

## Extração de Feições Cartográficas

Helien Eda Rodrigues Gato <sup>1</sup>

Nilton Nobuhiro Imai <sup>2</sup>

Antonio Maria Garcia Tommaselli <sup>3</sup>

<sup>1 2 3</sup> Universidade Estadual de São Paulo  
Faculdade de Ciências e Tecnologia  
Programa de Pós-Graduação em Ciências Cartográficas  
Rua Roberto Simonsen, 305, C.P. 957  
19.060-900 Presidente Prudente SP

<sup>1</sup> Universidade de Franca  
Av. Dr. Armando Salles Oliveira, 201, C.P. 082  
14.404-600 Franca SP

<sup>1</sup> ✉ helienbr@yahoo.com.br

<sup>2</sup> ✉ nnimai@prudente.unesp.br

<sup>3</sup> ✉ tomaselli@prudente.unesp.br

Conteúdo	
	1. Introdução
	2. Etapas do Processo de Extração de Feições
	2.1 Segmentação por crescimento de regiões
	2.1.1 Crescimento de região através de nível de brilho médio
	2.1.2 Critério estatístico
	2.1.3 Comparação de médias amostrais
	2.1.4 Segmentação através de limiares
	2.2 Refinamento
	2.2.1 Perseguição de contorno
	2.3 Vetorização
	2.3.1 Determinação das equações de retas detectadas
	3. Protótipo para extração de feições
	4. Resultados
	4.1 Segmentação por Crescimento de Região
	4.2 Rotinas de Extração
	5. Conclusões e Recomendações
	6. Referências Bibliográficas

**Resumo:** Este trabalho apresenta o estudo e implementação de técnicas de processamento digital de imagens utilizando-se uma abordagem semi-automática para extração de feições cartográficas. Para a realização dos testes, foi desenvolvido um protótipo na linguagem C++ Builder. Foi implementado um conjunto de rotinas capazes de realizar cada etapa necessária ao processo de extração de feições, além da adaptação do algoritmo de crescimento de região a diversos critérios de homogeneidade. Os resultados obtidos pelos testes efetuados com diversas combinações de processamento são apresentados e discutidos. Com a análise do banco de feições (retas) gerados no fim do processo, é possível mostrar que a abordagem semi-automática é viável e robusta.

**Palavras chave:** Segmentação por crescimento de região, Processamento Digital de Imagens, Extração de feições retas;

**Abstract:** This work presents the study and implementation of tools for digital image processing using a semiautomatic approach for cartographic feature extraction. Tests with real data were performed using a software prototype implemented in the C++ Builder Programming Environment. A bundle of functions were implemented in order to perform each step of feature extraction and, besides that, the region growing algorithm was adapted to several homogeneity criteria. The obtained results after several tests with different processes combination are presented and discussed. Considering the analysis of the generated straight lines parameters at the end of the process it is feasible to show that the proposed semiautomatic approach is viable and robust.

**Keywords:** Region Growing Segmentation, Digital Image Processing, Feature Extraction;

### 1. Introdução

A crescente evolução tecnológica, tanto em sistemas de aquisição de imagens quanto em computadores, resultou na diminuição dos custos com sistemas de aquisição de imagens e o aumento da velocidade de processamento. Esses fatores contribuíram fortemente no desenvolvimento de novas gerações de sistemas de informação geográfica com a integração de funcionalidades de processamento de imagens. Portanto a extração de informações espaciais a partir de imagens digitais ainda é um importante processo a ser aprimorado.

Como os métodos totalmente automatizados ainda não geram resultados satisfatórios, quando aplicados a imagens reais, o desenvolvimento de métodos semi-automáticos para a extração de feições cartográficas, a partir de imagens aéreas, é uma solução a ser investigada. As imagens utilizadas nesse processo de extração são de áreas urbanizadas (cidades), onde as construções

representam os alvos a serem identificados.

A dificuldade encontrada nos métodos totalmente automatizados contribui para que os métodos semi-automáticos (ou assistidos) sejam caracterizados pela interação de um operador com o sistema. Portanto o processo pode ser controlado pelo operador, que é responsável por tomar algumas decisões simples como indicar a região onde se encontra o alvo a ser extraído.

O presente trabalho tem como objetivo mostrar os resultados obtidos pela investigação de uma abordagem semi-automática para extração de feições cartográficas que representem regiões. Todas as ferramentas foram desenvolvidas num protótipo com base nas técnicas de processamento digital de imagens, métodos estatísticos e MMQ.

## 2. Etapas do Processo de Extração de Feições

As etapas envolvidas no processo de extração de feições desenvolvido são: Pré-processamento, Segmentação, Refinamento e Vetorização. Essas etapas devem seguir esta ordem e são explicadas a seguir:

*Pré-processamento:* possui a função de melhorar a qualidade da imagem quando ela apresenta distorções radiométricas e ruídos, como uma forma de tentar melhorar a qualidade da imagem e os resultados das fases posteriores;

*Segmentação:* visa particionar a imagem em diversas partes ou objetos constituintes. Um fator importante é a escolha do tipo de segmentador, que pode ser mais adequado a segmentar limites (bordas) ou regiões que é o caso tratado neste trabalho. Essa etapa é fundamental para se obter um bom resultado no processo de extração;

*Refinamento:* essa etapa é responsável pela filtragem dos objetos segmentados, pela identificação e extração das informações que caracterizam um objeto ou feição;

*Vetorização:* é responsável por definir os elementos que geram a representação vetorial da feição;

Além das etapas descritas acima, outro item que auxilia no processo com um todo é a *Base de conhecimento*, que é responsável por fornecer heurísticas e parâmetros adequados para direcionar o processo computacional nas etapas do processo. Sua alimentação pode ser dada através de algum parâmetro inicial fornecido pelo usuário, ou por heurísticas pertinentes a uma situação específica no processamento [Gonzalez e Woods, 1993].

