
 \end{array}$$

Como a menor distância na matriz é entre os grupos 3 e 5, eles são unidos para formar um novo grupo. As novas matrizes de distâncias obtidas para os algoritmos *single-link* e *complete-link*, respectivamente, são:

$$d_{1,(3,5)} = \min [d_{1,3}, d_{1,5}] = d_{1,3} = 3,4, \text{ e } d_{1,(3,5)} = \max [d_{1,3}, d_{1,5}] = d_{1,5} = 3,7,$$

$$d_{2,(3,5)} = \min [d_{2,3}, d_{2,5}] = d_{2,3} = 2,6, \text{ e } d_{2,(3,5)} = \max [d_{2,3}, d_{2,5}] = d_{2,5} = 4,6,$$

$$d_{4,(3,5)} = \min [d_{3,4}, d_{4,5}] = d_{3,4} = 4,2, \text{ e } d_{4,(3,5)} = \max [d_{3,4}, d_{4,5}] = d_{4,5} = 4,4.$$

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{ccccc}
 & 1 & 2 & 3,5 & 4 \\
 1 & \left[ \begin{array}{cccc}
 0,0 & 2,3 & 3,4 & 1,2 \\
 & 0,0 & 2,6 & 1,8 \\
 & & 0,0 & 4,2 \\
 & & & 0,0
 \end{array} \right]
 \end{array}$$

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{ccccc}
 & 1 & 2 & 3,5 & 4 \\
 1 & \left[ \begin{array}{cccc}
 0,0 & 2,3 & 3,7 & 1,2 \\
 & 0,0 & 4,6 & 1,8 \\
 & & 0,0 & 4,4 \\
 & & & 0,0
 \end{array} \right]
 \end{array}$$

A menor distância em ambas as matrizes acima é entre os grupos 1 e 4, os quais irão constituir um novo grupo. As novas matrizes de distâncias obtidas são:

$$d_{(1,4),2} = \min [d_{1,2}, d_{2,4}] = d_{2,4} = 1,8, \text{ e } d_{(1,4),2} = \max [d_{1,2}, d_{2,4}] = d_{1,2} = 2,3,$$

$$d_{(1,4),(3,5)} = \min [d_{1,3}, d_{1,5}, d_{3,4}, d_{4,5}] = d_{1,3} = 3,4, \text{ e}$$

$$d_{(1,4),(3,5)} = \max [d_{1,3}, d_{1,5}, d_{3,4}, d_{4,5}] = d_{4,5} = 4,4.$$

$$\begin{array}{c} \begin{array}{ccc} & 1,4 & 2 & 3,5 \\ 1,4 & \left[ \begin{array}{ccc} 0,0 & 1,8 & 3,4 \\ & 0,0 & 2,6 \\ & & 0,0 \end{array} \right] \\ 2 & & & \\ 3,5 & & & \end{array} & \begin{array}{ccc} & 1,4 & 2 & 3,5 \\ 1,4 & \left[ \begin{array}{ccc} 0,0 & 2,3 & 4,4 \\ & 0,0 & 4,6 \\ & & 0,0 \end{array} \right] \\ 2 & & & \\ 3,5 & & & \end{array} \end{array}$$

Para o algoritmo *single-link*, a menor distância agora é entre os grupos 2 e (1,4), que são unidos; para o algoritmo *complete-link*, a menor distância ocorre também entre os grupos 2 e (1,4), os quais são unidos. Porém, note-se que os valores de menor distância são diferentes, o que é possível de se ver no *dendrogram* de proximidade.

As matrizes seguintes mostram o próximo nível de agrupamento, para os algoritmos *single-link* e *complete-link*, respectivamente.

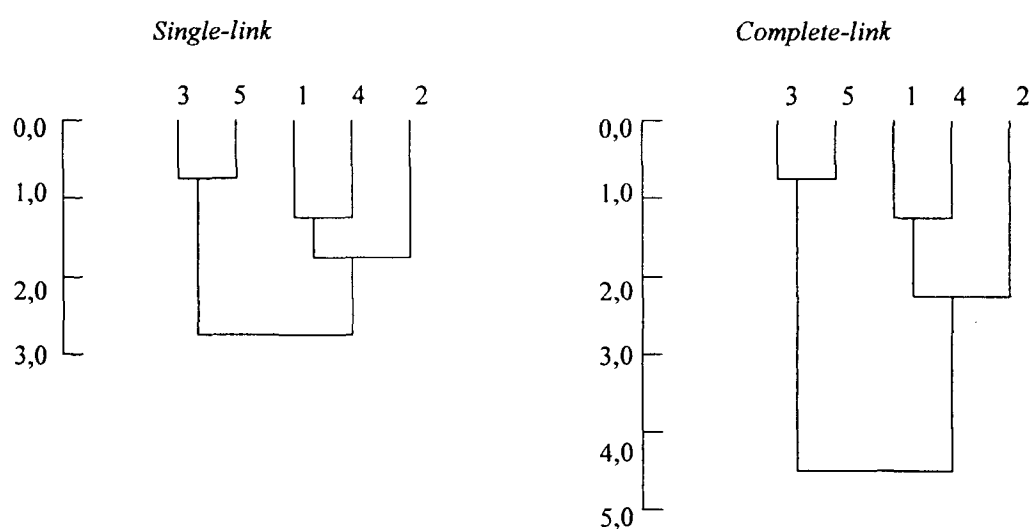
$$d_{(1,2,4),(3,5)} = \min [d_{1,3}, d_{1,5}, d_{2,3}, d_{2,5}, d_{3,4}, d_{4,5}] = d_{2,3} = 2,6, \text{ e}$$

$$d_{(1,2,4),(3,5)} = \max [d_{1,3}, d_{1,5}, d_{2,3}, d_{2,5}, d_{3,4}, d_{4,5}] = d_{4,5} = 4,4.$$

$$\begin{array}{cc}
 & \begin{array}{cc} 1,2,4 & 3,5 \end{array} \\
 \begin{array}{c} 1,2,4 \\ 3,5 \end{array} & \begin{bmatrix} 0,0 & 2,6 \\ & 0,0 \end{bmatrix}
 \end{array}
 \qquad
 \begin{array}{cc}
 & \begin{array}{cc} 1,2,4 & 3,5 \end{array} \\
 \begin{array}{c} 1,2,4 \\ 3,5 \end{array} & \begin{bmatrix} 0,0 & 4,6 \\ & 0,0 \end{bmatrix}
 \end{array}$$

Finalmente, os grupos (1,2,4) e (3,5) são combinados para formar um único grupo, para as duas matrizes acima. Os correspondentes *dendrograms* são exibidos na Figura 5.

FIGURA 5 - *DENDROGRAMS DE PROXIMIDADE*



### 2.2.2 Métodos Divisivos

Os métodos de agrupamento hierárquico divisivos são basicamente de dois tipos: monotético (*monothetic*), que dividem dados considerando um único atributo de cada vez, e politético (*polythetic*), onde as divisões são baseadas em valores tomados por todos os atributos dos dados de uma vez (DUBES e JAIN, 1988; EVERITT, 1993).

Esta classe de métodos de agrupamento é menos popular que os métodos aglomerativos, e pouco referenciada. No método divisivo politético de MACNAUGHTON-SMITH et al. (1964), um grupo “divisor” vai acumulando indivíduos seqüencialmente, para os quais a dissimilaridade total com o grupo restante, menos sua dissimilaridade total com o grupo “divisor”, é um máximo. Quando esta diferença se torna negativa, o processo é repetido com os dois subgrupos, e assim sucessivamente. A medida de dissimilaridade usual é a distância Euclidiana média.

Como exemplo, consideremos a matriz D de distâncias, para sete indivíduos (EVERITT, 1993). O indivíduo usado para iniciar o grupo “divisor” é aquele cuja distância média dos outros indivíduos é máxima. No caso, o indivíduo 1. Logo, tem-se os grupos (1) e (2,3,4,5,6,7). A seguir, a distância média de cada indivíduo no grupo restante para indivíduos no grupo “divisor” é encontrada, seguida pela distância média de cada indivíduo no grupo restante para outros indivíduos no mesmo grupo. A diferença entre essas duas médias é calculada, como pode ser visto na Tabela 1.

	1	2	3	4	5	6	7
D =	1	2	3	4	5	6	7
	0,0						
	10,0	0,0					
	7,0	7,0	0,0				
	30,0	23,0	21,0	0,0			
	29,0	25,0	22,0	7,0	0,0		
	38,0	34,0	31,0	10,0	11,0	0,0	
	42,0	36,0	36,0	13,0	17,0	9,0	0,0

A diferença máxima é 16.4 para o indivíduo 3, o qual é acumulado no grupo “divisor”: (1,3) e (2,4,5,6,7). Isto é, o indivíduo 3 está mais próximo do grupo divisor e mais distante do grupo restante, por que tem a diferença positiva máxima.

TABELA 1 - DISTÂNCIAS PARA OS GRUPOS “DIVISOR” E “RESTANTE”

INDIVÍDUO	DIST. MÉDIA PARA GRUPO DIVISOR (A)	DIST. MÉDIA PARA GRUPO RESTANTE (B)	B - A
2	10,0	25,0	15,0
3	7,0	23,4	16,4
4	30,0	14,8	-15,2
5	29,0	16,4	-12,6
6	38,0	19,0	-19,0
7	42,0	22,2	-19,8

Repetindo o processo de busca de um elemento do grupo restante para integrar o grupo divisor, tem-se:

TABELA 2 - DISTÂNCIAS PARA OS GRUPOS “DIVISOR” E “RESTANTE”

INDIVÍDUO	DIST. MÉDIA PARA GRUPO DIVISOR (A)	DIST. MÉDIA PARA GRUPO RESTANTE (B)	B - A
2	8,5	29,5	21,0
4	25,5	13,2	-12,3
5	25,5	15,0	-10,5
6	34,5	16,0	-18,5
7	39,0	18,7	-20,3

Agora, o indivíduo 2 é acumulado no grupo “divisor”: (1,2,3) e (4,5,6,7), e o processo se repete novamente, como mostra a Tabela 3.

Como todas as diferenças são negativas, o particionamento em dois grupos está completo, sendo que os elementos em cada um dos grupos obtidos estão muito próximos entre



si. A partir daqui, o processo pode continuar para cada grupo separadamente, para se obter 3 grupos, 4 grupos, e assim sucessivamente até que cada indivíduo constitua um grupo isolado.

TABELA 3 - DISTÂNCIAS PARA OS GRUPOS “DIVISOR” E “RESTANTE”

INDIVÍDUO	DIST. MÉDIA PARA GRUPO DIVISOR (A)	DIST. MÉDIA PARA GRUPO RESTANTE (B)	B - A
4	24,7	10,0	-14,7
5	25,3	11,7	-13,6
6	34,3	10,0	-24,3
7	38,0	13,0	-25,0

Segundo EVERITT (1993), os métodos divisivos do tipo *monothetic* são geralmente usados quando os dados consistem de variáveis binárias.

### 2.2.3 Métodos Grafo-teóricos

Agrupamentos hierárquicos podem ser obtidos aplicando-se a teoria dos grafos ao problema do agrupamento, pois podem ser expressos em termos de uma série de grafos (ZAHN, 1971; HUBERT, 1974; DUBES e JAIN, 1988). Partindo da matriz de proximidades  $D_{n \times n} = [d(i,j)]$ , que expressa as dissimilaridades entre  $n$  amostras, as  $n(n-1)/2$  entradas de um dos lados da diagonal da matriz correspondem à permutação dos inteiros de 1 até  $n(n-1)/2$ , sem empates, isto é, as proximidades estão em uma escala ordinal.

Posto isto, um grafo de limiar  $G(v)$  é um grafo não direcionado e sem peso, sobre  $n$  nodos, acíclico e sem arestas múltiplas. Cada nodo representa uma amostra. Um grafo de limiar  $G(v)$  é definido para cada nível de dissimilaridade  $v$  pela inserção de uma aresta entre os nodos  $i$  e  $j$  se as amostras  $i$  e  $j$  são menos dissimilares que  $v$ . Isto é:

$$(i,j) \in G(v) \text{ se e somente se } d(i,j) \leq v, \text{ sendo } d(i,i)=0, \forall i. \quad (20)$$

Na seqüência, são descritos algoritmos para os métodos de agrupamento *single-link* e *complete-link* baseados em grafos de limiar. Ambos os algoritmos assumem uma matriz de dissimilaridades ordinal não contendo empates, e produzem uma seqüência aninhada de agrupamentos que podem ser representados por um *dendrogram* de limiar.

### 2.2.3.1 Algoritmo *Single-link*

O algoritmo *single-link* começa com um grafo contendo os  $n$  nodos, representando as  $n$  amostras, e nenhuma aresta. Arestas são adicionadas, uma a cada vez. Cada subgrafo conexo é um grupo e cada nova aresta estabelece um subgrafo conexo (DUDA e HART, 1973; HUBERT, 1974; DUBES e JAIN, 1988). O algoritmo descrito por DUBES e JAIN (1988):

- Passo 1. Começa com o grafo de limiar  $G(0)$ , o qual não contém arestas, e cada amostra constitui um único grupo. Faz  $k=1$ .
- Passo 2. Forma o grafo de limiar  $G(k)$  pela adição de uma aresta ligando dois nodos no grafo de acordo com a condição estabelecida em (20). Se o número de componentes (subgrafos maximais conexos) é menor que o número de grupos no agrupamento corrente, redefine o agrupamento corrente fazendo cada componente de  $G(k)$  como um grupo.
- Passo 3. Se  $G(k)$  consiste de um único grafo conexo, pára. Senão, faz  $k=k+1$  e vai ao passo 2.

Em outras palavras, usando uma seqüência ordenada de grafos  $G(v)$  definida pelo aumento do valor de  $d(i,j)$ , o método *single-link* une dois conjuntos num determinado nível

que têm uma única aresta existente entre os correspondentes subgrafos maximais para o menor valor possível de  $d(i,j)$ .

### 2.2.3.2 Algoritmo *Complete-link*

Ao trocarmos a palavra “conexo” pela palavra “completo” na definição do agrupamento *single-link*, tem-se o método chamado *complete-link*. O método *complete-link* une dois subgrupos que possuem todas as possíveis arestas definidas entre os correspondentes subgrafos maximais para o menor valor possível de  $v$ . O algoritmo (DUBES e JAIN 1988):

- Passo 1. Começa com o agrupamento disjunto correspondente ao grafo de limiar  $G(0)$ , o qual não contém arestas, e cada amostra constitui um único grupo. Faz  $k=1$ .
- Passo 2. Forma o grafo de limiar  $G(k)$  pela adição de uma aresta ligando dois nodos no grafo de acordo com a condição estabelecida em (20). Se dois dos grupos correntes formam um subgrafo maximal completo em  $G(k)$ , redefine o agrupamento corrente pela união destes dois grupos em um único grupo.
- Passo 3. Se  $k=n(n-1)/2$ , tal que  $G(k)$  é um grafo completo sobre os  $n$  nodos, pára. Senão, faz  $k=k+1$  e vai ao passo 2.

Os algoritmos *single-link* e *complete-link* podem ser extendidos de forma a utilizar matrizes de dissimilaridades (ou proximidades) sobre escalas proporcionais (DUBES e JAIN, 1988). Observa-se que os algoritmos definidos anteriormente estão baseados em uma matriz de proximidades ordinal. O *dendrogram* de proximidades lista o nível de dissimilaridade para cada agrupamento, enquanto que o *dendrogram* de limiar registra os agrupamentos na ordem em que são formados, independentemente do nível de dissimilaridade no qual os agrupamentos aparecem primeiro. Comparativamente ao *dendrogram* de proximidades, um grafo de

proximidades é um grafo de limiar no qual cada aresta tem um peso de acordo com sua proximidade.

O agrupamento *single-link* sobre  $G(v)$  é definido em termos de subgrafos conexos em  $G(v)$ ; o agrupamento *complete-link* usa subgrafos completos. Entretanto, nem todos os subgrafos maximais completos precisam ser grupos *complete-link*. A ordem pela qual os grupos são formados é decisiva, como demonstra o seguinte exemplo, apresentado por DUBES e JAIN(1988).

Seja a seguinte matriz de proximidades:

$$D_t = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left[ \begin{array}{ccccc} 0,0 & & & & \\ 6,0 & 0,0 & & & \\ 8,0 & 1,0 & 0,0 & & \\ 2,0 & 5,0 & 10,0 & 0,0 & \\ 7,0 & 3,0 & 9,0 & 4,0 & 0,0 \end{array} \right] \end{matrix}$$

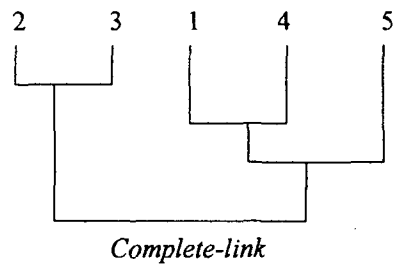
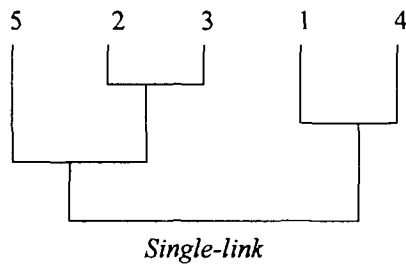
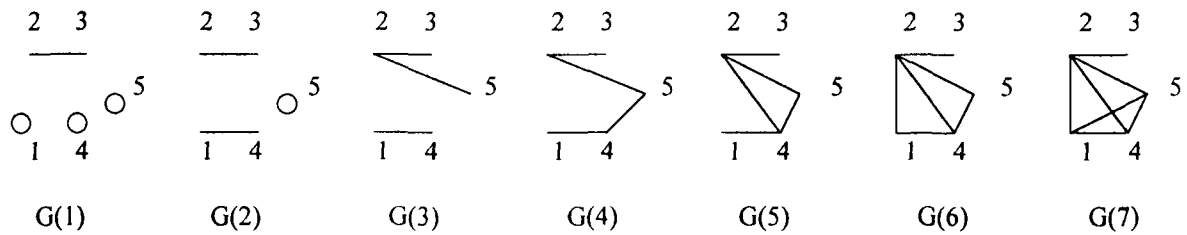
Os primeiros sete grafos de limiar na seqüência de 10 grafos de limiar são exibidos na Figura 6, com nodos rotulados tal que o nodo  $j$  denota a amostra  $x_j$ .

A hierarquia *single-link* inteira é definida pelos quatro primeiros grafos de limiar. Entretanto, os primeiros sete grafos de limiar são necessários para determinar a hierarquia *complete-link*. Uma vez obtidos os dois agrupamentos *complete-link*, grafos de limiar explícitos não são mais necessários porque os dois grupos somente serão unidos no agrupamento conjunto quando todas as  $n(n-1)/2$  arestas tiverem sido inseridas.

As amostras  $\{x_2, x_3, x_4\}$  formam um subgrafo maximal completo no grafo  $G(5)$ , mas as três amostras não são um grupo *complete-link*. Uma vez os grupos *complete-link*  $\{x_2, x_3\}$  e  $\{x_1, x_4\}$  estejam estabelecidos, a amostra  $x_5$  deve ser unida com um dos dois grupos estabelecidos; uma vez formados, os grupos não podem ser dissolvidos nem pode haver sobreposição de grupos.

FIGURA 6 - GRAFOS DE LIMIAR E *DENDROGRAMS* PARA OS AGRUPAMENTOS

HIERÁRQUICOS *SINGLE-LINK* E *COMPLETE-LINK*



FONTE: DUBES e JAIN(1988)

2.2.3.3 Árvore Geradora Mínima e o Algoritmo *Single-link*

Algoritmos eficientes e computacionalmente atrativos para agrupamentos hierárquicos podem ser obtidos da correspondência entre o método *single-link* e a árvore geradora mínima

sobre o grafo de proximidades completo conexo (ZAHN, 1971; DUDA e HART, 1973; DUBES e JAIN, 1980, 1988).

Uma árvore geradora para um grafo (conexo)  $G$  é um subgrafo de  $G$  tocando todos os nodos e não contendo ciclos. Uma árvore geradora contém exatamente  $(n-1)$  arestas. Uma árvore geradora mínima é a árvore geradora cuja soma dos pesos das aresta é mínima dentre todas as árvores geradoras (DUBES e JAIN, 1980).

A seguir, é descrito um algoritmo aglomerativo para o método *single-link* que assume a matriz de dissimilaridades (DUBES e JAIN, 1988).

- Passo 1. Inicia com o agrupamento disjunto, no qual cada amostra constitui um grupo isolado. Encontra a árvore geradora mínima sobre o grafo  $G$ .
- Passo 2. Une os dois grupos conexos pela aresta da árvore geradora mínima com o menor peso para definir o próximo agrupamento.
- Passo 3. Troca o peso da aresta selecionada no passo 2 por um peso maior que a maior proximidade. Repete os passos 2 e 3 até que todos os objetos formem um único grupo.

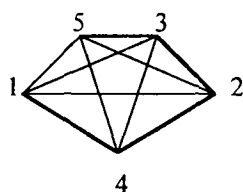
Da mesma forma, partindo da árvore geradora e removendo a aresta mais longa, produz-se um agrupamento de dois grupos; removendo a próxima aresta mais longa, produz-se um agrupamento de três grupos, e assim por diante. Tal procedimento pode definir um algoritmo divisivo. Assim, num algoritmo divisivo, simplesmente são feitos cortes na árvore geradora mínima por ordem de peso, cortando primeiro a maior. Cada corte define um novo agrupamento, com aquelas amostras ligadas na árvore geradora mínima em qualquer nível pertencendo ao mesmo grupo.

Para exemplificar, seja a matriz de proximidades  $D_1$  (DUBES e JAIN, 1988):

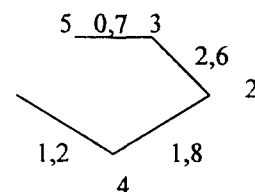
$$D_1 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0,0 & & & & \\ 2,3 & 0,0 & & & \\ 3,4 & 2,6 & 0,0 & & \\ 1,2 & 1,8 & 4,2 & 0,0 & \\ 3,7 & 4,6 & 0,7 & 4,4 & 0,0 \end{bmatrix} \end{matrix}$$

A Figura 7 mostra os agrupamentos aglomerativo e divisivo baseados na árvore geradora mínima, bem como o correspondente *dendrogram*.

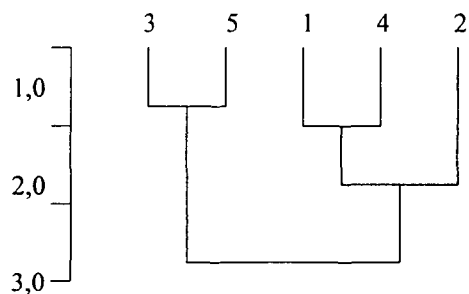
FIGURA 7 - EXEMPLOS DE ALGORITMOS *SINGLE-LINK* AGLOMERATIVO E DIVISIVO BASEADOS NA ÁRVORE GERADORA MÍNIMA



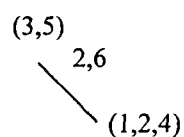
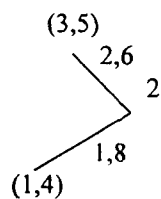
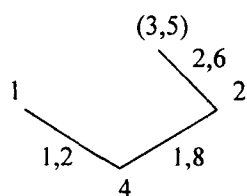
Grafo completo



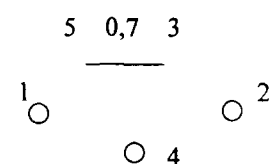
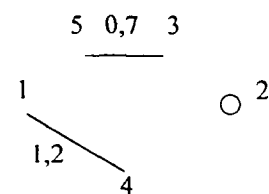
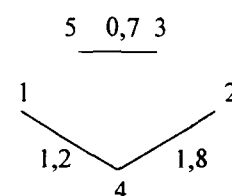
Árvore geradora mínima



*Dendrogram single-link*



Aglomerativo



Divisivo

FONTE: DUBES e JAIN (1988)

### 3 APLICAÇÕES

Neste capítulo, são abordados diversos trabalhos que tratam do problema do agrupamento ou que aplicam alguma técnica de agrupamento à segmentação de imagens.

Muitas destas abordagens têm como objetivo se aproximar do particionamento ótimo para o agrupamento, isto é, aquele que minimiza globalmente a função critério. Outros buscam por um método adequado para segmentar determinado tipo de imagem, com um custo computacional aceitável.



Também é comum encontrar trabalhos voltados a introduzir heurísticas a determinados algoritmos de agrupamento a fim de torná-los não supervisionados, isto é, executáveis sem interação humana.

A seguir, as diferentes abordagens são descritas e comentadas.

### 3.1 BASEADAS EM MÉTODOS PARTICIONAIS

3.1.1 A proposta de COLEMAN e ANDREWS (1979) descreve um procedimento para segmentação de imagens que opera num modo não supervisionado, isto é, sem interação humana.

Determinado o conjunto de características da imagem, o algoritmo *K-means* é utilizado para  $k = 2, 3, \dots, 16$  grupos, calculando a cada  $k$  a qualidade do agrupamento, que é o produto das medidas (*between* e *within*, ver Apêndice 2) inter e intra-grupo:

$$\beta = tr S_B \cdot tr S_W, \quad (21)$$

o qual alcança um valor máximo no número adequado de grupos (*tr* indica o traço da matriz). A medida intra-grupo baseia-se sobre os dados e médias de grupo, e a medida inter-grupo é definida sobre as médias de grupo e a média de todos os pontos:

$$S_W = (1/K) \sum_{k=1}^K \varepsilon \{ (x - \mu_k) (x - \mu_k)' \}, \text{ onde:} \quad (22)$$

$$\varepsilon \{ . \} = (1/M_k) \sum_{x_i \in S_k} (x_i - \mu_k) (x_i - \mu_k)', \quad (23)$$

e  $\mu_k$  é o centróide do  $k$ -ésimo grupo,  $M_k$  é o número de elementos no  $k$ -ésimo grupo,  $x_i$  é um elemento no  $k$ -ésimo grupo, e  $K$  é o número total de grupos.

$$S_B = 1/K \left( \sum_{k=1}^K (\mu_k - \mu_0) (\mu_k - \mu_0)' \right), \text{ onde:} \quad (24)$$

$$\mu_0 = 1/M \left( \sum_{i=1}^M x_i \right). \quad (25)$$

Aqui  $M$  representa o número total de pontos para ser agrupados.

A cada  $k$  é calculada a medida Bhattacharyya para cada característica. Esta medida é utilizada para a seleção do conjunto característico ideal, dada por:

$$B_n(S_i, S_j) = (1/4) \ln \{ (1/4) [(\sigma_i^2(n)/\sigma_j^2(n)) + (\sigma_j^2(n)/\sigma_i^2(n)) + 2] \} \\ + (1/4) \{ [\mu_i(n) - \mu_j(n)]^2 / [\sigma_i^2(n) + \sigma_j^2(n)] \}, \quad (26)$$

onde  $S_i$  e  $S_j$  referem-se aos grupos  $S_i$  e  $S_j$ ,  $n$  refere-se à  $n$ -ésima dimensão do espaço característico,  $\sigma_i$  e  $\sigma_j$  são as variâncias do  $i$ -ésimo e  $j$ -ésimo grupos na dimensão  $n$ .

O critério de rejeição de característica é para somente reter as características com valor Bhattacharyya grande. A redução de características melhora a qualidade da segmentação pelo descarte de características menos úteis, diminuindo o tempo de execução.

O agrupamento final é então realizado, utilizando o conjunto de características reduzido, sendo o número de grupos novamente determinado por  $\beta$ .

Os experimentos foram realizados, primeiramente, com uma imagem monocromática, a partir da qual foi gerado um conjunto com 12 características, relacionadas ao brilho e ao gradiente obtido pelo operador de Sobel (GONZALEZ e WOODS, 1992), com diferentes tamanhos de janela. Estas características foram sujeitas ao agrupamento, a partir do que se determinou o número de grupos pelo valor máximo de  $\beta$ . No valor máximo de  $\beta$ , as quatro melhores características foram selecionadas, através da medida Bhattacharyya. Todas as quatro características selecionadas correspondem a funções derivadas do brilho da imagem original. Finalmente, as características selecionadas foram usadas para novamente para segmentar a imagem em  $k = 2, 3, \dots, 16$  grupos, sendo a melhor partição no valor máximo de  $\beta$ .

