

UMA ABORDAGEM SEMI-AUTOMÁTICA PARA EXTRAÇÃO DE FEIÇÕES CARTOGRÁFICAS

Helien Eda Rodrigues Gato
Nilton Nobuhiro Imai
Antonio Maria Garcia Tommaselli

Universidade Estadual Paulista -Unesp
Faculdade de Ciência e Tecnologia
Programa de Pós-Graduação em Ciências Cartográficas
Rua Roberto Simonsen, 305, C.P. 957, 19.060-900 Presidente Prudente, S.P.
helieng@hotmail.com, [nnimai, tomaseli]@prudente.unesp.br

RESUMO

Este trabalho apresenta o estudo e implementação de uma abordagem semi-automática para extração de feições que representam regiões, a partir de imagens digitais. Os testes foram feitos por meio de um protótipo implementado na linguagem C++ Builder. Esse protótipo oferece um conjunto de ferramentas, baseadas nas técnicas de processamento digital de imagens e computação gráfica, com intuito de otimizar o processo de extração de feições através da semi-automatização de algumas operações manuais. Os resultados práticos obtidos, através de diversas combinações de processamento, são apresentados e discutidos, mostrando que a abordagem semi-automática é viável e robusta.

ABSTRACT

This work presents the study and implementation of a semiautomatic approach for feature extraction that represent areas, from digital images. The tests were made by a prototype that was implemented in the C++ Builder Programming Environment. This prototype offers a group of tools, based on techniques of digital image processing. The practical results obtained, through sundries combination of processing, are presented and discussed, showing that the semiautomatic approach is viable and robust.

Palavras Chaves: Processamento Digital de Imagens, Segmentação, Extração Semi-Automática de Feições.

1. INTRODUÇÃO

A obtenção de informações espaciais a partir de imagens digitais ainda é um processo a ser aprimorado.

Com a evolução da informática, a cada dia se torna mais viável o desenvolvimento de métodos mais eficientes para a extração de feições lineares em imagens digitais. Como os métodos totalmente automatizados ainda não são satisfatórios, pois geram bons resultados somente quando aplicados à imagens controladas, uma alternativa é procurar desenvolver métodos semi-automáticos para a extração de feições.

O método semi-automático (ou assistido) é caracterizado pela interação do operador com o sistema, ou seja, o operador é responsável por tomar decisões.

O presente trabalho tem como objetivo a investigação de uma abordagem semi-automática para extração de feições que representem regiões, baseada nas técnicas de processamento digital de imagens.

2. EXTRAÇÃO DE FEIÇÕES

As principais etapas de um processo de extração de feições podem ser classificadas da seguinte forma:

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

Segmentação: tem a função básica de 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. 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;

Base de conhecimento: não é considerada exatamente uma etapa, mas fornece heurísticas e parâmetros que direcionam o processo computacional. Pode-se integrar a qualquer uma das etapas acima citadas. Um exemplo de caso pode ser dado através de algum parâmetro inicial fornecido pelo usuário, ou heurísticas pertinentes a uma situação específica no processamento.

A figura 1 mostra como estas etapas interagem entre si.

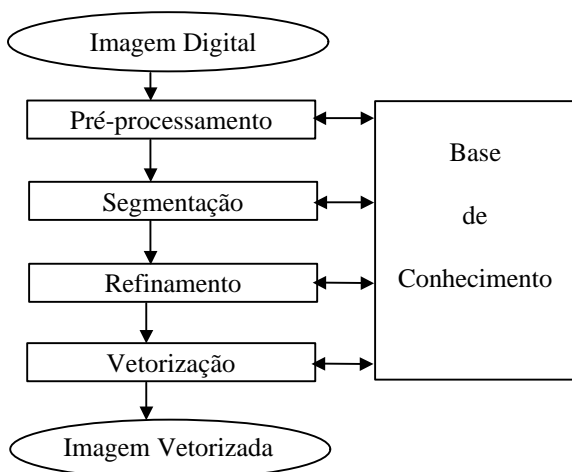


Fig.1 - Etapas do processo de extração de feições. (Adaptado de Gonzalez e Woods, 1993)

2.1 Segmentação

A Segmentação é uma etapa fundamental no processo de extração de feições, pois dela depende o eventual sucesso ou fracasso do mesmo. Vários métodos para segmentação de imagens digitais são descritos em Gonzalez e Woods (1993), Rosenfeld e Kak (1982), Hord (1982), Mascarenhas e Velasco (1989), Pavlidis (1982) e podem ser empregados no desenvolvimento de ferramentas para extração de feições.

Os algoritmos de segmentação para imagens monocromáticas geralmente são baseados em duas propriedades básicas dos valores de níveis de cinza, que são as discontinuidades e as similaridades (Gonzalez e

Woods, 1993). A discontinuidade é responsável por particionar uma imagem com base na informação de mudança abrupta do nível de cinza ou textura, sendo normalmente utilizada na detecção de pontos isolados, linhas e bordas. Por outro lado, os algoritmos baseados em limiares, algoritmos de crescimento de regiões, subdivisão e união (“splitting and merging”) tentam agrupar os pixels de acordo com alguma medida de similaridade.

