

**Uma nova abordagem para as operações de
Transformada da Erosão e
Transformada da Dilatação
através do uso da união da
decomposição de elementos estruturantes**

Alexandre Yukio Harano

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Ronaldo Fumio Hashimoto

Durante o desenvolvimento deste trabalho, o autor recebeu
auxílio financeiro do CNPq e da FAPESP.

São Paulo, Fevereiro de 2013

**Uma nova abordagem para as operações de
Transformada da Erosão e
Transformada da Dilatação
através do uso da união da
decomposição de elementos estruturantes**

Esta dissertação trata-se da versão original
do aluno Alexandre Yukio Harano.

Este trabalho é dedicado aos meus pais,
Edson Luiz Massonori Harano e
Helena Miyuki Ijiri Harano;
ao meu irmão, Fábio Hideki Harano; e
à minha amada companheira, Cintia Lumi Tho,
por me definirem como a pessoa que sou.

Agradecimentos

Ao longo dos anos do período do programa de mestrado, diversas pessoas e instituições me apoiaram em vários aspectos e, portanto, utilizo esse espaço para agradecê-las. As listas apresentadas utilizam a ordem lexicográfica somente por conveniência.

Gostaria de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelas respectivas bolsas concedidas para o desenvolvimento em tempo integral desta dissertação de mestrado.

Sou imensamente grato aos meus pais, Edson Luiz Massonori Harano e Helena Miyuki Ijiri Harano, por me apoiarem em todos os aspectos de minha vida, incluindo a liberdade de cometer falhas. Muito obrigado por permitir que eu seja quem sou hoje, seja por minhas qualidades quanto por meus defeitos. Agradeço ao meu irmão, Fábio Hideki Harano, por aprender a avaliar melhor meus defeitos como ser humano e a reconhecer as diversas visões sobre o mundo no qual vivemos.

Agradeço à minha amada companheira Cintia Lumi Tho, a qual me suporta no dia a dia, dividindo comigo as alegrias e os pesares da vida, e a quem sou grato de todo o meu coração. Aos meus sogros, Yuiti Tho e Eliza Shigueko Nishiya Tho, sou grato por todo o apoio recebido ao longo desses anos de convívio.

Agradeço o apoio moral de meus amigos: Fábio Cassarotti Parronchi Navarro; Gabriel Geraldo França Marcondes; Gustavo Keniti Yasuda; Hugo Poloni Amâncio da Silva; Luiz Carlos Irber Júnior; Mário Clemeêncio da Silva Júnior; e Raphael Nunes da Motta.

Ao Núcleo de Informação de Coordenação do Ponto BR (NIC.br), agradeço pelo tempo cedido para a finalização deste trabalho. Sou grato aos colegas que acompanharam de perto o desenlace: Antonio Marcos Moreiras; César de Tassis Filho; Eduardo Barasal Morales; Edwin Santos Cordeiro; Fátima Aparecida M. de Siqueira; Heitor de Souza Ganzeli; Milton Kaoru Kashiwakura; Rodrigo Matos Carnier; Rodrigo Regis dos Santos; Tiago Jun Nakamura; e Tuany Oguro Tabosa.

Por me auxiliar nos aspectos psicológicos e psiquiátricos, agradeço respectivamente à Angeline Nascimento Lima Uenoyama e à Doutora Belquiz Schiffnagel Avrichir.

Por último, mas não menos importante, sou grato ao Instituto de Matemática e Estatística pela oportunidade de realização do curso de mestrado, e ao Professor Doutor Ronaldo Fumio Hashimoto, por sua atenção, orientação e imensurável paciência para com este aluno.

Resumo

Dentro do âmbito da Morfologia Matemática aplicada a imagens binárias, o tempo requerido para a determinação das coordenadas de interesse a serem analisadas nas operações de Transformada da Erosão e Transformada da Dilatação usualmente é um dos termos de maior custo ao calcular a complexidade computacional de tais algoritmos.

Apresentamos neste trabalho algoritmos simples que implementam tais operações, seguido de refinamentos incrementais, até a exposição do algoritmo proposto, cuja finalidade é efetuar as operações com o menor número de comparações para a determinação das coordenadas a serem verificadas. O algoritmo proposto se difere dos apresentados em trabalhos anteriores por duas características: (i) uma decomposição de elemento estruturante é requerida como entrada, de modo que a união dos elementos dessa lista é utilizada para efetuar a Erosão ou a Dilatação; e (ii) uma estrutura de dados matricial foi desenvolvida de modo a relacionar as coordenadas de interesse da caixa delimitadora da imagem de entrada às coordenadas da união citada, sendo que o uso de tal estrutura permite a redução do número de comparações efetuadas para a determinação das coordenadas a serem verificadas.

Para todos os algoritmos discutidos, são apresentadas análises linha a linha de complexidade computacional. O algoritmo proposto possui o menor tempo computacional teórico dentre os verificados. O procedimento experimental aplicado também é discutido e o resultado obtido indicou que o método proposto é recomendado para imagens naturais.

Palavras-chave: Morfologia Matemática, imagens binárias, Transformada da Erosão, Transformada da Dilatação, decomposição de elemento estruturante.

Abstract

Within the Mathematical Morphology framework applied to binary images, the required time to determine the coordinates of interest to be analyzed within the Erosion Transform and Dilation Transform operators usually is one of the most expensive terms when calculating the computational complexity of their algorithms.

This work presents simple algorithms which implement those operations, followed by incremental refinements, up to the exposure of the proposed algorithm, whose purpose is to perform operations using the lowest number of comparisons to determine the coordinates to be checked. The proposed algorithm differs from the ones presented in previous works on two aspects: (i) a structuring element decomposition is required as input, insomuch that the union of the elements from that list is used to calculate the Erosion or the Dilation operation; and (ii) a matrix data structure was developed to relate the coordinates of interest from the input image's bounding box with the coordinates of the aforementioned union, in such manner that its use reduces the number of comparisons made to determinate the coordinates to be checked.

Line by line analysis for the computational complexity of each discussed algorithm is presented. The proposed algorithm has the lowest theoretical computational time from the verified analysis. The applied experimental procedure is also discussed and the result indicated that the proposed method is recommended for natural images.

Keywords: Mathematical Morphology, binary images, Erosion Transform, Dilation Transform, structuring element decomposition.

Conteúdo

Lista de Algoritmos	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Algoritmo Proposto para Transformada da Erosão e Transformada da Dilatação	1
1.2 Motivação	3
1.3 Organização do Trabalho	4
2 Fundamentos Matemáticos	5
2.1 Definições Gerais	5
2.2 Operações em Conjuntos	6
2.3 Morfologia Matemática	6
2.4 Algoritmos	9
2.5 Complexidade Computacional	11
3 Algoritmos Prévios para Transformada da Erosão e Transformada da Dilatação	15
3.1 Algoritmos Simples	15
3.1.1 Algoritmo Simples de Transformada da Erosão	15
3.1.2 Algoritmo Simples de Transformada da Dilatação	16
3.2 Algoritmo Dual	18
3.3 Algoritmo Dual de Vetor e Lista	19
3.4 Algoritmo Dual de Vetor e Lista de Consumo de Borda	24
4 Algoritmo Proposto para Transformada da Dilatação e Transformada da Erosão	29
5 Corretismo do Algoritmo Proposto	37
5.1 Demonstrações e Definições Comuns aos Dois Operadores	37
5.2 Algoritmo de Transformada da Erosão	39
5.3 Algoritmo de Transformada da Dilatação	47
6 Experimento	51
6.1 Procedimento experimental	51
6.2 Resultados	54
6.3 Análise do Dados Experimentais	58
6.3.1 Análise do Tempo Computacional Teórico	58

6.3.2	Análise do Tempo Médio de Execução	59
7	Conclusões	61
7.1	Considerações Finais	61
7.2	Sugestões para Pesquisas Futuras	61
A	Tabelas de Complexidade Computacional dos Algoritmos Apresentados	63
A.1	Complexidade Computacional dos Algoritmos da Seção 3.1	63
A.2	Complexidade Computacional dos Algoritmos da Seção 3.2	64
A.3	Complexidade Computacional dos Algoritmos da Seção 3.3	65
A.4	Complexidade Computacional dos Algoritmos da Seção 3.4	68
A.5	Complexidade Computacional dos Algoritmos do Capítulo 4	71
B	Dados experimentais	75
B.1	Tempos de Execução Obtidos	75
B.1.1	Elementos estruturantes 3×3	75
B.1.2	Elementos estruturantes 5×5	83
B.2	Contadores de Comparações e Acessos à Memória	90
B.2.1	Elementos estruturantes 3×3	90
B.2.2	Elementos estruturantes 5×5	98
	Bibliografia	107
	Índice	108

Lista de Algoritmos

1	EXEMPLO NÚMERO PAR(n)	9
2	TRANSFORMADA EROÇÃO SIMPLES(X, \mathcal{L}, k)	16
3	TRANSFORMADA DILATAÇÃO SIMPLES(X, \mathcal{L}, k)	17
4	TRANSFORMADA DILATAÇÃO DUAL(X, \mathcal{L}, k)	18
5	TRANSFORMADA EROÇÃO DUAL(X, \mathcal{L}, k)	18
6	TRANSFORMADA DUAL(X, \mathcal{L}, k , é Transformada Erosão)	19
7	TRANSFORMADA DILATAÇÃO LINEAR(X, \mathcal{L}, k)	20
8	TRANSFORMADA EROÇÃO LINEAR(X, \mathcal{L}, k)	20
9	TRANSFORMADA DUAL LINEAR(X, \mathcal{L}, k , é Transformada Erosão)	20
10	LINEARIZE ELEMENTO ESTRUTURANTE(\mathcal{L}, k)	22
11	LINEARIZE IMAGEM LINEAR(X , é Transformada Erosão)	22
12	ALGORITMO DUAL LINEAR(X , lista Coordenadas, elementos União, cardinalidade L , elementos L , k , é Transformada Erosão)	23
13	TRANSFORMADA DILATAÇÃO BORDA(X, \mathcal{L}, k)	24
14	TRANSFORMADA EROÇÃO BORDA(X, \mathcal{L}, k)	24
15	TRANSFORMADA DUAL LINEAR DE BORDA(X, \mathcal{L}, k , é Transformada Erosão)	25
16	LINEARIZE BORDA IMAGEM(X , cardinalidade União, elementos União, é Transformada Erosão)	25
17	ATUALIZAÇÃO DUAL LINEAR DE BORDA(imagem Trabalho, lista Coordenadas, imagem Candidato, a Atualizar, cardinalidade União, elementos União, é Transformada Erosão, iteração, resultado)	26
18	ALGORITMO DUAL LINEAR DE BORDA(X , lista Coordenadas, imagem Candidato, cardinalidade União, elementos União, cardinalidade L , elementos L , k , é Transformada Erosão)	27
19	TRANSFORMADA DILATAÇÃO PROPOSTA(X, \mathcal{L}, k)	30
20	TRANSFORMADA EROÇÃO PROPOSTA(X, \mathcal{L}, k)	30
21	TRANSFORMADA PROPOSTA(X, \mathcal{L}, k , é Transformada Erosão)	30
22	INICIALIZA MATRIZ ESPARSA(X , cardinalidade União, elementos União, é Transformada Erosão)	33
23	ATUALIZAÇÃO ALGORITMO PROPOSTO(imagem Trabalho, lista Nós Borda, matriz Nós Borda, imagem Candidato, vetor Listas União, a Atualizar, cardinalidade União, elementos União, é Transformada Erosão, iteração, resultado)	34
24	ALGORITMO PROPOSTO(X , lista Nós Borda, matriz Nós Borda, imagem Candidato, vetor Listas União, cardinalidade União, elementos União, cardinalidade L , elementos L , k , é Transformada Erosão)	35

Lista de Figuras

1.1	Exemplo de decomposição de elemento estruturante E em \mathbb{Z}^2	2
1.2	Exemplo de Transformada da Erosão (com a decomposição da Figura 1.1 como entrada).	3
2.1	Fluxograma para as implementações usuais dos operadores.	9
2.2	Fluxograma para as implementações apresentadas dos operadores.	10
3.1	Exemplo visual do Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE).	21
4.1	Apresentação dos tipos de nós utilizados: nó da borda e nó de ligação.	31
4.2	Exemplo visual da matriz esparsa bidimensional inicialmente calculada através de INICIALIZEMATRIZESPARGA para a operação de Transformada da Erosão (como continuação da Figura 3.1).	32
6.1	Imagem de teste oriunda de (Machado , 2008).	52
6.2	Divisão da Figura 6.1 de acordo com seus quadrantes.	52
6.3	Elementos estruturantes 3×3 utilizados no procedimento experimental.	53
6.4	Elementos estruturantes 5×5 utilizados no procedimento experimental.	54
6.5	Gráficos do tempo médio para a Figura 6.1 utilizando elementos estruturantes 3×3	55
6.6	Gráficos do tempo médio para a Figura 6.1 utilizando elementos estruturantes 5×5	55
6.7	Gráficos do tempo médio para a Figura 6.2a utilizando elementos estruturantes 3×3	55
6.8	Gráficos do tempo médio para a Figura 6.2a utilizando elementos estruturantes 5×5	56
6.9	Gráficos do tempo médio para a Figura 6.2b utilizando elementos estruturantes 3×3	56
6.10	Gráficos do tempo médio para a Figura 6.2b utilizando elementos estruturantes 5×5	56
6.11	Gráficos do tempo médio para a Figura 6.2c utilizando elementos estruturantes 3×3	57
6.12	Gráficos do tempo médio para a Figura 6.2c utilizando elementos estruturantes 5×5	57
6.13	Gráficos do tempo médio para a Figura 6.2d utilizando elementos estruturantes 3×3	57
6.14	Gráficos do tempo médio para a Figura 6.2d utilizando elementos estruturantes 5×5	58

Lista de Tabelas

2.1	Definição de <i>borda</i> de acordo com a operação desejada.	8
2.2	Definição de <i>borda</i> de acordo com a operação dual desejada.	8
2.3	Análise de Complexidade de EXEMPLO NÚMERO PAR.	11
2.4	Definição de I_x aplicada na Seção 3.1.	12
2.5	Definição de I_x aplicada aos algoritmos duais.	12
5.1	Definição de \mathcal{F}_{x+1}^u de acordo com a operação desejada.	38
6.1	Lista de parâmetros analisados no procedimento experimental.	51
6.2	Análise de Complexidade das implementações utilizadas.	58
A.1	Análise de Complexidade de TRANSFORMADA EROSÃO SIMPLES.	63
A.2	Análise de Complexidade de TRANSFORMADA DILATAÇÃO SIMPLES.	64
A.3	Análise de Complexidade de TRANSFORMADA DUAL.	65
A.4	Análise de Complexidade de LINEARIZE ELEMENTO ESTRUTURANTE.	66
A.5	Análise de Complexidade de LINEARIZE IMAGEM LINEAR.	66
A.6	Análise de Complexidade de ALGORITMO DUAL LINEAR.	67
A.7	Análise de Complexidade de TRANSFORMADA DUAL LINEAR.	67
A.8	Análise de Complexidade de LINEARIZE BORDA IMAGEM.	68
A.9	Análise de Complexidade de ATUALIZAÇÃO DUAL LINEAR DE BORDA.	69
A.10	Análise de Complexidade de ALGORITMO DUAL LINEAR DE BORDA.	70
A.11	Análise de Complexidade de TRANSFORMADA DUAL LINEAR DE BORDA.	70
A.12	Análise de Complexidade de INICIALIZAÇÃO MATRIZ ESPARSA.	71
A.13	Análise de Complexidade de ATUALIZAÇÃO ALGORITMO PROPOSTO.	72
A.14	Análise de Complexidade de ALGORITMO PROPOSTO.	73
A.15	Análise de Complexidade de TRANSFORMADA PROPOSTA.	73
B.1	Tempo de execução do Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 3×3	75
B.2	Tempo de execução do Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 3×3	76
B.3	Tempo de execução do Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 3×3	76
B.4	Tempo de execução do Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 3×3	77
B.5	Tempo de execução do Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 3×3	77
B.6	Tempo de execução do Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 3×3	78
B.7	Tempo de execução do Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 3×3	78

