

UM MÉTODO PARA A CONEXÃO E APROXIMAÇÃO POLIGONAL DE ELEMENTOS DE BORDAS EM IMAGENS DIGITAIS

*A method for edge pixel connection and straight-line segmentation
in digital images*

ALMIR OLIVETTE ARTERO
ANTONIO MARIA GARCIA TOMMASELLI

Universidade Estadual Paulista - Unesp
Faculdade de Ciências e Tecnologia
Departamento de Cartografia

Rua Roberto Simonsen, 305, 19.060-900 Presidente Prudente, S.P.
e-mail: almir@icmc.sc.usp.br , tomaseli@prudente.unesp.br

RESUMO

Este trabalho apresenta um método para a conexão de pixels de borda, com o objetivo de segmentar as retas definidas pelas discontinuidades nos tons de cinza em imagens digitais. Este método é baseado na técnica de Varredura e Rotulação, em relação à qual foram propostos vários passos adicionais para a efetiva extração de feições retas. O método também possui uma alternativa baseada no processo de perseguição de contorno. Todas as modificações foram desenvolvidas após estudos sobre imagens aéreas reais, sendo apresentados resultados obtidos com dados sintéticos e dados reais.

ABSTRACT

In this paper a method for edge pixel connection is presented, aiming at the segmentation of the straight-lines defined by gray level discontinuities in digital images. This proposed method is based on the Scan and Label technique, with was improved with several additional steps aiming at effective straight line extraction. An alternative based on the edge following technique was also implemented. All the improvements were developed after extensive experiments with real aerial images. Some results obtained with synthetic and real data are presented.

1. INTRODUÇÃO

A análise de imagens continua sendo uma tarefa de difícil implementação computacional e o que há disponível é um conjunto de técnicas de processamento de imagens, que podem ser utilizadas para extrair algumas informações sobre os objetos presentes nas cenas. A primeira etapa em um processo de análise de imagens é a segmentação de imagens (Gonzalez, 1993), que consiste em dividir a imagem em seus elementos constituintes.

Embora exista um grande número de técnicas para a segmentação (métodos baseados em classificação, bordas, regiões, formas e métodos híbridos), nenhum deles consegue apresentar bons resultados em todas as situações.

Os métodos baseados em bordas se justificam pelo fato de que elas definem o contorno dos objetos presentes na cena (Gonzalez, 1993). A maioria dos processos de detecção de descontinuidades baseia-se no fato de que tais descontinuidades são, na verdade, uma modificação do nível de cinza no pixel analisado em relação a seus vizinhos. Assim, tal modificação pode ser determinada pela derivada do sinal no pixel.

Os processos tradicionais para a segmentação de imagens digitais por bordas normalmente seguem uma seqüência, tal qual apresentam Paine e Lodwick (1988), que sugerem as cinco seguintes etapas: 1 – Suavização (Smoothing), com o objetivo de reduzir variações exageradas, que produzem bordas falsas na imagem (necessidade da utilização de filtros de suavização, porém, com máxima preservação de bordas); 2 – Detecção de Bordas (Edge Detection) – Consiste na aplicação de um detector de bordas que, normalmente, é baseado na aplicação de operadores de detecção de variações de brilho na imagem; 3 – Limiarização (Thresholding) – utilizada para eliminar algumas bordas detectadas, e que, por apresentarem baixas magnitudes, devem ser desconsideradas, a fim de simplificar o processamento posterior; 4 – Afinamento de bordas (Thinning) – as bordas obtidas precisam ser afinadas para uma melhor definição de sua verdadeira localização; 5 – Ligação (Linking) – Esta etapa deve ser capaz de agrupar todos os pontos pertencentes a cada uma das retas detectadas (pixels de borda).

O processo é direcionado para o trabalho com linhas retas por se tratar de uma feição de grande interesse na análise de imagens aéreas.

2. CONEXÃO DOS PIXELS DE BORDA

Apenas a definição dos pixels de borda não permite uma análise significativa do objeto, sendo importante que estes pixels sejam agrupados, de forma que possam definir objetos mais complexos como linhas retas ou outros objetos (círculos, elipses etc.). Esta tarefa pode ser executada por métodos globais, como a Transformada de Hough ou por métodos locais (com um processamento apenas em uma pequena região em torno do pixel em estudo), com o objetivo de determinar possíveis pixels vizinhos que possuem características parecidas (algumas das características mais utilizadas são a direção e a magnitude do gradiente). Um

método que apresenta bons resultados é o Método de Varredura e Rotulação (Scan & Label) (Venkateswar e Chellapa, 1992) e sobre este método serão apresentadas algumas modificações.

2.1 MÉTODO DA VARREDURA E ROTULAÇÃO

Neste método a imagem é varrida da esquerda para a direita e de cima para baixo, aplicando um valor (rótulo) aos pixels que apresentam características similares (direção e magnitude) (Zhou et al., 1989), (Venkateswar e Chellapa, 1992) e (Tommaselli, 1999). Ao mesmo tempo, uma estrutura de linhas é utilizada para guardar as características das retas que estão sendo definidas (pixel inicial - x_i, y_i , pixel final - x_f, y_f , direção - dir, número de pixels - n, parâmetros a e b da reta - parametrização $y=ax+b$ - e uma variável booleana que guarda informação da inclinação da reta - horizontal/vertical). A vizinhança a ser considerada no momento da rotulação depende da direção do pixel examinado (chamado pixel atual). Caso seja encontrado nas proximidades do pixel atual algum vizinho já rotulado e com as mesmas características, o pixel atual recebe o mesmo rótulo que o seu vizinho.

Figura 1 – Classificação dos pixels durante a conexão.

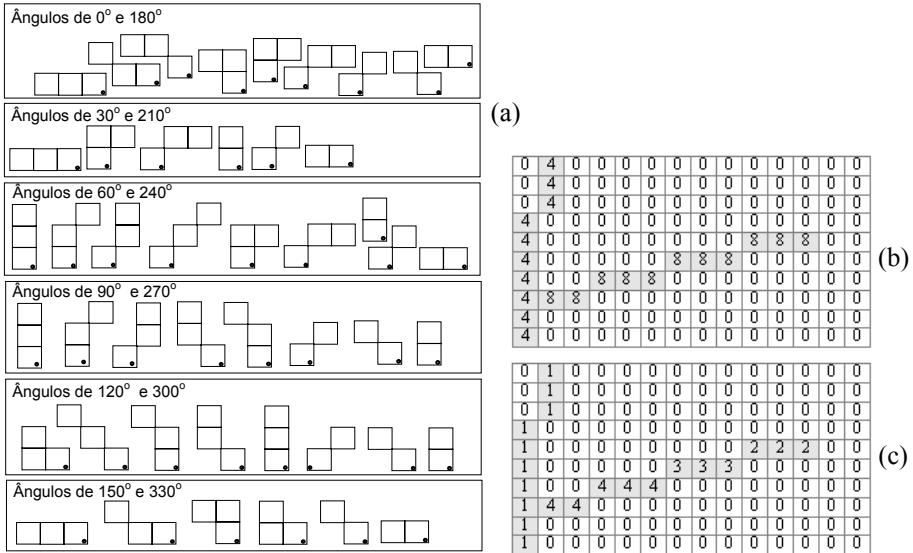


Na figura 1 observa-se que para o pixel em estudo (pixel atual) a vizinhança máxima verificada é a 24. No caso de ser encontrado um pixel na vizinhança 8, com as características do pixel atual, o pixel atual recebe o mesmo rótulo que o seu vizinho (pixel rótulo). Caso contrário, a vizinhança 24 é verificada, e se for encontrado um pixel rótulo nesta região, contendo características similares, o pixel atual recebe o rótulo do pixel rótulo, assim como o pixel ponte, que também recebe as informações de direção (a mesma direção que o pixel rótulo) e magnitude (pode ser a média entre as magnitudes dos pixels atual e rótulo).