2.1.1 Crescimento de região

Enquanto os métodos de detecção de borda utilizam a diferença de valores de pixel, ou seja, discontinuidades no nível de cinza da imagem, algoritmos como o de crescimento de regiões realizam uma busca por grupos de pixels com alguma propriedade de similaridade, caracterizando os pixels de uma área com brilho similar como pertencentes a 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 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.

2.1.2 Segmentação através de nível de brilho médio

Um dos critérios de uniformidade é baseado na 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 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).

2.1.3 Outros critérios de uniformidade

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.

2.1.4 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 “*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 (α) 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 (α) 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.

2.1.5 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

As técnicas de segmentação produzem dados na forma de limites (bordas) e regiões. A etapa de refinamento engloba todos os outros processos necessários para concluir a extração (detecção de contorno, afinamento, rotulação, ajustamento). 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 Inundação de contorno

A inundação de contorno é um algoritmo recursivo que analisa as extremidades da região segmentada por um processo de inundação (seed-fill). 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 pertencem ao contorno (borda).

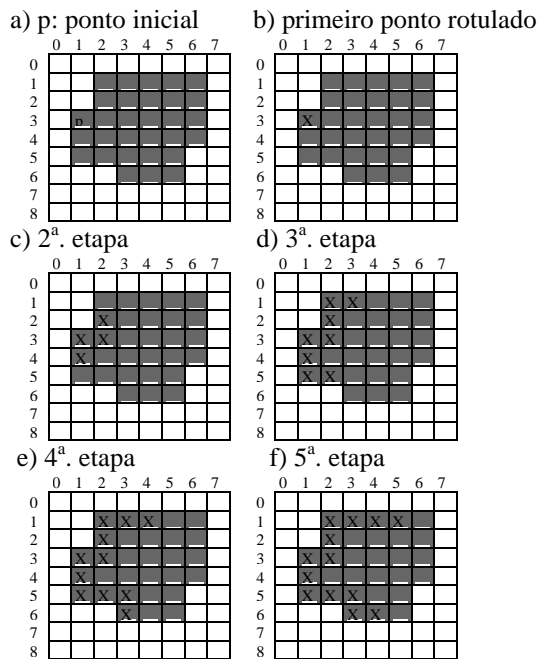


Fig. 2 – Etapas da rotina de inundação de contorno. (Fonte: Gato, 2000)

Observa-se, na figura 2, 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.

2.2.2 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.

2.2.2.1 Traçado de um único contorno

Os pixels de um contorno podem ser percorridos por um caminho, e sempre é possível escolher um caminho fechado para a travessia. O algoritmo TRAÇAR (Pavlidis, 1982), mostra uma forma particular de travessia. Ele usa o conceito de 8-vizinho com a notação mostrada na figura 3 para indicar a próxima direção a ser percorrida em relação ao pixel P .

3	2	1
4	P	0
5	6	7

Fig. 3 - Direção de travessia. (Adaptado de Pavlidis, 1982)

O algoritmo pode ser descrito como um observador que caminha ao longo de pixels que pertencem ao conjunto e seleciona o pixel disponível mais à direita. O pixel inicial A pode ser encontrado de vários modos, inclusive através de uma varredura na imagem de cima-para-baixo, da esquerda-para-direita. O traçado termina quando o pixel atual for igual ao pixel inicial. Considerando que essa também é a condição inicial, criou-se o rótulo “início” para distinguir o começo do retorno ao ponto inicial (fim).

O algoritmo TRAÇAR adota a seguinte notação:

- A é o ponto de partida do contorno da região R ;
- C é o ponto atual cuja vizinhança é examinada;
- S indica a direção de procura em termos do código da figura 3;
- B indica o elemento a ser investigado;
- *início* é o flag que é Verdade apenas quando o traçado começa; e
- *achou* é o flag que é Verdade quando um próximo ponto (elemento) no contorno é encontrado;
- Observação: os comentários sobre o algoritmo estão entre chaves “{}”.

Início do Algoritmo

```

A ← um ponto no contorno; {ponto de partida}
C ← A; S ← 6; início ← Verdade;
Enquanto C ≠ A ou início = Verdade faça
  Início
  achou ← falso;
  B ← S - 1; {o vizinho (S - 1) de C}
  Enquanto achou = falso faça
    Início
    Se B ∈ R
      então Início
        C ← B; S ← S - 2; achou ← Verdade;
      Fim
    senão Início
      B ← S; {o vizinho S de C}
      Se B ∈ R

```

```

então Início
  C ← B; achou ← Verdade;
  Fim
senão Início
  B ← S + 1; {o vizinho (S+1) de C}
  Se B ∈ R
  então Início
    C ← B; achou ← Verdade;
    Fim
  senão S ← S + 2;
  Fim
Fim
Fim
início ← falso;
Fim
Fim do Algoritmo.

```

Fig. 4 - Algoritmo TRAÇAR. (Adaptado de Pavlidis, 1982).

A figura 5 mostra uma pequena simulação do algoritmo descrito na figura 4.

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. Na verdade, no presente trabalho, esse processo encontra-se acoplado ao algoritmo de perseguição de contorno.

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 perseguido, 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).