Os mesmos procedimentos foram aplicados a imagens policromáticas, utilizando 15 características relacionadas ao brilho para os componentes *red*, *green* e *blue*, e janelas de tamanhos diferentes.

Sob o enfoque de nosso estudo, uma grande contribuição deste trabalho é a introdução de um critério para selecionar o número mais adequado de grupos, embora sejam necessários diversos agrupamentos para defini-lo.

3.1.2 HOFFMAN e JAIN (1987) utilizam o algoritmo CLUSTER para um esquema de segmentação utilizando como características as coordenadas dos pontos de superfície e suas normais associadas. O vetor normal unitário associado a cada *pixel* é estimado encontrando o melhor plano de ajuste em uma vizinhança  $m \times m$  do *pixel*. Nos experimentos realizados, a vizinhança 5x5 deu bons resultados no que diz respeito à minimização de efeitos de propagação de borda e de ruído.

A seqüência de agrupamentos é realizada para o intervalo de 2 a 20 grupos. A cada agrupamento, é calculada a distância média intra-grupo, definida como:

$$\text{CLAVGD}(i) = (1 / |G_i|) \sum_{x \in G_i} d(x, c(i)), \quad (27)$$

onde  $d$  indica a distância Euclidiana,  $c(i)$  é o centro do grupo  $i$ , e  $G_i$  é o número de pontos pertencentes ao grupo  $i$ .

Da mesma forma, para cada grupo é calculada a medida  $M(i)$  seguinte, a qual reflete a separabilidade e compactação do grupo  $i$ :

$$M(i) = (\min_{j:j \neq i} d^2(c(i), c(j))) / \text{CLAVGD}(i). \quad (28)$$

A partir de  $M(i)$ , para cada grupamento é calculada a medida  $M_{\text{ave}}$ , que é a média ponderada das  $M(i)$ 's, onde os pesos são os números de *pixels* nos grupos. Aqueles agrupamentos com valores grandes de  $M_{\text{ave}}$  são preferidos sobre aqueles com valores pequenos.

Na prática, observou-se que  $M_{\text{ave}}$  freqüentemente atinge um máximo global precoce (e.g., uma solução 3 grupos), o que tornou a idéia de aceitar o agrupamento cujo valor de  $M_{\text{ave}}$  foi o maior inadequada. Por este motivo optou-se pelo algoritmo CLUSTER com limite superior de 20 grupos. Assim, o critério utilizado foi a identificação, dentre a seqüência de agrupamentos, aquele com maior número de grupos fora do conjunto de agrupamentos com valores de  $M_{\text{ave}}$  que foram máximos locais.

Numa segunda fase, os segmentos de superfície obtidos são classificados como planos, côncavos ou convexos. Finalmente, uma técnica de *merging* é aplicada para a reconstrução das superfícies dos objetos, pois estes, muitas vezes, são super-segmentados na fase de agrupamento.

Neste trabalho, a solução encontrada para determinar o número ótimo de grupos  $k$  é custosa, pois necessita de uma seqüência de agrupamentos para valores diferentes de  $k$ , além do que utiliza um critério que não dá diretamente este número procurado.

3.1.3 VENKATESWARLU e RAJU (1992) propõem melhoramento no algoritmo *K-means* com distância Euclidiana no que se refere ao processo de encontrar uma partição ótima. Tal melhora é obtida pelo uso das abordagens de soma parcial (*partial sum*) e de distância de vizinhança mais próxima (NND *nearest neighbouring distance*). A finalidade dessas abordagens é acelerar o algoritmo *K-means*, já que é sabido que um dos problemas a ele relacionado é a sua convergência a uma solução mínima local, sendo necessárias múltiplas execuções para se aproximar de uma solução mínima global, o que demanda um esforço computacional muito grande.

Considerando a distância Euclidiana entre um elemento e os centros de grupos como regra de classificação, um elemento é designado para o grupo do qual é mais próximo, o que torna necessário o cálculo das distâncias para todos os centros de grupos. O que se busca é o grupo cuja distância para um determinado elemento é mínima. A abordagem de soma parcial, segundo estes autores, baseia-se na idéia de que o cálculo da distância para um grupo estará terminado, se sua soma acumulada excede o mínimo encontrado. Isto porque a distância de um elemento para um centro de grupo é um somatório de distâncias, pois cada elemento é definido

por uma série de características. Assim, calculada a distância do elemento para o primeiro grupo, a partir daí o cálculo em relação aos demais grupos pode ser interrompido se a soma parcial das distâncias em relação a cada característica superar o valor obtido para o primeiro grupo. Tal economia, considerando que o número de amostras de um conjunto é relativamente grande, leva a uma aceleração do algoritmo.

A NND de uma classe é a distância para sua classe mais próxima em espaço  $n$ -dimensional. Neste algoritmo, a distância Euclidiana quadrática é usada para o cálculo da NND. Inicialmente, para cada grupo ( $c$  grupos), a NND é calculada e metade de cada uma é guardada em uma matriz ( $NND(k)$ ,  $k=1, \dots, c$ ). Quando examinando a classe candidata  $c_1$ , ( $c_1 < c$ ), em relação a uma amostra, o exame das demais classes ( $c - c_1$  classes) pode ser ignorado se a distância Euclidiana da amostra para a classe é menor que o respectivo valor na matriz NND, pois a amostra definitivamente pertence à  $c_1$ -ésima classe. Tal método também provoca uma aceleração no algoritmo de segmentação. Maior velocidade pode ainda ser obtida pela combinação dos métodos de soma parcial e de distância de vizinhança mais próxima.

Outro problema relacionado ao algoritmo *K-means* com distância Euclidiana é que ele é bem adequado quando os grupos formam essencialmente nuvens compactas e bem separadas umas das outras. A utilização da distância Mahalanobis no lugar da distância Euclidiana permite a melhor identificação da estrutura de grupo presente no conjunto de amostras, porém requer um número maior de cálculos.

Assim, os autores propõem o uso da abordagem de matriz triangular inferior (*lower triangular matrix*) para reduzir os cálculos requisitados pelo uso da distância Mahalanobis, bem como propõem a abordagem de função de distância expandida (*expanded distance function*) para acelerar o processo de agrupamento com distância Mahalanobis. A distância Mahalanobis fica definida como:

$$d = (X_k - Y_j)^T \Sigma^{-1} (X_k - Y_j), \quad (29)$$

onde  $\Sigma$  é a matriz de covariância associada,  $X_k$  representa uma amostra e  $Y_j$  o centróide do grupo  $j$ . A matriz de covariância é definida como:

$$\Sigma = 1/N * \sum_{i=1}^N (y_i - Y)(y_i - Y)^T, \quad (30)$$

onde  $N$  é o número de amostras e  $Y$  a média global.

É possível representar  $\Sigma^{-1}$  em termos da matriz triangular inferior  $L$ :

$$\Sigma^{-1} = L^T L, \text{ onde:} \quad (31)$$

$$L = \begin{bmatrix} l_{11} & 0 & 0 & 0 & \dots \\ l_{21} & l_{22} & 0 & 0 & \dots \\ l_{31} & l_{32} & l_{33} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad (32)$$

Ao usar a matriz  $L$ , a função de distância Mahalanobis pode ser calculada com as equações:

$$Y_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \dots \\ y_{iN} \end{bmatrix} \quad (33)$$

$$Y_i = L(X - Y_i) = \begin{bmatrix} V_1^T \\ V_2^T \\ \dots \\ V_N^T \end{bmatrix} (X - Y_i) \quad (34)$$

$$y_{ik} = V_k^T (X - Y_i) \quad (35)$$

$$g_i(X) = \sum_{k=1}^N y_{ik}^2 \quad (36)$$

Aqui  $V_k^T$  é a  $k$ -ésima linha da matriz  $L$ . Como  $V_k^T$  é zero à partir do  $k$ -ésimo elemento, o número de multiplicações para calcular  $y_{ik}$  é somente  $k$ . Maior velocidade ainda pode ser obtida combinando-se a abordagem da matriz triangular inferior com a da soma parcial.

Da mesma forma, a abordagem da função de distância expandida pode ser utilizada para acelerar o algoritmo *K-means* com distância Mahalanobis. A Equação 29 da distância Mahalanobis entre  $X$  e o  $i$ -ésimo grupo pode ser escrita na forma expandida:

$$g_i(X) = X^T \Sigma^{-1} X - X^T \Sigma^{-1} Y_i - Y_i^T \Sigma^{-1} X + Y_i^T \Sigma^{-1} Y_i = X \Sigma^{-1} X - 2X \Sigma^{-1} Y_i + Y_i \Sigma^{-1} Y_i. \quad (37)$$

Da última equação, o primeiro termo é comum para cada grupo e pode ser eliminado no processo de busca. O termo  $2X \Sigma^{-1} Y_i$  é trocado por  $W_i$ , que é um vetor para o correspondente grupo. Logo,

$$g_i(X) = Y_i^T \Sigma^{-1} Y_i - X^T W_i. \quad (38)$$



Na equação acima, o primeiro termo é constante para cada grupo e pode ser calculado somente uma vez no laço do agrupamento em cada iteração, reduzindo o cálculo da distância à multiplicação dos vetores  $X$  e  $W_i$ . Além disso, a abordagem da soma parcial também pode ser associada à função de distância expandida, obtendo-se maior velocidade do algoritmo *K-means*.

No mesmo trabalho são propostas formas de acelerar o algoritmo de agrupamento dinâmico pelo uso das abordagens de soma parcial e de matriz triangular inferior.

O agrupamento dinâmico caracteriza-se por permitir o uso de modelos arbitrários para representação de grupo preferivelmente à média de grupo, o que produz uma melhor partição, quando comparado ao algoritmo *K-means*. O algoritmo de agrupamento dinâmico utilizado é assim definido:

Passo 1. Particiona arbitrariamente o conjunto  $Y$  de tamanho  $N$  em  $C$  grupos,  $w_i$ ,  $i=1, \dots, C$ , para inicializar o algoritmo.

Passo 2. Calcula  $N_i$ ,  $Y_i$  e  $\Sigma_i$ , que são o número de amostras do  $i$ -ésimo grupo, o vetor médio e a matriz de covariância, respectivamente.

Passo 3. Atribui cada (múltiplo) ponto  $y_j \in Y$ , e  $y_j \in w_j$ , no mais apropriado grupo  $w_r$  se

$$\ln |\Sigma_r| + g_r(y_j, Y_r) - 2 \ln(N/N) = \min \{ \ln |\Sigma_i| + g_i(y_j, Y_i) - 2 \ln(N_i/N) \}, \text{ onde:} \quad (39)$$

$$g_r(y_j, Y_r) = (y_j - Y_r)^T \Sigma_r^{-1} (y_j - Y_r). \quad (40)$$

Passo 4. Se nenhuma amostra troca sua associação a um grupo, então o algoritmo termina, senão volta ao passo 2.

Os resultados obtidos experimentalmente, utilizando um conjunto de 200 amostras, demonstraram que as abordagens NND e soma parcial em associação, resultou num melhor desempenho ao algoritmo *K-means*. Já no caso do algoritmo *K-means* com distância Mahalanobis, o método da função de distância expandida mostrou melhores resultados que a

abordagem da matriz triangular inferior. Quanto ao agrupamento dinâmico, o uso do método da matriz triangular inferior tornou mais rápido o processo de agrupamento.

Esta abordagem é interessante pois traz melhorias no algoritmo de agrupamento *K-means* com múltiplas execuções, quanto ao problema relacionado a requisitos computacionais. Como sabemos, entretanto, o resultado do agrupamento depende muito da configuração inicial dos grupos bem como do número de grupos. Desta forma, a associação destas técnicas de aumento de velocidade a outras que visem uma melhor distribuição inicial dos grupos, pode levar a um método mais confiável, e que cujos resultados se aproximem da solução ótima global.

3.1.4 AL-SULTAN (1995) desenvolve um novo algoritmo para resolver o problema do agrupamento, o qual apresenta soluções que são mínimos locais. O algoritmo proposto baseia-se na técnica de busca *tabu*, que é uma heurística que pode ser usada para resolver problemas de análise combinatorial. A idéia principal é gerar, a partir de uma solução corrente, outra solução relacionada com a solução corrente. Se o valor da função critério da solução gerada é melhor que o da solução corrente, ela passa a ser a configuração corrente. Caso contrário, poderá passar a ser a configuração corrente se determinada condição for satisfeita. O processo se repete iterativamente, fazendo com que chegue a uma solução melhor do que uma solução que é mínimo local. Cada solução que passa a ser a solução corrente, é guardada em uma lista (lista *tabu*), o que evita que ela seja reavaliada nas próximas iterações. O algoritmo desenvolvido é descrito a seguir.

Seja  $A$  uma matriz de dimensão  $m$  cujo  $i$ -ésimo elemento é um número representando o grupo para o qual a  $i$ -ésima amostra pertence. Assim, todos  $w_{ij}$ 's são definidos da seguinte maneira:

$$w_{ij} = \begin{cases} 1, & \text{se } A_i=j; \\ 0, & \text{caso contrário;} \end{cases} \quad (41)$$

$$\forall i=1,2,\dots,m, \text{ e } \forall j=1,2,\dots,c.$$

Por exemplo, seja a matriz  $A$  seguinte, para  $m=10$  e  $c=3$ :

3	2	1	2	2	3	1	1	2	3
---	---	---	---	---	---	---	---	---	---

A primeira amostra é atribuída ao terceiro grupo, isto é,  $w_{11}=0$ ,  $w_{12}=0$  e  $w_{13}=1$ . A segunda amostra é atribuída ao segundo grupo, isto é,  $w_{21}=0$ ,  $w_{22}=1$  e  $w_{23}=0$ , e assim por diante.

Dado um conjunto de  $w_{ij}$ 's, o centro  $z_j$  de cada grupo pode ser calculado como:

$$z_j = \left( \sum_{i=1}^m w_{ij} x_{ij} \right) / \left( \sum_{i=1}^m w_{ij} \right). \quad (42)$$

Dada a matriz  $A$ , e os centros  $z_j$ , podemos calcular a função objetiva  $J(w,z)$  como:

$$J(w,z) = \sum_{i=1}^m \sum_{j=1}^c w_{ij} \|x_i - z_j\|^2. \quad (43)$$

O algoritmo opera com uma configuração chamada corrente  $A_c$  e através de movimentos gera soluções teste  $A_t$ . A melhor solução encontrada é denotada por  $A_b$ . Correspondendo a estas matrizes, estão os correspondentes  $J_c$ ,  $J_t$  e  $J_b$  valores de funções objetivas. A seguir, o algoritmo:

**Inicialização.** Seja  $A_c$  uma solução arbitrária, e  $J_c$  o correspondente valor da função objetiva. Seja  $A_b=A_c$  e  $J_b=J_c$ . Selecciona valores para os seguintes parâmetros: tamanho da lista *tabu* (MTLS), limiar de probabilidade  $P$ , número de soluções teste (NTS) e número máximo de iterações ITMAX. Seja  $k=1$  e comprimento da lista *tabu* TLL=0.

**Passo 1.** A partir de  $A_c$ , gera soluções teste  $A_t^1, A_t^2, \dots, A_t^{\text{NTS}}$ , e avalia seus correspondentes valores de função objetiva  $J_t^1, J_t^2, \dots, J_t^{\text{NTS}}$ .

**Passo 2.** Ordena  $J_t^1, J_t^2, \dots, A_t^{\text{NTS}}$  em ordem ascendente. Se  $J_t^1$  não é *tabu*, ou se é *tabu* mas  $J_t^1 < J_b$ , então  $A_c=A_t^1$  e  $J_c=J_t^1$ ; caso contrário,  $A_c=A_t^l$  e  $J_c=J_t^l$ , onde  $J_t^l$  é a melhor função objetiva de  $J_t^2, \dots, J_t^{\text{NTS}}$  que não é *tabu*. Se todos  $J_t^2, \dots, J_t^{\text{NTS}}$  são *tabu* vai para o passo 1.

**Passo 3.** Insere  $A_c$  na base da lista *tabu* e TLL=TLL+1 (se TLL=MTLS=1, elimina o primeiro elemento da lista e TLL=TLL-1). Se  $J_b > J_c$  faz  $A_b=A_c$  e  $J_b=J_c$ . Se  $k=ITMAX$ , pára ( $A_b$  é a melhor solução encontrada e  $J_b$  o correspondente valor da melhor função objetiva); caso contrário,  $k=k+1$  e vai para o passo 1.

A partir da solução corrente  $A_c$ , uma solução teste pode ser gerada usando várias estratégias. A estratégia usada aqui estabelece um limiar de probabilidade  $P$ , para  $i=1, \dots, m$ , obtém um número aleatório  $R \sim u(0,1)$ . Se  $u(0,1) < P$ , então  $A_t(i)=A_c(i)$ ; caso contrário, obtém aleatoriamente um inteiro  $l$  do seguinte conjunto  $\{l: l=1,2,3,\dots,c, l \neq A_c(j)\}$  e  $A_c(j)=l$ .

Cada configuração analisada, presente ou não melhor valor para  $J$ , vai sendo inserida na lista *tabu*. Isto para evitar que uma certa configuração seja analisada mais de uma vez. Obviamente, o tamanho da lista *tabu* restringe tal procedimento, pois uma vez a lista *tabu* esteja cheia e uma nova configuração deva ser inserida na lista, a primeira configuração da lista é liberada (a lista *tabu* é circular).

Os resultados dos testes realizados com conjuntos de dados aleatórios e conjuntos de dados padrão, demonstraram a superioridade da proposta em relação ao algoritmo *K-means* e ao algoritmo *simulated annealing*. Mesmo não garantindo a solução global, é maior a probabilidade desta ser atingida. A técnica não define nenhuma heurística para se obter uma solução inicial para o algoritmo, a partir da qual são geradas as soluções teste. Parece-nos que, definindo uma razoável configuração inicial, é possível diminuir o número de iterações ou até mesmo melhorar ainda mais a solução final para o agrupamento.

3.1.5 ZHANG e BOYLE (1991) analisam o método do movimento (*Moving method*) para segmentação de conjuntos de dados.

Relatado primeiramente por DUDA e HART (1973), o método do movimento, a partir de uma configuração inicial de agrupamento, testa cada elemento para determinar se o custo (erro-quadrático) diminui se este elemento é movido para outro grupo. Se o custo diminui, o elemento é movido e recalculados os centróides dos dois grupos envolvidos. O processo se repete iterativamente como no algoritmo *K-means*, até que não mais hajam trocas na configuração de grupo, ou até que se atinja o número máximo de iterações. Desta forma, o método do movimento fica dependente da configuração inicial de agrupamento. Através de seus experimentos, os pesquisadores verificaram que múltiplas execuções do método do movimento, através da reinicialização com uma configuração aleatoriamente selecionada, levam a resultados superiores ao algoritmo *K-means*.

O trabalho introduz outra troca de configuração, não aleatória, baseada no estado de convergência prévio. O método, chamado troca de configuração controlada (*controlled configuration change*), torna o estado de convergência sujeito a uma seqüência de

perturbações, formando uma nova configuração. As duas operações primitivas são *delete* e *partition*. *Delete* cancela um grupo e atribui seus elementos para os grupos mais próximos remanescentes, diminuindo em um o número de grupos. *Partition* divide um grupo em dois grupos, sendo os elementos mínimo e máximo deste grupo (mais próximo e mais distante do centróide, respectivamente), os centróides dos novos grupos, e todos os elementos do grupo original são atribuídos a um dos novos grupos, de acordo com a maior proximidade. Após esta operação o número de grupos aumenta de um. Uma seqüência de tais operações ocorre de forma que, ao final, o número de grupos deve retornar ao valor estabelecido inicialmente.

Uma seqüência de operações é definida como uma string consistindo de dois caracteres,  $d$  e  $p$ , na qual o número de  $d$  é igual ao número de  $p$ . Nesta seqüência de operações,  $d$  representa *delete* e  $p$  representa *partition*. Uma troca controlada em configuração de grupo pode ser escrita como segue:

Passo 1. Cria uma seqüência de operações;

Passo 2. Para cada caracter nesta seqüência de operações:

- a. Seleciona um grupo;
- b. Implementa a operação associada com este caracter sobre o grupo selecionado.

O procedimento para criar uma seqüência de operações é baseado no limiar de probabilidade  $P_v(d)$  (Equação. 44) e esvazia a possibilidade de repetição de uma seqüência de operações:

Passo 1. Seja  $v=m$ , onde  $m$  é o número de grupos; inicializa a seqüência de operações como vazia;

Passo 2. Repete:

- a. Gera um valor aleatório  $(0,1)$ ; se este valor é menor que  $P_v(d)$ , então anexa o caracter  $d$  nesta seqüência de operações, senão anexa  $p$ ;

$$P_v(d) = 1 / (1 + e^{(-g(v-m))}), \quad (44)$$

onde o parâmetro  $g$  controla a probabilidade de desvios significativos de  $m$ ;

- b. Se o caracter corrente é  $d$ , decrementa  $v$ , senão incrementa  $v$ ;

Passo 3. Até  $v=m$ ;

Passo 4. Se esta seqüência de operações está incluída no conjunto de seqüência de operações corrente, vá para (1);

Passo 5. Saída desta seqüência de operações.

Nos experimentos realizados, foi escolhido o valor de  $g=1.0$ , por que valores pequenos de  $g$  geram seqüências longas, cujo efeito é pequeno no sucesso dos algoritmos.

Quanto à estratégia para a seleção de grupo para as duas operações primitivas, o objetivo de uma troca controlada é manter partes “boas” da estrutura e trocar partes “más”, a fim de alcançar melhores resultados no agrupamento. A estratégia para a operação *partition* é selecionar o grupo com o maior valor de erro quadrático médio. Já o critério para a escolha do grupo para *delete*, baseado numa propriedade entre pares de grupos, é determinar o par de centros de grupos que estão mais próximos em distância Euclidiana, e então selecionar o grupo deste par com o maior valor de erro quadrático médio.

O algoritmo CMA (*controlled moving algorithm*) proposto, que é um algoritmo de agrupamento com múltiplas execuções do método do movimento e que introduz a troca de configuração controlada a partir do estado de convergência prévio, tem os seguintes passos:

Passo 1. Entrada do parâmetro  $r$ , que é o número de configurações;

Passo 2. Inicializa aleatoriamente uma configuração de grupo;

Passo 3. Executa o método do movimento para convergir a configuração;

Passo 4. Salva o resultado do agrupamento como o melhor;

Passo 5. Para  $i=1$  até  $r-1$ :

- a. Forma uma nova configuração pela troca controlada da melhor configuração;
- b. Executa o método do movimento para convergir a nova configuração;
- c. Se este resultado do agrupamento é melhor que o melhor anterior, salva este como o melhor;

Passo 6. Saída do melhor resultado do agrupamento.

Experimentos realizados com o algoritmo CMA e com múltiplas execuções do método do movimento com troca de configuração aleatória, demonstraram que o algoritmo CMA usualmente alcança melhores resultados de agrupamento com o mesmo número de execuções, e com um custo computacional comparável à troca aleatória. Isto é, em alguns casos o algoritmo CMA alcança o mínimo global e mais rapidamente, porém não é possível garantir esta superioridade.

Outra contribuição deste trabalho é o algoritmo *Auto-Stop-CMA*, que pára automaticamente quando um resultado satisfatório de agrupamento é encontrado. Isto é feito apenas alterando a condição de parada do laço de repetição do algoritmo CMA que, ao invés de ser executado um número fixo de vezes, é repetido até que não hajam resultados melhores após um certo número de perturbações consecutivas. Para alguns conjuntos de dados este algoritmo alcança o mínimo global logo, não sendo necessário um valor de  $r$  muito grande para alcançar este estado final.

Concluindo, esta proposta é bastante interessante, pois introduz um algoritmo de agrupamento com múltiplas execuções do método do movimento. Neste algoritmo, a troca de configuração controlada é realizada no estado de convergência do método do movimento. O algoritmo pára automaticamente quando um resultado de agrupamento satisfatório é alcançado. Assim, na pior situação, este algoritmo chega aos mesmos resultados que o



algoritmo *K-means*. Porém, aumenta a probabilidade de se chegar a um resultado próximo do mínimo global. Embora tal resultado não possa ser generalizado, a proposta abre uma perspectiva para trabalhos suplementares para resolver o problema do agrupamento no que diz respeito a garantir que se alcance o mínimo global.

3.1.6 BENSALD et al. (1996) descreve um novo método de segmentação de imagens semi-supervisionado *fuzzy K-means*, visando superar os seguintes problemas: (1) escolha e validação do número de grupos na imagem; (2) garantia de que rótulos algorítmicos correspondam a rótulos físicos significativos dos grupos na imagem; e (3) tendência a parar em soluções que equalizam populações dos grupos. Todos os algoritmos de agrupamento sofrem de (1) e (2), e (3) é uma desvantagem bem conhecida dos algoritmos *hard K-means* e *fuzzy K-means*.

Os problemas algorítmicos na determinação do número de classes e seus rótulos (problemas (1) e (2)) são superados pelo uso de dados rotulados de cada classe (ao menos alguns). O algoritmo proposto é particularmente bem adequado à segmentação de imagens porque um humano pode examinar a imagem e selecionar claramente uns poucos *pixels* de cada classe, e subseqüentemente usá-los para guiar a rotulação dos *pixels* restantes. Esse pequeno conjunto de *pixels* rotulados dá ao algoritmo de agrupamento uma forma de supervisão parcial.

Quando, em qualquer algoritmo de agrupamento, incluindo *hard K-means* e *fuzzy K-means*, o conjunto de dados  $X$  é composto inteiramente de dados não rotulados, o método é completamente não supervisionado. Uma vez selecionado o número de classes  $c$ , partições de  $X$  têm a seguinte forma:

$$U = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ u_{21} & \dots & u_{2k} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ u_{c1} & \dots & u_{2k} & \dots & u_{cn} \end{bmatrix} \quad (45)$$

Como os  $n$  vetores rótulo não são conhecidos na inicialização, a matriz  $U$  inicial é aleatória. Ao final do agrupamento, cada linha de  $U$  corresponde a uma classe, pois  $u_{ik}$  indica o grau de relacionamento de  $x_k$  com a  $i$ -ésima classe.

Do outro extremo, se o processo se inicia com rótulos rígidos (1 ou 0) para todas as  $n$  amostras, então o método é supervisionado, não sofrendo dos problemas (1) e (2). Entretanto, a supervisão requer intervenção humana para a obtenção dos dados rotulados, o que pode ser impossível devido ao número de amostras em uma imagem.

O esquema proposto fica entre estes dois extremos. Os problemas (1) e (2) ficam esvaziados pelo uso de alguns dados rotulados de cada classe (amostras de treinamento), ficando a maioria das amostras na forma não rotulada na inicialização. Para garantir que o problema (3) se esvazie, é feita a introdução de pesos  $w$  às amostras rotuladas. Desta forma, é possível garantir que os grupos não sejam equalizados. Um grupo pequeno pode ser preservado pela introdução de altos pesos aos dados de treinamento correspondentes a este grupo.

A seguir, a descrição do algoritmo *fuzzy K-means* semi-supervisionado, no qual  $t$  é relativo a dados de treinamento,  $s$  é relativo a dados não rotulados, e  $n$  é o número de amostras:

Passo 1. Sejam dados parcialmente rotulados  $X=X^t \cup X^s$ ,  $n^t=|X^t|$ ,  $n^s=|X^s|$ , e  $n=|X|=n^t+n^s$ . Seja  $c$  o número de classes fixado pelos dados de treinamento.

Passo 2. Escolhe os parâmetros  $w$  (vetor de pesos),  $T$  (número máximo de iterações),  $\|\cdot\|_A$  (norma),  $m>1$  e  $\varepsilon>0$ .

