Reconhecimento de Mapas de Isovalores Digitalizados via *Scanners*

Luiz Eduardo Seabra Varella [1,2]
Luis Paulo Vieira Braga [2,3]

[1] PETROBRÁS-Petróleo Brasileiro S.A.
Depex-Departamento de Exploração
Av. Chile 65, sala 1402
20035 Rio de Janeiro, RJ, Brasil
g073@c53000.petrobras.anrj.br

[2] IME-Instituto Militar de Engenharia
Seção de Sistemas e Computação / Cartografia
Praça General Tibúrcio, 80
22290 Rio de Janeiro, RJ, Brasil

[3] IM/UFRJ-Instituto de Matemática
Caixa Postal 68530
21945 Rio de Janeiro, RJ, Brasil
mad02005@ufrj.bitnet

Abstract: This paper describes a semi-automatic procedure oriented to the recognition of scanned isovalues maps. It includes pre-processing, thinning and vectorization of bilevel images. Following these three steps, the separation of different graphics elements and its recognition are presented. Finally some results concerning the implementation are considered.

1.0-Introdução

A disponibilidade de equipamentos matriciais de alta resolução (*Scanners*) e o crescente aumento da velocidade de processamento dos computadores, vêm estimulando nos últimos anos a investigação de procedimentos destinados ao reconhecimento de informações pictóricas. Tais procedimentos permitem que gráficos impressos em papel sejam convertidos em arquivos digitais codificados em formato vetorial. Geralmente, a conversão é feita em duas etapas. Na primeira, chamada digitalização, é gerada uma imagem digital binária, a qual representa o retrato do gráfico original. Na segunda, os elementos gráficos são extraídos, reconhecidos e codificados em formato vetorial.

A transferência de gráficos impressos para arquivos digitais, com objetivo de se alimentar Sistemas Gráficos, permite não só uma maior segurança dos dados, mas também facilita a reprodução, manipulação e análise das informações. Neste sentido, o presente trabalho propõe um procedimento automatizado que visa agilizar a aquisição de grandes volumes de dados espaciais.

O texto foi organizado de maneira a apresentar o problema da aquisição de dados espaciais, formalizar a solução, descrever os módulos do procedimento proposto e analisar os resultados alcançados com a implementação.

2.0-Análise do Problema

Em função da natureza do problema, é conveniente que se inicie a discussão do assunto classificando alguns tipos de representação de dados. Para [Pavlidis (1982)] um gráfico armazenado em computador pode ser classificado da sequinte maneira: imagens coloridas ou com vários níveis de cinza (Classe 1); imagens binárias (Classe 2); curvas e linhas contínuas (Classe 3); retas, curvas, arcos, caracteres, etc. (Classe 4).

O reconhecimento, aqui proposto, pode ser visto como um procedimento que executa sucessivas transformações entre classes, com a entrada e saída se constituindo num mesmo gráfico representado pela Classe 2 e 4, respectivamente. A Figura 1 mostra um gráfico típico, no qual destacam-se as seguintes características: desenho preto e branco com dimensões que variam do formato A4 ao A0; a maior parte do gráfico corresponde ao seu fundo (branco) e a menor, a sua frente (preto); o desenho possui um sistema de

coordenadas representado por uma malha; os elementos gráficos possíveis são retas, curvas, curvas de isovalores, símbolos e caracteres; esses elementos possuem espessuras diversas e o tipo do traçado é sempre contínuo; as curvas de isovalores não admitem interseções e algumas possuem rotulação correspondente ao seu valor; os caracteres são de diversas formas, tamanhos e inclinações; os símbolos têm tamanhos e inclinações variadas, podendo ser de diversas formas.

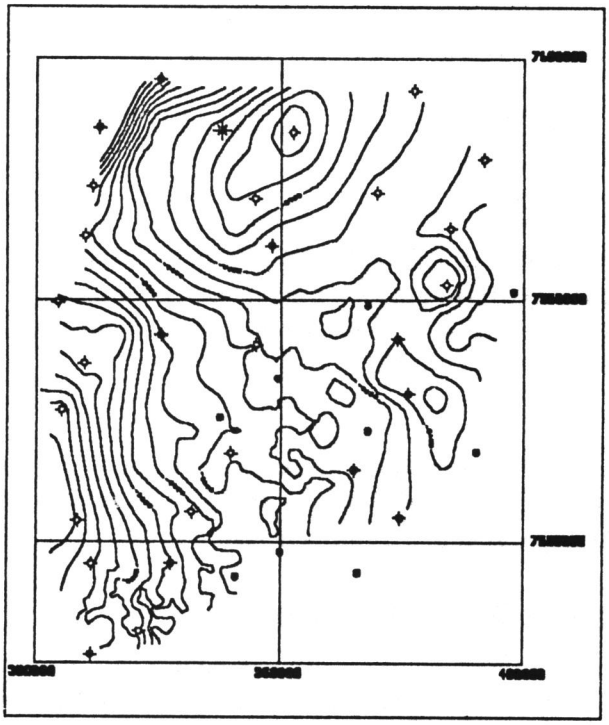

Figura 1 : Características do Gráfico.

O problema pode ser formulado como: dado um documento cartográfico, com as características citadas acima, deseja-se transferir as respectivas informações para um arquivo digital de formato vetorial, Classe 4. Este arquivo deve ocupar o menor espaço em disco possível, e em uma etapa posterior alimentar um Sistema Gráfico qualquer. A digitalização é feita através de *Scanners* , gerando-se uma imagem digital binária (Classe 2).