B.8	Tempo de execução do Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 3×3 .	79
B.9	Tempo de execução do Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 3×3 .	79
B.10	Tempo de execução do Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 3×3 .	80
B.11	Tempo de execução do Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 3×3 .	80
B.12	Tempo de execução do Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 3×3 .	81
B.13	Tempo de execução do Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 3×3 .	81
B.14	Tempo de execução do Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 3×3 .	82
B.15	Tempo de execução do Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 3×3 .	82
B.16	Tempo de execução do Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 5×5 .	83
B.17	Tempo de execução do Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 5×5 .	83
B.18	Tempo de execução do Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 5×5 .	84
B.19	Tempo de execução do Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 5×5 .	84
B.20	Tempo de execução do Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 5×5 .	85
B.21	Tempo de execução do Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 5×5 .	85
B.22	Tempo de execução do Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 5×5 .	86
B.23	Tempo de execução do Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 5×5 .	86
B.24	Tempo de execução do Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 5×5 .	87
B.25	Tempo de execução do Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 5×5 .	87
B.26	Tempo de execução do Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 5×5 .	88
B.27	Tempo de execução do Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 5×5 .	88
B.28	Tempo de execução do Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 5×5 .	89
B.29	Tempo de execução do Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 5×5 .	89
B.30	Tempo de execução do Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 5×5 .	90
B.31	Contadores para o Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 3×3 .	90
B.32	Contadores para o Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 3×3 .	91
B.33	Contadores para o Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 3×3 .	91
B.34	Contadores para o Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 3×3 .	92
B.35	Contadores para o Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 3×3 .	92
B.36	Contadores para o Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 3×3 .	93

B.37	Contadores para o Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 3×3 .	93
B.38	Contadores para o Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 3×3 .	94
B.39	Contadores para o Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 3×3 .	94
B.40	Contadores para o Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 3×3 .	95
B.41	Contadores para o Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 3×3 .	95
B.42	Contadores para o Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 3×3 .	96
B.43	Contadores para o Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 3×3 .	96
B.44	Contadores para o Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 3×3 .	97
B.45	Contadores para o Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 3×3 .	97
B.46	Contadores para o Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 5×5 .	98
B.47	Contadores para o Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 5×5 .	98
B.48	Contadores para o Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 5×5 .	99
B.49	Contadores para o Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 5×5 .	99
B.50	Contadores para o Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 5×5 .	100
B.51	Contadores para o Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 5×5 .	100
B.52	Contadores para o Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 5×5 .	101
B.53	Contadores para o Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 5×5 .	101
B.54	Contadores para o Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 5×5 .	102
B.55	Contadores para o Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 5×5 .	102
B.56	Contadores para o Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 5×5 .	103
B.57	Contadores para o Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 5×5 .	103
B.58	Contadores para o Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 5×5 .	104
B.59	Contadores para o Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 5×5 .	104
B.60	Contadores para o Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 5×5 .	105

Capítulo 1

Introdução

O trabalho aqui representado propõe um novo algoritmo para a Transformada da Erosão e Transformada da Dilatação, operações que utilizam as respectivas operações de Erosão e Dilatação, as quais compõem a base do arcabouço da Morfologia Matemática.

1.1 Algoritmo Proposto para Transformada da Erosão e Transformada da Dilatação

A Transformada da Erosão e a Transformada da Dilatação são operações da Morfologia Matemática que foram utilizadas em (Chen e Haralick , 1995), onde as mesmas foram definidas para imagens binárias bidimensionais (subconjuntos de \mathbb{Z}^2). Os operadores descritos neste texto utilizam imagens que são subconjuntos de \mathbb{Z}^d , onde $d > 1$. As operações utilizadas ao longo desse texto e toda a base necessária para seu entendimento estão definidas no Capítulo 2.

Visando apresentar de forma simplificada os algoritmos desenvolvidos nesse trabalho, inicialmente definimos os parâmetros de entrada para as implementações usuais dos operadores de Transformada da Erosão e Transformada da Dilatação como sendo:

- número de dimensões $d > 1$ para a definição das imagens;
- imagem binária de entrada $X \subset \mathbb{Z}^d$, e
- elemento estruturante $E \subset \mathbb{Z}^d$.

Estabelecemos como restrição para o algoritmo proposto, o resultado de uma decomposição prévia do elemento estruturante E e tal resultado é utilizado como parâmetro de entrada do algoritmo. A decomposição de elementos estruturantes é um tema bastante estudado, como podemos verificar em (Zhuang e Haralick , 1986), (Xu , 1991), (Hashimoto *et al.* , 2000), (Hashimoto e Barrera , 2001), (Hashimoto e Barrera , 2003), (Shih e Wu , 2005) e (Yang e Lee , 2005).

Para prosseguirmos com a apresentação do conteúdo, devemos definir a *adição de Minkowski*. Essa operação é calculada utilizando dois conjuntos G e H , sendo representada pela notação $G \oplus H$, e pode ser definida como

$$G \oplus H = \{j \in \mathbb{Z}^d : (j + H^r) \cap G \neq \emptyset\} . \quad (1.1)$$

A definição da restrição consiste no fato de que o elemento estruturante E deve ser decomponível em $k + 1$ elementos estruturantes listados em $\mathcal{L} = [L_0, L_1, \dots, L_k]$. A relação entre E e os elementos de \mathcal{L} é a seguinte:

$$E = L_0 \oplus L_1 \oplus \dots \oplus L_k . \quad (1.2)$$

Além da decomposição do elemento estruturante E , todos os elementos de \mathcal{L} possuem sua própria origem, ou seja,

$$\{o\} \subseteq L_i \forall L_i \in \mathcal{L} . \quad (1.3)$$

Para que tal decomposição seja efetuada, a origem o deve estar contida em E , i.e., $o \in E$. O elemento estruturante $L_0 = \{o\}$ é o elemento neutro das operações de Erosão e de Dilatação e é representado para fins do corretismo do algoritmo. A Figura 1.1 exemplifica uma decomposição para um elemento estruturante E em \mathbb{Z}^2 .

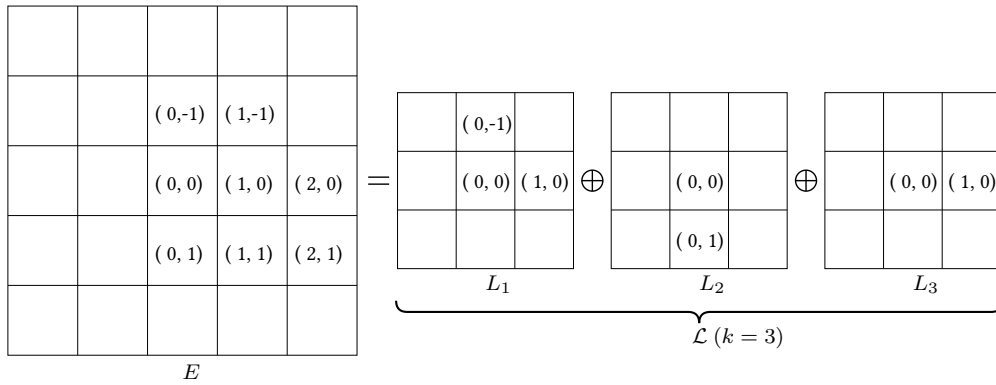


Figura 1.1: Exemplo de decomposição de elemento estruturante E em \mathbb{Z}^2 .

Com o intuito de apresentarmos de forma comparável os diversos algoritmos para as operações de Transformada da Erosão e Transformada da Dilatação descritas ao longo deste texto, adaptamos os parâmetros de entrada das implementações usuais para utilizarmos os seguintes parâmetros de entrada:

- número de dimensões $d > 1$ para a definição das imagens;
- imagem binária de entrada $X \subset \mathbb{Z}^d$; e
- uma lista $\mathcal{L} = [L_1, \dots, L_k]$ de k elementos estruturantes, de modo que a origem o pertence a cada elemento de \mathcal{L} .

A Figura 1.2 exemplifica o resultado esperado da Transformada da Erosão para o inteiro $d = 2$; a imagem de entrada X e a lista \mathcal{L} de $k = 3$ elementos estruturantes resultantes da decomposição representada pela Figura 1.1.

Assim como houve o desenvolvimento de algoritmos eficientes para as operações correlatas de Erosão e Dilatação aplicadas a imagens binárias apresentados em (Hirata , 1997), (Zampirolli , 2003) e (Machado , 2008), o trabalho aqui descrito propõe uma nova implementação para as operações de Transformada da Erosão e Transformada da Dilatação, sendo a utilização da união dos elementos de \mathcal{L} uma das principais características.

O algoritmo proposto visa minimizar o número de comparações necessárias para a determinação da borda — etapa a qual usualmente está atrelada o maior tempo computacional das operações — através do uso de dois tipos de nós, sendo um nó uma unidade de dado que compõe a matriz esparsa bidimensional.

Para detalharmos o desenvolvimento do algoritmo proposto, partimos da implementação mais trivial conhecida e aplicamos uma série de refinamentos incrementais para chegarmos ao algoritmo em questão. Com base na classificação realizada no Capítulo 3 de (Zampirolli , 2003) para os algoritmos de Erosão,

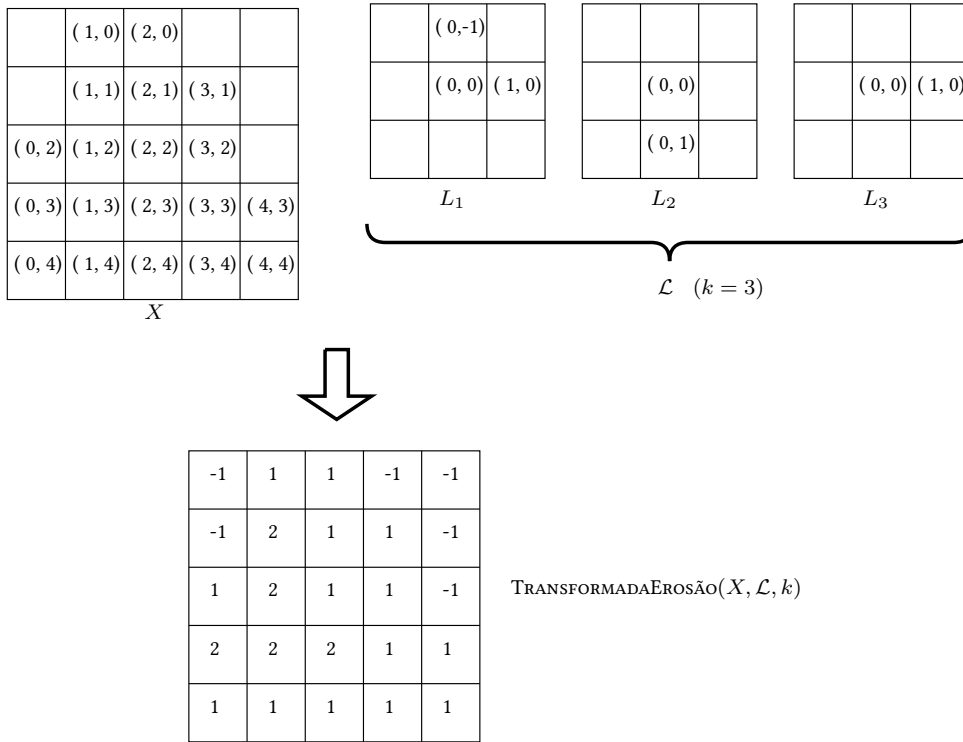


Figura 1.2: Exemplo de Transformada da Erosão (com a decomposição da Figura 1.1 como entrada).

podemos estender tal conceito aos algoritmos apresentados neste texto de modo que todos são algoritmos sequenciais.

1.2 Motivação

Os algoritmos de Transformada da Erosão e Transformada da Dilatação possuem como gargalo o tempo gasto para determinar quais são as coordenadas de interesse a serem analisadas e o tempo gasto para determinar quais serão os elementos a serem, respectivamente, removidos da ou inseridos na imagem fornecida como parâmetro de entrada em uma iteração dos algoritmos.

A motivação deste trabalho foi o desenvolvimento de um algoritmo que possui como premissa básica a redução do número de comparações para determinar quais serão as coordenadas analisadas. Tal motivação em conjunto com o fato de que a decomposição de elementos estruturantes foi um tema bastante estudado nos impulsionou a elaborar um algoritmo que utiliza a união dos elementos estruturantes resultantes da decomposição para alcançar o objetivo proposto.

Para que tal premissa fosse cumprida, optamos por desenvolver uma matriz bidimensional esparsa que armazena os dados relativos à borda da imagem fornecida como parâmetro de entrada, relativa à união dos elementos estruturantes resultantes da decomposição, sendo tal borda composta por coordenadas de interesse em \mathbb{Z}^d . Essa matriz possui como informações: (i) as coordenadas de interesse que se encontram na caixa delimitadora de X , (ii) as coordenadas dos elementos da união dos elementos estruturantes de \mathcal{L} , e (iii) a relação entre os dois itens anteriores.

Um fator importante na escolha das operações de Transformada da Erosão e Transformada da Dilatação ao invés de suas contrapartes, Erosão e Dilatação, se deve ao fato de que os custos computacionais de inicialização dos algoritmos apresentados possuem valores consideráveis, de modo a ser mais interessante a apresentação dos algoritmos discutidos como implementações das Transformadas.

1.3 Organização do Trabalho

Segue-se uma lista de como o trabalho foi apresentado ao longo do texto. No Capítulo 2, a base conceitual necessária para o entendimento dessa dissertação é introduzida. O Capítulo 3 apresenta os algoritmos de Transformada da Dilatação e Transformada da Erosão mais triviais e, a cada seção, são propostos refinamentos incrementais, os quais são ilustrados através de um algoritmo associado. O algoritmo proposto é apresentado no Capítulo 4, seguido da prova de seu corretismo no Capítulo 5. O procedimento experimental aplicado e seus resultados são apresentados no Capítulo 6. Finalmente, no Capítulo 7, discutimos as conclusões obtidas neste trabalho.

Efetuamos o cálculo de complexidade computacional dos algoritmos discutidos nos Capítulos 3 e 4 de modo a expor as tabelas relacionadas no Apêndice A e apresentamos dados obtidos através do procedimento experimental no Apêndice B.

Capítulo 2

Fundamentos Matemáticos

Neste capítulo, são apresentados conceitos utilizados ao longo do texto, agrupados de acordo com o contexto de aplicação.

2.1 Definições Gerais

O conjunto \mathbb{Z} é composto pelos números inteiros. A seguir, a notação utilizada ao longo do texto:

- letras minúsculas — como d , v e w — podem representar valores escalares inteiros ou coordenadas, conforme o contexto em que são utilizadas;
- letras maiúsculas — como X , E e L — representam imagens binárias de d dimensões;
- a letra em formatação especial \mathcal{L} representa uma lista;
- palavras com a formatação como em **verdadeiro** e **falso** são palavras reservadas;
- sequências de palavras com a formatação como em `TRANSFORMADADUAL`, `TRANSFORMADADUALLINEAR` e `TRANSFORMADAPROPOSTA` representam algoritmos; e
- sequências de palavras com a formatação como em *éTransformadaErosão*, *índiceElementoEstruturante* e *vetorListasUnião* são parâmetros ou variáveis utilizados nos algoritmos e são descritos quando utilizados.

Segue-se uma lista das estruturas de dados básicas utilizadas neste texto.

Vetor Contém elementos de um único tipo e é uma estrutura de dados linear que possui o tamanho fixado em sua instanciação. A estrutura de vetor permite acesso direto aos elementos contidos através da utilização de índices, sendo que o primeiro elemento de um vetor está atrelado ao índice 1, o segundo ao índice 2 e assim sucessivamente.

Fila Contém elementos de um único tipo e é uma estrutura de dados linear do tipo FIFO (*first in, first out*), de modo que o primeiro elemento a ser inserido será o primeiro a ser removido. Ao contrário do vetor, uma fila possui tamanho definido de acordo com as operações de inserção e de remoção, sendo que uma fila é instanciada sem elementos, i.e., vazia.

