

ALGORITMO DE PROGRAMAÇÃO DINÂMICA: FUNDAMENTOS E APLICAÇÕES EM EXTRAÇÃO DE FEIÇÕES

Aluir P. Dal Poz¹
Giovane Maia do Vale²

¹ Universidade Estadual Paulista – UNESP – Departamento de Cartografia – aluir@prudente.unesp.br

² Universidade Estadual Paulista – UNESP – Programa de Pós-Graduação em Ciências Cartográficas – gmvale@pos.prudente.unesp.br

RESUMO

O algoritmo de programação dinâmica vem sendo usado em processamento de imagens digitais e visão computacional por mais de trinta anos. A primeira aplicação foi no reconhecimento de caracteres tipografados e manuscritos em imagens digitais. Desde este trabalho pioneiro, ele vem sendo aplicado numa variedade de problemas de extração de feições envolvendo diferentes tipos de imagens, incluindo imagens aéreas e de satélite e imagens de microscópio. A programação dinâmica é uma técnica para resolver problemas de otimização quando nem todas as variáveis na função custo estão interrelacionadas simultaneamente. Dois exemplos de aplicação do algoritmo de programação dinâmica em extração de feições lineares são brevemente apresentados e discutidos. A primeira aplicação é na extração de rodovias em imagens de baixa resolução, como as imagens SPOT. Em qualquer aplicação da programação dinâmica o problema mais importante a ser resolvido é a definição da função custo. Para o problema de extração de rodovia esta função é definida levando em conta 5 propriedades básicas de rodovia, sendo três fotométricas (por exemplo, rodovias são feições lineares com alta intensidade de brilho) e duas geométricas (por exemplo, rodovias são feições suaves). Prova-se que apenas seis variáveis são interrelacionadas simultaneamente nesta função custo e, como tal, pode ser eficientemente resolvida pelo algoritmo de otimização por programação dinâmica. A segunda aplicação é uma técnica para detecção de bordas, cuja função custo é definida como uma soma ponderada entre as somatórias de magnitudes do gradiente e de curvaturas, as quais são tomadas ao longo de cadeias de pixels candidatas a constituir bordas. Mostra-se que esta função custo pode ser eficientemente resolvida pelo algoritmo de programação dinâmica. Este artigo apresenta e discute o algoritmo de otimização de programação dinâmica, bem como sua aplicabilidade em problemas de extração de feições, mas uma atenção especial é dada para dois exemplos específicos envolvendo extração de rodovia e detecção de borda.

Palavras-chave: Programação Dinâmica, Função Custo, Extração de Feições.

DYNAMIC PROGRAMMING ALGORITHM: FUNDAMENTALS AND APPLICATIONS IN FEATURE EXTRACTION

ABSTRACT

Dynamic programming algorithm has been used in digital image processing and computer vision for over thirty years. The earliest application was in the recognition of type and handwritten characters in digital images. Since this pioneer work, it has been applied in a plenty of feature extraction problems involving different kinds of image data, as for example aerial and satellite images, and microscope images. Dynamic programming is a technique for solving optimization problems when not all variables in the cost function are interrelated simultaneously. Two examples of application of the dynamic programming algorithm in linear feature extraction are briefly presented and discussed. The first application is on the road extraction from low-resolution images, like SPOT satellite image. In any dynamic programming application the key problem to be solved is the definition of the cost function. The cost function for this road extraction problem is formulated taking into account 5 basic road properties, three describing photometric (for example, roads are light linear features) and two describing geometric (for example, roads are smooth features) behavior. It is shown that only six variables are interrelated simultaneously in this cost function and, as such, it can be efficiently solved by the dynamic programming optimization algorithm. The second application is a technique for edge detection, whose cost function is defined as a weighted sum of cumulative edge strength and cumulative curvature along pixel chains candidate to

constitute edges. It is shown that this cost function can be efficiently solved by the dynamic programming algorithm. This paper presents the dynamic programming optimization algorithm, as well as its applicability in feature extraction problems, but special attention is given for two specific examples concerning road extraction and edge detection.