3.0-Estrutura Geral da Solução

O procedimento foi estruturado com seis módulos (vide Figura 2): Pré-Processamento, Afinamento, Codificação, Classificação, Reconhecimento Geométrico e Reconhecimento Não Geométrico. O Pré-Processamento consiste na eliminação dos ruídos da imagem. O Afinamento tem por objetivo gerar o esqueleto da imagem, simplificando-a para a vetorização. A Codificação inicia a vetorização. A Classificação separa os elementos gráficos em duas categorias: geométricos (retas, poligonais, curvas e curvas de isovalores) e não geométricos (caracteres e símbolos). O Reconhecimento Geométrico gera as curvas de isovalores, retas e poligonais. O Reconhecimento Não Geométrico é responsável pela codificação dos textos e símbolos.

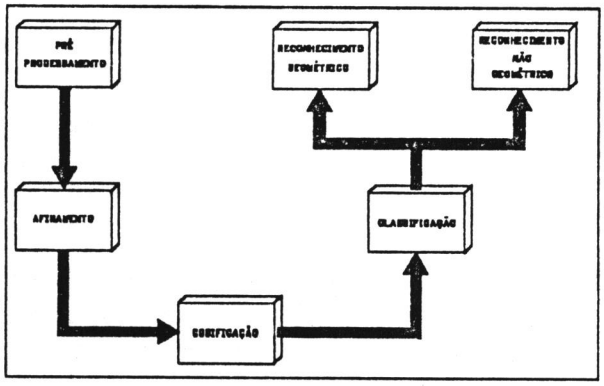

Figura 2 : Arquitetura Proposta.

4.0-Pré-Processamento

O Pré-Processamento elimina os ruídos da imagem, preparando-a para o afinamento. Com este fim, [Ramachandran (1980)] propõe dois métodos, a Remoção de Ruídos e o Preenchimento de Vazios, que exploram as características físicas das linhas do desenho: espessura mínima (0.010 in); distância entre linhas (0.012 in). A Figura 3 ilustra os métodos aplicados em uma imagem obtida por um *Scanner* cuja resolução impõe a existência de três pixels para a largura e distância mínima entre os elementos.

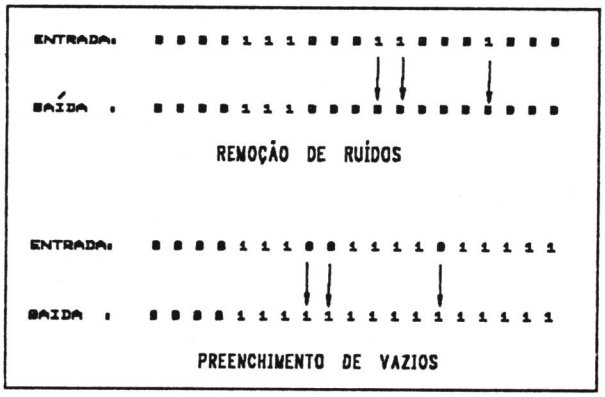

Figura 3 : Técnicas de Pré-Processamento.

[Ting (1980)] sugere três técnicas para o pré-processamento de imagens binárias: a remoção de pixels isolados, a reconstrução de linhas fragmentadas e a suavização da imagem.

A primeira, a remoção de pixels isolados, é feita utilizando-se a função OU EXCLUSIVO na vizinhança-8 para identificação dos padrões isolados.

A segunda, a reconstrução de linhas fragmentadas, também conhecida como suavização direcional, procura corrigir as falhas dos contornos identificando as configurações da Figura 4 e substituindo o pixel central por um valor preto, de forma a preservar a conectividade dos elementos. O valor D indica que o pixel pode estar aceso ou apagado.

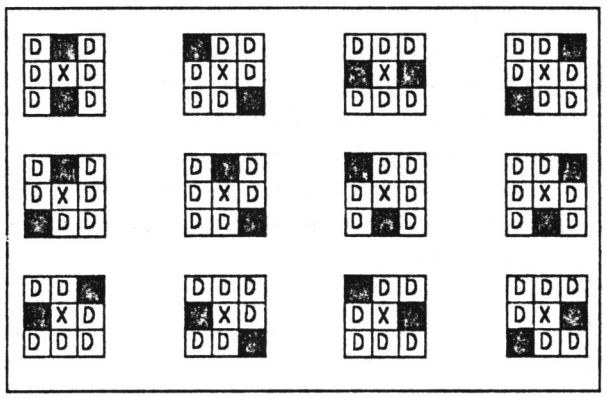

Figura 4 : Suavização Direcional.

A terceira, também chamada de suavização pela maioria com teste de contorno, o nível de cinza do pixel a ser processado é calculado em função do valor da maioria de seus oito vizinhos. O pixel é aceso se mais da metade de seus vizinhos são pretos, do contrário ele é apagado. O efeito produzido é de suavização dos contornos, podendo ocorrer quebra de conectividade das linhas com pequena espessura. A fim de se evitar este problema, deve-se incorporar um teste de modo a preservar os pixels de contorno. Para a vizinhança-8, as configurações que definem um pixel de contorno são mostradas na Figura 5. Desta forma, antes de se processar um pixel, sua vizinhança deve ser testada. Se esta coincidir com um dos padrões da Figura 5 o valor do pixel é preservado, caso contrário, o nível de cinza assumirá o valor da maioria.