### 2.1 Segmentação por Crescimento de regiões

O método de segmentação por crescimento de regiões realiza uma busca por grupos de pixels com alguma propriedade de similaridade, tentando agrupar os pixels de uma área com brilho similar, isto é, que caracterizam uma mesma região.

Inicialmente, o método começa com um pixel e, a partir daí, examina seus vizinhos, numa seqüência, para decidir se eles possuem brilho similar, segundo o critério de similaridade escolhido. Se os pixels vizinhos analisados forem aceitos como similares, então eles são agrupados ao pixel inicial (semente) para formar uma região. Desta forma, as regiões vão sendo desenvolvidas a partir de pixels sementes (únicos) (Pavlidis, 1982).

Uma forma melhorada é não começar com pixels, mas com uma partição da imagem em um conjunto de pequenas regiões. Um teste de uniformidade é, então, aplicado a cada região. Se o teste falhar, a região é subdividida em elementos menores até que todas as regiões fiquem uniformes. Então, as regiões se desenvolveriam de regiões menores. A vantagem em usar pequenas regiões ao invés de pixels é a redução da sensibilidade ao ruído (Gonzalez e Woods, 1993).

Foi realizada uma pequena adaptação desse método através da utilização de diversos critérios como medida de similaridade.

#### 2.1.1 Crescimento de região através de nível de brilho médio

Um dos métodos tem como critério de uniformidade, a comparação da máxima diferença entre o valor de um pixel ( $P$ ) e a média ( $m$ ) sobre a região. Para uma região  $R$  de tamanho  $N$  tem-se:

$$m = \frac{1}{N} \sum_{P \in R} f(P) \quad (1)$$

então pelo critério adotado a região é considerada uniforme se

$$\max_{P \in R} |f(P) - m| < T \quad (2)$$

para algum limiar  $T$ , onde  $T$  pode ser representado como uma heurística ao processo (Gonzalez e Woods, 1993).

#### 2.1.2 Critério estatístico

O critério de uniformidade da equação (2) apresenta-se pouco eficiente devido aos efeitos da região classificada segundo a confiabilidade das estimativas da região pela média, o que torna o teste muito exigente na decisão de incluir um pixel na região. Uma alternativa proposta neste trabalho é baseada em um valor da média ( $m$ ) e desvio padrão ( $s$ ) da vizinhança do pixel semente, podendo assim utilizar o critério de similaridade a seguir:

$$m - \Delta_m < f(P) < m + \Delta_m \quad (3)$$

onde:

$$\Delta_m = k \cdot s \quad (4)$$

$$s = \sqrt{\frac{\sum_{P \in R} (f(P) - m)^2}{N - 1}} \quad (5)$$

e  $k$ : representa o número de desvios padrões aceitos.

Para diminuir a grau de exigência no teste de um pixel, basta aumentar o número de desvios padrões aceitos como variação da população.

### 2.1.3 Comparação de médias amostrais

Uma abordagem um pouco diferente de uniformidade pode ser empregada comparando-se as estatísticas de uma região inicial (máscara da semente) com as estatísticas calculadas em outras regiões vizinhas (máscara de varredura). Se elas estão próximas uma da outra, isto é, se as regiões podem ser consideradas estatisticamente iguais, então elas são chamadas de uniformes e o pixel central da segunda é classificado como pertencente à região que está sendo segmentada.

Um critério adotado para a implementação desse método foi baseado na distribuição de "T de Student" (Spiegel, 1985) com o teste  $t$  para observações independentes para comparar duas populações (semente e varredura). Inicialmente é necessário estabelecer o nível de significância ( $\alpha$ ) e em cada iteração calcular:

- média da semente ( $m_1$ ) e da janela de varredura ( $m_2$ )
- variância da semente ( $s_1^2$ ) e da janela de varredura ( $s_2^2$ )
- variância ponderada

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{(n_1 + n_2 - 2)} \quad (6)$$

Valor de

$$t = \frac{m_2 - m_1}{\sqrt{s^2 \left( \frac{1}{n_1} + \frac{1}{n_2} \right)}} \quad (7)$$

onde:

$n_1$ : número de elementos do grupo 1, ou seja, da semente; e

$n_2$ : número de elementos do grupo 2, ou seja, da janela de varredura.

O valor de  $t$  é comparado com o valor da tabela (tabela estatística da distribuição  $t$  de Student, vide apêndice III de Spiegel, 1985.) ao nível de significância ( $\alpha$ ) estabelecido e com  $(n_1 + n_2 - 2)$  graus de liberdade. As duas populações (semente e varredura) são consideradas estatisticamente iguais se o valor de  $t$  (em valor absoluto) for menor que o valor da tabela.

### 2.1.4 Segmentação através de limiares

Nesta abordagem, o critério de uniformidade é definido a partir de limiares, verificando se o valor de brilho do pixel em análise ( $p$ ) enquadra-se na medida de similaridade adotada. Foram adotadas três formas possíveis para esse critério:

- $p < (\text{limiar } 1)$
- $p > (\text{limiar } 1)$
- $(\text{limiar } 1) < p < (\text{limiar } 2)$

Dessa forma, o pixel  $p$  será adicionado à região se a sentença selecionada for verdadeira. Apesar de o método ser bem simples, ele apresenta-se muito robusto quando se leva em consideração que os limiares são definidos pelo próprio usuário. Essa definição pode ser feita com base nos dados do histograma de uma sub-região da imagem onde o objeto a ser segmentado se encontra. Como os histogramas dessas sub-imagens do mundo real não possuem um modelo de representação padrão, tornou-se inviável a utilização de algoritmos de detecção de limiares automáticos.

## 2.2 Refinamento

A etapa de refinamento engloba a detecção dos elementos que compõem o contorno. O refinamento é responsável por filtrar e classificar os dados obtidos pelo processo de segmentação, encontrando o conjunto de pixels que representem as feições.

