

# Erosões e Dilatações Morfológicas Binárias Sequenciais Rápidas

Anderson Fraiha Machado<sup>1</sup>, Ronaldo Fumio Hashimoto<sup>1</sup>

<sup>1</sup>Instituto de Matemática e Estatística – Universidade de São Paulo (USP)  
Rua do Matão, 1010 – Cidade Universitária – CEP 05508-900  
São Paulo – SP – Brazil

{dandy, ronaldo}@ime.usp.br

**Abstract.** *This work proposes new fast methods for computing binary morphological erosions and dilations. Such methods are based on preprocessing techniques and new concepts presented and introduced by this research, such as Density Transformation and Set of Shells. The result of these preprocessing techniques is a speed up for computing binary erosions and dilations and also a new compact representation for binary images.*

**Resumo.** *Este trabalho propõe novos métodos para calcular a erosão e a dilatação morfológica binária rapidamente, fundamentados em técnicas de pré-processamento em tempo linear e novos conceitos apresentados e introduzidos nesta pesquisa. Como resultado, estes pré-processamentos tornam mais eficientes os algoritmos de erosão e dilatação, além de apresentar uma representação compacta para imagens binárias.*

## 1. Introdução

A *Morfologia Matemática* (MM), fundada por J. Serra e G. Matheron [Serra 1982] entre os anos de 1960 e 1969, vem se tornando uma disciplina muito importante na área de Processamento de Imagens Digitais (PID). Basicamente, a MM é uma teoria que estuda transformações de formas presentes nas imagens. Em particular, transformações entre imagens binárias são de especial interesse em MM e podem ser vistas como sendo mapeamentos entre conjuntos [Banon and Barrera 1994] e por esta razão são chamados de *operadores de conjunto*. Uma classe de operadores de conjuntos conhecida como operadores *invariantes por translação* (i.t.) são muito importantes em PID [Barrera and Salas 1996].

Um dos paradigmas centrais da MM é decompor os operadores i.t. em termos de dilatação, erosão e outras operações usuais de conjunto. Sendo assim, a *dilatação* e a *erosão* são tidos como pilares para a representação de operadores invariantes por translação. Existem vários métodos para calcular eficientemente a erosão e a dilatação, como decompor elementos estruturantes convexos em uma seqüência de erosões/dilatações por elementos estruturantes primitivos (com janelas  $3 \times 3$ ) centrados na origem [Xu 1991]; utilizar *Run-Length Encoding* para compactar intervalos contínuos de pontos da imagem [Kim et al. 2005]; aproveitar o paralelismo do processador em operações de E/OU lógico efetuadas sobre vetores de bits [Brambor 2002] e outros. Todavia, por mais rápidos que sejam todos estes algoritmos, ainda não são bastante eficientes quando aplicados em imagens de alta resolução e/ou ruidosas.

Este trabalho consiste em apresentar novos métodos que executam rapidamente a *erosão morfológica binária* e a *dilatação morfológica binária*, especialmente quando aplicados em imagens de alta resolução e/ou ruidosas.

## 2. Fundamentos Matemáticos

Uma imagem binária em  $\mathbb{Z}^n$  pode ser definida como uma função  $f : \mathbb{Z}^n \rightarrow \{0, 1\}$ . Ou seja, uma imagem binária contém apenas dois valores: o *background* (ou *fundo*) da imagem ou o *foreground* (ou *objeto*) da imagem binária. Uma imagem binária  $f$  também pode ser representada como um subconjunto  $X$  de  $\mathbb{Z}^n$ , de modo que um ponto  $x \in \mathbb{Z}^n$  pertence a  $X$  se, e somente se,  $x$  é um ponto do objeto de  $f$ , ou seja  $f(x) = 1$ . Assim, o conjunto  $X$  da imagem binária  $f$  é definido como  $X = \{x \in \mathbb{Z}^n : f(x) = 1\}$

Uma *relação de adjacência* sobre o espaço  $\mathbb{Z}^n$  pode ser representada por um conjunto de vetores  $V \subseteq \mathbb{Z}^n$  e definidas da seguinte maneira: Dados dois pontos  $p, q \in \mathbb{Z}^n$ , dizemos que  $q$  é  $V$ -*adjacente* a  $p$  se, e somente se, existe um  $v \in V$  tal que  $q = p + v$ . O conjunto  $V$  é chamado de *conjunto de adjacência*. As complexidades dos métodos deste trabalho dependem do tamanho do conjunto  $V$ . Tipicamente, em aplicações gerais (no espaço  $\mathbb{Z}^2$ ) o conjunto  $V$  é restrito a uma janela  $3 \times 3$  centrada na origem.