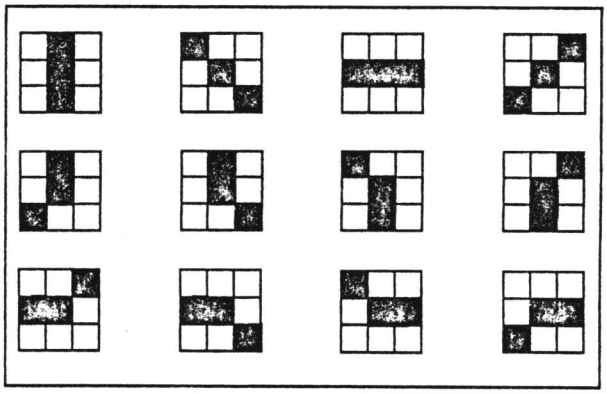

Figura 5 : Padrões de Contorno.

5.0-Afinamento

Existem, na literatura, diversos algoritmos de afinamento, não havendo um consenso em torno de um único. No entanto, sabe-se que um bom algoritmo deve preservar a topologia, a forma e a conectividade dos elementos do desenho, e na medida do possível, ser resistente aos ruídos de contorno. Com este intuito, foi adotado o algoritmo clássico, que apesar de ser lento, leva a esqueletos com elevado grau de acuracidade.

O método clássico, assim como outros, baseia-se na busca aos pixels de contorno, para em seguida verificar se estes fazem parte ou não do esqueleto da imagem. Os pixels ditos não esqueletais são eliminados e os esqueletais preservados. O processo é repetido até que toda imagem seja formada apenas de pixels esqueletais. Um pixel é dito de contorno se ele possui na sua vizinhança-8 pelo menos um vizinho direto nulo, ou seja, se pelo menos um de seus quatro vizinhos que compartilham um lado é nulo.

A obtenção do esqueleto reduz-se, inicialmente, a identificação dos pixels esqueletais, podendo ser feita através das definições de multiplicidade (pixels necessários à preservação da topologia e conectividade) estabelecidas por [Pavlidis (1982)] , as quais fazem uso de operações locais com janelas 3x3 sobre os pixels de contorno. Um pixel é dito esqueletal se sua vizinhança-8 coincidir com um dos padrões da Figura 6. Em todos os padrões no mínimo um elemento marcado por A ou B precisa ser não nulo. O primeiro padrão rotacionado uma vez de 90 graus e o segundo padrão rotacionado três vezes de 90 graus também devem ser considerados.

Isto corresponde a um total de seis padrões embutidos na Figura 6.

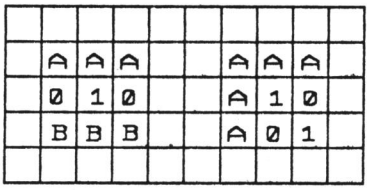

Figura 6 : Padrões de Pixels Esqueletais.

Se o algoritmo utilizar um processamento estritamente seqüencial, o esqueleto gerado preservará a conectividade apenas dos elementos que não possuem furos. Este problema é resolvido examinando-se a imagem em paralelo e não permitindo a remoção de um dado pixel se este foi, em uma iteração anterior, rotulado como esqueletal. Resta ainda um problema. Para um conjunto horizontal composto de duas linhas, todos os pixels são considerados não esqueletais e conseqüentemente serão descartados. Isto é resolvido intercalando-se procedimentos seqüenciais e paralelos. Desta forma não mais serão testados todos os pixels de contorno simultaneamente, mas apenas aqueles que possuem a vizinhança-N igual a zero, onde N assume os valores 0,2,4 e 6, que correspondem respectivamente as posições a direita, acima, a esquerda e abaixo do pixel a ser processado.

O método clássico consiste, então, para cada iteração de afinamento, de quatro varreduras na imagem, uma para cada valor de N. Se um dado pixel tiver sua vizinhança-N igual a zero, ele é testado com os padrões de multiplicidade. Se este for esqueletal, será marcado com um código, por exemplo 2, se não for esqueletal com 3. Ao final das quatro varreduras, os pixels com valor 3 são descartados e os com valor 2 preservados. O processo é repetido até que só reste o esqueleto.

6.0-Codificação

Uma vez afinada a imagem, o próximo passo é realizado pela codificação, com o objetivo de armazenar, apenas, a frente do desenho (pixels acesos). Esta etapa define uma compactação inicial dos dados.

Analisando a imagem de entrada, verifica-se que alguns elementos gráficos, de classes distintas, encontram-se conectados. Por exemplo, a malha do sistema de coordenadas e diversas curvas de isovalores formam um único componente conectado. Este fato reproduz um estado de conectividade indesejável, que é resolvido com a inclusão de um processo que visa separar convenientemente os diversos temas do desenho. Para tal, faz-se necessário a utilização de uma estrutura de dados que suporte a descrição de determinadas características topológicas dos elementos, em especial, os cruzamentos e bifurcações. Uma estratégia é o uso de uma descrição em grafos, onde a imagem é vista como um conjunto de componentes conectados, sendo que cada componente é definido por uma seqüência de nós ou por apenas um caminho, vide Figura 7. Um nó indica a descontinuidade do elemento, e a ele podem estar associados 3 ou 4 caminhos. Ao final da separação, aqui chamada segmentação, cada elemento gráfico é armazenado como um único componente conectado.