A figura 2(a) mostra as vizinhanças a serem verificadas no processo de conexão dos pixels, de acordo com as direções de seus gradientes. Neste método, segmentos com direções entre 0° e 30° não são devidamente conectados em um único segmento, apesar dos pixels terem sido detectados com uma mesma direção (na etapa de detecção), e estarem devidamente conectados em uma vizinhança 8. A figura 2(a) mostra esta situação.

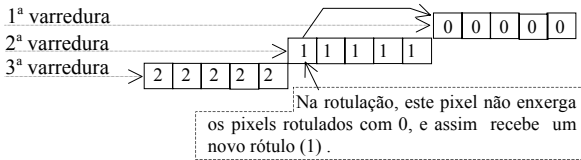
A figura 2(b) mostra um segmento com um conjunto de pixels, que, apesar de possuírem a mesma direção (8), não recebem o mesmo rótulo (figura 2(c)), provocando uma fragmentação da reta.

Figura 2 – a) Vizinhanças verificadas na etapa de conexão (Venkateswar e Chellapa, 1992), (Tommaselli, 1999), b) Direção dos pixels na borda e c) resultado da conexão utilizando o método da varredura e rotulação.



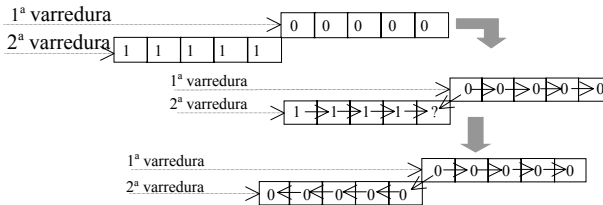
A figura 3 mostra o que acontece durante o processo de varredura e rotulação.

Figura 3 – Processamento realizado durante a Varredura e Rotulação.



Uma possível sugestão para a correção do problema é verificar, antes de fazer a rotulação, se o pixel está sob a ação de dois rótulos vizinhos e, neste caso, herdar o menor deles (na figura 3, o rótulo 0), fazendo uma propagação em todos os demais pixels já rotulados com o rótulo atual (na figura 3, o rótulo 1), de modo que todos recebam o rótulo anterior (na figura 3, o rótulo 0). Desta forma tem-se o procedimento apresentado na figura 4.

Figura 4 – Sugestão para a correção do problema no método da Varredura e Rotulação.



Esta etapa é importante pois previne a criação de um número excessivo de rótulos, embora não seja imprescindível, uma vez que, numa etapa posterior, os segmentos colineares serão reconectados.

2.2 ROTULAÇÃO POR INUNDAÇÃO

Uma outra forma de fazer a conexão consiste em fazer a rotulação por um processo de inundação (seed-fill). Neste caso, todos os pixels vizinhos a um pixel inicial, com características parecidas (magnitudes e direções próximas), recebem o mesmo rótulo, em uma operação equivalente a uma perseguição de bordas (edge following) (Ballard e Brown, 1982) Esta parece ser a forma como as pessoas fazem visualmente a identificação das linhas retas presentes em uma imagem, ou seja, identificam a reta toda no momento em que se dirige a atenção para uma parte da mesma. Uma grande vantagem deste método é que ele não apresenta o problema mostrado na figura 2(c).

2.3 DETERMINAÇÃO DOS PONTOS EXTREMOS DOS SEGMENTOS

Um problema que ocorre com os dois métodos apresentados é a definição dos pixels início e fim dos segmentos. Para o caso do método de varredura e rotulação, verifica-se que a simples estratégia de considerar que o último pixel anexado ao segmento é sempre o pixel final resulta em situações como a que é vista na figura 5(a), quando o que se espera é o que se observa na figura 5(b).

Figura 5 – (a) Definição incorreta dos pontos início e fim (b) resultado obtido com a aplicação do método de correção proposto.

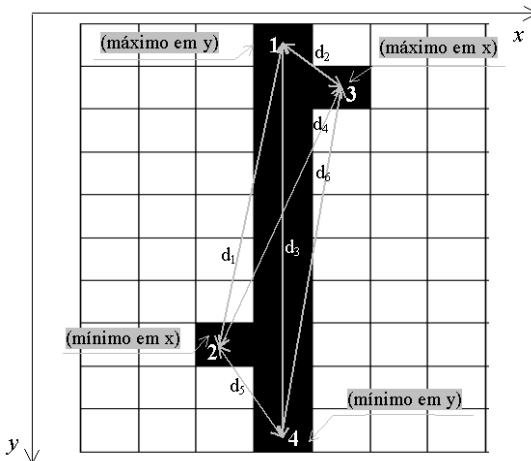


No caso do método de rotulação por inundação a situação é ainda mais crítica, pois durante o processo, a inundação se propaga em todas as direções.

A forma mais eficiente, desenvolvida para resolver este problema é mostrada na figura 6, que supõe que entre todos os pixels do segmento, os vértices são aqueles que apresentam a maior distância entre si.

Inicialmente são definidos os quatro pixels extremos em x e y . Em seguida determinam-se os dois pixels mais distantes ente os quatro possíveis (1, 2, 3 e 4, respectivamente, mínimo em y , mínimo em x , máximo em x e máximo em y). Para o caso da figura 6, especificamente, serão escolhidos os pixels 1 e 4 – mínimo e máximo em y , respectivamente. Na mesma figura, d_1 , d_2 , d_3 , d_4 , d_5 e d_6 são, respectivamente, as distâncias entre os pixels 1-2, 1-3, 1-4, 2-3, 2-4 e 3-4. Este método tem resolvido de forma adequada a determinação dos pixels extremos dos segmentos, de qualquer direção.

Figura 6 - Método usado para determinação dos extremos dos segmentos.



Como a rotina precisa percorrer todos os pixels com o mesmo rótulo, para medir as distâncias entre todos os pixels, pode-se também fazer uma recontagem do número de pixels que possuem o mesmo rótulo, informação importante para etapas posteriores do processo (ajustamento).