Lista duplamente encadeada com cabeça Contém elementos de um único tipo e é uma estrutura de dados linear que permite a inserção em qualquer posição, através do uso de ponteiros que indicam o elemento anterior e o próximo elemento. Assim como uma fila, uma lista duplamente encadeada com cabeça possui tamanho definido de acordo com as operações de inserção e de remoção, sendo que tal lista é instanciada sem elementos, i.e., vazia.

Imagem Quando utilizada como estrutura de dados, permite o acesso direto às informações atreladas a seus elementos através da utilização de coordenadas, as quais serão definidas na próxima seção.

Matriz É uma estrutura multidimensional a qual permite o acesso direto às informações nela contidas através da utilização de uma tupla de números inteiros não-negativos, cujo número de dimensões seja igual ao da matriz.

2.2 Operações em Conjuntos

Uma *imagem binária* é um subconjunto de \mathbb{Z}^d e um *elemento estruturante* é um tipo particular de imagem binária. Uma *coordenada* é uma tupla de d valores inteiros, a qual é utilizada para representar uma posição em \mathbb{Z}^d . A *origem* é uma coordenada de modo que toda a sua tupla é preenchida pelo valor zero (0).

Sejam c e d duas coordenadas. A notação $-c$ representa uma coordenada de modo que o valor de cada posição de sua tupla é calculada como o simétrico do valor contido na mesma posição em c em relação a 0. A expressão $c + d$ denota uma coordenada de modo que o valor de cada posição de sua tupla é calculada como a soma dos valores contidos na mesma posição em c e em d , enquanto que a expressão $c - d$ equivale a $c + (-d)$. A comparação $c \leq d$ é válida se e somente se o valor de cada posição da tupla de c for menor ou igual ao valor da mesma posição da tupla de d .

Seja $c \in X$ a coordenada de um elemento da imagem X . Uma *caixa delimitadora* (*bounding box*) de X pode ser representada por duas coordenadas a e z , de modo que, para toda coordenada c , valha a relação $a \leq c \leq z$.

Seja p uma coordenada e E uma imagem binária. A expressão $p + E$ denota uma imagem binária de modo que a coordenada de cada elemento é calculada como $p + e$, para todos os elementos de coordenadas $e \in E$. Essa operação é conhecida como a *translação* de E por p e pode ser reescrita como

$$p + E = \{h \in \mathbb{Z}^d : \forall e \in E, h = p + e\} . \quad (2.1)$$

Seja E uma imagem binária. E^c denota a imagem binária composta pelo complemento de E em relação ao universo \mathbb{Z}^d e é conhecido como o *complemento* de E e pode ser reescrita como

$$E^c = \{h \in \mathbb{Z}^d : h \notin E\} . \quad (2.2)$$

As notações $-E$ e E^r são equivalentes e representam uma imagem binária de modo que a coordenada de cada um de seus elementos é calculada como $-p$, para todos os elementos de coordenadas $p \in E$ e é conhecida como a *reflexão* de E , podendo ser reescrita como

$$-E = E^r = \{h \in \mathbb{Z}^d : -h \in E\} . \quad (2.3)$$

2.3 Morfologia Matemática

Sejam W um elemento estruturante e G uma imagem binária. A *Erosão* de G por W é representada por $\varepsilon_W(G)$ e a *Dilatação* de G por W é representada por $\delta_W(G)$. A operação de Erosão de G por W pode ser definida como

$$\varepsilon_W(G) = \{h \in \mathbb{Z}^d : (h + W) \subseteq G\} , \quad (2.4)$$

enquanto a operação de Dilatação de G por W pode ser definida como uma adição de Minkowski

$$\delta_W(G) = G \oplus W . \quad (2.5)$$

A dualidade entre as operações de Erosão e Dilatação permite que as mesmas sejam definidas para uma imagem binária G e um elemento estruturante W das seguintes formas:

$$\delta_W(G) = (\varepsilon_{W^r}(G^c))^c \quad (2.6)$$

e

$$\varepsilon_W(G) = (\delta_{W^r}(G^c))^c . \quad (2.7)$$

O *gradiente morfológico interno* de uma imagem binária G utilizando um elemento estruturante W é calculado através da expressão

$$\text{GradienteMorfológicoInterno}_W(G) = G \setminus \varepsilon_W(G) , \quad (2.8)$$

enquanto que o *gradiente morfológico externo* de G utilizando W é calculado através da expressão

$$\text{GradienteMorfológicoExterno}_W(G) = \delta_G(W) \setminus W . \quad (2.9)$$

Seja W um elemento estruturante. Seja $\{o\}$ um elemento estruturante composto somente pela origem. Seja Z um elemento estruturante W de raio r , sendo r um inteiro não-negativo. Define-se *raio* conforme a seguinte relação:

$$Z = rW = \{o\} \oplus \underbrace{W \oplus \dots \oplus W}_{r \text{ termos } W} . \quad (2.10)$$

Seja X uma imagem binária e E um elemento estruturante. Com base nas definições apresentadas em (Chen e Haralick , 1995), definimos os operadores de *Transformada da Erosão* de X por E e *Transformada da Dilatação* de X por E .

A Transformada da Erosão calcula uma imagem em níveis de cinza, cujos valores válidos para uma coordenada $c \in \mathbb{Z}^d$ são descritos como:

$$\text{TRANSFORMADAEROSÃO}(X, E)[c] = \begin{cases} \max\{w \mid c \in \varepsilon_{wE}(X)\} & : c \in X \\ -1 & : c \notin X \end{cases} \quad (2.11)$$

De modo análogo, a Transformada da Dilatação calcula uma imagem em níveis de cinza, cujos valores válidos para uma coordenada $c \in \mathbb{Z}^d$ são descritos como:

$$\text{TRANSFORMADADILATAÇÃO}(X, E)[c] = \begin{cases} \min\{w \mid c \in \delta_{wE}(X)\} & : \text{se } \exists w, c \in \delta_{wE}(X) \\ -1 & : \forall w, c \notin \delta_{wE}(X) \end{cases} \quad (2.12)$$

A *decomposição de elementos estruturantes* é um processo em que um determinado elemento estruturante E é decomposto numa lista $\mathcal{L} = [L_0, L_1, \dots, L_k]$ de $k+1$ elementos estruturantes, de modo a respeitar

a seguinte relação:

$$\begin{aligned} E &= L_0 \oplus L_1 \oplus \dots \oplus L_k \\ &= \delta_{L_k}(\dots \delta_{L_2}(\delta_{L_1}(L_0)) \dots) . \end{aligned} \quad (2.13)$$

O elemento estruturante $L_0 = \{o\}$ foi definido pelo fato dele ser o elemento neutro da Dilatação.

Adaptamos as operações de Transformada da Erosão e Transformada da Dilatação de modo a adotar uma decomposição do elemento estruturante E através da lista \mathcal{L} de k elementos estruturantes, conforme apresentado nas seguintes definições.

Substituindo a Equação 2.13 na Equação 2.11, obtemos a definição da operação de Transformada da Erosão utilizada ao longo do texto. A Transformada da Erosão calcula uma imagem em níveis de cinza, cujos valores válidos para uma coordenada $c \in \mathbb{Z}^d$ são descritos como:

$$\text{TRANSFORMADAEROSÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \max\{w \mid c \in \varepsilon_{wL_k}(\dots (\varepsilon_{wL_2}(\varepsilon_{wL_1}(X))) \dots)\} & : c \in X \\ -1 & : c \notin X \end{cases} \quad (2.14)$$

De modo análogo, substituindo a Equação 2.13 na Equação 2.12, obtemos a definição da operação de Transformada da Dilatação utilizada ao longo do texto. A Transformada da Dilatação calcula uma imagem em níveis de cinza, cujos valores válidos para uma coordenada $c \in \mathbb{Z}^d$ são descritos como:

$$\text{TRANSFORMADADILATAÇÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \min\{w \mid c \in \delta_{wL_k}(\dots (\delta_{wL_2}(\delta_{wL_1}(X))) \dots)\} \\ \quad : \mathbf{se} \exists w, c \in \delta_{wL_k}(\dots (\delta_{wL_2}(\delta_{wL_1}(X))) \dots) \\ -1 : \forall w, c \notin \delta_{wL_k}(\dots (\delta_{wL_2}(\delta_{wL_1}(X))) \dots) \end{cases} \quad (2.15)$$

Seja H um elemento estruturante e I uma imagem binária. Ao longo desse texto, utilizaremos o conceito de *borda*. Na Seção 3.1, onde não é aplicada a relação de dualidade adotada no restante do texto, valem as relações apresentadas na Tabela 2.1.

Tabela 2.1: Definição de borda de acordo com a operação desejada.

Operação	Definição de Borda
Transformada da Erosão	$borda_H(I) = \text{GradienteMorfológicoInterno}_H(I)$
Transformada da Dilatação	$borda_H(I) = \text{GradienteMorfológicoExterno}_H(I)$

Para os algoritmos duais do texto, valem as relações apresentadas na Tabela 2.2.

Tabela 2.2: Definição de borda de acordo com a operação dual desejada.

Operação	Definição de Borda
Transformada da Erosão	$borda_H(I) = \text{GradienteMorfológicoInterno}_H(I)$
Transformada da Dilatação	$borda_H(I) = \text{GradienteMorfológicoInterno}_{H^r}(I)$

2.4 Algoritmos

Inicialmente exemplificamos, através do Algoritmo 1 (EXEMPLONÚMEROPAR), os símbolos e a formatação utilizados para a representação de um procedimento.

Algoritmo 1 EXEMPLONÚMEROPAR(n)

- 1 $resposta \leftarrow \mathbf{falso}$ ▷ Comentário: variável $resposta$ recebe **falso**.
 - 2 se $n \bmod 2 = 0$ então ▷ Comentário: exemplo de indentação na linha seguinte.
 - 3 $resposta \leftarrow \mathbf{verdadeiro}$ ▷ Comentário: parâmetro n é par por ser divisível por 2.
 ▷ Comentário: o procedimento EXEMPLONÚMEROPAR verifica se n é par.
 - 4 **devolva** $resposta$
-

O fluxograma apresentado na Figura 2.1 sintetiza as implementações usuais das operações de Transformada da Erosão e Transformada da Dilatação entregando os resultados representados pelas Equações 2.11 e 2.12, respectivamente, de modo que os números apresentados no fluxograma representam os seguintes passos:

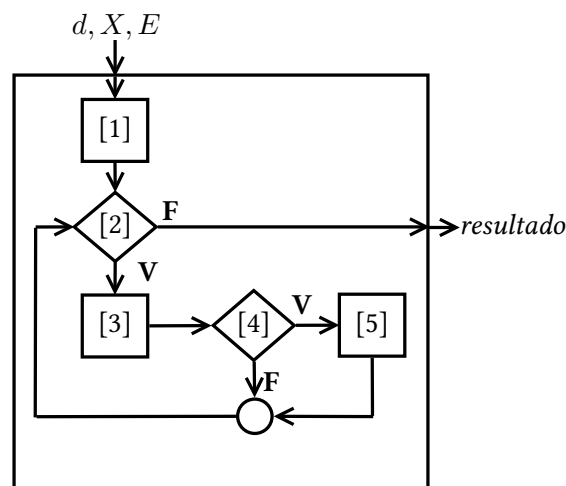


Figura 2.1: Fluxograma para as implementações usuais dos operadores.

1. A inicialização dos algoritmos, incluindo uma cópia de X para registro das alterações efetuadas a cada iteração como uma imagem de trabalho, uma variável para o controle de alterações e a instânciação de uma imagem em níveis de cinza cuja caixa delimitadora seja a mesma de X com o valor -1 atrelado a cada uma de suas coordenadas.
2. Houve alguma alteração na imagem de trabalho na iteração anterior ou esta é a primeira iteração?
3. Inicialização da iteração atual; cálculo da borda, de acordo com E , da imagem de trabalho fornecida como entrada para este passo; alteração da imagem de trabalho de acordo com a borda e o registro em $resultado$.
4. A borda calculada está vazia?
5. Contabilize que não houve mudança na imagem de trabalho.

Como houve a adoção da decomposição do elemento estruturante E em uma lista \mathcal{L} de k elementos estruturantes, com base no fluxograma apresentado na Figura 2.1, um novo fluxograma para representar

os algoritmos apresentados nos Capítulos 3 e 4 é apresentado na Figura 2.2, visando entregar os resultados representados pelas Equações 2.14 e 2.15, e os números apresentados no fluxograma representam os seguintes passos:

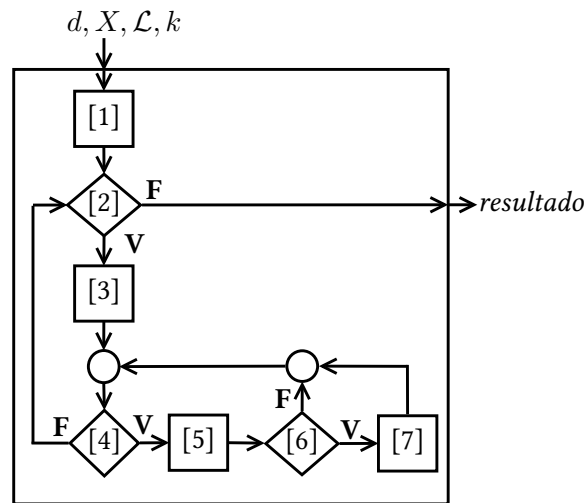


Figura 2.2: Fluxograma para as implementações apresentadas dos operadores.

1. A inicialização dos algoritmos, incluindo uma cópia de X para registro das alterações efetuadas a cada iteração como uma imagem de trabalho e uma variável para o controle de alterações.
2. Houve alguma mudança na imagem de trabalho na iteração anterior ou esta é a primeira iteração?
3. Inicialização da iteração atual.
4. Ainda há algum elemento estruturante de \mathcal{L} a ser verificado nessa iteração?
5. Calcule a borda, de acordo com o elemento estruturante de \mathcal{L} atualmente verificado, da imagem de trabalho fornecida como entrada para este passo; altere a imagem de trabalho de acordo com a borda e registre em *resultado*.
6. A borda calculada está vazia?
7. Contabilize que não houve mudança na imagem de trabalho para este elemento estruturante.

Uma vez que todos os algoritmos apresentados neste texto seguem o fluxograma representado pela Figura 2.2, a diferença consiste nas premissas adotadas durante seus respectivos desenvolvimentos, as quais afetam a complexidade computacional de cada implementação apresentada.

Uma *iteração* dos algoritmos discutidos é visualmente representada pelos Passos 2 a 5 do fluxograma da Figura 2.1 e pelos Passos 2 a 7 do fluxograma da Figura 2.2. Uma iteração é composta pelos passos referentes ao cálculo da borda, de acordo com o elemento estruturante E , da imagem de trabalho utilizada como imagem de entrada; e dos passos nos quais a imagem de entrada e a imagem resultante são alteradas de acordo com a borda calculada.

O Passo 2 dos fluxogramas representados pelas Figuras 2.1 e 2.2 representa o conceito de *condição de parada*. Independente da operação desejada, a condição é a mesma: se a imagem de trabalho não foi alterada após uma iteração, então a condição de parada foi alcançada e a imagem em níveis de cinza que armazena o resultado até a dada iteração pode ser devolvida como o resultado da operação.

A diferença entre os dois fluxogramas é que na Figura 2.1, a imagem de entrada da iteração é verificada somente por um único elemento estruturante E , enquanto que na Figura 2.2, são efetuadas verificações para os k elementos estruturantes de \mathcal{L} , de modo que é necessário contar o número de verificações em que a imagem de trabalho não foi alterada.

Um quesito que deve ser considerado antes de utilizar os algoritmos descritos neste texto é o fato de que os mesmos foram pensados já com a implementação em mente, pelo fato da definição de qual a caixa delimitadora ideal para o cálculo da Transformada da Dilatação. Os algoritmos apresentados utilizam para a Transformada somente as coordenadas que se encontrem na caixa delimitadora de X , de modo que as coordenadas p que se encontrem além da caixa são tratadas como não sendo elementos da imagem de trabalho, no caso da Transformada da Erosão, i.e., $p \notin \text{imagemTrabalho}$; e as coordenadas q que se encontrem além da caixa são tratadas como sendo elementos da imagem de trabalho, no caso da Transformada da Dilatação, i.e., $q \in \text{imagemTrabalho}$. Tal restrição utilizada pode impactar no resultado calculado, mas assim o fizemos para não utilizarmos um valor arbitrário no cálculo da nova caixa delimitadora de X .