O módulo de codificação foi decomposto em dois processos: vetorização e segmentação. A vetorização tem por objetivo representar os elementos gráficos por códigos de direção, ou seja, para uma seqüência de pixels conectados são armazenadas as coordenadas de um pixel terminal (x,y) e os demais representados por códigos de direção (leste = 0, nordeste = 1, norte = 2, noroeste = 3, oeste = 4, sudoeste = 5, sul = 6, sudeste = 7). Já a segmentação separa os diversos temas da imagem. As duas sub-seções seguintes descrevem estes dois processos da Codificação.

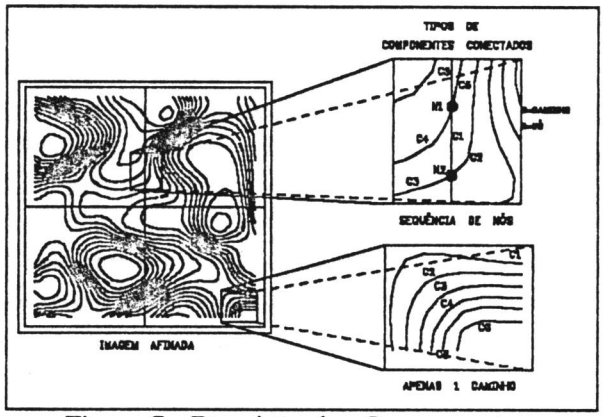

Figura 7 : Descrição dos Componentes.

6.1-Vetorização

A vetorização é composta do: cálculo da função vizinhança, modelagem dos componentes descontínuos e modelagem dos componentes contínuos.

A função vizinhança, V(p), de um dado pixel p, é calculada em termos da vizinhança-8 pela expressão:

$V(p) = \sum_{i=0}^{7} X_i$, onde
$X_i = 1$ para $C(X_i) \neq 0$ e $C(X_{i+1}) = 0$ ou
$X_i = 0$ caso contrário, sendo

$C(X_i)$ o nível de cinza do elemento i, vizinho a p (vide Figura 8). Segundo esta classificação, um pixel é dito isolado, terminal, de passagem, de bifurcação ou de cruzamento, para V(p) igual a 0, 1, 2, 3, e 4, respectivamente.

A vetorização é iniciada com o cálculo da função vizinhança de todos os pixels da imagem. Em seguida, a modelagem dos componentes descontínuos busca os elementos que dispõem de pelo menos um pixel de bifurcação ou cruzamento, transcrevendo-os como componentes conectados do tipo seqüência de nós. A modelagem dos componentes contínuos busca os demais elementos, que são os cíclicos e os abertos, codificando-os como componentes de apenas um caminho.

A modelagem de um componente descontínuo inicia com a busca a um pixel de bifurcação ou cruzamento. Esta característica indica a existência de três ou quatro caminhos associados ao nó. Este nó é inicialmente empilhado e um de seus caminhos percorrido, gerando-se uma seqüência de códigos de direção. A busca termina quando um novo pixel de bifurcação, cruzamento ou terminal for encontrado. Nos dois primeiros casos, empilha-se este novo nó e codifica-se um de seus caminhos. Se o pixel for terminal, retira-se da pilha um nó previamente processado, com o propósito de se esgotar todos os seus caminhos. O procedimento é repetido até que a pilha fique completamente vazia. Resumidamente, os passos do algoritmo são:

```
Buscar nó inicial e empilhar;
Enquanto existir nó na pilha faça
  Desempilhar nó;
  Obter caminho de trilha para o nó corrente;
  Se existir caminho então
    Reempilhar nó corrente;
    Trilhar caminho;
    Se fim da trilha = bif. ou cruz. então
      Empilhar novo nó;
    Fim-se;
  Fim-se;
Fim-enquanto.
```

A modelagem dos componentes contínuos inicia a busca de um caminho com dois tipos distintos de pixel, dependendo do elemento que se deseje vetorizar. A busca aos componentes cíclicos e abertos é iniciada com a identificação de um pixel de passagem ou terminal, respectivamente. A trilha de um dado caminho termina quando se encontrar um pixel terminal ou não existir nenhum outro pixel vizinho ao corrente que não tenha sido visitado.

X3	X2	X1
X4	P	X0
X5	X6	X7

Figura 8 : Vizinhança do Pixel P.

6.2-Segmentação

A segmentação pode ser formulada como: dado um componente conectado, que possui pelo menos dois elementos logicamente distintos, deseja-se transformar este componente em vários outros de forma a se representar individualmente cada um dos elementos. A separação é feita da seguinte maneira:

Identificar componente a ser segmentado;
Obter lista de caminhos do elemento;
Modelar separadamente este elemento;
Eliminar no componente de origem o elemento;
Recodificar o componente de origem.

Analisando o algoritmo, verifica-se que a maior parte dos passos são interações com o operador para uma posterior manipulação de dados. O ponto crítico reside na aquisição da lista de caminhos (definição do elemento a ser separado), que é feito com a intervenção humana. A solução recai, então, na especificação de uma rotina de busca a lista de caminhos que define um pequeno número de intervenções do operador.

Uma característica, a qual foi explorada na implementação, é a forma retilínea de trechos da malha do sistema de coordenadas e a não retilínea das curvas de isovalores. Esta característica foi considerada atributo fundamental para um classificador que mapeia um conjunto de pontos (caminhos) em um conjunto binário que representa a geometria dos elementos (reta e não reta).

Este classificador, que atua na busca como um extrator de propriedades geométricas dos caminhos, foi assim implementado: para um dado conjunto de pontos que descrevem um caminho da imagem,

calcula-se o coeficiente de correlação, Corr(X,Y), com o objetivo de se medir o grau de colinearidade dos pontos. Se $-0.7 \leq \text{Corr}(X,Y) \leq 0.7$, o caminho é dito não retilíneo, caso contrário, ele é retilíneo.