2.4 CONEXÃO DE SEGMENTOS COLINEARES

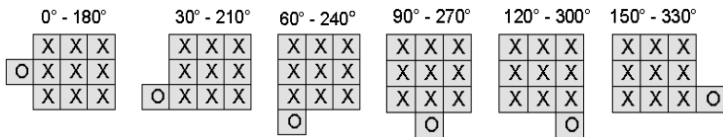
Após a etapa de rotulação, obtém-se um grande número de pequenas retas, que podem ser unidas umas às outras, de forma a formarem segmentos maiores. Uma estratégia que pode ser utilizada para fazer este processamento consiste em investigar as vizinhanças de cada pixel inicial e final de todos os segmentos obtidos

inicialmente e verificar se existe nas proximidades destes, o início ou fim de um outro segmento. Caso ocorra esta situação é preciso verificar ainda se eles possuem uma mesma direção e, neste caso, deve ser feita a união dos dois segmentos, formando um único. Por fim, os atributos do novo segmento precisam ser atualizados, sendo feitas as devidas alterações na estrutura de dados que mantém o controle da localização do início e fim deste segmento, bem como a eliminação dos segmentos que foram unidos. Na prática, o primeiro segmento pode ser atualizado para conter as características do novo segmento, e o segundo segmento deve ter seu rótulo desativado.

Todo este processamento pode ser realizado diretamente na estrutura de dados gerada na etapa anterior, permitindo uma boa performance no processamento, se comparada a uma nova varredura na imagem digital, pois, enquanto a imagem digital pode possuir alguns milhões de pixels, a estrutura deverá ter apenas alguns milhares de vetores.

Novamente, a informação da direção das linhas pode ser utilizada para uma melhoria na forma como as investigações na vizinhança de um pixel final ou inicial devem ser feitas. A figura 7 mostra as vizinhanças que devem ser investigadas, para o caso de uma linha reta com direção α .

Figura 7 – Detalhe das vizinhanças a serem investigadas no processo de conexão de segmentos colineares.

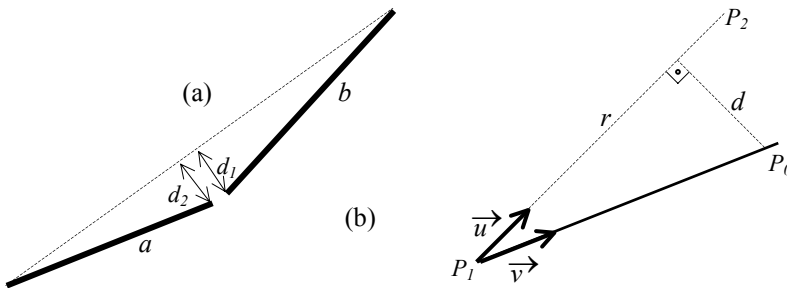


Para cada segmento i , obtém-se na estrutura de linhas a posição dos pixels inicial e final da reta x_i, y_i e x_f, y_f ; verifica-se no espaço da imagem, se na vizinhança existe algum pixel que pertença a um outro segmento j (isto ocorre se o pixel vizinho possui rótulo diferente do pixel atual). É importante também verificar se a reta j possui uma direção próxima à direção da reta i (uma diferença mínima é aceitável nesta etapa). No caso de se encontrar algum pixel nesta situação, o segmento j terá todos os seus pixels rotulados com o valor do rótulo do segmento i . A troca de rótulos é realizada no espaço imagem, e pode ser resolvida de uma forma suficientemente satisfatória, utilizando uma janela limite (bounding box), que contenha os pixels início e fim do segmento j . Por este método todos os pixels pertencentes à menor janela que contém o segmento serão processados. Uma outra alternativa mais eficiente é fazer a troca dos rótulos utilizando um algoritmo recursivo que se desloca dentro do segmento por um processo de inundação (seed-fill). Apesar da recursão ser mais difícil de ser construída, apenas os pixels com o mesmo rótulo são processados.

Um problema que precisa ser contornado nesta etapa é que o algoritmo de detecção de bordas não retorna os mesmos valores de direção para todos os pixels pertencentes a uma mesma borda. Assim, a etapa de conexão de segmentos colineares deve aceitar pequenas diferenças de direção, no momento em que analisa a possibilidade de unir dois segmentos.

Em todo caso, a união de dois segmentos precisa satisfazer uma condição mínima de alinhamento. Um critério que pode ser utilizado para definir em quais situações dois segmentos podem ser unidos é mostrado na figura 8.

Figura 8 – (a) Critério adotado para a conexão dos segmentos colineares, e (b) – Geometria utilizada para definir as distâncias d_1 e d_2 .



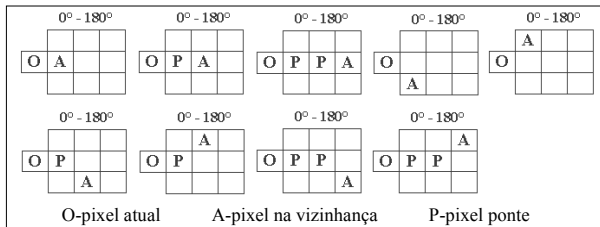
Na figura 8(a), os segmentos a e b devem ser unidos caso as distâncias d_1 e d_2 sejam menores que um fator de qualidade previamente definido. Os valores de d_1 e d_2 são definidos pela distância entre os pixels finais dos segmentos a e b , e a reta definida pelos extremos opostos dos segmentos, sendo estas distâncias obtidas conforme mostra a figura 8(b).

Na figura 8(b), a distância entre o ponto P_0 e a reta r é dada por :

$$d(P_0, r) = \frac{|\vec{u} \wedge \vec{v}|}{|\vec{u}|} \quad (1)$$

Finalmente, é preciso considerar que as máscaras apresentadas na figura 7, utilizadas para verificar a vizinhança dos pixels inicial e final dos segmentos (em busca de outros segmentos nas proximidades), implicam novamente no aparecimento de pixels ponte (pixels que devem ser ligados entre os dois segmentos que estão sendo unidos). No caso da direção igual a 0° e 180° podem ser verificadas as situações apresentadas na figura 10.

Figura 9 – Localização de alguns pixels pontes no processo de conexão de segmentos.



Usando as máscaras mostradas na figura 9, procura-se inicialmente em uma vizinhança mais próxima do pixel inicial (O); em seguida, caso não encontre o início ou final (A) de um outro segmento nas proximidades, desloca-se um pixel, até uma distância máxima de três pixels. Este procedimento permite a conexão de segmentos separados, no máximo, por três pixels. No caso de ocorrer a conexão (o que é possível se for atendida a condição apresentada na figura 8(a)), os pixels P são então anexados ao novo segmento, como pixels ponte, recebendo a direção dos demais pixels, e a magnitude é dada pela média das magnitudes do pixel O e A.

A figura 10 mostra o conjunto de máscaras utilizadas para fazer a conexão dos segmentos com as demais direções, sendo que, em todos os casos deve ser analisada a existência de algum outro segmento com um ângulo próximo (diferença menor ou igual a trinta graus) ao do segmento atual.

A figura 11 mostra como estas máscaras são utilizadas com os respectivos segmentos, sendo que a conexão deve ser verificada para segmentos de mesma direção e também segmentos valores próximos de direção. Nesta figura é possível observar que um segmento com orientação de 150° - 330° não consegue detectar na sua extremidade mais à direita segmentos de 0° - 180° , apesar de ser detectado pelos segmentos de 150° - 330° e 120° - 300° (caso existam). O mesmo ocorre para os segmentos de 0° - 180° , que não detectam segmentos de 150° - 330° e 120° - 300° a sua esquerda. Esta observação explica o resultado obtido na figura 12(a), quando o que se espera é o que se tem na figura 12(b).

Figura 10 – Regiões utilizadas na rotina de conexão de segmentos colineares.