Passo 3. Inicializa  $U=[U^t|U^s]$ ;

Passo 4. Calcula:

$$v_{i,0} = \left( \sum_{k=1}^{n_t} (u'_{ik,0})^m x'_k \right) / \left( \sum_{k=1}^{n_t} (u'_{ik,0})^m \right), \quad 1 \leq i \leq c. \quad (46)$$

Passo 5. Para  $t=1,2,\dots,T$ :

a. Calcula:

$$u^s_{ik,t} = \left\{ \sum_{j=1}^c [ \|x^s_k - v_{j,t-1}\|_A / \|x^s_k - v_{i,t-1}\|_A ]^{2/(m-1)} \right\}^{-1}, \quad 1 \leq i \leq c; \quad 1 \leq k \leq n_s. \quad (47)$$

b. Calcula:

$$E_t = \|U^s_t - U^s_{t-1}\| = \sqrt{\sum_{i=1}^c \sum_{k=1}^{n_s} (u^s_{ik,t} - u^s_{ik,t-1})^2}. \quad (48)$$

c. Se  $E_t \leq \varepsilon$  pára, senão calcula, para  $1 \leq i \leq c$ :

$$v_{i,t} = \left( \sum_{k=1}^{n_t} w_k (u'_{ik,t})^m x'_k + \sum_{k=1}^{n_s} (u^s_{ik,t})^m x^s_k \right) / \left( \sum_{k=1}^{n_t} w_k (u'_{ik,t})^m + \sum_{k=1}^{n_s} (u^s_{ik,t})^m \right), \quad (49)$$

Os experimentos foram realizados inicialmente com um conjunto de 43 vetores bi-dimensionais, que formam dois grupos naturais, um com 40 amostras e outro com 3 amostras, demonstrando que o algoritmo proposto pode resolver o problema da equalização de grupos. Também foram realizados experimentos com imagens de ressonância magnética tri-dimensionais.

Assim, uma pequena quantidade de dados de treinamento pode ser usada para influenciar fortemente o processo de agrupamento e, como demonstraram os experimentos, constitui uma boa alternativa de algoritmo de classificação. Além do mais, supera os principais problemas comuns em algoritmos de agrupamento, como mencionado anteriormente. Os resultados dos experimentos também demonstraram que o método dá bons resultados mesmo

quando a escolha dos dados de treinamento é difícil ou imperfeita, pois, mais importante que a escolha de dados de treinamento rotulados, é a escolha do peso para o conjunto de amostras de treinamento de cada classe.

Esta abordagem abre a perspectiva para a obtenção dos dados de treinamento sem supervisão humana, através de uma pré-segmentação e a subsequente extração de alguns *pixels*, os quais irão constituir os dados de treinamento.

3.1.7 KAMEL e SELIM (1994) propõem dois novos algoritmos *fuzzy K-means*, e apresentam uma série de teoremas demonstrando a sua convergência.

Desde que o algoritmo *fuzzy K-means* produz soluções mínimas locais permitindo, porém, soluções mínimas globais (ou próximas) isoladas, é comum aplicar o algoritmo várias vezes, começando cada vez com uma inicialização diferente. Dentre as várias soluções produzidas, a melhor delas é selecionada. Assim, a proposta visa acelerar o algoritmo *fuzzy K-means* de forma que, quando requisitado para múltiplas execuções, resulte em compensações computacionais.

No primeiro algoritmo proposto, os centros de todos os grupos são atualizados tão logo os graus de relacionamento de cada amostra são calculados. Assim, cada vez que os graus de relacionamento de todas as  $n$  amostras são calculados, os centros são atualizados  $n$  vezes.

#### **Algoritmo 1**

**Inicialização.** Estabelece o conjunto inicial de  $c$  centros de grupo e uma matriz de relacionamentos arbitrária  $n \times c$ . Escolhe um escalar  $\varepsilon > 0$ , faz  $k=1$  e  $r=0$ .

**Passo 1.** Faz  $r=r+1$ . Calcula os graus de relacionamento do modelo  $r$ .

Passo 2. Atualiza os centros de grupo. Se  $r < n$  vai para o passo 1.

Passo 3. Se a diferença entre o conjunto de centros atual e o anterior é menor que  $\varepsilon$ , pára, senão faz  $k=k+1$ ,  $r=0$ , e vai para o passo 1.

O algoritmo acima tem dois níveis de iterações. O nível mais externo é indexado por  $k$  e o nível mais interno é indexado por  $r$ . Em cada iteração interna os graus de relacionamento do modelo  $r$  são calculados, e a variável  $r$  é incrementada de um. A cada iteração interna, todos os centros são atualizados usando esses graus. Uma iteração externa é realizada quando os graus de relacionamento de todos os modelos são calculados. A atualização dos centros pode ser simplificada observando-se que a  $r$ -ésima matriz de relacionamentos difere da  $(r-1)$ -ésima apenas na  $r$ -ésima linha, e somente aqueles termos no conjunto de centros envolvendo essa linha precisam ser recalculados, enquanto os outros termos permanecem os mesmos.

A variável  $k$  determina o número de iterações externas, necessárias para a convergência. Embora o algoritmo não mencione,  $k$  pode ser usada para a terminação do algoritmo, pré-fixando-se o número máximo de iterações externas.

No segundo algoritmo, os graus de relacionamento são atualizados tão logo um centro de grupo é calculado.

## Algoritmo 2

Inicialização. Estabelece o conjunto inicial de  $c$  centros de grupo e uma matriz de relacionamentos arbitrária  $n \times c$ . Escolhe um escalar  $\varepsilon > 0$ , faz  $k=1$  e  $r=0$ .

Passo 1. Faz  $r=r+1$ . Calcula o  $r$ -ésimo centro de grupo.

Passo 2. Atualiza a matriz de relacionamentos para todos os modelos. Se  $r < c$  vai para o passo 1.

Passo 3. Se a diferença entre o conjunto de centros atual e o anterior é menor que  $\varepsilon$ , pára, senão faz  $k=k+1$ ,  $r=0$ , e vai para o passo 1.

Como no Algoritmo 1, este algoritmo tem dois níveis de iterações. O nível externo é indexado por  $k$ , e o nível interno é indexado por  $r$ . Em cada iteração interna, o centro de grupo  $r$  é calculado e, em seguida, os graus de relacionamento de todos os modelos são atualizados. Uma iteração externa está terminada quando todos os centros são calculados. O cálculo do  $r$ -ésimo centro de grupo requer os graus de relacionamento  $w_{1r}, w_{2r}, \dots, w_{nr}$ . Estes graus de relacionamento formam a  $r$ -ésima coluna da matriz de relacionamentos. Assim, economia computacional pode ser obtida no passo 1 se calcula-se a  $r$ -ésima coluna da matriz de relacionamentos antes que a matriz inteira.

A variável  $k$ , no Algoritmo 2, conta o número de iterações externas até a convergência e, como mencionado anteriormente, pode ser usada para a terminação algoritmo, desde que o número máximo de iterações seja pré-fixado.

Os algoritmos propostos são analisados e sua convergência é verificada. Um estudo empírico utilizando conjuntos de dados é também realizado a fim de verificar o desempenho dos novos algoritmos em relação ao algoritmo *fuzzy K-means* com distância Euclidiana. Observe-se que nos três algoritmos partiu-se do mesmo agrupamento inicial bem como foram obtidos os mesmos valores para as funções objetivas. Os resultados obtidos empiricamente demonstraram que o primeiro algoritmo requer o menor número de iterações, sendo consistentemente melhor que os outros dois algoritmos. O algoritmo *fuzzy K-means* requer o maior número de iterações.

O custo de uma iteração dos novos algoritmos é maior do que o custo de uma iteração do algoritmo *fuzzy K-means*. Entretanto, o número total de iterações dos novos algoritmos, até a convergência, é menor do que no algoritmo *fuzzy K-means*. Tal fato leva a boas compensações em tempo de CPU total para os novos algoritmos, o que pode ser significativo

quando os algoritmos são requisitados para repetidas execuções com diferentes inicializações, a fim de se obter um resultado mais próximo do mínimo global.

3.1.8. A abordagem de KRISHNAPURAM e MUNSHI (1991) para segmentação de imagens de profundidade combina características diferenciais geométricas como ângulo de orientação e ângulo de inclinação da normal, curvaturas média e Gaussiana, e as coordenadas de cada pixel. A combinação das características ângulo de orientação e curvatura média simultaneamente apresentou os melhores resultados tanto para regiões planares como curvas. Isto porque o ângulo de orientação é constante para regiões planares, e a curvatura média é constante para certos tipos de superfícies curvas, além do que, o ângulo de orientação contém mais informação que o ângulo de inclinação, e a curvatura média é menos sensível a ruído que a curvatura Gaussiana (BESL e JAIN, 1988; YOKOYA e LEVINE, 1989; ABDELMALEK, 1990).

Os algoritmos testados foram o *K-means*, o *fuzzy K-means* e o UFP-ONC (*unsupervised fuzzy partition — optimum number of clusters*), sendo que este último difere do tradicional *fuzzy K-means* por utilizar uma distância baseada na estimação de probabilidade máxima no lugar da distância Euclidiana.

Seja  $q$  um número real maior que 1,  $x_j$  o  $j$ -ésimo vetor característico  $n$ -dimensional,  $v_i$  o centróide do  $i$ -ésimo grupo,  $w_{ij}$  o grau de relacionamento de  $x_j$  no  $i$ -ésimo grupo,  $d^2(x_j, v_i)$  alguma métrica de distância entre  $x_j$  e  $v_i$ ,  $n$  o número de amostras, e  $C$  o número de grupos. O algoritmo UFP-ONC é o seguinte:

Passo 1. Escolhe os centróides iniciais (protótipos) e inicializa todos os relacionamentos para  $1/C$ ;

Passo 2. Enquanto [  $\max ( | w_{ij}^{old} - w_{ij}^{new} | ) > \varepsilon$  ] faça:

a. Atualiza os graus de relacionamento em todos os grupos usando:

$$w_{ij} = ( 1 / d^2(x_j, v_i) )^{1/(q-1)} / \sum_{k=1}^C ( 1 / d^2(x_j, v_k) )^{1/(q-1)}; \quad (50)$$

b. Calcula os novos centróides:

$$v_i = \sum_{j=1}^n (w_{ij})^q x_j / \sum_{j=1}^n (w_{ij})^q; \quad (51)$$

c. Calcula:

$\max ( | w_{ij}^{old} - w_{ij}^{new} | )$ , onde:

$$d^2(x_j, v_i) = ( [\det ( F_i ) ]^{1/2} / P_i(w_{ik}^d)^m) \exp[ (x_j - v_i)^T F_i^{-1}(x_j - v_i)/2], \quad (52)$$

e a matriz de covariância *fuzzy* é:

$$F_i = [ \sum_{j=1}^n w_{ij}^q (x_j - v_i)(x_j - v_i)^T ] / \sum_{j=1}^n w_{ij}^q \quad \text{e} \quad P_i = (1/n) ( \sum_{j=1}^n w_{ij} ). \quad (53) \text{ e } (54)$$

O algoritmo *K-means* apresentou resultados satisfatórios. As medidas utilizadas para se obter o número de grupos foram  $M_{ave}$  e o traço  $t$ . A medida  $M_{ave}$ , que reflete a compactação e separação dos grupos, é assim definida:

$$M_{ave} = \sum_{i=1}^K M[ i ] N_i, \text{ onde:} \quad (55)$$

$$M[ i ] = \min_{j, j \neq i} \{ d^2(v_j, v_i) \} / \{ (1/N_i) \sum_{x_k \in G_i} d^2(x_k, v_i) \}. \quad (56)$$

Aqui,  $d^2$  indica a distância Euclidiana,  $N_i$  é o número de elementos no grupo  $i$ ,  $G_i$  é o conjunto de pontos pertencentes ao grupo  $i$ , e  $v_i$  é o centróide do  $i$ -ésimo grupo. Obviamente,



agrupamentos com grandes valores de  $M_{ave}$  são os preferidos sobre aqueles com valores pequenos.

A medida de desempenho traço  $t$  é dada por:

$$t = \text{trace} [S_w^{-1} S_b], \quad (57)$$

onde  $S_w$  é a matriz intra-grupo e  $S_b$  é a matriz inter-grupo a seguir definidas:

$$S_w = \sum_{i=1}^K \sum_{x_j \in G_i} (x_j - v_i)(x_j - v_i)^T, \text{ e:} \quad (58)$$

$$S_b = \sum_{i=1}^K (v_i - c)(v_i - c)^T N_i, \quad (59)$$

onde  $c$  é o centro de todos os centros de grupos:

$$c = (1/K) \sum_{i=1}^K v_i. \quad (60)$$

Somente a medida traço  $t$  foi satisfatória para a maioria das imagens. Assim, obteve-se o número inicial de grupos  $K_0 = K_i + 1$ , onde  $K_i$  é o número de grupos para o qual o traço  $t$  é máximo.

O algoritmo *fuzzy K-means* com distância Euclidiana desempenhou-se um pouco melhor que o algoritmo *K-means* em todos os casos. O algoritmo UFP-ONC apresentou os melhores resultados, pela capacidade de localizar grupos alongados, o que não acontece com

os outros dois algoritmos, cuja tendência é gerar grupos esféricos e hiperelipsoidais. Nesse caso, o número correto de grupos foi obtido através das medidas hipervolume *fuzzy* e densidade de partição, respectivamente definidas a seguir:

$$F_{HV} = \sum_{i=1}^K [\det(F_i)]^{1/2}, \quad (61)$$

onde  $F_i$  é dado na Equação 53. A densidade de partição é dada por:

$$P_D = (\sum_{j=1}^K S_j) / F_{HV}, \quad (62)$$

onde  $S_i$ , a soma dos membros centrais, é dada por:

$$S_i = \sum w_{ij}, \quad \forall j, \quad \text{tal que:} \quad (63)$$

$$(x_j - v_i)^T F_i^{-1} (x_j - v_i) < 1. \quad (64)$$

O número de grupos selecionado correspondeu ao segundo pico em  $P_D$ , que é quase sempre o mesmo que o segundo declive em  $F_{HV}$ , em todos os casos. Isto corresponde ao critério de definição de bons grupos baseado no número máximo de amostras concentradas próximas ao centróide de cada grupo (densidade máxima), e no hipervolume mínimo dos grupos.

O usuário especifica um limite superior  $K_{\max}$  para o número de grupos. Nem todos os vetores representando as amostras são utilizados nos cálculos, simplificando o conjunto de amostras. Foi usada uma frequência de amostragem de 3, isto é, cada terceira linha e terceira coluna do conjunto de amostras foram usadas. O método gera uma seqüência de agrupamentos de 2 até  $K_{\max}$  grupos. Para cada valor de  $K$ , as medidas de desempenho são calculadas após convergência, sendo selecionado o número de grupos com base nestas medidas. Finalmente, os vetores característicos que não foram usados no processo de agrupamento são simplesmente atribuídos ao centro de grupo mais próximo. A razão de amostragem usada deu resultados quase que indistinguíveis daqueles quando todos os vetores característicos são usados no processo de agrupamento, e ainda com economia de tempo computacional.

O método apresentado constitui uma grande contribuição por apresentar um caminho para se determinar o número adequado de grupos, bem como detectar grupos com formas e tamanhos variados.

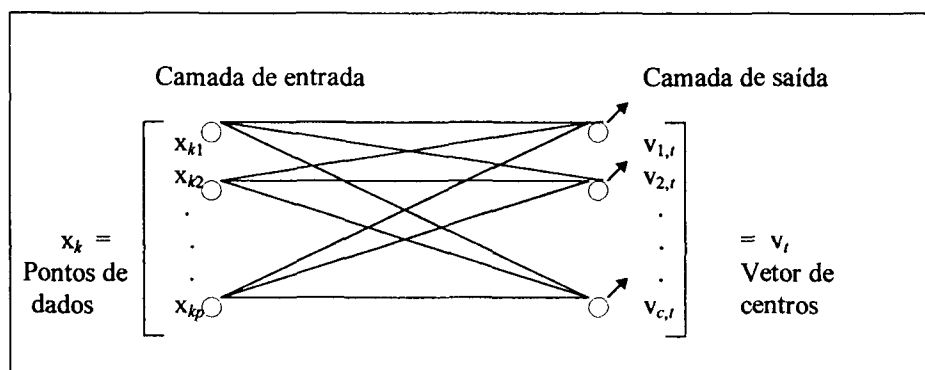
3.1.9 TSAO, BEZDEK e PAL (1994) integram o algoritmo *fuzzy K-means* às redes de agrupamento Kohonen, a fim de otimizar o problema do agrupamento na busca pelo melhor conjunto de centros de grupo.

Redes de agrupamento Kohonen (KCNs *Kohonen clustering networks*) são esquemas não-supervisionados que encontram o melhor conjunto de centros para agrupamentos rígidos, de forma seqüencial e iterativa. A estrutura de KCN consiste de duas camadas: uma camada de entrada e uma camada de saída, como mostrado na Figura 8.

Cada nodo de saída tem um protótipo ou vetor de centros ligado a ele, e é este vetor que é ajustado durante o aprendizado. Dado um vetor de entrada, os neurônios na camada de

saída competem entre si e o vencedor (aquele cujo centro tem a distância mínima da entrada) atualiza seus centros e aqueles de algum conjunto de vizinhos pré-definido. O processo continua até que os vetores de centros estabilizam.

FIGURA 8 - ESTRUTURA DE UM AGRUPAMENTO KOHONEN



FONTE: TSAO, BEZDEK e PAL (1994)

O algoritmo KCN pode ser brevemente descrito como segue:

Passo 1. Fixa o número de grupos  $c$ ,  $\| \cdot \|$ , e o parâmetro  $\varepsilon > 0$ ;

Passo 2. Inicializa  $v_0 = (v_{1,0}, v_{2,0}, \dots, v_{c,0}) \in \mathbb{R}^{cp}$ , taxa de aprendizado  $\alpha_{ik,0} \in (1,0)$  e atualiza vizinhança  $N_0 =$  um conjunto de índices em  $\{1, 2, \dots, c\}$ ;

Passo 3. Para  $t=1, 2, \dots, t_{\max}$ ; para  $k=1, 2, \dots, n$ :

a. Calcula  $d_{ik}^2 = \|x_k - v_{i,t}\|^2$ , para  $i=1$  até  $c$ ;

b. Ordena  $\{d_{ik}^2\}$  em ordem ascendente:  $d_{1k}^2 \leq d_{2k}^2 \leq \dots \leq d_{ck}^2$ ;

c. Atualiza o vencedor:

$$v_{1,t} = v_{1,t-1} + \alpha_{1k,t} (x_k - v_{1,t-1}). \quad (65)$$

d. Se  $|N_t| = n_t$ , atualiza os  $(n_t - 1)$  nodos que são mais próximos de  $x_k$  em:

$$v_{i,t} = v_{i,t-1} + \alpha_{ik,t} (x_k - v_{i,t-1}), \quad i = 2, 3, \dots, (n_t - 1). \quad (66)$$

e. Próximo  $k$ .

Passo 4. Calcula:

$$E_t = \|v_t - v_{t-1}\|^2 = \sum_i \|v_{i,t} - v_{i,t-1}\|^2; \quad (67)$$

Passo 5. Se  $E_t \leq \varepsilon$  pára, senão ajusta  $\{\alpha_{ik,t}\}$ , atualiza vizinhança  $N_t$  e próximo  $t$ .

A integração do algoritmo *fuzzy K-means* e das redes de agrupamento Kohonen KCN resulta no algoritmo FKCN seguinte:

Passo 1. Fixa  $c$ ,  $\|\cdot\|$ , e o parâmetro  $\varepsilon > 0$ ;

Passo 2. Inicializa  $v_0 = (v_{1,0}, v_{2,0}, \dots, v_{c,0})$ . Escolhe  $m_0 > 1$  e  $t_{\max}$  = limite de iterações;

Passo 3. Para  $t=1, 2, \dots, t_{\max}$ :

a. Calcula todos ( $cn$ ) taxas de aprendizado  $\{\alpha_{ik,t}\}$  com:

$$\alpha_{ik,t} = (u_{ik,t})^{m_t}; m_t = (m_0 - 1) / t_{\max}, \quad (68)$$

$m_0$  uma constante maior que 1, com:

$$u_{ik} = \left( \sum (\|x_k - v_i\|_A / \|x_k - v_j\|_A)^{2/(m-1)} \right)^{-1}, \text{ para todo } i, k; \quad (69)$$

b. Atualiza todos os ( $c$ ) vetores de centróides  $\{v_{i,t}\}$  com:

$$v_{i,t} = v_{i,t-1} + \left[ \frac{\sum_{k=1}^n \alpha_{ik,t} (x_k - v_{i,t-1})}{\sum_{s=1}^n \alpha_{is,t}} \right]. \quad (70)$$

c. Calcula:

$$E_t = \|v_t - v_{t-1}\|^2 = \sum_i \|v_{i,t} - v_{i,t-1}\|^2; \quad (71)$$

d. Se  $E_t \leq \varepsilon$  pára, senão próximo  $t$ .

A combinação das idéias de valores de relacionamento *fuzzy* para taxas de aprendizado, do paralelismo do *fuzzy K-means*, e a estrutura e regras de atualização das redes de agrupamento Kohonen KCNs, revela diversas vantagens.

Uma delas se refere à auto-organização do algoritmo FKCN, desde que o tamanho da vizinhança de atualização é ajustado durante o aprendizado. No algoritmo KCN, a escolha do tamanho da vizinhança, da taxa de aprendizado e das estratégias para operar estes parâmetros é crítica para a sua terminação (quando satisfaz  $E_t \leq \varepsilon$ ), pois diferentes conjuntos de parâmetros produzem resultados diferentes. A terminação do algoritmo KCN ocorre no limite do número de iterações, a menos que  $\alpha$  seja forçado para zero, o que é uma estratégia de terminação

artificial. Já a terminação do algoritmo combinado FKCN ocorre antes que se atinja o número máximo de iterações.

Outra vantagem é que o algoritmo FKCN não é seqüencial, por que as atualizações dos centros são realizadas depois de cada passada através de  $X$ . Assim, a seqüência de centros calculada é independente dos rótulos de dados, como no algoritmo *fuzzy K-means*, e ao contrário do algoritmo KCN.

Quanto à complexidade, o algoritmo KCN tem  $O(t^*n)$ , onde  $t^*$  é o número de iterações e  $n$  o número de dados de entrada. Como o algoritmo FKCN não é seqüencial, sua complexidade é  $O(t^*)$ , independente do número de dados de entrada.

O trabalho apresenta ainda uma prova de que o algoritmo FKCN é equivalente ao *fuzzy K-means* quando  $m$  é fixo, e ao *hard K-means* quando  $m=1$ .

Assim, o algoritmo FKCN tem a estrutura de auto-organização do algoritmo KCN e, ao mesmo tempo, está num grau ótimo com relação ao bem conhecido e amplamente usado modelo de agrupamento *fuzzy*, como puderam com provar os experimentos realizados pelos autores, através de um conjunto de dados.

3.1.10 BABU e MURTY (1994) apresentam um trabalho centrado no problema do agrupamento ótimo, explorando a aplicabilidade de abordagens de otimização estocástica para otimizar funções objetivas de agrupamento.

Estratégias de evolução (EES *evolution strategies*) são ferramentas poderosas de otimização de função e podem ser aplicadas a uma grande variedade de problemas em diversos campos. Estes métodos adotam princípios do processo evolutivo: seleção, recombinação e mutação, e pertencem a uma classe de abordagens baseadas em população.

Aqui, população consiste de um conjunto de pais,  $U$ , e um conjunto de filhos  $\Theta$ . Cada pai ou filho representa uma solução no espaço de busca e é denotado por uma *string* de parâmetros. Filhos na população são produzidos por operações de mutação e recombinação sobre os pais na mesma população. Pais na próxima população são gerados selecionando soluções potenciais na população prévia. O processo de desenvolvimento da próxima população a partir da população corrente é chamado geração. Este processo de geração de nova população termina quando soluções ótimas ou quase ótimas são obtidas, ou quando se alcança um limite no número de gerações.

No modelo de estratégias de evolução multi-associada utilizado nesse trabalho, mais de um pai participa na geração de filhos, ou seja,  $|U| = \mu > 1$ . Nesse modelo, o operador de recombinação é introduzido, no qual dois pais são selecionados e suas características são combinadas e alteradas para produzir um filho. Nesse mesmo contexto, o modelo  $(\mu + \lambda)$ -ES usa todos os indivíduos no processo de seleção de pais para a próxima população, enquanto que o modelo  $(\mu, \lambda)$ -ES usa somente filhos  $\Theta$  no processo. Assume-se que  $\mu < \lambda$  e cada pai contribui na geração de  $(\lambda/\mu)$  filhos em uma população. A descrição formal de estratégias de evolução multi-associada é dada a seguir.

Pais na população inicial são selecionados aleatoriamente com distribuição uniforme e filhos nessa população são produzidos usando operadores de recombinação e mutação. Pais na próxima população são selecionados da população prévia dependendo da estratégia usada (“+” ou “,”).

Na operação de recombinação  $R(I', I'') \rightarrow I$ , dois indivíduos  $I'$  e  $I''$  são selecionados com igual probabilidade do conjunto de pais  $U$ . Cada indivíduo compõe-se de um vetor solução e um vetor variância:

$$\begin{aligned}
 I' &= (x', \sigma'), \\
 I'' &= (x'', \sigma''), \\
 I &= (x, \sigma).
 \end{aligned}
 \tag{72}$$

Os operadores de recombinação utilizados nesse estudo foram:

Recombinação discreta  $R_1$ :

$$\begin{aligned}
 x_i &= x'_i \text{ or } x''_i, \\
 \sigma_i &= \sigma'_i \text{ or } \sigma''_i, \forall i.
 \end{aligned}
 \tag{73}$$

Recombinação intermediária  $R_2$ :

$$\begin{aligned}
 x_i &= (1/2) (x'_i + x''_i), \\
 \sigma_i &= (1/2) (\sigma'_i + \sigma''_i), \forall i.
 \end{aligned}
 \tag{74}$$

O operador de mutação  $M(I) \rightarrow I'$ , toma um indivíduo e os seguintes passos são executados para produzir um indivíduo alterado:

$$\begin{aligned}
 I &= (x, \sigma), \\
 I' &= (x', \sigma'), \\
 \sigma'_i &= \sigma_i \exp(N_0(0, \Delta\sigma)), \\
 x'_i &= x_i + N_0(0, \sigma'_i), \forall i,
 \end{aligned}
 \tag{75}$$

onde  $N_0(0, \sigma)$  gera valores aleatórios normalmente distribuídos com média zero e variância  $\sigma$ .



O operador de seleção  $S(\Omega) \rightarrow U$  seleciona os  $\mu$  indivíduos do conjunto  $\Omega$  para produzir pais na próxima geração. O conjunto  $\Omega$  é  $U \cup \Theta$  para  $(\mu + \lambda)$ -ES e é  $\Theta$  para  $(\mu, \lambda)$ -ES.

A seguir, o algoritmo para estratégias de evolução multi-associada:

**Input:**

$U$  = conjunto inicial de pais,

$\mu$  = número de pais,  $\lambda$  = número de filhos,

max\_gen = número máximo de gerações;

**Output:** string solução  $S$ ;

begin

$s = 0$ ;

no-generation = Max\_gen,

calcula valores de ajuste ( $f$ ) de soluções em  $U$ ,

while (no-generation > 0)

begin

$\Theta = \text{NIL}$ ;

while (  $|\Theta| < \lambda$  )

begin

$x = \text{Rand}(1, \mu)$ ;  $y = \text{Rand}(1, \mu)$ ;

seleciona  $I_x$  e  $I_y$  de  $U$ ;

$I' = M(R(I_x, I_y))$ ;

$\Theta = \Theta \cup \{I'\}$ ;

calcula  $f(I')$ ;

end

se ( $f(s) >$  valor de ajuste do melhor indivíduo) troque  $S$  por aquela string,

se ( $f(s)$  é ótima ou quase ótima) a saída é  $S$  e sai,

senão no-generation = no-generation - 1,

$U = S(U \cup \Theta)$ ; /\* se o operador de seleção "+" é usado \*/

end

end.

Nesse estudo, estratégias de evolução são usadas para otimizar a função objetiva soma de erro quadrático para agrupamento “hard”, e a função objetiva *fuzzy K-means* para agrupamento “fuzzy”. A função de avaliação  $f$  deve ser definida em cada caso.