Ao longo do texto, serão apresentados algoritmos que utilizam o conceito de dualidade entre as operações de Erosão e Dilatação, através da Equação 2.6 e para simplificar o entendimento desses algoritmos, definimos uma *coordenada de interesse* como uma coordenada da caixa delimitadora da imagem de trabalho de acordo com o tipo da operação desejada: no caso da Transformada da Erosão, as coordenadas de interesse c são as coordenadas dos elementos da imagem de trabalho, i.e., $c \in \text{imagemTrabalho}$, enquanto que no caso da Transformada da Dilatação, as coordenadas de interesse c pertencem ao complemento da imagem de trabalho em relação ao universo da caixa delimitadora da imagem de trabalho, i.e., $c \in \text{imagemTrabalho}^c$.

2.5 Complexidade Computacional

Para verificar o tempo e espaço que um determinado algoritmo consome, com base nos parâmetros de entrada, foram utilizadas as notações usuais da área de análise de algoritmos e notações próprias do texto. Para ilustrar como serão apresentados os dados relacionados à complexidade computacional, mostramos a Tabela 2.3, a qual apresenta o tempo e o espaço computacionais requeridos para executar o Algoritmo 1 (EXEMPLONÚMERO PAR), com base nos parâmetros de entrada utilizados.

Tabela 2.3: Análise de Complexidade de EXEMPLONÚMERO PAR.

Linha	Tempo Computacional	Espaço Computacional
1	$O(1)$	$O(1)$
2 -4	$O(1)$	
Total	$O(1)$	$O(1)$

Retirado da Seção 3.1 de (Cormen *et al.*, 2009), definimos a notação de O grande, a qual determina assintoticamente o limite superior de uma função. Para uma determinada função $g(n)$, denotamos $O(g(n))$ o conjunto de funções

$$O(g(n)) = \{f(n) : \text{existe constantes positivas } c \text{ e } n_0 \text{ tal que } 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}. \quad (2.16)$$

A utilização da notação de O grande nos permite descrever o tempo de execução de um algoritmo ape-

nas inspecionando a estrutura global do algoritmo. Para esse texto, utilizamos essa notação para descrever também o consumo de espaço computacional. Com base na Tabela 2.3, podemos verificar que as entradas $O(1)$ indicam que o tempo ou espaço computacional resultante da instrução em específico independe do tamanho do parâmetro de entrada n fornecido ao Algoritmo 1 (EXEMPLO NÚMERO PAR).

A seguir, definimos notações específicas utilizadas neste trabalho visando a análise de complexidade com base em atributos dos parâmetros de entrada fornecidos.

Seja E uma imagem binária. $|E|$ denota o número de elementos que estejam contidos em E . $[E]$ denota o número de coordenadas que estejam contidas na caixa delimitadora associada a E . A partir dessas duas definições, podemos estabelecer que $|E| \leq [E]$.

A partir do conceito de coordenadas de interesse aplicado aos algoritmos duais, definimos x como a seguinte relação:

$$x = \begin{cases} 0 & : \textit{iteração} = 0 \\ k \cdot (\textit{iteração} - 1) + \textit{índiceElementoEstruturante} & : \textit{iteração} > 0 \end{cases} \quad (2.17)$$

onde k é o número de elementos estruturantes de \mathcal{L} , $\textit{iteração}$ é um inteiro positivo, e $\textit{índiceElementoEstruturante}$ é um valor do intervalo $[1, k]$. O valor x é utilizado como índice com o intuito de relacionar para qual combinação de $\textit{iteração}$ e $\textit{índiceElementoEstruturante}$ que uma determinada imagem é avaliada.

A notação I_x utilizada na análise de complexidade dos algoritmos apresentados possui valor diferente na Seção 3.1 em relação ao restante do texto.

Para a Seção 3.1, a notação de I_x segue a Tabela 2.4.

Tabela 2.4: Definição de I_x aplicada na Seção 3.1.

Operação	Definição de I_x
Transformada da Erosão	$I_x = \varepsilon_{L_x}(\cdot(\varepsilon_{L_1}(\varepsilon_{L_0}(X)))\cdot)$
Transformada da Dilatação	$I_x = \delta_{L_x}(\cdot(\delta_{L_1}(\delta_{L_0}(X)))\cdot)$

Os algoritmos duais apresentados utilizam a Equação 2.6. Dessa forma, a notação de I_x adotada nesse contexto segue a Tabela 2.5.

Tabela 2.5: Definição de I_x aplicada aos algoritmos duais.

Operação	Definição de I_x
Transformada da Erosão	$I_x = \varepsilon_{L_x}(\cdot(\varepsilon_{L_1}(\varepsilon_{L_0}(X)))\cdot)$
Transformada da Dilatação	$I_x = (\varepsilon_{(L_x)^r}(\cdot(\varepsilon_{(L_1)^r}(\varepsilon_{(L_0)^r}(X^c)))\cdot))^c$

A notação L_x é utilizada para facilitar a visualização do cálculo de complexidade computacional. Para verificar qual elemento de \mathcal{L} que L_x identifica, basta verificar a seguinte relação:

$$L_x = \begin{cases} L_0 & : x = 0 \\ L_{((x-1) \bmod k)+1} & : x > 0 \end{cases} \quad (2.18)$$

onde $(u \bmod v)$ representa o módulo da divisão de inteiros u por v .

Como podemos verificar nas definições de x , I_x e L_x , as imagens estão atreladas à iteração em que são analisadas. Dessa forma, no contexto da análise de complexidade, definimos m como a última iteração do operador em que há alterações na imagem de trabalho. Esse valor está atrelado à condição de parada e seu uso será explicitado no próximo capítulo.

Para a análise de espaço computacional, cabe ressaltar que foi considerado que uma variável possui escopo local quanto à sua declaração. Tal fato implica que, ao finalizar o bloco lógico onde se encontra tal declaração de variável, implicitamente os dados armazenados dessa forma são desalocados. Como exemplo, temos a linha 14 do Algoritmo 2 (TRANSFORMADAEROSÃOSIMPLES), a qual alocará espaço relacionado às coordenadas que pertençam à borda de I_{x-1} através da fila *aRemove*. Esta variável, por sua vez, possui como escopo somente o intervalo da linha 11 à 18. Desse modo, consideramos somente a maior instância de *aRemove* para o quesito de espaço computacional.

Capítulo 3

Algoritmos Prévios para Transformada da Erosão e Transformada da Dilatação

Apresentaremos neste capítulo os algoritmos base de Transformada da Erosão e Transformada da Dilatação juntamente com os conceitos utilizados para os refinamentos incrementais até a apresentação de um único algoritmo proposto para efetuar as duas operações. Fixados os parâmetros de uma dada operação, dois algoritmos quaisquer que a implementem devem devolver a mesma imagem resultante.

O trabalho aqui apresentado foi efetuado para imagens e elementos estruturantes binários como entrada dos operadores estudados, sendo a adaptação dos algoritmos de modo a utilizar imagens e elementos estruturantes em níveis de cinza trivial de ser realizada.

Os parâmetros de entrada dos operadores são:

- número de dimensões $d > 1$ para a definição das imagens;
- imagem binária de entrada $X \subset \mathbb{Z}^d$; e
- uma lista $\mathcal{L} = [L_1, \dots, L_k]$ de k elementos estruturantes, de modo que a origem o pertence a cada elemento de \mathcal{L} .

A saída esperada e o fluxo de verificações efetuadas pelos algoritmos deste capítulo foram descritos na Seção 2.4. Note que em todos os algoritmos de Transformada foi utilizada a variável x como artifício para a análise da complexidade computacional dos mesmos.

3.1 Algoritmos Simples

3.1.1 Algoritmo Simples de Transformada da Erosão

O algoritmo simples de Transformada da Erosão aqui apresentado consiste em trabalhar com a imagem temporária *imagemTrabalho* para efetuar a operação da Transformada de modo que valham as seguintes relações:

$$I_0 = X \tag{3.1}$$

$$\begin{aligned} I_{k\text{-iteração}} &= \varepsilon_{(\text{iteração})} \cdot E(X) \\ &= \varepsilon_E(\varepsilon_{(\text{iteração}-1)} \cdot E(X)) \\ &= \varepsilon_E(I_{k\text{-iteração}-1}) \\ &= \varepsilon_{L_k}(\dots(\varepsilon_{L_1}(I_{k\text{-iteração}-1}))\dots) \end{aligned} \tag{3.2}$$

No início de uma dada iteração de valor *iteração*, *imagemTrabalho* equivale à imagem $I_{k \cdot (\textit{iteração} - 1)}$. Ao longo da iteração, são efetuadas k erosões sequenciais, sendo utilizados os elementos estruturantes de \mathcal{L} , de modo que ao final da iteração, *imagemTrabalho* equivalha à imagem $I_{k \cdot \textit{iteração}}$.

Já a condição de parada pode ser verificada em uma dada iteração de valor $m + 1$ através da relação

$$\begin{aligned} I_{k \cdot (m+1)} &= \varepsilon_E(I_{k \cdot m}) \\ &= \varepsilon_{L_k}(\dots(\varepsilon_{L_1}(I_{k \cdot m}))\dots) \\ &= I_{k \cdot m} \end{aligned} \quad (3.3)$$

No Algoritmo 2 TRANSFORMADAEROSÃOSSIMPLES, essa verificação é efetuada através da linha 6 com base em um único critério: o número de elementos estruturantes de \mathcal{L} verificados na iteração imediatamente anterior que não alteraram *imagemTrabalho* equivale ao número k , i.e., não houveram mudanças em *imagemTrabalho* durante toda a iteração de valor $\textit{iteração} = m + 1$.

O trecho entre as linhas 9 e 18 calcula a Erosão de *imagemTrabalho* por E , de modo que a Equação 3.2 será válida após a execução dessas instruções.

Algoritmo 2 TRANSFORMADAEROSÃOSSIMPLES(X, \mathcal{L}, k)

```

1  inicialize resultado como uma imagem de valores inteiros de mesmo tamanho que a caixa delimitadora de  $X$ 
   de modo que o valor  $-1$  seja atrelado a cada uma das coordenadas que estejam dentro dessa caixa
2  inicialize imagemTrabalho como cópia da imagem  $X$ 
3  inicialize índiceElementoEstruturante, iteração, nãoAlterou e  $x$  com o valor 0
4  para todo elemento de coordenadas  $c \in \textit{imagemTrabalho}$  faça
5     o valor de resultado na coordenada  $c$  recebe iteração
6  enquanto nãoAlterou <  $k$  faça
7     iteração  $\leftarrow$  iteração + 1
8     nãoAlterou  $\leftarrow$  0
9     para índiceElementoEstruturante de 1 até  $k$  faça
10         $x \leftarrow x + 1$ 
11        inicialize aRemover como uma fila vazia de coordenadas em  $\mathbb{Z}^d$ 
        ▷ Determina a borda de imagemTrabalho.
12        para todo elemento de coordenadas  $c \in \textit{imagemTrabalho}$  faça
13            se  $(c + L_{\textit{índiceElementoEstruturante}}) \cap \textit{imagemTrabalho} \neq (c + L_{\textit{índiceElementoEstruturante}})$  então
14                insira  $c$  em aRemover
        ▷ Remove a borda de imagemTrabalho e altera resultado de acordo com a borda.
15        para toda coordenada  $c$  em aRemover faça
16            imagemTrabalho  $\leftarrow$  imagemTrabalho  $\setminus \{c\}$ 
17            o valor de resultado na coordenada  $c$  recebe iteração
18        se aRemover estiver vazia então nãoAlterou  $\leftarrow$  nãoAlterou + 1
19  devolva resultado

```

3.1.2 Algoritmo Simples de Transformada da Dilatação

De modo análogo ao algoritmo apresentado na Subseção 3.1.1, o algoritmo simples de Transformada da Erosão aqui apresentado consiste em trabalhar com a imagem temporária *imagemTrabalho* para efetuar

a operação da Transformada da modo que valham as seguintes relações:

$$I_0 = X \quad (3.4)$$

$$\begin{aligned} I_{k \cdot \text{iteração}} &= \delta_{(\text{iteração}) \cdot E}(X) \\ &= \delta_E(\delta_{(\text{iteração} - 1) \cdot E}(X)) \\ &= \delta_E(I_{k \cdot (\text{iteração} - 1)}) \\ &= \delta_{L_k}(\dots(\delta_{L_1}(I_{k \cdot (\text{iteração} - 1)})) \dots) \end{aligned} \quad (3.5)$$

Algoritmo 3 TRANSFORMADADILATAÇÃOSIMPLES(X, \mathcal{L}, k)

```

1  inicialize resultado como uma imagem de valores inteiros de mesmo tamanho que a caixa delimitadora de  $X$ 
   de modo que o valor  $-1$  seja atrelado a cada uma das coordenadas que estejam dentro dessa caixa
2  inicialize imagemTrabalho como cópia da imagem  $X$ 
3  inicialize índiceElementoEstruturante, iteração, nãoAlterou e  $x$  com o valor 0
4  para todo elemento de coordenadas  $c \in \textit{imagemTrabalho}$  faça
5    o valor de resultado na coordenada  $c$  recebe iteração
6  enquanto nãoAlterou <  $k$  faça
7    iteração  $\leftarrow$  iteração + 1
8    nãoAlterou  $\leftarrow$  0
9    para índiceElementoEstruturante de 1 até  $k$  faça
10      $x \leftarrow x + 1$ 
11     inicialize alInserir como uma fila vazia de coordenadas em  $\mathbb{Z}^d$ 
        $\triangleright$  Determina a borda de imagemTrabalho.
12     para todo elemento de coordenadas  $c \in \textit{imagemTrabalho}$  faça
13       para todo elemento de coordenadas  $\textit{delta} \in (c + L_{\textit{índiceElementoEstruturante}})$  faça
14         se  $\textit{delta} \notin \textit{imagemTrabalho}$  então
15           insira  $\textit{delta}$  em alInserir
        $\triangleright$  Insira a borda em imagemTrabalho e altera resultado de acordo com a borda.
16     para toda coordenada  $\textit{delta}$  em alInserir
17        $\textit{imagemTrabalho} \leftarrow \textit{imagemTrabalho} \cup \{\textit{delta}\}$ 
18       o valor de resultado na coordenada  $\textit{delta}$  recebe iteração
19     se alInserir estiver vazia então nãoAlterou  $\leftarrow$  nãoAlterou + 1
20  devolva resultado

```

No início de uma dada iteração de valor *iteração*, *imagemTrabalho* equivale à imagem $I_{k \cdot (\textit{iteração} - 1)}$. Ao longo da iteração, são efetuadas k dilatações sequenciais, sendo utilizados os elementos estruturantes de \mathcal{L} , de modo que ao final da iteração, *imagemTrabalho* equivalha à imagem $I_{k \cdot \textit{iteração}}$.

Já a condição de parada pode ser verificado em uma dada iteração de valor $m + 1$ através da relação

$$\begin{aligned} I_{k \cdot (m+1)} &= \delta_E(I_{k \cdot m}) \\ &= \delta_{L_k}(\dots(\delta_{L_1}(I_{k \cdot m})) \dots) \\ &= I_{k \cdot m} \end{aligned} \quad (3.6)$$

No Algoritmo 3 TRANSFORMADADILATAÇÃOSIMPLES, essa verificação é efetuada através da linha 6 com base em um único critério: o número de elementos estruturantes de \mathcal{L} verificados na iteração imediatamente anterior que não alteraram *imagemTrabalho* equivale ao número k , i.e., não houveram mudanças

em *imagemTrabalho* durante toda a iteração de valor $iteração = m + 1$.

O trecho entre as linhas 9 e 19 calcula a dilatação de *imagemTrabalho* por E , de modo que a Equação 3.5 será válida após a execução dessas instruções.

