

1. Classificação <i>INPE-COM.1/TDL</i> C.D.U.: 681.3.01: 519.2		2. Período	4. Critério de Distribuição: interna <input type="checkbox"/> externa <input checked="" type="checkbox"/>
3. Palavras Chaves (selecionadas pelo autor) <i>RECONHECIMENTO DE PADRÕES</i> <i>PROCESSAMENTO DE IMAGENS</i> <i>CLASSIFICAÇÃO POR TABELAS</i> <i>AGREGAÇÃO DE DADOS</i>			
5. Relatório nº <i>INPE-1439-TDL/006</i>	6. Data <i>Março, 1979</i>	7. Revisado por <i>C. R. Souza</i> <i>Celso de Renna e Souza</i>	
8. Título e Sub-Título <i>CLASSIFICAÇÃO DE IMAGENS MULTI-ESPECTRAIS DE RECURSOS NATURAIS USANDO "TABLE LOOK-UP" E AGREGAÇÃO</i>		9. Autorizado por <i>N. Parada</i> <i>Nelson de Jesus Parada</i> <i>Diretor</i>	
10. Setor <i>DSE/DIN</i>	Código <i>2712</i>	11. Nº de cópias	<i>20</i>
12. Autoria <i>Nelson I. Tanaka</i> <i>Renato H.L. Pedrosa</i> <i>Nelson D.A. Mascarenhas</i>		14. Nº de páginas	<i>120</i>
13. Assinatura Responsável <i>Nelson Mascarenhas</i>		15. Preço	
16. Sumário/Notas <i>Visando otimizar o tempo de classificação de imagens multi-espectrais de recursos naturais, obtidas pelo LANDSAT, foram implementados dois métodos, ambos baseados na técnica de "table look-up": o das duas tabelas (Capítulo II) e outro usando a estrutura de dados acessada com "hashing" (Capítulo III). Foi implementado também um sistema de classificação utilizando agregação ("clustering"), segundo o algoritmo ISODATA (Capítulo IV).</i>			
17. Observações <i>Trabalho apresentado como "Trabalho de Graduação" para obtenção de grau no Instituto Tecnológico de Aeronáutica sob a orientação do Dr. Nelson Delfino d'Ávila Mascarenhas.</i>			

ÍNDICE

ABSTRACT	v
<u>CAPÍTULO I - INTRODUÇÃO</u>	1
<u>CAPÍTULO II - MÉTODO DAS DUAS TABELAS</u>	5
2.1 - Introdução	5
2.2 - Redução do espaço espectral	5
2.3 - A classificação	10
2.4 - Descrição dos programas	12
2.4.1 - Introdução	12
2.4.2 - TLUP0	12
2.4.3 - TLUP1	13
2.4.4 - TLUP2	15
2.4.5 - TLUP3	18
2.4.6 - TLUP4	18
2.4.7 - TLUP5	21
2.5 - Resultados e comentários	22
<u>CAPÍTULO III - CLASSIFICAÇÃO USANDO TABELA CONSTRUÍDA POR</u> <u>"HASHING"</u>	27
3.1 - Introdução	27
3.2 - Classificação por um passo usando "hash"	30
3.3 - Níveis de informação	33
3.4 - Colisões	34
3.5 - Tamanho necessário de tabela	34
3.6 - O Sistema MAXHAS	35
3.6.1 - MAXHAS	36
3.6.2 - MAXFIT	36
3.6.3 - CLRTAB	36
3.6.4 - MGL	36
3.6.5 - HASQIO	37
3.7 - Resultados e comentários	38
<u>CAPÍTULO IV - CLASSIFICAÇÃO POR AGREGAÇÃO</u>	41
4.1 - Introdução	41

4.2 - ISODATA	42
4.3 - Descrição de cada passo	44
4.4 - Programas	51
4.4.1 - COMSIG	52
4.4.2 - SELPTO	52
4.4.3 - CLTMAX	52
4.4.4 - CLSISO	53
4.5 - Resultados e comentários	53
<u>REFERÊNCIAS BIBLIOGRÁFICAS</u>	57
<u>APÊNDICE A</u>	A.1
<u>APÊNDICE B</u>	B.1
<u>APÊNDICE C</u>	C.1

ABSTRACT

Two computational methods were implemented, with the objective of optimizing the classification time of earth resources multispectral images, obtained through the Landsat. Both methods were based on the table look-up approach: one uses two tables (Chapter 2) and the other uses a data structure accessed through hashing (Chapter 3). Furthermore, a classification system using clustering was also implemented, based on the ISODATA algorithm.

CAPÍTULO I

INTRODUÇÃO

A cada dia a classificação de recursos naturais, utilizando o processamento de imagens multi-espectrais, adquire maior importância, sob vários aspectos. Como exemplo, atualmente está sendo desenvolvido no INPE (Instituto de Pesquisas Espaciais) do Conselho Nacional de Desenvolvimento Científico e Tecnológico, um projeto de reconhecimento de áreas produtoras de cana-de-açúcar, para todo o Estado de São Paulo. Consequentemente, são cada vez mais necessários sistemas de classificação rápidos, permitindo a otimização do tempo de máquina.

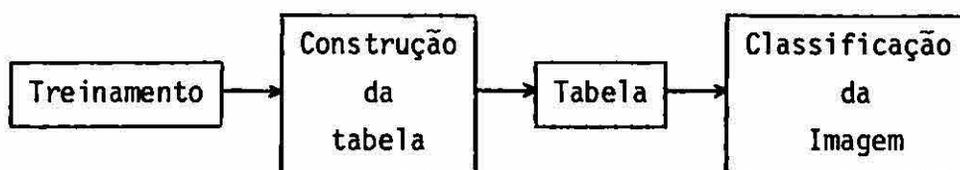
No INPE é utilizado, com mais frequência, o sistema MAXVER (Ref. 1), baseado na suposição de que a distribuição das diversas categorias de recursos naturais é gaussiana. A preferência é motivada por uma maior precisão da classificação, obtida pelo emprego do critério de máxima verossimilhança. O problema que esta técnica apresenta consiste no tempo relativamente longo que é necessário para os diversos cálculos efetuados durante a classificação. A atenção dos pesquisadores foi então atraída para a procura de técnicas que fornecessem alternativas mais eficientes, do ponto de vista de tempo de processamento.

Os pontos da imagem, durante a classificação, sofrem uma série de operações sobre suas coordenadas espectrais, operações estas que são repetidas para cada novo ponto. A informação sobre a classificação dos pontos anteriores é ignorada. A idéia básica dos métodos que empregam tabelas ("TABLE LOOK-UP") é, exatamente, organizar uma tabela rapidamente acessável, com a informação obtida dos pontos já classificados.

O uso de tabelas pode seguir duas linhas diferentes. A primeira, cujo esquema é mostrado na Fig. I.1.a, consiste em se construir a tabela após o treinamento e, posteriormente, utilizá-la na classificação; na segunda, esquematizada na Fig. I.1.b, a tabela vai sendo elaborada durante a própria classificação, sendo acessada para cada novo

ponto para se determinar se ele já está classificado e, caso contrário, classificando-o e guardando-o na tabela. Os dois métodos foram implementados, utilizando-se o próprio MAXVER na elaboração das tabelas. Os Capítulos II e III apresentam os métodos referentes aos esquemas das Figs. I.1.a e I.1.b, respectivamente.

(a)



(b)



Fig. I.1

Deve ser observado que o uso de tabelas só traz economia se o número de pontos com coordenadas diferentes for menor que o número total de pontos da imagem, de modo a compensar o acesso às tabelas, uma vez que esta é construída pelo sistema MAXVER. Empiricamente, verifica-se que existem cerca de 10000 vetores de intensidades diferentes por imagem obtida pelo LANDSAT, para um total de aproximadamente 7×10^6 pontos (Ref. 2).

Ainda na Fig. I.1, é mostrada outra etapa do processo, até aqui ignorada. O treinamento é a fase em que se obtêm os dados nos quais é baseada a separação da imagem em diversas classes, durante a classificação. A terceira parte do presente trabalho consiste na implementação

de uma técnica, que substitui o treinamento supervisionado, por uma partição do espaço espectral feita automaticamente, tornando o processo não supervisionado. Esta técnica é conhecida como agregação ("CLUSTERING").

CAPÍTULO II

MÉTODO DAS DUAS TABELAS

2.1. INTRODUÇÃO

Este método, apresentado em 1971 (Ref. 3), surgiu como uma das primeiras tentativas no sentido de se substituir os extensos cálculos dos métodos tradicionais de classificação (Refs. 1 e 4), por uma técnica que explorasse eficientemente o espaço de memória disponível, até então praticamente ocioso.

Como foi mencionado no primeiro capítulo, um dos caminhos possíveis no emprego de tabelas consiste em montar a tabela e, posteriormente, utilizá-la na classificação (Fig. I.1.a). A primeira idéia seria a de classificar o espaço espectral todo e guardar esta informação de modo que ela fosse rapidamente acessável. Mesmo considerando que o espaço espectral possa ser reduzido de 256 (8 "bits") para 64 (6 "bits") níveis por canal, uma vez que a imagem enviada pelo LANDSAT é caracterizada por 6 "bits", seria necessário acumular informação para cerca de $1,6 \times 10^7$ (64^4) vetores de intensidades diferentes. Como esta informação deve ficar na memória de massa do computador, torna-se necessária uma redução do espaço espectral.

2.2. REDUÇÃO DO ESPAÇO ESPECTRAL

A idéia é procurar uma maneira de minimizar a memória exigida. Experimentalmente, verifica-se que cada classe ocupa uma parcela reduzida do espaço de características. A Tab. II.1 apresenta o histograma obtido a partir do treinamento de área cultivada. Nota-se que o número de níveis ocupados por canal é relativamente pequeno. Existem, na literatura (Ref. 2), estudos estatísticos cujos resultados indicam que uma imagem do LANDSAT contém em torno de 10000 pontos com coordenadas diferentes. A partir destes dados, tem-se a indicação do caminho a ser seguido. O primeiro passo consiste em se preocupar apenas com as regiões de interesse para cada classe. A Fig. II.1 apresenta, do lado esquerdo, um

esquema simplificado para dois canais apenas.

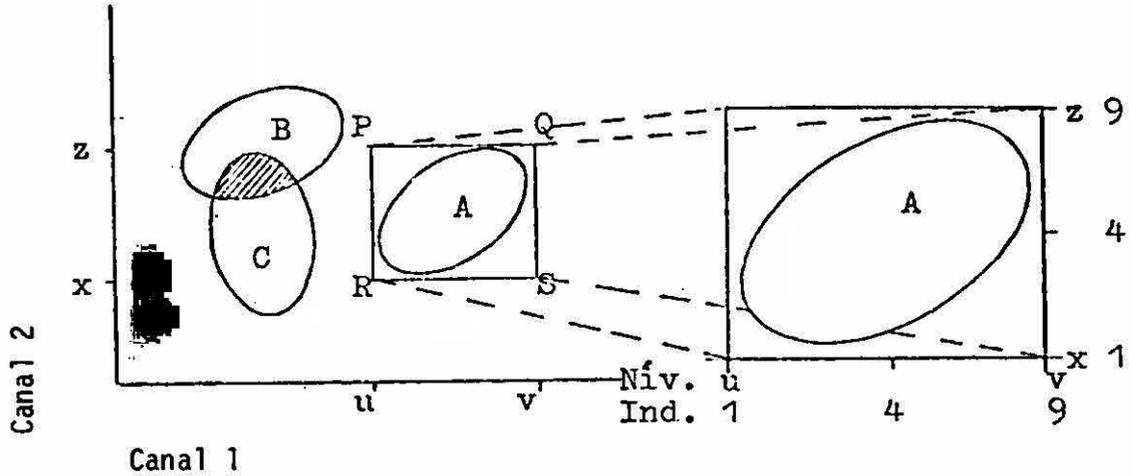


Fig. II-1

As classes estão representadas por suas bordas, contendo os pontos do espaço espectral que as caracterizam. A região de interesse para a classe A é o retângulo PQRS, ao qual estão associados os níveis u, v no canal 1 e x, z no canal 2. Vale notar que no caso quadri-dimensional se tem um hiper-paralelepípedo. Guardar estas regiões como estão é inconveniente, dado o tamanho imprevisível que cada uma pode assumir. Efetua-se, então, uma transformação sobre os níveis do espaço espectral, representada no lado direito da Fig. II.1. A transformação é realizada com a intenção de se obter em regiões de mesmas dimensões para todas as classes. Os níveis contidos entre os limites da região adquirem valores de 1 a 9, homogeneamente distribuídos. Aos pontos exteriores é atribuído o valor 0. A escolha do número 9 como base para a normalização se deve a dois fatores:

- necessidade de se economizar a memória;
- o número médio de níveis ocupados por canal.

Assim se obtêm a primeira tabela, dependente da classe e do canal, que atribui um indicador a cada nível. A Tab. II.2 apresenta a TABELA I referente à classe cujo histograma foi apresentado na Tab. II.1. Antes de prosseguir, cabe notar que, sendo N o número máximo de classes e M o número de níveis, o espaço de memória necessário para a TABELA I é de $4 \times N \times M$ "bytes" ou $2 \times N \times M$ palavras do computador PDP-11.

Nív.	Can. 1	Can. 2	Can. 3	Can. 4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	4	0	0
7	0	5	0	0
8	3	7	0	0
9	7	19	0	2
10	10	37	0	1
11	46	30	0	2
12	49	43	1	3
13	25	24	4	7
14	18	4	9	8
15	17	3	14	7
16	4	2	45	30
17	0	1	50	45
18	0	0	39	42
19	0	0	15	26
20	0	0	1	6
21	0	0	1	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0

TABELA II.1 - Histograma

TABELA II.2 - Tabela I

Nív.	Can. 1	Can. 2	Can. 3	Can. 4
2	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	1	0	0
7	0	2	0	0
8	1	3	0	0
9	2	3	0	1
10	3	4	0	2
11	4	5	0	3
12	5	6	1	3
13	6	6	2	4
14	7	7	3	5
15	8	8	4	6
16	9	8	4	6
17	0	9	5	7
18	0	0	6	8
19	0	0	7	8
20	0	0	8	9
21	0	0	9	0
22	0	0	0	0

Ainda observando a Fig. II.1, nota-se que as regiões correspondentes às classes B e C possuem área comum, sendo necessário estabelecer uma linha de separação (uma hiper-superfície, no caso quadri-dimensional) entre as classes cujas regiões se interpenetram. Além disso, o retângulo é uma aproximação muito grosseira para ser usada na classificação.

No Capítulo I, o sistema MAXVER foi apresentado como sendo o que fornece a maior confiabilidade na classificação, possibilitando, então, o refinamento exigido. O resultado é então guardado numa segunda tabela, cuja entrada é feita com os indicadores obtidos através da TABELA I. Como os indicadores variam de 0 a 9, o espaço de memória é 10^4 e, interessando apenas saber se o ponto pertence ou não à classe considerada, basta que a TABELA II indique a presença ou não da classe, para tal sendo suficiente um "bit". Assim, para um número máximo de 8 classes, basta um espaço de 10^4 "bytes" de memória, cada "byte" apontada para um conjunto de quatro indicadores, cada "bit" correspondendo a uma classe. A Tab. II.3 mostra um exemplo bi-dimensional da segunda tabela. Os pontos representados por "1" formam a região que realmente caracteriza a classe correspondente.

TABELA II.3 - Exemplo bi-dimensional da Tabela II

Indicadores do canal 2	9	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	1	1	1	0
	6	0	0	0	1	1	1	1	1	0
	5	0	0	1	1	1	1	1	1	0
	4	0	1	1	1	1	1	1	0	0
	3	0	1	1	1	1	1	1	0	0
	2	0	0	1	1	1	1	0	0	0
	1	0	0	0	1	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9
	Indicadores do canal 1									

Resta ainda resolver o problema do número de níveis. Como foi observado em 2.1, são suficientes 64 níveis para caracterizar a imagem e, além disso, o tamanho da TABELA I depende diretamente deste número. Feita então esta redução, o espaço de memória total ocupado pelas tabelas

é de 2048 mais 10000 "bytes", sendo possível a implementação no PDP-11.

Tendo em vista a finalidade de se reduzir o tempo de processamento, o tempo de construção das tabelas deve ser pequeno em relação ao necessário para a classificação que é atualmente feita. O primeiro motivo para que assim seja já foi mencionado no primeiro capítulo. O número de pontos classificados durante a elaboração das tabelas é bem menor que o da imagem total. Em segundo lugar, o tempo gasto na operação de ENTRADA/SAÍDA (cerca de 20% do tempo total de processamento), não existe aqui, as tabelas sendo completamente determinadas dentro do programa, sem usar nenhuma rotina auxiliar.

2.3. A CLASSIFICAÇÃO

O algoritmo de classificação é mostrado, na forma de diagrama de fluxo, na Fig. II.2. O primeiro passo consiste em supor uma terminada classe. Isto é mais importante do que pode parecer à primeira vista. Como a classificação é feita sequencialmente sobre pontos adjacentes, a primeira classe a ser testada é aquela à que foi atribuído o ponto anterior, exceção feita ao caso do ponto contíguo ser considerado não-classificado. Esta medida se deve ao fato de que se deve esperar a existência de áreas relativamente homogêneas, agrupadas numa mesma classe. Por outro lado, a sequência de teste pode ser determinada a partir de um conhecimento "a priori" da probabilidade de ocorrência de cada classe, na região geográfica considerada (ver resultados (2.5)), para comentários a cerca destes dois procedimentos.

A seguir, a TABELA I, acessada através da classe e das coordenadas espectrais do ponto, fornece quatro indicadores, um para cada canal. Estes indicadores apontam para um "byte" da TABELA II, cujo "bit" referente à classe assumida é então testado. Caso a resposta ao teste seja afirmativa ("bit" aceso), o ponto é atribuído à classe; caso contrário, a próxima classe da sequência é suposta e assim por diante. Se a sequência toda for percorrida sem resultado positivo, o ponto é considerado não-classificado.

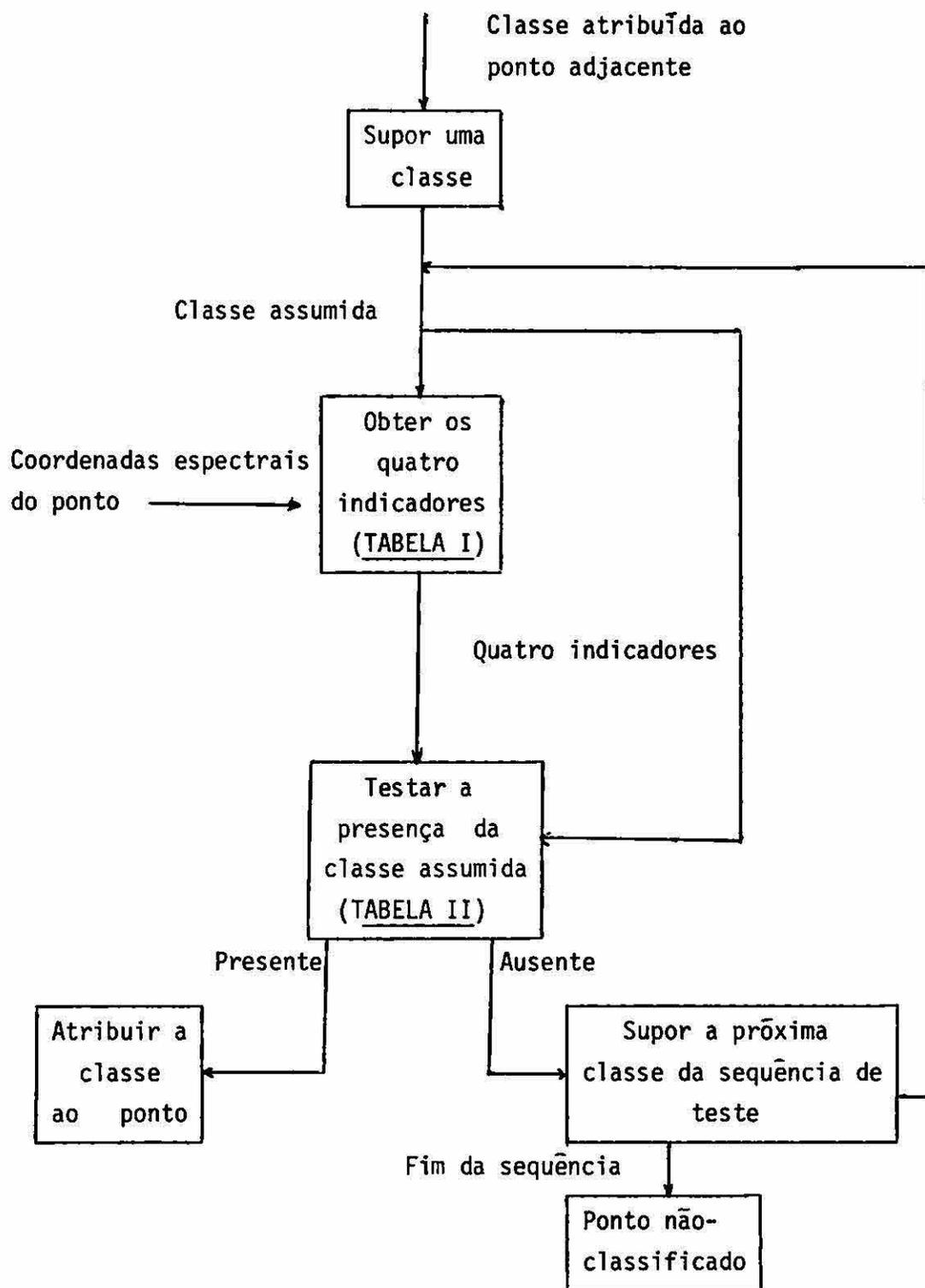


Fig. II.2 - Diagrama de Fluxo da Classificação.

2.4. DESCRIÇÃO DOS PROGRAMAS

2.4.1. INTRODUÇÃO

O método descrito foi implementado no computador PDP-11/45. As linguagens utilizadas foram FORTRAN e ASSEMBLY, a segunda quando o fator tempo se tornou crítico. O sistema se compõe de seis programas:

TLUP0 - Inicialização

TLUP1 - Treinamento e obtenção de parâmetros

TLUP2 - Tabelas

TLUP3 - Classificação da tela

TLUP4 - Classificação do disco

TLUP5 - Armazenamento para continuação

A seguir é apresentada a descrição dos programas e das rotinas que cada um emprega.

2.4.2. TLUP0 - INICIALIZAÇÃO

Programa que inicializa os parâmetros independentes e, em caso de continuação, lê os parâmetros de fita e de arquivo do disco. O nome do arquivo é lido, pelo programa, do terminal. A Fig. II.3 apresenta o diagrama de blocos de TLUP0.

Rotina utilizada:

PPSUB - Leitura de fita

2.4.3. TLUP1 - TREINAMENTO E OBTENÇÃO DE PARÂMETROS

Programa que realiza o treinamento das classes e calcula vários parâmetros necessários para a elaboração das tabelas. O programa

é rodado para cada classe, podendo a amostra ser adicionada a uma classe já existente (ver Apêndice A).