### 2.1. Morfologia Matemática

Visto que uma imagem binária pode ser representada por um subconjunto de  $\mathbb{Z}^n$ , vamos definir os principais operadores de conjunto pertinentes à MM.

Seja  $h \in \mathbb{Z}^n$ . O conjunto  $X_h$ , definido por  $X_h = \{x + h : x \in X\}$ , é uma *translação* de  $X$  por  $h$ . O conjunto  $X^t = \{-x : x \in X\}$  é a *transposta* de  $X$ .

Um operador de conjunto  $\psi$  é dito ser *invariante por translação* (i.t.) se, e somente se,  $\forall h \in \mathbb{Z}^n, \forall X \subseteq \mathbb{Z}^n, \psi(X_h) = \psi(X)_h$ .

Sejam  $X, Y, B \subseteq \mathbb{Z}^n$ . Os operadores de conjunto i.t.  $\delta_B$  e  $\varepsilon_B$  são, respectivamente, a dilatação e a erosão por  $B$  definidas por:

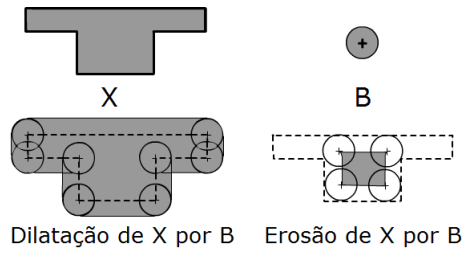
$$\delta_B(X) = \bigcup_{b \in B} X_b = \{x \in \mathbb{Z}^n : \exists b \in B \text{ tal que } x - b \in X\} \text{ e}$$

$$\varepsilon_B(X) = \bigcap_{b \in B} A_{-b} = \{x \in \mathbb{Z}^n : \forall b \in B \text{ tal que } x + b \in A\}.$$

O conjunto  $B$  é chamado de *elemento estruturante* (EE). Existe também uma definição alternativa para dilatação e erosão como segue:

$$\delta_B(X) = \{h : X \cap B_h^t \neq \emptyset\} \text{ e } \varepsilon_B(X) = \{h : B_h \subseteq X\}.$$

A Figura 1 mostra uma interpretação geométrica da dilatação e erosão de uma imagem binária  $X$  por um elemento estruturante  $B$ . Nesta figura, a dilatação de  $X$  por  $B$  equivale à “deslizar” o elemento estruturante  $B$  sobre cada ponto da imagem  $X$ , enquanto que a erosão de  $X$  por  $B$  equivale à marcar a origem do elemento estruturante  $B$  em todas as posições em que este se “encaixa” em  $X$ .



**Figura 1. Exemplos de dilatação e erosão de  $X$  por  $B$ . O sinal  $+$  indica a origem.**

Sabe-se que a dilatação e a erosão são operadores duais conforme segue:

$$\delta_B(X) = (\varepsilon_{B^t}(X^c))^c,$$

onde  $X^c = \{x : x \notin X\}$  corresponde ao *complemento* de  $X$ .

Assim, a implementação eficiente de um destes operadores pode ser reduzida a outro operador dual bem como na composição de outros operadores morfológicos<sup>1</sup>.

### 3. Técnicas de Pré-Processamento

A motivação em pré-processar os conjuntos que representam as imagens binárias vem de estudos sobre técnicas de pré-processamento de padrões e textos, utilizadas em buscas de padrões (*string-matching*). Tais técnicas possibilitam obter uma representação mais compacta dos conjuntos pré-processados, e assim, desenvolver alguns algoritmos para calcular mais rapidamente a erosão e a dilatação. Espera-se que os algoritmos de pré-processamento consumam tempo *linear* (ou *quase linear*) no tamanho da entrada.