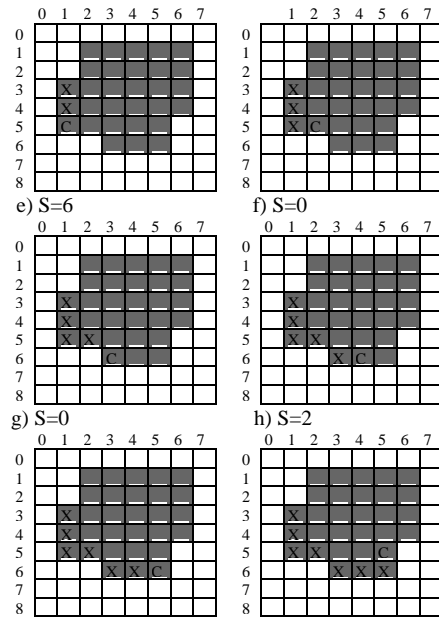
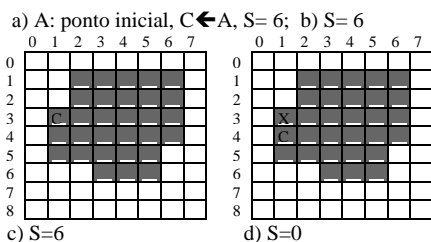


Fig. 5 – Exemplo de perseguição de contorno.

2.3.2 Ajustamento pelo Método dos Mínimos Quadrados

Tendo-se uma seqüência de pontos $S_{ij} = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, originados a partir de uma função desconhecida f , deseja-se determinar uma equação da reta g , que melhor se aproxima de f . Como a função aproximada g nem sempre é idêntica à f , tem-se a introdução de um erro ou resíduo (v_i) a cada elemento conhecido. Porém, para que g seja a melhor aproximação de f , é necessário que os resíduos (v_i) sejam minimizados. A equação (8) tem se mostrado adequada para a minimização dos resíduos:

$$\sum_{i=1}^n v_i^2 = \text{minimo} \quad (8)$$

De todas as funções que se ajustam ao conjunto de pontos S_{ij} , a que tem a propriedade de apresentar o

mínimo valor de $\sum_{i=1}^n v_i^2$ é definida como a melhor

função de ajustamento (Spiegel, 1985), isto é, ela ajusta os pontos com o critério dos mínimos quadrados. O método dos mínimos quadrados é amplamente utilizado nos problemas de ajustamento em diversas áreas de pesquisa. Considere a figura 6, na qual a seqüência S_{ij} de pontos está representada por $((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6))$. Observe os resíduos v_1, v_2, v_3, v_4, v_5 e v_6 formados pela diferença existente entre y_i e os valores correspondentes determinados na reta r .

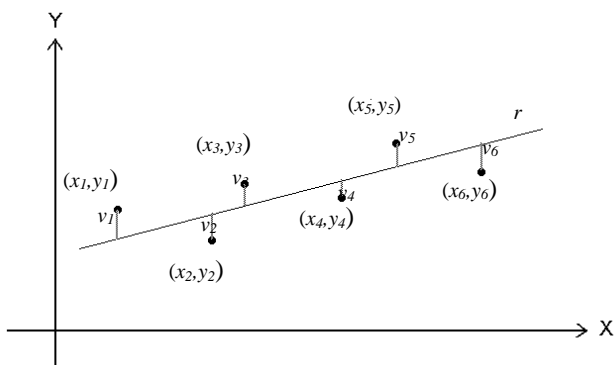


Fig. 6 - Reta r ajustada.

Neste trabalho, deseja-se obter as equações das retas que melhor definam o contorno de um objeto. Assim, a partir da equação da reta:

$$y = ax + b \quad (9)$$

onde são conhecidos os valores de x e y , deseja-se obter os valores de a e b que minimizam os quadrados dos resíduos em (8), ou seja,

$$\sum_{i=1}^n v_i^2 = \sum_{i=1}^n (y_i - ax_i - b)^2 \quad (10)$$

Este problema é caso particular de ajustamento conhecido por Regressão Linear (Humes et al., 1984). A solução do sistema é dada, de forma simplificada, por:

$$a = \frac{\sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{N}}{\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{N}} \quad (11)$$

e

$$b = \frac{\sum_{i=1}^n y_i}{N} - a \frac{\sum_{i=1}^n x_i}{N} \quad (12)$$

Observa-se que essa parametrização apresenta alguns problemas quando as retas são verticais, pois nessas situações, o denominador em (11) se anula. Uma forma de solucionar esse problema é através da utilização de duas parametrizações distintas, uma adequada ao modelo horizontal e a outra adaptada ao modelo vertical. Se no modelo horizontal, x é a variável independente e y é a dependente, no modelo vertical, x será a variável dependente. Deve-se considerar que a definição será modificada, pois agora os resíduos horizontais em vez dos verticais serão considerados. Isso corresponde a uma troca entre os eixos dos x e dos y . Essas duas parametrizações conduzem a retas de mínimos quadrados distintas.

3. PROTÓTIPO PARA EXTRAÇÃO DE FEIÇÕES POLIGONAIS

O protótipo foi desenvolvido através do ambiente C++ Builder, permitindo 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 e visualizar diversas imagens em tons de cinza.