Após a segmentação de uma imagem ou região, é necessário classificar os dados obtidos em uma estrutura adequada para que o processo de extração de feições seja concluído. Uma das formas de se realizar essa tarefa é pelo processo de conexão ("linking"), responsável por conectar todos os pixels pertencentes a uma mesma borda, constituindo, assim, uma única feição. Esse processo pode ser realizado pelo método de conexão por varredura e rotulação (Zhou et al., 1989). Porém, quando o resultado da segmentação é uma região, a tarefa seguinte será determinar, primeiramente, quais pixels pertencem ao contorno para, depois, classificá-los em segmentos distintos.

### 2.2.1 Perseguição de contorno

O contorno de um conjunto conectado de pixels  $R$  é definido como o conjunto de todos os pixels em  $R$  que têm, pelo menos, um vizinho que não pertença a  $R$ , onde  $R$  representa uma região.

Portanto, o processo de perseguição de contorno procura encontrar todos os elementos que formam o contorno de uma região, através de uma análise de vizinhança local.

Através desse algoritmo, são detectados e rotulados apenas os elementos pertencentes ao contorno da região.

Para a execução do algoritmo, é necessária a existência de um elemento ( $p$ ) como ponto de partida ao processo. Esse pixel inicial deve pertencer ao contorno e pode ser obtido através da etapa de segmentação por crescimento de região. A partir do pixel inicial, o método consiste em rotular todos os pixels vizinhos ao pixel atual, sendo que, para um elemento ser rotulado, ele deve ter, pelo menos, um vizinho pertencente ao fundo, ou seja, ser um elemento do contorno. Assim, o método vai classificando todos os elementos que pertencentes à borda.

Observa-se, na figura 1, o objeto segmentado (região cinza) e o seu fundo branco (não-cinza), sendo o pixel  $p$  o ponto inicial do processo, inicialmente rotulado. Pode-se observar que dos seus 8-vizinhos, apenas três são elementos do contorno, sendo esses também rotulados recursivamente. No exemplo, X é o rótulo adotado.

O processo termina quando todos os elementos pertencentes ao contorno estiverem rotulados.

Neste trabalho, o algoritmo de perseguição de contorno foi utilizado de forma adaptada. Nesse caso, a cada pixel anexado ao contorno, uma reta é ajustada a um conjunto e um erro vai sendo calculado. Se o erro ultrapassar um determinado limite, então o pixel atual passa a ser o elemento inicial de uma nova reta que é originada, e assim por diante. O resultado é armazenado numa estrutura de dados específica, neste caso, num banco de dados onde diversas retas vão sendo armazenadas para posterior refinamento.

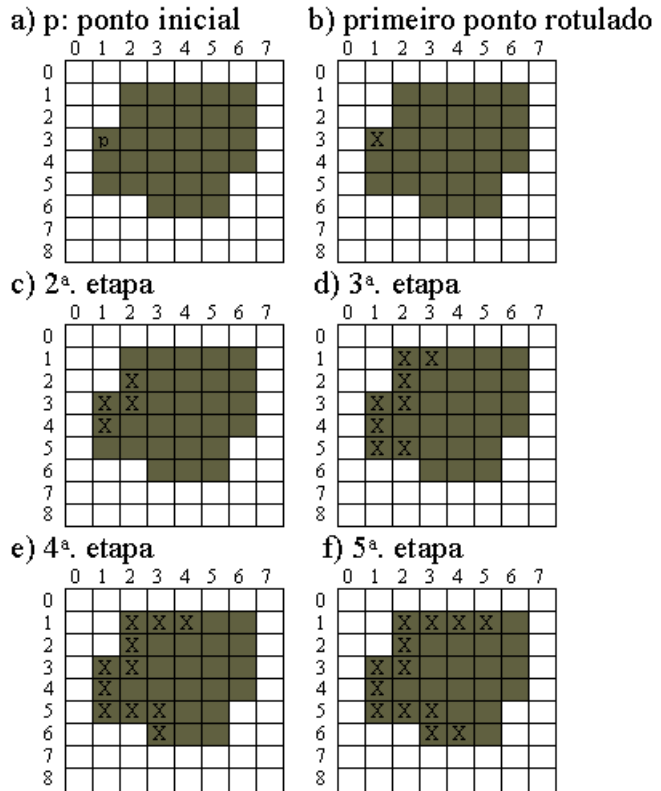


Figura 1 : Etapas da rotina de perseguição de contorno  
(Fonte: Gato, 2000)

## 2.3 Vetorização

Nesta etapa estão incluídos os processos para ajustar as retas detectadas (bordas/contorno) pelas etapas anteriores e encontrar, então, as interseções entre elas, ou seja, os vértices que definem a feição.

### 2.3.1 Determinação das equações de retas detectadas

A rotina de perseguição de contorno faz chamadas ao processo de ajustamento. Cada pixel do contorno vai sendo rotulado numa seqüência, sendo que o rótulo está associado a uma reta que é definida pelo ajustamento dos pixels deste. O processo de ajustamento é executado para cada novo pixel que é rotulado, a partir do momento em que o contorno possui pelo menos três pixels, pois com três pontos já se torna viável o ajuste. Após a identificação dos pixels que pertencem a uma dada reta (de mesma rotulação), é feita a determinação da equação da reta que melhor se ajusta aos mesmos (ajustamento). Para tanto, o método dos mínimos quadrados (M.M.Q.) foi utilizado. Este método consiste em minimizar as distâncias entre os pontos e a reta a que são ajustados, ou seja, obter a melhor equação da reta para representar esses pontos. No caso de uma minimização insuficiente, a reta pode ser quebrada em outras menores, até o momento que elas estejam dentro de um limite de qualidade (previamente estabelecido).

