

Um Algoritmo otimizado para o cálculo do gabarito da dilatação exata

MAURÍCIO FALVO¹
ODEMIR MARTINEZ BRUNO¹

USP - Universidade de São Paulo
ICMC - Instituto de Ciências Matemática e de Computação
SCC - Departamento de Ciências de Computação
Cx Postal 668 - CEP 13560-970 São Carlos (SP)
¹(mfalvo,bruno)@icmc.usp.br

Resumo. Este artigo apresenta uma proposta para a geração do gabarito das distâncias euclidianas exatas. O método desenvolvido se baseia no algoritmo de construção de circunferências, o qual faz uso de simetria. O método aqui descrito, realiza o mapeamento das distâncias euclidianas exatas de uma matriz quadrada de tamanho ímpar. Também são apresentados os resultados de um experimento comparativo, demonstrando um bom desempenho do método proposto.

Palavras-Chave: transformada da distância, dilatação euclidiana, simetria.

An optimized algorithm to yield a exact dilatation template

Abstract. This paper presents a novel algorithm to yield the exact Euclidean distance template. The proposed method is based on the circle symmetry algorithm and it performs the exact Euclidean distance mapping of an odd sized square matrix. There are shown results and a comparative experiment that illustrates the optimal performance of the novel method.

Keywords: exact distance transform, Euclidean dilatation, symmetry.

(Received June 3, 2006 / Accepted October 11, 2006)

1 Introdução

A transformada da distância é um método que possibilita obter a distância de um ponto \mathbf{P} em relação a qualquer outro ponto contido no mesmo espaço ao qual pertence \mathbf{P} [2], sendo a distância euclidiana um caso particular das distâncias de Minkowsky [9]. Este tipo de distância é largamente utilizada, em diversas aplicações, por ser uma maneira natural de como o homem percebe o espaço à sua volta. Recentemente foi proposto um método simples e exato para o cálculo da transformada da distância, baseada em dilatações exatas [3]. Seu principal componente é o gabarito da dilatação exata, que consiste de uma matriz ímpar, cujos elementos apresentam a distância euclidiana do elemento de

coordenadas (i, j) ao elemento central da matriz. Para isso, o cálculo do gabarito de dilatação exata é o primeiro passo do algoritmo, gerando um razoável custo computacional. Neste trabalho é apresentado um novo algoritmo otimizado para a geração do gabarito da dilatação exata, visando diminuir o tempo de cálculo.

A transformada da distância é um método numérico que apresenta aplicação direta em áreas como: robótica - no planejamento de trajetórias [11] e medicina - em neuromorfologia [4], entre outras. Além disso a transformada da distância pode ser utilizada como base para desenvolvimento de diversos métodos.

Em visão computacional e processamento de imagens, diversos algoritmos fazem uso intenso da transfor-

mada da distância, como os algoritmos de afinamento e esqueletização [2, 1, 6]; algoritmos de análise de complexidade de formas por dimensão fractal [4, 5], baseada no método de "Minkowsky-Sausage"[9] e algoritmos para segmentação, separação de objetos conexos [6] [8] e detecção de borda, como "watershed"[10].

1.1 O gabarito da dilatação exata

O primeiro passo no método da transformada da distância, é a obtenção da tabela de distâncias euclidianas, a partir do centro de uma matriz. A Figura 1a mostra o valor do elemento central com valor zero (informando a distância relativa de P a ele mesmo). Os demais pontos cobertos pela matriz têm sua distância relativa a P, determinada pela equação 1.

$$d = \sqrt{(c_i - i)^2 + (c_j - j)^2} \quad (1)$$

Por uma questão de maior facilidade de visualização das matrizes, e para a manipulação de números inteiros, não foram extraídas as respectivas raízes quadradas das distâncias assim, a distância euclidiana **D**, determinada pelo par ordenado (i, j) da matriz, fica definida por (2), onde c_i e c_j são os índices do elemento central da matriz $N \times N$, onde N é ímpar.

$$D = (c_i - i)^2 + (c_j - j)^2 \quad (2)$$

Pode-se observar que a geração das distâncias é um procedimento simples, pois cada elemento da matriz assume o valor da distância definida pelos seus respectivos índices (Figura 1b). Entretanto, ainda se faz necessário que as distâncias geradas e suas respectivas coordenadas da matriz, sejam armazenadas em uma estrutura de fácil recuperação - para que se possa saber quais coordenadas de pontos estão a uma distância **D** de **P** [4]. Esta estrutura deverá ser ordenada em função da distância (crescente) e deverá conter para cada elemento da estrutura a distância e a respectiva lista de coordenadas que a geram; conforme é exemplificado na Figura 2.