Seja  $x$  uma solução representada por um vetor de números reais  $x_1 x_2 \dots x_n$  e  $n=Cxd$ , onde  $d$  é o número de dimensões e  $C$  o número de grupos. A função de avaliação para agrupamento “hard”  $f_h(x)$  é calculada da maneira seguinte:

Passo 1. Faz:

$$c_i(l) = x((i-1) * d + l), 1 \leq i \leq C, 1 \leq l \leq d, \quad (76)$$

Passo 2. Atribui dados  $v_1, v_2, \dots, v_T$  ao centro mais próximo, e obtém a matriz de associação  $W$ ,

Passo 3. Recalcula centros  $c_1, c_2, \dots, c_C$ ,

Passo 4. Calcula erro  $J()$ ,

Passo 5. Retorna valor de  $J()$ .

No caso da função objetiva de agrupamento *fuzzy K-means*, os passos para o cálculo de  $f_f(x)$  são os seguintes:

Passo 1. Faz:

$$c_i(l) = x((i-1) * d + l), 1 \leq i \leq C, 1 \leq l \leq d, \quad (77)$$

Passo 2. Calcula medidas de associação  $u_{ij}$ ,

Passo 3. Recalcula centros  $c_1, c_2, \dots, c_C$ ,

Passo 4. Calcula erro  $J()$ ,

Passo 5. Retorna valor de  $J()$ .

No estudo experimental realizado, foram testados muitos conjuntos de dados com diferentes números de grupos. Em quase todos os casos, o modelo proposto encontrou uma

partição dos dados ótima ou próxima da ótima. Observou-se que a média de convergência inicial é muito alta, e com o progresso das gerações a média de convergência diminui rapidamente.

Os resultados deste trabalho são encorajadores, pois enfrentam um dos principais problemas dos algoritmos de agrupamento *K-means* e *fuzzy K-means*, que refere-se à solução mínima local. Embora o modelo proposto seja dispendioso com relação a tempo de execução, o trabalho apresenta um modelo de *hardware* paralelo, onde mais de um processador é utilizado, a fim de se obter maior velocidade.

3.1.11 SELIM e AL-SULTAN (1991) discutem a solução do problema do agrupamento, usualmente resolvido pelo algoritmo *K-means*, e que é conhecido por apresentar muitas soluções mínimas locais. A abordagem *simulated annealing* para resolver problemas de otimização é descrita e proposta para resolver o problema do agrupamento.

*Annealing* refere-se ao processo de aquecimento de um sólido a uma alta temperatura seguido por um lento resfriamento causado pela diminuição da temperatura do ambiente em etapas. A cada passo a temperatura é mantida constante por um período de tempo suficiente para o sólido alcançar o equilíbrio térmico. No equilíbrio o sólido pode ter muitas configurações, cada qual correspondendo a diferentes *spins* de seus elétrons e a um nível de energia específico.

No método para simular o processo de alcance de equilíbrio térmico (*simulated annealing*) num valor fixo de temperatura  $T$ , uma perturbação da configuração corrente do sólido – gerada aleatoriamente – é aplicada tal que uma configuração teste é obtida. Se o nível de energia da configuração teste é menor que o nível de energia da configuração corrente,

então a configuração teste é aceita e passa a ser a configuração corrente. Uma configuração com nível de energia maior que o da configuração corrente poderá ser aceita com uma determinada probabilidade. O processo continua até que o equilíbrio térmico é alcançado, depois de um número grande de perturbações.

QUADRO 1 - *SIMULATED ANNEALING* X PROBLEMA DO AGRUPAMENTO

<i>Simulated annealing</i>	O problema do agrupamento
1. O sólido sendo aquecido	1. O problema da otimização
2. A configuração corrente	2. A configuração de agrupamento corrente
3. A perturbação da configuração corrente	3. A geração de uma nova configuração
4. A configuração teste	4. A configuração teste
5. A energia associada a uma configuração	5. A função J associada a um agrupamento
6. O aceite de uma nova configuração se seu nível de energia é menor que o nível de energia corrente	6. O aceite de uma nova configuração que resulte em um melhoramento do valor da função J
7. O aceite de uma configuração com nível de energia mais alto, com alguma probabilidade	7. O aceite de uma configuração de agrupamento com um valor maior de J, esperando que um agrupamento melhor seja alcançável

A seguir, a descrição do algoritmo *simulated annealing* para resolver o problema do agrupamento. Os subscritos  $c$  e  $t$  denotam os agrupamentos corrente e teste, respectivamente, enquanto que o subscrito  $b$  denota o agrupamento que possui o menor valor para a função  $J$ .

**Inicialização.** Inicializa  $T_1$ ,  $\varepsilon$  e  $\mu$  com valores pré-especificados correspondendo a temperatura inicial, temperatura final e multiplicador de temperatura, respectivamente. Inicia com um agrupamento arbitrário das amostras em  $C$  grupos. Os vetores  $A_b=A_c$  denotam essa atribuição. Calcula o valor da correspondente função critério e atribui  $J_b=J_c$  para esse valor. Faz  $T=T_1$ .

**Passo 1.** Obtém um agrupamento teste  $A_t$ . Seja  $J_t$  o correspondente valor da função critério.

**Passo 2.** Se  $J_t > J_c$  vai para o passo 3, caso contrário aceita esse agrupamento teste, fazendo  $A_c=A_t$ , e  $J_c=J_t$ . Se  $J_t \geq J_b$  incrementa *count* de 1 e vai para o passo 4. Caso contrário, faz  $J_b=J_t$ ,  $A_b=A_t$ , *count*=0 e vai para o passo 4.

**Passo 3.** Obtém um número aleatório  $y \sim U(0,1)$ . Se  $y > \exp(-(J_t - J_c)/T)$  vai para o passo 4 (agrupamento teste não aceito); caso contrário, faz  $J_c=J_t$ ,  $A_c=A_t$  e vai para o passo 4.

Passo 4. Se  $count < N$  vai para o passo 1. Caso contrário, troca  $T$  por  $\mu T$ . Se  $T < \varepsilon$  pára, caso contrário vai para o passo 1.

Este algoritmo requer os seguintes parâmetros: a temperatura inicial  $T_1$ , o multiplicador de temperatura  $\mu$ , o limiar de probabilidade  $P$ , e o número de iterações  $N$  para alcançar o equilíbrio de uma dada temperatura.

Se  $J_i > J_c$  o agrupamento teste é aceito com probabilidade  $\exp(-(J_i - J_c)/T)$ . Se  $T$  é grande, a probabilidade de aceitação será grande. Inicialmente, é interessante aceitar agrupamentos com valor de função alto para explorar um número grande de agrupamentos. Assim, a temperatura inicial é escolhida tão grande quanto possível.

Se a melhor função  $J_b$  não melhora para um certo número de iterações, significa que o equilíbrio foi alcançado para esta temperatura. A temperatura deve ser diminuída. Isto é obtido multiplicando a temperatura por  $\mu$ , onde  $0 < \mu < 1$ .

O limiar de probabilidade  $P$  é usado no algoritmo para geração de um agrupamento teste. A ele devem ser atribuídos os valores  $P_1$  e  $P_2$ , sendo  $0 < P_1 < P_2 < 1$ .  $P = P_1$  é usado nas iterações iniciais, para se obter grandes números de trocas nos agrupamentos. Depois de um certo número de iterações,  $P_2$  deverá substituir  $P_1$  a fim de fazer os agrupamentos teste gerados mais próximos dos agrupamentos correntes.

Se durante as  $N$  últimas iterações em uma temperatura fixa, o valor de  $J_b$  não melhora, assume-se que o equilíbrio foi alcançado para este nível de temperatura. Assim, na próxima iteração a temperatura é reduzida multiplicando a temperatura corrente por  $\mu$ .

A respeito da convergência do processo *simulated annealing*, o trabalho em questão descreve o teorema que afirma que o algoritmo *simulated annealing* alcança a solução ótima, sob certas condições.

Um agrupamento teste é de alguma forma próximo ou relacionado ao agrupamento corrente. O procedimento usado para geração de um agrupamento teste segue abaixo:

Passo 1. Inicia com a primeira amostra, faz  $i=1$  e  $flag=false$ .

Passo 2. Obtém um número aleatório  $u \sim U(0,1)$ . Se  $u > P$  (um limiar de probabilidade pré-especificado) faz  $flag=true$  e vai para o passo 4. Caso contrário, mantém a amostra  $i$  atribuída a seu grupo corrente  $a_i$ .

Passo 3. Se  $i=n$  e  $flag=true$  pára. Caso contrário, troca  $i$  por  $i+1$  e vai para o passo 2.

Passo 4. Seja  $S_i = \{l \mid 1, 2, \dots, C, l \neq a_i\}$ . Gera aleatoriamente um elemento  $l$  do conjunto  $S_i$ . Atribui a amostra  $i$  para o grupo  $l$ , isto é, faz  $a_i=l$ . Vai para o passo 3.

O algoritmo acima irá parar se todas as amostras forem consideradas e ao menos uma tenha sido atribuída a um novo grupo. A proximidade entre um agrupamento e seu vizinho é controlada pelo limiar de probabilidade  $P$ .

Experimentos realizados com conjuntos arbitrários de dados levam a algumas considerações:

- A escolha de  $\mu$  depende da extensão do problema; geralmente,  $0,7 \leq \mu \leq 0,9$ ;
- Quanto ao limiar de probabilidade, este deve ser grande o suficiente para gerar uma configuração de agrupamento próxima, mas não muito próxima da configuração corrente; recomenda-se  $P=0,95$ ;
- Quanto maior o número de iterações  $N$ , melhor o resultado, porém também maior o tempo gasto para resolver o problema;  $N$  depende da extensão do problema, sendo recomendado um valor no intervalo  $[50,600]$ ;
- A temperatura inicial depende da magnitude da função objetiva; quanto maior a magnitude da função objetiva, maior deve ser a temperatura inicial; recomenda-se  $T_1=10$ .

Neste trabalho, faz-se referência a um teorema que assegura a convergência do algoritmo *simulated annealing* à solução ótima global, para um problema sob certas condições, bem como apresenta-se a prova de que o algoritmo proposto satisfaz tais condições e, portanto, converge para o mínimo global. Entretanto, não é apresentada nenhuma relação entre o número de iterações e a qualidade da solução produzida, isto é, não há como determinar o ponto de parada do algoritmo, no qual se atinje o mínimo global.

Ainda assim, o algoritmo apresentado permite que se alcance uma solução mínima local que se aproxima da solução mínima global, ficando a qualidade da solução obtida vinculada à habilidade do usuário de escolher o número de iterações para executar o algoritmo. Outra vantagem deste algoritmo é que o mesmo não requer capacidade computacional de memória grande.

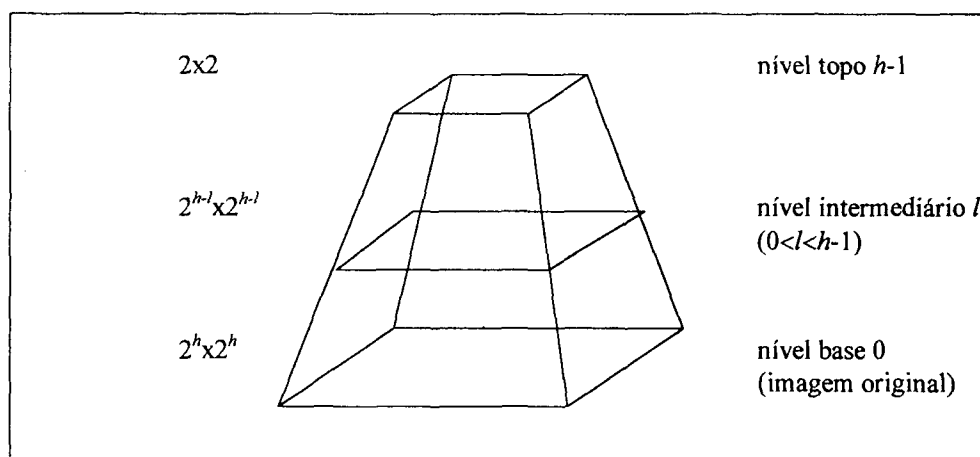
## 3.2 BASEADAS EM MÉTODOS HIERÁRQUICOS

3.2.1 O trabalho de SABATA, ARMAN e AGGARWAL (1993) é uma proposta de segmentação de imagens de profundidade que utiliza, em um de seus módulos, um algoritmo de segmentação piramidal. No primeiro módulo são utilizadas propriedades locais dos dados para realizar o agrupamento piramidal, tendo como resultado uma super-segmentação. No segundo módulo, um procedimento une regiões baseado num certo critério de homogeneidade, o qual depende da representação final da superfície, definida por informações de alto nível.

O algoritmo de agrupamento utilizado no primeiro módulo está baseado em sete características da imagem: a normal à superfície em cada ponto, e suas seis projeções sobre os planos  $xy$ ,  $xz$  e  $yz$  em cada uma das duas direções possíveis.

Será abordado aqui basicamente o algoritmo piramidal utilizado no primeiro módulo. O agrupamento piramidal baseia-se em estruturas chamadas pirâmides, que são estruturas de dados hierárquicas com uma matriz retangular de nodos em uma seqüência de níveis da base para o topo. O agrupamento piramidal é realizado em três estágios. No primeiro ocorre a inicialização, onde os nodos da estrutura de dados piramidal são inicializados. A imagem de entrada é armazenada na base de uma pirâmide na qual cada *pixel* da imagem representa um nodo.

FIGURA 9 - ESTRUTURA DE UMA PIRÂMIDE. O NÍVEL É DADO À DIREITA, E O TAMANHO DO NÍVEL EM *PIXELS* É DADO À ESQUERDA



FONTE : SABATA, ARMAN e AGGARWAL (1993)

Assumindo que a imagem original é de tamanho  $2^h \times 2^h$ , o nível zero (base) é de tamanho  $2^h \times 2^h$ . Os valores dos nodos em cada nível são obtidos pela média dos valores dos nodos vizinhos do nível abaixo, de uma área  $2(\text{span}) \times 2(\text{span})$ ,  $\text{span} \geq 1$ , o que determina o número de nodos designados como filhos válidos. A suposição de que o tamanho da imagem é  $2^h \times 2^h$  não reduz a generalidade dos tipos de imagens, porque qualquer imagem de tamanho  $W \times H$  pode ser estendida com "NULLS" para se ter  $2^h \times 2^h$ . Aqui  $h$  é o menor inteiro tal que



$2^h \geq \max(W, H)$ . Se  $p[i, j, l]$  é a propriedade do nodo  $n$  no nível  $l$ , e assumindo  $\text{span}=2$ , a inicialização é descrita como:

$$p[i, j, l] = \left[ \sum_{\substack{[i', j', l-1] \in X_{l-1} \\ [i', j', l-1] \in X_{l-1}}} p[i', j', l-1] w_{sn} \right] / \left[ \sum_{[i', j', l-1] \in X_{l-1}} w_{sn} \right], \forall [i, j, l] \in X_l, \quad (78)$$

onde  $w_{sn} = w[i', j', i, j, l]$  é a função de relacionamento para filhos válidos  $s$  (referenciado como  $[i', j', l-1]$ ), do nodo  $n$ . O nodo  $[i, j, l]$  é usado na determinação dos valores dos nodos localizados em:

$$\begin{aligned} & \lfloor (i-1)/2 \rfloor, \lfloor (j-1)/2 \rfloor, l+1, \\ & \lfloor (i-1)/2 \rfloor, \lfloor (j+1)/2 \rfloor, l+1, \\ & \lfloor (i+1)/2 \rfloor, \lfloor (j-1)/2 \rfloor, l+1, \\ & \lfloor (i+1)/2 \rfloor, \lfloor (j+1)/2 \rfloor, l+1, \end{aligned} \quad (79)$$

onde  $\lfloor x \rfloor$  designa a parte inteira de  $x$ . Estes quatro nodos são pais do nodo  $[i, j, l]$ .

Considerando  $\{father [i, j, l]\}$  o conjunto dos pais válidos do nodo  $[i, j, l]$ , tem-se:

$$w_{nf} = w([i, j, i', j', l+1]) = \begin{cases} 1 & \text{se } [i', j', l+1] \in \{father [i, j, l]\} \\ 0 & \text{caso contrário.} \end{cases} \quad (80)$$

No segundo estágio do agrupamento piramidal ocorre a ligação (*linking*) de nodos, que é um processo iterativo no qual cada nodo escolhe seu melhor pai baseado numa medida de proximidade. Tal medida exprime uma propriedade entre um nodo no nível  $l$  e um nodo no nível  $l+1$ :

$$d_{nf} = d([i, j, l], [i', j', l+1]) = \| p [i, j, l] - p [i', j', l+1] \|^2, \quad (81)$$

onde  $\| \cdot \|$  é a norma.

Um peso  $u_{nf}$  entre um nodo  $n$  e seu pai candidato  $f$  é então designado como:

$$u_{nf} = u([i, j, l], [i', j', l+1]) = (w_{nf} / d_{nf}) / \sum_{p \in X_{l+1}} (w_{np} / d_{np}). \quad (82)$$

Note que:

$$\sum_{f \in X_{l+1}} u_{nf} = 1, n \in X_l. \quad (83)$$

Após calculados os pesos entre cada nodo e seu pai, o valor  $p$  é recalculado para cada nodo (no nível  $l$ ):

$$p_n = p [i, j, l] = \left( \sum_{s \in X_{l-1}} (u^2)_{sn} \cdot p [i', j', l-1] \right) / \left( \sum_{s \in X_{l-1}} (u^2)_{sn} \right), \quad (84)$$

onde  $u_{sn}$  é o peso entre o nodo  $n$  e seu filho  $s$ . Para os casos em que  $d_{nf} = 0$ , o peso entre cada nodo e seu pai  $f$  é um e os outros três pais obtêm peso igual a zero.

Em cada nível,  $J_2$  pode ser definida:

$$J_2 = \sum_{n \in X_l} \sum_{f \in X_{l+1}} u_{nf}^2 d_{nf} w_{nf}. \quad (85)$$

Para cada nível  $l$  acima da base, os pesos  $u$  e os valores  $p[i,j,l]$  dos nodos são atualizados em uma iteração até que  $J_2$  alcança um ponto fixo (teste de convergência):

$$(J_2)^{(k)} - (J_2)^{(k+1)} < \delta, \quad (86)$$

onde  $(J_2)^{(k)}$  é o valor de  $(J_2)$  depois da  $k$ -ésima iteração.

Este processo continua em todos os níveis até alcançar o topo da pirâmide.

O algoritmo é geral para qualquer tipo de dados de entrada, trocando somente a definição da norma.

O terceiro estágio do algoritmo de agrupamento piramidal é a geração da árvore, no qual são utilizados os resultados do estágio de ligação para assinalar um rótulo de região para cada nodo. Todos os nodos com o mesmo rótulo estão na mesma região. Começando de um nível  $H \leq h-1$ , um rótulo distinto é designado para todos os nodos com valores de propriedade distintos. Então aos nodos no nível  $H-1$  são designados os rótulos dos seus pais escolhidos (o pai com o maior peso). Este processo é repetido para todos os níveis abaixo, cada filho sendo rotulado com o rótulo de seu pai escolhido. No final deste estágio, os nodos no nível base são rotulados com um dos rótulos do nível  $H$  escolhido. O menor número de possíveis rótulos é quando  $H=h-1$  (assumindo que o nível zero tem  $2^h \times 2^h$  nodos). Neste caso, quatro rótulos são gerados.

No segundo módulo, duas estratégias de *merging* foram examinadas, considerando que os retalhos de superfície são representados usando polinomiais bivariadas. Na primeira, os pontos de cada segmento sob consideração são usados para derivar uma função de aproximação. A função é testada para os segmentos vizinhos; o vizinho que melhor ajusta a função é escolhido para união. No segundo método, os dados de cada segmento vizinho são

usados para derivar uma função de aproximação. Uma segunda função é derivada assumindo que o vizinho foi unido com o segmento sob consideração; as duas funções são comparadas e, se similares, os dois segmentos são unidos.

Os experimentos realizados em várias imagens de profundidade demonstraram resultados bons, não apresentando restrições quanto ao tipo de superfície ou tamanho dos objetos na cena. Além disso, apresenta ainda a vantagem de uma dependência mínima de limiares determinados empiricamente, o que não ocorre em muitos algoritmos de segmentação. Deve ser considerado o custo elevado do algoritmo, o que pode ser resolvido através da implementação em paralelo.

3.2.2 Outra abordagem hierárquica é dada por HATTORI e TORII (1993), sendo apresentados dois algoritmos de agrupamento aglomerativo hierárquico.

Considerando o problema do agrupamento, no qual um conjunto de dados com  $N$  objetos para classificar em  $n$  grupos, e  $d_{ij}$  a distância entre os objetos  $A_i$  e  $A_j$ ,  $N$  grupos estão presentes no início. Os grupos vizinhos mais próximos são fundidos sob a definição de distância entre dois grupos, diminuindo o número de grupos. A fusão é realizada até que o número de grupos fique igual a  $n$ .

Seja  $d_{pq}$  a distância entre dois grupos  $C_p$  e  $C_q$ , que no método do vizinho mais próximo é definida como a distância mínima de todas as distâncias  $d_{ij}$ , para  $A_i \in C_p$  e  $A_j \in C_q$ :

$$d(p, q) = \min_{A_i \in C_p, A_j \in C_q} d_{ij}. \quad (87)$$

A proposta apresentada não trata as distâncias  $d_{ij}$  diretamente, mas um coeficiente de similaridade entre dois objetos  $A_i$  e  $A_j$ :

$$s_{ij} = \begin{cases} 1 - (d_{ij} / C\sigma), & \text{para } d_{ij} \leq C\sigma \\ 0, & \text{para } d_{ij} \geq C\sigma \end{cases} \quad (88)$$

onde  $C$  é um parâmetro fixado e adequadamente determinado, e  $\sigma$  o desvio padrão de todas as distâncias  $d_{ij}$  no conjunto de dados. O conjunto de todos os elementos  $s_{ij}$  constitui a matriz de similaridade  $S$ .

Considerando  $h$  um parâmetro característico para classificação de  $N$  objetos,  $A_i$  é ligado diretamente com objeto  $A_j$ , se  $s_{ij} \geq h$ . Caso  $s_{ij} < h$ ,  $A_i$  não é ligado diretamente com objeto  $A_j$ . Um grupo é formado por todos os objetos os quais são ligados direta ou indiretamente via outros objetos. Os objetos que não são conectados diretamente nem indiretamente pertencem a grupos diferentes.

Para  $h=1$ , cada objeto constitui um grupo, e  $N$  grupos estão presentes. Para  $h=0$ , somente um grupo está presente, ao qual pertencem todos os objetos. Portanto, o número de grupos é determinado através de  $h$ .

Quanto ao parâmetro  $C$ , este deve ser suficientemente grande para classificar o conjunto de dados no número específico de grupos. Se muito pequeno,  $s_{ij}$  é zero para muitos pares de objetos porque ele não é zero para  $d_{ij}$  muito pequeno. Para um  $C$  muito pequeno, muitas amostras não serão unidas em grupos, e cada uma delas constituirá um grupo.

No algoritmo seguinte, para classificar um conjunto de  $N$  dados em  $n$  grupos, todas as distâncias  $d_{ij}$  dos  $N$  objetos entre si já estão armazenadas em uma matriz de distâncias. Se os

objetos  $A_i$  e  $A_j$  pertencem (ou não) ao mesmo grupo é equívale a dizer que as variáveis  $P(A_i)$  e  $P(A_j)$  apontam (ou não) para o mesmo endereço.

- Passo 1. Calcula o desvio padrão  $\sigma$  de todas as distâncias. Determina adequadamente o valor de  $C$ . Constrói a matriz de similaridade  $S$ .
- Passo 2. Sejam  $h$  e  $\Delta h$  igual  $1/2$ . Prepara  $N$  diferentes endereços  $D(i)$ ,  $i=1, \dots, N$ . Prepara  $N$  variáveis apontadoras  $P(A_i)$ ,  $i=1, \dots, N$ .
- Passo 3. Faz cada  $P(A_i)$  apontar para o endereço  $D(i)$ ,  $i=1, \dots, N$ .
- Passo 4. Repete os seguintes procedimentos para todos os possíveis pares de  $(i,j)$  sob a condição de  $1 \leq i < j \leq N$ . Toma um par de  $(i,j)$ . Se  $s_{ij} \geq h$ , e o endereço apontado por  $P(A_i)$  é diferente do apontado por  $P(A_j)$ , busca por todas as variáveis ponteiro apontando para o endereço apontado por  $P(A_j)$ . Faz  $P(A_j)$  e todas estas variáveis ponteiro apontarem para  $P(A_i)$ .
- Passo 5. Repete o seguinte procedimento de  $i=1, \dots, N$ . Busca por  $P(A_i)$  que aponta o endereço  $D(i)$ , contando o número dessas variáveis ponteiros, e expressa-o como  $n(h)$ .
- Passo 6. Se  $n(h) > n$ , troca  $h$  por  $h - \Delta h/2$ , troca  $\Delta h$  por  $\Delta h/2$ , e retorna ao passo 3. Se  $n(h) < n$ , troca  $h$  por  $h + \Delta h/2$ , troca  $\Delta h$  por  $\Delta h/2$ , e retorna ao passo 3. Se  $n(h) = n$ , forma  $n$  grupos da seguinte maneira. Faz cada um dos  $n$  grupos como  $C(i;n)$ . O grupo  $C(i;n)$  corresponde à variável apontadora  $P(A_i)$ , a qual aponta para o endereço  $D(i)$ . Em primeiro lugar, faz cada grupo  $C(i;n)$  como um conjunto vazio. Repete o seguinte procedimento de  $j=1, \dots, N$ . Inclui o objeto  $A_j$  no grupo  $C(i;n)$  se  $P(A_j)$  aponta para  $D(i)$ . Como resultado, são obtidos  $n$  grupos.

No passo 3, no estado inicial,  $N$  grupos estão presentes. Cada grupo consiste de um único objeto. No passo 4, para todos os possíveis pares de  $(i,j)$ ,  $1 \leq i < j \leq N$ , o grupo incluindo o objeto  $A_j$  é unido ao grupo incluindo o objeto  $A_i$ , se  $s_{ij} \geq h$  e estes dois grupos são diferentes um do outro: cada grupo obtido da operação de variáveis apontadoras é formado pela fusão de outros grupos. O número de grupos diminui de um em cada operação de variáveis apontadoras. Desde que  $N$  grupos estão presentes no início, a operação de variáveis apontadoras é realizada ao menos  $N-1$  vezes. Os procedimentos do passo 3 até o passo 6 são repetidos até  $n(h)$  ficar igual a  $n$ .

Este algoritmo é eficaz, porém exige que o número de grupos seja conhecido antecipadamente. Já o segundo algoritmo proposto é semelhante ao primeiro, porém produz várias soluções prováveis para o agrupamento, no qual o número de grupos a ser obtido não é conhecido antecipadamente.