Keywords: *Dynamic Programming, Cost Function, Feature Extraction.*

1 INTRODUÇÃO

O algoritmo de programação dinâmica foi introduzido em processamento de imagens por Kovalevskii (1967), cujo problema tratado foi o reconhecimento de caracteres tipografados e manuscritos. Desde então, inúmeras pesquisas foram realizadas sobre extração de feições utilizando o algoritmo de programação dinâmica, estendendo-se pelas mais diversas áreas de conhecimento. Montanari (1971) contribuiu significativamente ao propor uma metodologia para a extração de feições lineares geometricamente suaves em imagens digitais. O problema colocado e resolvido por este autor consistia em detectar e extrair curvas caracterizadas como uma seqüência de n pixels adjacentes (p_1, \dots, p_n). Uma metodologia similar foi proposta por Martelli (1972, 1976), só que o problema é agora colocado como uma busca pelo caminho de custo mínimo em um grafo ponderado. Fischler et al. (1981) propuseram uma metodologia para a extração de rodovias em imagens (LANDSAT) de baixa resolução, nas quais as rodovias se manifestam como linhas. O problema básico consistia, primeiramente, em classificar cada pixel da imagem usando propriedades básicas (no caso, homogeneidade e contraste) de rodovia e, após, extrair os contornos usando o algoritmo de programação dinâmica, sob restrições geométricas de suavidade e continuidade. O algoritmo de programação dinâmica foi também aplicado no delineamento de glomérulos em imagens de microscópio de seções renais (Yamada et al., 1988). Os glomérulos são estruturas aproximadamente circulares, podendo então ser modelados por polígonos com curvatura constante. O método consistia em reconhecer feições na imagem com característica de modelo circular, tratando-se na verdade de um problema de correspondência usando programação dinâmica. Em Mortensen e Barret (1995) a programação dinâmica é transformada num problema de busca em grafo para gerar uma árvore de custo mínimo, na qual se tem caminhos ótimos entre pixels da imagem. Estes caminhos ótimos correspondem a feições na imagem. Uma metodologia versando sobre extração de rodovias em imagens SPOT de baixa resolução foi proposta em Gruen e Li (1995, 1997), onde o algoritmo de programação dinâmica foi utilizado para resolver um modelo de rodovia, formulado com base em propriedades radiométricas (por exemplo, contraste e homogeneidade) e geométricas (por exemplo, suavidade) de rodovia. Versões modificadas desta metodologia foram apresentadas em Dal Poz (2000) e Dal Poz e Agouris (2001).

Nos processos de extração de feições existem dois tipos básicos de feições lineares que são alvo de extração (Agouris et al., 1988, Bellman e Shortis, 2002): 1) feições descrevendo os contornos dos objetos, como por exemplo bordas e linhas; e 2) feições descrevendo eixos de simetria, como por exemplo o eixo de rodovia em imagens de alta resolução. Vale ressaltar que os trabalhos prévios, como alguns que foram brevemente descritos acima, destinam-se à extração de feições lineares de tipo 1, isto é, contornos dos objetos. Neste sentido, Vale e Dal Poz (2003) propuseram uma metodologia para tratar de forma rigorosa as feições do tipo 2, mas num contexto específico de extração de rodovias em imagens de média e alta resolução. Nesses tipos de imagem as rodovias têm que ser modeladas como faixas estreitas e alongadas, ficando como alvo de extração o eixo central de cada rodovia. Tendo por base o trabalho anterior de Dal Poz e Agouris (2001), uma importante modificação foi realizada por Vale e Dal Poz (2003) no modelo de rodovia para forçar a definição do eixo central de rodovia, possibilitando a obtenção de excelentes resultados com imagens de média e alta resolução.

A comprovada importância do algoritmo de programação dinâmica em aplicações envolvendo extração de feições em imagens de diferente natureza (imagens de microscópio, imagens aéreas e de satélite etc.) é uma das motivações para este trabalho. Entretanto, a principal motivação vem da constatação de que revisões compreensivas do algoritmo de programação dinâmica não têm sido apresentadas em publicações de grande circulação (principalmente as vinculadas em periódicos), ligadas à área de Fotogrametria Digital - alguns exemplos onde isso ocorre são Fischler et al. (1981), Gruen e Li (1995, 1997), Dal Poz (2000), Dal Poz e Agouris (2001) e Vale e Dal Poz (2003). As razões para este procedimento vão desde a relativa complexidade matemática inerente e a necessidade de um espaço relativamente grande para expô-lo, até a boa documentação existente em publicações da área de Visão Computacional - ver, por exemplo, Ballard e Brown (1982). Assim, geralmente se apresenta uma noção bastante superficial do algoritmo de programação dinâmica, ficando o foco do artigo quase que exclusivamente sobre o desenvolvimento da função custo e sobre os detalhes gerais de sua solução usando o algoritmo de programação dinâmica. Em vista do exposto, este artigo segue uma linha relativamente inversa, dando grande ênfase a uma revisão compreensiva do algoritmo de programação dinâmica (Seção 2) e apresentando dois exemplos de aplicação em extração de feições em imagens

digitais (Seção 3), isto é, extração de rodovias e bordas. O artigo é finalizado na Seção 4, onde algumas considerações finais são apresentadas.

2 ALGORITMO DE PROGRAMAÇÃO DINÂMICA

2.1 FUNDAMENTOS BÁSICOS

Quando em um problema, nem todas as variáveis envolvidas estão interrelacionadas simultaneamente, a solução pode ser eficientemente encontrada através de uma técnica de otimização conhecida como programação dinâmica (BALLARD e BROWN, 1982).

A figura 1 mostra o problema clássico da programação dinâmica (LI, 1997). A idéia básica é encontrar o caminho ótimo em um grafo, entre os nós A e N, cuja solução pode ser encontrada seqüencialmente. No primeiro estágio, é necessário escolher um dos caminhos entre o nó A e os nós 10, 11 ou 12. Supondo que o nó 11 é a melhor escolha, a próxima decisão deve ser tomada entre os nós 20, 21 e 22. Esta estratégia vai sendo repetida até que o nó N seja alcançado. O caminho definido pelos nós A, 11, 22, 32, 41 e N é o caminho ótimo procurado, correspondendo à trajetória de custo mínimo.