Para a obtenção dessa estrutura, seria necessário a manipulação de N^2 elementos da matriz, e o agrupamento de todas as coordenadas que geram uma mesma distância, em um único vetor *Vcd*. A proposta, aqui apresentada, visa otimizar o algoritmo de geração da tabela de distâncias euclidianas, atuando de duas formas: reduzindo o número de elementos a ser indexado - o que representou uma redução de alocação de memória considerável; e reduzindo o esforço de agrupamento de coordenadas de distâncias iguais - reduzindo o custo computacional envolvido na ordenação e agrupamento das coordenadas por distâncias.

18	13	10	9	10	13	18	-3,3	-2,3	-1,3	0,3	1,3	2,3	3,3
13	8	5	4	5	8	13	-3,2	-2,2	-1,2	0,2	1,2	2,2	3,2
10	5	2	1	2	5	10	-3,1	-2,1	-1,1	0,1	1,1	2,1	3,1
9	4	1	0	1	4	9	-3,0	-2,0	-1,0	0,0	1,0	2,0	3,0
10	5	2	1	2	5	10	-3,-1	-2,-1	-1,-1	0,-1	1,-1	2,-1	3,-1
13	8	5	4	5	8	13	-3,-2	-2,-2	-1,-2	0,-2	1,-2	2,-2	3,-2
18	13	10	9	10	13	18	-3,-3	-2,-3	-1,-3	0,-3	1,-3	2,-3	3,-3

(a)

(b)

Figura 1: (a) Matriz com os valores das distâncias determinadas pelos respectivos índices das células; (b) Representação da matriz com os respectivos índices das células.

I	D	Vetor de Coordenadas (Vcd)
0	0	{(0,0)}
1	1	{(0,1), (1,0), (-1,0), (0,-1)}
2	2	{(1,1), (1,-1), (-1,-1), (-1,-1)}
3	4	{(2,0), (0,2), (-2,0), (0,-2)}
4	5	{(1,2), (2,1), (1,-2), (2,-1), (-1,2), (-2,1), (-1,-2), (-2,-1)}
5	8	{(2,2), (-2,2), (2,-2), (-2,-2)}
6	9	{(3,0), (0,3), (-3,0), (0,-3)}
7	10	{(1,3), (3,1), (1,-3), (3,-1), (-1,3), (-3,1), (-3,-1), (-1,-3)}
8	13	{(2,3), (3,2), (2,-3), (3,-2), (-2,3), (-3,2), (-3,-2), (-2,-3)}
9	18	{(3,3), (3,-3), (-3,3), (-3,-3)}

Figura 2: Exemplo de uma estrutura de gabarito de distâncias euclidianas, ordenado pela distância - I é o índice do elemento na tabela, D é a respectiva distância e a terceira coluna representa o vetor de coordenadas.

2 O Método da Simetria

O método se baseou na geração de coordenadas de pontos de uma circunferência, utilizando-se de simetria, a partir de pontos contidos em uma seção de $\frac{1}{8}$ de circunferência [7]. Utilizando este princípio, pode-se otimizar o tempo de geração da tabela de distâncias euclidianas, ordenadas pela distância, considerando-se as características de simetria de uma matriz quadrada de largura ímpar. Assim, para uma matriz de tamanho 7×7 tem-se a matriz apresentada na Figura 1(a). Observando-se com mais atenção, pode-se verificar que todas as possíveis distâncias distintas se concentram na metade do primeiro quadrante, como é apresentada na Figura 3. O conjunto de coordenadas dos elementos assinalados na matriz da Figura 3 é aqui denominado de conjunto de coordenadas bases, o qual é definido por (3), onde i e

$j \in N$.

$$Cb = (i, j) | 0 \leq j \leq dim(Mn \times n); j \leq i \leq dim(Mn \times n) \quad (3)$$

18	13	10	9	10	13	18
13	8	5	4	5	8	13
10	5	2	1	2	5	10
9	4	1	0	1	4	9
10	5	2	1	2	5	10
13	8	5	4	5	8	13
18	13	10	9	10	13	18

Figura 3: Conjunto das células de distâncias distintas.