- Passo 1. Calcula o desvio padrão  $\sigma$  de todas as distâncias. Determina adequadamente o valor de  $C$ . Constrói a matriz de similaridade  $S$ .
- Passo 2. Determina os valores de  $p$ ,  $\Delta h$  e  $\delta$ . Prepara uma variável  $N_C$ . Prepara  $N$  diferentes endereços  $D(i)$ ,  $i=1, \dots, N$ . Prepara  $N$  variáveis apontadoras  $P(A_i)$ ,  $i=1, \dots, N$ . Prepara as variáveis  $a(n)$ ,  $h_1(n)$  e  $h_u(n)$ ,  $n=1, \dots, N$ .
- Passo 3. Troca  $h$  por  $1-\Delta h$ . Troca  $N_C$  por  $N$ . Faz cada  $P(A_i)$  apontar para o endereço  $D(i)$ ,  $i=1, \dots, N$ . Faz cada  $a(n)$  zero,  $n=1, \dots, N$ .
- Passo 4. Repete os seguintes procedimentos para todos os possíveis pares de  $(i, j)$  sob a condição de  $1 \leq i < j \leq N$ . Toma um par de  $(i, j)$ . Se  $h \leq s_{ij} < h + \Delta h$ , e o endereço apontado por  $P(A_i)$  é diferente do apontado por  $P(A_j)$ , busca por todas as variáveis ponteiro apontando para o endereço apontado por  $P(A_j)$ . Faz  $P(A_i)$  e todas estas variáveis ponteiro apontarem para  $P(A_i)$ , e reduz o número de grupos  $N_C$  de um.
- Passo 5. Armazena o resultado do passo 4 na forma de um conjunto de  $N$  variáveis apontadoras, que produz  $N_C$  grupos. Faz  $h_1(N_C)$  igual a  $h$ . Aumenta o valor de  $a(N_C)$  de um.
- Passo 6. Troca  $h$  por  $h - \Delta h$ . Retorna ao passo 4, e repete os procedimentos do passo 4 ao passo 6 até o valor de  $h$  estar igual a zero.
- Passo 7. Se possível, forma a seqüência decrescente  $(a(n_1), a(n_2), \dots, a(n_p), \dots, a(n_q), a(n_{q+1}))$ , onde  $q \geq p$ , e as duas desigualdades seguintes são satisfeitas:
- $$a(n_1) \geq a(n_2) \geq \dots \geq a(n_p) = a(n_{p+1}) = \dots = a(n_q) > a(n_{q+1}) \text{ e:} \quad (89)$$
- $$a(n_{q+1}) \geq a(n) \text{ para } 2 \leq n \leq N-1, \text{ e } n \neq n_1, n_2, \dots, n_{q+1}. \quad (90)$$
- Passo 8. Repete todos os seguintes procedimentos para  $k=1, 2, \dots, q$ . Faz cada elemento do conjunto de  $N$  variáveis apontadoras – que produzem  $n_k$  grupos, armazenadas no passo 5 – tornar-se  $P(A_i, n_k)$ ,  $i=1, 2, \dots, N$ . Repete o seguinte procedimento de  $i=1$  até  $N$ . Busca por  $P(A_i, n_k)$  que aponta para o endereço  $D(i)$ . Forma  $n_k$  grupos da seguinte maneira. Faz cada um dos  $n_k$  grupos como  $C(i, n_k)$ . O grupo  $C(i, n_k)$  corresponde à variável apontadora  $P(A_i, n_k)$ . Primeiramente, faz cada grupo  $C(i, n_k)$  um conjunto vazio. Repete o seguinte procedimento de  $j=1$  até  $N$ . Inclui o objeto  $A_j$  no grupo  $C(i, n_k)$ , se  $P(A_j, n_k)$  aponta para  $D(i)$ . Como resultado,  $n_k$  grupos são obtidos. Faz  $h_u(n_k)$  como  $(h_1(n_k) + (a(n_k) - 1)\Delta h^*)$ , onde  $\Delta h^*$  é o valor final de  $\Delta h$ , determinado no passo 7.

No passo 2, o número de soluções prováveis é suposto para ser no mínimo  $p$ . O parâmetro  $\Delta h$  denota o decremento de  $h$  na repetição dos procedimentos de agrupamento, e  $\delta$  é o parâmetro pelo qual o mínimo admissível de  $\Delta h$  é determinado. A variável  $N_C$  denota o número de grupos nos quais o conjunto de dados é classificado. A variável  $a(n)$  representa o número de casos onde o conjunto de dados é classificado em  $n$  grupos na repetição dos procedimentos de agrupamento pela diminuição de  $h$ . As variáveis  $h_1(n)$  e  $h_u(n)$  são respectivamente o limite inferior e superior de  $h$ , sendo o conjunto de dados classificado em  $n$  grupos.

No passo 3,  $h$ ,  $N_C$  e  $P(A_i)$  são inicializados. No passo 4, em cada repetição da operação de variáveis apontadoras, um grupo é unido a outro grupo, e o valor de  $N_C$  é reduzido de 1. No passo 5, o resultado no qual o conjunto de dados é classificado em  $N_C$  grupos é memorizado na forma de um conjunto de  $N$  variáveis apontadoras. Para este resultado, um limite inferior aproximado  $h_1(N_C)$  é memorizado como  $h$ . O número de casos nos quais o conjunto de dados é classificado em  $N_C$  grupos é também memorizado. Os procedimentos do passo 4 até o passo 6 são repetidos trocando o valor de  $h$ .

No passo 7, quando o número  $q$  satisfaz as Equações (89) e (90), uma seqüência decrescente  $(a(n_1), a(n_2), \dots, a(n_{q+1}))$  é obtida do procedimento de ordenação de  $a(n)$ ,  $n=2, 3, \dots, N-1$ . Seja  $W(n_k)$  o tamanho de  $h$  para o qual o conjunto de dados é classificado em  $n_k$  grupos. Tem-se  $(a(n_k)-1)\Delta h \leq W(n_k) < (a(n_k)+1)\Delta h$  para cada  $n_k$ . Se  $a(n_k) \geq a(n_{k+1})+2$ , tem-se  $W(n_k) > W(n_{k+1})$ . Neste caso, considera-se que o resultado de agrupamento onde o conjunto de dados é classificado em  $n_k$  grupos é melhor que a solução de agrupamento na qual o conjunto de dados é classificado em  $n_{k+1}$  grupos. Considera-se que o grau de separação de  $n_k$  grupos é melhor quando  $W(n_k)$  é maior.



Se  $a(n_k) \geq a(n_{k+1}) + 2$ , para  $k=1, 2, \dots, p$ , obtêm-se  $p$  prováveis soluções que satisfazem  $W(n_k) > W(n_{k+1})$ ,  $k=1, 2, \dots, p$ . Quando esta condição não é satisfeita, e  $\Delta h > \delta$ , troca-se  $\Delta h$  por  $\Delta h/2$  e repete os procedimentos do passo 3 ao 7. Se  $a(n_k) \geq a(n_{k+1}) + 2$ , para  $k=1, 2, \dots, p$ , ou se  $\Delta h$  já é muito pequeno e  $\Delta h \leq \delta$ , interrompe-se as repetições. Quando a seqüência de repetições que satisfaz as Equações (89) e (90) não pode ser obtida, os valores de  $p$ ,  $\Delta h$  e  $\delta$  determinados no passo 2 podem não ser bons. Os valores de  $p$  e  $\delta$  podem ser muito grandes. Em tais casos, a fim de recomeçar os procedimentos de agrupamento,  $p$  e/ou  $\delta$  devem ser menores, e um valor inicial menor de  $\Delta h$  seria mais adequado.

No passo 8, tem-se  $q=p$ , quando a condição  $a(n_k) \geq a(n_{k+1}) + 2$ , para  $k=1, 2, \dots, p$ , é satisfeita. Quando esta condição não é satisfeita apesar de  $\Delta h \leq \delta$ , tem-se  $q > p$ , se  $a(n_p) = a(n_{p+1})$ , e tem-se  $q = p$ , se  $a(n_p) > a(n_{p+1})$ . Em qualquer caso, obtêm-se  $q$  prováveis soluções do problema do agrupamento. Para a  $k$ -ésima provável solução na qual o conjunto de dados é classificado em  $n_k$  grupos, um limite superior aproximado  $h_u(n_k)$  é obtido como  $(h_1(n_k) + (a(n_k) - 1)\Delta h^*)$ . Limites acurados são dados por  $(h_1(n_k) - \delta_1(n_k))$  e  $(h_u(n_k) + \delta_2(n_k))$ , onde os parâmetros  $\delta_1(n_k)$  e  $\delta_2(n_k)$  estão no intervalo  $0 \leq \delta_1(n_k)$ ,  $\delta_2(n_k) < \Delta h^*$ . Portanto,  $W(n_k)$  é dado aproximadamente por  $(h_u(n_k) - h_1(n_k))$ , isto é  $W(n_k) \cong (a(n_k) - 1)\Delta h^*$ .

Nos experimentos realizados, foi considerado um conjunto bi-dimensional de 200 pontos. Os parâmetros  $C$  e  $p$  foram definidos como 1 e 3, respectivamente. O valor inicial de  $\Delta h$  foi estabelecido em 0.1, e o parâmetro  $\delta$  em 0,0125. Como resultado, o valor obtido para  $\Delta h^*$  é 0,0125. O número encontrado para  $q$  é 4, e 4 soluções prováveis são obtidas. Os valores de  $W(n_k)$  determinaram a ordem de preferência entre as soluções obtidas.

Estes algoritmos têm tempo de complexidade  $O(N^2)$  para o agrupamento de  $N$  objetos e são eficientes para o método do vizinho mais próximo. O algoritmo 1 é eficiente quando o

número de grupos é conhecido, e o algoritmo 2 é eficiente para se obter várias soluções prováveis para o agrupamento quando o número de grupos não é conhecido antecipadamente. Entretanto, com relação ao segundo algoritmo, não é seguro determinar a melhor solução a partir de  $W(n_k)$ , sendo necessárias outras informações para tal, o que constitui uma questão a ser investigada.

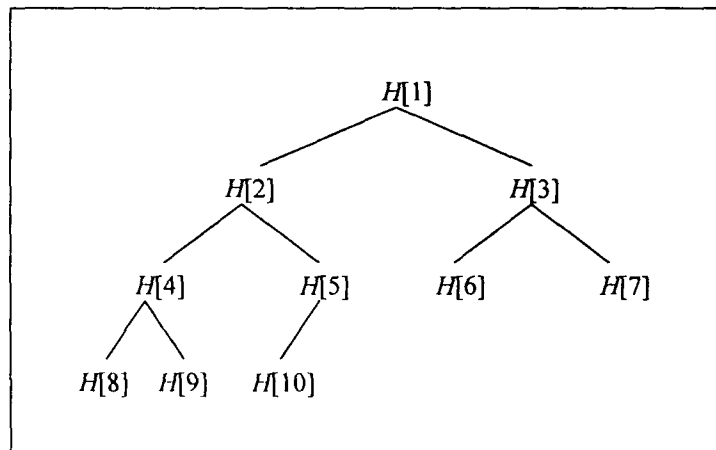
3.2.3 Outra abordagem hierárquica é sugerida por KURITA (1991), através de um algoritmo eficiente para o agrupamento aglomerativo, no qual as distâncias de todos os pares de grupos são armazenadas em um *heap*.

A proposta utiliza as características de um *heap* para acelerar a busca pelo par de grupos mais próximos. Logo, o algoritmo proposto difere do geral no que se refere ao armazenamento das distâncias. Um *heap* pode ser visto como uma árvore binária, sendo definido como uma seqüência de valores:

$$\begin{aligned}
 &H[s], H[s + 1], \dots, H[e], \text{ tal que:} \\
 &H[i] \leq H[2i], \\
 &H[i] \leq H[2i + 1], \forall i = s, \dots, e/2, \\
 &H[1] = \min( H[1], \dots, H[e] ). \tag{91}
 \end{aligned}$$

A mais importante característica de um *heap* é que o valor do nodo raiz da árvore binária associada é o menor valor, o que facilita a operação de busca pelo par de grupos mais próximos.

FIGURA 10 - UM EXEMPLO DE UMA ÁRVORE BINÁRIA CORRESPONDENDO A UM *HEAP* COM DEZ VALORES



FONTE : KURITA (1991)

Considerando  $d_{ij}$  a distância entre os objetos  $i$  e  $j$ , e  $d_{ij}=d_{ji}$ , todas as distâncias são armazenadas em uma matriz. Definida tal matriz, o algoritmo para agrupamento aglomerativo utilizando um *heap* é o seguinte:

- Passo 1. Inicia com  $n$  grupos, cada um consistindo de apenas de um objeto. Sejam os grupos rotulados com os números de 1 até  $n$ .
- Passo 2. Constrói um *heap* que contém todas as distâncias entre os grupos.
- Passo 3. Pega o par de grupos mais próximos que corresponde ao elemento  $H[1]$ . Sejam os grupos escolhidos  $p$  e  $q$ , e  $d_{pq}$  sua distância associada.
- Passo 4. Reduz o número de grupos de um pela união dos grupos  $p$  e  $q$ . Atualiza o *heap* para refletir as distâncias revisadas entre o novo grupo e os demais grupos existentes, eliminando elementos desnecessários no *heap*.
- Passo 5. Repete os passos 3 e 4  $N-1$  vezes no total, em cujo ponto todos os objetos são reunidos em um grupo.

Este algoritmo, em relação ao usual agrupamento aglomerativo, reduz o tempo de complexidade de  $O(N^3)$  para  $O(N^2 \log N)$ , o que permite tratar mais objetos.

Os experimentos foram realizados utilizando uma matriz de distâncias geradas aleatoriamente, com três algoritmos: o algoritmo aglomerativo básico, o algoritmo acima descrito, e um terceiro algoritmo que mantém um índice do grupo mais próximo para cada grupo, armazenando o conjunto de índices em uma matriz. Neste último algoritmo, o tempo de computação para busca do par mais próximo é  $O(N)$ . Entretanto, o tempo de computação total do algoritmo é  $O(N^2)$  no pior caso, porque é necessário procurar pelo grupo mais próximo quando as distâncias relacionadas são influenciadas por *merging* de par de grupos.

Os resultados dos experimentos demonstraram que o algoritmo proposto, que utiliza um *heap* no qual as distâncias de todos os pares de grupos são armazenadas, é eficiente.

3.2.4 BARONTI, CASINI e LOTTI (1990) apresentam uma variação do algoritmo piramidal geral pelo aumento do *span* nas últimas iterações a fim de corrigir a deficiência da ligação (*link*) dos nodos de borda, aplicando à segmentação de imagens.

Partindo da imagem de níveis de cinza original, uma matriz de *pixels*  $2^M \times 2^M$  está no nível base ( $l=0$ ); qualquer outra matriz no nível genérico  $l$  é gerada com  $1/4$  dos elementos da matriz no nível  $l-1$ , até um elemento no vértice ( $l=M$ ) da pirâmide.

O nível de cinza de cada nodo num nível  $l>0$  é usualmente calculado pela média dos nodos do nível  $l-1$ .

Múltiplos *links* podem ser estabelecidos entre nodos por meio de algum critério de homogeneidade e de uma escolha apropriada do *span* da pirâmide. O *span* determina o conjunto de todos os possíveis *links* para um nodo, e que é normalmente mantido constante durante todo o procedimento recursivo.

O *span*  $c$  ( $c \geq 1$ ) é o conceito chave em estruturas de pirâmides ligadas: ele é o tamanho da metade do lado de um bloco quadrado de  $2c \times 2c$  nodos no nível  $l-1$  o qual inclui todos os possíveis filhos para os quais um nodo interno em algum nível  $l$  ( $l > 0$ ) pode ser ligado; um nodo interno no nível  $l$  pode ser ligado com  $c \times c$  pais no nível  $l+1$ .

Em algoritmos piramidais todas as ligações de um nodo para seus pais podem ser levadas em conta e podem ser atribuídos diferentes pesos por meio de uma função de similaridade adequada. Para aliviar o ônus da computação, somente uma ligação para um único pai preferencial é comumente preservada.

A configuração das ligações em toda a pirâmide é recursivamente atualizada até que se alcance um estado estável, usualmente depois de poucas iterações. Em cada iteração, os valores associados com um nodo são recalculados de acordo com valores dos filhos ligados.

Em aplicações de segmentação, cada algoritmo tem uma maneira peculiar para separar alguns nodos da pirâmide, os quais podem se tornar as raízes de árvores, cujos nodos folha no nível base são os *pixels* dos segmentos candidatos.

Depois da inicialização, um *span*  $c=2$  é comumente usado durante todo o procedimento. Nesta proposta, ocorre a implementação de um *span* variável, isto é, aumenta-se o *span* para  $c=3$  nas iterações finais.

Esta variação é sugerida pela escassez de escolhas de ligações imposta pelo *span*  $c=2$  sobre nodos de borda e canto. De fato, em qualquer nível, os nodos de borda têm somente dois pais candidatos, e nodos de canto têm somente um candidato a pai.

Esta falta é propagada de cima para baixo na pirâmide, pelos filhos ligados, até o nível base, e pode causar erros notáveis de segmentação.

No algoritmo proposto, cada nodo liga-se a um único pai e torna-se uma raiz se a diferença de nível de cinza entre os dois excede um limiar pré-definido. Um limiar de área é também definido, relativo ao tamanho de área da região associada a uma raiz.

O algoritmo, que aceita como entrada o número máximo de grupos, é o seguinte:

Passo 1. Os limiares de nível de cinza  $T_g$  e de área  $T_a$  são inicializados adequadamente com base no conhecimento *a priori* sobre o contraste da imagem e o tamanho dos objetos considerados distintos.

Passo 2. A pirâmide é inicializada com  $span=2$ .

Passo 3. A religação é executada iterativamente um certo número de vezes. A cada iteração, nos níveis da pirâmide, um nodo é uma raiz candidata se:

- . a diferença de nível de cinza com o pai mais similar excede o limiar  $T_g$  ;
- . o número de *pixels* associados na imagem do nível base está acima do limiar  $T_a$ ; uma raiz candidata não contribui para as propriedades de um nodo nos níveis superiores.

Passo 4. Depois da última iteração, um procedimento de agrupamento é aplicado às raízes de vizinhança 8 quando as diferenças de seus níveis de cinza estão abaixo do limiar  $T_g$ . Cada um desses grupos é caracterizado pelo nível de cinza médio e pelo número total de *pixels* no nível base, ligados às raízes agrupadas.

Dois grupos são unidos se a diferença de seus níveis de cinza médios está abaixo de  $T_g$ , e se a condição de proximidade é verificada. Para definir esta última condição opera-se o seguinte:

- . No nível base, a janela retangular mínima contendo as projeções de todas as raízes de um grupo é separada e sua área é calculada ( $A_i$ ,  $i=1,2$ );
- . A área  $A_t$  da janela mínima contendo  $A_1$  e  $A_2$  é calculada e seja  $A_m$  o máximo entre eles;
- . A razão  $A_m/A_t$  é avaliada; se excede um valor predefinido, a condição de proximidade é verificada e o novo grupo está formado com um nível de cinza que é a média, e uma área que é a soma daqueles dois grupos.

O processo é repetido enquanto existem grupos para unir. Um grupo novo separa uma região.

Passo 5. Se o número total de regiões é menor que o máximo estabelecido, o procedimento é terminado. Caso contrário, os limiares  $T_g$  e  $T_a$  são atualizados e o processo continua do passo 3.

A fim de acelerar a convergência, os limiares são atualizados por meio de funções heurísticas de incremento que exploram os resultados intermediários.

Determinou-se como eficiente variar  $T_a$  somente se o número de regiões resultantes  $N_g$  é significativamente maior que o máximo fornecido pelo usuário (e.g., >70%).

Nesse caso, calculou-se os números de grupos  $N_1$  e  $N_2$  cujas áreas são, respectivamente, menores que  $T_a+dT_a$  e  $T_a+2dT_a$ . Se  $N_1$  é suficientemente maior (e.g.,  $N_1>0,25N_g$ ), o novo limiar de área  $T_a'$  é calculado como:

$$T_a' = T_a + dT_a(1 + N_1 / N_g); \quad (92)$$

caso contrário, se  $N_2$  é suficientemente maior (e.g.,  $N_2>0,3N_g$ ), então:

$$T_a' = T_a + 2 dT_a(1 + N_2 / N_g); \quad (93)$$

em todos os outros casos somente  $T_g$  é atualizado.

Com este propósito, empiricamente adotou-se a regra:

$$dT_g = 0,5 \sqrt{\sigma R_g}, \quad (94)$$

onde  $\sigma$  é o desvio padrão dos níveis de cinza das regiões.  $R_g$  é a razão entre  $N_g$  e o máximo fornecido pelo usuário.

O significado da Equação 94 é que  $dT_g$  aumenta com a variação de nível de cinza da imagem segmentada: quando as regiões têm níveis de cinza similares, um pequeno  $dT_g$  será suficiente para reduzir o número de segmentos nas iterações seguintes. A dependência sobre  $R_g$

sobrecarrega a distância do número de segmentos permitidos: se existem somente umas poucas regiões maior que o máximo dado pelo usuário, uma pequena variação de será suficiente para mudar abaixo do limite. A raiz quadrada pequena e o fator 0,5 deram bons resultados.

Com esta proposta, é possível corrigir os efeitos de borda. O uso de um  $span=3$  em todas as iterações também poderia superar este problema, porém com um tempo de computação dispendioso. Assim, o método proposto, no qual depois de algumas iterações com  $span=2$ , umas poucas iterações com  $span=3$ , é uma solução simples que torna a segmentação eficiente e com apenas um pequeno aumento do tempo de computação.

3.2.5 GROSKY e JAIN (1986) apresentam um esquema de segmentação baseado em uma arquitetura piramidal que utiliza *relinking*, cuja eficiência é demonstrada no contexto de *matching* de regiões através do cálculo de uma característica global para cada região segmentada.

O esquema tem a propriedade de que cada região no nível base da pirâmide compreende os nodos folha de uma árvore. Esta árvore representa um valor característico uniforme: cada nodo da árvore tem um rótulo similar.

Neste trabalho, uma arquitetura de pirâmide de  $N+1$  níveis pode ser comparada a uma pilha de  $N+1$  matrizes bi-dimensionais, estando a base no nível  $N$ , de tamanho  $2^N \times 2^N$ , enquanto cada uma das outras matrizes têm tamanho igual a  $1/4$  do tamanho da matriz abaixo dela. Assim, o ápice ou topo da pirâmide é uma matriz de tamanho  $1 \times 1$ .

A cada posição de uma matriz corresponde um elemento, que está ligado a um elemento na matriz acima (seu pai), 8 elementos na mesma matriz (seus irmãos), e 4 elementos



na matriz abaixo (seus filhos). Obviamente, elementos folha não têm filhos, elementos borda têm menos que 8 irmãos, e elementos raiz não têm pai.

Estabelecendo um sistema de coordenadas inteiro em cada matriz, e endereçando um elemento particular em uma dada matriz pela tripla ordenada  $(i,j,k)$ , onde  $i$  é o nível da matriz na pirâmide, e  $(j,k)$  são as coordenadas do elemento na  $i$ -ésima matriz, tem-se:

- para  $1 \leq i \leq N$ ,  $0 \leq j, k \leq 2^i$ , o pai do elemento  $(i,j,k)$  é o elemento  $(i-1, j \div 2, k \div 2)$ ;
- para  $1 \leq i \leq N-1$ ,  $0 \leq j, k \leq 2^i$ , os filhos do elemento  $(i,j,k)$  são os elementos  $(i+1, 2j, 2k)$ ,  $(i+1, 2j+1, 2k)$ ,  $(i+1, 2j, 2k+1)$ ,  $(i+1, 2j+1, 2k+1)$ .

Arquiteturas de pirâmide podem ser estendidas pelo aumento de conexões pai-filho. Uma pirâmide tem um *span*  $c \geq 1$ , para um  $c$  inteiro, se cada elemento no nível  $i$ ,  $0 \leq i \leq N-1$ , tem  $(2c)^2$  filhos no nível  $i+1$ . Estes filhos estão na forma de um quadrado  $2c \times 2c$  cujo núcleo central  $2 \times 2$  consiste dos 4 filhos padrão.

A pirâmide clássica tem um *span* de 1. Para *span's*  $c > 1$ , elementos próximos às bordas têm menos que  $(2c)^2$  filhos, assim como alguns de seus filhos potenciais estarão sobre a borda. Em tais pirâmides, um elemento cuja distância para qualquer borda é no mínimo  $c-1$ , tem  $c^2$  pais.

Esta última colocação leva à religação da pirâmide (*relinking*) que, em muitas aplicações, leva à seleção de um pai apropriado do conjunto de possíveis pais de um elemento, o que é feito pela remoção das ligações com todos os seus pais, com exceção de um. Assim, em uma pirâmide de *span*  $c=2$ , é possível para um elemento ter de zero a 16 filhos.

O objetivo desta abordagem é definir um algoritmo que utilize um esquema de *relinking* para segmentar e representar uma imagem digital armazenada na base de uma

pirâmide. Partindo de uma imagem digital de tamanho  $2^N \times 2^N$ , ela é encaixada no centro de uma imagem de fundo de tamanho  $2^{N+M} \times 2^{N+M}$ . A imagem resultante é armazenada na base de uma pirâmide de *span*  $c=2$ . Quanto ao valor de  $M$ , os autores mencionam que, quanto maior, melhor o desempenho do algoritmo, e permitindo a religação dos elementos de borda.

Utilizando conectividade-4, dois elementos de vizinhança 4 pertencem ao mesmo segmento se seus valores são não-fundo e próximos o suficiente um do outro. Cada árvore formada num dado nível da pirâmide irá corresponder a uma região ou sub-região e será uniformemente rotulada. Os rótulos, no nível base, habilitam duas sub-regiões de conectividade-4 a serem combinadas. Esta informação é levada à raiz da dada sub-árvore no nível  $i$ . A raiz então escolhe o novo rótulo para a região combinada. Subseqüentemente, a religação é feita, resultando em novos rótulos no nível  $i-1$ , os quais são, por sua vez, passados ao nível base.

Tal algoritmo está baseado no fato de que dois elementos vizinhos pertencem ao mesmo grupo se seus elementos são ambos não-fundo e próximos o suficiente um do outro. Isto pode ser trocado pela comparação do valor de um elemento com o valor médio de todos os elementos num segmento.

Para a combinação de regiões (*matching*), é introduzida uma técnica de mínimos quadrados que calcula várias características utilizando a arquitetura de pirâmide. Uma região em uma imagem é representada encontrando-se um parabolóide elíptico que melhor se ajuste à superfície de níveis de cinza correspondente à região. O cálculo das características é encaixado na arquitetura de forma que a representação de cada região em uma imagem é tão próxima do ápice quanto possível e tal que os parâmetros das superfícies de melhor ajuste podem ser calculados na pirâmide concorrentemente à religação.

Assim, é possível calcular os parâmetros das superfícies de níveis de cinza muito rapidamente, já que o *matching* das superfícies pode ser feito sem o retorno à imagem original. Através de experimentos com imagens binárias, os autores demonstram que o método produz bons resultados.

3.2.6 O trabalho de ZAHN (1971) demonstra como a árvore geradora mínima pode ser usada para detectar a estrutura de grupo existente em um conjunto de dados arbitrários, utilizando uma função de distância. A árvore geradora mínima constitui um poderoso mecanismo grafo-teórico para a detecção de grupos por satisfazer ao princípio básico de Gestalt, que favorece agrupamentos com menor distância entre as amostras.

A idéia consiste em construir a árvore geradora mínima para o conjunto de amostras. O segundo passo consiste em identificar arestas inconsistentes na árvore geradora mínima. Finalmente, a remoção das arestas inconsistentes decompõe o grafo em componentes conexos, onde cada componente conexo representa um grupo. A aplicação iterativa desses passos a cada componente conexo resultante, permite a identificação de subgrupos.

Vários critérios para a definição de inconsistência são considerados. Um deles está relacionado à separação de grupos, no qual uma aresta é inconsistente se seu peso (distância entre os nodos) é significativamente maior que a média dos pesos das arestas mais próximas. Outra maneira de identificar arestas inconsistentes é verificar em quantos desvios padrão o peso de uma aresta difere das médias de pesos de arestas mais próximas. Em qualquer um dos casos, uma aresta com um fator de inconsistência de dois usualmente significa que a separação é bem aparente, segundo o autor.

O algoritmo acima descrito dá bons resultados quando o conjunto de amostras é formado por grupos bem separados e homogêneos. Quando aplicada a conjuntos de amostras consistindo de grupos com formas variadas, incluindo grupos muito próximos entre si, grupos não homogêneos, ou grupos alongados, cujas amostras ficam alinhadas, semelhantes a galhos, heurísticas adequadas devem ser utilizadas a fim de identificar arestas inconsistentes. Por exemplo, no caso de grupos que estão muito próximos, a heurística sugerida consiste em determinar um *neck* no grafo, que é um subgrafo conexo que quando removido desconecta o grafo em seus componentes. Através do histograma dos pesos das arestas do diâmetro que contém o *neck*, é possível a identificação da aresta inconsistente.

Os experimentos computacionais foram realizados com vários conjuntos de dados, ilustrando vários tipos de estruturas de grupos. Segundo o autor, o conhecimento *a priori* das formas dos grupos é necessário a fim de selecionar a heurística adequada. Isso constitui uma desvantagem quando aplicado a conjuntos de dimensão maior que dois, quando a identificação das formas dos grupos é difícil.

3.2.7 O trabalho de HUBERT (1974) revê e expande o relacionamento entre a teoria dos grafos e o agrupamento de um conjunto de dados. Vários critérios grafo-teóricos são propostos para uso dentro de um paradigma de agrupamento geral no desenvolvimento de procedimentos “entre” os extremos “*complete-link*” e “*single-link*”.