#### 3.1. Transformada da Densidade

A *Transformada da Densidade* (TD) se assemelha muito à Transformada da Distância [Borgefors 1986]. O que difere estas duas é o fato que a TD permite um conjunto de adjacência (vizinhança) *assimétrico*. A TD pode ser entendida como uma forma rápida para calcular *Transformada de Erosões* [Chen and Haralick 1995].

Sejam  $p, q \in \mathbb{Z}^n$  e  $V$  um conjunto de adjacência qualquer. A seqüência de pontos  $P = (p = p_1, p_2, \dots, p_m = q)$  é um  $V$ -caminho que *liga*  $p$  a  $q$  se para  $i = 1, 2, \dots, m - 1$ , o ponto  $p_{i+1}$  é  $V$ -adjacente a  $p_i$  e  $p_i \neq p_j$  sempre que  $i \neq j$ . O comprimento do  $V$ -caminho  $P$  é definido como  $|P| - 1$ .

A  $V$ -densidade de um ponto  $p$  a um ponto  $q$ , denotada por  $d^V(p, q)$ , é o comprimento do menor  $V$ -caminho que liga  $p$  a  $q$ . Caso tal  $V$ -caminho não exista, dizemos que  $d^V(p, q) = +\infty$ . Observe que a  $V$ -densidade de  $p$  a  $q$  é uma *distância* somente se o conjunto de adjacência  $V$  é simétrico.

Seja  $X \subseteq \mathbb{Z}^n$ . A  $V$ -densidade de um ponto  $p$  em relação a  $X$ , denotado por  $\Delta_X^V(p)$ , é definido como a menor densidade  $d^V(p, q)$ , com  $q \in X^c$ .

A TD é um método<sup>2</sup> de atribuição das densidades  $\Delta_X^V(p)$  a cada ponto  $p$  de  $X$ . Utilizando-se uma fila (FIFO) é possível implementar uma TD em tempo  $\Theta(|X| \cdot |V|)$ . Considerando  $|V|$  constante, temos uma TD em tempo linear.

<sup>1</sup>Qualquer operador i.t. pode ser descrito como uma seqüência de erosões e/ou dilatações.

<sup>2</sup>Maiores informações estão em <http://score.ime.usp.br/~dandy/mestrado.php>

Uma vez definida a  $V$ -densidade de um ponto  $p$  com relação ao conjunto  $X$ , o conjunto  $COB_X^V(p) \subseteq X$  é o conjunto de pontos  $V$ -cobertos por  $p$  em  $X$ , também conhecido como *disco morfológico* [Dougherty and Lotufo 2003], definido como

$$COB_X^V(x) = \{q \in X : d^V(x, q) < \Delta_X^V(x)\}.$$

O ponto  $\zeta \in X$  é dito ser uma  $V$ -casca de  $X$  se e somente se,  $\zeta$  é a origem de um disco maximal  $COB_X^V(\zeta)$ , ou seja, não existe um outro ponto  $p \in X$  tal que  $COB_X^V(\zeta) \subseteq COB_X^V(p)$ . Assim,  $\zeta$  é uma  $V$ -casca de  $X$  se e somente se, para todo  $v \in V^t$ , temos que  $\Delta_X^V(\zeta + v) \leq \Delta_X^V(x)$  (máximos locais da TD).

Supondo que todos os pontos de  $X$  sejam visitados anteriormente pela TD, podemos facilmente extrair o conjunto de cascas de um conjunto qualquer em tempo  $\Theta(|X| \cdot |V|)$ . Novamente, se considerarmos que  $|V|$  é um valor constante, então o algoritmo extrator de cascas consome tempo linear. Denote por  $Z^V(X)$  o conjunto de todas as  $V$ -cascas de  $X$ . Observe que este conjunto corresponde ao *esqueleto morfológico da imagem  $X$*  pelo conjunto  $V$  (ver [Dougherty and Lotufo 2003]).

## 4. Algoritmos de Erosão e Dilatação

Mediante todas estas definições e propriedades, seguem algumas propostas de novos métodos para calcular a erosão (na Seção 4.1) e a dilatação (na Seção 4.2) morfológica binária. Estes novos algoritmos são a contribuição deste trabalho.

### 4.1. Erosão com pressupostos de busca de padrão

Nesta seção, apresentaremos dois algoritmos que calculam a erosão rapidamente, utilizando a Transformada da Densidade e as Cascas do EE combinadas com técnicas de busca de padrões (*string-matching*) [Cormen et al. 2001].