O protótipo oferece uma variedade de ferramentas para manipulação das imagens, destinadas ao pré-processamento, análise do usuário, configuração do "software", fornecimento de heurísticas e as ferramentas básicas para a realização da extração de feições. A geração de uma interface amigável e didática permite que o usuário abra a imagem a ser processada, escolha a opção de processamento desejada, determine o objeto a ser processado e visualize o resultado obtido. Isso permite a análise das etapas que compõem o processo de extração, ou seja, é possível realizar uma etapa apenas ou escolher uma combinação delas até chegar à realização total do processo.

A figura 7 mostra a interface do protótipo de extração de feições "SEF".



Fig. 7 - 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

Os testes foram realizados tanto com imagens padrão, geradas pelo programa “PaintBrush”, como em imagens reais. O tipo padrão possui figuras geométricas de vários formatos e posições. As reais correspondem a fotos aéreas da área urbana de uma cidade, na escala 1:8000. A figura 8 demonstra os dois casos.

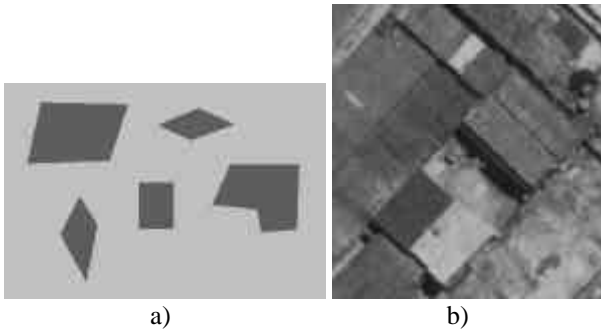


Fig. 8 - Exemplo de imagens: a) Padrão e b) Real.

4.1 Segmentação por Crescimento de Regiões

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

A figura 9 exemplifica o modelo de atuação do processamento, além do resultado do teste 1 quando aplicada a uma imagem padrão. 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”.

•Parâmetros: Máscara: 3x3 (representa a amostra da região definida pela máscara de 3 pixels de largura por 3 de comprimento) e $k=2$, onde k representa o número de desvios padrões ($\hat{\sigma}_{\hat{S}}$) aceitos em torno do valor médio estimado (\hat{m}). Os pixels da região são definidos como sendo $\{x | \hat{m} - k\hat{S} \leq x \leq \hat{m} + k\hat{S}\}$.

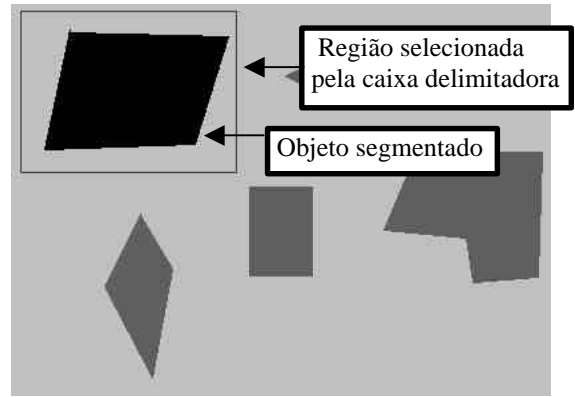


Figura 9 - Segmentação numa imagem padrão. (Teste 1)

•Teste 2: (Variação do tamanho da janela e K fixo)

Utilizou o algoritmo de segmentação por crescimento de região com critério de similaridade “média e desvio padrão” aplicado à imagem real original;

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

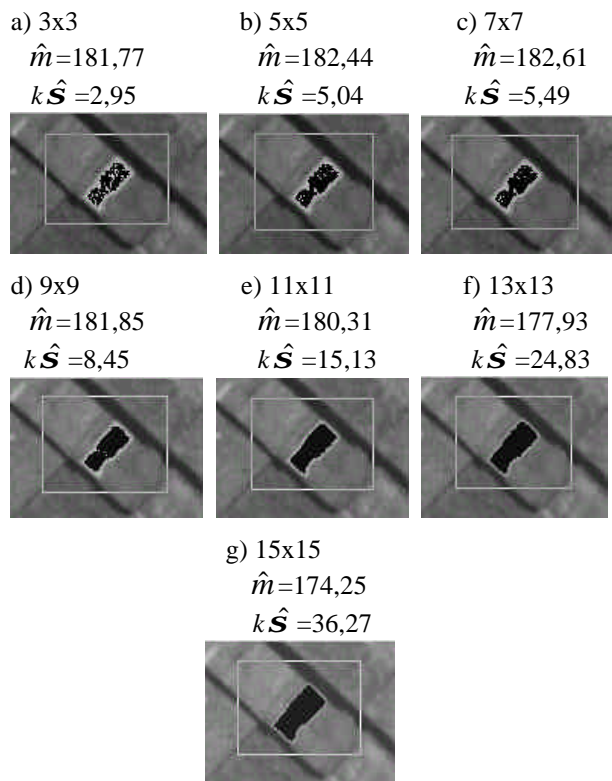


Fig. 10 - Variação no tamanho da máscara e $k=2$ fixo (Teste 2).

Pode-se observar na figura 10, 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 $(k\hat{S})$, 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 10, a máscara de tamanho 11x11 com $k=2$ gerou uma boa segmentação.