Os dois procedimentos de agrupamento clássicos que constróem hierarquias de partições a partir da matriz de proximidades ordinal  $\{s_{ij}\}$  são os chamados *single-link* e *complete-link*. Estes métodos formam um novo subconjunto na partição no nível  $u+1$  pela

união de um único par de subconjuntos do nível  $u$  e transferindo os remanescentes para o nível  $u+1$ , conforme definido na Seção 2.2. .

O autor utiliza a seguinte formulação geral para os agrupamentos hierárquicos *single-link* e *complete-link*. Suponha  $S$  um conjunto de  $n$  amostras distintas  $\{o_1, \dots, o_n\}$  e  $\{s_{ij}\}$  uma matriz representando as proximidades entre cada par de objetos  $o_i$  e  $o_j$ , assumindo que as medidas de proximidade são todas distintas, simétricas, e têm somente significado ordinal. Cada valor de proximidade  $c$  define um grafo finito correspondente  $G_c$ , sem arestas múltiplas ou reflexivas. Em particular, se os nodos de  $G_c$  são os elementos de  $S$ , então existe uma aresta entre dois nodos distintos  $o_i$  e  $o_j$  se e somente se  $s_{ij} \leq c$ . Se  $D$  é um subconjunto de  $S$ , então um subgrafo maximal de  $G_c$  definido pelo conjunto de nodos  $D$  é meramente um grafo que consiste dos nodos em  $D$  e onde  $o_i, o_j \in D$  são ligados se e somente se eles são ligados em  $G_c$ . Assim, um grafo  $D$  é dito conexo se para cada par de nodos  $o_i, o_j \in D, i \neq j$ , existe um caminho entre  $o_i$  e  $o_j$ , isto é, uma série de nodos intermediários distintos em  $D, o_{k1}, \dots, o_{km}$ , tal que  $o_i = o_{k1}, o_{km} = o_j$  e  $o_{ki}$  é ligado a  $o_{ki+1}$ . Um grafo  $D$  é dito completo se todos os pares de nodos em  $D$  estão ligados, e trivial se ele contém somente um nodo e nenhuma aresta.

Para cada par de subconjuntos no nível  $u$ , duas medidas  $Q_c$  e  $Q_s$ , podem ser definidas, correspondendo aos agrupamentos hierárquicos *complete-link* e *single-link*:

$$Q_c(L_{ur}, L_{us}) = \min\{s_{ij} \mid \text{o subgrafo maximal de } G_{s_{ij}} \text{ definido pelo conjunto de pontos } L_{ur} \cup L_{us} \text{ é completo}\};$$

$$Q_s(L_{ur}, L_{us}) = \min\{s_{ij} \mid \text{o subgrafo maximal de } G_{s_{ij}} \text{ definido pelo conjunto de pontos } L_{ur} \cup L_{us} \text{ é conexo}\}.$$

Os subconjuntos  $L_{up}$  e  $L_{uq}$  que são unidos na forma de um novo elemento ou na partição *single-link* ou na *complete-link* no nível  $u+1$ , devem satisfazer os seguintes relacionamentos:

$$\textit{Complete-link: } Q_c(L_{up}, L_{uq}) = \min Q_c(L_{ur}, L_{us});$$

$$\textit{Single-link: } Q_s(L_{up}, L_{uq}) = \min Q_s(L_{ur}, L_{us}), \text{ onde } r \neq s.$$

Em outras palavras, usando uma seqüência ordenada de grafos,  $\{G_{s_{ij}}\}$ , definida pelo aumento do valor de  $s_{ij}$ , o método *single-link* une aqueles dois subconjuntos no nível  $u$  que têm uma única ligação existente entre os correspondentes subgrafos maximais para o menor valor possível de  $s_{ij}$ . Inversamente, o método *complete-link* une aqueles dois subconjuntos que têm todas as possíveis ligações definidas entre os correspondentes subgrafos maximais para o menor valor possível de  $s_{ij}$ .

As propriedades de conectividade e compleição de um grafo podem ser tratadas como dois extremos, existindo muitos outros conceitos naturais e bem pesquisados em teoria dos grafos que sugerem critérios intermediários entre esses dois. Mais significativamente, vários destes conceitos parecem capazes de gerar procedimentos razoáveis para a construção de hierarquias usando somente a ordem das medidas de proximidade. As propriedades de agrupamento dadas a seguir constituem critérios alternativos:

(a) Conetividade  $k$ -nodo: um grafo é  $k$ -nodo conexo se o número mínimo de nodos cuja remoção resulta em um grafo desconexo ou trivial é maior ou igual a  $k$ . Alternativamente, um grafo é  $k$ -nodo conexo se e somente se cada par de nodos são unidos ao menos por  $k$  caminhos de nodos disjuntos.

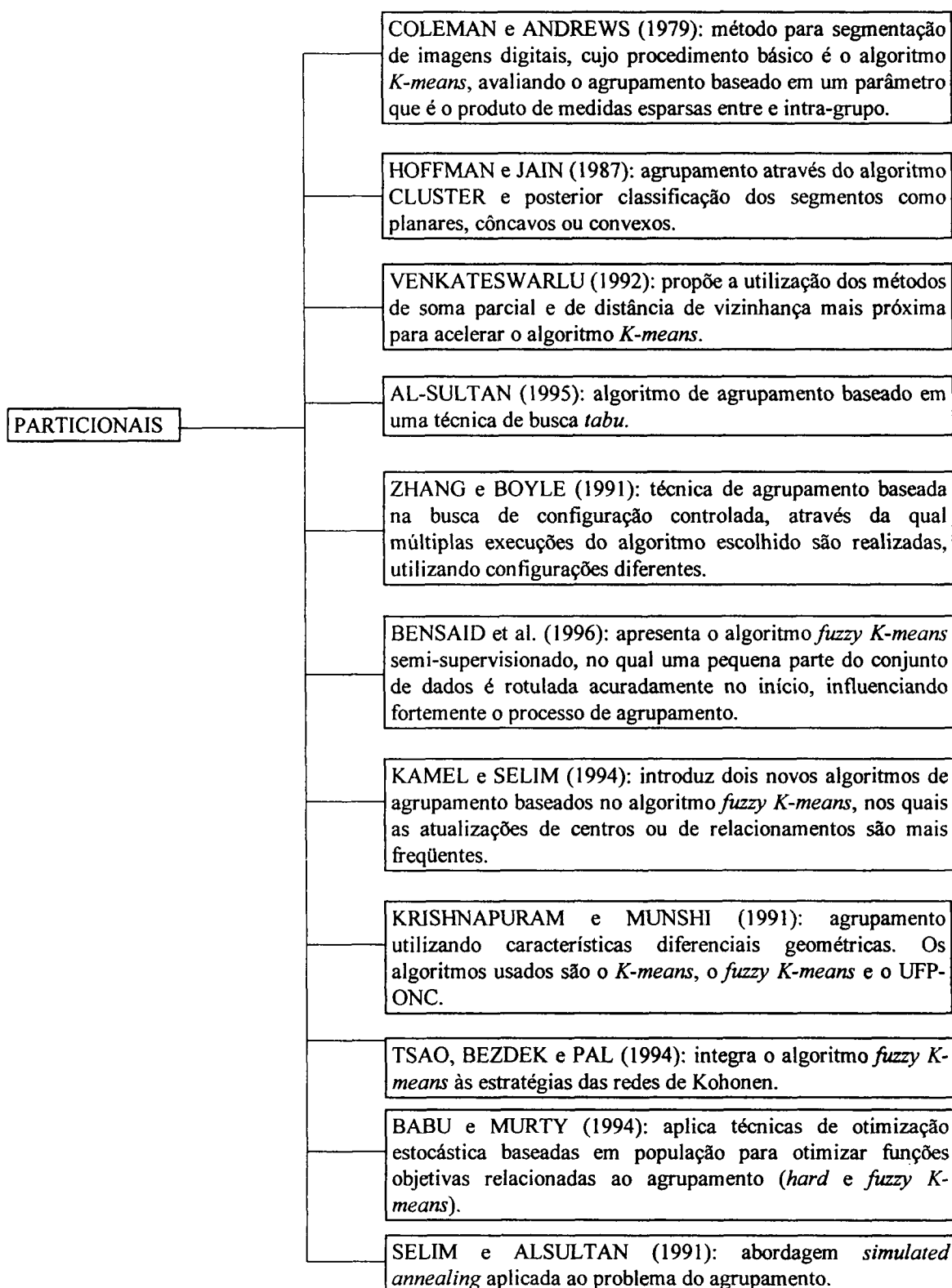
(b) Conetividade  $k$ -aresta: um grafo é  $k$ -aresta conexo se o número mínimo de arestas cuja remoção resulta em um grafo desconexo ou trivial é maior ou igual a  $k$ . Alternativamente, um grafo é  $k$ -aresta conexo se e somente se cada par de nodos são unidos ao menos por  $k$  caminhos de arestas disjuntas.

(c) Grau  $k$ -mínimo: um grafo tem um grau mínimo maior ou igual a  $k$  se cada nodo está unido ao menos a  $k$  outros nodos.

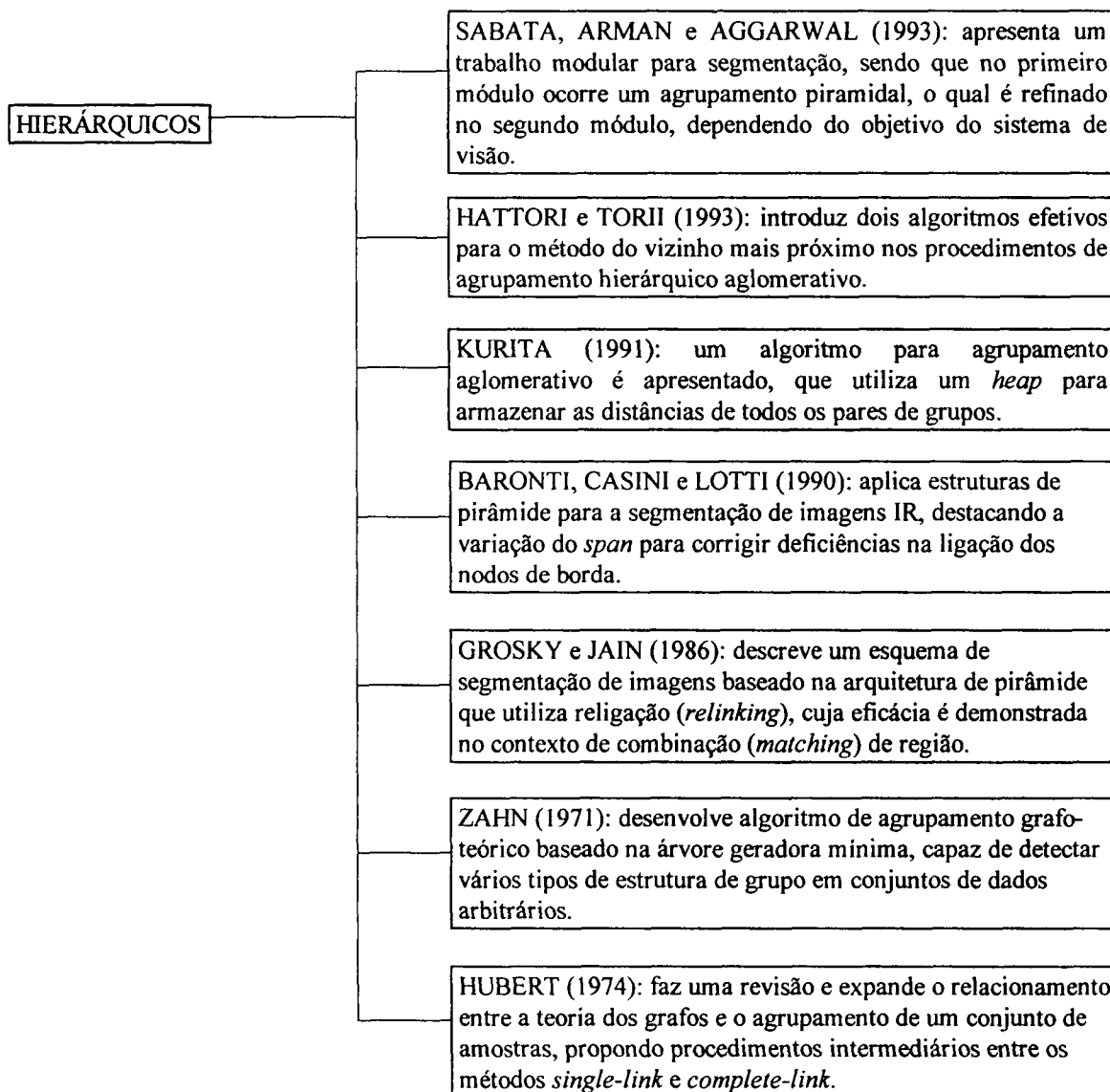
Nos três casos,  $k=1$  define o método *single-link* e  $k=n-1$  dá o método *complete-link*. Conseqüentemente, esses três conceitos caracterizam procedimentos de agrupamento intermediários entre os extremos definidos pelos critérios *single-link* e *complete-link*.

O trabalho relaciona, ainda, outros critérios grafo-teóricos intermediários que constituem simples variações sugerindo, desta forma, que muitos outros critérios podem ser formulados. Uma análise experimental através de valores de proximidade entre treze categorias de diagnóstico psiquiátrico, ilustrando os conceitos apresentados. Os resultados empíricos não apontam para uma direção, demonstrando que a finalidade desse trabalho é mostrar as várias direções, deixando o problema de aplicação prática das propriedades grafo-teóricas para trabalhos futuros.

## 3.3 DIAGRAMAS DE CLASSIFICAÇÃO







## 4 EXPERIMENTOS

Neste capítulo é feita a descrição dos experimentos. A Seção 4.1 faz uma exposição detalhada de cada algoritmo em termos de limiares de entrada e heurísticas utilizadas. A Seção 4.2 apresenta os resultados obtidos dos testes, e a Seção 4.3 define uma nova abordagem, desenvolvida para melhorar os resultados do agrupamento.

O conjunto de algoritmos selecionado para implementação e testes compreende os seguintes algoritmos: *K-means*, ISODATA, CLUSTER, suas respectivas versões *fuzzy*, e o algoritmo divisivo descrito na Seção 2.2.2.

Em se tratando de imagens de profundidade, a orientação das superfícies visíveis de um objeto é uma característica útil para a fase de reconhecimento. Assim, para a segmentação de imagens de profundidade em segmentos com orientações diferentes, são utilizadas nos experimentos os coeficientes das normais à superfície, como características de cada *pixel*. Para o cálculo de distância entre *pixels* e centros de grupos é considerada a distância Euclidiana.

Os algoritmos são implementados em linguagem C através do software de processamento de imagens Khoros 2.1, em um microcomputador com processador Pentium MMX 166 Mhz, 32Mbytes de memória RAM e 256 Kbytes de memória cache, sob o sistema operacional Linux. Os testes são conduzidos utilizando imagens de profundidade, reais e sintéticas obtidas em <http://www.prip.tuwien.ac.at/ogi-bin/listobj>.

#### 4.1 ALGORITMOS IMPLEMENTADOS

Para o algoritmo *K-means*, os limiares fornecidos pelo usuário considerados na implementação são o número de grupos  $K$  e o número máximo de iterações. O programa, em sua inicialização, utiliza  $K$  centros de grupo que correspondem a valores de  $K$  *pixels* da imagem (mais precisamente, as normais de cada *pixel*), aleatoriamente selecionados.

No algoritmo ISODATA, além do número de grupos  $K$  e do número máximo de iterações, são também considerados o limiar de divisão e união  $T$ , e o número mínimo de amostras dentro de um grupo  $min\%$ .

Na inicialização do ISODATA, todas as amostras pertencem a um único grupo ( $K=1$ ), sendo o centróide deste grupo o vetor médio de todas as amostras. A partir da inicialização, em cada iteração do ISODATA é possível “remover” grupos contendo menos que *min%* do total de amostras da imagem. A remoção corresponde a “marcar” os *pixels* do grupo a ser removido para, ao final das iterações, rotular esses *pixels* de acordo com a sua posição física na imagem. Isto é, atribuir a esses *pixels*, possivelmente ruído, um rótulo de acordo com os *pixels* vizinhos.

Também é possível, em cada iteração do ISODATA, a divisão ou a união de grupos, de acordo com o número de grupos atual  $K'$ : se  $K' < K$ , pode ocorrer a divisão de um grupo, caso contrário pode ocorrer a união de dois grupos. A bibliografia consultada não define explicitamente as heurísticas para união e divisão. Assim, as heurísticas adotadas na implementação do ISODATA são descritas a seguir.

- Admitida a divisão, o grupo que possui o maior erro-quadrático médio é dividido se o erro-quadrático médio deste grupo é maior que o erro-quadrático médio de todos os grupos vezes o limiar  $T$ . O *pixel* mais distante (em distância Euclidiana quadrática) do centro deste grupo constitui um novo centro de grupo.
- Admitida a união, os dois grupos que possuem a menor distância quadrática entre seus centros são unidos se esta distância é menor que a distância quadrática média entre todos os pares de centros de grupos vezes o limiar  $T$ . O novo centro de grupo resulta da média ponderada entre os dois centros de grupo envolvidos na união, sendo que estes dois últimos são removidos do conjunto de centros de grupo.

A cada operação de remoção, divisão ou união, o algoritmo *K-means* é executado para a rotulação adequada conforme o novo conjunto de centros estabelecido em cada caso.

O programa correspondente ao algoritmo CLUSTER tem como limiares fornecidos pelo usuário o número máximo de grupos  $K$  e o número máximo de iterações. A inicialização do CLUSTER é como no ISODATA.

Cada iteração do CLUSTER corresponde à execução das duas fases: uma de divisão de 2 até  $K$  grupos, e outra de união de  $K$  até 2 grupos. Na primeira fase, o critério utilizado para selecionar o grupo para divisão é buscar pelo grupo com o maior erro-quadrático médio, sendo que o *pixel* mais distante do centro deste grupo constitui um novo centro de grupo. As divisões ocorrem iterativamente até que se obtenha  $K$  grupos. Ao final de cada operação de divisão, é executado o algoritmo *K-means*.

Na segunda fase, a cada iteração é escolhido o par de grupos cujos centros são mais próximos, em distância Euclidiana quadrática. O novo centro de grupo é a média ponderada dos dois centros de grupo envolvidos, os quais são removidos do conjunto de centros de grupo. A cada união o algoritmo *K-means* é executado.

As implementações *fuzzy* dos algoritmos *K-means*, ISODATA e CLUSTER incluem ainda, como entrada a ser fornecida pelo usuário, o escalar  $m > 1$ , que é o expoente para o caso *fuzzy*. Como os algoritmos *fuzzy* geram uma matriz de relacionamentos, é definida uma heurística para forçar, ao final, uma saída rígida, ou seja, um único rótulo para cada *pixel*, a partir da matriz de relacionamentos: a cada *pixel* é atribuído o rótulo correspondendo ao grupo para o qual apresenta o maior grau de relacionamento.

O algoritmo divisivo descrito na Seção 2.2.2, em sua implementação, tem como único limiar a ser fornecido pelo usuário, o número de grupos  $K$ . O mesmo  $K$  controla o número de iterações do algoritmo, já que as divisões ocorrem até que se obtenha o número de grupos

desejado  $K$ . A matriz de distâncias entre todos os *pixels* é gerada utilizando como medida a distância Euclidiana quadrática.

A cada iteração do algoritmo divisivo, ocorre a busca pelo grupo a ser dividido, o grupo “restante”, que é o grupo que possui a maior distância média entre seus elementos. Obtido o grupo “restante”, cada elemento deste grupo é testado para fazer ou não parte do novo grupo.

## 4.2 RESULTADOS OBTIDOS

Numa primeira fase, os algoritmos implementados são testados para as imagens sintéticas e respectivas imagens reais das Figuras 11 e 12. A Seção 4.2.1 mostra os resultados obtidos da execução desses algoritmos. Numa fase seguinte, as imagens segmentadas resultantes são submetidas à rotina Labeling do software Khoros, que faz uma nova rotulação dos *pixels* dessas imagens. Esses resultados são exibidos na Seção 4.2.2.

FIGURA 11 - IMAGENS (A) SINTÉTICA E (B) REAL DE AGPART2-1

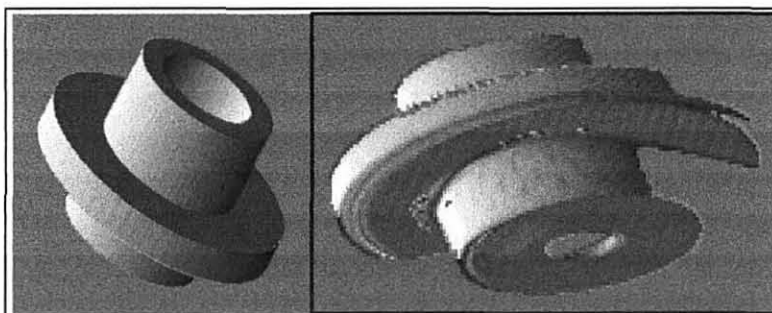
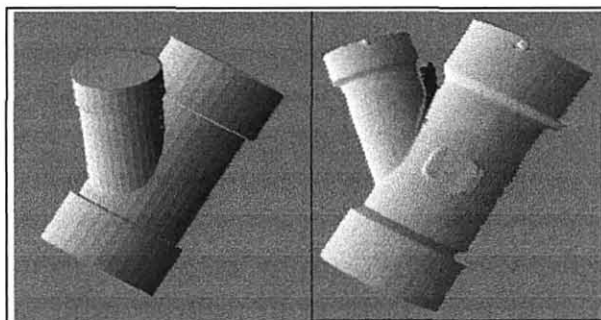


FIGURA 12 - IMAGENS (A) SINTÉTICA E (B) REAL DE BIGWYE-1



#### 4.2.1 Primeira Fase

O algoritmo *K-means* é executado para todas as imagens, sendo 50 o número máximo de iterações. Por considerar visualmente mais adequados, o número de grupos  $K$  utilizado nos testes para as imagens da Figura 11, está na faixa de 10 a 15. Para as imagens da Figura 12, a faixa de valores para o número de grupos  $K$  é de 11 a 16. Algumas das imagens segmentadas obtidas podem ser vistas nas Figuras 13 e 14, onde fica evidente que a segmentação das imagens sintéticas apresenta resultados sensivelmente melhores em relação às imagens reais, devido à presença de grande quantidade de ruído nas imagens reais.

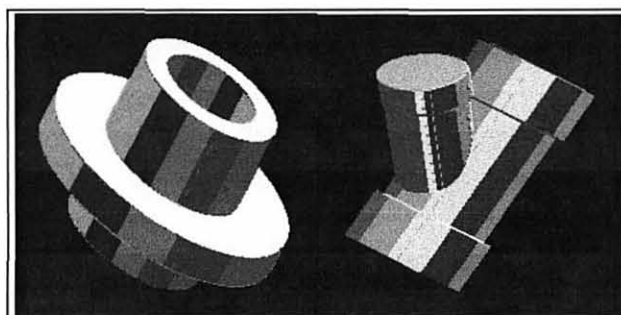
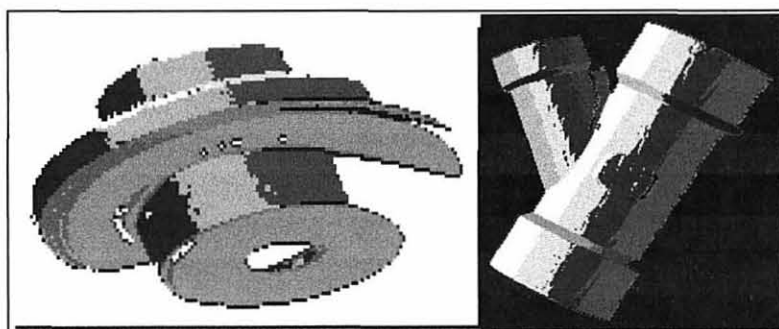
FIGURA 13 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *K-MEANS*: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS

FIGURA 14 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *K-MEANS*:  
(A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS



Os resultados da segmentação com o algoritmo *K-means* são razoáveis, mas apresentam *outliers*, ou seja, pontos fora de contexto, o que é visível nos limites entre uma região e outra, e nas bordas. Em termos de distribuição de grupos, estes pontos formam grupos, algumas vezes muito pequenos, que não refletem a estrutura real dos dados. Deteta-se que esse efeito indesejável tende a aumentar à medida que o número de grupos aumenta.

O algoritmo CLUSTER é, da mesma forma, testado para todas as imagens, considerando 50 iterações. Para as imagens da Figura 11, é usado um  $K=15$ , e para as imagens da Figura 12,  $K=16$ , o que permite a formação de imagens segmentadas de 2 até  $K$  grupos. Os resultados são semelhantes aos obtidos pelo algoritmo *K-means*, porém permitindo a obtenção automática de uma série de agrupamentos de 2 até  $K$  grupos.

As Figuras 15 e 16 apresentam as imagens *agpart2-1* e *bigwye-1*, segmentadas em 10 e 11 grupos, respectivamente. A Figura 17 mostra a seqüência de imagens obtidas da segmentação da imagem sintética *agpart2-1*, de 5 a 12 grupos, onde é possível observar a formação de grupos fora do contexto, os quais ficam visíveis nas regiões de borda.

FIGURA 15 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO CLUSTER: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS

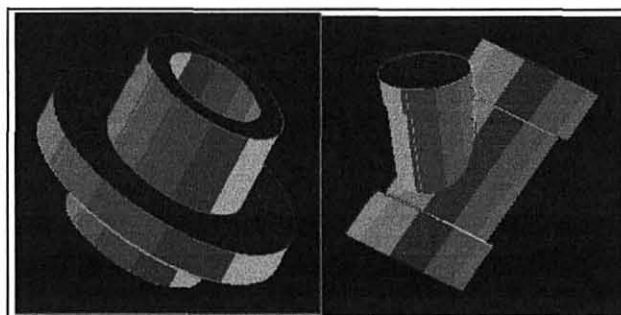
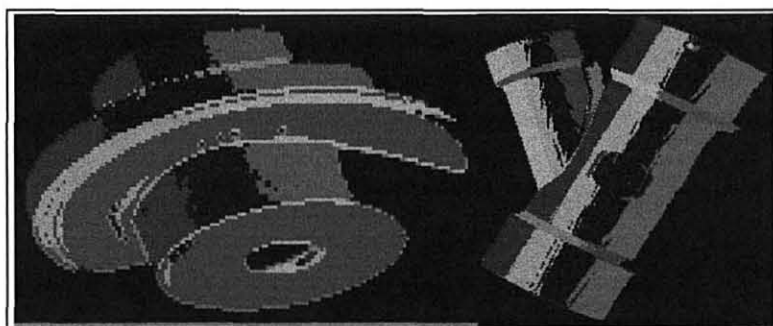


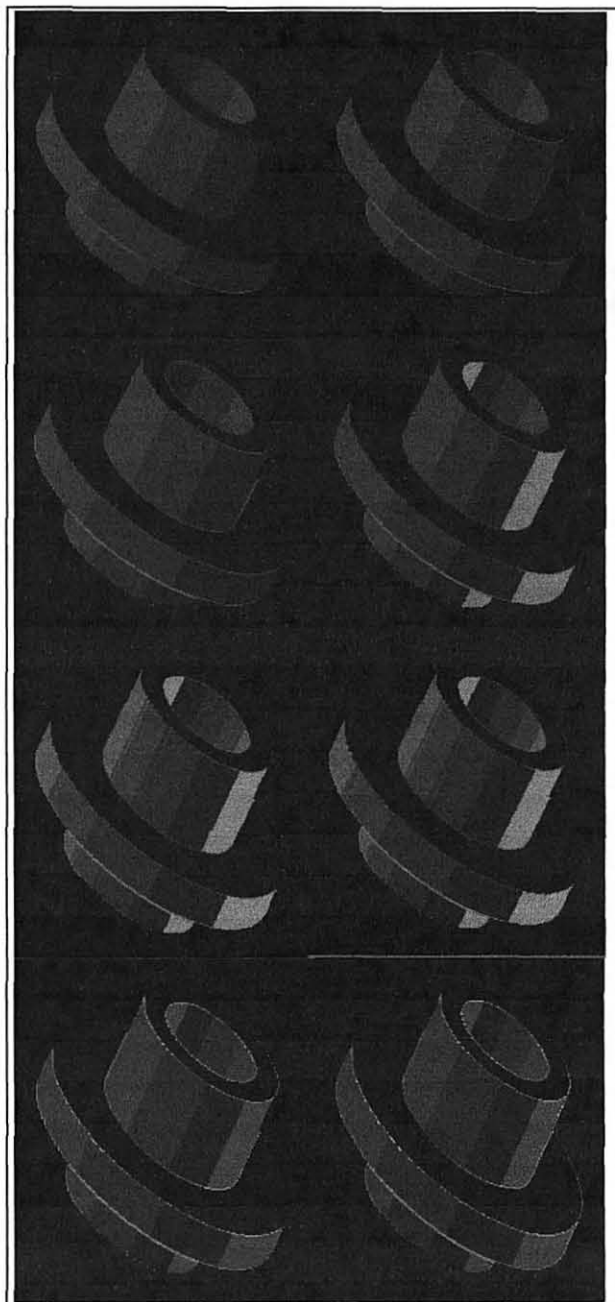
FIGURA 16 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO CLUSTER: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS



Para os testes com o algoritmo ISODATA, o número de grupos segue as mesmas faixas de valores estabelecidos para o algoritmo *K-means*. O número de iterações é ajustado em 30, o limiar de união e divisão em 0,5, e a porcentagem de amostras dentro de um grupo em 1%. Esses dois últimos valores são ajustados empiricamente, por levarem a resultados mais adequados, quando aplicados às imagens em estudo. Valores grandes para o limiar de divisão e união favorecem a união de grupos, e pequenos favorecem a divisão de grupos; logo, 0,5 é um valor intermediário. Paralelamente, o número mínimo de amostras dentro de um grupo, ajustado em 1%, leva a resultados mais adequados, por diminuir a tendência de formação de grupos fora de contexto (*outliers*).