Uma característica interessante presente em alguns algoritmos de busca de padrão em textos é a propriedade de realizar saltos (*jumps*) na busca, na proporção em que se garante que uma determinada seqüência consecutiva de caracteres do padrão não se encaixa (*miss-matching*) como subsequência no texto. Assim, o objetivo desta seção é reduzir o problema de realizar a erosão binária ao problema de busca de padrões textuais. Neste caso, o alfabeto  $\Sigma = \{0, 1\}$  corresponde aos pixels de uma imagem qualquer, o padrão corresponde ao EE e o texto corresponde a uma imagem de entrada.

Pela definição alternativa de erosão temos que  $\varepsilon_B(X) = \{h \in \mathbb{Z}^n : B_h \subseteq X\}$ ,  $X, B \subseteq \mathbb{Z}^n$ . Se garantirmos que  $\varepsilon_B(X) \subseteq X$  então o cálculo da erosão é reduzido a um problema de busca por ocorrências do elemento estruturante  $B$  dentro da imagem de entrada  $X$ . No entanto, sabe-se que se a origem está no elemento estruturante  $B$  então certamente  $\varepsilon_B(X) \subseteq X$ . Caso a origem não esteja em  $B$ , temos que  $\varepsilon_B(X) = (\varepsilon_{B-b}(X))_b$  para ponto qualquer  $b \in B$ , e a origem está em  $B_{-b}$ .

Tanto a imagem de entrada  $X$  quanto o elemento estruturante  $B$  são submetidos aos passos de pré-processamento utilizando o mesmo conjunto de adjacência  $V$ . O pré-processamento da TD e do conjunto de cascas é único, tanto para o EE, quanto para a imagem de entrada. Sendo assim, não é necessário recalculá-lo para os conjuntos  $B_h$ , para cada  $h$  distinto, visto que a TD é invariante por translação.

### 4.1.1. Erosão Direta

Dado um vetor  $v \neq o \in V^t$ . Tal vetor indica um sentido de propagação das densidades na TD. O algoritmo de erosão direta realiza buscas do EE sobre a imagem de entrada no mesmo sentido  $v$  em que se propaga as densidades na TD da imagem de entrada.

**Proposição 1** *Sejam  $X, B, V \subseteq \mathbb{Z}^n$ . Então  $B \subseteq X$  se, e só se, para toda  $V$ -casca  $\zeta$  de  $\mathcal{Z}^V(B)$  existe um ponto  $x$  em  $X$  tal que  $COB_B^V(\zeta) \subseteq COB_X^V(x)$ .*

Sendo assim, dado um  $h \in \mathbb{Z}^n$  de modo que para toda  $V$ -casca  $\zeta \in \mathcal{Z}^V(B_h)$  em  $B_h$  vale que  $COB_B^V(\zeta) \subseteq COB_X^V(x)$ , então  $h \in \varepsilon_B(X)$ . Tal conclusão nos leva a entender que não é necessário verificar se todos os pontos de  $B_h$  estão contidos em  $X$ : basta examinar as cascas de  $\mathcal{Z}^V(B_h)$ .

Pela contrapositiva da Proposição 1 temos: “Então  $B \not\subseteq X$  se, e só se, existe uma  $V$ -casca  $\zeta$  de  $\mathcal{Z}^V(B)$  onde para todo  $x \in X$  vale que  $COB_B^V(\zeta) \not\subseteq COB_X^V(x)$ .” Conseqüentemente, caso  $\Delta_{B_h}^V(\zeta) > \Delta_X^V(\zeta)$  para alguma casca  $\zeta$  de  $B_h$ , então  $B_h \not\subseteq X$ . Note que é possível conhecer de antemão para quais posições  $h$  temos que  $B_h \not\subseteq X$ . Nestas posições  $h$  não há necessidade do algoritmo de erosão direta efetuar comparações (ver Prop. 2). O algoritmo de erosão direta efetua uma busca do EE sobre a imagem de entrada, realizando eventuais saltos (*jumps*) sobre os pontos  $h \in X$  tais que  $B_h \not\subseteq X$ .