•Teste 3: (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 11. O valor médio estimado (\hat{m}) através dos pixels da região definida pela máscara 3x3 é 181,77. Os valores de “ $k\hat{S}$ ” calculados em cada seqüência mostra a medida de dispersão obtida com a variação de k .

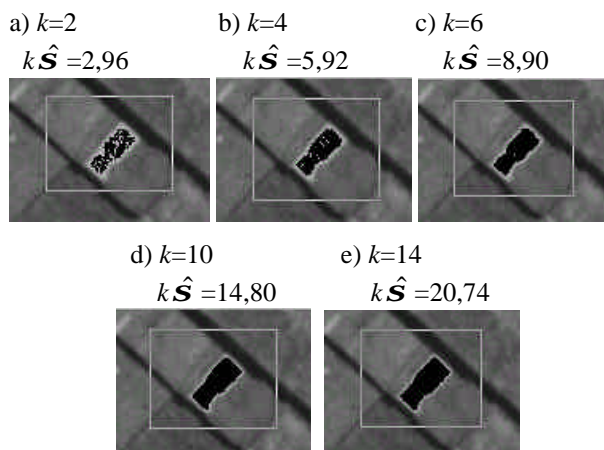


Fig. 11 - Máscara 3x3 fixa com k variando.

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. Observa-se que nesse caso específico (objeto e fundo bastante distintos), o valor adequado de k foi acima de 6.

•Teste 4: (variação da máscara e em k)

Utilizou o algoritmo de segmentação por crescimento de região com critério de similaridade “média e desvio padrão” aplicado a imagem real original;

A Figura 12 mostra os resultados obtidos com diversos tamanhos de máscaras e valores de k .



a) 3x3 b) 9x9 c) 15x15 d) 9x9 e) 3x3
 $k=2;$ $k=2;$ $k=2;$ $k=3;$ $k=6;$

Fig. 12 - Análise de processamento (1).

Conforme os resultados mostrados na figura 12, pode-se verificar que a melhor configuração foi obtida no experimento d). A dificuldade em se obter uma boa segmentação ocorre devido à grande variação nos níveis de cinza do objeto e sua semelhança com certas áreas do fundo. Outro fator que também influencia é o posicionamento da semente. Este ponto inicial, fornecido pelo usuário, define os pixels da amostra, cujos valores de brilho serão usados para calcular os parâmetros, para comparação com o critério de similaridade. Portanto, para se obter uma boa segmentação é necessário que o objeto possua limites definidos e uma certa homogeneidade no seu interior.

•Teste 5:

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 “ α ”.

Os parâmetros usados em a), b) e c) da figura 13 foram:

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

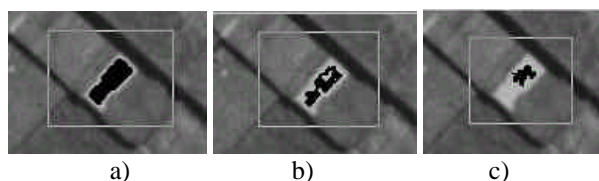


Fig. 13 - Resultados da segmentação.

Comparando as imagens da figura 13, pode-se verificar que, nesse caso (imagens do mundo real), quando a amostra da semente é maior e α 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. Esse critério de similaridade é muito rígido para segmentar imagens reais e sua utilização é mais adequada na segmentação de padrões de textura, pois, uma pequena região semente (inicial) pode conter toda a variação de níveis de cinza necessária para classificar o restante do objeto. A figura 14 mostra os objetos segmentados b) e c) que possuíam, anteriormente, o mesmo padrão de textura do objeto a).

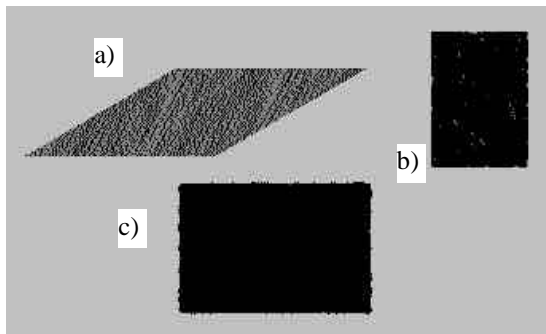


Figura 14 - a) Padrão original, b) Segmentação com $\alpha=10\%$, c) Segmentação com $\alpha=0,5\%$.

Percebe-se que, quando aplicado a padrões de textura, esse critério apresenta-se bem eficiente, segmentando o objeto facilmente por diversos critérios de similaridade (variações em α).

•**Teste 6:**

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 15 mostra esse processo de segmentação aplicado em duas regiões (a) e (c) e seus respectivos histogramas (b) e (d).

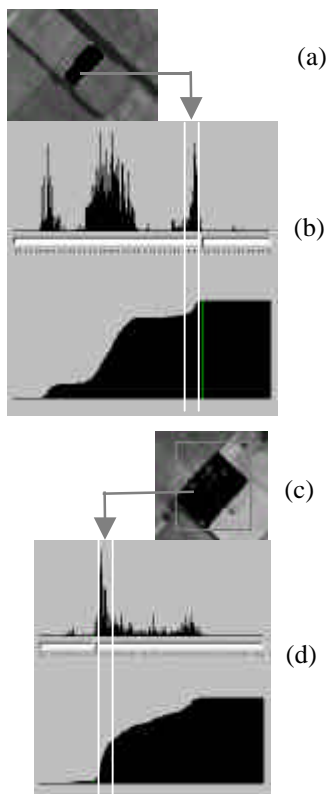


Fig. 15 - Segmentação através de limiares.