Rotinas utilizadas:

TLPAR - Rotina responsável pelo treinamento, acumulação da média, do histograma e da matriz de auto-correlação. Basicamente, é equivalente à rotina DISTR1 do sistema MAXVER, acrescida da obtenção do histograma;

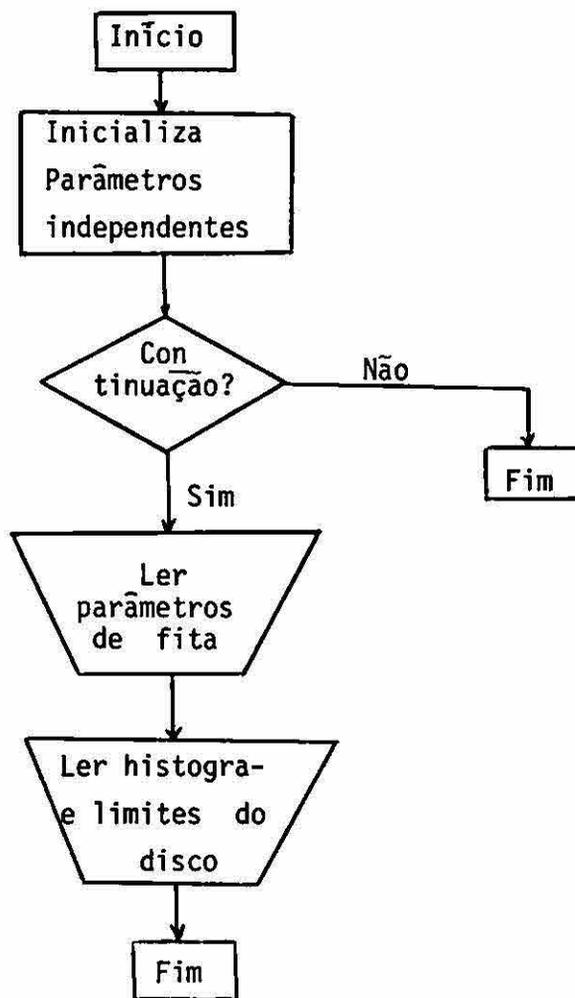


Fig. II.3 - TLUP0

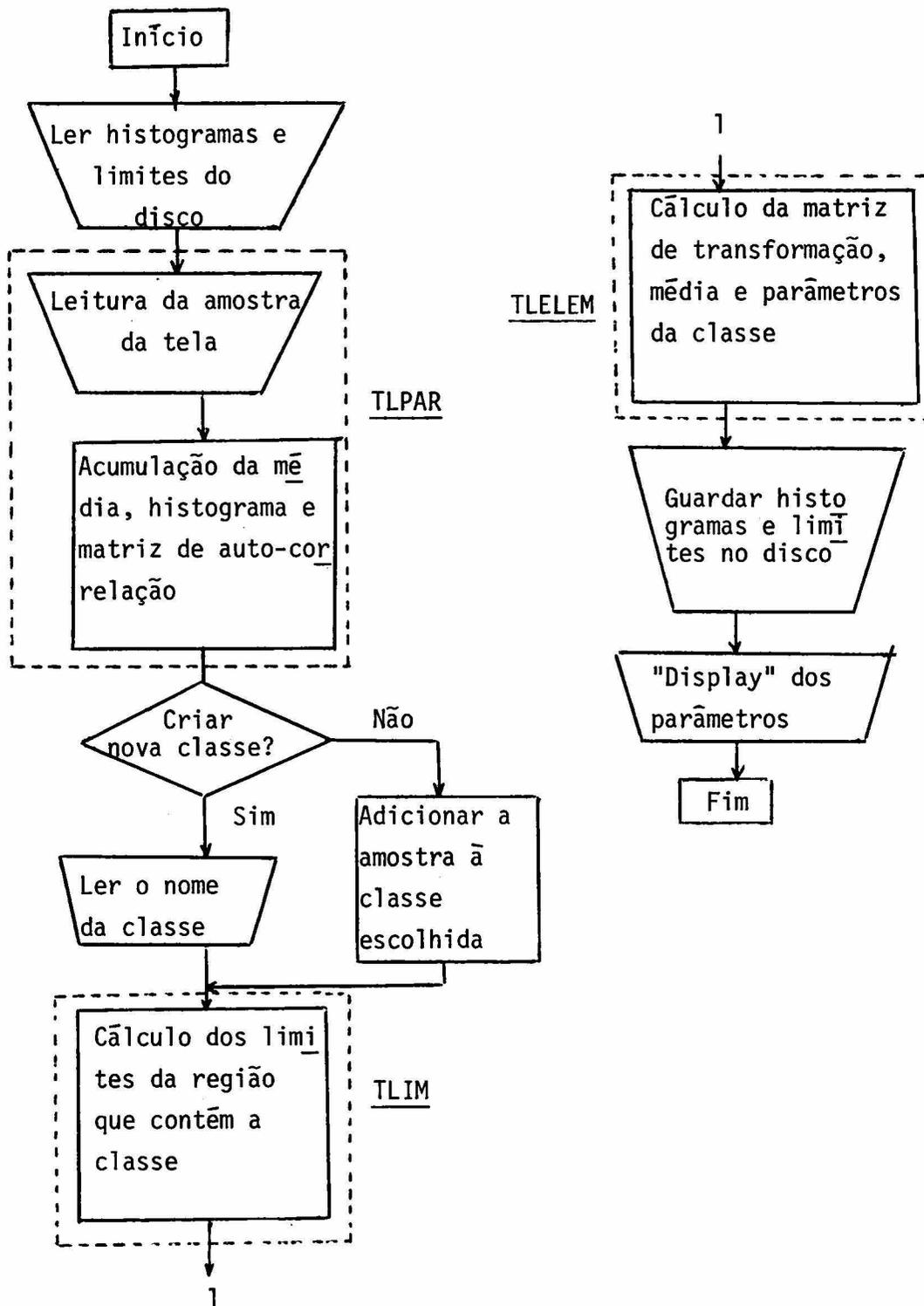


Fig. II.4 - TLUP1

TLIM - Calcula os limites da região de interesse da classe, a partir do histograma. Não deixa de fora nenhum ponto da amostra. Nas Tabs. II.1 e II.2 pode-se notar que todos os níveis que estão ocupados na Tab. II.1 são cobertos pelos indicadores da TABELA I, mostrada na Tab. II.2;

TLELEM - Calcula as transformadas necessárias para a classificação pelo critério de máxima verossimilhança. É equivalente à rotina ELEM1 do sistema MAXVER;

INTCNV - Faz a aproximação de real para o inteiro mais próximo.

A Fig. II.4 mostra o diagrama de blocos de TLUP1.

2.4.4. TLUP2 - TABELAS

O programa monta as tabelas a partir dos dados obtidos por TLUP1. A TABELA I é elaborada pelo programa principal. Os limites em que estão confinadas as regiões de interesse das classes são lidos do disco e a transformação no espaço espectral é então efetuada. A rotina MONTB2 constrói a TABELA II, pelo critério de máxima verossimilhança, sendo necessário entrar com o limiar a ser utilizado. O valor deste limiar determina o tamanho do hiper-elipsóide. Um valor típico é 5.0, usado também pelo sistema MAXVER. Após a TABELA II ter sido completada, é necessário entrar com o nome do arquivo do disco onde serão guardadas as tabelas, ou as tabelas e os parâmetros, caso se vá classificar no disco. O nome é da forma nome.TAB, onde nome tem um máximo de seis letras. No caso de classificação no disco, é possível se classificar na tela, o arquivo sendo o mesmo. As Figs. II.5 e II.6 apresentam os diagramas de blocos de TLUP2 e MONTB2.

Rotinas utilizadas:

INTCNV - Aproximação inteiro-real;

MONTB2 - Esta rotina classifica a menor região do espaço espectral que contém todas as regiões de interesse, acumulando o resultado na TABELA II e é programada em linguagem ASSEMBLY, devido à neces

sidade de se otimizar o tempo de processamento. Para cada ponto do espaço encontram-se as classes para as quais todos os indicadores (TABELA I) são diferentes de 0. Classifica-se por máxima verossimilhança, apenas para estas classes, e o resultado é guardado na TABELA II.

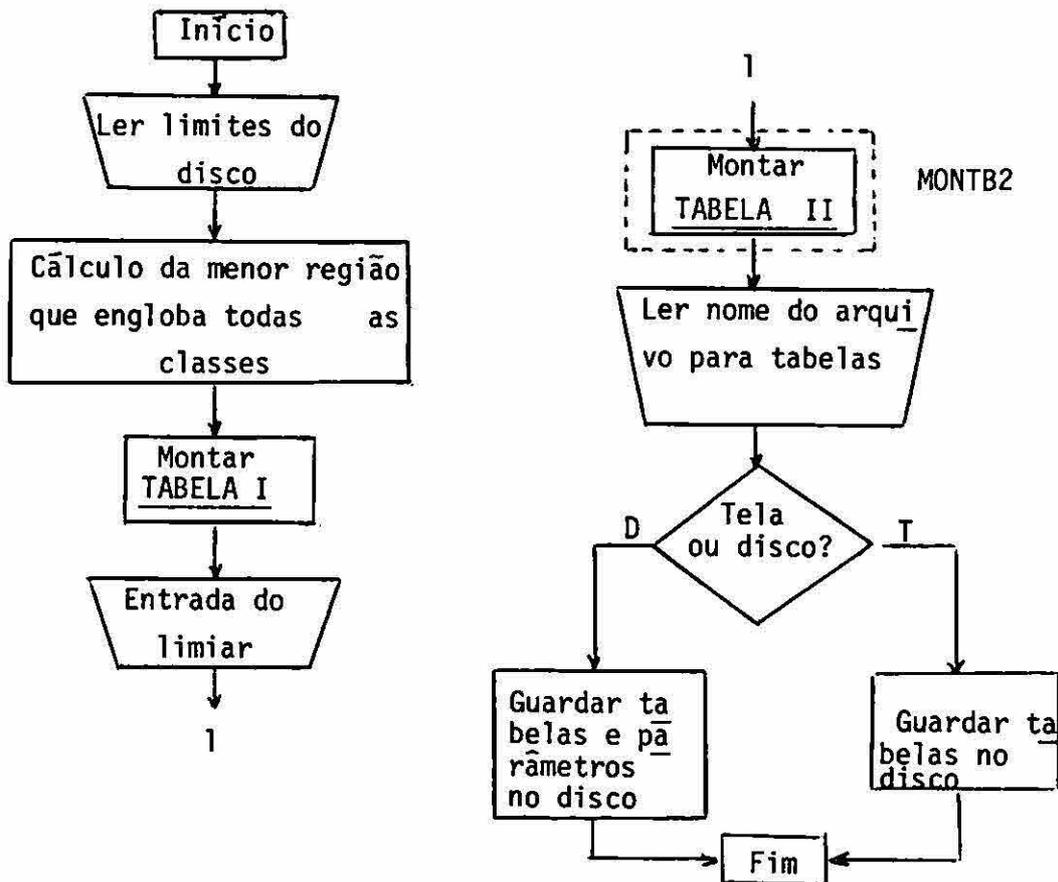


Fig. II.5 - TLUP2

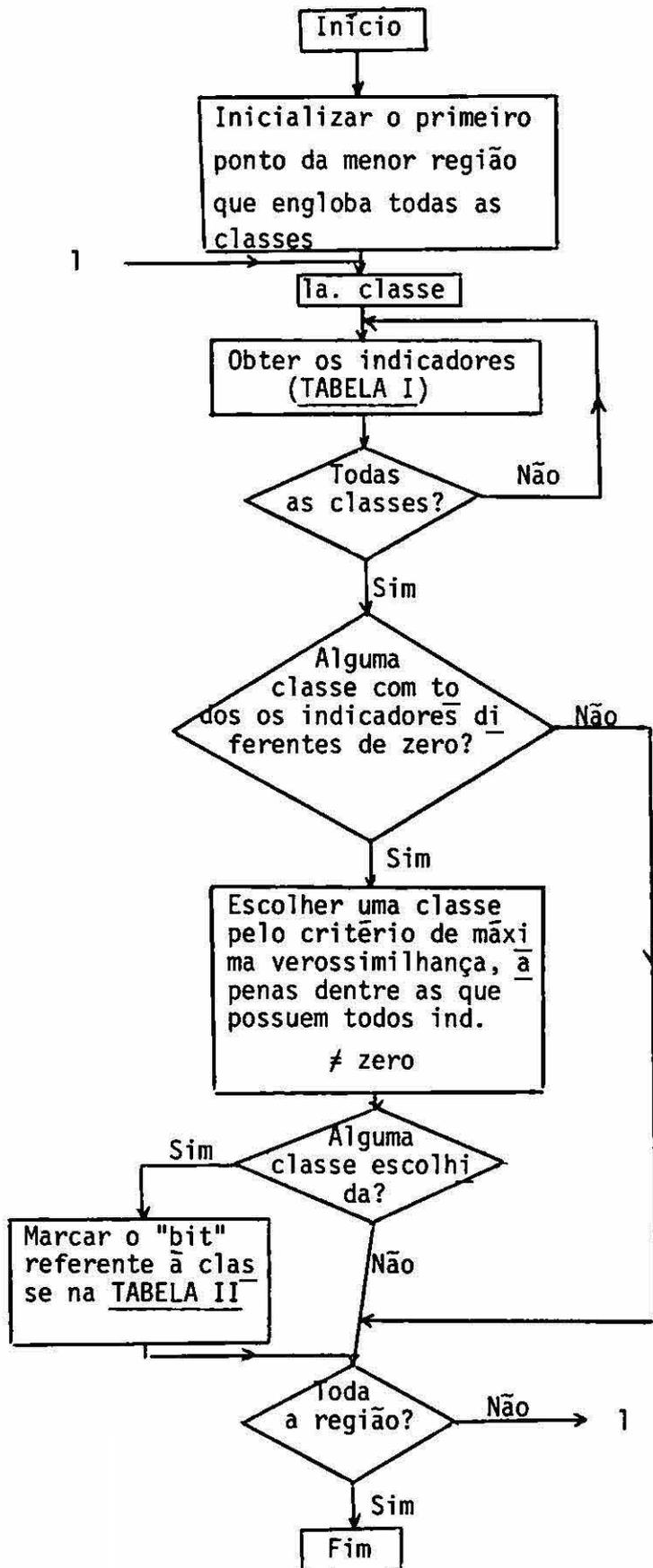


Fig. II.6 - MONTB2

2.4.5. TLUP3 - CLASSIFICAÇÃO DA TELA

A classificação pode ser efetuada na tela toda ou em área determinada pelo cursor. O programa apresenta as classes na ordem em que foi feito o treinamento. Devem então ser associados os temas às classes, a entrada feita na ordem em que serão testadas na classificação (ver Apêndice A para o diálogo). Notar que podem ser escolhidas apenas as classes desejadas. É realizada a classificação, o resultado aparecendo na tela, através dos temas. Finalmente, são apresentadas as estatísticas da classificação, sendo desprezadas as laterais, onde não há pontos. Nas Figs. II.7 e II.8 são apresentados os diagramas de blocos de TLUP3 e TLCLS2.

Rotina utilizada:

TLCLS2 - Rotina, programada em ASSEMBLY, que classifica uma linha (ou pedaço dela) de cada vez. As tabelas são passadas como parâmetros. A sequência de teste é aquela que foi escolhida no programa principal. A primeira classe testada é sempre a que foi atribuída ao ponto adjacente, exceção feita ao caso em que o ponto adjacente foi considerado não-classificado. Para cada classe, obtemos os indicadores (TABELA I), passando para a próxima logo que um dos indicadores for 0. Caso todos sejam diferentes de zero, testamos o "bit" do "byte" apontado pelos indicadores na TABELA II. Se o "bit" estiver aceso, a classe é atribuída ao ponto, caso contrário passamos à próxima classe, considerando o ponto não-classificado se todas as classes forem testadas sem sucesso. O resultado da classificação volta em um vetor, IMACLS.

2.4.6. TLUP4 - CLASSIFICAÇÃO DO DISCO

Neste caso, a imagem a ser classificada se encontra no disco e a classificação pode ser total ou parcial. As classes devem ser ordenadas em ordem decrescente de probabilidade (ver diálogo) como em TLUP3. As tabelas e os parâmetros são lidos do disco, do arquivo cujo nome foi dado em TLUP2. As estatísticas são dadas no final, em porcentagens e em quilômetros quadrados. Diagrama de blocos na Fig. II.9.

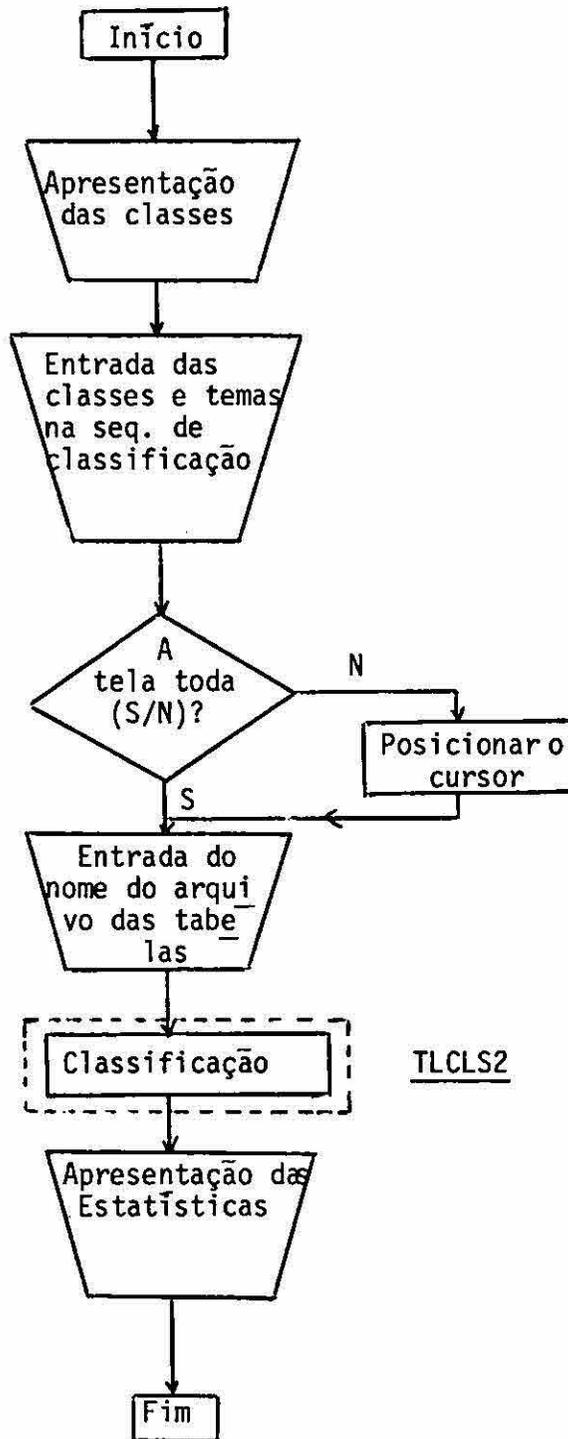


Fig. II.7 - TLUP3

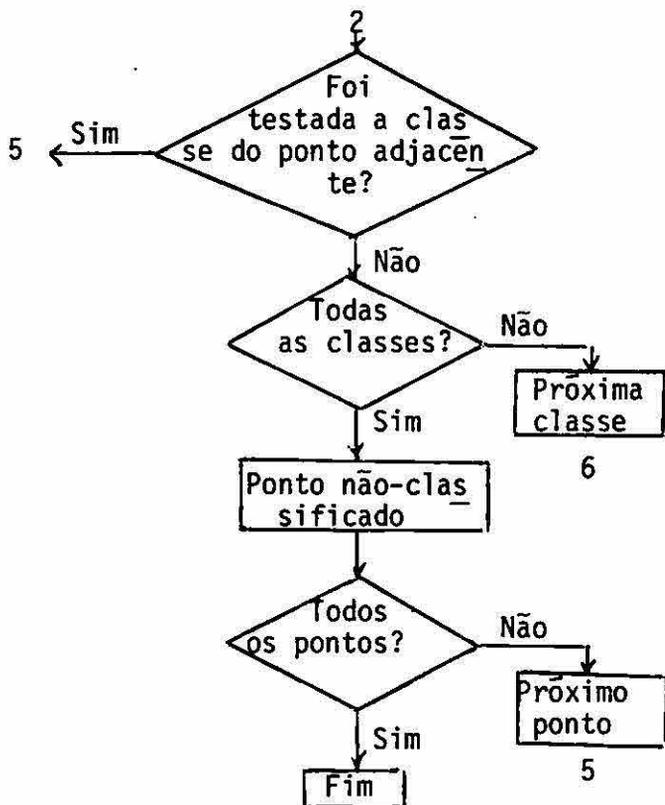
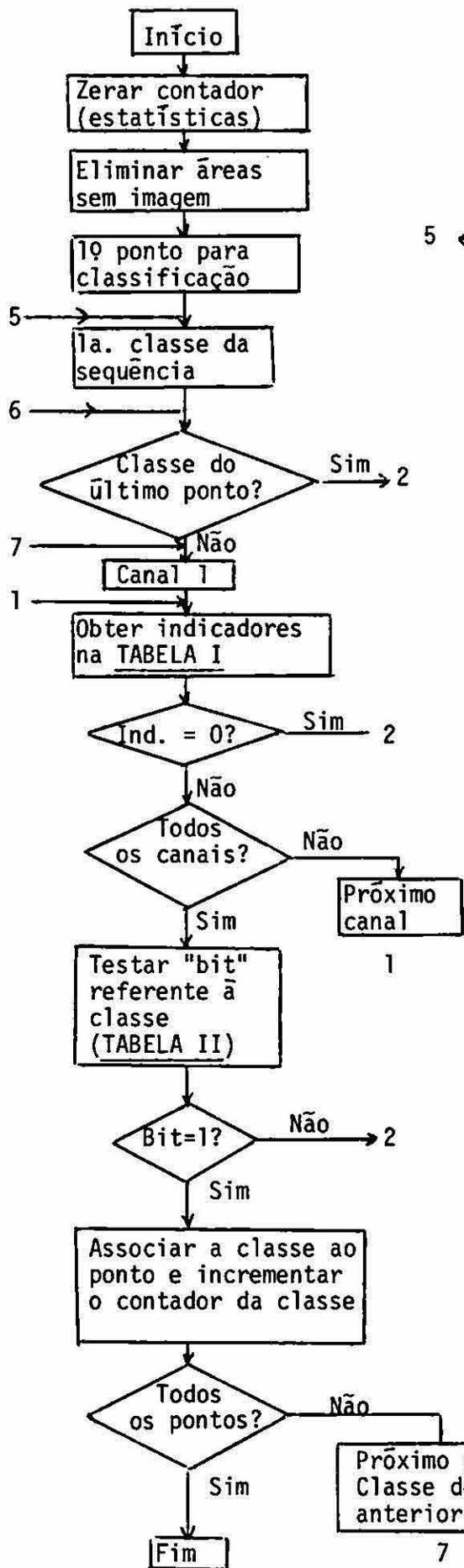


Fig. II.8 - TLCLS2

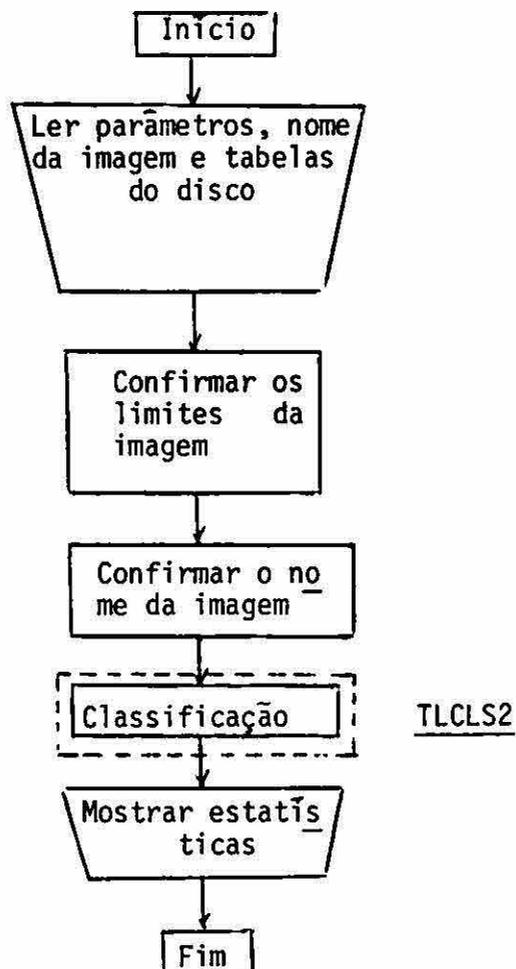


Fig. II.9 - TLUP4

Rotina utilizada:

TLCLS2 - Ver 2.4.5, Fig. 2.8

2.4.7. TLUP5 - ARMAZENAMENTO PARA CONTINUAÇÃO

O programa armazena em fita de papel os dados necessários para a continuação, como MAXV00. É dada entrada ao nome do arquivo, onde serão acumulados os histogramas e os limites de cada classe. O nome é da forma nome.DAT, onde nome possui um máximo de seis letras.