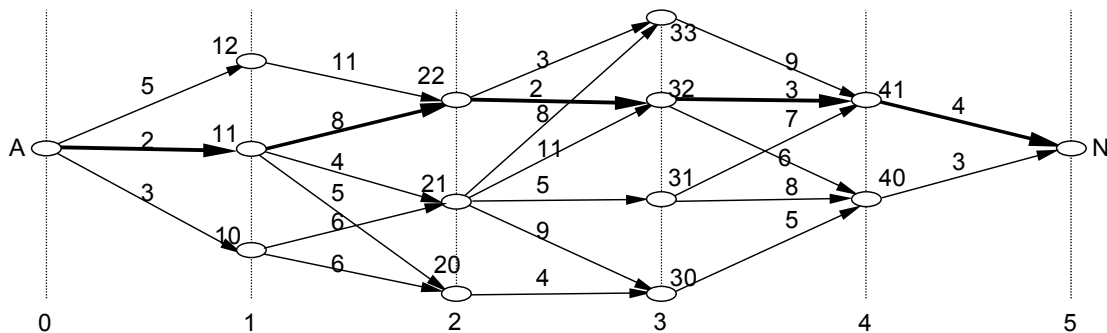


Figura 1 - Problema Clássico de Programação Dinâmica

A seguir, o problema é apresentado de maneira mais formal (LI, 1997). Sejam $\{P_i\}$, $i= 1, 2, \dots, n$, um conjunto de nós e $[C_{ij}]$ uma matriz custo, onde C_{ij} é o custo para ir do nó P_i para o nó P_j . O caminho ótimo é tal que o custo para ir do nó inicial P_A para o nó final P_N seja mínimo.

Sejam agora $g(i,j)$ a função (ou função custo) que descreve o custo para mover do nó P_i para o nó P_j e s os nós ao longo do caminho. Pode-se escrever,

$$g(i,j) = g(i,s) + g(s,j) \quad (1)$$

Para encontrar o caminho ótimo entre os nós P_i e P_j é necessário encontrar $g(i,j)$ e um conjunto s que satisfaça,

$$g(i,j) = \min_s [g(i,s) + g(s,j)] \quad (2)$$

Se nenhuma restrição for imposta, a expressão 2 estabelece um problema geral de busca em grafo. Entretanto, se o grafo for similar àquele apresentado na figura 1, onde apenas alguns nós estão diretamente conectados, então a busca pelo caminho ótimo envolve um procedimento de otimização seqüencial e, como tal, um algoritmo eficiente para resolver o problema é o da programação dinâmica.

O algoritmo para resolver o problema estabelecido acima é apresentado na seção 2.2.

2.2 ALGORITMO PARA RESOLVER O PROBLEMA DE PROGRAMAÇÃO DINÂMICA

Considerar o problema de encontrar o máximo M da função,

$$g = g(x_1, x_2, \dots, x_n) \quad (3)$$

Portanto, o objetivo é encontrar a ênupla $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ que maximiza a função 3. Se esta função é analiticamente desconhecida, e as variáveis independentes (x_1, \dots, x_n) assumem valores discretos, a solução pode ser encontrada através da exaustiva combinação dessas variáveis. Contudo, se nem todas as variáveis são simultaneamente interrelacionadas, então um algoritmo de programação dinâmica pode ser aplicado para resolver o problema.

Supor então, que,

$$g(x_1, x_2, \dots, x_n) = g_1(x_1, x_2, x_3) + \dots + g_{n-2}(x_{n-2}, x_{n-1}, x_n) \quad (4)$$

onde, a função g_1 depende apenas de x_1, x_2 e x_3 , a função g_2 apenas de x_2, x_3 , e x_4 , e assim por diante.

Assim posto o problema, o valor máximo da função 4 pode ser encontrado através de um processo de eliminação seqüencial das variáveis envolvidas.

O processo inicia então com a eliminação da variável x_1 . Como as funções g_2, \dots, g_{n-2} não dependem de x_1 , somente a função g_1 deve ser considerada nesse momento. A eliminação de x_1 é realizada através da seleção do valor de x_1 que maximiza g_1 , para cada realização do par (x_2, x_3) , isto é,

$$f_1(x_2, x_3) = \max_{x_1} [g_1(x_1, x_2, x_3)] \quad (5)$$

O processo continua com a eliminação da variável x_2 , sendo que o mesmo princípio é aplicado, como segue,

$$f_2(x_3, x_4) = \max_{x_2} [f_1(x_2, x_3) + g_1(x_2, x_3, x_4)] \quad (6)$$

A eliminação das demais variáveis segue o mesmo princípio da etapa anterior. Entretanto, é conveniente formalizar a eliminação das duas últimas variáveis (x_{n-1}, x_n) , como segue,

$$f_{n-1}(x_n) = \max_{x_{n-1}} [f_{n-2}(x_{n-1}, x_n)] \quad (7)$$

$$M = \max(g) = \max_{x_n} [f_{n-1}(x_n)] \quad (8)$$

Finalmente, pode-se generalizar o algoritmo descrito acima através das seguintes equações recursivas,

$$f_0(x_1, x_2) = 0 \quad (9)$$

$$f_k(x_{k+1}, x_{k+2}) = \max_{x_k} [f_{k-1}(x_k, x_{k+1}) + g_k(x_k, x_{k+1}, x_{k+2})] \quad (10)$$