Em função da natureza do classificador utilizado, este método de busca destina-se, principalmente, à segmentação de elementos que tendem a descrever formas retilíneas, podendo dispor de inclinações diversas. A decisão da escolha de que caminho deve ser incorporado à lista é tomada da seguinte maneira: Dado um nó, classificam-se todos os caminhos a ele associados. Os não retilíneos são descartados, enquanto os retilíneos têm suas inclinações calculadas via método dos mínimos quadrados. O caminho a ser incorporado à lista é aquele cuja inclinação mais se aproxima do caminho imediatamente anterior. Se todos os caminhos foram classificados como não retilíneos, a decisão é tomada pelo operador.

Como não foi estabelecido nenhum parâmetro de rejeição dos caminhos retilíneos, é de se esperar que em alguns casos o método eleja um caminho que não faça parte do elemento em processo de busca. Este erro com certeza será propagado, acarretando a identificação de um elemento logicamente inexistente na imagem. Este problema é resolvido com a implementação de uma interface gráfica que informa ao operador cada decisão tomada. Se por ventura ocorrer algum desvio, o usuário solicita uma interrupção e auxilia a identificação do caminho.

7.0-Classificação

O reconhecimento é iniciado com a classificação dos componentes conectados em duas categorias: geométricos e não geométricos. Para o gráfico em estudo, cujos textos e símbolos encontram-se aleatoriamente distribuídos, com diversas formas, tamanhos e inclinações, faz-se necessário a utilização de um método que seja pouco sensível a variações de escala, rotação e forma. Adotou-se o método proposto por [Fletcher-Kasturi (1988)], que explora os atributos área, densidade de pixels acesos e razão altura/largura dos componentes conectados da imagem.

O primeiro atributo, o de área (L_{ar}), tem como objetivo estabelecer uma fronteira numérica entre os elementos mais extensos dos menos extensos, ou seja, discriminam-se em um primeiro instante, as grandes curvas e retas, dos textos, símbolos e pequenas informações gráficas. Seu valor é definido por uma função de duas variáveis: A_{mf} e A_{avg}. A variável A_{mf} é a área de maior número de incidência na imagem, que é obtida calculando-se o histograma de freqüência das áreas. A variável A_{avg} corresponde à média aritmética das áreas dos componentes. O limiar de área de uma imagem é dado pela expressão:

$L_{ar}(A_{mf}, A_{avg}) = 0.10 \times (A_{\max} - A_{\min})$, onde:
$A_{\max} = \max(A_{mf}, A_{avg})$, e
$A_{\min} = \min(A_{mf}, A_{avg})$.

Como os textos e símbolos, em geral, apresentam um comportamento não muito alongado e relativamente denso, pode-se estabelecer para estes elementos uma razão altura/largura que varie de 1:10 à 10:1, e para densidade, um valor mínimo de 10%.

Utilizando-se estes atributos, a classificação automática é feita da seguinte maneira:

```
Calcular histograma de área dos componentes;
Obter área de maior freqüência (A_mf);
Calcular área média dos componentes (A_avg);
Calcular limiar de área (L_ar);
Para cada componente da imagem, faça
  Obter área (A_com);
  Obter densidade (D_com);
  Obter razão altura/largura (R_com);
  Se (A_com < L_ar) e (D_com > 0.10) e
    (R_com ≥ 0.10 e R_com ≤ 10.00) Então
      Componente é não geométrico;
  Senão
      Componente é geométrico;
  Fim-se;
Fim-para.
```

Em função da natureza robusta e experimental do método é de se esperar que em determinadas situações alguns elementos sejam classificados erradamente. A fim de contornar este problema, foi implementado, também, um processo complementar que permite ao operador manipular os limiares e classificar individualmente os componentes da imagem.

8.0-Reconhecimento Geométrico

Uma vez separados os elementos geométricos em duas categorias, o passo seguinte é realizado pelo

reconhecimento geométrico que visa codificar os elementos geométricos, descrevendo-os segundo um formato vetorial definitivo. Para tal, convencionou-se que as informações devem ser modeladas sobre um conjunto de temas gráficos, onde cada tema é constituido de diversos elementos, que por sua vez definem uma ou mais primitivas gráficas. A Figura 9 mostra esquematicamente a hierarquia proposta.

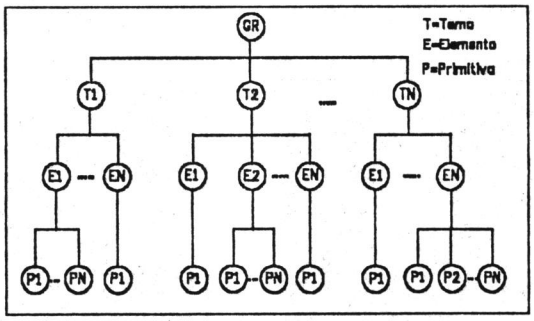

Figura 9 : Descrição Vetorial dos Dados.

Para se chegar a esta modelagem de dados, o reconhecimento geométrico foi projetado com dois processos: geração e manipulação. A geração codifica os componentes geométricos sem a preocupação de organizar as informações hierarquicamente, restringindo-se, apenas, a reduzir a quantidade de pontos dos componentes. A manipulação, completa o reconhecimento, estruturando os dados segundo a convenção adotada. As duas sub-seções seguintes descrevem estes dois processos.