Rotina utilizada:

PPSUB - Escrita em fita

A seguir, são apresentadas as conclusões e os comentários sobre os resultados obtidos com o sistema implementado, além de algumas extensões possíveis.

2.5. RESULTADOS E COMENTÁRIOS

O sistema foi testado para diversas regiões. O tempo de construção da TABELA II é muito variável, uma vez que depende do número de pontos da menor região que engloba todas as classes. Esta região vai depender, por sua vez, da variância das classes. Para imagens normais, onde não foram efetuadas transformações sobre os diversos canais, os valores típicos variam de 30 seg. a 3 min., para 8 classes; para uma imagem onde três canais haviam sofrido "stretch" e aumento da variância, a elaboração da tabela consumiu cerca de 10 min. de processamento, também para 8 classes. Considerando que o tempo para a classificação do disco é da ordem de 10 a 15 vezes maior, uma melhora de cerca de 10% no tempo de classificação já compensa estes casos extremos.

Uma imagem, da região de Belo Horizonte, foi classificada na tela pelos sistemas MAXVER e TLUP. Foram treinadas 8 classes. A comparação dos resultados se encontra na Tab. II.4.

TABELA II.4 - COMPARAÇÃO DE RESULTADOS

	Tempo	Não-classificados
MAXV10	2m10s	10,56 %
TLUP3	53s	19,24 %

Nota-se que houve uma economia de cerca de 60% no tempo de processamento, bem maior do que o necessário para compensar a construção

da TABELA II, apesar de que, neste caso, este tempo foi de cerca de 1m40s. Observe-se que foi feita somente uma classificação parcial da tela.

O resultado da classificação merece alguns comentários. O número de pontos não-classificados foi o dado que apresentou a maior discrepância entre os dois métodos. A explicação se encontra na maneira como as tabelas são construídas. Como as regiões de interesse (hiper-paralelepípedos) das classes são obtidas a partir do treinamento, é possível, para classes cuja distribuição não possa ser considerada aproximadamente gaussiana, que o hiper-elipsóide ultrapasse as faces do hiper-paralelepípedo. Uma vez que os pontos fora do hiper-paralelepípedo não são levados em conta durante a classificação, mais pontos seriam considerados não-classificados. A diferença é visualmente notável quando são processadas imagens com áreas cobertas com água. Feito o treinamento em pequena parcela das áreas úmidas, a classificação pelo MAXVER resulta em área consideravelmente maior, classificada como pertencente à mesma classe que a da região treinada. Realmente, a suposição de distribuição gaussiana para a água é crítica, uma vez que os histogramas obtidos mostram que existem cortes abruptos na distribuição, sem a suavidade necessária. Note-se ainda que isto não significa que foi diminuído o limiar utilizado pelo critério de máxima verossimilhança, mas sim que foram eliminados pontos que seriam incluídos pela hipótese de distribuição normal. O fato que ilustra este ponto é o de que a região treinada não sofre variação, no resultado de classificação, por um e outro método. Mais pontos são considerados não-classificados mas, em contrapartida, o número de classificados incorretamente diminui (Tabela IV da Ref. 3).

Outro ponto deve ser ressaltado. O tempo de classificação é praticamente independente do número de classes, para mais de 4 classes. A Fig. II.10 mostra o gráfico do tempo de processamento em função do número de classes.

Considerando que são em ENTRADA/SAÍDA são gastos cerca de 20 segundos, de 1 para 8 classes temos apenas o dobro de tempo de processamento de algoritmo. Para o sistema MAXVER, este tempo é diretamente proporcional ao número de classes, já que todas são testadas para cada ponto.

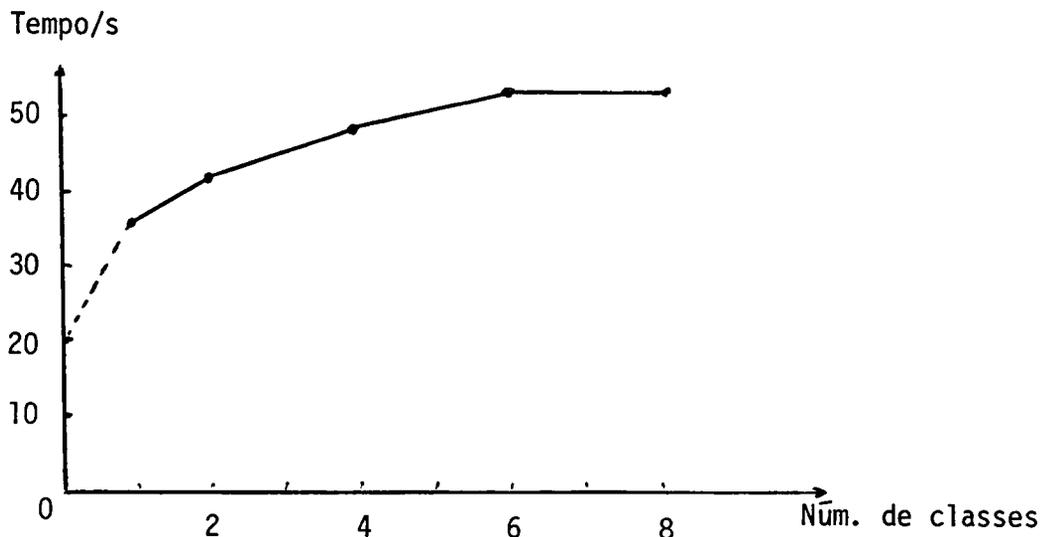


Fig. II.10

Foi verificado também, que as estatísticas de classificação de algumas classes se modificava, se fosse considerada uma sequência de teste diferente ou se sã se classificasse em determinadas classes. O problema está novamente na elaboração das tabelas. No caso em que se têm classes com interpenetração nas regiões de interesse e, além disso, o número de níveis ocupados por canal é maior que 9, existe a possibilidade de vários vetores de intensidade diferentes, serem associados aos mesmos indicadores, para classes diferentes, através da TABELA I. Quando é feita a classificação por máxima verossimilhança, durante a construção da TABELA II, pode-se ter dois pontos do espaço espectral associados a classes diferentes, mas com o mesmo conjunto de indicadores. Assim, o "byte" apontado pelos indicadores, para as classes em questão, seria o mesmo e teria mais de um "bit" aceso. Durante a classificação, a classe testada primeiro seria associada ao ponto, causando a diferença nas estatísticas. Mas o problema sã ocorre no caso de variância grande para classes, com interpenetração. Nos testes realizados, as diferenças não chegaram a 1%.

Podem ser feitas extensões para o sistema, como, por exemplo, um programa para subtrair amostras durante o treinamento e outro para realizar a classificação da tela sem utilização da área de COMMON, COMML, o que resultaria em economia de memória, necessária para a execução de outros programas, processados em "time-sharing".

Pode-se prever, além disso, a seleção das classes a serem examinadas em ordem decrescente de probabilidade, que deve ser estimada a partir da frequência relativa das mesmas.

CAPÍTULO III

CLASSIFICAÇÃO USANDO TABELA CONSTRUÍDA POR "HASHING"

3.1. INTRODUÇÃO

No intuito de diminuir o tempo de classificação de imagens resolveu-se utilizar a estrutura de "hashing" proposta por Ambler & Knuth (Ref. 5) na construção da tabela. Pretendeu-se com isto melhorar o desempenho do programa em relação às duas características mais críticas nesta tarefa:

1. Minimizar o tempo de procura a um determinado vetor na tabela.
2. Minimizar a memória de massa necessária no computador.

Naturalmente há um compromisso entre tais metas, obrigando a busca de doses ótimas de prioridade.

Considerando os quatro canais, se a cada vetor (coordenadas espectrais de um pixel) for associada uma posição na tabela, serão necessários 256^4 lugares, pois cada canal compreende níveis de cinza entre 0 e 255 (8 bits).

Levando em conta o alto custo de memória, isto é inviãvel, pelo menos nas condições atuais. Seria também contraproducente uma vez que as componentes espectrais da imagem do LANDSAT são altamente correlacionadas. No máximo 10.000 vetores seriam suficientes para cobrir toda a gama espectral que aparece numa imagem típica deste satélite.

Devido a tal limitação de dados, uma idêia seria a de guardar os vetores classificados à medida que fossem aparecendo, para serem reaproveitados quando as intensidades se repetissem.

Surge naturalmente a pergunta: como armazenar os vetores?

Poderiam ser colocados um abaixo do outro, na sequência de aparecimento. Haveria um ponteiro indicando sempre a próxima posição vaga e, assim, seria mínimo o tempo de construção da tabela. O problema é que para acessar um determinado vetor, teria que ser percorrida a tabela toda. Aliás, o mesmo procedimento deve ser tomado para evitar guardar duas vezes a mesma classificação.

Decorre daí a procura de formas mais elaboradas de estruturas de dados.

A primeira delas é o "binary search" (Ref. 6). Nesta permutação, vai se dividindo o espaço de dados em metades até resultar o endereço final do vetor procurado. É uma boa idéia se for considerado apenas o aspecto de procura do vetor. Este é acessado rapidamente porém, em contrapartida, é bastante difícil a construção da tabela. Inserir novos elementos altera toda a estrutura, levando um tempo muito grande.

Procurou-se, então, um método de acesso e construção rápida. Isto naturalmente implicou em memória ociosa, conforme será visto adiante, mas o acréscimo não é tão considerável assim.

Foi deste modo que a escolha acabou recaindo sobre o "hash".

Certamente haverá outros métodos de se estruturar uma tabela, porém é certo também que o escolhido daria conta das especificações desejadas-fácil construção, fácil acesso e uma memória requerida razoável.

Dentre os vetores que aparecem numa imagem, existem aqueles que são mais populosos, ou melhor, são mais frequentes. Uma tabela mais representativa e mais eficiente dentro do enfoque do "table look-up" deveria conter estes elementos. Acontece que selecioná-los custaria tempo e justamente isto é o que se deseja diminuir.

Além disso, escolhendo os elementos, é quase certa a necessidade de dividir a tarefa em duas partes: uma de seleção dos pontos e posterior construção da tabela, outra de acesso e classificação propriamente dita.

Foi preferida a classificação por um passo apenas, que consiste em ir acessando a classe dos "pixels" ao mesmo tempo que se vai construindo a tabela. Se o tempo para armazenar é praticamente o mesmo para acessar, nada impede que se façam ambos concomitantemente.

É possível que, fazendo-se a seleção dos elementos, a tabela fique tão eficiente que desconte e talvez até supere o tempo da seleção dos pontos, economizando assim mais tempo que o algoritmo de um passo.

Realmente, ainda não há base para afirmar nada sobre qual dos dois é mais rápido. O procedimento normal seria implementar as duas maneiras e ficar com a melhor.

Optou-se pelo método do "one pass approach", confiando no fato dos pontos mais populosos terem mais chance de aparecerem primeiro, antes de esgotada a capacidade total da tabela. Os restantes, supostamente os menos populosos, são classificados toda vez que aparecerem na imagem, sem haver armazenamento de suas classificações.

A atribuição de uma classe para um vetor é feita pelo sistema MAXVER (Ref. 1) implementado anteriormente, aqui no INPE, para as condições do sistema analisador IMAGE-100 da GE, acoplado a um PDP-11/45. O citado sistema adota na classificação o critério da máxima verossimilhança.

O uso do "hash" foi baseado no artigo publicado por Shlien (Ref. 7).

A seguir é dada uma descrição do funcionamento do algoritmo.

3.2. CLASSIFICAÇÃO POR UM PASSO USANDO "HASH"

O esquema de "hash" consiste em definir uma função que dá a um conjunto de coordenadas espectrais o seu endereço na tabela. Utilizando esta mesma função, pode-se facilmente buscar o vetor e a classe quando se necessitar deles, nas repetições.

Foi escolhida a seguinte função: - Colocar as quatro intensidades lado a lado, resultando num escalar. O resto da divisão do escalar por um dado número daria o endereço.

Qual seria este número?

À primeira vista, seria a capacidade máxima da tabela, porém aí ocorre o risco de um "loop" infinito. Querendo contornar o problema, foi usado o maior número primo ainda dentro dos limites da tabela.

Como de vários dividendos pode resultar um mesmo resto, haveria várias composições sendo indicadas para um mesmo endereço.

Como guardar tantos elementos num só lugar?

Certamente não há meios de guardar. A solução tomada foi: - Indicado um endereço, o ponteiro da tabela dirige-se para a posição e verifica se já há algum elemento ali. Se não houver, guarda a composição e a classe. Porém, se houver, testa se não é a sua própria classificação. Se for, toma a classe e sai do algoritmo. Se não for, busca um novo endereço.

Batizou-se a operação de tomada de novo endereço por "double hash" ou 2º "hash". Consiste em dividir o escalar obtido anteriormente pelo segundo maior número primo sem sair dos limites da tabela. Toma-se o resto da operação e soma-se ao endereço obtido na vez anterior. Se a soma cair dentro do intervalo da tabela, será o novo endereço. Caso contrário, este é obtido subtraindo-se da soma aquele maior número primo, já considerado na primeira tentativa.

É preciso atenção para o fato de se acabar fatalmente caindo num "loop" infinito se duas tentativas de "hash" resultarem num mesmo endereço. A utilização de números primos como divisores pretendeu justamente evitar este problema.

Quando o resto da divisão no "double hash" é zero, também se enfrenta o problema do "loop". Em tal caso adiciona-se uma unidade ao resto da divisão.

Terá ocorrido uma colisão quando um vetor encontrar um outro na posição indicada.

De que maneira se diferenciará uma colisão da própria classificação? Como saber se a classe inserida numa determinada posição refere-se ao vetor à mão ou a um outro vetor que saiu da função com o mesmo endereço?

A resposta é: - ao construir a tabela, guarda-se, junto com a classificação, o número resultante da linearização das intensidades. Assim, acessando um endereço, verifica-se a igualdade dos identificadores (o escalar). Em caso afirmativo, é só retirar a classe e passar para o próximo ponto. Em caso negativo, novo endereço é tomado por "double hash". A nova posição é acessada. Testam-se os identificadores e segue-se neste esquema até encontrar a classificação ou então uma posição vazia. O vazio significa que nenhum vetor acessou a posição, e, portanto, deve ser classificado e armazenado exatamente naquele lugar.

Se o "pixel" não foi anteriormente classificado, a rotina só sai do "loop" quando encontra o branco (vazio). Então, em caso da tabela estar muito cheia, serão necessárias muitas tentativas até ser alcançado o vazio. Isto compromete bastante a velocidade do algoritmo.

Provendo tal problema, limita-se o número de vetores armazenados. Após a tabela ser preenchida 75% da sua capacidade total, nenhum vetor a mais é armazenado. Conforme já foi dito, os pontos além da tabela são classificados a cada vez que aparecerem.

O diagrama de blocos é mostrado na Fig. III.1 para melhor visualização do processo.

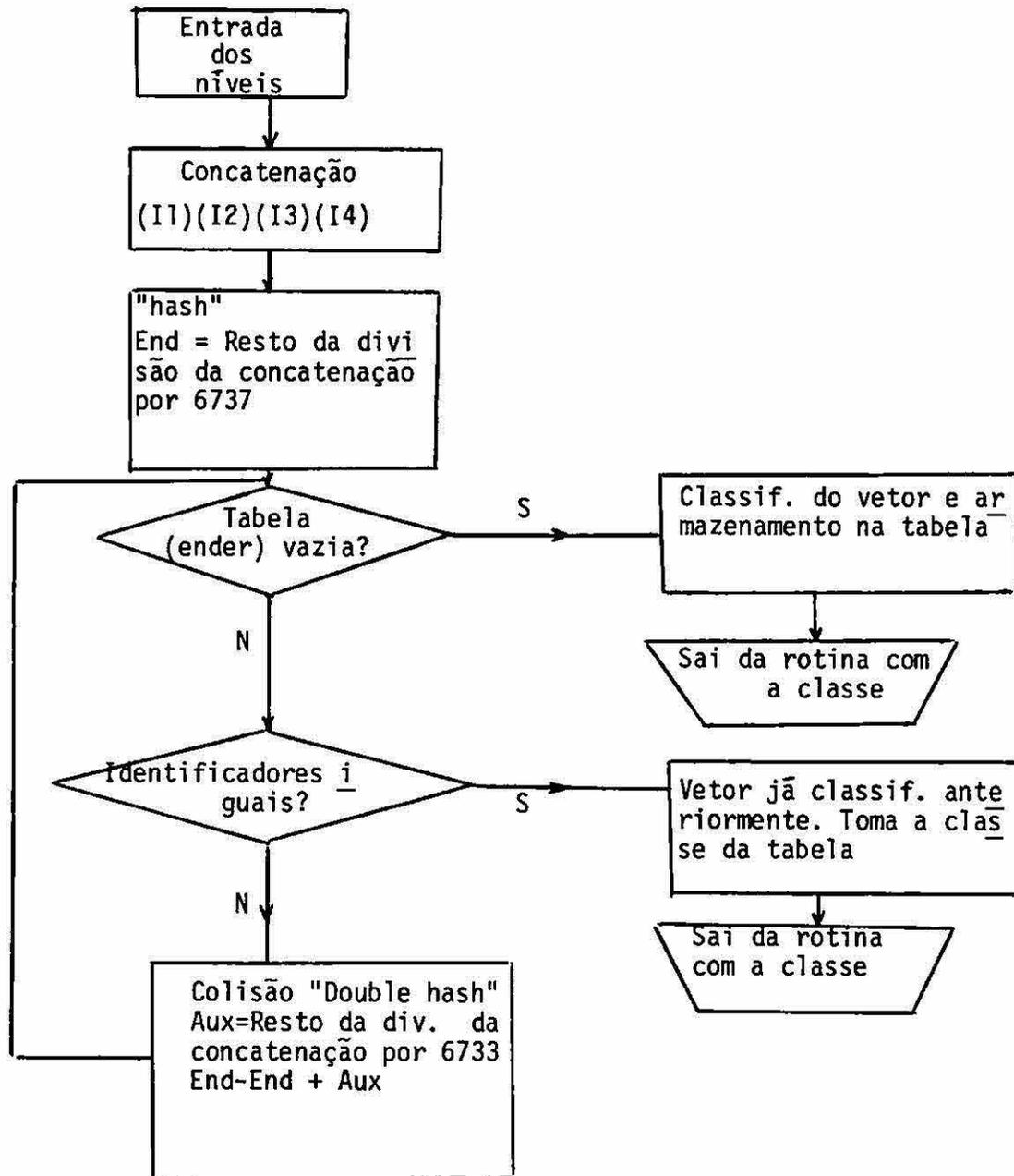


Fig. III.1

3.3. NÍVEIS DE INFORMAÇÃO

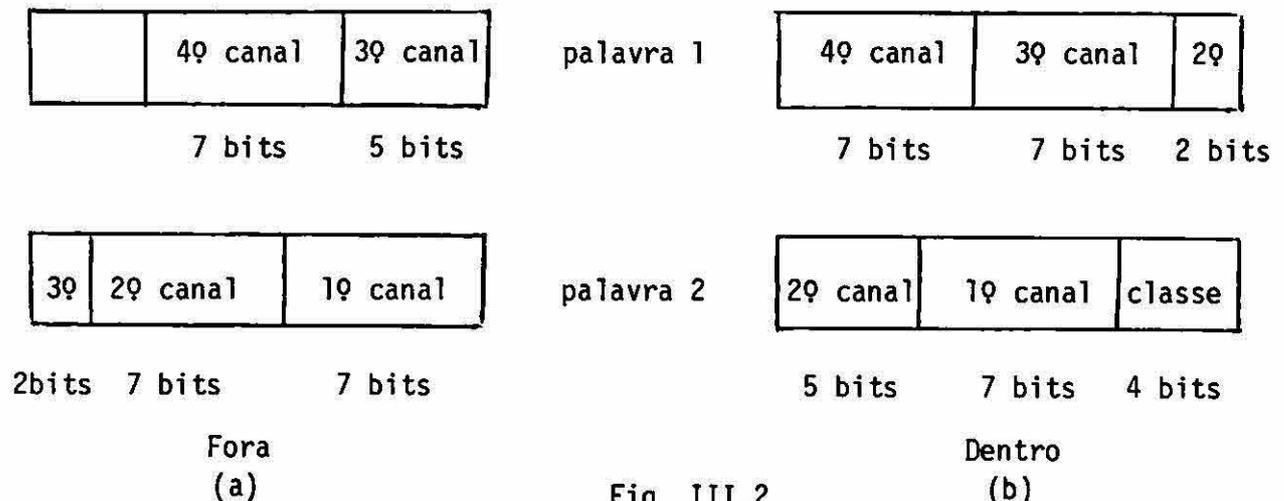
No algoritmo proposto por Shlien (Ref. 7) são considerados até 64 níveis de cinza (6 bits) para cada um dos canais 4, 5, 6 e 7 do LANDSAT (o que implica num escalar de 24 bits). Aqui no INPE trabalha-se com 8 bits, resultando num escalar de 32 bits. Ao que tudo indica, o centro canadense (Ref. 7) dispõe de computador que manipula palavras de 32 bits. O PDP aqui usado manipula 16 bits.

O problema foi contornado trabalhando-se em "assembly", permitindo deste modo considerar duas palavras por endereço, sendo elas capazes de trocarem bits entre si. Resulta daí se terem 32 bits utilizáveis por posição.

Ainda assim, ficavam faltando bits pois era necessário guardar também a classe.

Desprezar bits das coordenadas espectrais significava perder em resolução mas, não havendo outra saída, foi jogado fora o bit menos significativo de cada coordenada. Utilizaram-se então 28 bits como identificadores e o resto para a classe. A disponibilidade de apenas 4 bits condiciona a um número máximo de 15 classes mas de qualquer modo este valor é bem próximo ao de outros programas (Ref. 1).

Mostra-se em seguida, na Fig. III.2 a concatenação das intensidades e da classe, fora e dentro da tabela.



É claro que o identificador poderia ocupar menos bits (15 seriam suficientes para os 10.000 vetores) mas para isto se teria que complicar a função geradora de endereços, ou pelo menos ela necessitaria de um bom número de instruções, o que levaria tempo para ser processado.

3.4. COLISÕES

Uma preocupação normal para o implementador é saber a frequência de colisões, retardando o tempo de classificação.

Surpreendentemente, conforme estudos feitos por Shlien (Ref. 7), é bastante rara a ocorrência de longas cadeias de colisões mesmo que nenhuma medida seja tomada para preveni-las.

Foi construída uma tabela 74,34% cheia. Contou-se o número de tentativas necessárias para localizar um determinado vetor. Mais da metade dos vetores foram acessados sem colisão. Após cinco tentativas apenas 350 dos 5.955 vetores não foram encontrados. O maior número de tentativas foi 80. Em média precisou-se 2,21 tentativas por vetor.

3.5. TAMANHO NECESSÁRIO DA TABELA

É patente a limitação de memória disponível. Além disso, para continuar eficiente a rapidez de acesso, foi deixada memória ociosa, o que diminuiu ainda mais a capacidade de armazenamento.

Depois da tabela estar cheia, os pontos ainda não armazenados necessitarão, além do tempo normal de classificação, aquele para se constatar que não estão na tabela. Então, é altamente indesejável classificar "pixels" nesta situação.

Em suma, fica evidente que o programa proposto funciona melhor (no sentido de gastar menos tempo) à medida que a imagem trabalhada seja mais homogênea. A correlação entre as intensidades de cinza dos "pixels" deve ser alta.

Querendo manter a eficiência constante, certamente se te
rã que aumentar a capacidade da tabela.

Mais uma vez Shlien (Ref. 7) foi quem estudou a relaçã
o tamanho da tabela/homogeneidade da imagem. As suas conclusões são mostra
das na Tab. III.1. Nesta aparece a quantidade mínima de posições necessã
rias em função do número de canais considerados, a porcentagem (α) de
coordenadas espectrais não cobertas pela tabela por unidade de raiz qua
drada do determinante da matriz de covariância da imagem toda.

No caso do LANDSAT o comportamento da covariância em fun
ção dos vários tipos de imagem é apresentado na Tab. III.2. O determinan
te da matriz de covariância $|C|$, e a fração de "pixels" não cobertos pe
la tabela (α) são estimados de uma amostra de 250.000 "pixels". Na últi
ma coluna está o valor teórico desta fração, se for suposta uma distri
buição gaussiana das classes. Como era de se esperar, o modelo produz
estimativas melhores em áreas uniformes, contendo apenas um tipo de pa
drão. Sua validade decai totalmente em áreas complexas.