Considerando-se que, para manter sua simetria, a matriz de distâncias euclidianas deve sempre ter tamanho ímpar N , o qual pode ser definido por: $N = (dim * 2) + 1$. Onde dim é o comprimento em número de elementos de matriz, do elemento central até à sua borda, excluindo o próprio elemento central. No caso da matriz da Figura 3 o valor de dim é três. Dessa forma $N = (3 * 2) + 1$, de onde se obtêm o tamanho da matriz 7×7 , cujo centro que contém o valor zero, está na célula (4, 4), o qual de ser determinado em função de dim por: $C = dim + 1$.

A quantificação do número total de elementos de coordenadas base é determinada pelo número de elementos do centro da matriz à borda, isto é, $dim + 1$. deste empilhamento é de comprimento $dim + 1$, o número total de elementos de distância base (Ndb) é dado por (eq. 4):

$$Ndb = \sum_{k=0}^{dim} (dim+1-k) = 0.5dim^2 + 1.5dim + 1 \quad (4)$$

Para medir o desempenho do método, pode-se comparar a razão entre o número de elementos de distância base, pelo número total de elementos da matriz. Essa razão indica qual o percentual de redução ($Rn\%$) do número de elementos a serem manipulados para a construção da tabela de distância euclidiana (eq. 5).

$$Rn\% = \frac{Ndb}{(2dim + 1)^2} \quad (5)$$

Quando dim é 1, a matriz será de 3×3 e o número de distâncias base será 3. O que corresponde a aproximadamente 33% do número total de elementos da matriz. Quando dim tende a infinito, teremos (eq. 6):

$$\lim_{dim \rightarrow \infty} \frac{Ndb}{(2dim + 1)^2} = 12,5\% \quad (6)$$

Nesta primeira etapa apresentada, foi possível obter uma considerável redução no número distâncias a serem geradas, o que representa um fator de ganho em termos de esforço computacional. Na segunda etapa desse trabalho, são obtidas todas as coordenadas possíveis de cada distância gerada, fazendo-se uso de simetria. Para facilitar a análise da característica de simetria da matriz, as Figuras 4(a-d) apresentam a mesma matriz da Figura 3, destacando os grupos de valores das distâncias. A ordem de análise dos grupos de números é seguida pela seqüência das Figuras: 4a, 4b, 4c e 4d.

- Primeiro grupo (Figura 3a): ocorrência. (\mathbf{x}, \mathbf{y}) .
Para $\{(x, y) | x = y = 0\}$;
- Segundo grupo (Figura 3b): ocorrências $(\mathbf{x}, \mathbf{y}), (\mathbf{y}, \mathbf{x}), (-\mathbf{x}, \mathbf{y})$ e $(\mathbf{x}, -\mathbf{y})$.
Para: $\{(x, y) | (x = i, y = 0), \text{ onde } 0 < i \leq dim\}$
- Terceiro grupo (Figura 3c): ocorrências $(\mathbf{x}, \mathbf{y}), (\mathbf{x}, -\mathbf{y}), (-\mathbf{x}, -\mathbf{y})$ e $(-\mathbf{x}, \mathbf{y})$.
Para: $\{(x, y) | x = y = i, \text{ onde } 0 < i \leq dim\}$
- Quarto grupo (Figura 3b): ocorrências: $(\mathbf{x}, \mathbf{y}), (\mathbf{y}, \mathbf{x}), (-\mathbf{x}, \mathbf{y}), (-\mathbf{y}, \mathbf{x}), (-\mathbf{x}, -\mathbf{y}), (-\mathbf{y}, -\mathbf{x}), (\mathbf{x}, -\mathbf{y})$ e $(\mathbf{y}, -\mathbf{x})$.
Para: $\{(x, y) | x \neq y \text{ e } x, y \neq 0\}$