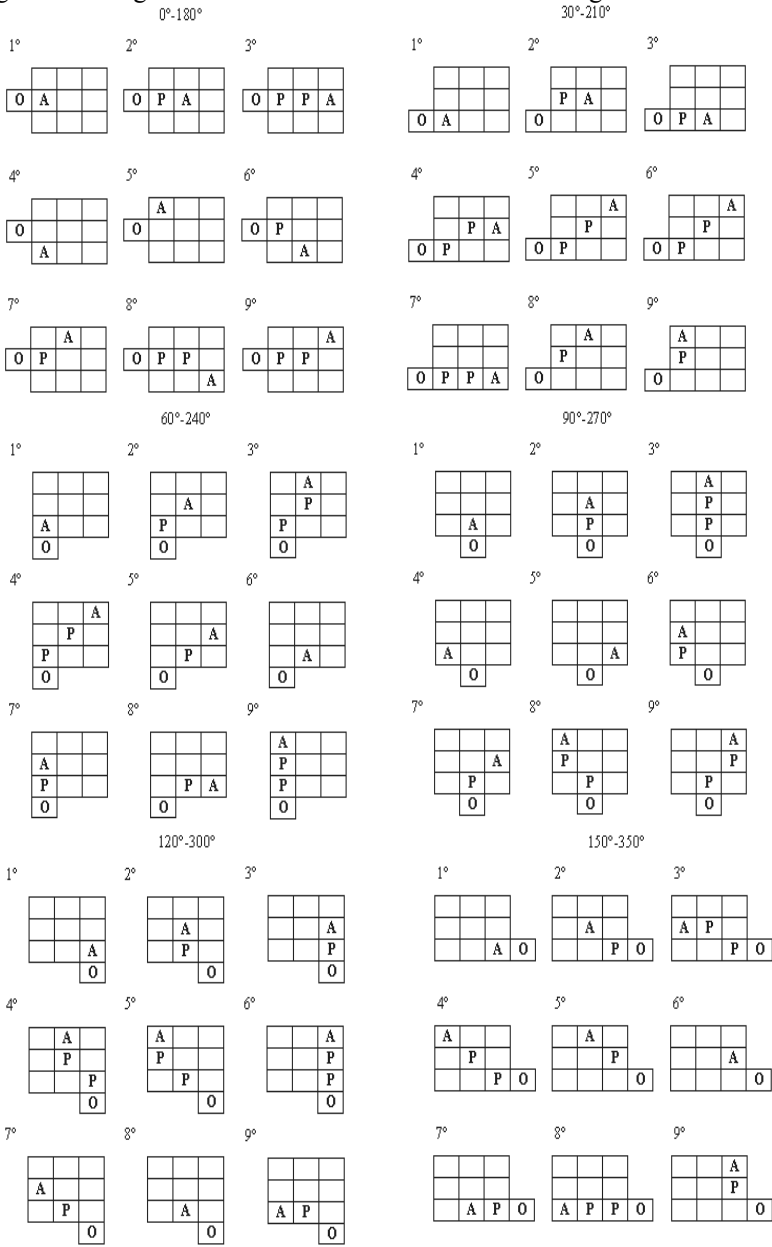


Figura 11 – Relacionamento entre os segmentos na rotina de conexão de segmentos colineares

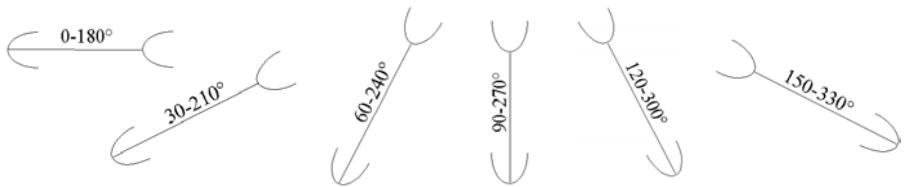
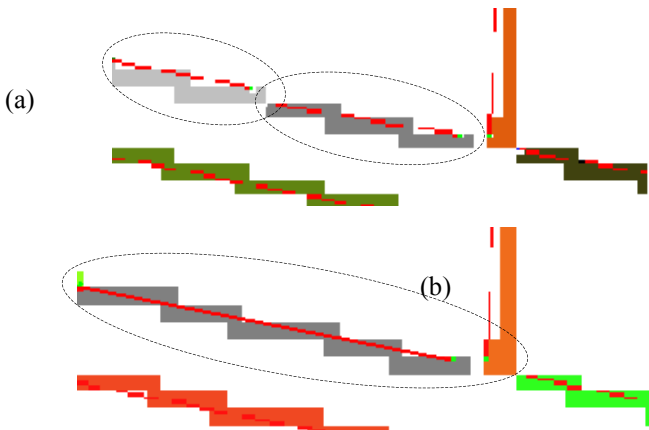


Figura 12 – (a) Vários segmentos colineares não conectados devido a falha apontada na figura 11 e (b) resultado utilizando a máscara da figura 13.



Uma possível solução para esta situação é substituir as máscaras da figura 10 (0° - 180°) pela máscara que aparece na figura 13.

Figura 13 – Máscara para a conexão de segmentos colineares com direção 0° - 180° .

X	X	X		X	X	X
X	X	X	O	X	X	X
X	X	X		X	X	X

2.5 DEFINIÇÃO DA DIREÇÃO DOS SEGMENTOS CONECTADOS

Após a junção de dois segmentos, as informações sobre ele devem ser atualizadas na estrutura de linhas. A quantidade de pixels pertencentes ao segmento obtido é dada pela soma dos pixels pertencentes a cada um dos dois segmentos que foram unidos, além dos pixels ponte (caso existam).

Como dois segmentos com direções próximas (porém diferentes) podem ser unidos, caso seja satisfeita a condição imposta na figura 8(a), o segmento obtido pela união precisa receber uma direção, que deve ser próxima das direções dos segmentos conectados.

Neste caso, é possível adotar para todos os pixels do segmento obtido, a direção global dele, que pode ser definida exclusivamente pelos seus pixels extremos, conforme a equação (2).

A partir dos pixels início e fim do segmento, determina-se o ângulo aproximado do segmento (direção aproximada). O ângulo do segmento é dado pela equação 2.

$$\alpha = \begin{cases} \arctan\left(\frac{y_f - y_i}{x_f - x_i}\right) & \text{se } x_f > x_i \\ \arctan\left(\frac{y_i - y_f}{x_i - x_f}\right) & \text{se } x_i > x_f \end{cases} \quad (2)$$

Em (2) não está sendo considerada a direção da borda (direção do gradiente) nos pixels, mas a direção do segmento, obtida conforme descrito anteriormente, exclusivamente através da localização dos pixels extremos deste. A informação de direção obtida nesta etapa é importante, pois auxilia a definição da parametrização mais adequada a ser utilizada no processo de ajustamento de retas.

Após a determinação dos pixels que pertencem a cada segmento de reta (todos os pixels possuem um mesmo rótulo), pode ser realizado o ajustamento das retas sobre eles.

2.6 ELIMINAÇÃO DE LINHAS INSIGNIFICANTES

Após a realização da rotina de varredura e rotulação, conforme descrita anteriormente, verifica-se a presença de uma grande quantidade de linhas que apresentam um tamanho insignificante (em muitos casos, apenas um pixel).