A classificação de áreas de recursos naturais, segundo da
dos da Tab. III.2, exigiria 12.000 posições ($|C| \cong 400$). Pela capacidade
disponível de memória, foi conseguido um máximo 5.100 posições. Mesmo
com apenas metade do tamanho indicado da tabela, a presente idéia é ain
da mais eficiente que calcular todas as probabilidades para todos os
"pixels". Certamente não se obteve a redução de 30 vezes no tempo, como
conseguiu Shlien.

3.6. O SISTEMA MAXHAS

É dada, a seguir, uma breve explanação dos programas im
plementados nesta fase do trabalho.

Ao lado do programa de "hash" propriamente dito (CLSRAP)
são necessários outros que servem de base na classificação.

Tem-se, assim, o MAXHAS (MAXFIT), CLRTAB, MGL e HASQIO.

3.6.1. MAXHAS

Classificação na tela do IMAGE-100. Seu funcionamento é semelhante ao MAXV10 (Ref. 1). É quem toma os parâmetros gerados pelo sistema MAXVER (MAXV01) através de uma área de COMMON. Associa, à cada classe existente, um tema.

Pode classificar a tela toda ou a região interna ao cursor.

Lê os níveis dos 4 canais de uma linha inteira (512 "pixels"). O resultado da classificação é escrito no canal 5.

Ao mesmo tempo que a classificação, é feita uma contabilidade da porcentagem de "pixels" em cada classe e em nenhuma das classes.

3.6.2. MAXFIT

É praticamente igual ao programa anterior. A diferença é a transferência de parâmetros, agora feita por fita de papel. Dispensa-se assim a área de COMMON, economizando memória.

3.6.3. CLRTAB

Zera todas as posições da tabela antes de começar a classificação. Pela maneira como é feito o teste de vazio, foi necessário o cuidado de limpar esta região de possíveis dados de programas anteriores.

3.6.4. MGL

É um programa subordinado à estrutura do CLSRAP.

Faz a classificação por máxima verossimilhança.

Prevê o caso do "pixel" não pertencer a nenhuma das classes adquiridas por treinamento.

3.6.5. HASQIO

Classificação no disco.

Divide uma linha do disco em 7 partes de 512 "pixels" cada. Usa o mesmo CLSRAP na classificação.

A leitura dos parâmetros é feita de fita de papel. O resultado é escrito num arquivo do disco.

TABELA III.1

Dimensão/ α	50	25	10	5	2,5(%)
1	,7	1,1	1,6	1,9	2,2
2	4,4	8,7	14,5	18,8	23,2
3	15,	34,	65,	91,	119
4	55,	143	298	444	607
5	131	376	862	1364	1948
6	791	2489	6153	10335	15427
7	3048	10332	28281	48509	77409

Número mínimo de vetores na tabela "hash" versus porcentagem de "pixels" não cobertos na imagem (α), em função de p: número de canais (dimensão). O determinante da matriz de covariância é tomado igual a 1. Para obter o número de vetores no caso de determinante da matriz de covariância diferente de 1, multiplica-se o valor indicado na tabela pela raiz quadrada do determinante da referida matriz.

TABELA III.2

Tipo de padrão	C	V	α (%)	α^*
Vegetação	32,	3371	0	2,7
Neve	85,	5698	0	2,5
Agricultura	344,	5818	5,8	8,8
Vegetação+Água+Atmosfera difusa	1460,	5481	1,8	24,7
R.urbana+vegetação	296,	5455	1,6	9,2
R.urbana+vegetação + rocha+neve+água	300000,	5573	26,7	83,2

Estatística de uma amostra de 250.000 "pixels". $|C|$ é o determinante da matriz de covariância, α é a fração de "pixels" não cobertos por tabela contendo V vetores e α^* é o valor estimado de α .

3.7. RESULTADOS E COMENTÁRIOS

Classificou-se uma imagem da região de Ribeirão Preto, relativamente homogênea, com agricultura e cidade (A17827.777). Foram definidas 6 classes.

O sistema MAXVER (MAXV10) precisou aproximadamente 6:00 minutos, enquanto o MAXHAS aproximadamente 2:30 minutos, para classificar a tela toda do I-100. Na ocasião deste teste usavam-se ainda 64 níveis de informação por canal.

Em termos de confiabilidade, os resultados de ambos os sistemas foram idênticos.

Trabalhando depois com uma foto de Manaus (1976) definiram-se 5 classes: água do Rio Negro, água do Rio Solimões, Floresta Amazônica propriamente dita, vegetação de beira-rio e cidade.

Para toda a tela foram necessários 5:00 minutos usando o MAXVER, e mais ou menos 2:30 com o MAXHAS. A relação entre tempos baixou porque , na segunda imagem, as diferentes classes são mais interpenetradas causando maior heterogeneidade de reflexão.

Desta vez, comprova-se a igualdade das classificações facilmente olhando as estatísticas apresentadas no Apêndice B.

CAPÍTULO IV

CLASSIFICAÇÃO POR AGREGAÇÃO

4.1. INTRODUÇÃO

A idéia agora já não é mais calcular probabilidades ou otimizar algoritmos que façam tais cálculos, nem construir tabelas devido às redundâncias da imagem.

Quer-se aproveitar o fato das coordenadas espectrais de uma classe estarem mais ou menos agrupadas no espaço de atributos.

Vão ser agregados, iterativamente, objetos similares em conjuntos denominados "clusters". O critério de similaridade é a distância mínima, sendo a mais típica delas a Euclidiana.

Durante esta descrição não se fará distinção entre "cluster" e classe, assim como objeto, dado, "pixel" ou amostra.

Trata-se de uma tentativa no sentido da classificação automática. Sem nenhum treinamento prévio, procura-se definir as classes existentes em uma imagem.

Vários algoritmos foram criados com esta orientação* porém foi feita a opção pelo ISODATA (Iterative Self-Organizing Data Analysis Technique A) (Ref. 9), principalmente por ele produzir resultados bons e ter-se apresentado uma boa bibliografia sobre o assunto ao nosso alcance (Refs. 8 e 10).

* "K-médio" (Ref. 8), "One pass clustering" (Ref. 10), "Adding algorithm" (Ref. 10), "Non parametric clustering scheme" (refs. 11 e 12), e vários outros.

É interessante notar que o algoritmo é válido não só na análise de imagens multiespectrais, mas é usado também no agrupamento de dados sociológicos e atmosféricos (rastreamento do movimento das nuvens).

Percebe-se que em nenhum ponto o algoritmo está restrito à análise de imagens.

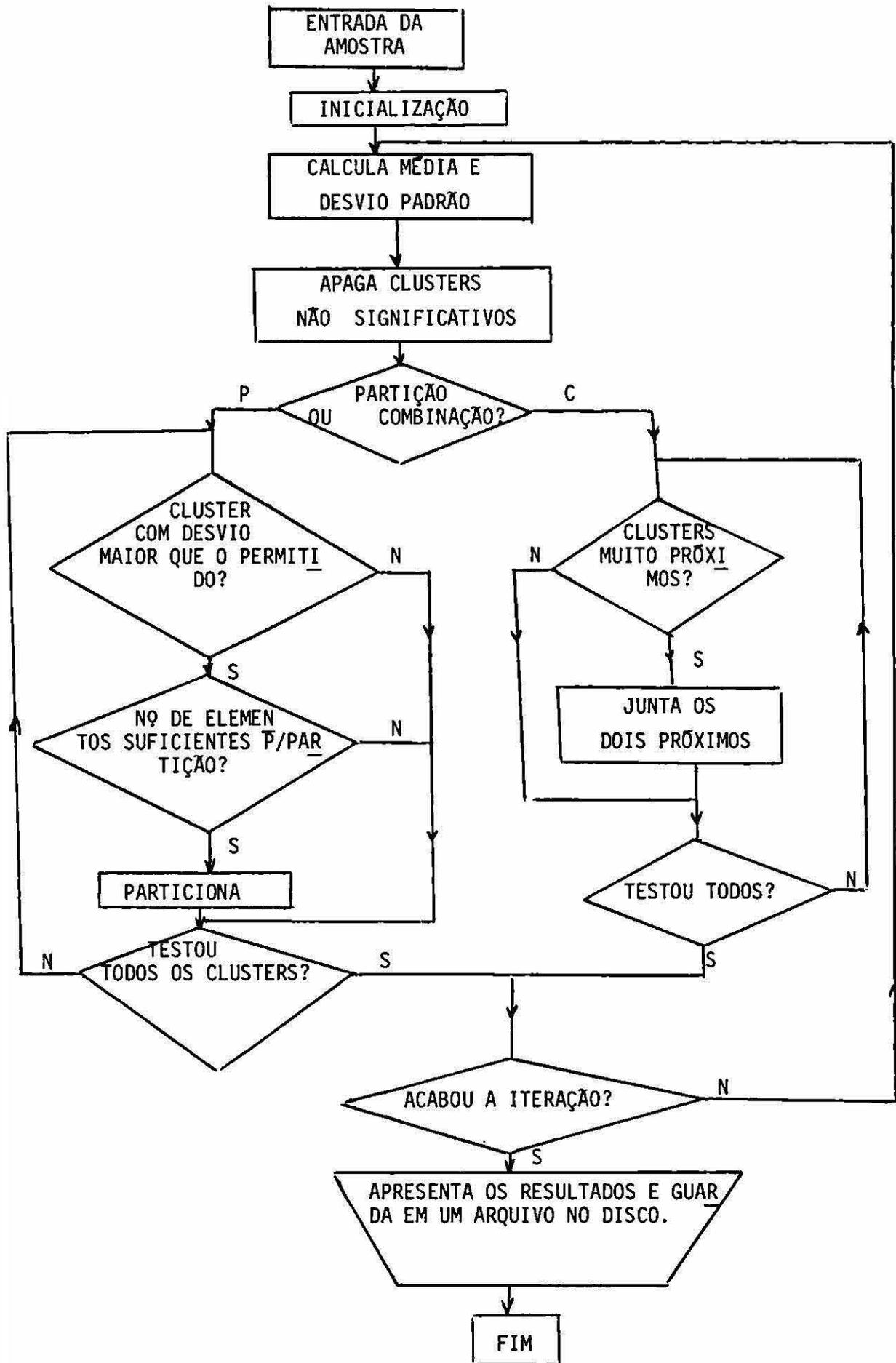
4.2. ISODATA

Inicialmente o algoritmo toma as amostras, considerando-as pertencerem a um grande "cluster". Em seguida vai particionando-o em sub-"clusters" ou combinando os mesmos, se houver conveniência, segundo a sequência partição-combinação fornecida pelo usuário.

Uma descrição passo a passo facilita a compreensão:

1. Toma as amostras do arquivo.
2. Inicializa as variáveis (média e desvio padrão), toma os parâmetros e lê a sequência de operações (C/P).
3. Das amostras calcula média e desvio padrão para os "clusters" existentes.
4. Contabiliza as amostras pertencentes a cada "cluster".
5. Muda a estrutura dos "clusters" conforme os critérios estabelecidos.
 - a. Apaga "clusters" com menos que um número mínimo de elementos.
 - b. Particiona ou combina, conforme o caso.
6. Testa se chegou ao final da sequência. Desvia para o fim e mostra os resultados se a resposta for afirmativa. Retorna para 3 com possíveis novos centros, caso contrário.

Na Fig. IV.1 mostramos o diagrama de blocos.



4.3. DESCRIÇÃO DE CADA PASSO

1. Tomada das amostras

Escolhemos os pontos mais populosos, isto é, os que aparecem mais vezes na imagem.

2. Inicialização

2.a. Média e desvio padrão - Como são calculados iterativamente, deverão ser zerados.

2.b. Sequência partição-combinação - A versão original do ISODATA faz uma alternância entre estas duas operações. Pela experiência, tem-se notado que isto é um tanto arbitrário e não produz resultados ótimos. Queremos nos referir com isto à obtenção de agregados que se aproximem mais da realidade, dentro das limitações do método e do nosso conhecimento. Uma sequência típica, proposta quando não há um conhecimento prévio das amostras, é SSSCSCSCSCSCCC. (S-partição; C-combinação). Quando houver classes definidas a priori, recomenda-se a sequência CSCSCSCSCCC. Através da experiência também se tem notado que a definição inicial de classes é absolutamente dispensável, bastando considerar todos os dados como pertencendo a um único "cluster" global. Apesar da sequência indicada conter 14 iterações o programa comportará até 16, para possíveis eventualidades. Ela será lida, permitindo ao usuário modificá-la com base na experiência sobre a amostra trabalhada.

2.c. Parâmetros

- Desvio padrão máximo. As classes são particionadas quando seu desvio alcança ou ultrapassa um certo valor máximo. Este limiar de partição é aproximadamente 4,5 para processamento de imagens do MSS (Multispectral Scanner) contendo 8 "bits" de informação. No caso de agregamento de outros tipos de dados sugerem-se testes preliminares que auxiliem na escolha do valor crítico.

- Limiar de combinação. Dois "clusters" com centros distantes do menos que um certo valor são combinados. Alguns autores indicam a escolha de valores entre 2,8 a 3,0 (Ref. 13). Estudos posteriores apontam 3,2. Observações empíricas fazem crer que este valor independe do número de canais utilizados.
- Número mínimo de elementos para se caracterizar um "cluster" - Kan & Holley propõem 20 elementos por classe (Ref. 10), certamente considerando uma amostra razoavelmente grande. Se se tiver, por exemplo, 50 dados, não terá sentido considerar 20 deles. Deixa-se este valor a critério do usuário que, pelo tamanho de sua amostra, saberá intuir o valor adequado.

3. Classificação dos "pixels"

Será associada a cada "pixel", uma classe, medindo a distância do "pixel" em relação aos centros dos vários "clusters". Atribuir-se-á o "pixel" à classe cujo centro estiver mais próximo. Poderia ser usada, a distância Euclidiana ponderada $d_w(x_i, x_k)$:

$$d_w^2(x_i, x_k) \triangleq \sum_{j=1}^{ndim} w_j^2 (x_{ij} - x_{kj})^2$$

com

$$w_j \triangleq 1/\text{sig}_j \quad ; \quad j=1, \dots, ndim$$

sig_j = desvio padrão na direção j .

Ou então, como Ball & Hall (Ref. 9), a distância ponderada:

$$d^2(x_i, x_k) \triangleq \sum_{j=1}^{ndim} (x_{ij} - x_{kj})^2$$

Foi usada a última por exigir menos cálculos (economia de tempo de processamento) e não dar resultados muito diferentes da primeira. Uma terceira alternativa é usar o módulo das diferenças das coordenadas, igualmente com bons resultados (Ref. 10).

$$d_1(x_i, x_k) \triangleq \sum_{j=1}^{ndim} |x_{ij} - x_{kj}|$$

Note-se que todas estas distâncias só terão sentido quando se tratar de dados com mesmo número de dimensões. Diz-se que $X_k = (X_{k1}, X_{k2}, \dots, X_{k1})^T$ pertence à classe I se

$$d^2(X_k, m^{(J)}) > d^2(X_k, m^{(I)}); \quad J \neq I, \quad J = 1, \dots, n^{\circ} \text{ tot. de "clusters"}$$

4. Cálculo da média e do desvio padrão

A média é avaliada por:

$$m^{(p)} = (m_1^{(p)}, \dots, m_{ndim}^{(p)})^T = (1/K) \sum_{i=1}^K X_i$$

K = tamanho da amostra

O desvio padrão é calculado fazendo:

$$(\text{sig}_j^{(p)})^2 = (1/K) \sum_{i=1}^K (X_{ij} - m_j)^2 \quad ; \quad j = 1, \dots, ndim$$

Lembrando a recomendação de Kan & Holley poder-se-ia ser tentado a calcular o desvio usando:

$$s\bar{i}g = (1/K) \sum_{i=1}^K |X_i - m|$$

O inconveniente principal de utilizar $s\bar{i}g$ é exigir um novo limiar de combinação (discutido no tópico 2.c) cuja avaliação ainda não é perfeitamente conhecida.

$$sig = (1/K) \sum_{i=1}^K (X_i - m)^2 = (1/K) \sum_{i=1}^K [X_i^2 - 2X_i m + m^2] =$$

$$(1/K) \sum_{i=1}^K X_i^2 - 2m (1/K) \sum_{i=1}^K X_i + (K/K)m^2 = (1/K) \sum_{i=1}^K X_i^2 - m^2$$

Kan & Holley (Ref. 10) propuseram cálculos da média e do desvio padrão através da acumulação, ao mesmo tempo, da classificação:

$$m_j^{(I)} = \frac{\text{npontos}(I) - 1}{\text{npontos}(I)} m_j^{(I)} + \frac{1}{\text{npontos}(I)} X_{kj}$$

$$\text{lamb}_j^{(I)} = \frac{\text{npontos}(I) - 1}{\text{npontos}(I)} \text{lamb}_j^{(I)} + \frac{1}{\text{npontos}(I)} X_{kj}^2$$

$$\text{sig}_j^{(I)} = \left\{ \text{lamb}_j^{(I)} - (m_j^{(I)})^2 \right\}^{1/2} \quad j = 1, \dots, \text{ndim}$$

onde

lamb é uma variável auxiliar

npontos(I) é o no de pixels assumidos como pertencentes à classe I.

Notar que, nas formlas anteriores, npontos(I) já computou o elemento X_{kj} da iteração à classe I. Teoricamente não há restrição alguma em proceder assim. Porém, na prática, isto não é muito aconselhável pelo fato da divisão dar quocientes que são arrendodados (por limitação de "bits"). O erro de arredondamento, por ser acumulativo, pode tornar-se muito grande.

Por isto, foi feito o seguinte:

- A medida que se concluisse a classe I para X_{ij}

$$\begin{aligned} X(I)_j &= X(I)_j + X_{ij} & i &= \text{índice relativo a amostra} \\ & & j &= 1, \dots, \text{ndim} \\ Y(I)_j &= Y(I)_j + X_{ij}^2 & I &= 1, \dots, \text{ncluster} \end{aligned}$$

- Ao final da amostra, para todas as classes:

$$\begin{aligned} m_j^{(I)} &= \frac{1}{\text{npontos}(I)} \sum_{i=1}^{Kc1} X_i(I)_j \\ \text{sig}_j^2(I) &= \frac{1}{\text{npontos}(I)} \sum_{i=1}^{Kc1} \frac{X_i^2(I)_j}{Y(I)_j} - m_j^2(I) \end{aligned}$$

5. Verificação dos "clusters" sem um número significativo de elementos

Fazer um teste para ver se o número de componentes de um "cluster" é maior que o número fornecido pelo usuário (tópico 2.c.). Se for maior, sua existência é confirmada. Caso contrário, ele é apagado.

6. Partição

Nas primeiras versões do ISODATA definiu-se o parâmetro AD (average distance) como critério de decisão para particionar um "cluster".

$$AD \triangleq (1/K) \sum_{i=1}^K d_w(X_i, m)$$

onde

d_w já foi definido anteriormente.

X_i , em nº de K , são os elementos da classe.

m a média ou centro do "cluster"

A partir de um certo valor crítico era determinada a partição. Procurou-se um valor ideal que não causasse partições desnecessárias, nem permitisse a existência de um "cluster" excessivamente generoso, em termos de aceitar elementos não suficientemente semelhantes. Primeiramente pensou-se que o valor $(N-0, 16)/\sqrt{N}$, para o caso n -variado era o procurado. Mais tarde foi verificado que este critério só funcionava bem em casos particulares, como quando o "cluster" congrega dois grupos com aproximadamente o mesmo número de elementos. Notou-se que, adicionando um "pixel" bastante afastado do centro, em relação ao novo centro devido a esta adição, o desvio padrão aumentava bastante enquanto que a soma das distâncias absolutas nem tanto. A distância média acabava não crescendo, apesar da introdução do elemento estranho. Resolveu-se, então, adotar o desvio padrão como medida da "não-pureza" de um "cluster". Um valor crítico superior é obtido empiricamente, conforme já discutido anteriormente. Quando particionado, o "cluster" dá origem a dois outros, com as médias:

$$(m_1^{(I)}, m_2^{(I)}, \dots, m_j^{(I)} \pm sig_j^{(I)}, \dots, m_L^{(I)})^T$$

(partição na direção j)

Os desvios padrões das duas novas classes serão dados (Ref. 14), pelo desvio da classe de origem dividido por $\sqrt{2}$. Lembrar que este procedimento é particularmente efetivo se a partição distribuiu os elementos igualmente. Como poderá ser observado nos programas, é testado apenas o desvio padrão na direção em que ele é maior. Na realidade, pode haver mais de uma direção ultrapassando o limiar de partição. Tou & Gonzalez (Ref. 8) ordenam decrescentemente os desvios e os vão testando; e, se for o ca

so, vão particionando, até obterem um desvio menor que o crítico. Sem a ordenação, espera-se, no presente trabalho, que na iteração seguinte seja detectada a outra direção de maior espalhamento e se faça devidamente a partição.

7. Combinação

A idéia agora é bastante simples - não tem sentido querer sustentar dois "clusters" cujos centros estão muito próximos porque, se isto ocorrer, provavelmente, seus elementos estarão se confundindo. O que se faz é calcular as várias distâncias entre centros e verificar quais delas são menores que o limite mínimo de distinção entre classes. Há, então, um problema: nada impede que três centros estejam próximos além do limite estipulado. Se forem calculadas as distâncias entre todos os centros e depois foram feitos os testes, uma vez combinados dois "clusters", um novo centro será definido, anulando a distância ao terceiro centro calculado anteriormente. Não haverá lógica em decidir pela combinação com base na antiga distância. Por outro lado, se for feito o teste à medida que as distâncias são calculadas, o algoritmo fica demasiadamente complicado. Sabendo que a ocorrência de três centros próximos é rara, foi descartada, de vez, a capacidade de convergir três "clusters" num só. Foi associado, a cada "cluster", o seu nº de ordem ($P(I) = I; I=1, \dots, n_{tot. clust.}$). Ao combinar com a classe J faz-se $P(I) = P(J)$. O fato de $P(I) \neq I$ será uma indicação que o "cluster" I já foi combinado e portanto o seu centro já mudou. Uma vez decidida a combinação dos "clusters" I e J o novo centro será dado por:

$$m^{(I)} = \frac{n_{pontos(I)}}{n_{pontos(I)} + n_{pontos(J)}} m(I) + \frac{n_{pontos(J)}}{n_{pontos(I)} + n_{pontos(J)}} m(J)$$

A distância para se decidir pela combinação será calculada por:

$$d_w^2(m^{(p)}, m^{(q)}) = \sum_{j=1}^K \frac{1}{\text{sig}_j^{(p)} \text{sig}_j^{(q)}} (m_j^{(p)} - m_j^{(q)})^2$$

Dois cuidados devem ser tomados neste cálculo: Se o produto dos desvios padrões for muito pequeno, a função peso pode crescer de tal maneira que anulará totalmente o efeito da diferença entre centros. Por outro lado, se for muito grande, resulta um peso muito pequeno, distorcendo novamente o teste. Toma-se, então, o seguinte caminho:

(a) Se $p = \text{sig}_j \cdot \text{sig}_i < P_{\min} \rightarrow p = P_{\min}$

(b) Se $p = \text{sig}_j \cdot \text{sig}_i \geq P_{\max} \rightarrow p = P_{\max}$

Não se sabe ainda com exatidão os valores ideais para P_{\min} e P_{\max} , porém, em caso de dados de agricultura do MSS, os valores 1. e 100., respectivamente, parecem ser satisfatórios (Ref. 14).

8. Saída dos resultados

Após a última iteração, o programa guarda a média dos "clusters" (tantas quanto o número deles) e o número de canais considerados num arquivo criado na memória do disco. O arquivo é utilizado por dois outros programas descritos em seguida.

4.4. PROGRAMAS DESTE SISTEMA

Toda a parte descrita anteriormente compreende um programa - o ISOCLT em Fortran.

Existem outros: COMSIG, SELPTO, CLTMAX, CLSISO; a disposição de cada um no sistema é mostrada na Fig. IV.2.