FIGURA 17 - IMAGENS SINTÉTICAS DE AGPART2-1 SEGMENTADAS PELO ALGORITMO CLUSTER EM 5 ATÉ 12 GRUPOS



Assim, o algoritmo ISODATA leva a resultados melhores em relação aos algoritmos *K-means* e CLUSTER, por diminuir esse efeito relacionado a *outliers*, embora não o elimine por

completo. As Figuras 18 e 19 mostram resultados da segmentação com o algoritmo ISODATA.

FIGURA 18 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO ISODATA: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS

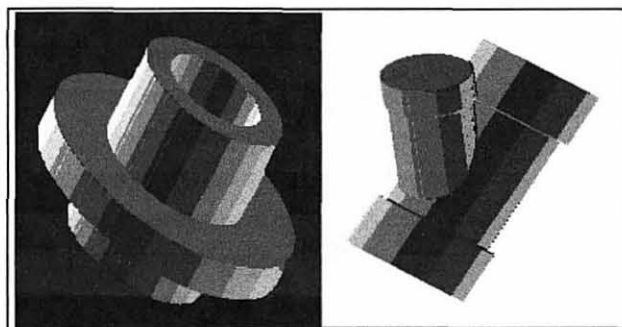
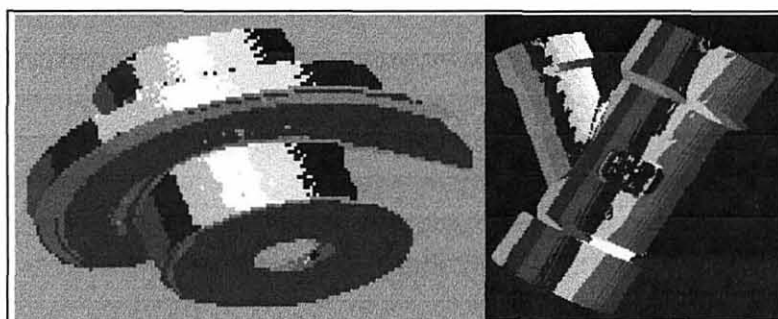


FIGURA 19 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO ISODATA: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS



A versão *fuzzy* do algoritmo *K-means* é testada com as mesmas faixas de valores para o número de grupos  $K$ , utilizadas para o algoritmo *K-means*, porém considerando 20 iterações, devido ao tempo de computação que aumenta consideravelmente. Quanto ao expoente  $m$  para agrupamento *fuzzy*, os testes demonstram que o valor  $m=2$  ou próximo apresenta os melhores resultados. Os testes levam a resultados visualmente superiores aos obtidos pelo algoritmo *K-*

*means*, reduzindo o efeito já mencionado relativo a *outliers*, como pode ser observado nas Figuras 20 e 21.

FIGURA 20 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS*: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS

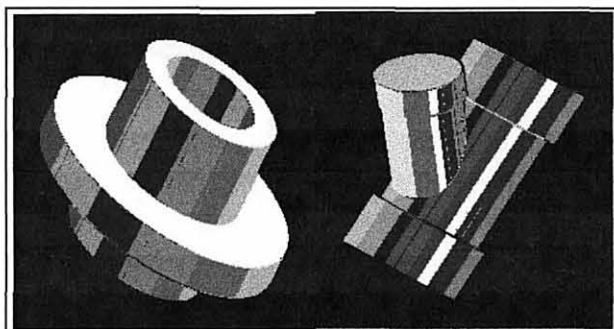
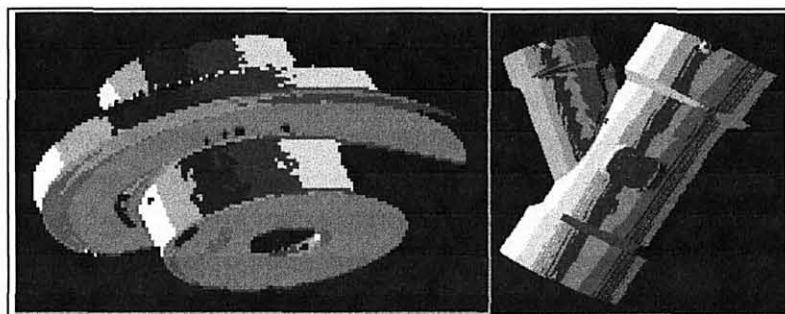


FIGURA 21 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS*: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS



Nos testes com o algoritmo *CLUSTER*, em sua versão *fuzzy*, o número de iterações é igual a 30, o valor de  $m=2$ , e o número máximo de grupos para as diferentes imagens é o mesmo ajustado para os testes do algoritmo *CLUSTER*, para efeito de comparação de resultados. Nota-se que os resultados obtidos são um pouco melhores em relação ao algoritmo

CLUSTER, reduzindo os grupos fora de contexto. Nas Figuras 22 e 23 é possível observar esses resultados.

FIGURA 22 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY* CLUSTER: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS

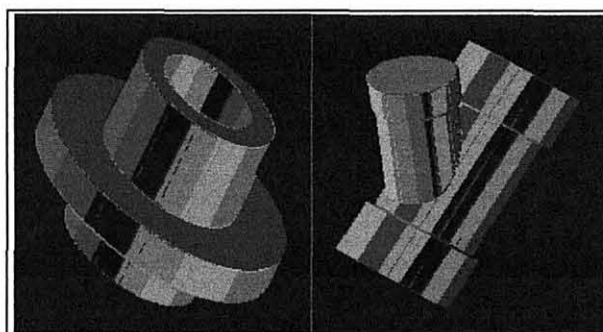
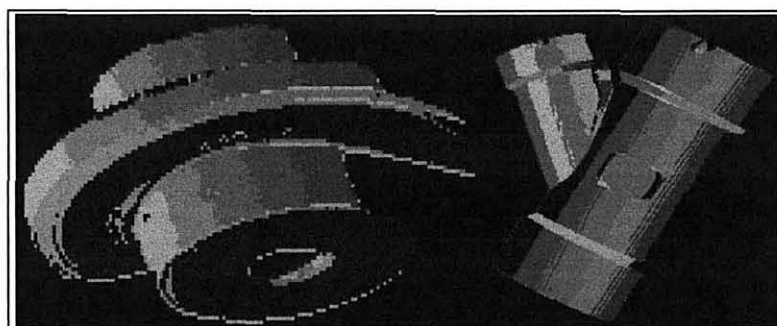


FIGURA 23 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY* CLUSTER: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS



Para o algoritmo *fuzzy* ISODATA, são ajustados os valores de entrada tal como no algoritmo ISODATA, além do valor de  $m=2$ , com 20 iterações. Entretanto, os resultados obtidos podem ser considerados os melhores dentre os algoritmos *fuzzy* testados, pois apresenta o menor número de grupos fora de contexto. Nas Figuras 24 e 25 pode-se observar uma melhora na segmentação das imagens em análise.

FIGURA 24 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY* ISODATA: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS

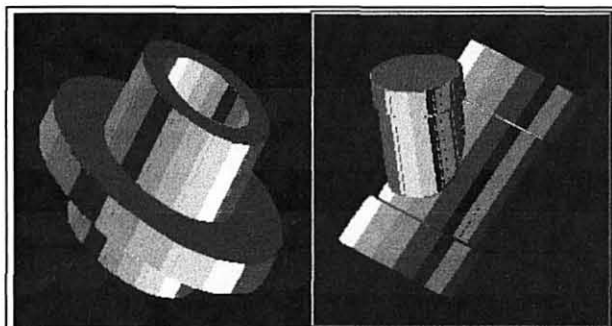
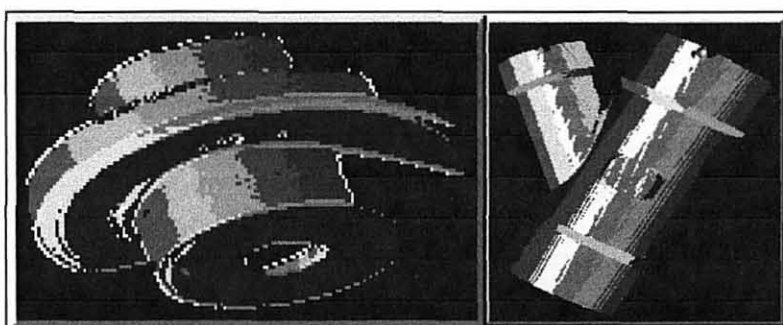


FIGURA 25 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY* ISODATA: (A) AGPART2-1 EM 10 GRUPOS E (B) BIGWYE-1 EM 11 GRUPOS



Já os testes com o algoritmo divisivo encontram como dificuldade um elevado custo computacional em termos de capacidade de memória. Isso devido às dimensões da matriz de proximidades, em função das imagens utilizadas nos testes. Dessa forma, não é possível fazer uma análise do algoritmo divisivo nos mesmos termos que os demais algoritmos, pois os testes ficam restritos a regiões pequenas, extraídas das imagens selecionadas para teste. Uma solução para esse problema é a implementação do algoritmo em paralelo, mas esta abordagem não é desenvolvida neste trabalho.

Paralelamente à análise visual dos resultados obtidos, é realizada a comparação dos algoritmos em termos de tempo de computação e de valor de erro do agrupamento. A Tabela 4 exibe dados comparativos dos algoritmos *K-means*, ISODATA, CLUSTER e suas respectivas versões *fuzzy*, com relação às imagens sintéticas Agpart2-1 (10 grupos) e Bigwye-1 (11 grupos), ambas com dimensões de 240x240 *pixels*.

Para o algoritmo CLUSTER, o número máximo de grupos utilizado é igual a 15 para a imagem Agpart2-1 e igual a 16 para a imagem Bigwye-1, e o valor do erro na Tabela 4 refere-se às imagens segmentadas em 10 e 11 grupos, respectivamente.

TABELA 4 - TEMPO DE COMPUTAÇÃO E VALOR DO ERRO DOS ALGORITMOS

	IMAGEM AGPART2-1		IMAGEM BIGWYE-1	
	TEMPO(seg)	ERRO	TEMPO(seg)	ERRO
K-MEANS	6	25,49	6	30,87
CLUSTER	167	25,97	152	27,04
ISODATA	70	14,86	60	17,99
FUZZY K-MEANS	337	27,32	404	37,64
FUZZY CLUSTER	15675	24,08	18908	25,53
FUZZY ISODATA	6084	23,11	6871	25,33

Quanto ao tempo de computação, evidencia-se que o algoritmo *K-means* é o mais rápido, seguido do ISODATA e do CLUSTER. Quanto às versões *fuzzy*, o tempo aumenta consideravelmente em relação às respectivas versões não *fuzzy*.

Em termos de erro, o algoritmo ISODATA, em suas versões *fuzzy* e não *fuzzy*, apresenta os melhores resultados para as imagens em análise, pois o mesmo apresenta a capacidade de excluir grupos pequenos do processo de segmentação, além da capacidade de divisão e união de grupos.

Embora a análise visual dos resultados obtidos das versões *fuzzy* em relação às respectivas versões não *fuzzy* mostre uma melhora em termos de diminuição dos grupos fora de contexto, os resultados em termos de valor do erro não confirmam tal análise. Tal fato sugere um estudo mais detalhado que venha a explorar a utilização dos algoritmos de agrupamento *fuzzy* na segmentação de imagens de profundidade.

#### 4.2.2 Refinamento da Segmentação

Em todos os casos vistos até aqui, destaca-se que regiões de mesma orientação mas profundidade diferente recebem o mesmo rótulo, o que não é desejável. A solução aponta para uma nova rotulação, a partir da imagem segmentada, atribuindo rótulos diferentes a essas regiões.

A partir dos resultados de segmentação obtidos pelos algoritmos implementados executa-se uma nova rotulação das imagens através de uma rotina chamada Labeling, presente no Khoros. Quando da execução da rotina Labeling, são utilizadas como entrada as imagens segmentadas obtidas na primeira fase. Os demais parâmetros ajustados são os seguintes:

- . Medida de distância: Euclidiana;
- . Conetividade: 4;
- . Tamanho da borda: zero (*pixels* da borda não são atualizados);
- . Quantidade mínima de *pixels* em uma região: 0,1% a 0,5% do total de *pixels* da imagem;
- . Processo de *merging*: automático, com fator de *split and merge* de 0,07;

- Tratamento de regiões pequenas: TRUE (regiões pequenas são unidas a uma região conexa).

As imagens segmentadas resultantes após a execução da rotina Labeling apresentam um número de grupos maior que o valor anterior, pois as regiões desconexas de mesma orientação recebem rótulos diferentes. Além disso, o efeito de má segmentação das bordas e pequenos grupos fora de contexto diminui sensivelmente. Esses resultados podem ser observados nas Figuras 26 a 37.

FIGURA 26 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *K-MEANS* EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 23 GRUPOS E (B) BIGWYE-1 COM 21 GRUPOS

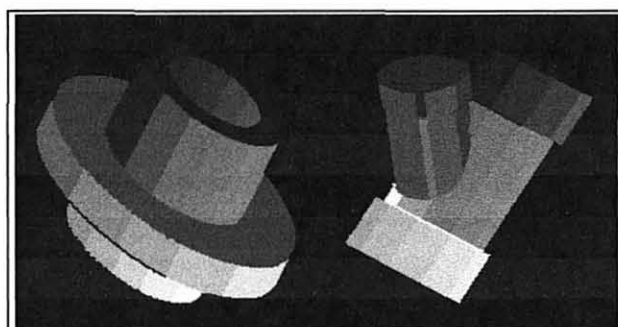


FIGURA 27 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *K-MEANS* EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 16 GRUPOS E (B) BIGWYE-1 COM 33 GRUPOS

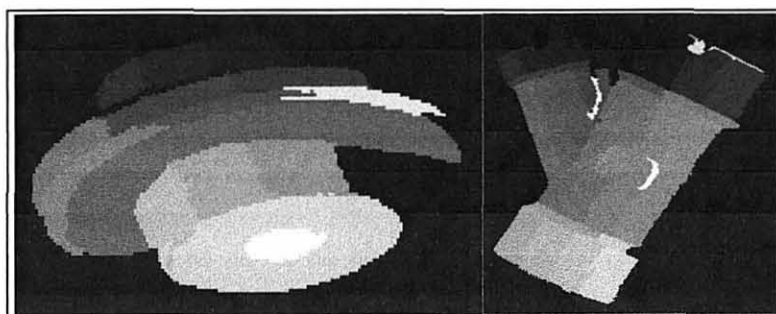




FIGURA 28 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO CLUSTER EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING:  
(A) AGPART2-1 COM 24 GRUPOS E (B) BIGWYE-1 COM 22 GRUPOS

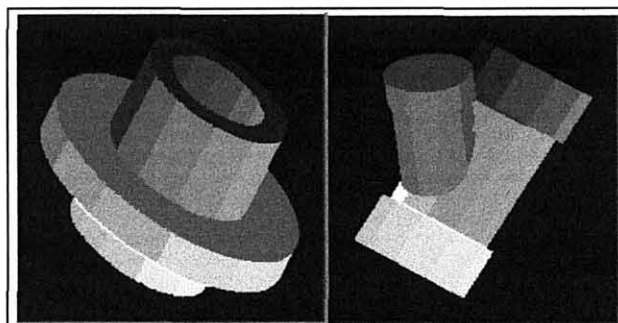


FIGURA 29 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO CLUSTER EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING:  
(A) AGPART2-1 COM 22 GRUPOS E (B) BIGWYE-1 COM 27 GRUPOS

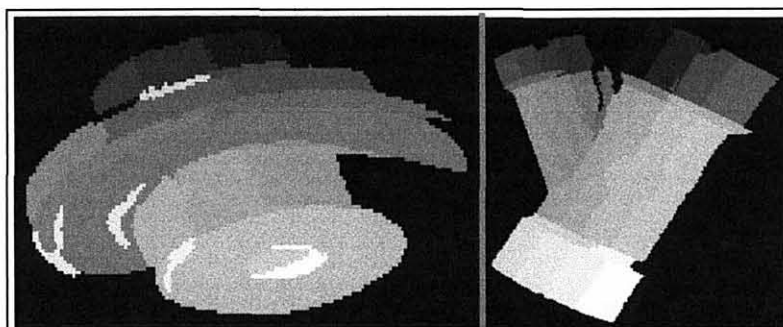


FIGURA 30 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO ISODATA EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING:  
(A) AGPART2-1 COM 16 GRUPOS E (B) BIGWYE-1 COM 21 GRUPOS

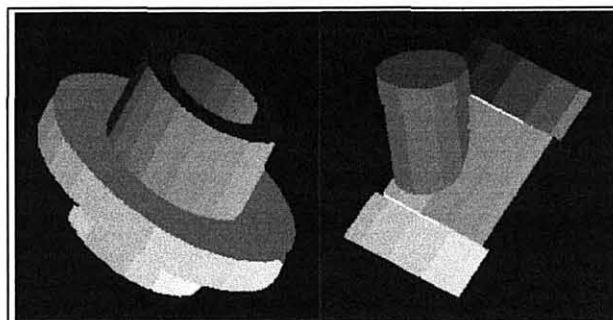


FIGURA 31 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO ISODATA EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 17 GRUPOS E (B) BIGWYE-1 COM 32 GRUPOS

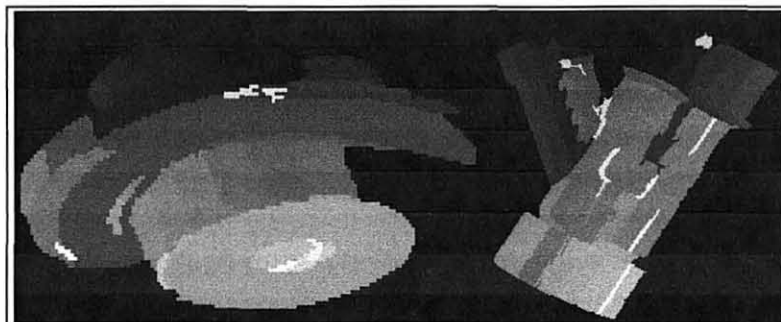


FIGURA 32 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS* EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 26 GRUPOS E (B) BIGWYE-1 COM 39 GRUPOS

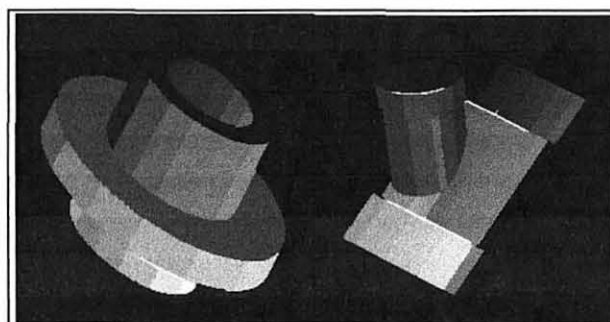


FIGURA 33 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS* EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 18 GRUPOS E (B) BIGWYE-1 COM 14 GRUPOS

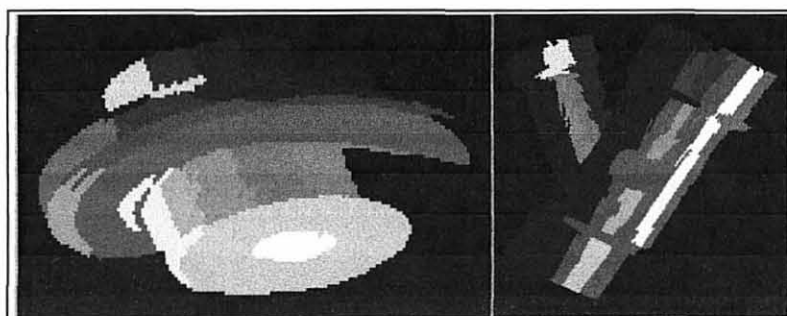


FIGURA 34 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY* CLUSTER EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 28 GRUPOS E (B) BIGWYE-1 COM 30 GRUPOS

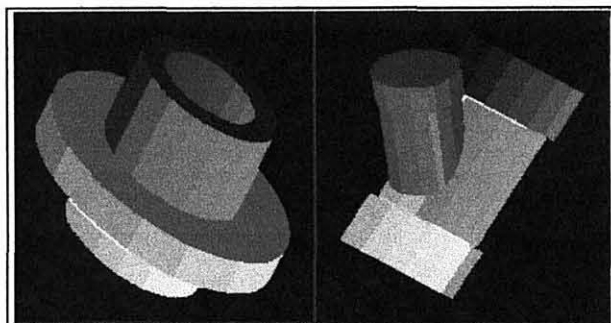


FIGURA 35 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY* CLUSTER EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 39 GRUPOS E (B) BIGWYE-1 COM 41 GRUPOS

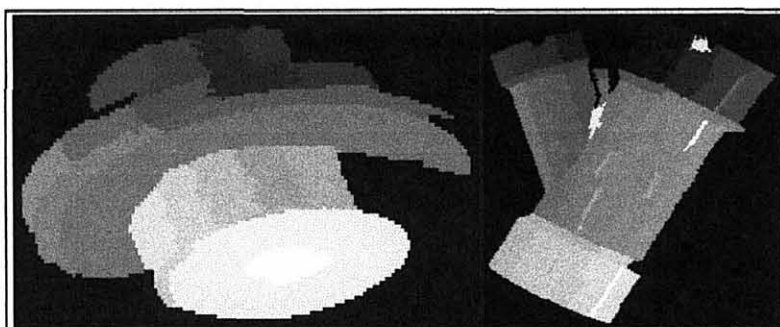


FIGURA 36 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY* ISODATA EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 31 GRUPOS E (B) BIGWYE-1 COM 34 GRUPOS

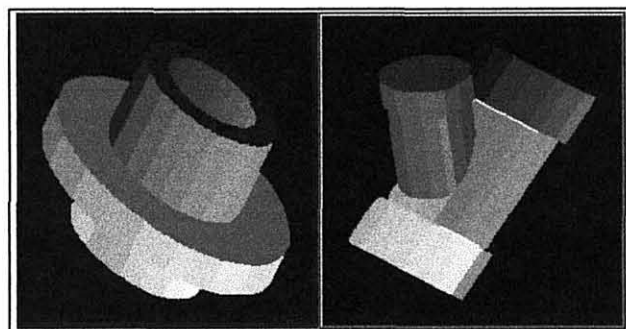
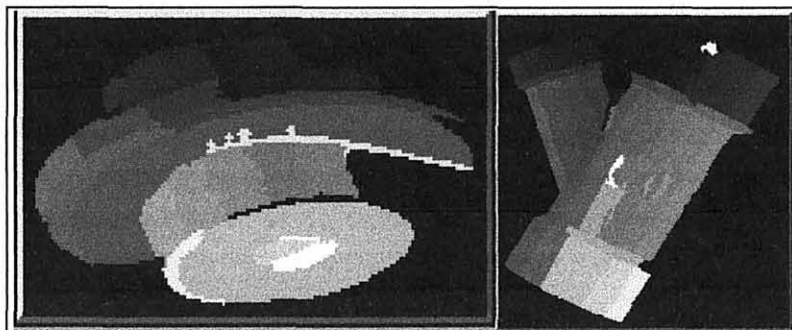


FIGURA 37 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY* ISODATA EM 10 E 11 GRUPOS, RESPECTIVAMENTE, SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 23 GRUPOS E (B) BIGWYE-1 COM 42 GRUPOS



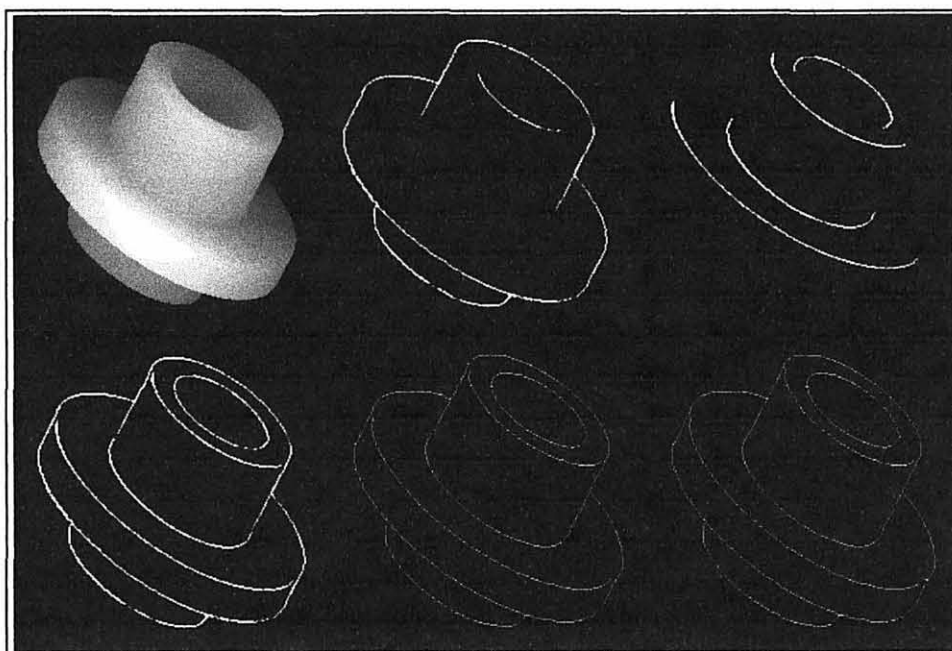
As imagens segmentadas obtidas pela execução da rotina Labeling vêm reforçar a superioridade do algoritmo ISODATA em relação ao *K-means* e ao CLUSTER, tanto na versão *fuzzy* como na versão não *fuzzy*. Essa superioridade diz respeito à obtenção de uma configuração de grupo que melhor represente a imagem inicial.

#### 4.3 ABORDAGEM DESENVOLVIDA

Um resultado melhor para a segmentação de imagens de profundidade pode ser obtido se o conjunto inicial de centros for adequadamente selecionado. A seleção de centros para a inicialização é feita a partir da detecção de bordas, a qual realiza uma pré-segmentação da imagem, fornecendo uma indicação inicial coerente sobre as regiões presentes na imagem (HOFFMAN e JAIN, 1987; YOKOYA e LEVINE, 1989; BHANDARKAR e SIEBERT, 1992).