$$f_{n-1}(x_n) = \max_{x_{n-1}} [f_{n-2}(x_{n-1}, x_n)] \quad (11)$$

$$M = \max[g] = \max_{x_n} [f_{n-1}(x_n)] \quad (12)$$

O princípio representado pelas equações 5, 6, 7 e 8 é usado para construir uma série de tabelas, possibilitando posteriormente a obtenção da ênupla $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ que maximiza a função 4. Para facilitar o entendimento do algoritmo, BALLARD; BROW (1982) apresentam um exemplo artificial bem simples, onde $n = 4$ e as funções g_i dependem apenas de 2 variáveis. Como mostra a figura 2, cada variável x_i assume três valores discretos. Como um exemplo, a variável x_1 pode assumir os valores 0, 1 e 2. As funções g_i são definidas pelas tabelas da figura 2. Por exemplo, $g_2(1, -1) = 1$.

		x_2		
		1	2	3
x_1	0	5	7	3
	1	2	1	8
	2	6	3	3
		$g_1(x_1, x_2)$		

		x_3		
		-1	0	1
x_2	1	1	7	1
	2	1	1	3
	3	5	6	2
		$g_2(x_2, x_3)$		

		x_4		
		1	2	3
x_3	-1	7	9	8
	0	2	3	6
	1	5	4	1
		$g_3(x_3, x_4)$		

Figura 2 - Representação das Funções g_1, g_2 e g_3 Através de Tabelas.

A figura 3 mostra o primeiro passo para a solução do problema de determinação de máximo da função definida pela soma das funções g_1, g_2 e g_3 . Como mostra a figura 3, cada valor de x_2 é associado com o valor de x_1 que maximiza $g_1(x_1, x_2)$, obtendo-se assim os valores máximos de g_1 .

x_2	f_1	x_1
-------	-------	-------

1	6	2
2	7	0
3	8	1

Figura 3 – Primeiro Passo

As fórmulas de recursão 9 e 10 são usadas para a construção da função $f_1(x_2)$:

$$f_1(x_2) = \max_{x_1} [f_0(x_1) + g_1(x_1, x_2)], \text{ com } f_0(x_1) = 0 \quad (13)$$

Aplicando a equação 13 para os possíveis valores de x_2 (i. e., 1, 2 e 3), obtém-se os respectivos valores de $f_1(x_2)$ (figura 3):

$$f_1(x_2 = 1) = \max_{x_1} [f_0(x_1) + g_1(x_1 = 2, x_2 = 1)] = 6 \quad (14)$$

$$f_1(x_2 = 2) = \max_{x_1} [f_0(x_1) + g_1(x_1 = 0, x_2 = 2)] = 7 \quad (15)$$

$$f_1(x_2 = 3) = \max_{x_1} [f_0(x_1) + g_1(x_1 = 1, x_2 = 3)] = 8 \quad (16)$$

			x_3		
			-1	0	1
		1	7	13	7
x_2	2	8	8	10	
	3	13	14	10	

(a)

x_3	f_2	x_2
-1	13	3
0	14	3
1	10	2

(b)

Figura 4 - Segundo Passo. (a) $f_1(x_2) + g_2(x_2, x_3)$; e (b) $f_2(x_3)$

As figuras 4(a) e 4(b) mostram o segundo passo da solução do problema de otimização. Os elementos da tabela da figura 4(a) são obtidos pela adição dos elementos da função $f_1(x_2)$ às respectivas linhas de $g_2(x_2, x_3)$, tendo por base a seguinte fórmula de recursão:

$$f_2(x_3) = \max_{x_2} [f_1(x_2) + g_2(x_2, x_3)] \quad (17)$$

O argumento do operador $\max[\cdot]$ (equação 17) é usado para construir a tabela da figura 4(a). Por exemplo, para montar a primeira linha (valores 7, 13 e 7) basta fixar $x_2 = 1$ e atribuir à x_3 os valores -1, 0 e 1.

A exemplo do passo 1, a função $f_2(x_3)$ (figura 4(b)) é obtida selecionando-se os máximos (em cinza) na tabela da figura 4(a), como segue:

$$f_2(x_3 = -1) = \max_{x_2} [f_1(x_2 = 3) + g_2(x_2 = 3, x_3 = -1)] = 13 \quad (18)$$

$$f_2(x_3 = 0) = \max_{x_2} [f_1(x_2 = 3) + g_2(x_2 = 3, x_3 = 0)] = 14 \quad (19)$$

$$f_2(x_3 = 1) = \max_{x_2} [f_1(x_2 = 2) + g_2(x_2 = 2, x_3 = 1)] = 10 \quad (20)$$

As tabelas das figuras 5(a) e 5(b) são montadas como no passo anterior. A seguinte fórmula de recursão é utilizada:

$$f_3(x_4) = \max_{x_3} [f_2(x_3) + g_3(x_3, x_4)] \quad (21)$$

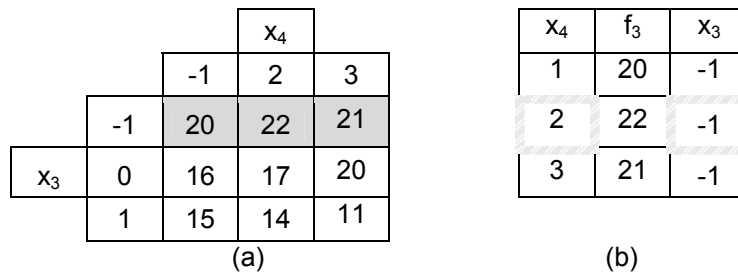


Figura 5 - Terceiro Passo. (a) $f_2(x_3) + g_3(x_3, x_4)$; e (b) $f_3(x_4)$.