Por outro lado, pode ocorrer também que estes pequenos segmentos, podem ser reconectados uns aos outros, formando segmentos maiores. Assim, parece mais interessante que, antes de se fazer a eliminação dos segmentos com poucos pixels, se faça a conexão dos segmentos colineares, descrita na seção anterior.

2.7 APROXIMAÇÃO DAS BORDAS POR LINHAS RETAS

A próxima etapa a ser realizada após a identificação dos pixels que pertencem a um dado segmento, é a determinação da equação da reta que melhor se ajusta a estes pixels (ajustamento). O Método dos Mínimos Quadrados (M.M.Q.) pode ser utilizado nesta etapa, apresentando resultados adequados. O método é bem simples e consiste em minimizar o somatório dos quadrados das distâncias entre os pixels e a reta em que são ajustados. Pode-se, ainda, no caso de se obter uma minimização

insuficiente, dividir a reta em outras menores, até que sejam obtidas retas adequadas aos pixels detectados, atendendo a um fator de qualidade previamente estabelecido.

2.7.1 Método dos Mínimos Quadrados (M.M.Q.)

Dada uma seqüência de pontos $S_{i,j} = ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$, obtida por meio de uma função f , desconhecida, deseja-se determinar uma função g , que melhor se aproxima de f .

A tentativa de aproximar uma função f , por uma função g , introduz um erro em cada um dos pontos conhecidos, que normalmente é chamado resíduo v_i . Como o que se procura é obter g , de forma que a mesma seja a melhor aproximação de f , o objetivo a ser alcançado é fazer com que estes resíduos sejam minimizados. Uma primeira abordagem leva então a concluir que o que se deve fazer é obter :

$$\sum_{i=1}^n v_i^2 = \text{minimo} \quad (3)$$

Este critério para a aproximação de duas funções é conhecido como Método dos Mínimos Quadrados M.M.Q., sendo muito utilizado nos problemas de ajustamento, nas mais diversas áreas.

Neste trabalho, deseja-se obter as equações das retas que melhor se ajustam aos conjuntos de pixels rotulados na etapa anterior como pertencentes às bordas. Assim, dada a equação da reta:

$$y = ax + b \quad (4)$$

onde, são conhecidos os valores de x e y , deseja-se obter os valores de a e b .

O caso particular de ajustamento de linhas retas é conhecido por Regressão Linear, sendo os parâmetros a e b obtidos pelas equações (5) e (6).

$$a = \frac{\sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i y_i}{N}}{\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{N}} \quad (5)$$

e

$$b = \frac{\sum_{i=1}^n y_i}{N} - a \frac{\sum_{i=1}^n x_i}{N} \quad (6)$$

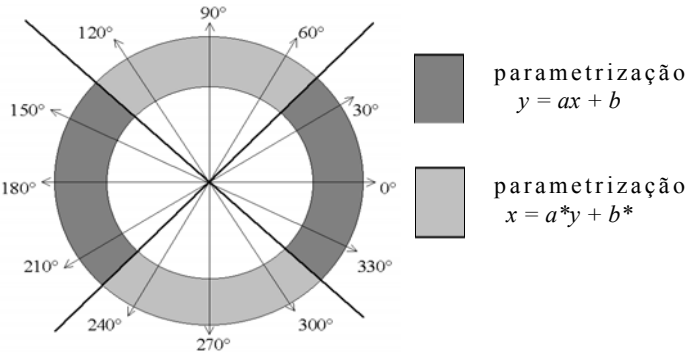
A parametrização utilizada em (4) não pode ser utilizada para representar linhas verticais ($a \rightarrow \pm\infty$). Uma solução que pode ser adotada para evitar o problema com as retas verticais é utilizar duas parametrizações diferentes, cada uma adequada

as retas apresentadas. A sugestão é utilizar para segmentos aproximadamente horizontais (onde ocorre uma boa variação na coordenada x dos pixels) ou com pequena inclinação, a equação (4). No caso de segmentos aproximadamente verticais ou com grande inclinação, uma nova parametrização (7) pode ser utilizada (Tommaselli e Tozzi, 1996).

$$x = a*y + b^* \quad (7)$$

A figura 14 mostra as regiões onde utilizar cada uma destas parametrizações para representar as retas.

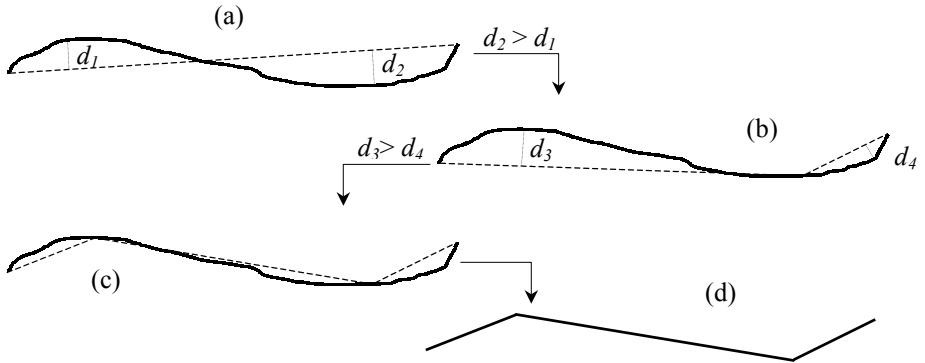
Figura 14 – Parametrizações usadas em cada região.



2.8 APROXIMAÇÃO DE POLIGONAIS

Nos processos posteriores à conexão, dependendo da curvatura dos segmentos detectados, o ajustamento não deverá funcionar adequadamente. Uma solução que pode ser adotada para resolver este problema é a quebra do segmento (splitting), em dois menores. Entre os extremos do segmento é traçada uma linha reta, e o pixel do segmento que apresentar a maior distância (euclidiana) a esta linha será o pixel de quebra. Para os dois novos segmentos também pode ser aplicado o mesmo procedimento, caso necessário, e o processo todo pode se repetir, até que os segmentos obtidos atendam às necessidades de linearidade exigidas (após o ajustamento apresentem um pequeno resíduo). A figura 15 mostra este método sendo aplicado em um conjunto de pixels de borda. Após uma quebra, os segmentos resultantes podem ser reconectados com outros segmentos nas proximidades, pela repetição da etapa de conexão de segmentos colineares, o que caracteriza uma operação de reconexão (merge), caso seja atendida a condição apresentada na figura 8(a).

Figura 15 – Processo de quebra de segmentos (Roberts, 1965).

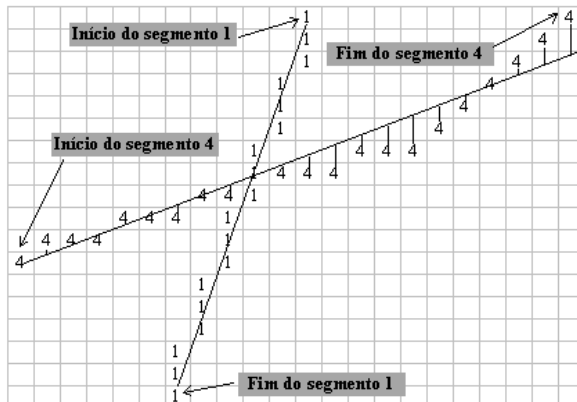


2.9 CONEXÃO DOS SEGMENTOS COM O AJUSTAMENTO

Neste trabalho, está implementada uma rotina de ajustamento, que trabalha apenas com linhas retas (regressão linear – supondo que os erros estão apenas em y , ou em x , dependendo da parametrização). Durante o processamento também é realizada uma soma dos valores dos resíduos, e caso seja obtido um número acima de um valor previamente estabelecido, um processo de quebra do segmento (splitting) é executado.