8.1-Geração dos Elementos Geométricos

A geração vetoriza definitivamente os componentes conectados. O processo, como já citado anteriormente, não se preocupa em organizar, ou mesmo, definir os temas representados no documento. Todos os elementos são, nesta etapa, associados a um mesmo tema, sem qualquer significado lógico. Na verdade, o processo restringe-se a descrever vetorialmente somente os componentes geométricos com a preocupação de reduzir o número de pontos. Para sua implementação foram estudados três algoritmos de detecção de pontos dominantes em curvas digitais fechadas: [Rosenfeld-Johnston (1973)], [Rosenfeld-Weszka (1975)] e [Teh-Chin (1989)].

O algoritmo Rosenfeld-Johnston estima, inicialmente, as curvaturas discretas de cada ponto. Em seguida, aplica, em duas etapas, um processo de supressão de modo a preservar apenas os pontos dominantes, ou seja, aqueles que possuem maior valor de curvatura. Na primeira etapa de supressão, as curvaturas são comparadas com um valor limite, com o intuito de descartar os pontos de baixa curvatura. Na segunda, aplica uma supressão não máxima, onde são eliminados os pontos que não dispõem de curvaturas máximas locais.

Para uma seqüência de N coordenadas que descrevem uma curva fechada C, $C = \{p_i = (x_i, y_i), i = 1,...,N\}$, onde p_{i+1} é o ponto vizinho a p_i, o algoritmo de Rosenfeld-Johnston é composto dos seguintes passos:

a - Definir para cada ponto p_i, seus k-vetores:
$\vec{a}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k})$ e
$\vec{b}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k})$,
e os k-cosenos de p_i:

$$\cos_{ik} = \frac{\vec{a}_{ik} \bullet \vec{b}_{ik}}{|\vec{a}_{ik}|\,|\vec{b}_{ik}|};$$

b - Escolher um fator de suavização m(N/15) baseado no nível de detalhe de C.
Para cada ponto p_i, computar
$\{\cos_{ik}, k = 1,...,m\}$;

c - Atribuir a cada ponto p_i a região de suporte h_i e o valor de curvatura $\cos_{i,hi}$ para o maior valor de h tal que
$\cos_{i,m} < \cos_{i,m-1} < ... < \cos_{i,hi} \geq \cos_{i,hi-1}$;

d - Definir como dominantes os pontos cujos $\cos_{i,hi} \geq \cos_{j,hj}$, para todo j tal que
$|i - j| \leq h_i/2$.

Além do parâmetro de entrada (m), o procedimento acima apresenta outro problema: a descaracterização de curvas que dispõem de cantos contíguos. Isto é resolvido com o algoritmo de Rosenfeld-Weszka, que é uma versão melhorada do procedimento Rosenfeld-Johnston, diferenciando-se, apenas, no cálculo dos k-cosenos, que é feito através das expressões:

$$\cos_{ik} = \frac{2}{k+2} \sum_{j=k/2}^{k} \cos_{ij} \quad \text{,para k = par}$$

$$= \frac{2}{k+3} \sum_{j=(k-1)/2}^{k} \cos_{ij} \quad \text{,para k = ímpar.}$$

Analisando os dois procedimentos de Rosenfeld, constata-se que o ponto crítico reside na determinação da região de suporte (passo c). Nota-se que se for escolhido um inadequado parâmetro de suavização m, as regiões de suporte,

por sua vez, serão determinadas de maneira errada, com a conseqüente propagação do erro para o cálculo das medidas de curvatura discreta.

O problema da determinação da região de suporte tem sido estudado por um grande número de autores que afirmam que cada ponto de uma curva fechada possui sua própria visão da curva, portanto, a vizinhança de cada ponto que delimita a região de suporte deve variar de ponto a ponto. Este conceito vai de encontro aos algoritmos de Rosenfeld, onde uma vizinhança fixa (m) é estabelecida, inicialmente, para o cálculo.

Teh-Chin sugere uma alternativa extremamente interessante para obtenção da região de suporte, que além de não necessitar de nenhum parâmetro de entrada, analisa vizinhanças diferentes para cada ponto da curva. O método, de natureza iterativa, é assim definido:

- Seja o comprimento de corda que une os pontos p_{i-k} a p_{i+k}, $l_{ik} = |\overline{p_{i-k}p_{i+k}}|$, e d_{ik} a distância perpendicular do ponto p_i a corda l_{ik}.
- Para k iniciando em 1, computar l_{ik} e d_{ik} até que se satisfaça uma das condições abaixo:

  a - $l_{ik} \geq l_{i,k+1}$, ou

  b - $\frac{d_{ik}}{l_{ik}} \geq \frac{d_{i,k+1}}{l_{i,k+1}}$

Finalizando, o algoritmo de detecção de pontos dominantes, proposto por Teh-Chin, é definido da seguinte forma:

- Determinar a região de suporte de cada ponto da curva utilizando o método acima;
- Seja $S(p_i)$ a medida de significância de Rosenfeld-Wezka (k-cosenos). Calcular para cada ponto p_i a função $S(p_i)$;
- Executar supressão não máxima, retendo os pontos p_i onde $S(p_i) \geq S(p_j)$ para todo j tal que $|i-j| \leq \frac{k_i}{2}$;
- Para os pontos restantes, testar:

  Se $D(p_i) = 1$ e existe (p_{i-1} ou p_{i+1}) e ($S(p_i) \leq S(p_{i-1})$) ou ($S(p_i) \leq S(p_{i+1})$)

  Suprimir p_i

Este último algoritmo foi o que apresentou melhores resultados, aliando acuracidade à redução de pontos. A Figura 10 ilustra os resultados.