## 3. Protótipo para extração de feições

O protótipo foi desenvolvido no ambiente C++ Builder, que permite a utilização de técnicas de programação orientadas a objeto, a

eventos e a tradicional programação estruturada. Os recursos do ambiente oferecem grande facilidade para manipulação de imagens, bancos de dados e interfaces gráficas no ambiente "Windows".

A definição do sistema foi baseada na necessidade de gerar um ambiente versátil, capaz de abrir, processar, visualizar diversas imagens em tons de cinza e os resultados provenientes das ferramentas desenvolvidas.

As etapas do processo de extração de feições abordadas neste trabalho foram implementadas no protótipo "SEF", que oferece também uma variedade de ferramentas de manipulação de imagens destinadas ao pré-processamento, análise do usuário, configuração do software e fornecimento de heurísticas pelo usuário.

Foi implementada uma interface amigável e didática, permitindo ao usuário escolher a imagem a ser processada e a opção de processamento desejada. O usuário pode também determinar o objeto a ser processado e visualizar o resultado obtido. Isso permite uma análise crítica do processo o que pode acarretar na modificação das heurísticas adotadas a partir da tomada de decisões pelo usuário, ou seja, é possível realizar uma etapa apenas ou escolher uma combinação delas até chegar à realização total do processo.

A figura 2 mostra a interface do protótipo de extração de feições "SEF".



Figura 2 : Interface do sistema protótipo "SEF"

O sistema trabalha apenas com imagens no formato de mapa de bits (bitmap), ou seja, com extensão "bmp".

O menu principal disponibiliza as principais opções do sistema. O menu *Arquivo* oferece acesso às operações correspondentes à manipulação dos arquivos de imagens e de banco de dados, além de permitir a finalização do programa. O menu *Visualização* oferece as opções "plotagem" e "banco de dados". "Plotagem" corresponde ao processo de visualização das feições já extraídas que foram armazenadas numa base de dados e "banco de dados", à visualização dessa base de feições. O menu *Ferramentas* disponibiliza as diversas técnicas de processamento de imagem existentes no sistema.

#### 4. Resultados

Para a realização dos testes foi utilizada uma imagem aérea de área urbana na escala de 1:8000, como mostra a figura 3.



Figura 3 : Exemplo de imagem real

#### 4.1 Segmentação por Crescimento de Região

A seguir, serão apresentados os diversos resultados obtidos com as técnicas de segmentação implementadas no sistema.

Para cada teste, é necessário, que o usuário selecione a área de extração e posteriormente forneça um ponto de partida (pixel semente) ao processo, que é dado através de um simples “clique” do mouse (botão esquerdo) sobre o objeto a ser segmentado.

- **Teste 1:** Utilizou o algoritmo de segmentação por crescimento de região com critério de similaridade “média e desvio padrão” com variação do tamanho da janela e K fixo.

**Parâmetros:** Máscara: 3x3, 5x5, 7x7, 9x9, 11x11, 13x13 e 15x15 respectivamente, com  $k=2$ , onde:

- máscara 3x3 :- representa uma pequena amostra de elementos, definida por uma janela de 3 pixels de largura por 3 de comprimento. A mesma definição é aplicada às outras máscaras, mudando apenas a dimensão da janela.

-  $k=2$  :- representa o número de desvios padrões ( $\hat{\sigma}$ ) aceitos em torno do valor médio estimado ( $\hat{m}$ ).

Nesta classificação os pixels da região são definidos como sendo  $\{x \mid \hat{m} - k\hat{\sigma} \leq x \leq \hat{m} + k\hat{\sigma}\}$ .

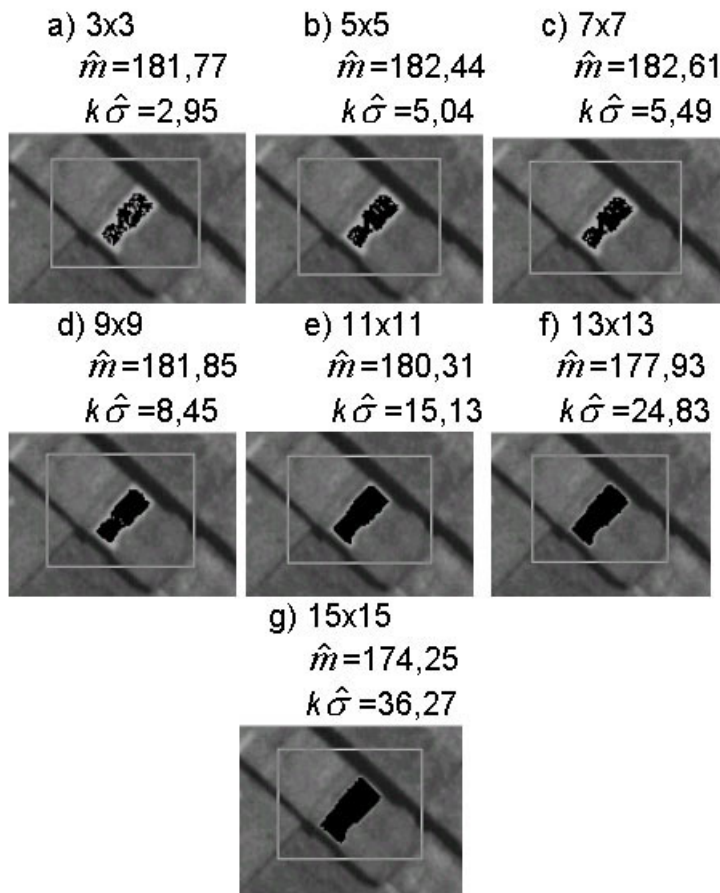


Figura 4: Variação no tamanho da máscara e  $k=2$  fixo.

Pode-se observar na figura 4, que os resultados obtidos com as máscaras 3x3, 5x5, 7x7 e 9x9 não foram adequados para a continuação do processo, pois o objeto não foi totalmente segmentado. Isso ocorreu devido às imagens do mundo real possuírem grande variação de tons de cinza e, sendo assim, uma pequena amostra pode não ser representativa do comportamento do objeto em questão.