Na figura 15 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. Os valores de

limiares utilizados no item a) são: limite inicial: 163 e limite final: 187, já em b) limite inicial: 64 e final: 83.

Pode-se observar que a segmentação no experimento b) 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.

Na figura 16 a segmentação ultrapassou os limites do objeto. Neste caso, não é possível o prosseguimento do processo de extração, pois a próxima etapa necessária de uma borda coerente para ser perseguida.

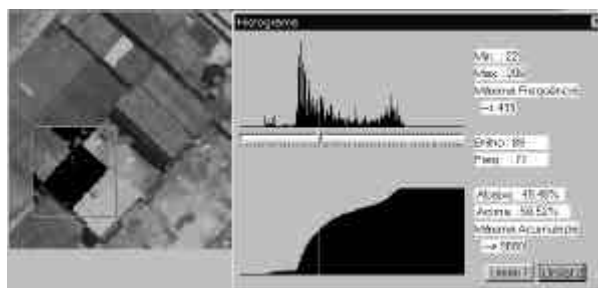


Fig. 16 - Segmentação com limiares 63 e 89.

Existem casos, em que a caixa delimitadora pode solucionar alguns problemas. Por exemplo, quando se sabe que a segmentação vazou 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 17 mostra o resultado obtido pela segmentação da mesma área da figura 16, com uma delimitação que eliminou o canto inferior esquerdo (vazamento).

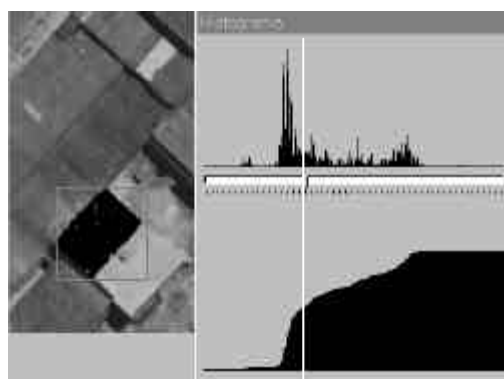


Fig. 17 - Eliminação dos cantos problemáticos.

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

4.2 Testes das Rotinas de Extração

A figura 18 indica o tipo de referencial empregado no processo de extração do SEF.

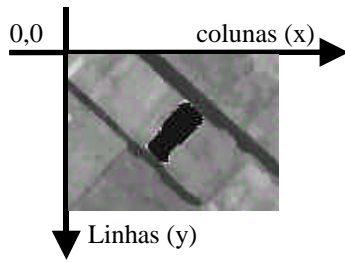


Fig. 18 - Sistema referencial da imagem.

A seguir serão apresentados os resultados obtidos na segunda fase do processo de extração de feições.

•Teste 1

Antes da execução desta rotina, é necessária a criação ou seleção da tabela de banco de dados onde as retas que definem o objeto serão armazenadas. O banco de feições armazena cada reta que compõe o objeto extraído. A tabela 1 mostra o banco de feições gerado para o objeto extraído na figura 19.

TABELA 1 - BANCO DE FEIÇÕES

pol	id	xi	yi	xf	yf	np	a	b	flag_h
1	1	18	145	92	167	76	0.27385662288528	140518744506072	True
1	2	93	168	120	179	28	0.42802881723399	128506911330049	True
1	3	121	179	137	182	17	0.171310629514364	159037435200034	True
1	4	138	181	176	161	41	-0.492684285941731	2492393711192486	True
1	5	176	160	167	137	24	0.393076923076923	112504615384615	False
1	6	166	136	172	113	24	-0.160769230769231	191815384615385	False
1	7	173	112	102	91	22	-0.482213438735178	227136339920949	False
1	8	182	90	176	70	21	0.28728991596637	156511029411766	False
1	9	176	70	39	96	138	-0.165926116652168	893243681048227	True
1	10	38	97	19	143	47	-0.41581868640148	787705827937095	False

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 ($y=ax+b$);
- flag_h: variável booleana que indica o tipo de referência utilizada para o ajustamento da reta, podendo ser horizontal (true) ou vertical (false);

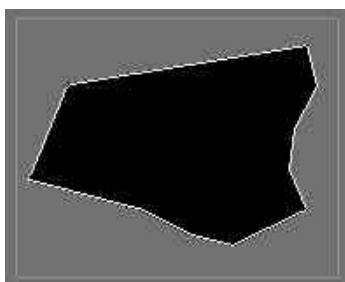


Fig. 19 - Polígono resultante do ajustamento de retas inicial.

O sistema permite um refinamento dos resultados, através da configuração de alguns parâmetros. Neste caso, considerando-se os coeficientes angulares com a diferença entre -0,2 e +0,2 como similares, caso as retas possuam a mesma direção de ajustamento. Na tabela 2 e figura 20 são mostrados os resultados após o refinamento.