Finalmente, a solução para o problema de otimização é encontrada seguindo-se, pela ordem, as tabelas das figuras 5(b), 4(b) e 3. Conforme mostra a figura 5(b), o máximo de $f_3(x_4)$ está associado com $x_4 = 2$ e $x_3 = -1$. Retrocedendo-se à tabela da figura 4(b), nota-se que o valor a ser considerado de $f_2(x_3)$ é 13, pois é este que está relacionado com $x_2 = 3$ e $x_3 = -1$, sendo que este último foi previamente selecionado na tabela anterior. Em seguida, seguindo o caminho inverso, encontra-se $x_1 = 1$, que é o valor relacionado a $x_2 = 3$, em $f_1(x_2)$. Assim, tem-se a solução ótima $g(x_1, x_2, x_3, x_4) = 22$, com $x_1 = 1, x_2 = 3, x_3 = -1$ e $x_4 = 2$.

Retornando-se agora ao caso mais geral (eq. 4), e levando-se em conta as equações 9 e 10, pode-se resumir o algoritmo descrito através dos seguintes passos :

- 1- fazer $K= 1$ e $f_0(x_1, x_2)= 0$;
 - 2- construir a tabela de máximos para a função $f_k(x_{k+1}, x_{k+2})$;
 - 3- se $K < n-1$, então fazer $K= K + 1$ e ir para o passo 2;
 - 4- tomar o valor máximo de f_{n-1} e retroagir pelas $n-1$ tabelas para obter a ênupla solução.
- O fluxo acima pode ser melhor visualizado através da figura 6.

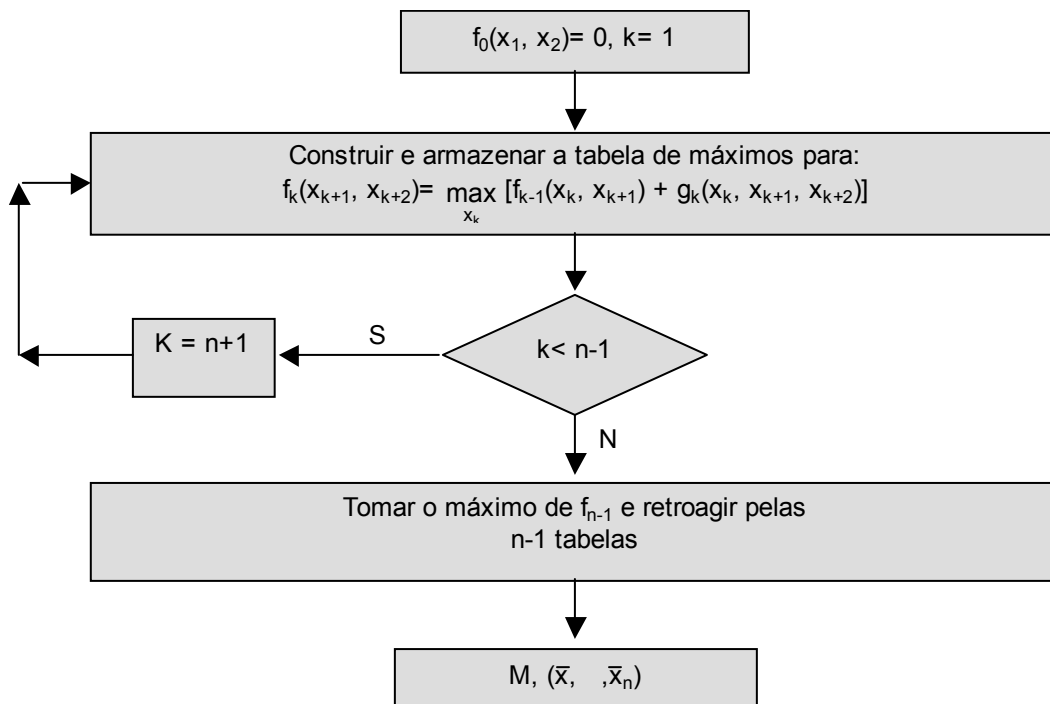


Figura 6 - Fluxo do algoritmo de programação dinâmica

Abaixo são apresentados alguns comentários adicionais sobre o algoritmo apresentado (LI, 1997):

- o procedimento apresentado é ótimo, pois todas as possibilidades são avaliadas;
- considerando que cada uma das n variáveis assume m valores discretos, o esforço computacional é $O(nm^3)$;
- uma possibilidade interessante desse algoritmo é a introdução de inunções de desigualdade. Por exemplo, na extração de uma rodovia, pontos que superam uma determinada limiar de curvatura não são avaliados;
- o algoritmo pode ser utilizado para encontrar o valor mínimo, em vez do máximo. Para tanto, basta substituir o operador $\max(\cdot)$ por $\min(\cdot)$;
- a função 4 pode ser estendida para um caso mais geral, onde cada variável depende de p subseqüentes variáveis. Para o caso analisado $p= 3$. A complexidade computacional cresce exponencialmente com o aumento de p ; e

- a equação 4 é conhecida como função custo.