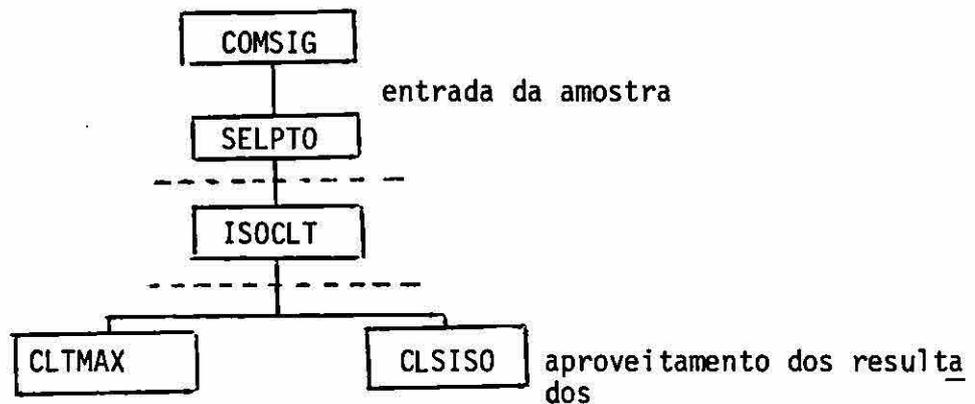


Fig. IV.2

1. COMSIG

É um programa aproveitado de outra estrutura. Não é de autoria dos presentes autores. Toma a amostra de uma imagem. Guarda num arquivo os quatro níveis de cinza de um "pixel" e o número de vezes que ele aparece na amostra. Permite amostrar uma determinada região da imagem através do cursor do I-100, no caso de conhecimento prévio de alguma classe.

2. SELPTO

Aproveita o arquivo criado pelo COMSIG e ordena os vetores nele contidos, pela ordem decrescente de vezes em que aparece na amostragem. Considera apenas os vetores cuja população (frequência na amostra) seja maior que um valor fornecido pelo usuário. Guarda num segundo arquivo apenas os níveis dos vetores mais populosos. A capacidade máxima do arquivo dos pontos selecionados é de 2.000 lugares. Os vetores servirão de amostra para o ISOCLT.

3. CLTMAX

Compatibiliza o resultado do agregamento com o sistema MAXVER (Ref. 1). Pretende-se uma aquisição de parâmetros mais automática do que aquela feita pelo MAXV01. As amostras selecionadas por SELPTO são classificadas segundo os parâmetros gerados por ISOCLT através do critério da mínima distância. Cada agrupamento irá gerar os parâmetros de uma classe requeridos pelo MAXVER

Duas razões explicam porque não se usou a classificação feita no ISOCLT:

- a) Guardar as médias significa utilizar 132 palavras, enquanto guardar a classificação pode necessitar até 1.000 palavras.
- b) O CLSISO não usa a classificação da amostra. Computa-se:
 - Matriz de correlação
 - Médias das classes são colocadas num vetor linear único
 - Matriz de covariância
 - Matriz triangular inferior
 - O determinante da matriz de covariância
 - A inversa da matriz triangular inferior. Ela é a matriz de transformação

Só não foi possível acertar a variável NUMAM e o vetor AMOSTR (390). O vetor guarda a região da tela (define o contorno) onde foi adquirida a amostra de uma determinada classe. A variável é um registro do número da aquisição. Como a aquisição é feita "automaticamente" não tem sentido se falar da região da tela nem número de aquisição.

4. CLSISO

Outra opção de utilização dos parâmetros obtidos pelo ISOCLT. Atribui a cada "pixel" lido do I-100, aquela classe cujo centro está mais próximo de suas coordenadas espectrais. Se algum "pixel" estiver distante de todos os centros, além de um limiar, será dito não pertencer a nenhuma classe. O limiar acaba definindo para uma classe um espaço de atributos de forma hiper-esférica e o que é pior, o espaço terá o mesmo volume, qualquer que seja a classe. Fica-se bem longe do hiper-elipsóide resultante do modelo gaussiano. Infelizmente não houve tempo disponível para melhorar o programa.

4.5. RESULTADOS E COMENTÁRIOS

Como teste do ISOCLT tomaram-se amostras de uma imagem também da região de Manaus (1976). Apesar do programa poder separar amostras

sem definição prévia de classes, apenas para efeito de teste foram tomadas das cinco regiões distintas que, no treinamento normalmente feito para o sistema MAXVER, representariam classes diferentes.

Utilizou-se a amostra para ser ordenada e selecionada pelo SELPTO. Considera-se o número mínimo da população igual a 1 porque se pretendia não perder nenhum elemento da amostragem.

Foram armazenados 236 vetores.

O ISOCLT separou estes vetores em 5 classes, com médias quase iguais às obtidas pelo sistema MAXVER, conforme pode ser verificado no apêndice.

Percebe-se que as médias obtidas por "clustering" foram sempre menores. Uma explicação possível para o fato é que, por agregação, fica-se menos sujeito a tomar "pixels" não pertencentes à classe, ou seja, "pixels" ruidosos.

Tomou-se uma outra amostra, agora de uma imagem do Amazonas (1977). O COMSIG gerado foi bem maior. Os elementos foram tomados de 8 regiões distintas.

Selecionada pelo SELPTO, a amostra produziu 1915 vetores distintos.

Foi experimentado agregá-los usando o valor 5 para o número mínimo de elementos para um "cluster" ser classe (tópico 2.2.c da seção 4.3).

Ao final obtiveram-se 13 classes, número bem superior às 8 tomadas. Aumentou-se o número mínimo para 25 e o resultado já foi bem melhor. Observaram-se 9 nuvens distintas. O fato de aumentar em 1 o número de classes não implica em erro porque sempre há problemas de ruído, que eventualmente geram uma nova classe.

O tempo para processar 236 vetores é relativamente peque
no, porém para 1915 ele já fica da ordem de alguns minutos.

REFERÊNCIAS BIBLIOGRÁFICAS

1. VELASCO, F.R.D.; PRADO, L.O.C.; SOUZA, R.C.M. *Sistema MAXVER: Manual do usuário.* INPE-1315-NTI/110, Jul. 1978.
2. SHLIEN, S. & SMITH, A. *A rapid method to generate spectral theme classification of LANDSAT imagery.* Remote sensing of Environment: Vol. 4(1), 1975, 67-77.
3. EPPLER, W.G.; HELMKE, C.A.; EVANS, R.H. *Table look-up approach to pattern recognition.* Proc. 7th. Symp. on Remote sensing of Environment (University of Michigan, Ann Arbor), May 1971, 1415 - 1425.
4. FU, K.S.; LANDGREBE, D.A.; PHILLIPS, T.L. *Information processing of remotely sensed agricultural data.* Proc. IEEE, Vol. 57(4), apr. 1969, 639-653.
5. AMBLE, O.; KNUTH, D.E. *Ordered hash table.* Comput. Journ., 17, 1974, 135-142.
6. KNUTH, D.E. *The art of computer programming.* Sorting and searching Vol. 3, Addison-Wesley, 1973.
7. SHLIEN, S. *Practical aspects related to automatic classification of LANDSAT imagery using look-up tables.* Research Report of Canada Centre for Remote Sensing, 75-2, Feb.
8. TOU, J.T.; GONZALEZ, R.C. *Pattern recognition principles.* Addison-Wesley, Massachusetts, 1974.
9. BALL, G.H.; HALL, D.J. *ISODATA, An iterative method of multivariate analysis and pattern recognition.* Proc. of the Int. Comun. Conf., Philadelphia, Pa., Jun., 1966.

10. KAN, E.P.F.; HOLLEY, W.A. *More on clustering techniques with final recommendations on ISODATA.* Lockheed Electronics Co., Inc., Houston, Texas, NAS 9-12200, May, 1972.
11. NARENDRA, P.M.; GOLDBERG, M. *A non parametric clustering scheme for LANDSAT.* Pattern Recognition, Vol. 9(4), 1977, 207-215.
12. KOONTZ, W.L.; NARENDRA, P.M.; FUKUNAGA, K. *A graph theoretic approach to non parametric cluster analysis.* IEEE Transaction on Comput., 25, 1976, 936-944.
13. MANNED SPACECRAFT CENTER. *IBM360/44 ISODATA clustering program*, Earth Resources Group, Information System Division, Houston, Texas, 1969.
14. KAN, E.P.F.; HOLLEY, W.A. *Experience with ISODATA.* Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Mem. TM-642-354.

APÊNDICE A

DIÁLOGOS E LISTAGENS DO SISTEMA TLUP

A seguir são apresentados os diálogos dos programas implementados. Os programas TLUP0 e TLUP1 possuem dois diálogos cada um. TLUP0 pode ser usado para inicializar nova sessão ou continuação de sessão anterior. TLUP1 pode criar nova classe com a amostra treinada ou adicionar a amostra a classe já existente. Finalmente são apresentadas as listagens.

```
>RUN TLUP0
*** INICIALIZACAO ***
NOVA SESSAO(N) OU CONTINUACAO(C)? >N

*** INICIALIZACAO TERMINADA ***

>
>RUN TLUP0
*** INICIALIZACAO ***

NOVA SESSAO(N) OU CONTINUACAO(C)? >C
COLOQUE A FITA DE PAPEL >

ENTRE C/O NOME DO ARQUIVO
DOS LIMITES E DO HISTOGRAMA >HILI.DAT
PARAMETROS DO TREINAMENTO

NO. DE DIMENSOES = 4
NO. DE CLASSES = 2
NO. DE AMOSTRAS = 2

CLASSES:
1.MATA
2.AGUA

*** INICIALIZACAO TERMINADA ***
```

>
>
>RUN TLUP1

AQUISICAO DE PARAMETROS

POSICIONE O CURSOR SOBRE A AMOSTRA >
PARAMETROS DA AMOSTRA
NO. DE PONTOS = 36.
MEDIA = 15.03 12.97 5.67 10.36
MATRIZ DE AUTOCORRELACAO = 226.64 195.75 169.75 86.72 76.11
38.28 156.44 135.78 81.56 109.42
ADICIONAR AMOSTRA(A) OU CRIAR NOVA CLASSE(C)? >C
ENTRE COM O NOME DA CLASSE >CIDADE

PARAMETROS DA CLASSE 1 CIDADE

NO. DE PONTOS = 36.
MEDIA = 15.03 12.97 5.67 10.36
MATRIZ DE COVARIANCIA:
0.80 0.81 1.56 0.74
0.81 1.47 2.60 1.37
1.56 2.60 6.17 2.84
0.74 1.37 2.84 2.06
CANAL LIM.INF. LIM.SUP.
1 13 19
2 11 18
3 1 13
4 7 15

AQUISICAO COMPLETADA

RUN TLUP1

AQUISICAO DE PARAMETROS

POSICIONE O CURSOR SOBRE A AMOSTRA >
PARAMETROS DA AMOSTRA
NO. DE PONTOS = 36.
MEDIA = 15.74 14.03 9.42 11.69
MATRIZ DE AUTOCORRELACAO = 255.39 224.61 198.36 151.31 133.14
93.69 187.42 165.22 112.11 138.69
ADICIONAR AMOSTRA(A) OU CRIAR NOVA CLASSE(C)? >A
CLASSES ATÉ AGORA

- 1. CIDADE
- 2. MATÁ
- 3. VALE

QUAL CLASSE(?)? >1

PARAMETROS DA CLASSE 1 CIDADE

NO. DE PONTOS = 72.
MEDIA = 15.49 13.50 7.54 11.03
MATRIZ DE COVARIANCIA:
1.19 1.12 2.22 1.15
1.12 1.81 2.81 1.63
2.22 2.81 9.11 3.67
1.15 1.63 3.67 2.44
CANAL LIM.INF. LIM.SUP.
1 13 20
2 11 19
3 1 16
4 7 17

AQUISICAO COMPLETADA

>RUN TLUP2

**** TLUP2: TABELAS ****

CANAL	LMIN.	LMAX.
1	10	22
2	6	24
3	1	35
4	6	38

**** TABELA I COMPLETADA

ENTRE C/O LIMIA R>S.

**** CONSTRUINDO TABELA II

**** TABELA II COMPLETADA

ENTRE C/O NOME DO ARQ. P/ TABELAS>TABELA.TAB

CLASSIFICAR NO (D)ISCO OU NA (T)ELA?>T

**** TLUP2: FIM DE EXECUCAO ****

>RUN TLUP5

ARMAZENAMENTO

ENTRE C/O NOME DO ARQUIVO
DOS LIMITES E DO HISTOGRAMA >MILI.DAT

ARMAZENAMENTO TERMINADO

>RUN TLUP3

CLASSIFICACAO

C: 0000

CLASSES:

1. CIDADE
2. NATA
3. VALE
4. AGUA
5. ENCOS
6. CIDAD2
7. PLAN.
8. NATA2

ENTRE COM AS CLASSES E OS TEMAS ASSOCIADOS
EM ORDEM DECRESCENTE DE PROBABILIDADE

CLASSE(0).TEMA(0).NAD MAIS(N) >1 1
CLASSE(0).TEMA(0).NAD MAIS(N) >2 2
CLASSE(0).TEMA(0).NAD MAIS(N) >3 3
CLASSE(0).TEMA(0).NAD MAIS(N) >4 4
CLASSE(0).TEMA(0).NAD MAIS(N) >5 5
CLASSE(0).TEMA(0).NAD MAIS(N) >6 6
CLASSE(0).TEMA(0).NAD MAIS(N) >7 7

ERRO >>> CLASSE INVALIDA
CLASSE(0).TEMA(0).NAD MAIS(N) >7 7
CLASSE(0).TEMA(0).NAD MAIS(N) >8 8

ENTRE C/O NOME DO ARQUIVO >TABELA.TAB

TEMA A TELA?(S/N) >N

POSICIONE O CURSOR >

INICIO: 13:09:32

FIN: 13:10:25

RESULTADO DA CLASSIFICACAO

CLASSE: 8 NATA2	(TEMA 8):AREA	13116.4	(16.56%)
CLASSE: 7 PLAN.	(TEMA 7):AREA	9256.4	(11.69%)
CLASSE: 6 CIDAD2	(TEMA 6):AREA	4265.4	(5.39%)
CLASSE: 5 ENCOS	(TEMA 5):AREA	23901.4	(30.19%)
CLASSE: 4 AGUA	(TEMA 4):AREA	286.4	(0.36%)
CLASSE: 3 VALE	(TEMA 3):AREA	476.4	(0.60%)
CLASSE: 2 NATA	(TEMA 2):AREA	2910.4	(3.68%)
CLASSE: 1 CIDADE	(TEMA 1):AREA	3445.4	(4.36%)

NAD CLASSIFICADOS-AREA = 15237.4 (19.34%)
NO. TOTAL DE PONTOS = 79180.

CLASSIFICACAO TERMINADA

```

C INICIALIZACAO
C
C NOME: TLUPO
C
C DESCRICAO: INICIALIZA OS PARAMETROS INDEPENDENTES DO TREINA-
C          MENTO (NO. MAX. DE CLASSES, DE AMOSTRAS E ESCALA).
C          CASO SEJA UMA NOVA SESSAO.
C          LE OS PARAMETROS DE FITA DE PAPEL CASO SEJA CONTI-
C          NUACAO E LE O HISTOGRAMA E OS LIMITES DO DISCO.
C
C SUBROTINAS: PFSUB (PRSUB,FRATT)
C
0001      IMPLICIT INTEGER(A-S-Z)
0002      LOGICAL*1 W(74),TARN(10)
0003      DOUBLE PRECISION NOMECL(16)
0004      COMMON/COMMON1/
          1 NDIM,NUMCL,NUMAM,
          2 NOMECL,DETERM(16),RNFTCL(16),MECL(64),RMECL(64),
          3 TRNSCL(160),RAUTCL(160),AMOSTR(390),
          4 NMAXCL,NMAXAM,RSCALE,FAT
0005      DIMENSION CONTAD(8,4,64),LIMI(8,4),LIMS(8,4)
C
0006      CALL ASSIGN(6,'TI:')
0007      WRITE(6,100)
0008      100  FORMAT(21X,'*** INICIALIZACAO ***',///)
C
C PARAMETROS INDEPENDENTES
C
0009      NMAXCL=8
0010      NMAXAM=78
0011      RSCALE=30.
0012      NDIM=4
C
0013      120  WRITE(6,140)
0014      140  FORMAT('NOVA SESSAO(N) OU CONTINUACAO(C)? >')
0015      CALL OUTPUT(7)
0016      READ(6,150)W
0017      150  FORMAT(74A1)
0018      CALL FRONT(W,74)
0019      IF(W(1).EQ.'103') GO TO 260
0020      IF(W(1).EQ.'116') GO TO 180
0021      GO TO 120
C
C NOVA SESSAO
C
0022      180  NUMCL=0
0023          NUMAM=0
0024          GO TO 400
C
C CONTINUACAO
C
C LEITURA DOS PARAMETROS DA FITA DE PAPEL
C
0025      260  WRITE(6,270)
0026      270  FORMAT('COLOGUE A FITA DE PAPEL >')
C
0027      READ(6,150)
0028      CALL PRATT
0029      CALL PRSUB(NDIM,4)
0030      NBAITS=B*NUMCL
0031      CALL PRSUB(NOMECL,N - IITS)
0032      NBAITS=2*NUMCL
0033      CALL PRSUB(DETERM,N - IITS)
0034      NBAITS=4*NUMCL
0035      CALL PRSUB(RNFTCL,NBAITS)
0036      NBAITS=8*NUMCL
0037      CALL PRSUB(MECL,NBAITS)
0038      NBAITS=16*NUMCL
0039      CALL PRSUB(RMECL,NBAITS)
0040      NBAITS=20*NUMCL
0041      CALL PRSUB(TRNSCL,NBAITS)
0042      NBAITS=40*NUMCL
0043      CALL PRSUB(RAUTCL,NBAITS)
0044      NBAITS=10*NUMAM
0045      CALL PRSUB(AMOSTR,NBAITS)
C
C ENTRADA DO NOME DO ARQUIVO
C DOS LIMITES E HISTOGRAMA
C
0046      271  CALL OUTPUT(7)
0047          WRITE(6,272)
0048      272  FORMAT(/,1X,'ENTRE C/O NOME DO ARQUIVO',/,
          1 'DOS LIMITES E DO HISTOGRAMA >')
0049      READ(6,150)W
0050      CALL FRONT(W,74)
0051      IF(W(1).EQ.'40')GO TO 999
0052      DO 273 I=1,10
0053          TARN(I)='40'
0054      273  CONTINUE
0055          K=0
0056          DO 274 I=1,74
0057              IF(W(I).EQ.'40')GO TO 274
0058              K=K+1
0059              IF(K.GT.10)GO TO 271
0060              TARN(K)=W(I)
0061      274  CONTINUE
C
C LER LIMITES E HISTOGRAMA DO DISCO
C
0062      NN=K
0063      CALL FORSET(2)
0064      CALL ASSIGN(2,TARN,NN)
0065      READ(2)(LIMI(I,1),I=1,32),(LIMS(J,1),J=1,32),
          1 (CONTAD(K,1),K=1,2048)
0066      ENDFILE 2
0067      CALL CLOSE(2)
C
0068      CALL OUTPUT(27,12)
0069      WRITE(6,275)
0070      275  FORMAT(1X,'PARAMETROS DO TREINAMENTO',//)