3.2 Algoritmo Dual

O primeiro dos refinamentos incrementais para o algoritmo proposto é a utilização de um único algoritmo dual para as operações de Transformada da Erosão e Transformada da Dilatação, de modo que o comportamento do algoritmo é controlado através de um parâmetro binário. Todos os algoritmos apresentados a partir desta seção utilizam como base a Transformada da Erosão com o intuito de poder visualizar pontualmente as diferenças. A seguinte relação de dualidade, baseada nas Equações 2.6 e 2.13, foi adotada:

$$\begin{aligned} \delta_E(X) &= (\varepsilon_{E^r}(X^c))^c \\ &= (\varepsilon_{(L_k)^r}(\dots((\varepsilon_{(L_1)^r}(X^c))^c)\dots))^c, \end{aligned} \quad (3.7)$$

onde X^c é o complemento da imagem binária X no universo \mathbb{Z}^d e E^r é a reflexão de E pela origem $o \in \mathbb{Z}^d$. O parâmetro *TransformadaErosão* determina a operação desejada: se **verdadeiro**, então é escolhida a Transformada da Erosão, senão é escolhida a Transformada da Dilatação.

A escolha de basearmos os algoritmos duais apresentados neste capítulo de acordo com a Transformada da Erosão pode ser justificada com base nos algoritmos apresentados na seção anterior. Todas as coordenadas de *aRemover* verificadas pela linha 15 do Algoritmo 2 (TRANSFORMADAEROSÃOSIMPLES) serão analisadas por essa linha uma única vez, pois o número máximo de remoções do elemento de *imagemTrabalho* atrelado a essa coordenada ao longo do algoritmo é 1^* , enquanto que pode ocorrer a repetição de coordenadas em *aInserir* analisadas pela linha 16 do Algoritmo 3 (TRANSFORMADADILATAÇÃOSIMPLES), as quais podem decorrer de múltiplas inserções da coordenada na linha 15 ocorridas em uma mesma combinação de *iteração* e $L_{índiceElementoEstruturante}$.

A condição de parada do Algoritmo 6 (TRANSFORMADADUAL) visa atender às Equações 3.3 e 3.6, de acordo com a operação desejada, com o auxílio da definição de coordenadas de interesse, a qual foi efetuada na Seção 2.4. Essa condição é verificada na linha 7 e é composta por um único critério: o número de elementos estruturantes de \mathcal{L} verificados na iteração imediatamente anterior que não alteraram *imagemTrabalho* equivale ao número k , i.e., não houveram mudanças em *imagemTrabalho* durante toda a iteração de valor $iteração = m + 1$.

Algoritmo 4 TRANSFORMADADILATAÇÃODUAL(X, \mathcal{L}, k)

1 **devolva** TRANSFORMADADUAL(X, \mathcal{L}, k , **falso**)

Algoritmo 5 TRANSFORMADAEROSÃODUAL(X, \mathcal{L}, k)

1 **devolva** TRANSFORMADADUAL(X, \mathcal{L}, k , **verdadeiro**)

*Essa constatação é garantida pelo fato de todos os elementos de \mathcal{L} possuírem suas respectivas origens.

Algoritmo 6 TRANSFORMADADUAL($X, \mathcal{L}, k, \acute{e}$ TransformadaErosão)

```

1  inicialize resultado como uma imagem de valores inteiros de mesmo tamanho que a caixa delimitadora de  $X$ 
   de modo que o valor  $-1$  seja atrelado a cada uma das coordenadas que estejam dentro dessa caixa
2  inicialize imagemTrabalho como c3pia da imagem  $X$ 
3  inicialize indiceElementoEstruturante, iteracao, n3oAlterou e  $x$  com o valor 0
4  inicialize removeCandidato com o valor falso
5  para todo elemento de coordenadas  $c \in \textit{imagemTrabalho}$  faa
6    o valor de resultado na coordenada  $c$  recebe iteracao
7  enquanto n3oAlterou  $< k$  faa
8    iteracao  $\leftarrow \textit{iteracao} + 1$ 
9    n3oAlterou  $\leftarrow 0$ 
10   para indiceElementoEstruturante de 1 at3  $k$  faa
11      $x \leftarrow x + 1$ 
12     inicialize aAtualizar como uma fila vazia de coordenadas em  $\mathbb{Z}^d$ 
      $\triangleright$  Determina a borda de imagemTrabalho.
13     para toda coordenada de interesse  $c$  na caixa delimitadora de imagemTrabalho faa
14       se 3TransformadaEros3o ent3o
15          $\textit{removeCandidato} \leftarrow (c + L_{\textit{indiceElementoEstruturante}}) \cap \textit{imagemTrabalho}$ 
            $\neq (c + L_{\textit{indiceElementoEstruturante}})$ 
16       sen3o
17          $\textit{removeCandidato} \leftarrow (c + (L_{\textit{indiceElementoEstruturante}})^r) \cap \textit{imagemTrabalho}^c$ 
            $\neq (c + (L_{\textit{indiceElementoEstruturante}})^r)$ 
18       se removeCandidato ent3o
19         insira  $c$  em aAtualizar
      $\triangleright$  Altera imagemTrabalho e resultado de acordo com a borda.
20     para toda coordenada de interesse  $c$  em aAtualizar faa
21       se 3TransformadaEros3o ent3o
22          $\textit{imagemTrabalho} \leftarrow \textit{imagemTrabalho} \setminus \{c\}$ 
23       sen3o
24          $\textit{imagemTrabalho}^c \leftarrow \textit{imagemTrabalho}^c \setminus \{c\}$ 
25       o valor de resultado na coordenada  $c$  recebe iteracao
26       se aAtualizar estiver vazia ent3o n3oAlterou  $\leftarrow \textit{n3oAlterou} + 1$ 
27  devolva resultado

```

3.3 Algoritmo Dual de Vetor e Lista

Na defini33o das linhas 13, 15 e 17 do Algoritmo 6 (TRANSFORMADADUAL), existem verifica33es impl3citas: todas as coordenadas das caixas delimitadoras de *imagemTrabalho*, $L_{\textit{indiceElementoEstruturante}}$ e $L_{\textit{indiceElementoEstruturante}}$ — respectivamente — s3o verificadas cada vez que essas instru33es s3o executadas. Visando reduzir o n3mero de compara33es efetuadas em coordenadas que n3o s3o classificadas como coordenadas de interesse, surge o segundo dos refinamentos incrementais: armazenar em vetores e listas as coordenadas de interesse, de modo que todas as coordenadas da caixa delimitadora de X e dos elementos estruturantes de \mathcal{L} sejam percorridas uma 3nica vez para a inser33o nessas estruturas de dados. Nos vetores s3o armazenados dados que n3o s3o alterados ao longo da execu33o do algoritmo, i.e., constantes, enquanto que nas listas duplamente encadeadas s3o armazenados dados que possam ser alterados ao longo da execu33o do algoritmo, i.e., vari3veis. Apesar da lineariza33o dos dados, ainda 3 necess3ria a utiliza33o de uma imagem como estrutura de dados, pois a mesma permite acesso direto 3s coordenadas, como podemos verificar nas instru33es das linhas 22 e 24 de Algoritmo 12 (ALGORITMODUALLINEAR).

Visando facilitar o entendimento, o Algoritmo 9 (TRANSFORMADADUALLINEAR) foi dividido em tr3s etapas: (i) a inicializa33o de dados referente aos elementos estruturantes de \mathcal{L} (Algoritmo 10 — LINEARIZEELEMENTOESTRUTURANTE), (ii) a inicializa33o de dados referente 3 imagem bin3ria X (Algo-

ritmo 11 — `LINEARIZEIMAGEMLINEAR`), e (iii) o algoritmo propriamente dito (Algoritmo 12 — `ALGORITMODUALLINEAR`).

Algoritmo 7 `TRANSFORMADADILATAÇÃOLINEAR`(X, \mathcal{L}, k)

1 **devolva** `TRANSFORMADADUALLINEAR`($X, \mathcal{L}, k, \text{falso}$)

Algoritmo 8 `TRANSFORMADAEROSÃOLINEAR`(X, \mathcal{L}, k)

1 **devolva** `TRANSFORMADADUALLINEAR`($X, \mathcal{L}, k, \text{verdadeiro}$)

Algoritmo 9 `TRANSFORMADADUALLINEAR`($X, \mathcal{L}, k, \text{éTransformadaErosão}$)

1 $\text{cardinalidadeUnião}, \text{elementosUnião}, \text{cardinalidadeL}, \text{elementosL} \leftarrow \text{LINEARIZEELEMENTOESTRUTURANTE}(\mathcal{L}, k)$
 2 $\text{listaCoordenadas} \leftarrow \text{LINEARIZEIMAGEMLINEAR}(X, \text{éTransformadaErosão})$
 3 **devolva** `ALGORITMODUALLINEAR`($X, \text{listaCoordenadas}, \text{elementosUnião}, \text{cardinalidadeL}, \text{elementosL}, k, \text{éTransformadaErosão}$)

No caso específico do Algoritmo 9 (`TRANSFORMADADUALLINEAR`), note que a variável *cardinalidadeUnião* calculada não é utilizada. Isso ocorre pelo fato de que o Algoritmo 10 (`LINEARIZEELEMENTOESTRUTURANTE`) foi desenhado de modo a ser utilizado também pelos algoritmos posteriores, os quais utilizam a informação contida nessa variável.

O Algoritmo 10 (`LINEARIZEELEMENTOESTRUTURANTE`) visa linearizar os dados referentes aos elementos estruturantes de \mathcal{L} em quatro etapas: (i) o cálculo da menor caixa delimitadora da união dos elementos estruturantes de \mathcal{L} , obtido entre as linhas 5 e 7, e a alocação da imagem binária *união*, na linha 8, com o objetivo de fornecer acesso direto à informação efetuado na linha 13, (ii) enfileiramento das coordenadas dos elementos da união dos elementos estruturantes de \mathcal{L} , obtido entre as linhas 9 e 16, (iii) a criação e o preenchimento de um vetor que armazene as coordenadas enfileiradas, calculado entre as linhas 17 e 21, e (iv) a criação e o preenchimento de k vetores de modo que cada vetor de índice *índiceElementoEstruturante* armazena os índices dos elementos dos respectivos $L_{\text{índiceElementoEstruturante}}$ com base no vetor da união dos elementos estruturantes calculado no passo anterior, efetuado entre as linhas 22 e 31.

Para facilitar o entendimento do Algoritmo 10 (`LINEARIZEELEMENTOESTRUTURANTE`), optamos por apresentar um exemplo visual com 3 elementos estruturantes em \mathbb{Z}^2 de tamanho 3×3 através da Figura 3.1: a ordem lexicográfica de seus itens reflete a ordem em que as estruturas representadas são utilizadas ou alocadas. Relacionaremos os itens apresentados na listagem de etapas do algoritmo à figura apresentada. Note que, como uma das restrições relativa aos elementos estruturantes de \mathcal{L} é a existência de suas respectivas origens, então a cardinalidade armazenada nas estruturas difere em um do número de elementos que cada elemento estruturante de \mathcal{L} possui.

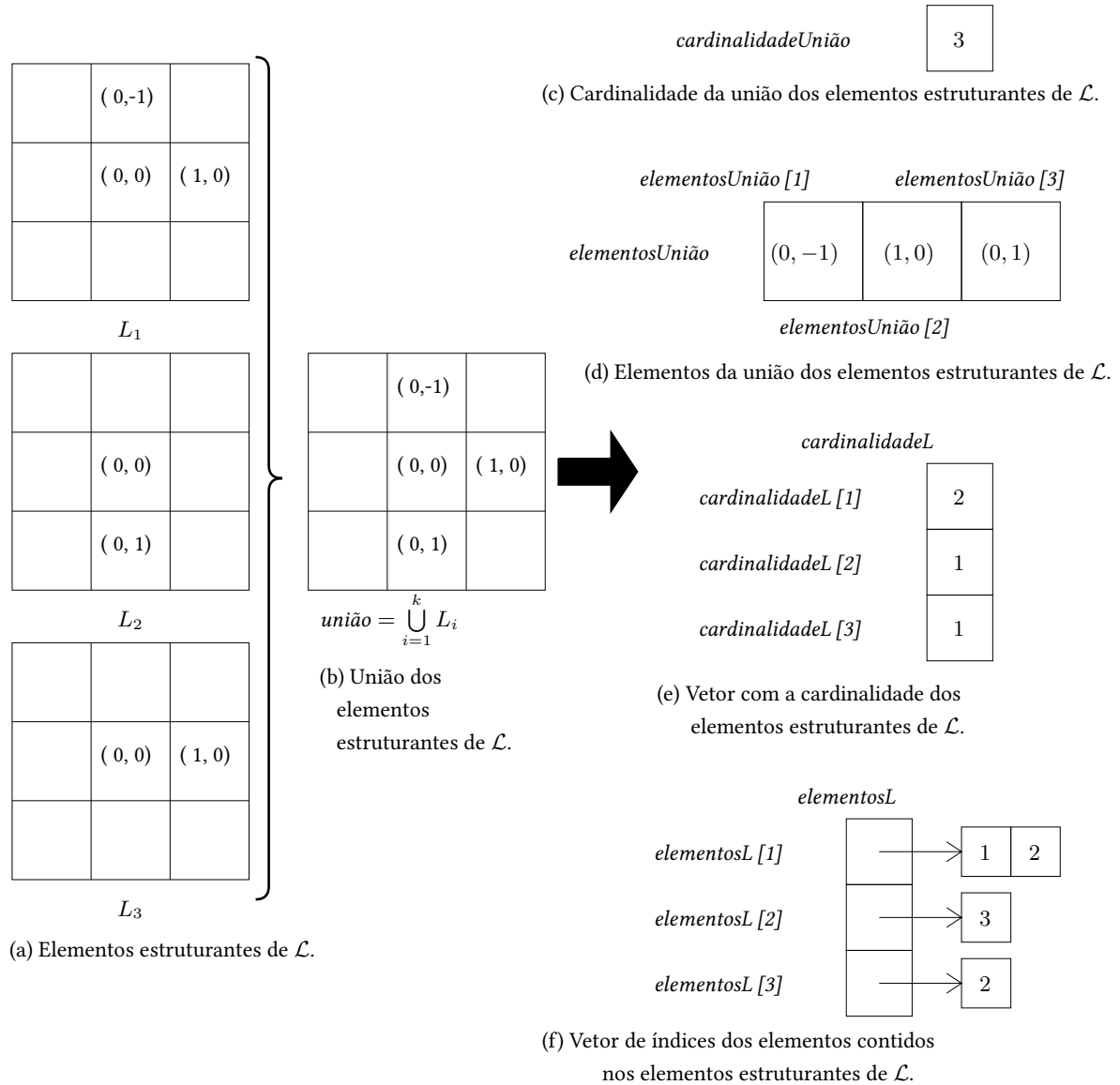


Figura 3.1: Exemplo visual do Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE).

Relativo à Figura 3.1: o item (a) representa os elementos estruturantes contidos na lista \mathcal{L} . O cálculo da união dos elementos estruturantes de \mathcal{L} — representado pelo item (b) — e da cardinalidade de elementos contidos na união — representado pelo item (c) — são efetuados nas etapas **i** e **ii**. A estrutura representada pelo item (d) e seu conteúdo são calculados na etapa **iii**, enquanto que a linearização dos elementos de \mathcal{L} é armazenada nas estruturas representadas pelos itens (e) e (f), sendo estes calculados com base no vetor representado pelo item (d) na etapa **iv**.