3 EXEMPLOS DE APLICAÇÃO EM EXTRAÇÃO DE FEIÇÕES

Dois exemplos envolvendo extração de feições através do algoritmo de otimização por programação dinâmica são apresentados abaixo. O primeiro exemplo refere-se à extração de rodovias em imagens de baixa resolução. O segundo exemplo mostra uma aplicação envolvendo o problema de extração de bordas. Em ambos os casos não há a intenção de descrever em detalhes as respectivas soluções, uma vez que uma descrição básica da função custo relativa a cada aplicação é suficiente para entender o processo inerente de otimização pelo algoritmo de programação.

3.1 EXTRAÇÃO DE RODOVIAS EM IMAGENS DE BAIXA RESOLUÇÃO

Uma rodoviária em uma imagem de baixa resolução pode ser modelada como uma feição linear suave e de alta intensidade de brilho. O modelo matemático de rodovia baseia-se em cinco propriedades básicas, sendo três fotométricas e duas geométricas.

A primeira propriedade é fotométrica e estatui que os pixels de uma rodovia são mais claros que os de fundo. Isto significa que uma rodovia em uma imagem digital pode ser considerada como uma faixa estreita e contínua de alta intensidade de brilho, margeada por regiões (fundos) de baixa intensidade. Portanto, a soma dos quadrados dos tons de cinza da imagem ($G(x,y)$) ao longo da rodovia é máxima, isto é,

$$E_{p1} = \sum_i G^2(x_i, y_i) \rightarrow \text{Máx} \quad (22)$$

A segunda propriedade é também fotométrica e fundamenta-se no conhecimento a priori de que o material de rolagem, geralmente asfalto ou concreto, não varia muito em curtas distâncias. Isso implica em respostas espectrais similares em segmentos curtos. Portanto, uma expressão matemática baseada no conceito de homogeneidade pode ser escrita, como segue,

$$E_{p2} = \sum_i \sum_j [G(x_j, y_j) - G_{d^m}^{DS_i}]^2 \rightarrow \text{Mín} \quad (23)$$

onde, $G_{d^m}^{DS_i}$ é dada pela expressão 24.

$$G_{d^m}^{DS_i} = \frac{\sum_j G(x_j, y_j)}{DS_i} \quad (24)$$

Nas equações 23 e 24, o índice j é usado para indexar pontos ao longo de cada vetor ($i, i+1$). A distância DS_i é dada por,

$$DS_i = \sqrt{(y_{i+1} - y_i)^2 + (x_{i+1} - x_i)^2} \quad (25)$$

A terceira propriedade é uma generalização das 2 primeiras. Visto que uma rodovia é uma feição linear de alta intensidade de brilho, e que os pontos mais distantes dessa curva têm menor influência na definição da mesma, pode-se escrever a seguinte expressão matemática para representar esta propriedade,

$$E_{p3} = \sum_i e^{-d_i^2} G^2(x_i, y_i) \rightarrow \text{Máx} \quad (26)$$

onde,

- d_i é a distância entre um ponto (x_i, y_i) e a curva discreta representando a rodovia; e

- $e^{-d_i^2}$ é uma função gaussiana, inversamente proporcional ao quadrado de d_i .

A quarta propriedade reflete uma característica geométrica fundamental de uma rodovia, isto é, a suavidade. Por questões práticas, uma rodovia é normalmente composta por segmentos retos e curvas suaves, geralmente circulares. Esta propriedade dá origem ao seguinte modelo matemático (figura 7),

$$E_g = \sum_i [2 - 2 \cos(a_i - a_{i+1})] / DS_i \rightarrow \text{Mín} \quad (27)$$

onde,

- $a_i - a_{i+1}$ é o ângulo de deflexão entre dois vetores sucessivos da entidade discreta representando a rodovia; e
- DS_i é a distancia entre o vértice i e $i-1$ ou $i+1$.

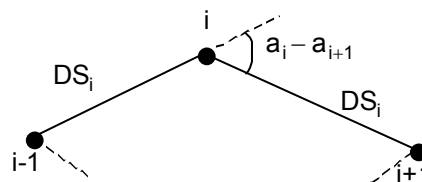


Figura 7 - Curvatura no Vértice i

A quinta propriedade estatui que a curvatura local de uma rodovia possui um limite máximo, isto é,

$$C_{gi} = |a_i - a_{i+1}| < T \quad (28)$$

onde, T é um limiar.

Considerando agora que uma rodovia pode ser representada por uma linha poligonal de n vértices, o seguinte par de equações pode ser usado para modelar o objeto rodovia (DAL POZ e AGOURIS, 2001),

$$E = \sum_{i=1}^{n-1} ((Ep_1 - bEp_2 + cEp_3) [1 + \cos(a_i - a_{i+1})] / DS_i) = \sum_{i=1}^{n-1} E_i(p_{i-1}, p_i, p_{i+1}) \quad (29)$$

$$C_i = |a_i - a_{i+1}| < T, \quad i = 1, \dots, n-1 \quad (30)$$

onde, b e c são duas constantes positivas e Ep_1 , Ep_2 e Ep_3 são dados respectivamente pelas equações 22, 23 e 26.