**Proposição 2** *Sejam  $X, B, V \subseteq \mathbb{Z}^n$ . Considere ainda  $\zeta \in \mathcal{Z}^V(B)$ . Se  $\Delta_X^V(\zeta) < \Delta_B^V(\zeta)$ , então existem pelo menos  $\Delta_B^V(\zeta) - \Delta_X^V(\zeta)$  pontos  $h \in \mathbb{Z}^n$  tais que  $B_h \not\subseteq X$ .*

Esta proposição nos indica a existência de pontos  $h \in \mathbb{Z}^n$  para os quais  $B_h \not\subseteq X$ . Tais pontos são facilmente localizáveis, uma vez que se conhece o conjunto de adjacência  $V$ . Podemos implementar o algoritmo de erosão direta gastando tempo proporcional a  $O(|X| + |B| + |X| \cdot |\mathcal{Z}^V(B)|)$  no pior caso.

### 4.1.2. Erosão Inversa

A erosão inversa efetua uma busca do EE sobre a imagem de entrada, marcando-se previamente as posições  $h$  em  $\mathbb{Z}^n$  das quais se tem certeza que  $B_h \subseteq X$ . Neste caso, o conjunto de adjacência utilizado nos passos de pré-processamento das imagens de entrada (TD e Conjunto de Cascas) deste algoritmo de erosão é um conjunto  $V \subseteq \mathbb{Z}^n$  que possui apenas um vetor  $v \in \mathbb{Z}^n$ . Todavia, neste caso o sentido da busca é contrário ao sentido da propagação das densidades na imagem de entrada.

**Proposição 3** *Sejam  $X, B, V \subseteq \mathbb{Z}^n$ . Se  $\forall \zeta$  de  $\mathcal{Z}^V(B)$  temos que  $\Delta_B^V(\zeta) \leq \Delta_X^V(\zeta)$  então existem  $\beta$  pontos  $h \in \mathbb{Z}^n$  tais que  $B_h \subseteq X$ , onde  $\beta = \min_{\zeta \in \mathcal{Z}^V(B)} \{\Delta_X^V(\zeta) - \Delta_B^V(\zeta)\} + 1$ .*

Tal propriedade acima nos garante a possibilidade do algoritmo conhecer de antemão quais posições  $h \in \mathbb{Z}^n$  garantimos que  $B_h \subseteq X$ . Novamente, em tais posições não há necessidade de se verificar  $B_h \subseteq X$ , e por isso pode-se realizar eventuais saltos (*jumps*) sobre os pontos  $h \in X$  tais que  $B_h \subseteq X$ . O consumo de tempo deste no pior caso também é  $O(|X| + |B| + |X| \cdot |\mathcal{Z}^V(B)|)$ .

## 4.2. Dilatação por Cascas

Este efetua a dilatação de uma imagem de entrada  $X$  por um elemento estruturante  $B$  usando basicamente as cascas de  $X$  e as cascas de  $B$ . Considere os seguintes resultados:

1. Sejam  $X, B \subseteq \mathbb{Z}^n$ . Considere que  $X = X_1 \cup X_2 \cup \dots \cup X_N$  e  $B = B_1 \cup B_2 \cup \dots \cup B_M$ . Então pela propriedade distributiva da dilatação (ver [Dougherty and Lotufo 2003]) temos que  $\delta_B(X) = \cup_{1 \leq i \leq N} \cup_{1 \leq j \leq M} \delta_{B_j}(X_i)$ .
2. Sejam  $Y, V \subseteq \mathbb{Z}^n$ . Por definição  $Y = \cup_{\zeta \in \mathcal{Z}^V(Y)} COB_Y^V(\zeta)$ .

Combinando ambos resultados, temos que:

$$\delta_B(X) = \bigcup_{\zeta_1 \in \mathcal{Z}^V(B)} \bigcup_{\zeta_2 \in \mathcal{Z}^V(X)} \delta_{COB_B^V(\zeta_2)}(COB_X^V(\zeta_1)),$$

para um conjunto de adjacência  $V$  qualquer.