TABELA 2 - BANCO DE FEIÇÕES APÓS REFINAMENTO.

pol	id	xi	yi	xf	yf	np	a	b	flag_h
1	1	18	144	135	183	121	0.332747727708233	137755840274502	True
1	4	135	163	176	162	41	-0.462684285941731	2492393711192486	True
1	5	176	162	167	138	24	0.393076923076923	112504615384615	False
1	6	167	138	171	117	24	-0.160769230769231	191815384615385	False
1	7	171	117	103	92	22	-0.482213438735178	227136339920949	False
1	8	183	92	177	70	21	0.28728991596637	156511029411766	False
1	9	177	70	40	93	138	-0.165926116652168	893243681048227	True
1	10	40	93	19	144	47	-0.41581868640148	787705827937095	False

Percebe-se que as retas 1, 2 e 3 foram agrupadas em uma única reta e seus pontos reajustados. A figura 20 mostra a visualização desses resultados.

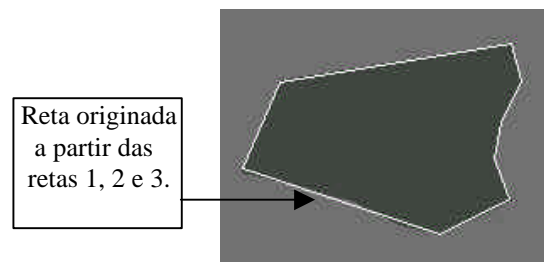


Fig. 20 - Polígono resultante após o refinamento.

•Teste 2:

Esse teste foi efetuado sobre uma imagem real original (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.

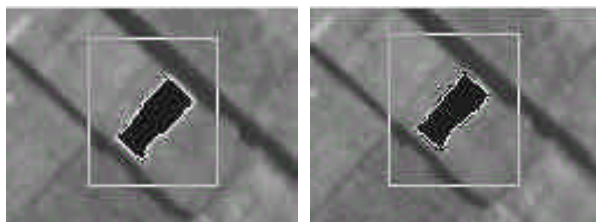
A tabela 3 mostra o banco de feições gerado para o objeto extraído da figura 21 b).

TABELA 3 - BANCO DE FEIÇÕES.

pol	id	xi	yi	xf	yf	np	a	b	flag_h
1	1	108	53	119	62	12	0.77425274725275	-29.0140659340659	True
1	2	120	61	123	55	7	-0.523809523809524	161.619047619048	False
1	3	124	54	136	44	13	-0.83956043956044	158.562637362637	True
1	4	137	41	141	38	7	-0.6580865808658	166.402597402597	False
1	5	140	37	130	28	11	0.828671328671329	-79.529627029627	True
1	6	129	39	110	51	21	-0.820063241106719	162.262964426297	False

A figura 21 mostra uma sequência 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

Fig. 21 - 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 4 e a figura 22 mostram o resultado da plotagem das retas extraídas após o refinamento com os parâmetros acima.

TABELA 4 - BANCO DE FEIÇÕES APÓS REFINAMENTO.

pol	id	xi	yi	xf	yf	np	a	b	flag_h
1	1	107	53	117	61	12	0,774725274720275	-29,0340659340059	True
1	3	117	61	143	39	13	-0,83966040856044	168,562637362637	True
1	5	143	39	129	29	11	0,028671320671329	-79,539627639627	True
1	6	129	29	107	53	23	-0,8288063241196719	152,252964426877	False

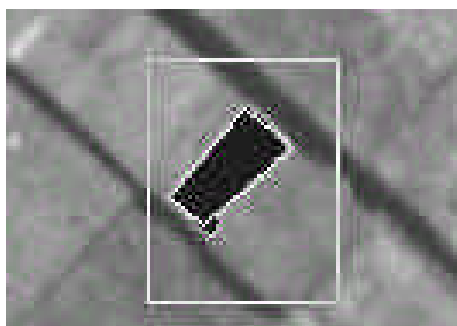


Fig. 22 - Retas extraídas após refinamento.

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 3:

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 23 e a tabela 5 mostram os resultados dessa extração.



Fig. 23 - Exemplo de extração.

TABELA 5 - RESULTADOS INICIAIS DO PROCESSO DE EXTRAÇÃO.

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

Nesse caso onze retas foram geradas para representar o objeto da figura 23. Analisando a tabela 5 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 24 e tabela 6 tem-se os resultados obtidos com esse refinamento.

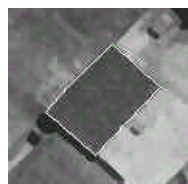


Fig. 24 - Mínimo de pontos igual a nove.

TABELA 6 - RESULTADOS DO REFINAMENTO.

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

Observando a figura 24, 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 6 pode-se refinar o processo novamente e, nesse caso, o número mínimo de pontos será alterado para pelo menos 16.

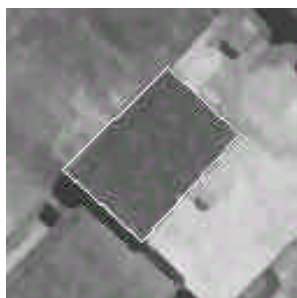


Fig. 25 - Mínimo de pontos igual a dezesseis.