Portanto, o modelo matemático de rodovia é composto por 2 funções:

- E : corresponde à função objetivo (eq. 29), podendo ser decomposta em funções E_i dependendo apenas de três pontos vizinhos (p_{i-1} , p_i , p_{i+1}) da linha poligonal que representa a rodovia; e
- C_i : devido a sua definição (eq. 30), também é denominada de inunção de desigualdade e permite limitar o espaço de busca do problema, pois a função E é apenas avaliada para pontos que satisfaçam $C_i < T$.

Como mostra a equação 29, o modelo matemático de uma rodovia em uma imagem de baixa resolução depende simultaneamente de três vértices sucessivos da linha poligonal representado a rodovia. Isto significa que seis variáveis estão interrelacionadas simultaneamente. Assim, a estrutura da equação (29) é semelhante à da equação 4, podendo então ser resolvida eficientemente pelo algoritmo de otimização por programação dinâmica. O resultado desse processo de otimização é uma linha poligonal que maximiza a equação (29) e satisfaz a inunção de desigualdade (eq. 30).

3.2 EXTRAÇÃO DE BORDA

Os detectores de borda podem ser agrupados em duas classes (ZIOU e TABBONE, 1998): 1- autônomos: estes detectores não usam conhecimento a priori sobre a imagem ou borda a ser detectada,

sendo que uma característica em comum dos detectores autônomos é o processamento local dos dados da imagem. Exemplos: os detectores de borda de Sobel e Canny; e 2- contextual: estes detectores têm sido propostos para aplicação em conteúdos mais específicos de imagem, incluindo dessa forma algum conhecimento a priori de objetos presentes na imagem. Exemplos: detectores baseados em contorno ativo (snakes) (KASS et al., 1987) e programação dinâmica (BALLARD e BROWN, 1982).

A metodologia a ser apresentada abaixo faz parte da classe dos detectores contextuais. Novamente são apresentados detalhes essenciais relacionados com a função custo do método e argumentos sobre a aplicabilidade do algoritmo de otimização por programação dinâmica. Detalhes adicionais podem ser encontrados em BALLARD e BROWN (1982).

A definição da função custo depende da determinação a priori da imagem de orientação do gradiente ($d(p_i)$, sendo p_i a posição de um pixel da imagem) e da imagem de magnitude do gradiente ($m(p_i)$), procedimentos bastante comuns em textos básicos de processamento de imagens digitais. A função custo é definida com base em duas características das bordas que se pretende detectar. A primeira característica é básica em qualquer detector de bordas, pela qual as bordas são constituídas de uma seqüência de pixels associados com máximos locais do gradiente. A segunda característica impõe uma restrição importante sobre as bordas que se pretende detectar, segundo a qual as mesmas devem ser suaves. Assume-se então que as bordas possuem aspecto de feições suaves. Assim, como as rodovias são geralmente feições suaves, esta característica é favorável à detecção de bordas que delimitam rodovias.

Do exposto, a função custo pode ser definida a partir das seguintes parcelas:

- Soma das magnitudes do gradiente ao longo dos pontos que definem uma borda: como estes pontos correspondem a máximos locais do gradiente, esta soma deve ser máxima; e
- Soma das curvaturas locais ao longo dos pontos que definem uma borda: como se quer forçar que as bordas sejam suaves, esta soma deve ser mínima.

Matematicamente, tendo em vista as características das bordas que se quer detectar, a função custo é formulada como segue:

$$g(p_1, \dots, p_2) = \sum_{i=1}^n m(p_i) + \alpha \sum_{i=1}^{n-1} |d(p_i) - d(p_{i+1})|, \quad \text{com } \alpha < 0 \quad (31)$$

onde,

- $| \cdot |$ é o operador valor absoluto;
- $p_i, i= 1, \dots, n$, são pixels vizinhos de uma borda qualquer; e
- $g(p_1, \dots, p_2)$ é a função custo, que deverá ser máxima para seqüências de pixels descrevendo bordas.

Notar que o parâmetro α é definido como negativo, justamente para forçar a detecção de bordas suaves. Em outras palavras, seqüências de pixel com maior curvatura serão mais penalizadas. Observando a equação 31, conclui-se facilmente que a mesma pode ser decomposta na forma que segue,

$$g(p_1, \dots, p_2) = g_1(p_1, p_2) + \dots + g_{n-1}(p_{n-1}, p_n) \quad (32)$$

A equação 32 mostra que somente dois pontos de borda estão interrelacionados simultaneamente na função custo. Em outras palavras, apenas quatro variáveis estão interrelacionadas simultaneamente, permitindo concluir que a equação 31 possui a mesma estrutura da equação 4. Dessa forma, a programação dinâmica é o algoritmo apropriado para resolver o problema de extração de bordas colocado.