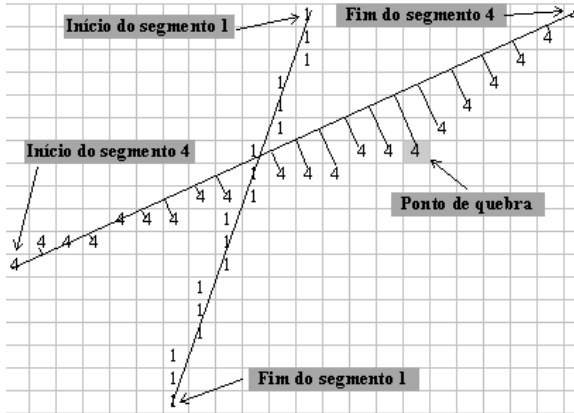
A figura 16 mostra um exemplo obtido nos experimentos, onde se verifica tal situação. Neste caso, o valor do resíduo está acima do limite previamente estabelecido, e portanto deve ser iniciado o processo de divisão do segmento (splitting), conforme mostra a figura 16, segmento número 4. No caso da situação apresentada nesta figura, o método de quebra implementado localiza a maior distância entre os pixels do segmento 4 e a reta que liga os extremos do segmento.

Figura 16 – Ajustamento de retas sobre dois segmentos.



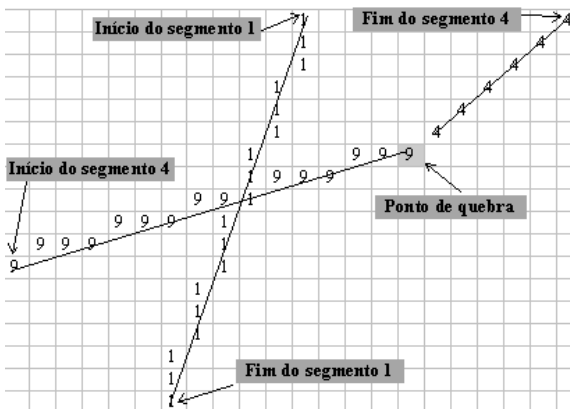
O pixel destacado na figura 17, e indicado como pixel de quebra, deve se tornar o pixel final do segmento que se inicia no pixel de início do segmento 4.

Figura 17 – Segmentos 1 e 4, e a identificação do pixel de quebra no processo de ajustamento.



Em seguida, todos os pixels a partir do pixel de início do segmento 4 devem ser substituídos por um novo valor de rótulo (um rótulo ainda não utilizado - no caso 9), constituindo-se assim um novo segmento de rótulo 9, conforme mostra a figura 18.

Figura 18 – Segmentos resultantes após quebra em dois menores (rótulos 4 e 9).



Na rotina implementada, o pixel início do segmento novo (rótulo 9) coincide com o início do segmento de rótulo 4, enquanto que o fim do segmento novo coincide com o pixel de quebra do segmento antigo (rótulo 4). Como o pixel fim do segmento 4 não é alterado, o novo pixel de início do mesmo pode ser determinado pela rotina de determinação de pixels extremos já apresentada anteriormente (que também atua por um processo de inundação – perseguição de pixels), sendo inicializada neste pixel (pixel início do segmento 4).

Para que a rotina (baseada em um processo de inundação) funcione adequadamente, conseguindo percorrer o segmento do início até o fim, e transpor os cruzamentos (segmento 1 na figura 16), é preciso que seja verificada uma vizinhança 24 (a rotina de inundação deve ser capaz de realizar saltos de um pixel, para continuar a percorrer o segmento com quebras). Por outro lado, para que o pixel de quebra não seja ultrapassado, apenas a vizinhança 8 deve ser utilizada. Com o objetivo de resolver esta situação o sistema utiliza o algoritmo apresentado na figura 19.

Figura 19 – Algoritmo utilizado para fazer a troca de rótulos no processo de quebra de segmentos.

```

quebra_proxima=Quebra_Na_Vizinhanca(i,j,1); // quebra viz-8 ?
se (quebra_proxima) // se estiver
    troca_local_rotulos(i,j,1); //troca rótulos na viz-8
senão // se quebra não está na viz-8
{
    possivel_caminhar=possivel_deslocar_vizinhanca(i,j,1);
    // caminhar na viz-8 ?
    se (possivel_caminhar) // se sim
        trocar_rotulo_vizinhanca_1(i,j); //troca rótulos na viz-8
    senão // se não é possível caminhar na viz-8
    {
        quebra_proxima=Quebra_Na_Vizinhanca(i,j,2); //quebra na viz-24
        se (quebra_proxima) // se sim
            troca_local_rotulos(i,j,2); // troca os rótulos na vizinhança 24
        senão // se a quebra não está na viz-24
            trocar_rotulo_vizinhanca_2(i,j); // troca rótulos na viz-24
    }
}

```

Conforme se observa no algoritmo (figura 19), inicialmente é verificado se o pixel de quebra está nas proximidades (vizinhança 8) e, caso isto ocorra, apenas os pixels dentro desta vizinhança terão seu rótulos verificados e trocados pelo novo valor, pela função `troca_local_rotulos(i,j,1)`. Caso o pixel de quebra não esteja nas proximidades (vizinhança 8), o algoritmo verifica se é possível caminhar utilizando uma vizinhança 8 (isto será possível se não houver um pixel de interrupção no segmento). Neste caso, percorre o segmento utilizando uma vizinhança 8. Não havendo a possibilidade de caminhar na vizinhança 8 (devido a alguma interrupção

do segmento ou devido a uma interseção com um outro segmento) a rotina então verifica se o pixel de quebra está na vizinhança 24. Neste caso, executa a função `troca_local_rotulos(i,j,2)`, que finaliza o processo verificando uma possível troca apenas nos pixels da vizinhança 24. Caso a quebra não esteja ainda na vizinhança 24, é verificada a possibilidade de se caminhar em uma vizinhança 24, e desta forma, é possível transpor o pixel de interrupção e continuar o processo até que o pixel de quebra seja atingido.

As funções `trocar_rotulo_vizinhanca_2(i,j)` e `trocar_rotulo_vizinhanca_1(i,j)` também são recursivas e participam do processo, conforme descrito.

Após a quebra do segmento em dois, os mesmos também precisam passar pelo processo de ajustamento, e caso ocorra dos resíduos continuarem acima de um limite aceitável, os segmentos precisam ser quebrados novamente. A figura 20 mostra o algoritmo utilizado para resolver esta etapa do processamento.