```

```
C
0071 WRITE(6,28)NDIM,NUMCL,NUMAM
0072 280 FORMAT(IX,'NO. DE DIMENSÕES =
1 IX,'NO. DE CLASSES = ',I2,/,
2 IX,'NO. DE AMOSTRAS = ',I2,/)
0073 WRITE(6,290)
0074 290 FORMAT(IX,'CLASSES: '//)
0075 DO 300 I=1,NUMCL
0076 WRITE(6,320)I,NDRECL(I)
0077 CONTINUE
0078 300 FORMAT(IX,I2,' ',AB)
C
C
C ESCREVER NOS ARQUIVOS COMUNS
C
0079 400 CALL FDRSET(4)
0080 CALL ASSIGN(4,'LIMIT.TAB')
0081 WRITE(4)(LIMI(I,1),I=1,32),(LIMS
0082 /J,1),J=1,32)
0083 CALL CLOSE(4)
C
0084 CALL FDRSET(3)
0085 CALL ASSIGN(3,'HISTOG.TAB')
0086 WRITE(3)(CONTAD(I,1),I=1,2048)
0087 ENDFILE 3
0088 CALL CLOSE(3)
C
0089 FAT=(NDIM*NDIM*NDIM)/2
0090 WRITE(6,900)
0091 900 FORMAT(///,2IX, '### INICIALIZ ACAO TERMINADA ###')
0092 CALL OUTPUT(7)
0093 999 CALL EXIT
0094 END
```

```
C AQUISICAO DE AMOSTRAS
C
C
C NOME: TLUP1
C
C DESCRICAO: CALCULA A DISTRIBUICAO DA AMOSTRA INDICADA
C PELO CURSOR, CRIANDO UMA NOVA CLASSE OU ADICIO-
C NANDO A AMOSTRA A UMA CLASSE JA EXISTENTE.
C CALCULA OS LIMITES DENTRO DOS QUAIS FICARA
C CONFINADA A CLASSE.
C
C SUBROTINAS: TLPAR
C 3LEEN
C 3LIM
C
0001 IMPLICIT INTEGER(A-D,S-Z)
0002 LOGICAL L1,L2(74)
0003 DOUBLE PRECISION NOMECL(16)
0004 COMMON/COM1/
0005 1 NDIM,NUMCL,NUMAM,
0006 2 MORECL,ME.TI(16),RMPTCL(16),MEDCL(64),RMEDCL(64),
0007 3 TRANSCL(160),RAUFEL(160),AMOSTR(390),
0008 4 MAXCL,MAXAM,FCDL,FAT
0009 DIMENSION CURSOR(5)
0010 COMMON/P/PAH/P/PTAM,RMEDAM(4),RAUTAM(10)
0011 COMMON/CDU/XXI,YF,YI,YF
0012 COMMON/HIST/CONT(4,64),CONTAD(8,4,64),MEDIA(4),
0013 1 LIMI(8,4),LIMS(8,4)
C
C
0009 RPNMAX=40
0010 CALL ASSIGN(5,'LPT')
0011 CALL ASSIGN(6,'TI')
0012 CALL OUTPUT(27,12)
C
C LER HISTOGRAMA E LIMITES DO DISCO
C
0013 CALL FDRSET(3)
0014 CALL ASSIGN(3,'HISTOG.TAB')
0015 READ(3)(CONTAD(I,1),I=1,2048)
0016 ENDFILE 3
0017 CALL CLOSE(3)
0018 CALL FDRSET(4)
0019 CALL ASSIGN(4,'LIMIT.TAB')
0020 READ(4)(LIMI(I,1),I=1,32),(LIMS(J,1),J=1,32)
0021 ENDFILE 4
0022 CALL CLOSE(4)
C
0023 WRITE(6,60)
0024 60 FORMAT(2IX, '###AQUISICAO DE PARAMETROS###')
0025 CALL OUTPUT(7)
0026 IF (NUMAM.LT.MAXAM) GO TO 73
0027 WRITE(6,70)
0028 70 FORMAT(IX,'EPRO >>> EXCEDEU NO. MAXIMO DE AMOSTRAS')
0029 CALL OUTPUT(7)
0030 GO TO 999
```

```
C
0031 73 WRITE(6,75)
0032 75 FORMAT('POSICIONE O CURSOR SOBRE A AMOSTRA >')
0033 CALL OUTPUT(7)
0034 READ(6,120)M

C
C LER CURSOR
C
0035 80 CALL IKK(CURSOR)
C
C TESTAR SE O MODO ESTÁ CERTO
C
0036 IF(CURSOR(1).EQ.1)GO TO 95
0037 WRITE(6,90)
0038 90 FORMAT(1X,'ERRO >>> CURSOR NAO ESTA NO MODO RETANGULO')
0039 CALL OUTPUT(7)
0040 GO TO 73

C
C CALCULO DOS PARAMETROS DA AMOSTRA
C
C CALCULO DO NO. DE FONTOS
C
0041 95 RC3=CURSOR(3)
0042 RCS=CURSOR(5)
0043 RNPAN=164AC3+RCS+89*(RC3+RCS)+4

C
C CALCULO DOS LIMITES XI,XF,YI,YF
C
0044 XI=CURSOR(2)-2*CURSOR(3)-2
0045 XF=CURSOR(2)+2*CURSOR(3)-1
0046 YI=CURSOR(4)-2*CURSOR(5)
0047 YF=CURSOR(4)+2*CURSOR(5)+1

C
0048 100 CALL TLPAR
C
C APRESENTACAO DOS PARAMETROS DA AMOSTRA
C
0049 WRITE(6,103)RNPAN,(RNPAN(I),I=1,NDIM)
0050 103 FORMAT(1X,'PARAMETROS DA AMOSTRA',/,
1 1X,'NO. DE FONTOS = ',F7.0,/,
2 1X,'MEDIA = ',F4.2)
0051 WRITE(6,107)(RNPAN(I),I=1,FAT)
0052 107 FORMAT(1X,'MATRIZ DE AUTOCORRELAÇÃO = ',5F8.2,/,2BX,5FB,
2)

C
C NOVA CLASSE?
C
0053 105 WRITE(6,110)
0054 110 FORMAT('ADICIONAR AMOSTRA(A) OU CRIAR NOVA CLASSE(C)?')
0055 CALL OUTPUT(7)
0056 READ(6,120)M
0057 120 FORMAT(74)
0058 CALL FRONT(6,74)
0059 IF(M(1).EQ.-103)GO TO 400
0060 IF(M(1).EQ.-101)GO TO 130
0061 IF(M(1).EQ.-130)GO TO 999
0062 GO TO 105

C
C ADICIONAR AMOSTRA
C
0063 130 IF(MUCL.EQ.0) GO TO 135
0064 WRITE(6,133)
0065 133 FORMAT(1X,'ERRO >>> NAO HA CLASSES')
0066 CALL OUTPUT(7)
0067 GO TO 105

C
0068 135 CALL OUTPUT(27,12)
C
0069 WRITE(6,140)
0070 140 FORMAT(1X,'CLASSES ATE AGORA',/)
0071 DO 150 I=1,MUCL
0072 WRITE(6,160)I,NOMECL(I)
0073 CONTINUE

C
0074 160 FORMAT(1X,I2,' ',F4B)
0075 170 WRITE(6,175)
0076 175 FORMAT(1X,'QUAL CLASSE(0) >')
0077 CALL OUTPUT(7)
0078 READ(6,120)M
0079 CALL FRONT(6,74)
0080 IF(M(1).EQ.-130)GO TO 999

C
0081 I=0
0082 CALL INTFF(I,M,74,CLAT)
0083 IF (CLAT.GE.1.AND.CLAT.LE.MUCL)GO TO 178
0084 WRITE(6,177)
0085 177 FORMAT(1X,'ERRO >>> CLASSE INVALIDA')
0086 CALL OUTPUT(7)
0087 GO TO 170

C
C ADICIONAR AMOSTRA
C
C
C CALCULO DOS LIMITES DOS PARAMETROS DA CLASSE CLAT:
C J1,J2 - RMEICL
C J1,J2 - RNUICL
C
0088 178 J1=4+(CLAT-1)+1
0089 J2=11*NDIM-1
0090 J1=10+(CLAT-1)+1
0091 J2=11*FAT-1

C
C ATUALIZACAO DOS PARAMETROS DA CLASSE
C
0092 DO 179 J=1,NDIM
0093 DO 179 K=1,64
0094 CONTAD(CLAT,J,K)=CONTAD(CLAT,J,K)+CONT(J,K)
0095 179 CONTINUE
0096 NT=FNPTEL(CLAT)+RNPAN
0097 PC=FNPTEL(CLAT)/RT
0098 PA=FNPAN/RT
0099 PNPTEL(CLAT)=RT
```

```
0100      J=0
0101      DO 100 I=J1,I2
0102      J=J+1
0103      RNECL(I)=RABR*RDAR(J)+RCBR*RNEDCL(I)
0104      MEDIA(J)=INTCNV(RNECL(I))
0105      100 CONTINUE
      :
0106      J=0
0107      DO 200 I=J1,J2
0108      J=J+1
0109      RAUTCL(I)=RABRAUTAH(J)+RCBRAUTCL(I)
0110      200 CONTINUE
0111      GO TO 800
      C
      C CRIAR NOVA CLASSE
      C
0112      400 IF(NUMCL.LT.NMAXCL)GO TO 420
0113      WRITE(6,410)NMAXCL
0114      410 FORMAT(1X,'ERRO >> NO. MAXIMO DE CLASSES EM ',I3)
0115      CALL OUTPUT(7)
0116      GO TO 105
      C
0117      420 NUMCL=NUMCL+1
0118      CLAT=NUMCL
0119      DO 425 J=1,NDIM
0120      DO 425 K=1,64
0121      CONTAD(CLAT,J,K)=CONT(J,K)
0122      425 CONTINUE
      C
0123      I1=4+(CLAT-1)+1
0124      I2=I1+NDIM-1
0125      J1=10*(CLAT-1)+1
0126      J2=J1+FAT-1
      C
      C NOME DA NOVA CLASSE
      C
0127      WRITE(6,430)
0128      430 FORMAT('NITRE COM O NOME DA CLASSE >>')
0129      CALL OUTPUT(7)
0130      READ(6,440)NONECL(CLAT)
0131      440 FORMAT(A8)
      C
      C PREENCHER DADOS DA CLASSE I
      C RNPTCL+RNECL+RAUTCL
      C
0132      RNPTCL(CLAT)=RNPTAN
      C
0133      J=0
0134      DO 450 I=I1,I2
0135      J=J+1
0136      RNECL(I)=RNEFDAR(J)
0137      MEDIA(J)=INTCNV(RNECL(I))
0138      450 CONTINUE
      C
0139      J=0
0140      DO 460 I=J1,J2
      :
0141      J=J+1
0142      RAUTCL(I)=RAUTAH(J)
0143      460 CONTINUE
      C
      C CALCULO DO LIMITE (LIM1:LIM2)
      C
0144      800 CALL TLEM(CLAT)
      C
      C PREENCHER DADOS DA AMOSTRA
      C CLAT=XI-YF-YI-YF
      C
0145      NUMAN=NUMAN+1
0146      AN=SPNUMAN
0147      AMOSTRAN(4)=CLAT
0148      AMOSTRAN(3)=XI
0149      AMOSTRAN(2)=YF
0150      AMOSTRAN(1)=YI
0151      AMOSTRAN(5)=YF
      D
0152      WRITE(5,B1)CLAT
0153      8014 FORMAT(1X,'CLASSE N.',I2,/)
      D
0154      DO 816 I=1,64
0155      WRITE(5,B15)I,(CONTAD(CLAT,K,I),K=1,4)
0156      816 CONTINUE
0157      815 FORMAT(5(1X,I5))
      C
      C GUARDAR HISTOGRAMA NO DISCO
      C
0158      CALL FDBSET(3,'HISTOGRAMA')
0159      CALL ASSIGN(3,'HISTOGRAMA')
0160      WRITE(3)(CONTAD(I,1),I=1,2048)
0161      ENDFILE 3
0162      CALL CLOSE(3)
      C
      C GUARDAR LIMITES NO DISCO
      C
0163      CALL FDBSET(4,'LIMITES')
0164      CALL ASSIGN(4,'LIMITES')
0165      WRITE(4)(LIM1(I),I=1,32),(LIM2(J),J=1,32)
0166      ENDFILE 4
0167      CALL CLOSE(4)
      C
      C CALCULO DA MATRIZ DE TRANSFORMACAO
      C
0168      CALL TLEM(CLAT)
      C
      C DISPLAY DOS PARAMETROS
      C
0169      WRITE(6,830)CLAT,NUMCL(CLAT),RNPTCL(CLAT),(RNECL(I),I=
0170      I1,I2)
      D
0171      WRITE(6,840)(MEDCL(I),I=I1,I2)
0172      WRITE(6,850)(RAUTCL(J),J=J1,J2)
0173      WRITE(6,855)
0174      DO 820 I=1,NDIM
0175      WRITE(6,857)I,LIM1(CLAT,I),LIM2(CLAT,I)
0176      820 CONTINUE
0177      WRITE(6,860)(TRNSCL(J),J=J1,J2)
```

```
0030 FORMAT(IX,'PARAMETROS DA CLASSE',I3,2X,AB//,
D 1 IX,'NO. DE PUNTOS = ',I7,0//,
D 2 IX,'MEDIA = ',D4,2)
0040 FORMAT(IX,'MEDCL',I4)
0050 FORMAT(IX,'MATRIZ DE AUTOCORRELACAO = ',5F8,2//,20X,5F8,
2)
0147 055 FORMAT(//,20X,'CANAL',I3,'LIN. INF.',I3,'LIN. SUP.')
```

```
0148 057 FORMAT(30X,I1,9X,I2,10X,I2)
0060 0640 FORMAT(IX,'TRANSCL',I10I2)
C
C
0169 070 WRITE(6,B70)
0170 070 FORMAT(///,2IX,'***ADQUISICAO COMPLETADA***')
0171 CALL OUTPUT(7)
C
0172 999 CALL EXIT
0173 END
```

```
C DISTRIBUICAO
C
C NOME: TLPAR
C
C DESCRICAO: CALCULA OS SEQUINTES PARAMETROS: MEDIA (RMEAN),
C MATRIZ DE AUTOCORRELACAO (RAUTAN) DA AMOSTRA DADA
C PELOS LIMITES XI, XF, YI, YF.
C CALCULA O HISTOGRAMA DA AMOSTRA (CONT).
C
0001 SUPPORTIVE TLPAR
0002 IMPLICIT INTEGER(A-D,S-Z)
0003 LOGICAL*1 BUF1(512),BUF2(512),BUF3(512),BUF4(512)
0004 DOUBLE PRECISION NOMECL(14)
0005 COMMON/COMN1/
I NDIR,MNCL,NMAM,
I NMECL,DETER(16),RNFCL(16),MEDCL(64),RMECL(64),
I TRNSCL(160),RAUTCL(160),AMOSTR(370),
4 MNAXCL,MNAXAM,ASCALA,FAT
COMMON/COM2/XI,XF,YI,YF
COMMON/NIST/CONT(4,64),CONTAD(8,4,64),MEDIA(4),
I LIMI(8,4),LIMS(8,4)
COMMON/PARAM/RMPTAN,RMEDAN(4),RAUTAN(10)
0006 DIMENSION RPTO(4),PTO(4)
0007
C
C INICIALIZACAO
C
0010 DO 10 I=1,NDIR
0011 RMEAN(I)=0.
0012 DO 10 J=1,64
0013 CONT(I,J)=0
0014 10 CONTINUE
C
0015 DO 15 I=1,FAT
0016 RAUTAN(I)=0.
0017 15 CONTINUE
C
0018 DO 40 LINHA=YI,YF
C
C LEITURA DAS LINHAS
C
0019 CALL IRV(1,LINHA,BUF1)
0020 CALL WAIT
0021 CALL IRV(2,LINHA,BUF2)
0022 CALL WAIT
0023 CALL IRV(3,LINHA,BUF3)
0024 CALL WAIT
0025 CALL IRV(4,LINHA,BUF4)
0026 CALL WAIT
0027 DO 40 COLUN=XI,XF
C
C LEITURA DO PONTO
C
0028 PTO(1)=IBYTE(0,PUF1(COLUN))/4
0029 PTO(2)=IBYTE(0,PUF2(COLUN))/4
0030 PTO(3)=IBYTE(0,PUF3(COLUN))/4
```

```
0031      PTO(4)=IBYTF(0,SUF4(COLUN))/4
C
C ACUMULAR MEDIA E FAZER HISTOGRAMA
C
0032      DO 25 I=1,4
0033      RPTO(I)=PTO(I)
0034      DO 25 I=1,NDIM
0035      RMEDAN(I)=RMEDAN(I)+RPTO(I)
0036      CONT(I,PTO(I)+1)=CONT(I,PTO(I)+1)+1
0037      CONTINUE
0038      DO 25 I=1,4
C
C ACUMULAR AUTOCORRELAÇÃO
C
0039      K=0
0040      DO 30 I=1,NDIM
0041      DO 30 J=1,I
0042      K=K+1
0043      RAUTAN(K)=RAUTAN(K)+RPTO(I)*RPTO(J)
0044      CONTINUE
0045      CONTINUE
C
C CALCULO DA MEDIA
C
0046      DO 50 I=1,NDIM
0047      RMEDAN(I)=RMEDAN(I)/RNPTAN
0048      CONTINUE
C
C CALCULO DA AUTOCORRELAÇÃO
C
0049      DO 60 I=1,FAT
0050      RAUTAN(I)=RAUTAN(I)/RNPTAN
0051      CONTINUE
C
0052      RETURN
0053      END
```

```
C PARÂMETROS DA CLASSIFICAÇÃO
C
C NOME: TLELEH
C
C DESCRIÇÃO: CALCULA, A PARTIR DA AUTOCORRELAÇÃO DA CLASSE CLAT
2      (RAUTCL(J1,J2)) E DA MEDIA (RMEDCL(I1,I2)), A MATRI
C      DE TRANSFORMAÇÃO (TRNSCL(J1,J2)) DA CLASSE CLAT, A
MEDIA      (MEDCL(I1,I2)) E O LOGARITMO DO DETERMINANTE
C      (DETERM(CLAT)).
C
C SUBROTINAS: INTCRV
C
*0001      SUBROUTINE TLELEH(CLAT)
0002      IMPLICIT INTEGER(A-G,S-Z)
0003      REAL*8 DO
0004      DOUBLE PRECISION NOMECL(16)
0005      COMMON/COMPI/
1      NDIM,NUMCL,NMVAR,
2      NMECL,NELEM(16),RNFYCL(16),MEDCL(64),RMEDCL(64),
3      TRNSCL(160),RAUTCL(160),AMOSTR(390),
4      NMXCL,NMXAN,NMCAI,FAT
0006      DIMENSION RCOVAR(10),RELE(10),RELEI(10),RSAI(4)
C
0007      I1=4*(CLAT-1)+1
0008      I2=I1+NDIM-1
0009      J1=10*(CLAT-1)+1
0010      J2=J1+FAT-1
C
C CALCULO DA MATRIZ DE COVARIANCIA
C
C CALCULO DE NMECL
C
0011      K=0
0012      DO 10 I=I1,I2
0013      DO 10 J=I1,I1
0014      K=K+1
0015      RCOVAR(K)=RMEDCL(I)*RMEDCL(J)
0016      CONTINUE
C
0017      J=0
0018      DO 20 I=J1,J2
0019      J=J+1
0020      RCOVAR(J)=RAUTCL(I)-RCOVAR(J)
0021      CONTINUE
D      WRITE(6,25)(RCOVAR(J),J=1,FAT)
D25      FORMAT(1X,'COV',/,1X,10F7,2)
C
C CALCULO DE L
C
0022      IF(RCOVAR(1).LE.0)GO TO 200
0023      NELL(1)=SUM(RCOVAR(1))
0024      DO 30 I=1,NDIM
0025      J=(I-1)/2+1
0026      RELE(I)=RCOVAR(J)/RELE(1)
0027      CONTINUE
```

```
C
C
0028 DO 40 I=2,NDIM
0029 L=(I+1-1)/2
0030 DO 40 J=2,I
0031 R=(J+J-1)/2
0032 RF=0.
0033 K2=J-1
0034 DO 35 K=1,K2
0035 RF=RF+RELE(L,K)*RELE(M,K)
0036 CONTINUE
0037 IF(I.EQ.J) GO TO 37
0038 IF(RELE(M+J).EQ.0) GO TO 200
0039 RELE(L+J)=(KCOVAR(L+J)-RF)/RELE(M+J)
0040 GO TO 40
0041 37 RF=RCOVAR(L+J)-AF
0042 IF(RF.LT.0) GO TO 200
0043 RELE(L+J)=SORT(RF)
0044 40 CONTINUE
D WRITE(6,41)(RELE(J),J=1,PAT)
041 FORMAT(1X,'ELE',/,/,1X,10F7.2)
C
C CALCULO DE DETERM
C
0045 RDET=1.
0046 DO 50 I=1,NDIM
0047 J=I+1/2
0048 RDET=RDET*RELE(J)
0049 CONTINUE
D WRITE(8,55)RDET
055 FORMAT(1X,'RDET=',F7.2)
0050 RDETQ=RDET*KDET
0051 IF(RDETQ.EQ.0) GO TO 200
0052 RDQL=LOG(RDETQ)
D WRITE(4,60)RDQL
060 FORMAT(1X,'RDQL',F7.2)
0053 RTEMP=RSCALE*SCALE*RNOL
0054 DETERM(CLAT)=INTCNVTEMP)
D WRITE(4,65)DETERM(CLAT)
065 FORMAT(1X,'DETERM=',18)
C
C INVERSAD DE L
C
0055 DO 150 I=1,NDIM
0056 DO 150 JJ=1,I
0057 J=I-JJ
0058 L=(I+1-1)/2+J
0059 IF(I.EQ.J)GO TO 145
0060 RF=0.
0061 K2=I-1
0062 DO 143 K=J,K2
0063 M=(K+K-1)/2+J
0064 N=(I+1-1)/2+K
0065 N=RF+HILE(N)*RELE(M)
0066 CONTINUE
0067 143 N=(M+M)/2
RELE(M)=(N-RF)/RELE(M)
0068 GO TO 150
0069 IF (RELE(L).EQ.0) GO TO 200
0070 RELE(M)=(N-1)/RELE(L)
0071 CONTINUE
0072 150 CONTINUE
C
D WRITE(4,155)(RELE(J),J=1,PAT)
0155 FORMAT(1X,'ELE1',/,/,1X,10F7.2)
C
C ESCALE E GUARDE A INVERSA DE L
C
0073 J=0
0074 DO 160 I=J1,J2
0075 J=J+1
0076 RTEMP=RSCALE*RELE(M)
0077 TANSCL(I)=INTCNVTEMP)
0078 160 CONTINUE
C
C GUARDAR MEDIAS DAS CLASSES
C
0079 DO 170 I=1,I2
0080 MEDCL(I)=INTCNV(RMEDCL(I))
0081 170 CONTINUE
C
C APRESENTACAO DOS NOVOS PARAMETROS
C
0082 WRITE(6,172)CLAT,HOMECL(CLAT)
0083 172 FORMAT(//,1X,'PARAMETROS DA CLASSE',I3,2X,AB,/)
0084 WRITE(6,174)RNFTCL(CLAT),(RMEDCL(I),I=1,I2)
0085 174 FORMAT(14,'NO. DE PONTOS = ',F7.0,/,
1 IX,'MEDIA = ',F8.2)
0086 WRITE(6,176)
0087 176 FORMAT(1X,'MATRIZ DE COVARIANCIAS')
C
DO 184 I=1,NDIM
0088 DO 182 J=1,NDIM
0089 IF (I.LT.J) GO TO 178
0090 A=(I+1-1)/2+J
0091 GO TO 180
0092 B=(J+J-1)/2+I
0093 178 KSAI(J)=RCOVAR(K)
0094 180 KSAI(J)=RCOVAR(K)
0095 CONTINUE
0096 WRITE(6,184)(KSAI(J),J=1,NDIM)
0097 184 FORMAT(23X,F8.2)
0098 184 CONTINUE
C
C
0099 RETURN
C
0100 WRITE(4,300)
0101 300 FORMAT(1X,'AVISO >>> AMOSTRA INSUFICIENTE PARA
1 'CLAT'DIFICACAO')
0102 CALL DUFFUT(7)
C
0103 RETURN
0104 END
```

```

C SUB-ROTINA
C
C NOME: TLIN
C
C DESCRICAO: SUB-ROTINA PARA CALCULAR OS LIMITES
C DA CLASSE (CLAT) EM CADA CAML.
C
0001 SUBROUTINE TLIN(CLAT)
0002 IMPLICIT INTEGER(A-D,S-Z)
0003 COMMON/COMMI/
1 NDIR,MUMCL,MUMAM,
2 NOMECL,DETERM(16),RNPTCL(16),MEDCL(64),RMEDCL(64),
3 TRNCL(160),RAUTCL(160),AMOST(390),
0004 NMAXCL,NMAXAM,RSCLAL,FAT
COMMON/HLST/CONT(4,64),CONTAD(8,4,64),MEDIA(4),
1 LIMI(8,4),LIMS(8,4)
C
0005 DO 50 J=1,NDIR
0006 RNPT=0
0007 DO 4 I=1,64
0008 RNPT=RNPT+CONTAD(CLAT,J,I)
0009 CONTINUE
0010 RLIMI=RNPT-1.
0011 M=MEDIA(J)+1
0012 RM=CONTAD(CLAT,J,M)/2.
0013 RSONA=RHM
0014 DO 5 I=1,M-1
0015 RSONA=RSONA+CONTAD(CLAT,J,I)
0016 CONTINUE
0017 RLIMI=RLIMI+RSONA/RNPT
0018 RSONA=RHM
0019 DO 4 I=M+1,64
0020 RSONA=RSONA+CONTAD(CLAT,J,I)
0021 CONTINUE
0022 RLIM=RLIMI+RSONA/RNPT
0023 IF(M.GT.2)GO TO 7
0024 LIMI(CLAT,J)=1
0025 GO TO 25
0026 7 RSONA=RHM
0027 DO 10 I=M-1,2,-1
0028 IF(RSONA.GT.RLIMI)GO TO 20
0029 RSONA=RSONA+CONTAD(CLAT,J,I)
0030 CONTINUE
0031 10 LIMI(CLAT,J)=I
0032 20 IF(M.LT.6)GO TO 26
0033 LIMS(CLAT,J)=64
0034 GO TO 50
0035 24 RSONA=RHM
0036 DO 30 I=M+1,63
0037 IF(RSONA.GT.RLIM)GO TO 40
0038 RSONA=RSONA+CONTAD(CLAT,J,I)
0039 30 CONTINUE
0040 40 LIMS(CLAT,J)=I
0041 50 CONTINUE
0042 RETURN
0043 END

```

C CALCULO DE TABELAS

C NOME: TLUP2

C DESCRICAO: A PARTIR DOS LIMITES MONTA A TABELA I E,
PELO CRITERIO DE MAXIMA VELOCIDADE,
A TABELA II. CASO VA SER FEITA CLASSIFI-
CACAO NO DISCO, GUARDA PARAMETROS NO DISCO.