4 CONSIDERAÇÕES FINAIS

Este artigo apresentou o algoritmo de programação dinâmica em seus aspectos teórico e algorítmico, bem como mostrou sua aplicabilidade em problemas de extração de feições em imagens digitais de diversas naturezas. Em especial, foram apresentados dois exemplos de aplicação em problemas relacionados com a coleta de informações espaciais. No que se refere ao primeiro exemplo, foi apresentada uma metodologia para a extração de rodovias em imagens de baixa resolução (pixels > 2 m). Como pode ser constado em DAL POZ (2000) e DAL POZ e AGOURIS (2001), esta metodologia é útil também para aplicação em imagens de média resolução (pixels entre 0,7 m e 2 m). Entretanto, modificações realizadas na função custo (VALE e DAL POZ, 2003) possibilitaram a aplicação da metodologia em imagens aéreas de alta resolução (pixels < 0,7 m), permitindo a extração acurada do eixo de rodovia. Referente ao método contextual para detecção de bordas, não se tem disponível

implementação computacional para realização de testes. Entretanto, embora nenhum teste tenha sido realizado, a metodologia para extração de rodovias pode ser utilizada, à princípio sem modificação, para a extração de bordas em uma imagem de magnitude do gradiente. Isto é teoricamente possível porque as bordas, que são descontinuidades numa imagem qualquer, manifestam-se como linhas na correspondente imagem de magnitude do gradiente, isto é, da mesma forma que rodovias em imagens de baixa resolução.

5 REFERÊNCIAS BIBLIOGRÁFICAS

- AGOURIS, P., GYFTAKIS, S., STEFANIDIS, A. **Using a fuzzy supervisor for object extraction within an integrated geospatial environment.** In SYMPOSIUM ON OBJECT RECOGNITION AND SCENE CLASSIFICATION FROM MULTISPECTRAL AND MULTISENSOR PIXELS, v. 32, 1998, Columbus, Ohio. International Archives of the Photogrammetry and Remote Sensing, Columbus - EUA, 1998. p. 191-195.
- BALLARD, D. H.; BROWN, C. M. **Computer Vision.** Englewood Cliffs, New Jersey: Prentice Hall, 1982.
- BELLMAN, C. J., SHORTIS, M. R. **A machine learning approach to building recognition in aerial photographs.** In: ISPRS COMMISSION III, SYMPOSIUM 2002, PCV02 (Part A), v. 34, 2002, Graz - Austria. International Archives of the Photogrammetry, Remote Sensing and Information Sciences, Graz - Austria, 2002. p. 50-54.
- DAL POZ, A. P. **Processo automático para reconhecer rodovias georeferenciadas de uma fotografia aérea digitalizada.** Relatório FAPESP de Pós-Doutorado, Universidade do Maine, EUA, 225 p., 2000.
- DAL POZ, A. P.; AGOURIS, P. **Um algoritmo de otimização global para a extração de rodovias de imagens digitais.** Revista Brasileira de Cartografia, Rio de Janeiro 53, p. 65-72, 2001.
- FISCHLER, M. A., TENENBAUM, J. M., WOLF, H. C. **Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique.** Computer Graphics and Image Processing, v. 15, p. 201-223, 1981.
- GRUEN, A.; LI, H. **Road extraction from aerial and satellite images by dynamic programming.** ISPRS Journal of Photogrammetry and Remote Sensing, v. 50, n. 4, p. 11-20, 1995.
- GRUEN, A.; LI, H. **Semi-automatic linear feature extraction by dynamic programming and LSB-snakes.** Photogrammetric Engineering and Remote Sensing, v. 63, n. 8, p. 985-995, 1997.
- KASS, M., WITKIN, A., TERZOPOULOS, D. (1987). **Snakes: Active Contour Models.** In: 1ST INT. CONF. ON COMPUTER VISION, London, pp. 259-268.
- KOVALEVSKII, V. A. (1967), **An optimal recognition algorithm for some sequences of patterns.** Cybernetics, 3(4), pp. 62-66.
- LI, H. **Semi-automatic road extraction from satellite and aerial images.** PhD thesis, Report No. 61, Institute of Geodesy and Photogrammetry, ETH-Zurich, Switzerland, 161p., 1997.
- MONTANARI, U. **On the optimal detection of curves in noisy pictures.** Communications of the ACM, v. 14, n. 5, p. 335-345, 1971.
- MARTELLI A. (1972), **Edge detection using heuristic search methods.** Computer graphics and Image Processing, 1(12), pp. 169-182.
- MARTELLI, A. (1976), **An application of heuristic search methods to edge and contour detection.** Communication of the ACM, 19(2), pp. 73-83.
- MORTENSEN, E. N., BARRETT, W. A. **Intelligent Scissors for image composition.** SIGGRAPH 95, 1995, Los Angeles - EUA. 22nd Annual ACM Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, 1995. p. 191-198.

YAMADA, H., MERRITT, C., KASVAND, T. **Recognition of kidney glomerulus by dynamic programming matching method.** IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 10, n. 5, p. 731-737, 1988.

VALE, G., DAL POZ, A. P. **Metodologia modificada de programação dinâmica para a extração acurada do eixo de rodovia em imagens digitais.** Revista Brasileira de Cartografia – RBC, Rio de Janeiro, n. 55/1, junho de 2003.

ZIOU, D. TABBONE, S. **Edge detection techniques- an overview.** International Journal of Pattern Recognition and Image Analysis, v.8 pp 537-559, 1998.