Algoritmo 10 LINEARIZEELEMENTOESTRUTURANTE(\mathcal{L}, k)

```

1  inicialize cardinalidadeL como um vetor de  $k$  inteiros
2  inicialize elementosL como um vetor de vetores não alocados de  $k$  posições
3  inicialize cardinalidadeUnião, índiceCoordenada e índiceElementoEstruturante com o valor 0
4  inicialize fila como uma fila vazia de coordenadas em  $\mathbb{Z}^d$ 
   ▷ Calcula a união dos elementos estruturantes de  $\mathcal{L}$ .
5  calcule a menor caixa delimitadora de  $o \in \mathbb{Z}^d$  como MCD
6  para índiceElementoEstruturante de 1 até  $k$  faça
7     atualize MCD como a menor caixa delimitadora a cobrir MCD e  $L_{\text{índiceElementoEstruturante}}$ 
   ▷ Calcula a união dos elementos estruturantes de  $\mathcal{L}$ .
8  inicialize união como uma imagem binária vazia em  $\mathbb{Z}^d$  de tamanho MCD
9  para índiceElementoEstruturante de 1 até  $k$  faça
10     o valor de cardinalidadeL na posição índiceElementoEstruturante recebe 0
11     para todo elemento de coordenadas  $p \in L_{\text{índiceElementoEstruturante}}$  faça
12         o valor de cardinalidadeL na posição índiceElementoEstruturante aumenta em 1
13         se  $p \notin \textit{união}$  então
14             o valor de cardinalidadeUnião aumenta em 1
15              $\textit{união} \leftarrow \textit{união} \cup \{p\}$ 
16             enfileire  $p$  em fila
17     inicialize elementosUnião como um vetor de coordenadas em  $\mathbb{Z}^d$  de cardinalidadeUnião posições
18     enquanto fila não está vazio faça
19         desenfileire  $p$  de fila
20          $\textit{índiceCoordenada} \leftarrow \textit{índiceCoordenada} + 1$ 
21         o valor de elementosUnião na posição índiceCoordenada recebe  $p$ 
   ▷ Lineariza os dados relacionados aos elementos estruturantes de  $\mathcal{L}$ .
22     para índiceElementoEstruturante de 1 até  $k$  faça
23         inicialize vetorAtual como um vetor de inteiros de  $\textit{cardinalidadeL}[\textit{índiceElementoEstruturante}]$  posições
24         inicialize índiceUnião com o valor 0
25         inicialize índiceVetorAtual com o valor 0
26         enquanto  $\textit{índiceUnião} < \textit{cardinalidadeUnião}$  e
            $\textit{índiceVetorAtual} < \textit{cardinalidadeL}[\textit{índiceElementoEstruturante}]$  faça
27              $\textit{índiceUnião} \leftarrow \textit{índiceUnião} + 1$ 
28             se  $\textit{elementosUnião}[\textit{índiceUnião}] \in L_{\text{índiceElementoEstruturante}}$  então
29                  $\textit{índiceVetorAtual} \leftarrow \textit{índiceVetorAtual} + 1$ 
30                 o valor de vetorAtual na posição índiceVetorAtual recebe  $\textit{índiceUnião}$ 
31             o valor de elementosL na posição índiceElementoEstruturante recebe vetorAtual
32     devolva cardinalidadeUnião, elementosUnião, cardinalidadeL, elementosL

```

O Algoritmo 11 (LINEARIZEIMAGEMLINEAR) é responsável por verificar todas as posições da caixa delimitadora de X e armazenar as coordenadas de interesse[†] na lista duplamente ligada *listaCoordenadas*, a qual será alterada no Algoritmo 12 (ALGORITMODUALLINEAR).

Algoritmo 11 LINEARIZEIMAGEMLINEAR(X, \acute{e} TransformadaErosão)

```

1  inicialize listaCoordenadas como uma cabeça de lista duplamente encadeada de coordenadas em  $\mathbb{Z}^d$ 
2  para toda coordenada  $c$  na caixa delimitadora de  $X$  faça
3     se  $((c \in X) = \acute{e}\textit{TransformadaErosão})$  então
   ▷ Insere todas as coordenadas de interesse de  $X$  em listaCoordenadas.
4     insira  $c$  em listaCoordenadas
5  devolva listaCoordenadas

```

[†] Conforme definido na Seção 3.2

Algoritmo 12 ALGORITMODUALLINEAR(X , listaCoordenadas, elementosUnião, cardinalidadeL, elementosL, k , \acute{e} TransformadaErosão)

```

1  inicialize resultado como uma imagem de valores inteiros de mesmo tamanho que a caixa delimitadora de  $X$ 
  de modo que o valor  $-1$  seja atrelado a cada uma das coordenadas que estejam dentro dessa caixa
2  inicialize imagemTrabalho como c3pia da imagem  $X$ 
3  inicialize 3ndiceElementoEstruturante, 3ndiceVetorAtual, iteração, n3oAlterou e  $x$  com o valor 0
4  inicialize removaCandidato com o valor falso
5  para todo elemento de coordenadas  $c \in \textit{imagemTrabalho}$  faa
6    o valor de resultado na coordenada  $c$  recebe iteração
7  enquanto listaCoordenadas n3o contiver somente sua cabea e n3oAlterou  $< k$  faa
8    iteração  $\leftarrow \textit{iteração} + 1$ 
9    n3oAlterou  $\leftarrow 0$ 
10   para 3ndiceElementoEstruturante de 1 at3  $k$  faa
11      $x \leftarrow x + 1$ 
12     inicialize aRemover como uma fila vazia de coordenadas em  $\mathbb{Z}^d$ 
      $\triangleright$  Determina a borda de imagemTrabalho.
13     para toda coordenada de interesse  $c$  em listaCoordenadas faa
14       inicialize vetorAtual como elementosL[3ndiceElementoEstruturante]
15       3ndiceVetorAtual  $\leftarrow 0$ 
16       removaCandidato  $\leftarrow$  falso
17       enquanto n3o removaCandidato e
         3ndiceVetorAtual  $<$  cardinalidadeL[3ndiceElementoEstruturante] faa
18         3ndiceVetorAtual  $\leftarrow \textit{3ndiceVetorAtual} + 1$ 
19         3ndicePosi33oUn3o  $\leftarrow \textit{vetorAtual}$ [3ndiceVetorAtual]
20         delta  $\leftarrow \textit{elementosUn3o}$ [3ndicePosi33oUn3o]
21         se  $\acute{e}$ TransformadaEros3o ent3o
22           removaCandidato  $\leftarrow (c + \textit{delta}) \notin \textit{imagemTrabalho}$ 
23         sen3o
24           removaCandidato  $\leftarrow (c - \textit{delta}) \notin \textit{imagemTrabalho}^c$ 
25         se removaCandidato ent3o
26           insira  $c$  em aAtualizar
27           remova  $c$  de listaCoordenadas
      $\triangleright$  Altera imagemTrabalho e resultado de acordo com a borda.
28     para toda coordenada de interesse  $c$  em aAtualizar faa
29       se  $\acute{e}$ TransformadaEros3o ent3o
30         imagemTrabalho  $\leftarrow \textit{imagemTrabalho} \setminus \{c\}$ 
31       sen3o
32         imagemTrabalho $c$   $\leftarrow \textit{imagemTrabalho}^c \setminus \{c\}$ 
33       o valor de resultado na coordenada  $c$  recebe iteração
34       se aAtualizar estiver vazia ent3o n3oAlterou  $\leftarrow \textit{n3oAlterou} + 1$ 
35  devolva resultado

```

O Algoritmo 12 (ALGORITMODUALLINEAR) 3 a cerne do algoritmo dual de vetor e lista para os operadores de Transformada da Dilata33o e Transformada da Eros3o. Quando comparado ao Algoritmo 6 (TRANSFORMADADUAL), verificamos que, devido 3 lineariza33o dos dados referentes 3 imagem de entrada, existem dois crit3rios que comp3em a condi33o de parada, a qual 3 verificada na linha 7: (i) *listaCandidatos* n3o possui nenhuma coordenada de interesse, ou (ii) o n3mero de elementos estruturantes de \mathcal{L} verificados na itera33o imediatamente anterior que n3o alteraram *imagemTrabalho* equivale ao n3mero k , i.e., n3o houveram mudan3as em *imagemTrabalho* durante toda a itera33o de valor *iteração* $= m + 1$.

No Algoritmo 6 (TRANSFORMADADUAL), a verifica33o de cada elemento de $L_{\textit{3ndiceElementoEstruturante}}$ 3 realizada de maneira impl3cita nas linhas 15 e 17. Com a lineariza33o dos elementos estruturantes de \mathcal{L} , o Algoritmo 12 (ALGORITMODUALLINEAR) efetua a mesma verifica33o de modo expl3cito para cada elemento

de $L_{\text{índiceElementoEstruturante}}$ nas linhas 18 a 20, como consequência direta do uso de vetores para linearizar os dados representativos de cada elemento estruturante de \mathcal{L} .

3.4 Algoritmo Dual de Vetor e Lista de Consumo de Borda

Na linha 4 do Algoritmo 11 (LINEARIZEIMAGEMLINEAR), o único critério para a inserção em *listaCoordenadas* é que c seja uma coordenada de interesse. Esse fato juntamente com o que decorre na linha 13 do Algoritmo 12 (ALGORITMODUALLINEAR) resultam na questão de que todas as coordenadas de interesse remanescente em *listaCoordenadas* são analisadas para cada vez que algum elemento $L_{\text{índiceElementoEstruturante}}$ é verificado.

Com o intuito de reduzir o número de coordenadas de interesse a serem analisadas para cada vez que algum elemento $L_{\text{índiceElementoEstruturante}}$ é verificado, aplicamos o terceiro refinamento incremental: a adoção da invariante, a qual garante que, no início de cada vez que algum elemento $L_{\text{índiceElementoEstruturante}}$ for verificado no algoritmo de Transformada propriamente dito, *listaCoordenadas* conterá somente as coordenadas de interesse que pertençam à borda de *imagemTrabalho* de acordo com a união dos elementos estruturantes de \mathcal{L} , conforme definição de acordo com o descrito no final da Seção 2.3. Ao adotarmos uma invariante num algoritmo, devemos garanti-la através das fases de inicialização, manutenção e término. Tais fases serão verificadas nos algoritmos apresentados nesta seção.

Note que as listas *aRemover*, *aInserir* e *aAtualizar* utilizadas nos algoritmos apresentados anteriormente já armazenavam a borda para cada combinação de *iteração* e *índiceElementoEstruturante*, ainda que implicitamente, e assim foram apresentadas para demonstrar as modificações proporcionadas pelos refinamentos incrementais de forma mais transparente. Uma outra maneira de validar o fato é observar as tabelas de complexidade computacional apresentadas no Apêndice A para verificar em quais instruções tais bordas eram utilizadas nos algoritmos anteriores.

Assim como na seção anterior, o Algoritmo 15 (TRANSFORMADADUALLINEARDEBORDA) foi dividido em três etapas: (i) a inicialização de dados referente aos elementos estruturantes de \mathcal{L} (Algoritmo 10 – LINEARIZEELEMENTOESTRUTURANTE), (ii) a inicialização de dados referente à imagem binária X (Algoritmo 16 – LINEARIZEBORDAIMAGEM), e (iii) o algoritmo propriamente dito (Algoritmo 18 – ALGORITMODUALLINEARDEBORDA).

Algoritmo 13 TRANSFORMADADILATAÇÃOBORDA(X, \mathcal{L}, k)

1 **devolva** TRANSFORMADADUALLINEARDEBORDA ($X, \mathcal{L}, k, \text{falso}$)

Algoritmo 14 TRANSFORMADAEROSÃOBORDA(X, \mathcal{L}, k)

1 **devolva** TRANSFORMADADUALLINEARDEBORDA ($X, \mathcal{L}, k, \text{verdadeiro}$)

Algoritmo 15 TRANSFORMADADUALLINEARDEBORDA($X, \mathcal{L}, k, \acute{e}$ TransformadaErosão)

```

1  cardinalidadeUnião, elementosUnião, cardinalidadeL, elementosL ← LINEARIZEELEMENTOESTRUTURANTE( $\mathcal{L}, k$ )
2  listaCoordenadas, imagemCandidato
   ← LINEARIZEBORDAIMAGEM( $X, cardinalidadeUnião, elementosUnião, \acute{e}$ TransformadaErosão)
3  devolva ALGORITMODUALLINEARDEBORDA( $X, listaCoordenadas, imagemCandidato, cardinalidadeUnião,$ 
    $elementosUnião, cardinalidadeL, elementosL, k, \acute{e}$ TransformadaErosão)

```

O Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE) é utilizado com o intuito de linearizar os dados referentes aos elementos estruturantes de \mathcal{L} da mesma forma que fora exemplificado na seção anterior.

O Algoritmo 16 (LINEARIZEBORDAIMAGEM) armazena as coordenadas de interesse que pertençam à borda inicial, de modo a cumprir a fase de inicialização da invariante apresentada nesta seção. Além dessa modificação, é inicializada uma imagem binária *imagemCandidato* com o intuito de marcar as coordenadas que já foram inseridas em *listaCoordenadas* para a manutenção da invariante.

Algoritmo 16 LINEARIZEBORDAIMAGEM($X, cardinalidadeUnião, elementosUnião, \acute{e}$ TransformadaErosão)

```

1  inicialize listaCoordenadas como uma cabeça de lista duplamente encadeada de coordenadas em  $\mathbb{Z}^d$ 
2  inicialize índicePosiçãoUnião com o valor 0
3  inicialize insiraCandidato com o valor falso
4  inicialize imagemCandidato como uma imagem binária vazia de mesmo tamanho que
   a caixa delimitadora de  $X$ 
   ▷ Efetua a inicialização da invariante adotada para TRANSFORMADADUALLINEARDEBORDA.
5  para toda coordenada  $c$  na caixa delimitadora de  $X$  faça
6     se  $((c \in X) = \acute{e}$ TransformadaErosão) então
7         índicePosiçãoUnião ← 0
8         insiraCandidato ← falso
9         enquanto não insiraCandidato e índicePosiçãoUnião < cardinalidadeUnião faça
10            índicePosiçãoUnião ← índicePosiçãoUnião + 1
11            delta ← elementosUnião[índicePosiçãoUnião]
12            se  $\acute{e}$ TransformadaErosão então
13                insiraCandidato ←  $(c + delta) \notin$  imagemTrabalho
14            senão
15                insiraCandidato ←  $(c - delta) \notin$  imagemTrabalhoc
16            se insiraCandidato então
17                ▷  $c$  é uma coordenada de interesse que pertence à borda de imagemTrabalho de
18                acordo com a união dos elementos de  $\mathcal{L}$ .
19                insira  $c$  em listaCoordenadas
20            imagemCandidato ← imagemCandidato  $\cup \{c\}$ 
21 devolva listaCoordenadas, imagemCandidato

```

Ao compararmos o Algoritmo 12 (ALGORITMODUALLINEAR) com o Algoritmo 18 (ALGORITMODUALLINEARDEBORDA), podemos verificar que as diferenças entre eles consistem em: (i) a ausência da variável *nãoAlterou*, o qual tornou-se desnecessário devido à invariante utilizada; e (ii) o uso do Algoritmo 17 (ATUALIZAÇÃOODUALLINEARDEBORDA) na linha 27 de ALGORITMODUALLINEARDEBORDA, o qual é responsável pela fase de manutenção da invariante apresentada nesta seção. Apesar do conteúdo da linha 13 de ALGORITMODUALLINEAR e da linha 12 do ALGORITMODUALLINEARDEBORDA serem o mesmo, quando fixadas as variáveis *iteração* e *índiceElementoEstruturante* (combinação representada por x), o número de coordenadas de interesse armazenadas em *listaCoordenadas* do algoritmo apresentado nessa seção

é menor ou igual ao número de coordenadas de interesse armazenadas na variável homônima do algoritmo apresentado na seção anterior, como consequência do uso da invariante apresentada nesta seção.

As linhas 7 a 23 do Algoritmo 17 (ATUALIZAÇÃODUALLINEARDEBORDA) efetuam a manutenção da invariante apresentada através da inserção de coordenadas de interesse que nunca estiveram em *listaCandidatos*, fato que pode ser verificado com o auxílio de *imagemCandidato*. Para determinarmos os possíveis candidatos, é calculada a borda de *imagemTrabalho* através da iteração dos elementos da lista *aAtualizar* e da iteração dos elementos que pertençam à união dos elementos estruturantes de \mathcal{L} .