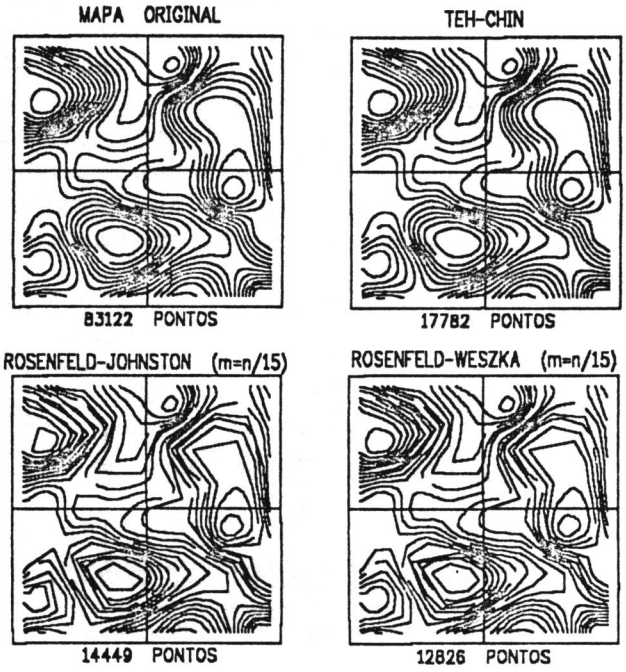

Figura 10: Resultados da Detecção de Pontos Dominantes.

8.2-Manipulação dos Elementos Geométricos

Uma vez gerada a representação vetorial dos elementos, a manipulação completa o reconhecimento geométrico, reconstruindo formas, reconhecendo primitivas e organizando hierarquicamente as informações. A manipulação foi projetada com três processos distintos: ajustamento geométrico, reconhecimento de curvas de isovalores e edição geométrica.

O ajustamento visa reconstruir feições com formas bem definidas e que devido a problemas de digitalização sofreram algum tipo de distorção. Para se corrigir estes elementos, foram implementadas funções que interpolam retas, retângulos, círculos, arcos e elipses, via método dos mínimos quadrados.

A identificação das curvas de isovalores foi implementada de duas maneiras: busca individual e via definição de feixes. A primeira, busca individual, o operador posiciona o dispositivo de interação sobre o elemento. Em seguida, o processo realça a primitiva e solicita a entrada do nível a ser atribuído à curva. A segunda maneira, a identificação via definição de feixes, explora a regularidade de variação numérica dos níveis, e também, a característica de que não existem cruzamentos entre

curvas de nível. A identificação consiste na definição de um feixe (uma reta) que intercepta um grande número de elementos que devem ser transformados em curvas de nível. Definido o feixe, o processo responde realçando as poligonais que interceptam a reta, e solicita a entrada do nível inicial, o intervalo de variação dos níveis e a especificação do acidente geográfico, ou seja, se o feixe, em relação ao primeiro ponto da reta, define um aclive ou um declive.

Finalizando, a edição geométrica organiza hierarquicamente as informações segundo a estrutura de dados proposta e oferece um conjunto de ferramentas que permitem alterações, quer na localização espacial das feições, quer na especificação de atributos de visualização gráfica.

9.0-Reconhecimento Não Geométrico

O Reconhecimento Não Geométrico se constitui, principalmente, na utilização de um método de reconhecimento de caracteres e símbolos com variações de escala e rotação. Apesar do assunto ser de grande importância para o problema em estudo, ele não foi aqui descrito face a sua extensão. Para maiores detalhes, ver [Varella (1992)] .

10.0-Análise dos Resultados

O procedimento foi implementado em ambiente IBM, utilizando estações gráficas 5080 e um computador de grande porte 3090-600J/6VF. Todos os módulos foram escritos em FORTRAN/VS, visando explorar toda a capacidade de processamento vetorial da instalação.

Para a realização dos testes dos módulos do procedimento, foi digitalizado um gráfico em perfeito estado de conservação, produzindo-se uma imagem de 2808x2040 pixels, vide Figura 11. A seguir são apresentados os resultados obtidos.

Iniciando a análise pelo módulo de Pré-Processamento, pode-se afirmar que, em função da natureza dos filtros projetados, é de grande importância que se aplique a remoção de pixels isolados, a suavização direcional e a suavização pela maioria com teste de contorno em todas as imagens. A fim de se quantificar as alterações produzidas na imagem digitalizada, a remoção de pixels isolados alterou 84 pixels, a suavização direcional alterou 25 pixels e a suavização pela maioria com teste de contorno alterou 4568 pixels.

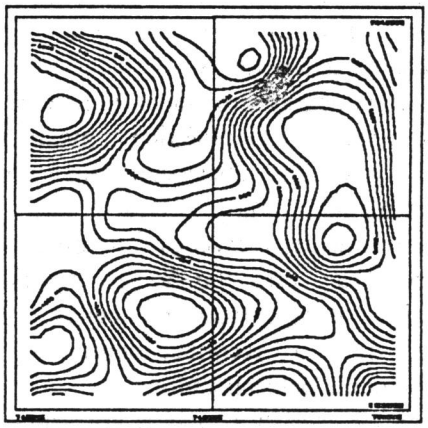

Figura 11: Gráfico Digitalizado.