TABELA 7 - RESULTADOS DO SEGUNDO REFINAMENTO.

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

Após o refinamento com os parâmetros acima, a figura 25 e a tabela 7 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 sequência de extração.

Mesmo quando o resultado inicial é satisfatório, a rotina de refinamento deve ser executada, pois dela dependem os valores precisos das coordenadas dos pontos iniciais e finais de cada reta.

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.

O resultado da extração fica armazenado no banco de feições, como mostram as figuras dos bancos de dados gerados. Portanto podem-se utilizar esses dados para uma futura reconstrução utilizando as coordenadas ou os parâmetros. A partir desses resultados, é possível também a geração de arquivos DXF, que poderiam ser utilizados em outros aplicativos, mas para tanto, os vetores devem sofrer um tratamento geométrico adequado.

8 CONCLUSÕES E RECOMENDAÇÕES

A utilização da abordagem semi-automática permitiu que problemas geralmente encontrados no processamento global pudessem ser reduzidos a uma análise local. Com isso algumas etapas do processo manual foram otimizadas, mostrando que essa abordagem é viável e robusta, necessitando de poucos aprimoramentos para se chegar a uma ferramenta comercial.

Através dos experimentos observou-se que a etapa de segmentação é fundamental para definir o sucesso do processo. Quando as bordas do objeto não foram razoavelmente segmentadas, torna-se impossível chegar a um bom resultado na extração das feições. Esta etapa ainda deve ser objeto de pesquisas adicionais.

A forma com que os dados de cada feição foram armazenados, além da rotina de refinamento, tornou a geração dos resultados robusta, de forma que o usuário pode refinar o processo até chegar ao resultado desejado.

A interface de usuário é uma questão importante, pois facilita o desenvolvimento da abordagem semi-automática, em que o usuário deve interagir constantemente no processo.

Essas ferramentas asseguram uma interface mais poderosa e reduzem o tempo para gerar uma base de dados de feições.

REFERÊNCIAS BIBLIOGRÁFICAS

- CRÓSTA, Alvaro P. **Processamento digital de imagens de sensoriamento remoto**. Campinas: IG/UNICAMP, 1992.
- EKESTRON, M. P. **Digital image processing techniques**. London: Academic Press, 1983.
- 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.
- GATO, H. E. R. **Uma abordagem semi-automática para extração de feições por crescimento de regiões em imagens digitais**. Presidente Prudente, 2000. 91p. Dissertação de Mestrado em Ciências Cartográficas. FCT-Unesp, 2000.
- GEMAEL, C. **Introdução ao ajustamento de observações: aplicações geodésicas**. Curitiba: Editora UFPR, 1994.
- GOMES, J e VELHO, L. **Computação gráfica: imagem**. Rio de Janeiro: Sociedade Brasileira de Matemática, 1994. 421p.
- GONZALEZ, R.C., WOODS, R.E. **Digital image processing**. New York: Addison Wesley, 1993.
- 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.
- GÜLCH, E. 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.

- HORD, R. M. **Digital image processing of remotely sense data**. New York: Academic Press, 1982.
- 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.
- JAIN, R., KASTURI, R., SCHUNCK, B.G. **Machine vision**. Singapore: McGraw-Hill International, 1995.
- LIRA, J., FRULLA, L. An automated region growing algorithm for segmentation of texture regions in SAR images, **Intenational Journal of Remote Sensing**, Southampton, vol.19, n.18, p.3595-3606, 1998.
- MARQUES FILHO, O., VIEIRA NETO, H. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.
- MASCARENHAS, N. A., VELASCO, F. R. D. **Processamento de Imagens**. 2.ed. São José dos Campos: MCT-INPE, 1989.
- NEWTON, W. **An 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.
- NOVO, E. M. L. M. **Sensoriamento remoto: princípios e aplicações**. 2.ed. São Paulo: Edgard Blucher Ltda, 1992.
- PAVLIDIS, T. **Algorithms for graphics and image processing**. Rockville: Computer Science Press, 1982.
- PRATT, W. K.. **Digital image processing**. 2.ed. New York: John Wiley & Sons, 1991.
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T. e FLANERRY, B. P. **Numerical recipies in C**. New York: Cambridge University Press, 1992.
- RICHARDS, John A. **Remote sensing digital Image analysis : an introduction**. Berlin : Springer - Verlag, 1986.
- ROSENFELD, A. e KAK, A. C. **Digital picture processing**. 2.ed. London: Academic Press., 1982.
- SCHOWENGERDT, R. A. **Techniques for image processing and classification in remote sensing**. London: Academic Press., 1983.
- 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.
- SPIEGEL, M. R. **Estatística**. São Paulo: McGraw-Hill do Brasil, 1985. 454p.
- TOMMASELLI, A.M.G. e TOZZI, C. L. Extração de linhas retas em imagens digitais. IN: **Congresso Brasileiro de Cartografia**, 16., 1993, Rio de Janeiro, 1993. v.2, p399-409.
- ZHOU, Y. T., VENKATESWAR, V. e CHELLAPA, R. Edge detection and linear feature extracting using a 2-d random field model, **IEEE Transactions on PAMI**, 1989, v.11, n.1.

SITES CONSULTADOS PELA INTERNET

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