Identificadas as situações de simetria, pode-se agora definir um algoritmo de geração do gabarito da transformada da distância euclidiana, por simetria, a partir do conjunto de regras de construção das coordenadas de distância base:

1. Adicionar a coordenada base (x, y) no Vcd
2. Se $x \neq y$ OU $y \neq 0$
adiciona-se ao conjunto Vcd o par $(-x, y)$
3. Se $x \neq y$
adiciona-se ao conjunto Vcd os pares (y, x) e $(y, -x)$
4. Se $y \neq 0$
adiciona-se ao conjunto Vcd os pares $(x, -y)$ e $(-x, -y)$
5. Se $x \neq y$ E $y \neq 0$
adiciona-se ao conjunto Vcd os pares $(-y, x)$ e $(-y, -x)$

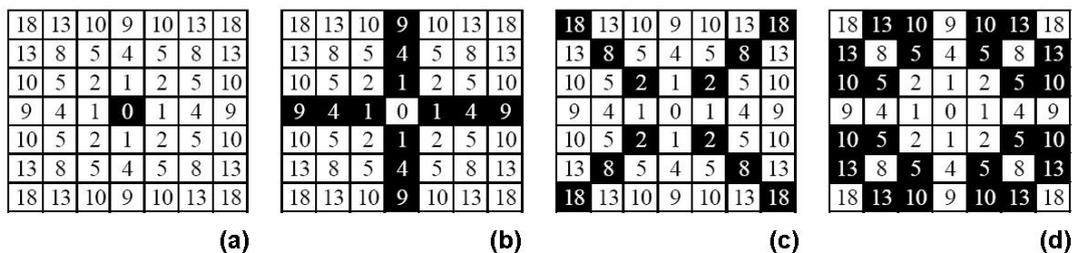


Figura 4: Regiões de simetria: (a) grupo 1; (b) grupo 2; (c) grupo 3; (d) grupo 4.

2.1 Algoritmo para a geração da tabela de distâncias euclidianas

1. Para cada coordenada base (x,y):

- gerar a distância euclidiana da coordenada base, e o Vcd, utilizando-se das Regras de Construção do Vetor de Coordenadas da Distância.

- Inserir na Tabela de Distâncias Euclidianas a distância gerada e seu respectivo vetor de coordenadas Vcd.

2. Ordenar a Tabela de Distâncias Euclidianas pela distância.

3. Varrer a Tabela de Distâncias Euclidianas Ordenada, atualizando o índice I (Figura 2) com o primeiro valor 0(zero) para a primeira distância, e incrementando o valor deste índice para todo elemento da tabela que possuir a distância diferente do elemento anterior. Caso o elemento tenha distância igual ao anterior, deve-se agrupar o Vcd deste ao Vcd do elemento anterior. Este passo se faz necessário, pois as coordenadas bases podem gerar distâncias iguais. Um exemplo que pode ser citado, é a distância 25 que pode ser gerada pela coordenada base (0, 5) e seus simétricos, mas também pode ser gerada por outra coordenada base (3, 4) e seus simétricos, apesar de não possuírem (0, 5) e (3, 4) nenhuma relação de simetria. O índice I comumente é utilizado para mapear os níveis de distâncias em vez de se utilizar o valor da distância propriamente dito (Figura 1a).

3 Resultados Experimentais

Os testes de tempo de geração do gabarito das distâncias euclidianas, foram feitos em um computador Pentium IV 1.5GHz com 512MB de RAM. Foi feito um teste comparativo com um algoritmo convencional e os resultados são apresentados na Tabela 1. Como pode-se observar, o algoritmo de simetria apresentou um bom

desempenho em relação ao algoritmo convencional. Fazendo-se a média dos tempos de cada tipo de algoritmo e calculando-se a razão entre elas, tem-se como resultado a constatação de um tempo de 17 vezes menor para o algoritmo por simetria, mesmo possuindo a mesma complexidade que o algoritmo convencional - $O(n^2)$. Esta melhora no desempenho atribui-se à redução do número de interações obtida graças à observação e aproveitamento das características de simetria dos elementos, não presentes no algoritmo convencional.