Algoritmo 17 ATUALIZAÇÃODUALLINEARDEBORDA(*imagemTrabalho*, *listaCoordenadas*, *imagemCandidato*, *aAtualizar*, *cardinalidadeUnião*, *elementosUnião*, *éTransformadaErosão*, *iteração*, *resultado*)

```

    ▷ Altera imagemTrabalho e resultado de acordo com a borda.
1  para toda coordenada de interesse c em aAtualizar faça
2      se éTransformadaErosão então
3          imagemTrabalho ← imagemTrabalho \ {c}
4      senão
5          imagemTrabalhoc ← imagemTrabalhoc \ {c}
6      o valor de resultado na coordenada c recebe iteração
7  para toda coordenada de interesse c em aAtualizar faça
8      ▷ Efetua a manutenção da invariante adotada para TRANSFORMADA DUAL LINEAR DE BORDA.
9      para índicePosiçãoUnião de 1 a cardinalidadeUnião faça
10         ▷ É necessário verificar cada c com cada delta para a determinação das
11         coordenadas de interesse que pertençam à nova borda (manutenção).
12         achouCandidato ← falso
13         delta ← elementosUnião[índicePosiçãoUnião]
14         se éTransformadaErosão então
15             possívelCandidato ← (c - delta)
16         senão
17             possívelCandidato ← (c + delta)
18         se possívelCandidato está dentro da caixa delimitadora de imagemTrabalho então
19             se possívelCandidato ∉ imagemCandidato então
20                 se éTransformadaErosão então
21                     se possívelCandidato ∈ imagemTrabalho então achouCandidato ← verdadeiro
22                 senão
23                     se possívelCandidato ∈ imagemTrabalhoc então achouCandidato ← verdadeiro
24         se achouCandidato então
25             ▷ A coordenada de interesse possívelCandidato será inserida em listaCoordenadas.
26             insira possívelCandidato em listaCoordenadas
27             imagemCandidato ← imagemCandidato ∪ {possívelCandidato}
28 devolva imagemTrabalho, listaCoordenadas, imagemCandidato, resultado

```

Algoritmo 18 ALGORITMO DUAL LINEAR DE BORDA(X , listaCoordenadas, imagemCandidato, cardinalidadeUnião, elementosUnião, cardinalidadeL, elementosL, k , \acute{e} TransformadaErosão)

```

1  inicialize resultado como uma imagem de valores inteiros de mesmo tamanho que a caixa delimitadora de  $X$ 
  de modo que o valor  $-1$  seja atrelado a cada uma das coordenadas que estejam dentro dessa caixa
2  inicialize imagemTrabalho como cópia da imagem  $X$ 
3  inicialize índiceElementoEstruturante, índiceVetorAtual, iteração e  $x$  com o valor 0
4  inicialize removeCandidato com o valor falso
5  para todo elemento de coordenadas  $c \in$  imagemTrabalho faça
6    o valor de resultado na coordenada  $c$  recebe iteração
7  enquanto listaCoordenadas não contiver somente sua cabeça faça
8    iteração  $\leftarrow$  iteração + 1
9    para índiceElementoEstruturante de 1 até  $k$  faça
10      $x \leftarrow x + 1$ 
11     inicialize aAtualizar como uma fila vazia de coordenadas em  $\mathbb{Z}^d$ 
      $\triangleright$  Determina a borda de imagemTrabalho.
12     para toda coordenada de interesse  $c$  em listaCoordenadas faça
13       inicialize vetorAtual como elementosL[índiceElementoEstruturante]
14       índiceVetorAtual  $\leftarrow$  0
15       removeCandidato  $\leftarrow$  falso
16       enquanto não removeCandidato e
         índiceVetorAtual  $<$  cardinalidadeL[índiceElementoEstruturante] faça
17         índiceVetorAtual  $\leftarrow$  índiceVetorAtual + 1
18         índicePosiçãoUnião  $\leftarrow$  vetorAtual[índiceVetorAtual]
19         delta  $\leftarrow$  elementosUnião[índicePosiçãoUnião]
20         se  $\acute{e}$ TransformadaErosão então
21           removeCandidato  $\leftarrow$   $(c + \textit{delta}) \notin$  imagemTrabalho
22         senão
23           removeCandidato  $\leftarrow$   $(c - \textit{delta}) \notin$  imagemTrabalhoc
24         se removeCandidato então
25           insira  $c$  em aAtualizar
26           remova  $c$  de listaCoordenadas
      $\triangleright$  Altera imagemTrabalho e resultado de acordo com a borda.
27     imagemTrabalho, listaCoordenadas, imagemCandidato, resultado
      $\leftarrow$  ATUALIZAÇÃO DUAL LINEAR DE BORDA(imagemTrabalho, listaCoordenadas, imagemCandidato,
     aAtualizar, cardinalidadeUnião, elementosUnião,  $\acute{e}$ TransformadaErosão, iteração, resultado)
28 devolva resultado

```

Capítulo 4

Algoritmo Proposto para Transformada da Dilatação e Transformada da Erosão

No capítulo anterior, inicialmente foram apresentados algoritmos simples para a Transformada da Dilatação e Transformada da Erosão seguidos de três refinamentos incrementais, sendo apresentado um novo algoritmo após a aplicação de cada um desses refinamentos, de maneira incremental. Segue-se uma lista dos refinamentos apresentados:

1. a utilização de um único algoritmo dual, em conjunto com o conceito de coordenadas de interesse, para atender as operações de Transformada da Dilatação e Transformada da Erosão, o qual devolve o resultado adequado à escolha da operação;
2. a linearização dos dados de entrada, sendo representados por vetores os elementos estruturantes de \mathcal{L} e por listas as coordenadas de interesse que pertençam à caixa delimitadora de X ; e
3. a análise das coordenadas de interesse que se encontrem na borda calculada através da união dos elementos de \mathcal{L} .

Note que essas modificações já foram amplamente adotadas em diversos algoritmos relacionados, como podemos verificar em (Hirata, 1997) e (Zampirolli, 2003).

Conforme descrito no capítulo anterior, os parâmetros de entrada dos operadores de Transformada da Dilatação e Transformação da Erosão são:

- número de dimensões $d > 1$ para a definição das imagens;
- imagem binária de entrada $X \subset \mathbb{Z}^d$; e
- uma lista $\mathcal{L} = [L_0, L_1, \dots, L_k]$ de $k + 1$ elementos estruturantes.

A única restrição aplicada é o fato de que cada um dos elementos estruturantes L_i de \mathcal{L} sejam imagens binárias de \mathbb{Z}^d que possuam as suas respectivas origens, conforme a Equação 1.3. O elemento estruturante $L_0 = \{o\}$ é o elemento neutro das operações de Erosão e de Dilatação e é representado para fins do corretismo do algoritmo proposto.

Seja c uma coordenada válida em \mathbb{Z}^d . Os valores válidos para a imagem em níveis de cinza resultante para cada operador na coordenada c seguem as Equações 2.14 e 2.15 para os operadores de Transformada da Erosão e Transformada da Dilatação, respectivamente.

Conforme as duas seções posteriores do capítulo anterior, o Algoritmo 21 (TRANSFORMADAPROPOSTA) foi dividido em três etapas: (i) a inicialização de dados referente aos elementos estruturantes de \mathcal{L} (Algoritmo 10 – LINEARIZEELEMENTOESTRUTURANTE), (ii) a inicialização de dados referente à imagem binária

X de acordo com a união dos elementos estruturantes de \mathcal{L} (Algoritmo 22 – INICIALIZEMATRIZESPARGA), e (iii) o algoritmo propriamente dito (Algoritmo 24 – ALGORITMOPROPOSTO).

Algoritmo 19 TRANSFORMADADILATAÇÃOPROPOSTA (X, \mathcal{L}, k)

1 **devolva** TRANSFORMADAPROPOSTA ($X, \mathcal{L}, k, \text{falso}$)

Algoritmo 20 TRANSFORMADAEROSÃOPROPOSTA (X, \mathcal{L}, k)

1 **devolva** TRANSFORMADAPROPOSTA ($X, \mathcal{L}, k, \text{verdadeiro}$)

Algoritmo 21 TRANSFORMADAPROPOSTA($X, \mathcal{L}, k, \acute{e}$ TransformadaErosão)

1 *cardinalidadeUnião, elementosUnião, cardinalidadeL, elementosL* \leftarrow LINEARIZEELEMENTOESTRUTURANTE (\mathcal{L}, k)
 2 *listaNósBorda, matrizNósBorda, imagemCandidato, vetorListasUnião*
 \leftarrow INICIALIZEMATRIZESPARGA($X, \text{cardinalidadeUnião}, \text{elementosUnião}, \acute{e}$ TransformadaErosão)
 3 **devolva** ALGORITMOPROPOSTO($X, \text{listaNósBorda}, \text{matrizNósBorda}, \text{imagemCandidato}, \text{vetorListasUnião},$
 cardinalidadeUnião, elementosUnião, cardinalidadeL, elementosL, k, \acute{e}TransformadaErosão)

O Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE) é utilizado com o intuito de linearizar os dados referentes aos elementos estruturantes de \mathcal{L} da mesma forma que fora exemplificado nas duas seções posteriores do capítulo anterior.

Para o entendimento do restante do algoritmo proposto, apresentamos a matriz esparsa bidimensional. Para sua utilização, foram desenvolvidos dois tipos de nós: o *nó da borda* e o *nó de ligação*.

O *nó da borda* possui como objetivo representar uma coordenada de interesse que se encontre na borda para a próxima combinação de *iteração* e *índiceElementoEstruturante* a ser analisada no Algoritmo 24 (ALGORITMOPROPOSTO). Um *nó da borda* possui quatro campos: (i) as coordenadas de interesse que se encontram na caixa delimitadora de X , (ii) um ponteiro que indica o *nó anterior* em *listaNósBorda*, (iii) um ponteiro que indica o *próximo nó* em *listaNósBorda*, e (iv) um ponteiro que indica os *nós de ligação*. Convém notar que, ao compararmos com os algoritmos anteriores, *listaNósBorda* adotou a função que era designada a *listaCoordenadas*, com a diferença de que a unidade de dados de *listaNósBorda* é a estrutura *nós da borda* e essa nova lista também atua como uma lista duplamente ligada com cabeça, visando facilitar a remoção de um *nó da lista*.

O *nó de ligação* possui como objetivo relacionar uma dada coordenada de interesse a um elemento da união dos elementos estruturantes de \mathcal{L} . A utilização do dado representado por essa estrutura é o mecanismo que minimiza a necessidade de comparações para a determinação da borda, conforme explicado posteriormente. Um *nó de ligação* possui quatro campos: (i) um ponteiro que indica a qual *nó da borda* esse dado está relacionado, (ii) um ponteiro que indica a *próxima ligação* desse *nó da borda*, (iii) um ponteiro que indica o *nó anterior* da lista duplamente ligada, e (iv) um ponteiro que indica o *próximo nó* da lista duplamente ligada.

A Figura 4.1 apresenta um exemplo de como esses nós se relacionam. O intuito dessa figura é indicar como os campos descritos são utilizados. O item (a) representa a cabeça de *listaNósBorda*, enquanto que

os item (b) e (c) representam as cabeças das listas contidas nas posições 3 e 2 de *vetorListasUnião*, respectivamente. O item (d) representa uma coordenada de interesse c qualquer e os itens (e) e (f) representam os nós de ligação que vinculam c aos elementos da união dos elementos estruturantes de \mathcal{L} que são representados pelos índices 3 e 2 de *elementosUnião*, respectivamente. Convém apontar que a matriz esparsa bidimensional é calculada após a execução do Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE).

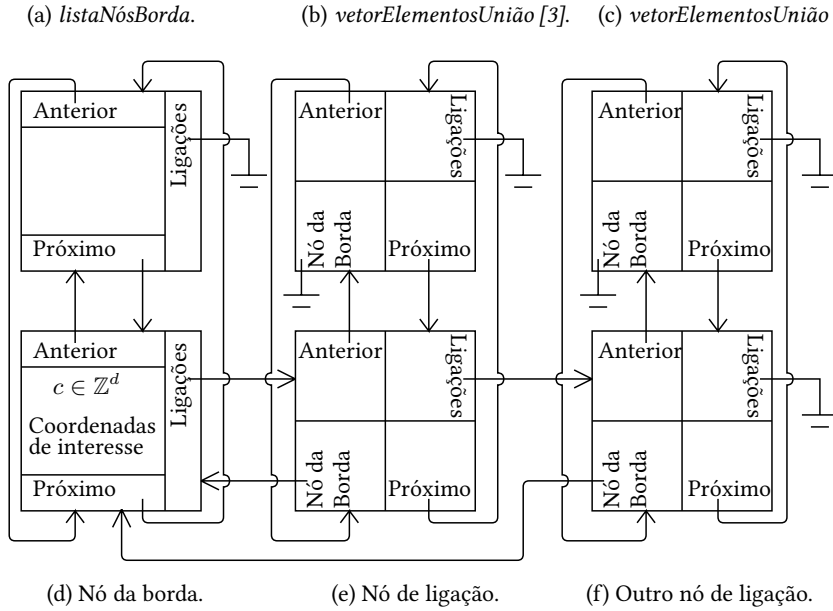


Figura 4.1: Apresentação dos tipos de nós utilizados: nó da borda e nó de ligação.

A matriz esparsa bidimensional é composta por três estruturas:

1. uma lista denominada *listaNósBorda*, o qual desempenha a mesma função designada por *listaCoordenadas* nos algoritmos anteriores, sendo tal função representar as coordenadas de interesse pertencentes à borda;
2. uma matriz de mesmo tamanho que a caixa delimitadora de X denominada *matrizNósBorda*, a qual existe para permitir o acesso direto aos nós da borda de acordo com o valor da coordenada de interesse, de modo a armazenar os mesmos nós de *listaNósBorda* nas respectivas posições; e
3. um vetor denominado *vetorListasUnião*, o qual possui *cardinalidadeUnião* posições, sendo que cada posição armazena uma cabeça de lista duplamente ligada de nós de ligação.

Para explicar a utilidade dessa matriz, tomemos como base o operador de Transformada da Erosão: seja c a coordenada de interesse representada por um nó da borda e seja δ uma coordenada que pertença à união dos elementos estruturantes de \mathcal{L} . A existência de um dado nó de ligação que vincule c à δ equivale ao fato de que a coordenada de interesse c poderá ser removido de *imagemTrabalho* ao final da iteração, pois no início da iteração, já vale a relação $c + \delta \notin \text{imagemTrabalho}$. Desse modo, o Algoritmo 24 (ALGORITMOPROPOSTO) é responsável somente por processar os nós de ligação que estejam na matriz, de acordo com as listas correspondentes aos elementos de um dado $L_{\text{índiceElementoEstruturante}}$.

Visando exemplificar uma inicialização da matriz esparsa bidimensional, apresentamos a Figura 4.2. Essa figura possui como premissa a utilização a mesma lista \mathcal{L} de 3 elementos estruturantes apresentada na Figura 3.1.

No Algoritmo 22 (INICIALIZEMATRIZESPARGA), inicialmente são alocadas as estruturas *listaNósBorda*, *matrizNósBorda*, *imagemCandidato* e *vetorListasUnião*. A estrutura *listaNósBorda* é representada por uma cabeça para armazenar a lista duplamente ligada de nós da borda, enquanto que *matrizNósBorda* é uma matriz vazia do mesmo tamanho que a caixa delimitadora de X , a qual permite o acesso direto dos nós da borda de acordo com as suas respectivas coordenadas de interesse. *imagemCandidato* é uma imagem para indicar quais coordenadas de interesse já foram inseridas em *listaNósBorda*. *vetorListasUnião* é inicializado com tamanho *cardinalidadeUnião*, o qual possui valor 3 nesse caso, e possui uma cabeça de lista duplamente ligada de nós de ligação em cada uma de suas posições.