Analisando a variação do valor obtido por ( $k\hat{\sigma}$ ), pode-se notar que o aumento do número de elementos da amostra está correlacionado a um incremento significativo da variância estimada, como mostra a figura 4, a máscara de tamanho 11x11 com  $k=2$  gerou uma boa segmentação.

- **Teste 2 (variação de k):** Utilizou o algoritmo de segmentação por crescimento de região com critério de similaridade “média e desvio padrão” aplicado a imagem real;

**Parâmetros:** Máscara: 3x3 e  $k=2$ , 6, 10 e 14 respectivamente.

Nesse teste, uma máscara de tamanho 3x3 é aplicada com os diferentes valores de  $k$  conforme mostra a seqüência de imagens na figura 5. O valor médio estimado ( $\hat{m}$ ) através dos pixels da região definida pela máscara 3x3 é 181,77. Os valores de “ $k\hat{\sigma}$ ” calculados em cada seqüência mostra a medida de dispersão obtida com a variação de  $k$ .

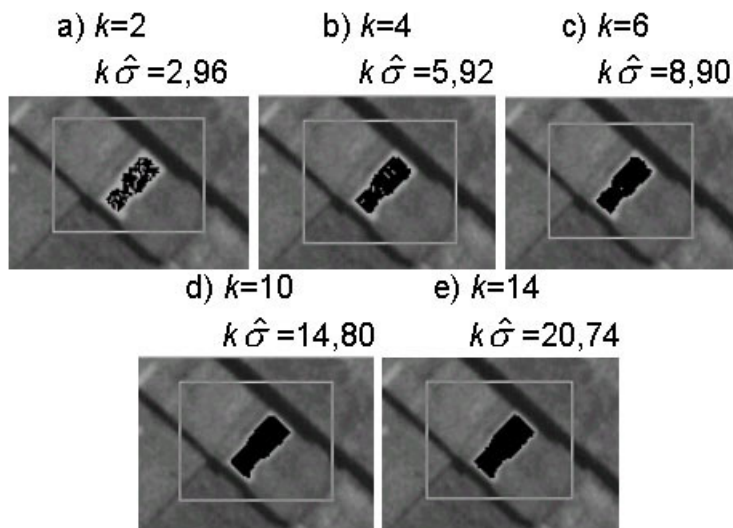


Figura 5 : Máscara 3x3 fixa com  $k$  variando

É possível perceber que os experimentos d) e e) apresentaram os melhores resultados. Mesmo com uma amostra pequena (3x3), o aumento do limiar de aceitação em torno da média ( $k$ ) conseguiu representar a variação existente na região.

- **Teste 3:** Utilizou o algoritmo de segmentação por crescimento de região com critério de similaridade por comparação de médias amostrais, baseado no teste "T de Student". Neste caso é necessário configurar alguns parâmetros, dentre eles "semente", "varredura" e " $\alpha$ ".

Os parâmetros usados em a), b) e c) da figura 6 foram:

- a) semente 11x11, varredura 3x3 e  $\alpha = 0,5\%$ ;
- b) semente 11x11, varredura 3x3 e  $\alpha = 10\%$ ;
- c) semente 3x3, varredura 3x3 e  $\alpha = 0,5\%$ .

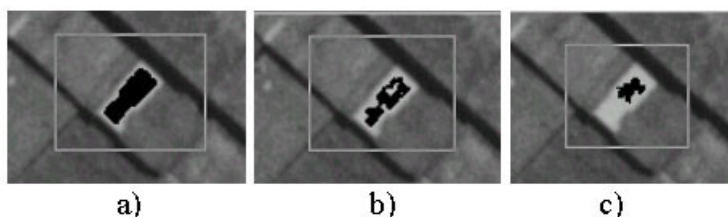
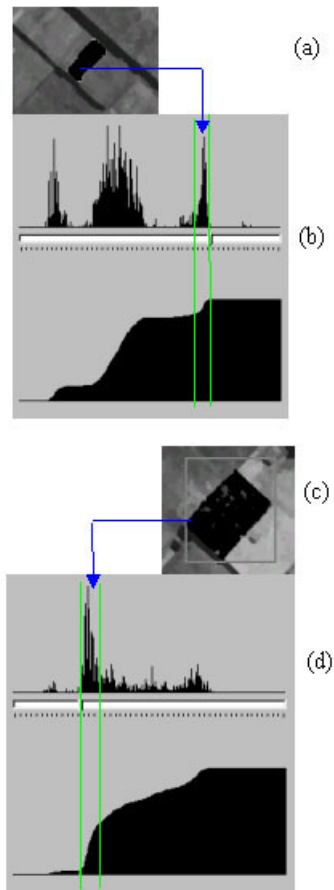


Figura 6 : Resultados da segmentação

Comparando as imagens da figura anterior, pode-se verificar que, nesse caso (imagens do mundo real), quando a amostra da semente é maior e a menor, o resultado dessa segmentação torna-se mais adequado (experimento a), pois, quanto maior a semente, as estatísticas da região inicial serão mais representativas do comportamento do objeto e as regiões vizinhas tem maior chance de serem classificadas como pertencentes ao mesmo objeto.

- **Teste 4:** Segmentação através de limiares

A segmentação através de limiares necessita do histograma da região a ser segmentada, a partir do qual os valores de limiar são fornecidos. A figura 7 mostra esse processo de segmentação aplicado em duas regiões (a) e (c) e seus respectivos histogramas (b) e (d). Cada pixel vizinho da semente será classificado como pertencente ao objeto se seu valor de brilho estiver entre os dois limiares já escolhidos. A configuração para a classificação do objeto em relação aos limiares pode ser mudada para utilizar apenas um limiar, ou, como nesse caso, 2 limiares.