Figura 20 – Algoritmo utilizado para medir a qualidade do ajustamento e decidir as situações onde a quebra (splitting) é necessária.

```

se linha_horizontal(i); // se segmento i é aprox. horizontal
{
  ajusta_horizontal(i); // ajusta com param.  $y=ax+b$ 
   $Q = \frac{\sum \text{residuos}}{N}$  // Q = qualidade do ajustamento
  se  $Q >$  limiar Quebre_o_segmento(i);
} // Quebra o segmento em dois (i e novo)
senão // então segmento i é aproximadamente vertical
{
  ajusta_vertical(i); // ajusta com parametrização  $x=ay+b$ 
   $Q = \frac{\sum \text{residuos}}{N}$  // Q = qualidade do ajustamento
  se  $Q >$  limiar Quebre_o_segmento(i); // Quebra o segmento em dois (i e novo)
}
  
```

repita para os segmentos que foram quebrados():

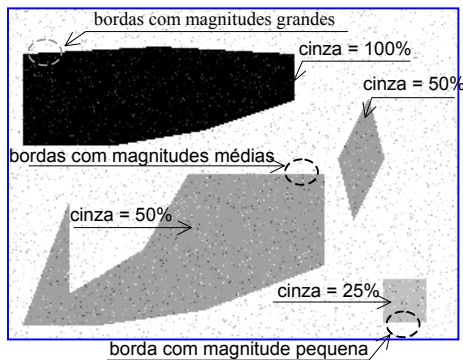
O valor de Q utilizado no algoritmo foi obtido empiricamente, sendo igual a 1 pixel, podendo ainda ser modificado, de acordo com os resultados obtidos. O valor de N (número de pixels no segmento) utilizado no denominador é justificado por ser mais interessante manter os segmentos com um grande número de pixels (segmentos compridos). Assim, um mesmo resíduo resultará em um valor de Q maior que aquele que seria obtido se o segmento fosse maior.

3 – Resultados

Os procedimentos discutidos neste trabalho foram implementados em um ambiente para vetorização automática. Entre outras facilidades o ambiente realiza a conexão de pixels e várias tarefas de visualização.

A figura 21 mostra uma das imagens utilizadas na realização dos experimentos. Trata-se de uma imagem sintética, gerada pelo próprio ambiente desenvolvido, que passou por um processo de degradação. O fato de a imagem ter sido construída pelo sistema garante o conhecimento prévio dos parâmetros a e b dos segmentos de retas presentes. Isto é necessário para a realização da análise quantitativa dos resultados obtidos. O processo de degradação foi composto de um borramento (filtragem passa-baixa) seguido de um acréscimo de 10% de ruídos (aleatórios).

Figura 21 – Imagem sintética utilizada nos experimentos.



A imagem foi projetada de modo a conter bordas com diferentes inclinações (parâmetros a) e, com contraste variável, de modo que o processo de detecção de bordas obtivesse pixels de bordas com diferentes faixas de magnitudes.

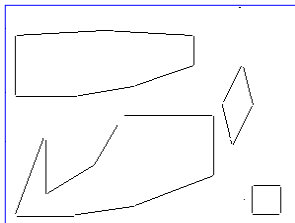
A tabela 1 mostra os valores verdadeiros das coordenadas dos vértices da figura 21 e também dos parâmetros a e b das retas definidas pelas bordas. Em todos os experimentos as etapas de afinamento e ligação foram realizadas com os métodos da Supressão Não Máxima e Varredura e Rotulação (utilizando o ajustamento em conjunto com a rotina de conexão e quebra de segmentos), respectivamente. A figura 22 mostra os resultados na imagem sintética, utilizando-se a seguinte seqüência: suavização pela mediana com máscaras rotativas (Sonka e Halavac, 1998); detecção da bordas pelo operador de Sobel; limiarização pelo método de Otsu Local (Artero e Tommaselli, 2000); afinamento e conexão pelo método da varredura e rotulação, com os passos adicionais sugeridos neste trabalho.

Tabela 1 – Parâmetros conhecidos na imagem padrão.

reta	Vértice 1	vértice 2	tipo *	parâmetros da reta	
	x_1, y_1	x_2, y_2		a	b
1	10,210	60,210	H	0,000000	210,000000
2	60,210	130,200	H	-0,142857	218,571428
3	130,200	210,170	H	-0,375000	248,750000
4	210,170	210,110	V	0,000000	210,000000
5	210,110	120,110	H	0,000000	110,000000
6	120,110	90,160	V	-0,600000	186,000000
7	90,160	40,190	H	-0,600000	214,000000
8	40,190	40,130	V	0,000000	40,000000
9	40,130	10,210	V	-0,375000	88,750000
10	10,190	70,90	H	0,000000	90,000000
11	70,90	130,80	H	-0,166666	101,666666
12	130,80	190,60	H	-0,333333	123,333333
13	190,60	190,30	V	0,000000	190,000000
14	190,30	100,25	H	0,055555	19,444444
15	100,25	10,30	H	-0,055555	30,555555
16	10,30	10,90	V	0,000000	10,000000
17	210,110	240,60	V	-0,500000	270,000000
18	240,60	250,100	V	0,250000	225,000000
19	250,100	230,140	V	-0,500000	300,000000
20	230,140	210,110	V	0,250000	195,000000
21	250,180	280,180	H	0,000000	180,000000
22	280,180	280,210	V	0,000000	280,000000
23	280,210	250,210	H	0,000000	210,000000
24	250,210	250,180	V	0,000000	250,000000

* H indica a parametrização $y = ax + b$ e V indica $x = a'y + b'$.

Figura 22 – Resultado obtido com a imagem padrão.



Os valores ajustados para os parâmetros das retas foram comparados aos valores verdadeiros conhecidos da simulação e foi calculado o erro médio

quadrático para todas as retas, que é apresentado na tabela 2.

Tabela 2 – E.M.Q. dos parâmetros a e b.

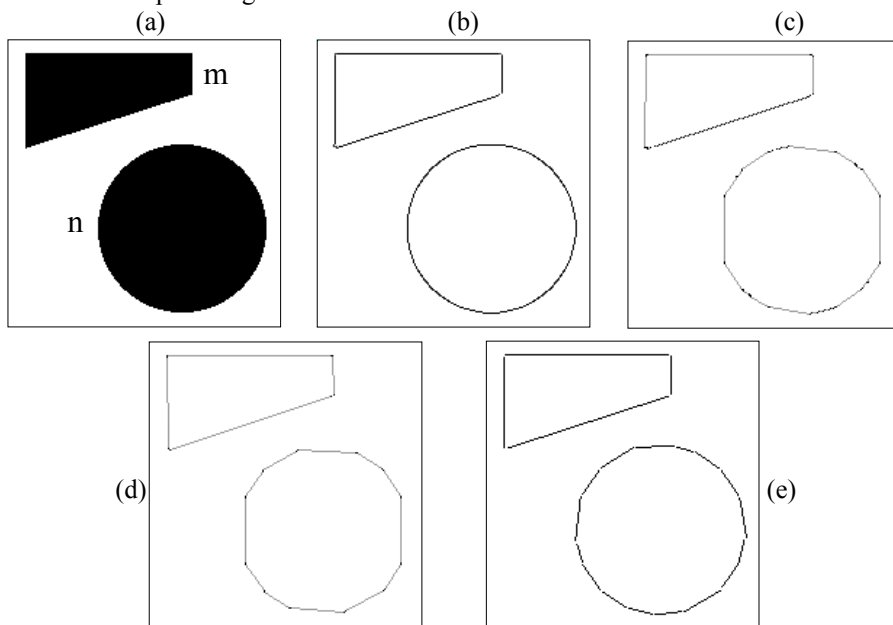
EMQ(a)	EMQ(b) em pixels
0,00597018	1,34346758

Pode-se perceber que a acurácia do método está em torno de 1 pixel, o que é aceitável. Além disto, não ocorreram erros grosseiros em nenhum dos experimentos realizados.