Tabela 1: A coluna N indica o tamanho N da matriz distância; as colunas Alg. Simetria e Alg. Convencional indicam o tempo gasto em milissegundos para os vários valores de N.

N	Alg.Simetria	Alg.Convencional
101	20	40
251	65	531
501	420	6.300
1001	5.980	184.460
1251	18.607	323.195
1501	41.640	717.862
1751	80.526	1.356.871
2001	139.721	2.343.650

4 Conclusão

A partir das propriedades de simetria, utilizadas para a construção de uma circunferência, foi possível identificar as propriedades de simetria de uma matriz distância, que por sua vez permitiu:

- identificar e quantificar o número de distâncias bases, sendo que a quantidade de distâncias necessárias a serem geradas cai, no pior caso, para 33% do número de elementos da matriz (3x3) do montante original. Essa tende a cair rapidamente para 12, 5% (Figura 5), a medida que a largura da matriz aumenta e;
- gerar, a partir de uma coordenada base, de uma dada distância, todas as demais coordenadas onde

essa distância pode ser encontrada na matriz.

O algoritmo apresentado tornou a geração do gabarito da dilatação exata, 17 vezes mais rápida, em relação ao algoritmo convencional devido: a sensível redução das distâncias a serem calculadas e ordenadas e à diminuição do esforço de agrupamento das coordenadas de mesma distância.

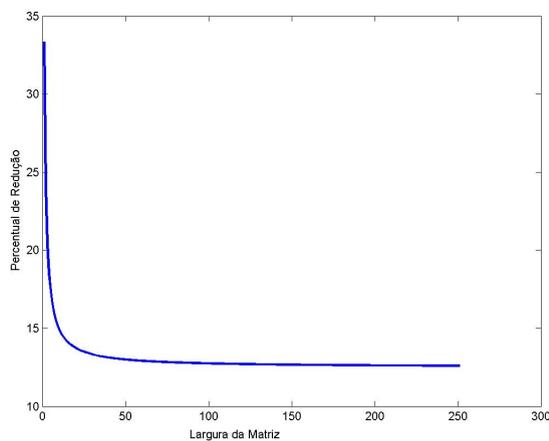


Figura 5: Gráfico Largura de Matriz X Percentual de Redução./

5 agradecimentos

Os autores agradecem ao CNPq processo Nº 303746/2004-1 e a CAPES pelo apoio fornecido.

Referências

- [1] Baxes, G. A. *Digital Image Processing Principles and Applications*. John Wiley and Sons, 2005.
- [2] Chassery, J. M. and Annick, M. *Géometrie Discrète en analyse d'images*. Hermes, 1991.
- [3] da Fontoura Costa, L. Gauss' law in image processing and analysis via fast numerical calculation of Vector fields. *Real-Time Imaging*, 5(4):243–251, 1990.
- [4] da Fontoura Costa, L. Robust Skeletonization through Exact Euclidean Distance Transform and its Application to Neuromorphometry. *Real-Time Imaging*, 6:415–431, 2000.
- [5] de Oliveira Plotze, R., Pádua, J. G., Falvo, M., Bernacci, L. C., Oliveira, G. C. X., Vieira, M. L. C., and Bruno, O. M. Leaf shape analysis by the multiscale minkowski fractal dimension, a new morphometric method: a study in passiflora L. (Passifloraceae). *Canadian Journal of Botany-Revue Canadienne de Botanique*, 83(3):287–301, 2005.
- [6] Dougherty, E. R. *An Introduction to Morphological Image Processing*. SPIE Optical Engineering Press, 1992.
- [7] Foley, J. D., van Dam, A., Ferner, S. K., and Hugher, J. *Computer Graphics Principles and Practice*. Addison Wesley Publishing Company, 1992.
- [8] Serra, J. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [9] Tricot, C. *Curves and Fractal Dimension*. Springer-Verlag, 1995.
- [10] Wright, A. and Acton, S. Watershed pyramids for edge detection. pages 578–581.
- [11] Zelinsky, A. A Mobile Robot Navigation Exploration Algorithm. *IEEE Transactions of Robotics and Automation*, 8(6):707–717, 1992.