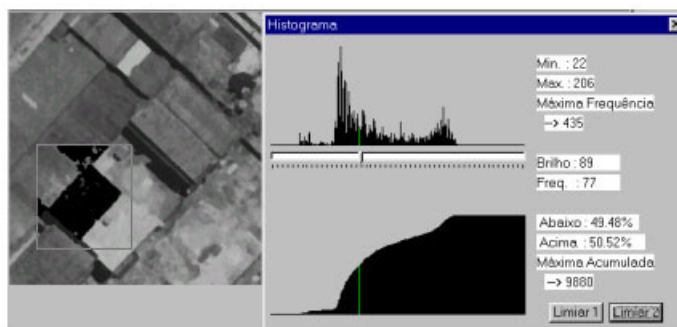


**Figura 7:** Segmentação através de limiares

Na figura 7 os valores de limiares utilizados no item a) são: limite inicial: 163 e limite final: 187, já em c) limite inicial: 64 e final: 83. Pode-se observar que a segmentação no experimento c) não conseguiu classificar todos os pixels do objeto como pertencentes à mesma região. Porém, mesmo quando isso acontece, se as bordas externas ficarem bem definidas, ainda é possível continuar o processo.

• **Teste 5:** Segmentação através de limiares

Neste teste, os valores de limiares utilizados foram: limite inicial 63 e final 89. Como mostra a figura 8, a segmentação ultrapassou os limites do objeto. Neste caso, não é possível o prosseguimento do processo de extração, pois a próxima etapa necessitaria de uma borda coerente para ser perseguida.

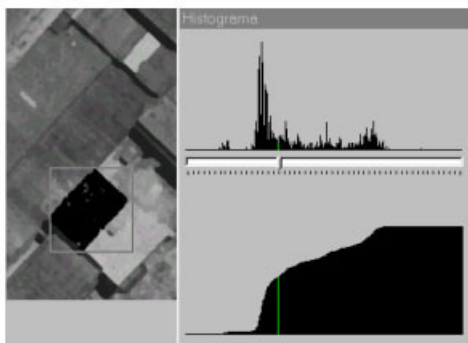


**Figura 8 :** Segmentação com limiares 63 e 89

Existem casos, em que a caixa delimitadora pode solucionar alguns problemas. Por exemplo, quando se sabe que a

segmentação vaza em algum canto do objeto, é conveniente eliminar esse canto problemático da área de processamento. Caso a distância entre os limites (limiares) fosse diminuída, o objeto não seria completamente segmentado, e se fosse aumentada, os limites do objeto seriam ultrapassados. Portanto, apenas alterar os limiares não seria suficiente.

A figura 9 mostra o resultado obtido pela segmentação da mesma área da figura 8, com uma delimitação que eliminou o canto inferior esquerdo (vazamento).

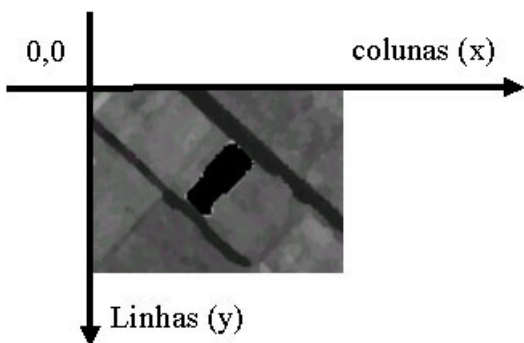


**Figura 9** : Eliminação dos cantos problemáticos

Em todas as segmentações, a região segmentada fica pintada de preto para indicar que essa área foi classificada como pertencente ao objeto.

#### 4.2 Rotinas de Extração

A figura 10 indica o tipo de referencial empregado no processo de extração do SEF.



**Figura 10** : Sistema referencial da imagem

Nesta etapa é gerado um banco de dados, onde ficam armazenadas todas as feições (retas) que constituem o objeto (polígono) detectado. A seguir serão apresentados os resultados obtidos na segunda fase do processo de extração de feições.

Cada registro do banco de feições é composto pelos campos *pol*, *id*, *xi*, *yi*, *xf*, *yf*, *np*, *a* e *b*, onde :

- pol**: indica o polígono que foi extraído;
- id**: é o número identificador da reta;
- xi**, **yi** e **xf**, **yf**: são as coordenadas correspondentes ao ponto inicial e final da reta;
- np**: número de ponto da reta;
- a** e **b**: parâmetro da equação da reta ( $y=ax+b$ );

- **Teste 1:**

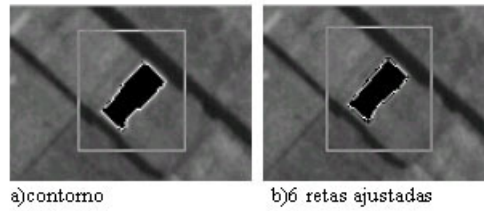
Esse teste foi efetuado sobre a imagem mostrada na figura 11 b), que não foi suavizada. Foi utilizado o processo de segmentação por crescimento de região, critério de similaridade "média e desvio padrão", com máscara de tamanho 15x15 e  $k=2$ .

A tabela 1 mostra o banco de feições gerado para o objeto segmentado da figura 11 b).

**Tabela 1 - Banco de feições**

pol	id	xi	yi	xf	yf	np	a	b
1	1	108	53	119	62	12	0.774725274725275	-29.9340659340659
1	2	120	61	123	55	7	-0.523809523809524	151.619047619048
1	3	124	54	136	44	13	-0.83956043956044	158.562637362637
1	4	137	43	141	38	7	-0.658008658008658	165.402597402597
1	5	140	37	130	28	11	0.828671328671329	-79.539627039627
1	6	129	29	110	51	23	-0.828063241106719	152.252964426877

A figura 11 mostra uma seqüência de processamento. O item a) representa o resultado da execução da rotina de inundação de contorno para classificar os "pixels", que pertencem ao objeto segmentado, que correspondem aos elementos de borda. Em b) foi realizado o ajustamento de retas em relação aos pixels de contorno.