A partir dos segmentos definidos pela detecção de bordas, é possível selecionar *pixels* desses segmentos, cujos valores vão constituir o conjunto de centros iniciais para o agrupamento. Dessa forma, os algoritmos *K-means* e *fuzzy K-means*, em suas implementações, suportam também como entrada opcional um arquivo de entrada que corresponde a *K* centros de grupo adequadamente selecionados, permitindo uma análise quanto à escolha dos centros em relação a uma seleção aleatória.

FIGURA 38 - FASES DA DETECÇÃO DE BORDAS  
PARA A IMAGEM SINTÉTICA AGPART2-1



Para ilustrar, a Figura 38 exibe as várias fases da detecção de bordas aplicadas à imagem sintética *agpart2-1*, obtidas por rotinas desenvolvidas junto ao software *Khoros*. Seguindo a ordem da esquerda para a direita, (a) refere-se à imagem da coordenada *z*, necessária para a detecção de bordas; (b) e (c) referem-se às imagens de bordas de profundidade e de orientação,

respectivamente; (d) refere-se à soma das imagens (b) e (c); (e) refere-se à imagem obtida pela execução da operação *thinning*, que reduz regiões de borda da imagem às linhas centrais destas bordas; e (f) refere-se à imagem final após a execução da rotina *closing*, que faz o fechamento dos mapas de bordas de profundidade e orientação, além de eventuais falhas nas linhas de borda obtidas (BELLON, SILVA e WEINGAERTNER, 1998; SILVA e WEINGAERTNER, 1998).

Com base na imagem de bordas é possível extrair alguns *pixels* que vão constituir os centros de grupo para a inicialização dos algoritmos de segmentação. Nesta imagem, por exemplo, pode-se extrair um conjunto de dez centros de grupo, assim distribuídos: um centro de cada uma das duas regiões planares e dois centros de cada uma das quatro regiões curvas.

Alguns dos resultados de segmentação obtidos dos algoritmos *K-means* e *fuzzy K-means*, com a inicialização dos centros pela detecção de bordas, são exibidos nas Figuras 39 a 42. As Figuras 43 a 46 mostram os resultados aplicando-se a rotina *Labeling* às imagens segmentadas resultantes.

FIGURA 39 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS

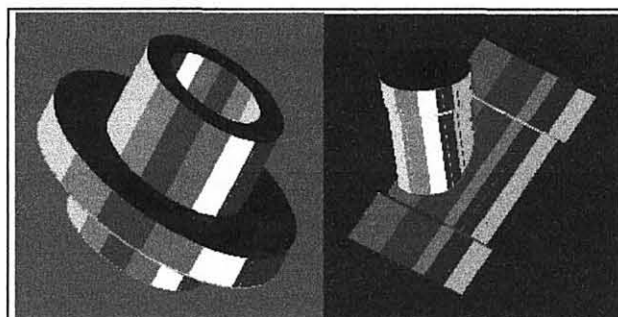


FIGURA 40 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS

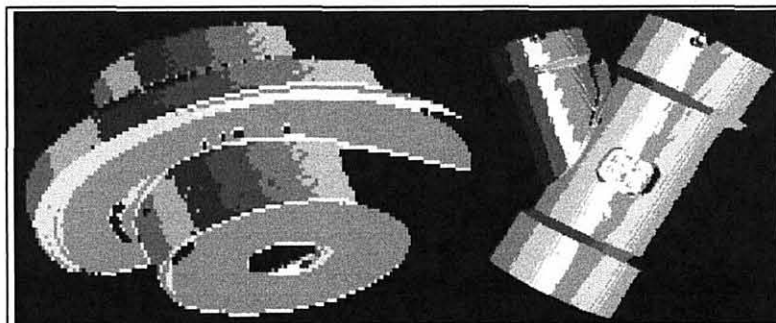


FIGURA 41 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS

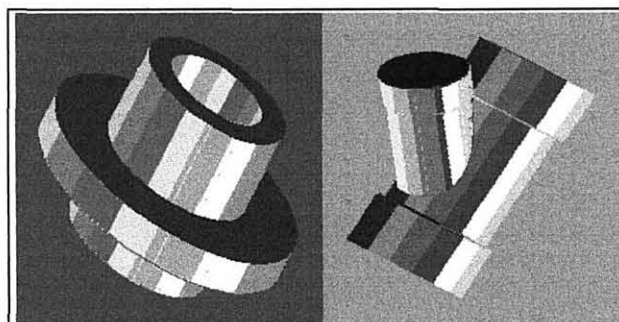


FIGURA 42 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS

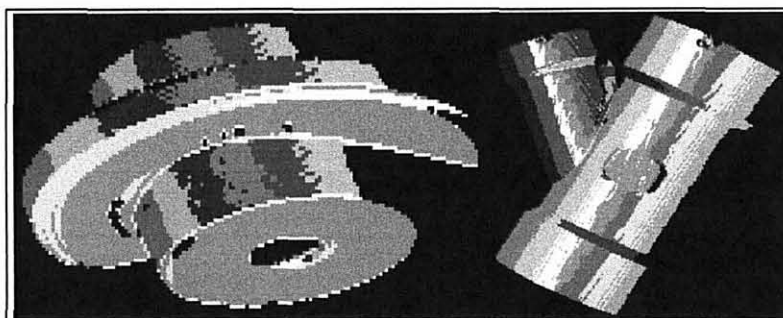




FIGURA 43 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 28 GRUPOS E (B) BIGWYE-1 COM 30 GRUPOS

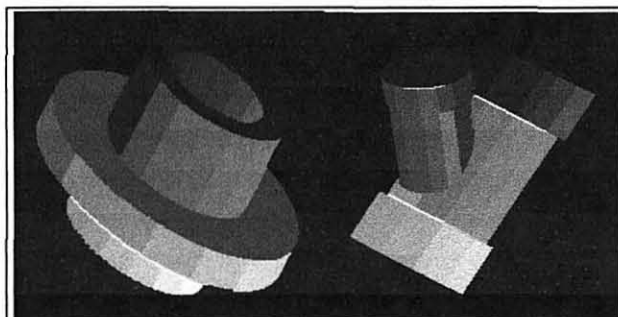


FIGURA 44 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 25 GRUPOS E (B) BIGWYE-1 COM 35 GRUPOS

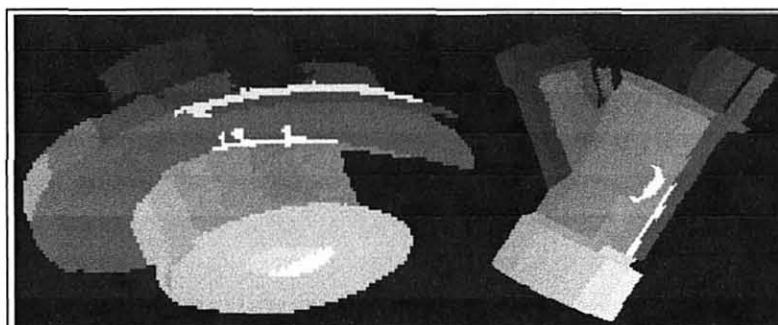


FIGURA 45 - IMAGENS SINTÉTICAS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 29 GRUPOS E (B) BIGWYE-1 COM 26 GRUPOS

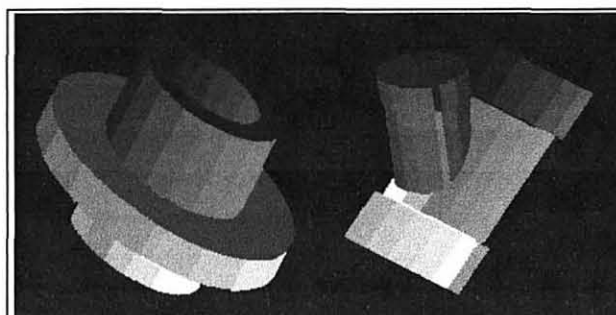
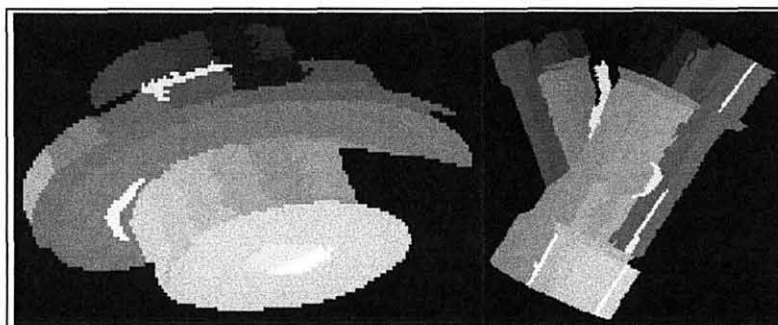




FIGURA 46 - IMAGENS REAIS SEGMENTADAS PELO ALGORITMO *FUZZY K-MEANS* EM 10 E 11 GRUPOS, TENDO COMO ENTRADA OS RESPECTIVOS CONJUNTOS DE CENTROS DE GRUPO, ADEQUADAMENTE SELECIONADOS SEGUIDO DA ROTINA LABELING: (A) AGPART2-1 COM 26 GRUPOS E (B) BIGWYE-1 COM 30 GRUPOS



As comparações entre as imagens segmentadas resultantes dos algoritmos *K-means* e *fuzzy K-means* com conjunto de centros aleatórios e as imagens resultantes dos respectivos algoritmos com conjunto de centros obtidos pela detecção de bordas, demonstram que a escolha adequada do conjunto de centros leva a uma configuração de grupo com um número ainda menor de grupos fora de contexto. Em alguns casos, como pode ser observado na Figura 45 (A), a qualidade da segmentação é muito boa, após a execução da rotina Labeling.

As oito tabelas seguintes exibem valores de erro obtidos para os algoritmos *K-means* (50 iterações) e *fuzzy K-means* (20 iterações) pela Equação 4 (Seção 2.1.1), considerando valores iniciais de centros de grupo aleatórios e obtidos da pré-segmentação por detecção de bordas, para diversas imagens, com diferentes números de grupos.

Observa-se que, para ambos os algoritmos, o valor do erro diminui quando a entrada de centros é obtida da pré-segmentação, para o mesmo número de grupos, demonstrando que o agrupamento assim obtido está mais próximo da solução mínima global.

Somente no caso do algoritmo *fuzzy K-means* com centros definidos pela pré-segmentação, para a imagem real *agpart-2* (Tabela 11), houve aumento no valor do erro. Tal fato pode ser justificado por se tratar de uma imagem real, na qual é difícil a escolha de *pixels*

que não sejam ruído. Neste caso em especial, alguns *pixels* escolhidos para centros provavelmente constituem ruído. Para solucionar esse problema, pode ser realizada uma análise prévia dos valores dos centros, descartando aqueles *pixels* ruidosos. Nessa análise, pode-se utilizar informações de adjacência dos *pixels* sob análise, para verificar se os seus valores diferem de seus vizinhos.

TABELA 5 - VALORES DE ERRO PARA O ALGORITMO *K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM SINTÉTICA AGPART2-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
10	25.49	23.79
13	22.32	21.34
15	21.46	20.65

TABELA 6 - VALORES DE ERRO PARA O ALGORITMO *K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM SINTÉTICA BIGWYE-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
11	30.87	26.58
13	29.74	24.33
16	25.55	22.32

TABELA 7 - VALORES DE ERRO PARA O ALGORITMO *K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM REAL AGPART2-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
10	16.80	15.97
13	15.37	14.02
15	14.95	13.57

TABELA 8 - VALORES DE ERRO PARA O ALGORITMO *K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM REAL BIGWYE-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
11	28.41	26.33
13	26.77	21.81
16	25.89	19.38

TABELA 9 - VALORES DE ERRO PARA O ALGORITMO *FUZZY K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM SINTÉTICA AGPART2-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
10	27.32	24.34
13	24.95	21.40
15	23.59	20.28

TABELA 10 - VALORES DE ERRO PARA O ALGORITMO *FUZZY K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM SINTÉTICA BIGWYE-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
11	37.64	25.17
13	37.18	25.73
16	22.46	21.12

TABELA 11 - VALORES DE ERRO PARA O ALGORITMO *FUZZY K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM REAL AGPART2-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
10	16.12	16.22
13	14.76	17.25
15	14.26	14.71

TABELA 12 - VALORES DE ERRO PARA O ALGORITMO *FUZZY K-MEANS* COM CENTROS ALEATÓRIOS E COM CENTROS DEFINIDOS (IMAGEM REAL BIGWYE-1)

NÚMERO DE GRUPOS	ERRO (CENTROS ALEATÓRIOS)	ERRO (CENTROS DEFINIDOS)
11	29.50	28.49
13	28.37	23.13
16	28.47	20.97

## 5 CONCLUSÃO

Este trabalho apresenta uma discussão sobre métodos e algoritmos de agrupamento, dentro das duas principais categorias: hierárquicos e particionais. Diversos algoritmos são testados utilizando imagens de profundidade, e os resultados são apresentados.

Uma nova abordagem para inicialização dos algoritmos é introduzida. Nessa abordagem, o conjunto inicial de centros de grupo é gerado a partir de uma pré-segmentação por detecção de bordas. Os resultados dos testes demonstram uma melhora significativa na segmentação de imagens de profundidade.

### 5.1 CONTRIBUIÇÕES

Este trabalho faz uma explanação sobre métodos de agrupamento, hierárquicos e particionais, e os principais algoritmos de agrupamento que implementam os métodos. Uma questão é a escolha entre procedimentos hierárquicos ou particionais. Cada qual tem suas

peculiaridades, como a escolha de parâmetros, a apresentação dos resultados, a adequação do algoritmo aos dados. Porém, o fator determinante é a quantidade de dados, definida pelo número de amostras e pelo número de características.

Métodos hierárquicos, incluindo nestes os grafo-teóricos, organizam os dados numa seqüência aninhada de agrupamentos, que são representados na forma de um *dendrogram*. Para se obter um agrupamento do *dendrogram*, é necessário especificar um valor de limiar de similaridade para fazer o corte e se obter uma partição.

Os métodos particionais, ou de minimização de custo, que levam a uma partição dos dados, são mais fáceis de implementar e mais largamente utilizados, porque permitem trabalhar com grandes conjuntos de dados. Como a solução obtida é local, várias heurísticas são usadas para melhorar o resultado do agrupamento, embora não garantam a otimização global.

Foi verificado na prática que os algoritmos podem não gerar os mesmos agrupamentos quando aplicados a um conjunto de amostras. Não há, ainda, como dizer que um certo algoritmo é o “melhor”, mas algumas características foram exaustivamente abordadas neste trabalho, para fins de comparação. Portanto, cabe ao usuário analisar as diferenças entre os vários algoritmos de agrupamento ou mesmo estar preparado em incluir heurísticas aos algoritmos existentes, a fim de torná-los mais adequados ao tipo de dados em análise, já que cada aplicação tem suas próprias necessidades. Assim, a seleção de características, de critérios e metodologia de agrupamento, requer familiaridade com a área na qual os dados se inserem.

Desta forma, através do estudo comparativo de métodos e algoritmos de agrupamento, uma contribuição deste trabalho é servir como referencial para usuários potenciais desses métodos, especialmente na área de processamento de imagens.

Outra contribuição se dá pela introdução de uma nova abordagem para melhoramento do desempenho dos algoritmos de agrupamento *K-means* e *fuzzy K-means* para segmentação

de imagens de profundidade. A abordagem permite a extração do conjunto de centros de grupo através de uma técnica de detecção de bordas, para a inicialização dos algoritmos. Os resultados obtidos experimentalmente demonstram a eficiência da abordagem, por levar a um melhor desempenho dos algoritmos *K-means* e *fuzzy K-means*. Tal melhora ocorre no sentido da obtenção de agrupamentos com um erro menor em comparação aos mesmos algoritmos com inicialização de centros aleatória, isto é, soluções mínimas mais próximas das respectivas soluções mínimas globais.

## 5.2 TRABALHOS FUTUROS

A abordagem desenvolvida para segmentação de imagens de profundidade produz resultados razoáveis, mas pode ser melhorada. Nota-se que, quando as normais à superfície são usadas como características, ocorrem problemas devido aos *pixels* de borda, ocasionando grupos fora de contexto. Assim, um trabalho futuro pode ser feito no sentido de excluir os *pixels* de borda durante o agrupamento, pela análise dos valores de curvatura dos *pixels*.

Quanto à obtenção dos centros de grupo a partir da detecção de bordas para inicializar os algoritmos de agrupamento, pode ser realizada uma análise da variação de curvatura das regiões curvas, de forma a definir quantos centros devem ser considerados para cada uma dessas regiões.

No caso de imagens reais que, em geral, são ruidosas, é conveniente um exame dos valores iniciais de centros, descartando aqueles que constituem ruído. Assim, resultados melhores podem ser obtidos utilizando, junto ao agrupamento, informações de adjacência dos *pixels* em adição ao conjunto característico.

Também é possível realizar a rotulação de alguns *pixels* de cada uma das regiões obtidas por uma pré-segmentação (dados de treinamento), e usar esses *pixels* para influenciar a segmentação dos demais. Atribuindo pesos adequados aos dados de treinamento, pode-se preservar certos grupos, pois os centros de grupo finais são atraídos para os valores de treinamento. Esta idéia pode resolver o problema de equalização dos grupos, próprio dos algoritmos do tipo erro-quadrático (BENSAID et al., 1996).

Finalmente, este trabalho pode ser visto como um módulo de um sistema de visão, mais precisamente o módulo de segmentação. Um trabalho futuro é o desenvolvimento de um módulo subsequente à segmentação, que faz a classificação dos segmentos de superfície obtidos, de acordo com um critério pré-estabelecido. Realizando esse processo para diferentes vistas do objeto, e formando um catálogo das diferentes vistas dos segmentos obtidos, é possível a representação 3D do objeto (RAJA e JAIN, 1994).

## ANEXO 1 - MATRIZES ESPARSAS

Seja um conjunto de  $n$  objetos descritos por amostras  $d$ -dimensionais classificadas em  $K$  grupos. As amostras no  $k$ -ésimo grupo, não normalizadas, em número  $n_k$ , são denotadas pelos vetores (coluna) (DUDA e HART, 1973; DUBES e JAIN, 1988):

$$[ \mathbf{x}_1^{*(k)}, \dots, \mathbf{x}_{n_k}^{*(k)} ]^T, \text{ onde:} \quad (95)$$

$$\mathbf{x}_j^{*(k)} = [ x_{j1}^{*(k)}, \dots, x_{jd}^{*(k)} ]^T. \quad (96)$$

A média da  $i$ -ésima característica para o  $k$ -ésimo grupo é:

$$m_i^{(k)} = (1/n_k) \sum_{j=1}^{n_k} x_{ji}^{*(k)}. \quad (97)$$

O vetor característico médio para o  $k$ -ésimo grupo é:

$$\mathbf{m}^{(k)} = [ m_1^{(k)}, \dots, m_d^{(k)} ]^T. \quad (98)$$

A média associada,  $\mathbf{m}$ , é o vetor médio global para todas as amostras:



$$\mathbf{m} = (1/n) \sum_{k=1}^K n_k \mathbf{m}^{(k)} \quad \text{onde } n = \sum_{k=1}^K n_k. \quad (99) \text{ e } (100)$$

A normalização é obtida pela subtração do vetor médio global de todas as amostras:

$$\mathbf{x}_j^{(k)} = \mathbf{x}_j^{*(k)} - \mathbf{m}. \quad (101)$$

A matriz esparsa  $S$  para a amostra associada é definida como:

$$S = \sum_{k=1}^K \sum_{j=1}^{n_k} (\mathbf{x}_j^{(k)})(\mathbf{x}_j^{(k)})^T. \quad (102)$$

A matriz esparsa para o  $k$ -ésimo grupo é definida como:

$$S^{(k)} = \sum_{j=1}^{n_k} (\mathbf{x}_j^{*(k)} - \mathbf{m}^{(k)})(\mathbf{x}_j^{*(k)} - \mathbf{m}^{(k)})^T. \quad (103)$$

A matriz esparsa intra-grupo (*within-group*)  $S_W$  é definida como a soma das matrizes esparsas dos grupos:

$$S_W = \sum_{k=1}^K S^{(k)}. \quad (104)$$

Finalmente, a matriz esparsa inter-grupo (*between-group*)  $S_B$ , é definida como a matriz esparsa para as médias de grupo:

$$S_B = \sum_{k=1}^K \sum_{j=1}^{n_k} (\mathbf{m}^{(k)} - \mathbf{m})(\mathbf{m}^{(k)} - \mathbf{m})^T = \sum_{k=1}^K n_k \mathbf{m}^{(k)} \mathbf{m}^{(k)T} - n \mathbf{m} \mathbf{m}^T. \quad (105)$$

A forma da primeira equação de  $S_B$  é similar à equação de  $S$  acima.

As três matrizes esparsas são relacionadas de uma maneira simples:

$$\mathbf{x}_j^{(k)} = (\mathbf{m}^{(k)} - \mathbf{m}) + (\mathbf{x}_j^{*(k)} - \mathbf{m}^{(k)}). \quad (106)$$

Formando o produto  $\mathbf{x}_j^{(k)} \mathbf{x}_j^{(k)T}$ , adicionando sobre  $k$  e  $j$ , e realizando de forma que o produto oposto na linha somem para zero, tem-se

$$S = S_B + S_W. \quad (107)$$

Assim, a matriz esparsa total fica dividida nas matrizes esparsas inter-grupo e intra-grupo.

## ANEXO 2 - DISTÂNCIA MAHALANOBIS

A distância Mahalanobis quadrática entre as amostras  $\mathbf{x}_i$  e  $\mathbf{x}_k$ , também usada como medida de distância em agrupamentos do tipo erro-quadrático, é dada por (DUDA e HART, 1973; DUBES e JAIN, 1988; EVERITT, 1993):

$$d(i,k) = (\mathbf{x}_i - \mathbf{x}_k)^T S^{-1} (\mathbf{x}_i - \mathbf{x}_k). \quad (108)$$

onde a matriz  $S$  é a matriz de covariância das amostras definida no apêndice A. A distância Mahalanobis incorpora a correlação entre características e normaliza cada característica para média zero e variância unitária. Se  $S$  é a matriz identidade, a distância Mahalanobis quadrática é a mesma que a distância Euclidiana quadrática.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABDELMALEK, N. Algebraic error analysis for surface curvatures and segmentation of 3D images. *Pattern Recognition*, v. 23, p. 807-817, 1990.
- AL-SULTAN, K. S. A tabu search approach to the clustering problem. *Pattern Recognition*, v. 28, p. 1443-1451, 1995.
- BABU, G. P.; MURTY, M. N. Clustering with evolution strategies. *Pattern Recognition*, v. 27, p. 321-329, 1994.
- BARONTI, S. et al. Variable pyramid structures for image segmentation. *Computer Vision, Graphics and Image Processing*, v. 49, p. 346-356, 1990.
- BELLON, O. R. P. *Imagens de profundidade: segmentação e representação por superfícies planares*. Campinas, 1997. Tese (Doutorado em Engenharia Elétrica) - DCA-FEEC, UNICAMP.
- BELLON, O. R. P.; SILVA, L.; WEINGAERTNER, D. Estudo sobre detecção de bordas em imagens de profundidade. Em: Congresso da Sociedade Brasileira de Computação SBC (1998: Belo Horizonte).
- BENSAID, A. M. et al. Partially supervised clustering for image segmentation. *Pattern Recognition*, v. 29, p. 859-871, 1996.
- BESL, P. J.; JAIN, R. C. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Machine Intell.*, v. 10, p. 167-192, 1988.
- BHANDARKAR, S. M.; SIEBERT, A. Integrating edge and surface information for range image segmentation. *Pattern Recognition*, v. 25, p. 947-962, 1992.
- COLEMAN, G. B.; ANDREWS, H. C. Image segmentation by clustering. *Proceedings of the IEEE*, v. 67, p. 773-785, 1979.
- DUBES, R. C.; JAIN, A. K. *Algorithms for clustering data*. New Jersey: Prentice Hall, 1988.
- DUBES, R.; JAIN, A. K. Clustering and methodologies in exploratory data analysis. In: *Advances in Computers*, v. 19. New York: Academic Press, 1980. p. 113-215.

- DUBES, R.; JAIN, A. K. Clustering techniques: the user's dilemma. *Pattern Recognition*, v. 8, p. 247-260, 1976.
- DUDA, R. O.; HART, P. E. *Pattern classification and scene analysis*. New York: Wiley, 1973.
- EVERITT, B. S. *Cluster analysis*. New York: Wiley, 1993.
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Addison-Wesley, 1992.
- GROSKY, W. I.; JAIN, R. A pyramid-based approach to segmentation applied to region matching. *IEEE Trans. Pattern Anal. Machine Intell.*, v. 8, p. 639-650, 1986.
- HATHAWAY, R. J.; BEZDEK, J. C. Nerf c-means: non-Euclidean relational fuzzy clustering. *Pattern Recognition*, v. 27, p. 429-437, 1994.
- HATTORI, K.; TORII, Y. Effective algorithms for the nearest neighbor method in the clustering problem. *Pattern Recognition*, v. 26, p. 741-746, 1993.
- HOFFMAN, R.; JAIN, A. K. Segmentation and classification of range images. *IEEE Trans. Pattern Anal. Macch. Intell.*, v. 9, p. 608-620, 1987.
- HUBERT, L. J. Some applications of graph theory to clustering. *Psychometrika*, v. 39, p. 283-309, 1974.
- JAIN, R.; KASTURI, R.; SCHUNCK, B. G. *Machine vision*. McGraw-Hill, 1995.
- JOHNSON, S. C. Hierarchical clustering schemes. *Psychometrika*, v. 32, p. 241-254, 1967.
- KAMEL, M. S.; SELIM, S. Z. New algorithms for solving the fuzzy clustering problem. *Pattern Recognition*, v. 27, p. 421-428, 1994.
- KRISHNAPURAM R.; MUNSHI, A. Cluster-based segmentation of range images using differential-geometric features. *Optical Engineering*, v. 30, p. 1468-1478, 1991.
- KURITA, T. An efficient agglomerative clustering algorithm using a heap. *Pattern Recognition*, v. 24, p. 205-209, 1991.
- MACNAUGHTON-SMITH et al. Dissimilarity analysis. *Nature*, v. 202, p. 1034-1035, 1964.
- NEWMAN, T. S.; FLYNN, P. J.; JAIN, A. K. Model-based classification of quadric surfaces. *CVGIP: Image Understanding*, v. 58, p. 235-249, 1993.

- RAJA, N. S.; JAIN, A. K. Obtaining generic parts from range images using a multi-view representation. *CVGIP: Image Understanding*, v. 60, p. 44-64, 1994.
- SABATA, B.; ARMAN, F.; AGGARWAL, J. K. Segmentation of 3D range images using pyramidal data structures. *CVGIP: Image Understanding*, v. 57, p. 373-387, 1993.
- SELIM, S. Z.; ALSULTAN, K. A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, v. 24, p. 1003-1008, 1991.
- SELIM, S. Z.; ISMAIL, M. A. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Machine Intell.*, v. 6, p. 81-86, 1984.
- SILVA, L.; WEINGAERTNER, D. *Deteção de bordas e extração de características em imagens de profundidade*. Curitiba, 1998. Relatório técnico RT002/98 (Bacharelado em Informática) - Departamento de Informática, UFPR.
- TAYLOR, R. W.; SAVINI, M.; REEVES, A. P. Fast segmentation of range imagery into planar regions. *Computer Vision, Graphics and Image Processing*, v. 45, p. 42-60, 1989.
- TSAO, E. C.; BEZDEK, J. C.; PAL, N. R. Fuzzy Kohonen clustering networks. *Pattern Recognition*, v. 27, p. 757-764, 1994.
- VENKATESWARLU, N. B.; RAJU, P. S. V. S. K. Fast ISODATA clustering algorithms. *Pattern Recognition*, v. 25, p. 335-342, 1992.
- YOKOYA, N.; LEVINE, M. D. Range image segmentation based on differential geometry: a hybrid approach. *IEEE Trans. Pattern Anal. Machine Intell.*, v. 2, p. 643-649, 1989.
- ZAHN, C. T. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. On Computers*, v. 8, p. 68-86, 1971.
- ZHANG, Q.; BOYLE, R. D. A new clustering algorithm with multiple runs of iterative procedures. *Pattern Recognition*, v. 24, p. 835-848, 1991.