Logo, podemos efetuar dilatações das cascas  $\mathcal{Z}^V(B)$  do EE pelas cascas  $\mathcal{Z}^V(X)$  da imagem de entrada através de uma simples atualização das densidades das cascas  $\zeta \in \mathcal{Z}^V(X)$ . Ao final, o algoritmo necessita de uma fase de propagação das densidades, recuperando cada ponto na imagem resultante  $\delta_B(X)$ . Considerando  $|V|$  constante, o consumo de tempo no pior caso deste algoritmo é  $\Theta(|X| + |B| + |\delta_B(X)| + |\mathcal{Z}(X)| \cdot |\mathcal{Z}(B)|)$ .

## 5. Resultados e Discussões

Esta seção visa apresentar alguns resultados experimentais dos algoritmos propostos aplicados em imagens bidimensionais, bem como compará-los com outros algoritmos.

A imagem de entrada é uma imagem que possui dimensão  $2500 \times 2500$  e possui quatro quadrantes. Dois quadrantes são compostas por quatro silhuetas, um quadrante é composto por um ruído sal & pimenta com distribuição uniforme e o último quadrante é composto por linhas igualmente bem espaçadas e orientadas a  $45^\circ$ . A Figura 2 mostra a imagem de entrada em um tamanho reduzido.

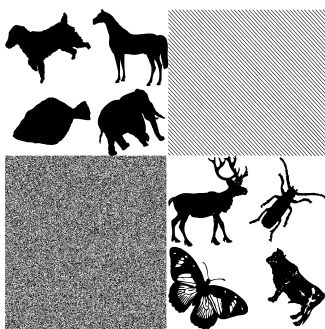


Figura 2. A imagem de entrada de dimensão  $2500 \times 2500$ .

Os algoritmos de erosão e dilatação utilizados neste experimento serão os seguintes: os que usam decomposição de EE (Alg. 3), os que usam RLE (Alg. 5), os que usam decomposição de EE e o vetor de bits (Alg. 7.1), dilatação que usa decomposição de EE e o contorno da imagem de entrada (Alg. 7.2), o algoritmo de erosão direta (Alg. A), o algoritmo de erosão inversa (Alg. B) e o algoritmo de dilatação por cascas (Alg. C). Implementações destes algoritmos e outros, bem

como uma inspeção mais detalhada deste e outros experimentos podem ser encontradas em <http://score.ime.usp.br/~dandy/mestrado.php>.

Os gráficos das Figuras 3 e 4 mostram os tempos médios de 10 execuções repetidas dos algoritmos de erosão e dilatação, respectivamente, usando a imagem de entrada proposta e quadrados de tamanho  $n$  como elementos estruturantes. Embora foram usados EEs quadrados, os algoritmos apresentados podem utilizar EEs arbitrários.

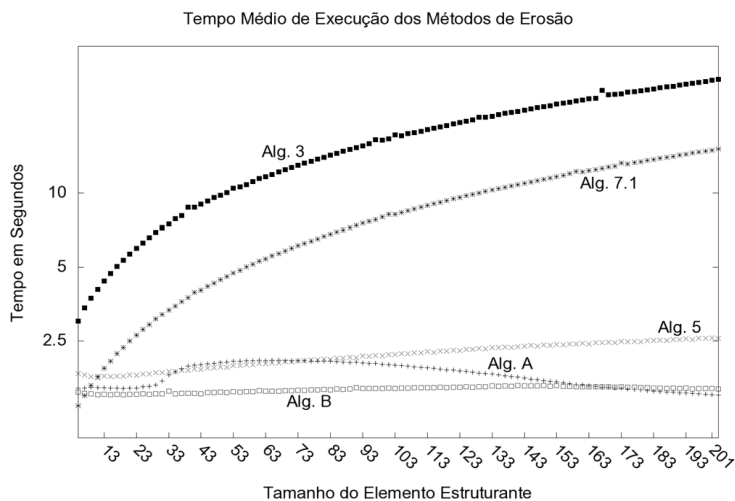


Figura 3. Tempo médio de execução dos algoritmos de erosão.