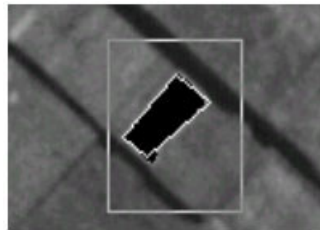
a)contorno b)6 retas ajustadas

**Figura 11** : Resultado da extração inicial

Após a etapa de extração, ainda é possível refinar os resultados. Caso seja útil a eliminação das retas com menor número de pontos, o refinamento pode ser configurado para considerar apenas retas com uma quantidade mínima de pontos. Nesse caso, mínimo de 8 pontos (np) e então o refinamento deve ser executado. A tabela 2 e a figura 12 mostram o resultado da plotagem das retas extraídas após o refinamento com os parâmetros acima.

**Tabela 2 - Banco de feições após refinamento**

pol	id	xi	yi	xf	yf	np	a	b
1	1	107	53	117	61	12	0,774725274725275	-29,9340659340659
1	3	117	61	143	39	13	-0,83956043956044	158,562637362637
1	5	143	39	129	28	11	0,828671328671329	-79,539627039627
1	6	129	28	107	53	23	-0,828063241106719	152,252964426877



**Figura 12** : Retas extraídas após refinamento

Nesse caso é possível perceber que todo cuidado deve ser tomado ao modificar os parâmetros do refinamento, pois nem sempre as retas geradas levam à melhor definição do objeto.

• **Teste 2:**

Esse teste foi efetuado sobre uma imagem suavizada e segmentada pelo algoritmo de crescimento de região através de dois limiares (limite inicial: 63 e limite final: 85). Pode-se observar que a caixa delimitadora da área de processamento eliminou um pequeno pedaço do canto inferior esquerdo do objeto, que proporcionava problemas de segmentação. A figura 13 e a tabela 3 mostram os resultados dessa extração.



**Figura 13** : Exemplo de extração

**Tabela 3 - Resultados iniciais do processo de extração**

pol	id	xi	yi	xf	yf	np	a	b
1	1	40	196	42	203	8	0,33	-26,00
1	2	43	204	70	228	29	0,89	165,23
1	3	71	227	85	210	18	-0,85	263,95
1	4	85	209	87	207	4	-0,26	230,62
1	5	88	208	90	206	4	-0,53	198,88
1	6	90	205	108	186	22	-0,94	284,15
1	7	108	185	110	179	7	-0,14	135,43
1	8	109	178	81	152	29	0,92	77,77
1	9	80	153	77	158	6	-0,76	197,41
1	10	76	158	54	179	25	-1,02	235,53
1	11	53	180	42	194	15	-0,80	197,87

Nesse caso onze retas foram geradas para representar o objeto da figura 13. Analisando a tabela 3 observa-se que a quarta e quinta reta (id) possuem número de pontos (np) igual a quatro, na nona reta "np" vale seis e na primeira oito. Como existe uma grande diferença entre o número de pontos dessas retas e das outras, pode-se tentar aprimorar esse resultado pela execução do processo de refinamento, alterando o mínimo de pontos necessários para 9. Na figura 14 e tabela 4 tem-se os resultados obtidos com esse refinamento.



**Figura 14** : Mínimo de pontos igual a nove

**Tabela 4 - Resultados do refinamento**

pol	id	xi	yi	xf	yf	np	a	b
1	2	38	199	70	228	29	0,89	165,23
1	3	70	228	112	182	40	-0,92	280,33
1	8	112	182	81	153	29	0,92	77,77
1	10	81	153	52	182	25	-1,02	235,53
1	11	52	182	38	199	15	-0,80	197,87

Observando a figura 14, verifica-se que as retas extraídas são uma boa representação do objeto em questão, porém ainda existem cinco retas aos invés de apenas quatro (retângulo).

A partir dos dados da tabela 4 pode-se refinar o processo novamente e, nesse caso, o número mínimo de pontos será alterado para pelo menos 16.



**Figura 15** : Mínimo de pontos igual a dezesseis

**Tabela 5 - Resultados do segundo refinamento**

pol	id	xi	yi	xf	yf	np	a	b
1	2	36	197	70	228	29	0,89	165,23
1	3	70	228	112	182	40	-0,92	280,33
1	8	112	182	81	153	29	0,92	77,77
1	10	81	153	36	197	25	-1,02	235,53

Após o refinamento com os parâmetros acima, a figura 15 e a tabela 5 mostram que a reta 11 foi eliminada, porém esse tipo de ajuste pode, em alguns casos, gerar uma deformação no objeto devido à eliminação de elementos importantes. Uma sugestão seria analisar os coeficientes angulares (a) de todas retas na mesma forma de parametrização, na seqüência de extração.

É possível a realização de vários processos de refinamento, podendo chegar a vários tipos de resultados, cada qual com a precisão necessária, em termos de ajustamento das retas.

## 5. Conclusões e Recomendações

A abordagem semi-automática permite que problemas geralmente encontrados no processamento totalmente automático (global) sejam reduzidos a partir de uma análise local. Com a evolução dessas técnicas é possível se chegar a uma ferramenta comercial para extração assistida de feições cartográficas.

Neste trabalho a interface com o usuário foi desenvolvida de forma que o protótipo ficasse maleável tanto com forma de utilização das ferramentas disponíveis quanto com a escolha da imagem e áreas a serem processadas. Isso facilitou a interação do usuário, que é um fator predominante da abordagem semi-automática, além de questões como a facilidade de uso e aprendizagem do sistema.

Através dos experimentos observou-se que o sucesso do processo depende fortemente de um adequado processo de segmentação e, portanto, este foi alvo de grande atenção, através do desenvolvimento de várias técnicas de segmentação para melhorar o desempenho final do processo de extração.