Quanto ao afinamento, o método clássico, apesar de lento, mostrou resultados altamente satisfatórios, definindo corretamente os eixos médios e preservando a conectividade e a topologia dos elementos. A Figura 12 mostra o resultado do afinamento clássico, onde foram realizadas 64 varreduras na imagem para a obtenção do esqueleto.

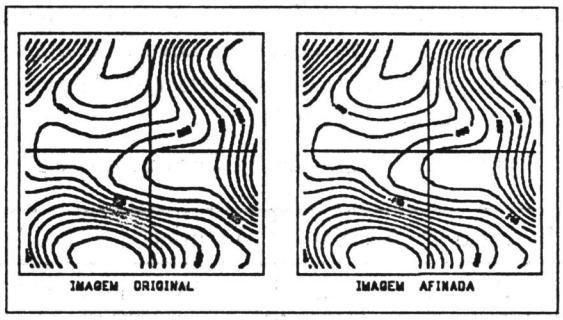

Figura 12: Resultado do Afinamento.

No que se refere a compactação das informações, a vetorização gerou 223 componentes conectados, organizados em 125 nós e 431 caminhos. Após a segmentação, esta mesma imagem passou a ser descrita por 296 componentes conectados, distribuídos em 214 nós e 427 caminhos. O gráfico que antes era descrito por uma imagem de 2808x2040 pixels, ocupando 1.07 mbytes, passou a ocupar, em formato vetorial, 77.98 kbytes.

Na classificação, 83% dos textos foram identificados corretamente através do procedimento automático. No entando, ajustando-se o limiar de área, interativamente, conseguiu-se atingir 100%. Este índice de 83% deveu-se à existência de um grande número de pequenos componentes não geométricos, gerados durante a digitalização, que na verdade se constituem em ruídos. Estes minúsculos

componentes deslocaram o limiar de área para baixo, classificando os textos de maior área em componentes geométricos.

As ferramentas projetadas para o módulo de Reconhecimento Geométrico mostraram um elevado grau de eficiência, em especial, o reconhecimento via feixes que agilizou consideravelmente a execução do processo. A fim de quantificar esta tarefa, o reconhecimento geométrico foi executado em apenas 4 min. Vale lembrar que o reconhecimento é realizado interativamente pelo operador.

Uma característica, que se apresenta de forma casual, é que até à vetorização todos os processos são completamente automáticos. Este fato permite a implementação de um procedimento único que engloba o Pré-processamento, Afinamento e a Vetorização, podendo ser executado sem a intervenção do operador. Conseqüentemente, esta etapa, apesar de consumir um elevado tempo de CPU, não se constitui em tempo de operação. A título de ilustração, a Tabela 1 mostra os tempos de execução, em instruções escalares e vetoriais, do procedimento para o processamento da imagem correspondente a Figura 11.

Tabela 1 : Tempos de Processamento (seg).

Processo	Escalar	Vetorial	Total
Pré-Proc.	10.812	17.184	27.996
Afin.	247.854	176.557	424.411
Vetor.	170.908	62.391	233.299
Totais	429.574	256.132	685.706

11.0-Conclusões

O reconhecimento de elementos gráficos, representados em forma de imagens, se constitui em um problema não trivial, que é resolvido, em alguns casos, através da intervenção do usuário. Este fato impôs a especificação de procedimentos que exploraram, tanto a interface homem-máquina, quanto algumas características geométricas dos elementos digitalizados.

A acuracidade da representação das informações cartográficas foi considerada como um dos aspectos de maior importância. Entretanto, se forem utilizados *Scanners* de baixa resolução, inferior a 400 dpi, o desempenho pode ser seriamente comprometido.

Dos processos implementados, o afinamento é o que apresenta o maior esforço computacional, em decorrência do elevado número de varreduras na imagem. O algoritmo sugerido, não deve ser executado em tempo real, especialmente, em instalações com baixa capacidade de processamento.

O trabalho sugeriu um procedimento computacional semi-automático para a aquisição de *grandes* volumes de dados espaciais, que propôs a utilização de *Scanners*, com o objetivo de agilizar o processo e definir um ambiente amigável e preciso para a execução da tarefa.

Referências

- T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Rockville, 1982.
- K. Ramachandran, Coding Method for Vector Representation of Engeneering Drawings, Proc. of the IEEE, 1980, Vol.68, Num.7.
- D. Ting & B. Prasada, Digital Processing Techniques for Encoding of Graphics, Proc. of the IEEE, 1980, Vol.68, Num.7.
- L.A. Fletcher & R. Kasturi, A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images, IEEE Trans. on Patt. Anal. and Mach. Int., 1988, Vol.10, Num.6, PP.910-918.
- A. Rosenfeld & E. Johnston, Angle Detection on Digital Curves, IEEE Trans. Comput., 1973, Vol.C-22, PP.875-878.
- A. Rosenfeld & J.S. Weszka, An Improved Method of Angle Detection on Digital Curves, IEEE Trans. Comput., 1973, Vol.C-22, PP.875-878.
- C. Teh & R. Chin, On the Detection of Dominant Points on Digital Curves, IEEE Trans. on Pattern Analysis and Machine Inteligence, 1989, Vol.11, Num.8, PP.859-872.
- L.E.S. Varella, Reconhecedor de Elementos Gráficos Digitalizados via Scanners, Dissertação de Mestrado, Instituto Militar de Engenharia, RJ, 1992.