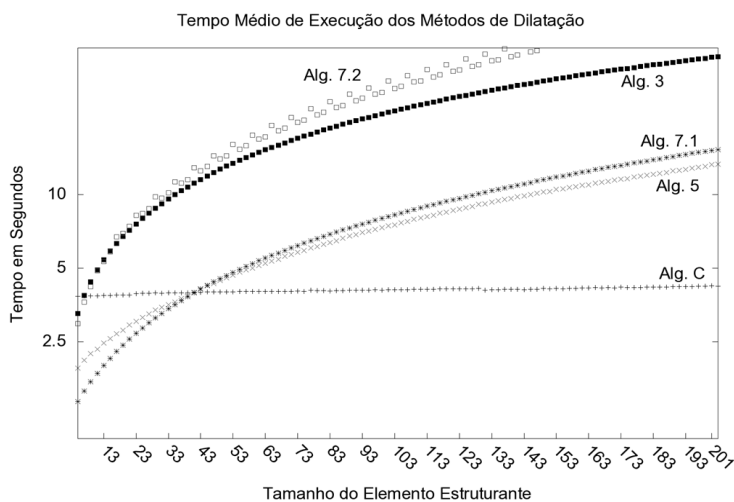


Figura 4. Tempo médio de execução dos algoritmos de dilatação.

De acordo com os resultados, os Algoritmos 3, 7.1 e 7.2 sofrem influências significativas com o aumento do tamanho do EE. No Algoritmo 5 para erosão observamos uma razoável *performance* mediante ao crescimento do elemento estruturante, ou seja, o tempo de execução deste algoritmo é pouco influenciado pelo aumento do tamanho do EE. Ambos os algoritmos de erosão propostos por este trabalho se destacam por sua eficiência. Tal eficiência é, em parte, justificada para ambos os algoritmos: “quanto maior

a densidade de uma casca do EE, espera-se que o algoritmo efetue maiores saltos em sua busca”. O algoritmo C obteve bons resultados devido a simplicidade da forma do EE (quadrado) e o conjunto de adjacência  $V$  escolhido. Podemos ainda utilizar um dos algoritmos de erosão rápida para resolver o problema da dilatação, visto que ambos são duais [Dougherty and Lotufo 2003].

## 6. Conclusão

Os resultados experimentais comprovam a análise teórica de que os algoritmos propostos por este trabalho são boas alternativas de erosão e dilatação eficientes. Cada um destes rápidos algoritmos deve ser utilizado em casos específicos: o algoritmo de *erosão direta* é aplicável em imagens ruidosas; o algoritmo de *erosão inversa* é aplicável em pontos de altas densidades; por fim o algoritmo de dilatação por cascas é apropriado na presença de EE simples (compacto quanto ao conjunto de  $V$ -cascas). Durante o mestrado, realizamos uma apresentação em forma de um resumo estendido<sup>3</sup> no *International Symposium on Mathematical Morphology*<sup>4</sup> (ISMM-2007) de resultados preliminares deste trabalho. Atualmente, estamos trabalhando para publicar estes resultados em periódicos internacionais da área e em estender estes resultados em outras aplicações.

## Referências

- Banon, G. J. F. and Barrera, J. (1994). *Bases da Morfologia Matemática para a Análise de Imagens Binárias*. IX Escola de Computação, Recife.
- Barrera, J. and Salas, G. P. (1996). Set Operations on Closed Intervals and Their Applications to the Automatic Programming of Morphological Machines. *Electronic Imaging*, 5(3):335–352.
- Borgefors, G. (1986). Distance Transformations on Digital Images. *Computer Vision and Image Processing*, 51(34):344–371.
- Brambor, J. (2002). Implementation Notes of Binary Dilation and Erosion on 64-bit SH-5 Processor.
- Chen, S. and Haralick, R. M. (1995). Recursive erosion, dilation, opening, and closing transforms. *IEEE Transactions on Computers*, 4(3):335–345.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press and McGraw-Hill.
- Dougherty, E. R. and Lotufo, R. A. (2003). *Hands-on Morphological Image Processing*. SPIE International Society for Optical Engine.
- Kim, W. J., Kim, S. D., and Kim, K. (2005). Fast Algorithms for Binary Dilation and Erosion Using Run-Length Encoding. *ETRI Journal*, 27(6):814–817.
- Serra, J. (1982). *Image Analysis and Mathematical Morphology*. Academic Press, New York.
- Xu, J. (1991). Decomposition of Convex Polygonal Morphological Structuring Elements into Neighborhood Subsets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):153–162.

<sup>3</sup><http://ismm.dpi.inpe.br/col/dpi.inpe.br/ismm@80/2007/06.16.00.48/doc/extendedabstract2.pdf>

<sup>4</sup>Uma das principais conferências na área de MM