C SUB-ROTINAS: MONI2
INTEN2

```

0001 IMPLICIT INTEGER(A-D,S-Z)
0002 LOGICAL B1 (174),TAB1(8,4,64),TAB2(10,10,10,10),TABN(10)
0003 LOGICAL B11 MONI(15)
0004 DOUBLE PRECISION NOMECL(16)
0005 COMMON/COMMI/
1 NDIR,MUMCL,MUMAM,
2 NOMECL,DETERM(16),RNPTCL(16),MEDCL(64),RMEDCL(64),
3 TRNCL(160),RAUTCL(160),AMOST(390),
0006 NMAXCL,NMAXAM,RSCLAL,FAT
0007 DIMENSION P(4),MYC(10),LIMI(8,4),LIMS(8,4),PTCL(9)
DIMENSION LMIN(4),LMAX(4)
C
C INICIALIZACAO
C
C LEITURA DOS LIMITES DO DISCO
0008 CALL FORSET(4)
0009 CALL ASSIGN(4,'LIMI.TAB')
0010 READ(4)((LIMI(I,1),I=1,32),(LIMS(J,1),J=1,32)
0011 ENDFILE 4
0012 CALL CLOSE(4)
C
0013 CALL ASSIGN(6,'TII')
0014 CALL OUTPUT(37,12)
0015 WRITE(6,10)
0016 10 FORMAT(21X,'===== TABELAS =====')
C
C CALCULO DA MENOR REGIAO QUE CONTEM
C TODAS AS REGIOES DE INTERESSE.
C
0017 DO 40 J=1,NDIR
0018 LMIN(J)=LIMI(1,J)
0019 DO 30 I=1,MUMCL
0020 IF(LIMI(I,J).GE.LMIN(J))GO TO 30
0021 LMIN(J)=LIMI(I,J)
0022 CONTINUE
0023 IF(LMIN(J).EQ.1)GO TO 31
0024 LMIN(J)=LMIN(J)+1
0025 31 LMAX(J)=LIMS(1,J)
0026 DO 35 I=2,MUMCL
0027 IF(LIMS(I,J).LE.LMAX(J))GO TO 35

```

```
0028 LMAX(J)=LIMS(I,J)
0029 35 CONTINUE
0030 IF(LMAX(J).EQ.64)GO TO 40
0031 LMAX(J)=LMAX(J)-1
0032 40 CONTINUE
C
0033 WRITE(4,43)
0034 43 FORMAT(//,IX,'CANAL LMIN, LMAX,')
0035 DO 44 I=1,NDIM
0036 WRITE(4,45)I,LMIN(I),LMAX(I)
0037 44 CONTINUE
0038 45 FORMAT(IX,J(12,5X))
C
C CALCULO DAS MASCARAS P/ ESCREVER
C NA TABELA II.
0039 DO 50 I=1,NUMCL
0040 MASC(I)=2*(I-1)
0041 50 CONTINUE
C
0042 RLIMAX=181.019/RSCALA
C
C
C MONTAR TABELA I
C
0043 DO 120 I=1,NUMCL
0044 DO 120 J=1,NDIM
0045 IF(LIMI(I,J).GT.1)GO TO 90
0046 LIM(I,J)=0
0047 GO TO 105
0048 90 DO 100 NIV=1,LIMI(I,J)
0049 TABI(I,J,NIV)=0
0050 100 CONTINUE
0051 105 IF(LIMS(I,J).LT.64)GO TO 107
0052 LIMS(I,J)=65
0053 GO TO 115
0054 107 DO 110 NIV=LIMS(I,J)+64
0055 TABI(I,J,NIV)=0
0056 110 CONTINUE
0057 115 DELTA=(LIMS(I,J)-LIMI(I,J))-1
0058 RDELTA=DELTA
0059 RD=2.
0060 D=2
0061 IF(DELTA.LE.11)GO TO 112
0062 IF(DELTA.LE.25)GO TO 111
0063 D=INTCNV(RDELTA/P.)
0064 RD=D
0065 111 PT=1.56/RD
0066 RAZAO=P./RDELTA
0067 IF(RAZAO.GT.RT)GO TO 112
0068 RDELTA=RDELTA+.1
0069 RD=RD+.1
0070 GO TO 111
0071 112 INIC=LIMI(I,J)+D-RD
0072 RAZAO=P./RDELTA
0073 DO 120 NIV=LIMI(I,J)+1,LIMS(I,J)-1
0074 RDELTA=NIV-INIC

0075 TABI(I,J,NIV)=INTCNV(RDELTA*RAZAO)
0076 CONTINUE
0077 WRITE(4,125)
0078 125 FORMAT(//,IX,'#### TABELA I COMPLETADA')
C
C
0079 CALL OUTPUT(7)
C
C LEITURA DO LIMIAZ
C
0080 130 CALL OUTPUT(7)
0081 WRITE(4,140)
0082 140 FORMAT(//,'ENTRE C/O LIMIAZ:')
0083 CALL OUTPUT(7)
0084 READ(4,150)M
0085 150 FORMAT(74I1)
0086 CALL FRONT(M,74)
0087 IF(M(1).EQ.'130)GO TO 999
0088 I=0
0089 CALL REAUFF(I,M,74,RLIM)
0090 IF(PLIM.LE.RLIMAX)GO TO 170
0091 WRITE(4,160)ALIMAX
0092 160 FORMAT(IX,'EQU ',' LIMIAZ MAXIMO EM:',F8.2)
0093 GO TO 130
0094 170 THRESH=RLIM*RLIM/RSCALA*PRSCALA
C
C MONTAR TABELA II
C
0095 WRITE(4,180)
0096 180 FORMAT(//,IX,'#### CONSTRUINDO TABELA II')
C
0097 CALL MONTA2(NUMCL,THRESH,TRNSCL,MECL,
I DETERM,LMIN,LMAX,MASC,TAR1,TAR2)
0098 215 WRITE(4,220)
0099 220 FORMAT(//,IX,'#### TABELA II COMPLETADA')
C
C
C ENTRADA DO NOME DO ARQUIVO P/ TABELAS.
C
0100 230 CALL OUTPUT(7)
0101 WRITE(4,240)
0102 240 FORMAT(//,'ENTRE C/O NOME DO ARG. P/ TABELAS:')
0103 READ(4,150)M
0104 CALL FRONT(M,74)
0105 IF(M(1).EQ.'40)GO TO 999
0106 DO 250 I=1,10
0107 TAP(I)=*0
0108 250 CONTINUE
0109 N=0
0110 DO 260 I=1,74
0111 IF(M(I).EQ.'40)GO TO 260
0112 N=N+1
0113 IF(N.GT.10)GO TO 230
0114 TAP(N)=M(I)
0115 260 CONTINUE
0116 MN=N
```

```

C
C 'AI CLASSIFICAR NO DISCO OU NA TELAS
C
0117 CALL OUTPUT(37,12)
0118 CALL OUTPUT(7)
0119 WRITE(A,310)
0120 31 FORMAT(//,'CLASSIFICAR NO (D)DISCO OU NA (T)TELAS')
0121 REW(4,150)W
0122 CALL FRONT(N,74)
0123 IF(M(1).EQ.'124)GO TO 300
0124 IF(M(1).EQ.'104)GO TO 400
0125 GO TO 300
C
C LER NOME DA IMAGEM
C
0126 400 CALL OUTPUT(7)
0127 WRITE(A,400)
0128 405 FORMAT(//,'ENTRE C/O NOME DA IMAGEM >')
0129 REW(4,150)W
0130 IF(M(1).EQ.'401)GO TO 999
0131 DO 410 I=1,15
0132 NOME(I)=' '
0133 410 CONTINUE
0134 K=0
0135 DO 420 I=1,74
0136 IF(M(I).EQ.'401)GO TO 420
0137 K=K+1
0138 IF(K.GT.15)GO TO 400
0139 NOME(K)=M(I)
0140 420 CONTINUE
C
C CLASSIFICACAO DO DISCO:
C GUARDAR PARAMETROS E TABELAS NO DISCO.
C
0141 CALL FDISK(2)
0142 CALL ASSIGN(2,TABN,NM)
0143 WRITE(2)(TAB1(I,1:1),I=1,2048),(TAB2(J,1:1),J=1,10000)
0144 WRITE(2)NMCL
0145 WRITE(2)(NOME(I),I=1,8),(NOME(J),J=1,15)
0146 ENDFILE 2
0147 CALL CLOSE(2)
0148 GO TO 600
C
C CLASSIFICACAO DA TELA:
C GUARDAR AS TABELAS NO DISCO.
C
0149 500 CALL FDISK(2)
0150 CALL ASSIGN(2,TABN,NM)
0151 WRITE(2)(TAB1(I,1:1),I=1,2048),(TAB2(J,1:1),J=1,10000)
0152 ENDFILE 2
0153 CALL CLOSE(2)
C
C
0154 600 WRITE(A,600)
0155 800 FORMAT(///,20X,' *** TLUF? FIN DE EXECUCAO ***')
0156 999 CALL EXIT
0157 END

```

```

) CALL MONTB2(NMCL,THRESH,TRNSCL,MEDCL,
) DETERM,LMIN,LMAX,RASC,TAB1,TAB2)
)
) FUNCAO:
) MONTA A TABELA II, CLASSIFICANDO
) AFERIR OS PONTOS DO ESCALO ESPECIFICO CON-
) FERINDO ENTRE LMIN E LMAX, PARA CADA
) CANAL, UTILIZANDO A TABELA I, A CLAS-
) SIFICACAO E FEITA PELO CRITERIO DE VEROS-
) SIMILHANCA, DESCRITO A SEGUIR.
)
) DESCRICAO:
) PARA CADA CLASSE, A SUBROTINA ADA O PONTO TRANSFORMADO PTO'
) ATRAVES DA FORMULA PTO'=TANSEI*(1/3,4)-MEDCL). AEM ENTAO CALCULA
) A NORMA DA QUADRADO DE PTO' (PTO'(1)+2*PTO'(2)+PTO'(3)+PTO'(4)+2)
) SOMA COM DET E COMPARA COM DTNO A MEDIDA QUE VAO SENDO
) CALCULADAS AS PARCELAS DA SOMA.
) TAO LOGO ESTA SEJA MAIOR QUE DTNO (INICIALMENTE
) FEITO IGUAL A THRESH, QUE EM O LIMIAR DO ESCALADO), OUTRA
) CLASSE EN TESTADA, SE A SOMA FOR MENOR QUE DTNO, A CLASSE EM ATUA-
) LIZADA, SEM CINGO DTNO, QUE PASSA A SER A SOMA, OU SEJA, A NORMA QU-
) BRADA DE PTO' MAIS DET.
)
)
) .TITLE MONTB2
)
)
) NMCL: 0
) CLRES: 0
) TESTE: 0
) THRESH: 0
) TRNSCL: 0
) TPN: 0
) MEDCL: 0
) MED: 0
) CSEL: 0
) DETERM: 0
) DET: 0
) RASC: 0
) DIF1: 0
) DIF2: 0
) DIF3: 0
) DTNO: 0
) P1: 0
) P2: 0
) P3: 0
) P4: 0
) LMIN1: 0
) LMIN2: 0
) LMIN3: 0
) LMIN4: 0
) LMAX1: 0
) LMAX2: 0
) LMAX3: 0
) LMAX4: 0
) ETD: .PLW 8.
) DLCL: .PLW 8.
) TAB1: 0
) TAB2: 0

```



```
CLASSIFICACAO DA TELA
00401 TLUP)
DESCRICAO: CLASSIFICA A PARTE DA TELA OBTIDA
PADA PELO CURSOP, QUANDO O PA-
SULTADO EM TEMAS ESPECIFICADOS.
SUBROTINAS TUCLOS2
0001 1=POLICIF INTEGER(A=0, S=2)
0002 2=C2CAL(1,474),RLE(512,43)
1 1=ACLS(512),TAB1(A,4,84),TAB2(10,10,10),L=AC(9),C2CL(10),
2 TAB(12)
0003 DOUBLE PRECISION AD=FC(10),NOMA
0004 C3=023AC(20)
1 NOM1,4=NCL,NUM84,
2 NOM2,DETEM(10),PMPICL(10),MECL(64),PMLCCL(64),
3 INASCL(10),NAUICL(10),ANJSTR(10),
4 NMA2CL,NAXAM,NSCALA,PAT
0005 DENYS(10),RMC(9),LPC(8),CURSOR(5),PICL(4),
1 PRAPRI(9)
0006 DATA 7CMA/0,1,2,4,8,16,32,64,128/
INICIALIZACAO
0007 CALL ASSIGN(,'T1:')
0008 C2CL(10)
0009 DO 70 1=1,NUNCL=1
0010 RMC(10)
0011 CONTINUE
0012 CALL OUTPUT(27,12)
0013 #RITC(0,75)
0014 75 FORMATE(10, 'FORMCLASSIFICACAO****', //)
0015 CALL OUTPUT(7)
0016 IF(NUNCL.GT.0) GO TO 85
0017 #RITC(0,40)
0018 90 FORMATE(10, 'ERRO: >>> NAO HA CLASSES')
0019 GO TO 99
0020 85 #RITC(0,90)
0021 90 FORMATE(10, 'CLASSE:', //)
0022 DO 95 1=1,NUNCL
0023 #RITC(0,100),RMC(10)
0024 CONTINUE
0025 95 FORMATE(10,12,*,*,AB)
LEITURA DAS CLASSES E DOS TEMAS CORRESPONDENTES
0026 105 #RITC(0,110)
0027 110 FORMATE(10,10, 'ENTRE COM AS CLASSES E OS TEMAS ASSOCIADOS',
1 //,10, 'EM ORDEM DECRESCENTE DE PRIORIDADE', //)
0028 111
0029 115 #RITC(0,120)
0030 120 FORMATE(10,CLASSE(A),TEMATE(A),NOM NOME(A) //)
0031 CALL OUTPUT(7)
0032 #RMC(0,151)
0033 CALL FORMATE(0,74)
0034 IF (L=1,CJ,*10) GO TO 989
0035 IF (L=1,CJ,*110) GO TO 148
0036 148
0037 CALL 1=FFFC(0,74,CJ)
0038 IF (CJ.GE.1,AND,CJ.LE.NUNCL) GO TO 130
0039 #RITC(0,120)
0040 125 FORMATE(10, 'ERRO: >>> CLASSE INVALIDA')
0041 GO TO 115
0042 130 CALL 1=FFFC(0,74,10)
0043 IF (10.GE.0,AND,10.LE.0) GO TO 145
0044 #RITC(0,140)
0045 140 FORMATE(10, 'ERRO: >>> TEMA INVALIDO')
0046 GO TO 115
0047 145 C2C=(C2C+1)=TE=ALTE(1)
0048 TE=C2C=C2C
0049 PRAPRI(10)=NCL=A(1)NCL
0050 #RMC(0)
0051 IF (C2C=NUNCL) GO TO 148
0052 GO TO 115
0053 146 #C2C=NUNCL
0054 2=FC(10)=NCL=C2C
0055 DO 147 1=1,NUNCL
0056 PRAPRI(10)=PRAPRI(10)+FC(10)
0057 CONTINUE
LEITURA DO NOME DO ARQUIVO ONDE ESTAO AS TABELAS
0058 148 #RITC(0,150)
0059 150 1=1=ATE(10, 'TAB.TEM.C/D NOME DO ARQUIVO //')
0060 CALL OUTPUT(7)
0061 #RMC(0,151)
0062 FORMATE(10,10)
0063 CALL FORMATE(0,74)
0064 IF (L=1,CJ,*10) GO TO 999
0065 DO 152 1=1,10
0066 TAB(10)=740
0067 152 CONTINUE
0068 ***
0069 DO 153 1=1,74
0070 IF (L=1,CJ,*10) GO TO 153
0071 #RMC(0)
0072 IF (C2C.GT.10) GO TO 148
0073 TAB(10)=L(1)
0074 153 CONTINUE
LEITURA DAS TABELAS DO DISCO
```

```
0075      N=4
0076      CALL FONGE(2)
0077      CALL ASSIGNE(,FAMA,AB)
0078      READ(1,1) (I,1,1), (I,1,1), (I,1,1), (I,1,1), (I,1,1), (I,1,1), (I,1,1), (I,1,1)
0079      L=0
0080      CALL CLASSE(2)

      DOJA A TELA DM #A0?
0081      160      WRITE(6,165)
0082      165      FORMATE(, 'Tela A TELAS(1/4) 2')
0083      CALL SAIR(7)
0084      READ(5,151)
0085      CALL FONGE(,7)
0086      IF(=1),E3, *12) GO TO 175
0087      IF(=1),E2, *11) GO TO 165
0088      IF(=1),E2, *10) GO TO 999
0089      GO TO 160

      DOJA A TELA
0090      175      L=1
0091      IF(=1)
0092      T=1
0093      T=512
0094      GO TO 225

      DO METANGULO
0095      180      WRITE(6,185)
0096      185      FORMATE(, 'OPÇÕES D CURSOR 2')
0097      READ(6,151)
0098      CALL IN(CURSOS)
0099      IF(CURSOS(1),E3,1) GO TO 195
0100      WRITE(6,190)
0101      190      FORMATE(11, 'EMP >>> CURSOR EM 1000 ERRO')
0102      GO TO 160

      CALCULO DAS COORDENADAS
0103      195      X=CURSOS(2)+2*CURSOS(1)-2
0104      Y=CURSOS(2)+2*CURSOS(1)+1
0105      X=CURSOS(4)+2*CURSOS(5)
0106      Y=CURSOS(4)+2*CURSOS(5)+1

      DOJA A TELA
0107      225      L=1
0108      CLASSIFICADA
0109      DELTA=X-Y
0110      CALL T=C(MURA)
0111      WRITE(6,439)MURA

0112      475      FORMATE(, 'INICIO ',AB)
0113      DO 410 LINHA=1,17

0114      CALL IN(1,LINHA,MUR(1,1))
0115      CALL =A1
0116      CALL IN(2,LINHA,MUR(1,2))
0117      CALL =A1
0118      CALL IN(3,LINHA,MUR(1,3))
0119      CALL =A1
0120      CALL IN(4,LINHA,MUR(1,4))
0121      CALL =A1

0122      CALL T=C(LINHA,MUR(1,1),MUR(1,2),MUR(1,3),MUR(1,4),TABLIN)
0123      PC(1)=PC(1)+PTC(1)
0124      UJ 33 181,NCULS
0125      J=PC(1)+PTC(1)
0126      PC(1)=PC(1)+PTC(1)
0127      CONTINUE

      CONTINUA A ESCREVER LINHA CLASSIFICADA
0128      DO 170 LINHA=1,17
0129      J=PC(1)+PTC(1)
0130      INACLS(1)=ACSL(J)
0131      CONTINUE

0132      CALL IN(5,LINHA,INACLS)
0133      CALL =A1

0134      CONTINUE

0135      CALL T=C(MURA)
0136      CALL SAIR(27,17)
0137      WRITE(6,411)MURA
0138      FORMATE(, 'FIM ',AB)
0139      CONTINUE

0140      PC(1)=PC(1)
0141      DO 415 LINHA=1,17
0142      J=PC(1)+PTC(1)
0143      INACLS(1)=ACSL(J)
0144      CONTINUE

0145      475      FORMATE(230, '*****SULFADO DA CLASSIFICADA*****')

0146      DO 455 LINHA=1,17
0147      J=PC(1)+PTC(1)
```



```

)
) VAMOS ELIMINAR AS AREAS SEM PONTOS
)
TESTE1:
MOV      R4,R0
MOV      P1,R1
ADD      R3,R1
NCOOR1:
TSTB    BR1
BNE     TESTE2
ADD     R5,R1
SOB    R0,NCOOR1
)
INC     R3
DEC     R4
BR     TESTE1
)
TESTE2:
MOV      R4,R0
MOV      P1,R1
ADD      R3,R1
ADD      R4,R1
NCOOR2:
TSTB    BR1
BNE     LINTEM
ADD     R5,R1
SOB    R0,NCOOR2
)
DEC     R4
BR     TESTE2
)
) LIMPAR TEMAS FORA DA AREA A SER CLASSIFICADA
)
)
LINTEM:
MOV      R3,R0
BEQ     PULA
MOV      IMACLS,R1
VOLTA1:
CLRB    BR1
INC     R1
SOB    R0,VOLTA1
)
PULA:
MOV      R5,R1
MOV      R4,R0
ADD      R3,R0
CMP     R1,R0
BEQ     INICID
SUB     R2,R1
MOV      IMACLS,R0
DEC     R5
ADD     R5,R0
VOLTA2:
CLRB    BR0
DEC     R0
SOB    R1,VOLTA2
)
)
) INICIALIZACAO
) INICIO:
MOV      R4,R1
DEC     R4
ADD     R4,R3
)
ADD     R3,P1
ADD     R3,P2
ADD     R3,P3
ADD     R3,P4
ADD     R3,IMACLS
)
SUB     R2,PRAPRI
DEC     R2
)
NOVTO:
CLR     AVISO
MOV     NUMCL,R0
MOV     R0,R3
ASL     R3
ADD     PRAPRI,R3
)
NOVCL:
CMP     BR3,ULTCLA
BEQ     CONT2
)
ULTCLA:
)
) CANAL 1
)
MOV     EP1,R5
BIC     $177400,R5
ASR     R5
ASR     R5
MUL     $32,R5
ADD     BR3,R5
ADD     R2,R5
MOV     BR2,R4
BEQ     CONT1
)
) CANAL 2
)
MOV     EP2,R5
BIC     $177400,R5
ASR     R5
ASR     R5
MUL     $32,R5
ADD     $0,R5
ADD     BR3,R5
ADD     R2,R5
MOV     BR5,R5
BEQ     CONT1
MUL     $10,R5
ADD     R5,R4

```



```
C ARMAZENAMENTO
C
C NOME: TLUPD
C
C DESCRICAO: ARMAZENA EM FITA DE PAPEL OS PARAMETROS ACQUI-
C BOS PARA POSTERIOR CONTINUACAO DO TREINAMENTO.
C O HISTOGRAMA E DE LIMITES SAO GUARDADOS NO DISCO.
C CONTINUACAO NE USADA EM TLUPD.
C
C SUBROTINAS: PPSUB (PPATT)
C
0001      IMPLICIT INTEGER (A-D,S-Z)
0002      LOGICAL I, TARN(10),U(74)
0003      DOUBLE PRECISION NOHECL(16)
0004      COMMON/COMMON/
1 NDIM,NUMCL,NUMMH,
2 NOMECL,PE,TERM(16),RNPTEL(16),MEDCL(64),RMECL(64),
3 TRNSCL(16),FNTCL(16),ANOSTR(190),
4 MAXCL,MAXAN,RSCALE,FAT
0005      DIMENSION CONTAD(0,4,64),LIM(8,4),LINS(8,4)
C
0006      CALL OUTPUT(27,12)
0007      CALL ASSIGN(6,'TI:')
0008      WRITE(6,100)
0009      100  FORMAT(71X,'***ARMAZENAMENTO***')
0010      CALL OUTPUT(7)
C
C LER DOS ARQUIVOS COMUNS
C
0011      CALL FDBSET(2)
0012      CALL ASSIGN(2,'LIMIT.TAB')
0013      READ(2)((LIM(I,J),J=1,32),(LINS(J,I),J=1,32))
0014      ENDFILE 2
0015      CALL CLOSE(2)
C
0016      CALL FDBSET(3)
0017      CALL ASSIGN(3,'HISTO.TAB')
0018      READ(3)((CONTAD(I,K,I),I=1,2048))
0019      ENDFILE 3
0020      CALL CLOSE(3)
C
0021      IF (NUMCL.GT.0)GO TO 210
0022      WRITE(6,200)
0023      200  FORMAT(1X,'ERRO >>>NAO HA CLASSES')
0024      GO TO 999
C
C ENTRADA DO NOME DO ARQUIVO
C PARA LIMITES E HISTOGRAMA
C
0025      210  CALL OUTPUT(7)
0026      CALL OUTPUT(27,12)
0027      WRITE(6,220)
0028      220  FORMAT(7,1X,'ENTRE C/D NOME DO ARQUIVO',/,
1 'DOS LIMITES E DO HISTOGRAMA >')
0029      READ(6,230)M
C
0030      230  FORMAT(74A1)
0031      CALL FRONT(M,74)
0032      IF(M(1).EQ.'40' GO TO 999
0033      GO 240 I=1,10
0034      TARN(I)=M(1)
0035      240  CONTINUE
0036      K=0
0037      GO 250 I=1,74
0038      IF(M(I).EQ.'40'GO TO 250
0039      K=K+1
0040      IF(K.GT.10)GO TO 10
0041      TARN(K)=M(I)
0042      CONTINUE
C
C ESCREVER LIMITES E HISTOGRAMA NO DISCO
C
0043      MM=K
0044      CALL FDBSET(4)
0045      CALL ASSIGN(4,TARN,MM)
0046      WRITE(4)((LIM(I,J),J=1,32),(LINS(J,I),J=1,32),
1 (CONTAD(K,I,I),K=1,2048))
0047      ENDFILE 4
0048      CALL CLOSE(4)
0049      CALL PPATT
0050      CALL PPSUB(NDIM,6)
0051      NPAITS=16*NUMCL
0052      CALL PPSUB(NDHECL,NPAITS)
0053      NPAITS=2*NUMCL
0054      CALL PPSUB(NDTERM,NPAITS)
0055      NPAITS=4*NUMCL
0056      CALL PPSUB(RNPTEL,NPAITS)
0057      NPAITS=8*NUMCL
0058      CALL PPSUB(RMECL,NPAITS)
0059      NPAITS=16*NUMCL
0060      CALL PPSUB(FNTCL,NPAITS)
0061      NPAITS=32*NUMCL
0062      CALL PPSUB(ANOSTR,NPAITS)
0063      NPAITS=40*NUMCL
0064      CALL PPSUB(NTCL,NPAITS)
0065      NPAITS=104*NUMMH
0066      CALL PPSUB(ANOSTR,NPAITS)
C
0067      WRITE(6,900)
0068      900  FORMAT(77,1X,'***ARMAZENAMENTO TERMINADO***')
C
0069      999  CALL EXIT
0070      END
```

APENDICE B

Resultado da classificação da região de Manaus (1976) feito pelo MAXV10.