A implementação do banco de feições permitiu, a partir da análise dos resultados, a criação de rotinas de refinamento (Pós-Processamento), as quais tornaram o processo mais robusto. Dessa forma, os dados de cada feição armazenados podem ser pós-processados até se chegar ao resultado desejado.

Com o estudo, implementação e aperfeiçoamento destes tipos de técnicas, é possível um aumento da qualidade das ferramentas de apoio em sistemas CADs e SIGs.

## 6. Referências Bibliográficas

1. **CRÓSTA, Alvaro P.** Processamento digital de imagens de Sensoriamento Remoto. Campinas: IG/UNICAMP, 1992. 170p.
2. **EKESTRON, M.P.** Digital image processing techniques. London: Academic Press, 1983. 372p.
3. **FIRESTONE, L. et al.** Automated feature extraction: the key to future productivity, *Journal of the American Society for Photogrammetry and Remote Sensing*, Maryland, vol.62, n.6, pp.671-674, 1996.
4. **GATO, H. E. R.** Uma abordagem semi-automática para extração de feições por crescimento de regiões em imagens digitais. Presidente Prudente, 2000. 91p. Dissertação (Mestrado em Ciências Cartográficas) - Faculdade de Ciências e Tecnologia da Universidade Estadual Paulista, 2000.
5. **GEMAEL, C.** Introdução ao ajustamento de observações: aplicações geodésicas. Curitiba: Editora UFPR, 1994. 319p.
6. **GOMES, J e VELHO, L.** Computação gráfica: imagem. Rio de Janeiro: Sociedade Brasileira de Matemática, 1994. 421p.
7. **GONZALEZ, R.C., WOODS, R.E.** Digital image processing. New York: Addison Wesley, 1993. 716p.
8. **GÜLCH, E., MÜLLER, H., LÄBE, T., RAGIA, L.** On the performance of semi-automatic building extraction. IN: SYMPOSIUM ISPRS, 1998, Ohio, Proceedings... Ohio: ISPRS, 1998, Commission III.
9. **GÜLCH, Eberhard.** Application of semi-automatic building acquisition. IN: ASCONA WORKSHOP. Automatic Extraction of Man-Made Objects from Aerial and Space Images (II), Basel: A. Grün , 1997.
10. **R. Michael.** Digital image processing of remotely sense data. New York: Academic Press, 1982. 256p.
11. **HUMES, A. F. P. C., MELO, I. S. H., YOSHIDA, L. K. e MARTINS, W. T.** Noções de cálculo numérico. São Paulo: McGraw-Hill, 1984.
12. **R., KASTURI, R., SCHUNCK, B.G.** Machine vision. Singapore: McGraw-Hill International, 1995. 549p.
13. **J., FRULLA, L.** An automated region growing algorithm for segmentation of texture regions in SAR images, *International Journal of Remote Sensing*, Southampton, vol.19, n.18, p.3595-3606, 1998.
14. **MARQUES FILHO, O., VIEIRA NETO, H.** Processamento digital de imagens. Rio de Janeiro: Brasport, 1999. 409p.
15. **MASCARENHAS, N. A., VELASCO, F. R. D.** Processamento de Imagens. 2.ed. São José dos Campos: MCT-INPE, 1989. 271p.
16. **NEWTON, W.** Na approach to the identification of forest in thematic mapper imagery within the context of a change detection system. London, 1993. Tese (Mestrado em Sensoriamento Remoto) – University College London.
17. **E. M. L. M.** Sensoriamento remoto: princípios e aplicações. 2.ed. São Paulo: Edgard Blucher Ltda, 1992. 308p.
18. **PAVLIDIS, T.** Algorithms for graphics and image processing. Rockville: Computer Science Press, 1982. 416p.
19. **PRATT, W. K.** Digital image processing. 2.ed. New York: John Wiley & Sons, 1991. 698p.
20. **PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. e FLANERRY, B. P.** Numerical recipies in C. New York: Cambridge University Press, 1992. 994p.
21. **RICHARDS, John A.** Remote sensing digital Image analysis : an introduction. Berlin : Springer - Verlag, 1986. 281p.
22. **ROSENFELD, A. e KAK, A. C.** Digital picture processing. 2.ed. London: Academic Press., 1982. 349p.
23. **SCHWENGERDT, R. A.** Techniques for image processing and classification in remote sensing. London: Academic Press., 1983. 249p.
24. **SILVA, E. A.** Extração de feições cartográficas de imagens multiespectrais fundidas. São Paulo, 1995. 109p. Tese (Doutorado em Engenharia) - Escola Politécnica da Universidade de São Paulo, 1995.
25. **SPIEGEL, M. R.** Estatística. São Paulo: McGraw-Hill do Brasil, 1985. 454p.
26. **TOMMASELLI, Antonio. Maria. G. e TOSSI, C. L.** Extração de linhas retas em imagens digitais. IN: CONGRESSO BRASILEIRO DE CARTOGRAFIA, 16., 1993, Rio de Janeiro. Anais... Rio de Janeiro: SBC, 1993. v.2, p399-409.
27. **Y. T., VENKATESWAR, V. e CHELLAPA, R.** Edge detection and linear feature extracting using a 2-d random field model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, v.11, n.1.

### Sites Consultados pela Internet

1. **ANTUNES, A.F.B.:** Análise de Textura em Imagens de Radar.  
<http://www.cieg.ufpr.br/textura.htm>
2. **GRUEN, Armin, LI, Haihong.:** Semi-automatic Road Extraction from Digital Images., 1997.  
<http://www.geod.ethz.ch/p02/projects/safe/safe.html>
3. **GÜLCH,E.; MÜller,H.; RAGIA,L.:** Semi-Automatic Building Acquisition, 1998.  
<http://www.ipb.uni-bonn.de/ipb/projects/semi-automatic.html>