A figura 23 mostra mais alguns resultados obtidos com as funções apresentadas neste trabalho; (a) é a imagem original, (b) é a imagem de bordas, obtida após a etapa de afinamento, (c) é o resultado obtido após a aplicação da conexão por varredura e rotulação (nota-se o grande número de pequenos segmentos, (d) é o resultado obtido com a conexão dos segmentos colineares, que, embora tenha conseguido eliminar os pequenos segmentos (colineares) do objeto m, conecta alguns segmentos não muito alinhados do objeto n. Após o ajustamento em (e), esta situação está totalmente resolvida nos dois objetos (o ajustamento também resolve o deslocamento ocorrido na borda vertical do objeto m – figura 23(c)).

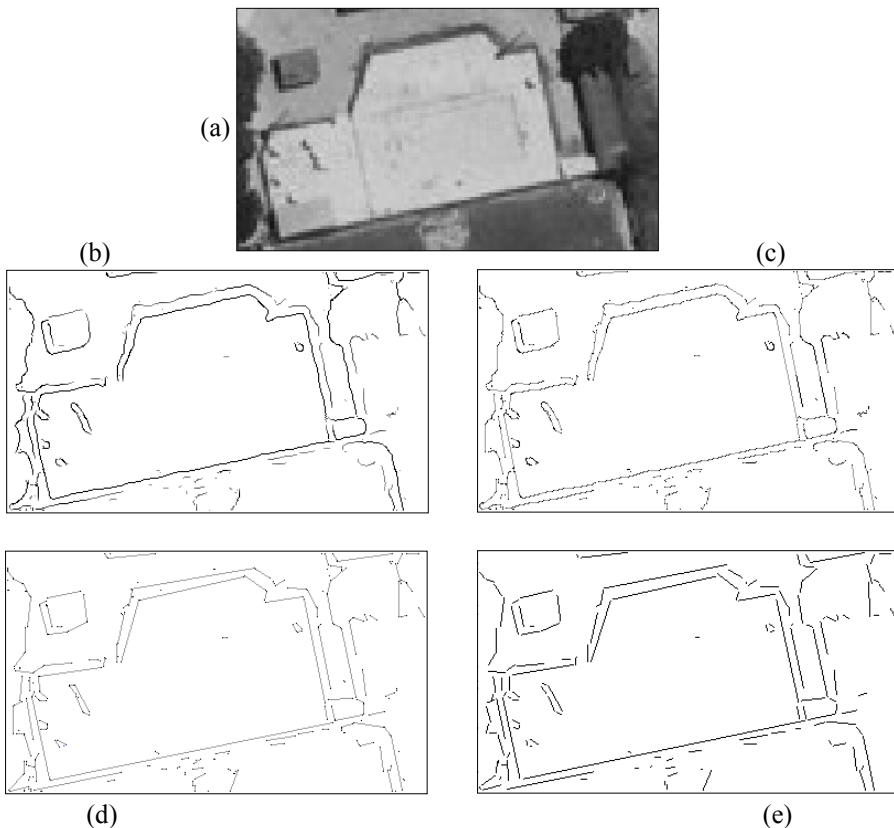
A figura 24 permite uma análise dos resultados obtidos com uma imagem real. A imagem original (a) consiste de uma parte de uma imagem aérea (área urbana); (b) é a imagem de bordas, após a etapa de afinamento; (c) é o resultado obtido após a etapa de conexão pelo método de varredura e rotulação (nota-se um excessivo número de pequenos segmentos); (d) é o resultado obtido após a etapa de conexão de segmentos colineares e (e) mostra o resultado final, após o ajustamento combinado com o split-merge.

Figura 23 – Resultados com a imagem de testes (os pontos escuros em (c) e (d) indicam os extremos dos segmentos) e (e) após o ajustamento com split-merge.



Pode-se notar que todas as feições significativas foram extraídas e há um delineamento adequado da edificação central e das estruturas vizinhas.

Figura 24 – Resultados obtidos com a imagem aérea a) imagem original (fotografia aérea 1:8000) , b) Imagem de bordas, c) Conexão por varredura e rotulação, d) Após a conexão dos segmentos e e) após o ajustamento.



5 - Conclusão

Neste trabalho foram apresentadas algumas técnicas desenvolvidas para realizar a extração de feições em imagens digitais. Além de uma análise visual dos resultados, também foi apresentada uma análise numérica dos erros obtidos em experimentos com uma imagem sintética.

A técnica apresentada é uma adaptação com melhorias de métodos já existentes na literatura, mas que apresentam problemas em situações especiais complexas, como é o caso de imagens aéreas.

Esta técnica desenvolvida pode ser ainda melhorada se forem agregados procedimentos de extração de bordas com precisão subpixel. Mesmo sem esta característica, entretanto, pôde-se observar que a acurácia, em torno de um pixel, é

adequada à maioria das aplicações.

O método mostrou-se eficiente em imagens aéreas, extraindo todas as feições de interesse.

BIBLIOGRAFIA

- ARTERO, A. O., TOMMASELLI, A.M.G., Limiarização Automática de Imagens Digitais, Boletim de Ciências Geodésicas, vol. 6, nº 1, pp. 38-48, 2000.
- BALLARD, D.H., BROWN, C. M., Computer Vision, Prentice Hall, 1982.
- GONZALEZ, R. C., Woods, R. E., Digital Image Processing, Addison Wesley, 1993.
- PAINE, S. H., LODWICK, G. D., Edge Detection and Processing of Remotely Sensed Digital Images, Photogrammetria, vol. 43, n. 6, pp. 323-336, 1989.
- ROBERTS, L. G. Machine Perception of the three-dimensional solids, In: Optical and Electro-optical Information Processing, MIT Press, 1965.
- SONKA, M., HLAVAC, V., BOYLE, R., Image Processing, Analysis, and Machine Vision, PWS Publishing, 1998.
- TOMMASELLI, A. M. G., TOZZI, C. L. A Recursive Approach to Space Resection using Straight Lines, Photogrammetric Engineering and Remote Sensing, vol. 62, n. 1, pp. 57 – 66, 1996.
- TOMMASELLI, A.M.G., Extração Automática de Feições Lineares em Imagens Digitais para Aplicações Cartográficas, Relatório de Bolsa de Produtividade em Pesquisa, cnpq, 1999
- VENKATESWAR, V.; CHELLAPA, R., Extraction of Straight Lines in Aerial Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, nº11, 1992.
- ZHOU, Y. T.; VENKATESWAR, V.; CHELLAPA, R., Edge Detection and Linear Feature Extracting using a 2-d Random Field Model, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, nº1, 1989.

(Recebido em 18/04/00. Aceito para publicação em 19/06/02.)