CLASSIFICACAO

CLASSES:

1. C1
2. C2
3. C3
4. C4
5. C5

ENTRE COM OS TEMAS ASSOCIADOS AS CLASSES

CLASSE > 1 TEMA > 1
CLASSE > 2 TEMA > 2
CLASSE > 3 TEMA > 3
CLASSE > 4 TEMA > 4
CLASSE > 5 TEMA > 5

ENTRE COM O LIMAR >4.5

TODA A TELA*(S/N) >S

RESULTADO DA CLASSIFICACAO

CLASSE:	1 C1	(TEMA 1):AREA	38247.(14.59%)
CLASSE:	2 C2	(TEMA 2):AREA	37751.(14.40%)
CLASSE:	3 C3	(TEMA 3):AREA	20115.(10.73%)
CLASSE:	4 C4	(TEMA 4):AREA	135926.(51.85%)
CLASSE:	5 C5	(TEMA 5):AREA	13034.(5.28%)

NAO CLASSIFICADOS-AREA	=	8261.(3.15%)
NO. TOTAL DE FONTOS	=	262144.
LIMAR	=	4.50

CLASSIFICACAO TERMINADA


```

CALL I=1,CLINA,MUF1)
CALL MUF
CALL I=2,CLINA,MUF2)
CALL MUF
CALL I=3,CLINA,MUF3)
CALL MUF
CALL I=4,CLINA,MUF4)
CALL MUF

CALL CLSPAP(I,MH54,MUCL,SI,DELTA,
1 IN=CL,PAUCL,OLICR,PICL,IMACL,
2 DIF1,IMUF,MUF,IMUF,IMUL)

3) 100 1000,00
4) 100 1000,00
5) 100 1000,00
6) 100 1000,00
7) 100 1000,00
8) 100 1000,00
9) 100 1000,00
10) 100 1000,00
11) 100 1000,00
12) 100 1000,00
13) 100 1000,00
14) 100 1000,00
15) 100 1000,00
16) 100 1000,00
17) 100 1000,00
18) 100 1000,00
19) 100 1000,00
20) 100 1000,00
21) 100 1000,00
22) 100 1000,00
23) 100 1000,00
24) 100 1000,00
25) 100 1000,00
26) 100 1000,00
27) 100 1000,00
28) 100 1000,00
29) 100 1000,00
30) 100 1000,00
31) 100 1000,00
32) 100 1000,00
33) 100 1000,00
34) 100 1000,00
35) 100 1000,00
36) 100 1000,00
37) 100 1000,00
38) 100 1000,00
39) 100 1000,00
40) 100 1000,00
41) 100 1000,00
42) 100 1000,00
43) 100 1000,00
44) 100 1000,00
45) 100 1000,00
46) 100 1000,00
47) 100 1000,00
48) 100 1000,00
49) 100 1000,00
50) 100 1000,00
51) 100 1000,00
52) 100 1000,00
53) 100 1000,00
54) 100 1000,00
55) 100 1000,00
56) 100 1000,00
57) 100 1000,00
58) 100 1000,00
59) 100 1000,00
60) 100 1000,00
61) 100 1000,00
62) 100 1000,00
63) 100 1000,00
64) 100 1000,00
65) 100 1000,00
66) 100 1000,00
67) 100 1000,00
68) 100 1000,00
69) 100 1000,00
70) 100 1000,00
71) 100 1000,00
72) 100 1000,00
73) 100 1000,00
74) 100 1000,00
75) 100 1000,00
76) 100 1000,00
77) 100 1000,00
78) 100 1000,00
79) 100 1000,00
80) 100 1000,00
81) 100 1000,00
82) 100 1000,00
83) 100 1000,00
84) 100 1000,00
85) 100 1000,00
86) 100 1000,00
87) 100 1000,00
88) 100 1000,00
89) 100 1000,00
90) 100 1000,00
91) 100 1000,00
92) 100 1000,00
93) 100 1000,00
94) 100 1000,00
95) 100 1000,00
96) 100 1000,00
97) 100 1000,00
98) 100 1000,00
99) 100 1000,00
100) 100 1000,00

```

Listação do programa C.R.A.I

```

CALL CLSPAP(I,MH54,MUCL,SI,DELTA,
1 IN=CL,PAUCL,OLICR,PICL,IMACL,
2 DIF1,IMUF,MUF,IMUF,IMUL)

DESCRICAO
DADA UMA LINHA DA TABELA, CONSTITUIDA DE 4 CANAIS
(MH54,MUF1,MUF2,MUF3), CLSPAP FORNECE UMA LINHA DE 100 POSICOES
COM O RESULTADO DA CLASSIFICACAO. O TOTAL DE PONTOS DA
CLASS. 1 EM ANEXO E ENCONTRADO EM PICL(1), FICANDO A POSICAO
PICL PARA O TOTAL DE PONTOS DAS CLASSIFICACOES DA LINHA

```

TITLE CLSPAP

```

MH54 0
MUCL 0
SI 0
DELTA 0
TRN2CL 0
MUCL 0
MH54 0
PICL 0
IMACL 0
PT01 0
PT02 0
PT03 0
PT04 0
DIF1 0
DIF2 0
DIF3 0
DIF4 0
TRN1 0
TRN2 0
TRN3 0
TRN4 0
TRN5 0
TRN6 0
TRN7 0
TRN8 0
TRN9 0
TRN10 0
TRN11 0
TRN12 0
TRN13 0
TRN14 0
TRN15 0
TRN16 0
TRN17 0
TRN18 0
TRN19 0
TRN20 0
TRN21 0
TRN22 0
TRN23 0
TRN24 0
TRN25 0
TRN26 0
TRN27 0
TRN28 0
TRN29 0
TRN30 0
TRN31 0
TRN32 0
TRN33 0
TRN34 0
TRN35 0
TRN36 0
TRN37 0
TRN38 0
TRN39 0
TRN40 0
TRN41 0
TRN42 0
TRN43 0
TRN44 0
TRN45 0
TRN46 0
TRN47 0
TRN48 0
TRN49 0
TRN50 0
TRN51 0
TRN52 0
TRN53 0
TRN54 0
TRN55 0
TRN56 0
TRN57 0
TRN58 0
TRN59 0
TRN60 0
TRN61 0
TRN62 0
TRN63 0
TRN64 0
TRN65 0
TRN66 0
TRN67 0
TRN68 0
TRN69 0
TRN70 0
TRN71 0
TRN72 0
TRN73 0
TRN74 0
TRN75 0
TRN76 0
TRN77 0
TRN78 0
TRN79 0
TRN80 0
TRN81 0
TRN82 0
TRN83 0
TRN84 0
TRN85 0
TRN86 0
TRN87 0
TRN88 0
TRN89 0
TRN90 0
TRN91 0
TRN92 0
TRN93 0
TRN94 0
TRN95 0
TRN96 0
TRN97 0
TRN98 0
TRN99 0
TRN100 0

```


APÊNDICE C

PARÂMETROS ADQUIRIDOS PELO MAXVER

ENTRE COM O TEMA > 1
(C CURSOR OU (T JODO TEMA ? > C
POSICIONE O CURSOR SOBRE A TR. AREA >
ENTRE COM O NOME DA CLASSE >C1

PARAMETROS DA CLASSE 1 C1

NO. DE PONTOS = 676.
MEDIA = 15.15 10.95 3.41 1.15
MATRIZ DE COVARIANCIA:
2.00 0.12 1.27 0.89
0.12 1.85 1.00 0.60
1.27 1.00 10.21 7.01
0.89 0.80 7.01 4.19

ADQUISICAO COMPLETADA

ENTRE COM O TEMA > 2
(C CURSOR OU (T JODO TEMA ? > C
POSICIONE O CURSOR SOBRE A TR. AREA >
ENTRE COM O NOME DA CLASSE >C2

PARAMETROS DA CLASSE 2 C2

NO. DE PONTOS = 676.
MEDIA = 19.39 15.64 52.83 49.55
MATRIZ DE COVARIANCIA:
2.83 0.43 3.39 2.91
0.43 2.40 2.48 2.90
3.39 2.48 22.72 21.49
2.91 2.90 21.49 30.63

ADQUISICAO COMPLETADA

ENTRE COM O TEMA > 3
(C CURSOR OU (T JODO TEMA ? > C
POSICIONE O CURSOR SOBRE A TR. AREA >
ENTRE COM O NOME DA CLASSE >C3

PARAMETROS DA CLASSE 3 C3

NO. DE PONTOS = 364.
MEDIA = 33.64 37.96 17.05 1.29
MATRIZ DE COVARIANCIA:
9.12 4.15 1.43 -0.51
4.15 4.84 2.21 -2.34
1.43 2.21 14.08 11
-0.51 -2.34 11.71 13

ADQUISICAO COMPLETADA
ENTRE COM O TEMA > 4
(C) CURSOR OU (T) TODO TEMA > C
POSICIONE O CURSOR SOBRE A TK. AREA >
ENTRE COM O NUME DA CLASSE > C4

PARAMETROS DA CLASSE 4 C4

NO. DE PONTOS = 344.
MEDIA = 44.02 47.91 47.45 52.58
MATRIZ DE COVARIANCIA:
312.32 306.08 143.45 54.60
306.08 470.20 261.74 93.61
143.45 261.74 252.24 147.09
54.60 93.61 147.09 140.67

ADQUISICAO COMPLETADA
ENTRE COM O TEMA > 5
(C) CURSOR OU (T) TODO TEMA > C
POSICIONE O CURSOR SOBRE A TK. AREA >
ENTRE COM O NUME DA CLASSE > C5

PARAMETROS DA CLASSE 5 C5

NO. DE PONTOS = 108.
MEDIA = 21.79 17.64 47.19 32.82
MATRIZ DE COVARIANCIA:
5.20 3.27 19.52 19.54
3.27 5.36 18.00 17.48
19.52 18.00 191.03 203.22
19.54 17.48 203.22 234.42

ADQUISICAO COMPLETADA

1. COM APRESENTA INSUFICIENTE
2. COM APRESENTA INSUFICIENTE
3. COM APRESENTA INSUFICIENTE

ENTRE COM O NUME DA CLASSE > 4

17.82 14.80 10.07
1.03 2.73
0.18 0.27
COM APRESENTA INSUFICIENTE

ADQUISICAO COMPLETADA

Classificaco obtida pelo LIMITE

CLASSIFICACAO

CLASSES:

- 1. C1
- 2. C2
- 3. C3
- 4. C4
- 5. C5

ENTRE COM OS TEMAS ASSOCIADOS AS CLASSES

CLASSE > 1 TEMA > 1
CLASSE > 2 TEMA > 2
CLASSE > 3 TEMA > 3
CLASSE > 4 TEMA > 4
CLASSE > 5 TEMA > 5

ENTRE COM O LIMITE > 5.0

TODA A TELA (S/N) > S

RESULTADO DA CLASSIFICACAO

CLASSE: 1 C1 (TEMA 1)AREA 26749.1 (10.21%)
CLASSE: 2 C2 (TEMA 2)AREA 134401.4 (51.30%)
CLASSE: 3 C3 (TEMA 3)AREA 4606.4 (2.22%)
CLASSE: 4 C4 (TEMA 4)AREA 14595.1 (5.57%)
CLASSE: 5 C5 (TEMA 5)AREA 49537.1 (26.03%)

NAD CLASSIFICADOS-AREA = 10157.4 (3.87%)
NO. TOTAL DE PONTOS = 262144.
LIMITE = 5.00

CLASSIFICACAO TERMINADA

RUN SELPTO
 ORDENAÇÃO - 2000 PONTOS MAIS POPULOSOS
 ENTRE COM LIMITE INFERIOR DE POPULACAO >1
 336 VETORES **ARQUIVO PRONTO**

RUN ISDCLT
 QUANTAS ITERACOES DE PARTICAO-COMBINACAO**
 MAX=16/14
 FORNECA A SEQUENCIA**
 PARTICAO=1 COMBINACAO=0
 1 ESIMA . ITERACAO=1
 2 ESIMA . ITERACAO=1
 3 ESIMA . ITERACAO=1
 4 ESIMA . ITERACAO=0
 5 ESIMA . ITERACAO=1
 6 ESIMA . ITERACAO=0
 7 ESIMA . ITERACAO=1
 8 ESIMA . ITERACAO=0
 9 ESIMA . ITERACAO=1
 10 ESIMA . ITERACAO=0
 11 ESIMA . ITERACAO=1
 12 ESIMA . ITERACAO=0
 13 ESIMA . ITERACAO=0
 14 ESIMA . ITERACAO=0
 NUMERO MINIMO DE VET. P/ SER CLASSE>5
 NUMERO DE CANAIS CONSIDERADOS E>4
 MEDIA DAS CLASSES
 1 49.83 41.00 47.25 48.75
 2 15.38 10.95 3.41 0.00
 3 21.04 17.20 61.02 59.02
 4 33.14 37.82 14.80 0.07
 5 19.71 15.95 51.74 47.49
 DESVIO PADRAO
 1 12.49 16.43 11.08 8.29
 2 2.20 2.04 1.99 0.00
 3 2.04 2.78 5.49 5.24
 4 2.84 3.03 2.24 0.33
 5 2.00 2.48 4.02 4.11

RUN ISDCLT
 QUANTAS ITERACOES DE PARTICAO-COMBINACAO**
 MAX=16/14
 FORNECA A SEQUENCIA**
 PARTICAO=1 COMBINACAO=0
 1 ESIMA . ITERACAO=1
 2 ESIMA . ITERACAO=1
 3 ESIMA . ITERACAO=1
 4 ESIMA . ITERACAO=0
 5 ESIMA . ITERACAO=1
 6 ESIMA . ITERACAO=0
 7 ESIMA . ITERACAO=1
 8 ESIMA . ITERACAO=0
 9 ESIMA . ITERACAO=1
 10 ESIMA . ITERACAO=0
 11 ESIMA . ITERACAO=1
 12 ESIMA . ITERACAO=0
 13 ESIMA . ITERACAO=0
 14 ESIMA . ITERACAO=0
 NUMERO MINIMO DE VET. P/ SER CLASSE>5
 NUMERO DE CANAIS CONSIDERADOS E>4
 MEDIA DAS CLASSES
 1 45.10 34.30 81.80 78.24
 2 73.82 74.67 87.30 63.48
 3 45.43 35.54 23.40 4.15
 4 52.78 47.24 23.80 39.12
 5 72.74 78.30 13.25 13.89
 6 42.31 58.48 39.23 9.55
 7 73.40 73.40 99.17 79.07
 8 82.40 86.55 99.15 73.85
 9 72.33 73.65 74.40 49.40
 DESVIO PADRAO
 1 3.08 2.59 4.31 4.40
 2 4.43 4.30 3.76 4.45
 3 3.38 2.43 3.08 2.28
 4 2.10 3.67 4.12 9.20
 5 3.93 4.78 4.61 3.27
 6 4.44 4.20 4.08 3.44
 7 3.59 5.40 4.70 6.71
 8 4.49 5.89 4.89 4.57
 9 4.28 5.81 4.37 4.

ANÁLISE DE VARIÁVEIS DE FATORIAL DISCRIMINANTE

PARÂMETROS DE FATORIAL DISCRIMINANTE

1	27.0000	1
2	27.0000	1
3	27.0000	1
4	27.0000	1
5	27.0000	1
6	27.0000	1
7	27.0000	1
8	27.0000	1
9	27.0000	1
10	27.0000	1
11	27.0000	1
12	27.0000	1
13	27.0000	1
14	27.0000	1

STATISTICS FOR CLASS 1

1	3.00	4.00	0.00	0.00
2	44.17	34.7	14.00	27.00
3	50.42	37.3	14.00	4.12
4	27.34	45.54	24.92	40.61
5	27.34	29.43	20.09	27.44
6	27.34	27.00	20.09	27.44
7	47.19	47.31	5.00	20.00
8	13.49	20.0	4.00	20.14
9	62.50	20.0	3.00	9.00
10	27.34	20.0	24.00	13.00
11	27.40	20.0	10.10	22.00
12	20.14	20.0	10.10	22.00
13	20.14	47.17	10.10	22.00

STATISTICS FOR CLASS 2

1	4.00	1.00	5.00	2.00
2	2.00	2.00	1.00	2.00
3	3.00	2.00	3.00	2.00
4	2.00	1.00	4.00	2.00
5	4.00	2.00	3.00	4.00
6	4.00	4.00	3.00	4.00
7	4.00	3.00	4.00	2.00
8	2.00	3.00	3.00	4.00
9	4.00	4.00	3.00	2.00
10	3.00	4.00	4.00	3.00
11	3.00	2.00	3.00	4.00
12	4.00	4.00	4.00	2.00
13	3.00	4.00	1.00	2.00

RUN CLINIX
 CLASSE 1 COM AMOSTRA INDEPENDENTE
 CLASSE 2 COM AMOSTRA INDEPENDENTE

PARÂMETROS DA CLASSE 1

NO. DE PONTOS	394			
MEDIA	45.42	35.00	23.34	4.12
MATRIZ DE COVARIÂNCIA:				
	11.42	2.00	0.00	1.00
	0.00	0.00	0.00	0.00
CLASSE 4 COM AMOSTRA INDEPENDENTE				
CLASSE 5 COM AMOSTRA INDEPENDENTE				
CLASSE 6 COM AMOSTRA INDEPENDENTE				
CLASSE 7 COM AMOSTRA INDEPENDENTE				
CLASSE 8 COM AMOSTRA INDEPENDENTE				

PARÂMETROS DA CLASSE 2

NO. DE PONTOS	217			
MEDIA	6.00	29.00	13.92	9.03
MATRIZ DE COVARIÂNCIA:				
	30.00	20.10	21.00	15.00
	20.10	10.00	10.00	10.00
CLASSE 10 COM AMOSTRA INDEPENDENTE				
CLASSE 11 COM AMOSTRA INDEPENDENTE				
CLASSE 12 COM AMOSTRA INDEPENDENTE				
CLASSE 13 COM AMOSTRA INDEPENDENTE				

STATISTICAL COMPLETE


```
      LAMP
      M0V=1+MDES/P(4,1)
142  C0=FINDE
      IF(DES0P(LK,1),LT,MSC441CGU TO 150
      *M0V*P(1,M0V)
      IF(LAMP(1),LT,M0V1CGU TO 150
      M0L0(LK,1)+M0L0(LK,1)+MDES/P(LK,1)
      MDES/P(LK,1)+MDES/P(LK,1)+MDES/P(LK,1)
      NCLST=NCLST+1
      M0L0(LK,NCLST)+M0L0(LK,1)+MDES/P(LK,1)
      MDES/P(LK,NCLST)+MDES/P(LK,1)
      M0 170 M01,M004
      IF(LK,1),M01CGU TO 170
      M0M0(LK,NCLST)+M0M0(LK,1)
      MDES/P(LK,NCLST)+MDES/P(LK,1)
170  C0=FINDE
182  I0=1
      IF(LK,NCLST)GO TO 112
      G0 13 547
      C0=M0V1CGU
192  M0 110 101,(NCLST-1)
      M0 120 101,1,NCLST
      M0M0(LK,1)
      M0 130 101,M004
      M0J1(LK,M0(LK,1)+M0(LK,1))
      M0J2(MDES/P(LK,1)+MDES/P(LK,1))
      IF(M0J2,LT,1,)+M0J2(LK)
      IF(M0J2,GT,100)+M0J2(LK)
192  M0M0(LK,M0J2)+M0J2(LK)+M0J2(LK)
200  M0(LK,1)+M0(LK,1)+M0(LK,1)
210  IF(LK,1)
220  J0=1
230  IF(LK(1,1),GE,M0L0(M01CGU) TO 160
      IF(J0M0(LK,1))
      M0 130 101,M004
      M0J1(LK,M0(LK,1)+M0(LK,1))
      M0J2(LK,M0(LK,1)+M0(LK,1))
235  M0M0(LK,1)+M0M0(LK,1)+M0M0(LK,1)
      M0 11 170
240  J0=1
      IF(LK,NCLST)GO TO 140
270  I0=1
      IF(LK,NCLST)GO TO 540
      IF(LK,1),IF(1)GO TO 150
      G0 13 370
280  M0 110 101,NCLST
      M0 120 101,M004
      M0J1(LK,1)+M0(LK,1)
      M0M0(LK,1)+M0M0(LK,1)
      M0L0(LK,1)+M0L0(LK,1)
290  MDES/P(LK,1)+M0L0(LK,1)+M0L0(LK,1)
300  MDES/P(LK,1)+M0L0(LK,1)+M0L0(LK,1)
302  C0=FINDE
302  CALL F0M5F(2)
```

```
      CALL ABSIGN(2,'DESEY.DAT')
      DEFIL FILE 2(00,2,0,0,0,2)
      WRITE(1)10004
      WRITE(2)1+NCLST
      M0 11 170
      M0L0(LK,1)+M0L0(LK,1)
      F0M0(104,'M0M0 DAS CLASSE00')
      M0 110 101,NCLST
      M0 115 101,M004
      M0M0(LK,1)+M0M0(LK,1)
415  M0L0(LK,1)+M0L0(LK,1)
410  M0L0(LK,1)+M0L0(LK,1)+M0L0(LK,1)
420  F0M0(LK,10,5,4,0,0,0)
      CALL CLOS(2)
      M0L0(LK,1)+M0L0(LK,1)
430  F0M0(LK,1)+M0M0(LK,1)+M0M0(LK,1)
      M0 110 101,NCLST
440  M0L0(LK,1)+M0M0(LK,1)+M0M0(LK,1)
      CALL EXIT
      END
```

Lista de programa SELPTO

```
IMPLICIT INTEGER (A-Z,S=2)
LOGICAL L1 (E,F),Z1(400),Z2(400),Z3(400)
DIMENSION NPON(400),PC(10),L4(10)
DIMENSION IAU(5),APAU(5)
EQUIVALENCE (L4(1),*N)
CALL ASSIGN(4,'F1')
CALL ASSIGN(5,'M1')
WRITE(1,100)
100 FJ=AC('MDEMCAU = 2000 PONTOS MAIS POPULOSOS')
WRITE(1,101)
101 FJ=AC('S LEMO COM LIMITE INFERIOR DE POPULACAO S')
102 FJ=AC('M1')
103
CALL FONS(1,74)
CALL INIT(1,74,L1=1)
CALL FONS(2)
CALL ASSET(2,'SIG,JAT')
DEFINE FILE 2(0,0,2,2,0,0)
READ(2)I1PEZ,IPC
104
DO 420 M=1,NPC
  READ(2)M1,IX
  M1
  CNOX1
  DO 410 I=10,12,2
    DO 410 J=7,9,2
      DO 410 K=6,8,2
        DO 410 L=1,3,2
          IF(I*J)(M1,*)EQ.0 GO TO 410
          IF(L*(M1,LP,LI*1))GO TO 410
          NPON(I)=I*(M1)
          Z1(I)=L4(I)
          Z2(I)=L4(I)
          Z3(I)=L4(I)
          I=1
          M1=M1L*1
          CNOX=CNOX+1
        410
      420
    END DO
  END DO
  CALL CLOSE(2)
  M1=1
  NPON=0
  202
  DO 201 J=1,PC(J)=1
    IF(NPON(J),L1,NPON(J),)GO TO 201
    AU(1)=J*(J)
    AU(2)=J*(J)
    AU(3)=J*(J)
    AU(4)=J*(J)
    AU(5)=J*(J)
    NPON(J)=PC(J)
    Z1(J)=Z1(J)
    Z2(J)=Z2(J)
    Z3(J)=Z3(J)
  201
  CONTINUE
  IF(L1,CU,3)GO TO 205
  300
  301
  GO TO 202
  M1=1
  205
  IF(L1,CU,3)M1
  M1=M1+2
  CALL FONS(2)
  CALL ASSIGN(2,'SELPTO,JAT')
  DEFINE FILE 2(0,0,2,0,0,0)
  WRITE(2)M1
  DO 430 I=1,M1
  M1=M1+2
  WRITE(2)Z1(I),Z2(I),Z3(I),Z4(I),Z5(I)
  400
  CONTINUE
  M1=4,450
  FORMAT(10Z,10,'PONTOS',10Z,'*MQUIVO PONTOS')
  CALL CLOSE(2)
  END
```