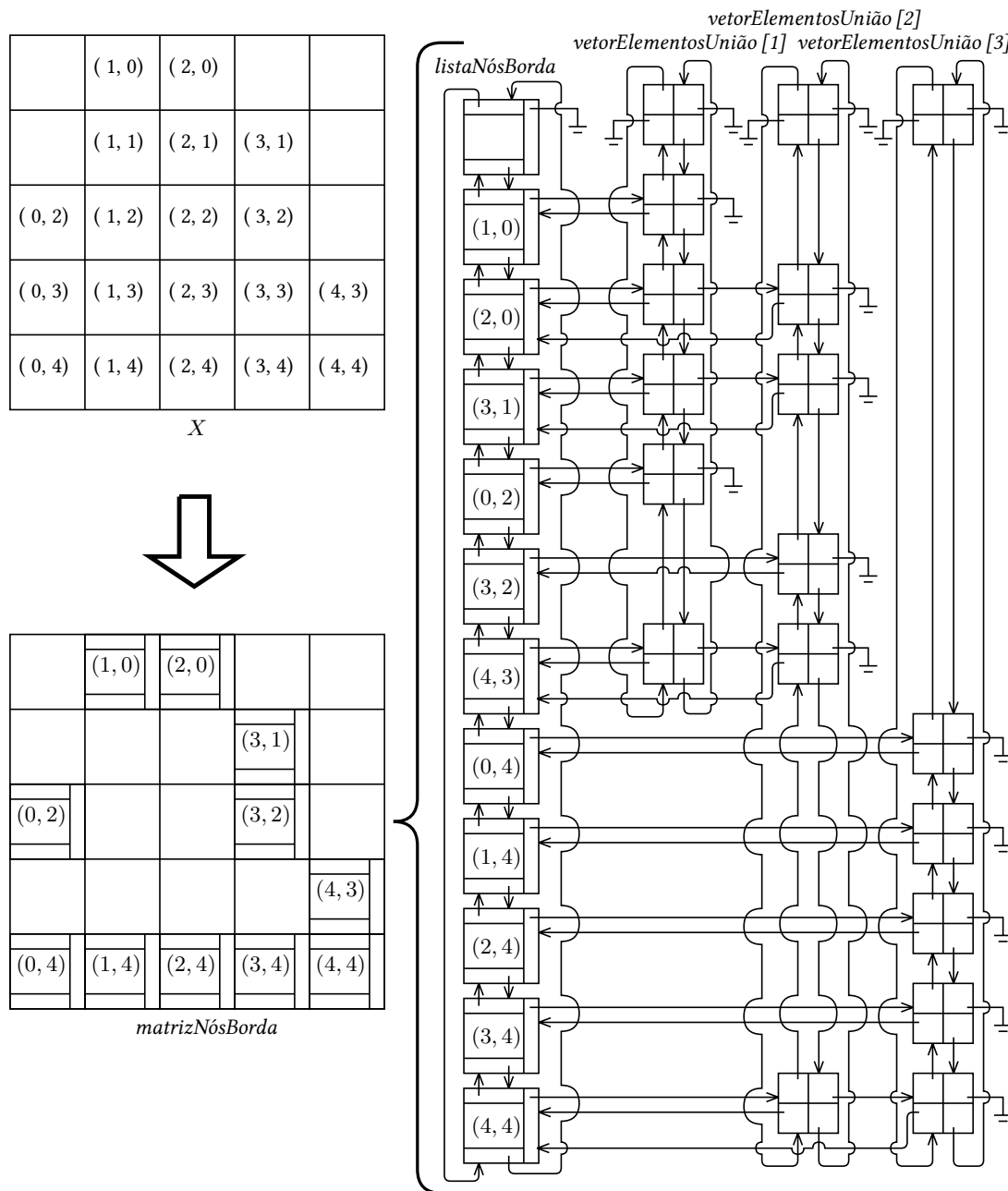


Figura 4.2: Exemplo visual da matriz esparsa bidimensional inicialmente calculada através de INICIALIZEMATRIZESPARGA para a operação de Transformada da Erosão (como continuação da Figura 3.1).

Após a inicialização dessas estruturas, todas as coordenadas da caixa delimitadora de X são verificadas. Caso alguma coordenada de interesse dentro dessa caixa delimitadora seja encontrada, é verificado

o fato dessa coordenada pertencer à borda. A principal diferença entre o algoritmo discutido e o Algoritmo 16 (LINEARIZEBORDAIMAGEM) consiste no fato de, além de inserir uma coordenada de interesse em *listaNósBorda*, categorizar o motivo da coordenada de interesse em questão pertencer à borda de acordo com um dado elemento da união dos elementos estruturantes de \mathcal{L} . Em paralelo aos vínculos efetuados através dos nós de ligação, as coordenadas de interesse são marcadas em *imagemCandidato*.

O lado direito da Figura 4.2 apresenta uma possível instância da matriz esparsa bidimensional para a imagem $X \subset \mathbb{Z}^2$ mostrada na mesma figura, após o processo de inicialização. Note que a instância da matriz esparsa bidimensional está atrelada à ordem de como as coordenadas da caixa delimitadora de X são verificadas.

Algoritmo 22 INICIALIZEMATRIZESPARSA(X , cardinalidadeUnião, elementosUnião, éTransformadaErosão)

```

1  inicialize vetorListasUnião como um vetor de cardinalidadeUnião posições de modo que
    cada posição armazene uma cabeça de lista duplamente encadeada de nós de ligação
2  inicialize ligaçãoEncontrada com o valor falso
3  inicialize listaNósBorda como uma cabeça de lista duplamente encadeada de nós da borda
4  inicialize matrizNósBorda como uma matriz vazia de nós da borda de mesmo tamanho que
    a caixa delimitadora de  $X$  vinculada a uma função de mapeamento entre as coordenadas dessa
    caixa e as posições da matriz
5  inicialize imagemCandidato como uma imagem binária vazia de mesmo tamanho que
    a caixa delimitadora de  $X$ 
    ▷ Efetua a inicialização da invariante adotada para TRANSFORMADAPROPOSTA.
6  para toda coordenada  $c$  na caixa delimitadora de  $X$  faça
7    se  $((c \in X) = \text{éTransformadaErosão})$  então
8      para índicePosiçãoUnião de 1 até cardinalidadeUnião faça
9        ligaçãoEncontrada  $\leftarrow$  falso
10       delta  $\leftarrow$  elementosUnião[índicePosiçãoUnião]
11       se éTransformadaErosão então
12         ligaçãoEncontrada  $\leftarrow$   $(c + \textit{delta}) \notin X$ 
13       senão
14         ligaçãoEncontrada  $\leftarrow$   $(c - \textit{delta}) \notin X^c$ 
15       se ligaçãoEncontrada então
16         ▷ A coordenada de interesse  $c$  será vinculada a delta através de um novo nó de ligação.
17         se a posição de  $c$  em matrizNósBorda estiver vazia então
18           imagemCandidato  $\leftarrow$  imagemCandidato  $\cup$   $\{c\}$ 
19           inicialize um nó da borda relacionado a  $c$  e o insira em matrizNósBorda e listaNósBorda
20           inicialize nóBorda como o nó da borda da posição de  $c$  em matrizNósBorda
21           inicialize um nó de ligação que vincule delta a  $c$  e o insira em nóBorda e
22           na lista da posição índicePosiçãoUnião de vetorListasUnião
23 devolva listaNósBorda, matrizNósBorda, imagemCandidato, vetorListasUnião

```

O Algoritmo 23 (ATUALIZAÇÃOALGORITMOPROPOSTO) é responsável por uma parte significativa da manutenção do algoritmo proposto, representado pelo Algoritmo 24 (ALGORITMOPROPOSTO), pois armazena os resultados obtidos pela operação através das instruções contidas entre as linhas 1 e 6 e efetua a inserção de novos nós da borda e nós de ligação através das instruções contidas entre as linhas 7 e 26.

Algoritmo 23 ATUALIZAÇÃOALGORITMOPROPOSTO(*imagemTrabalho*, *listaNósBorda*, *matrizNósBorda*, *imagemCandidato*, *vetorListasUnião*, *aAtualizar*, *cardinalidadeUnião*, *elementosUnião*, *éTransformadaErosão*, *iteração*, *resultado*)

▷ Altera *imagemTrabalho* e *resultado* de acordo com a borda.

- 1 para toda coordenada de interesse c em *aAtualizar* faça
- 2 se *éTransformadaErosão* então
- 3 $imagemTrabalho \leftarrow imagemTrabalho \setminus \{c\}$
- 4 senão
- 5 $imagemTrabalho^c \leftarrow imagemTrabalho^c \setminus \{c\}$
- 6 o valor de *resultado* na coordenada c recebe *iteração*
- ▷ Efetua manutenção da invariante adotada para TRANSFORMADAPROPOSTA.
- 7 para toda coordenada de interesse c em *aAtualizar* faça
- 8 para *índicePosiçãoUnião* de 1 a *cardinalidadeUnião* faça
- 9 ▷ É necessário verificar cada c com cada *delta* para a determinação dos novos nós de ligação (manutenção).
- 10 *ligaçãoEncontrada* \leftarrow **falso**
- 11 *delta* $\leftarrow elementosUnião[índicePosiçãoUnião]$
- 12 se *éTransformadaErosão* então
- 13 *possívelCandidato* $\leftarrow (c - delta)$
- 14 senão
- 15 *possívelCandidato* $\leftarrow (c + delta)$
- 16 se *possívelCandidato* está dentro da caixa delimitadora de *imagemTrabalho* então
- 17 se *possívelCandidato* $\notin imagemCandidato$ então
- 18 se *éTransformadaErosão* então
- 19 se *possívelCandidato* $\in imagemTrabalho$ então *ligaçãoEncontrada* \leftarrow **verdadeiro**
- 20 senão
- 21 se *possívelCandidato* $\in imagemTrabalho^c$ então *ligaçãoEncontrada* \leftarrow **verdadeiro**
- 22 se *ligaçãoEncontrada* então
- 23 ▷ A coordenada de interesse *possívelCandidato* será vinculada a *delta*, após a atualização referente a c , através da criação de um nó de ligação.
- 24 se a posição de *possívelCandidato* em *matrizNósBorda* estiver vazia então
- 25 $imagemCandidato \leftarrow imagemCandidato \cup \{possívelCandidato\}$
- 26 inicialize um nó da borda relacionado a *possívelCandidato* e o insira em *matrizNósBorda* e *listaNósBorda*
- 27 inicialize *nóBorda* como o nó da borda da posição de *possívelCandidato* em *matrizNósBorda*
- 28 inicialize um nó de ligação que vincule *delta* a *possívelCandidato* e o insira em *nóBorda* e na lista da posição *índicePosiçãoUnião* de *vetorListasUnião*
- 29 **devolva** *imagemTrabalho*, *listaNósBorda*, *matrizNósBorda*, *imagemCandidato*, *vetorListasUnião*, *resultado*

O Algoritmo 24 (ALGORITMOPROPOSTO) calcula a operação desejada através do consumo dos nós de ligação relacionados aos elementos de um dado L_x , inserindo as coordenadas de interesse em questão na lista *aAtualizar*, como podemos verificar nas instruções contidas nas linhas 8 a 24. Conforme descrito anteriormente, a execução da linha 25 finaliza o processo de manutenção do algoritmo proposto. Existem dois critérios que compõem a condição de parada, a qual é verificada na linha 6: (i) *listaNósBorda* não possui nenhuma coordenada de interesse, ou (ii) o número de elementos estruturantes de \mathcal{L} verificados na iteração imediatamente anterior que não alteraram *imagemTrabalho* equivale ao número k , i.e., não houveram mudanças em *imagemTrabalho* durante toda a iteração de valor $iteração = m + 1$.

Algoritmo 24 ALGORITMOPROPOSTO(X , listaNósBorda, matrizNósBorda, imagemCandidato, vetorListasUnião, cardinalidadeUnião, elementosUnião, cardinalidadeL, elementosL, k , éTransformadaErosão)

```

1  inicialize resultado como uma imagem de valores inteiros de mesmo tamanho que a caixa delimitadora de  $X$ 
  de modo que o valor  $-1$  seja atrelado a cada uma das coordenadas que estejam dentro dessa caixa
2  inicialize imagemTrabalho como cópia da imagem  $X$ 
3  inicialize índiceElementoEstruturante, índiceElemento, iteração e  $x$  com o valor 0
4  para todo elemento de coordenadas  $c \in$  imagemTrabalho faça
5    o valor de resultado na coordenada  $c$  recebe iteração
6  enquanto listaNósBorda não contiver somente sua cabeça faça
7    iteração  $\leftarrow$  iteração + 1
8    para índiceElementoEstruturante de 1 até  $k$  faça
9      inicialize vetorAtual como elementosL[índiceElementoEstruturante]
10      $x \leftarrow x + 1$ 
11     inicialize aAtualizar como uma lista vazia de coordenadas em  $\mathbb{Z}^d$ 
    ▷ Determina a borda de imagemTrabalho.
12     para índiceElemento de 1 até cardinalidadeL[índiceElementoEstruturante] faça
13       índicePosiçãoUnião  $\leftarrow$  vetorAtual[índiceElemento]
14       enquanto vetorListasUnião[índicePosiçãoUnião] não contiver somente sua cabeça faça
15         desenfileire um nó de ligação de vetorListasUnião[índicePosiçãoUnião]
          como primeiro
16         inicialize nBorda como o nó da borda de primeiro
17         inicialize  $c$  como a coordenada de nóBorda
18         remova nóBorda de listaNósBorda
19         remova nóBorda de matrizNósBorda
20         inicialize próximaLigação como o nó de ligação apontado por nóBorda
21         enquanto próximaLigação não apontar para um nó inválido faça
22           remova próximaLigação de sua lista duplamente encadeada
23           atribua a próximaLigação o nó de ligação apontado por próximaLigação
24           insira  $c$  em aAtualizar
    ▷ Altera imagemTrabalho e resultado de acordo com a borda.
25     imagemTrabalho, listaNósBorda, matrizNósBorda, imagemCandidato, vetorListasUnião,
      resultado  $\leftarrow$  ATUALIZAÇÃOALGORITMOPROPOSTO(imagemTrabalho, listaNósBorda,
        matrizNósBorda, imagemCandidato, vetorListasUnião, aAtualizar,
          cardinalidadeUnião, elementosUnião, éTransformadaErosão, iteração, resultado)
26  devolva resultado

```

Observe que a proposta inicial do algoritmo, a qual consiste em minimizar o número de comparações para a determinação da borda, foi efetuada entre as linhas 14 e 20. Utilizando a invariante proposta para esse algoritmo, a determinação da borda da imagem de entrada I_{x-1} é calculada através da remoção das filas duplamente ligadas de nós de ligação, cujas cabeças estão armazenadas em *vetorListasUnião*.

Capítulo 5

Corretismo do Algoritmo Proposto

O Algoritmo 21 (TRANSFORMADAPROPOSTA) foi projetado com a Equação 2.6 como base, visando atender tanto a Transformada da Erosão quanto a Transformada da Dilatação, assim como os algoritmos duais previamente apresentados.

Este capítulo foi dividido de modo a inicialmente apresentar um conjunto de provas utilizados nas seções referentes a ambas as Transformadas. Após tais demonstrações, são apresentados os corretismos referentes aos operadores de transformada da Erosão e Transformada da Dilatação.

5.1 Demonstrações e Definições Comuns aos Dois Operadores

Proposição 5.1. *Sejam G e H duas imagens binárias em \mathbb{Z}^d de modo que $H \subseteq G$. Então,*

$$H = G \setminus (G \setminus H) .$$

Demonstração.

$$\begin{aligned} G \setminus (G \setminus H) &= \{z \in G : z \notin (G \setminus H)\} && \text{(definição de diferença de conjuntos)} \\ &= \{z \in G : z \notin (G \cap H^c)\} && \text{(propriedade da diferença de conjuntos)} \\ &= \{z \in G : z \in (G \cap H^c)^c\} && \text{(propriedade da complementação)} \\ &= \{z \in G : z \in (G^c \cup H)\} && \text{(Lei de De Morgan)} \\ &= \{z \in G : z \in H\} && (z \text{ deve pertencer a } G) \\ &= \{z \in H\} && (H \subseteq G) \\ &= H . \end{aligned}$$

□

O Corolário 5.2 segue-se como consequência direta da Proposição 5.1.

Corolário 5.2. *Seja P um elemento estruturante que contenha a origem. Então, para qualquer imagem binária $A \subset \mathbb{Z}^d$, temos que*

$$\varepsilon_P(A) = A \setminus (A \setminus \varepsilon_P(A)) .$$

Demonstração. Dado que $o \in P$, a seguinte sentença é verdadeira: $\varepsilon_P(A) \subseteq A$. Portanto, pela Proposição 5.1, $\varepsilon_P(A) = A \setminus (A \setminus \varepsilon_P(A))$. □

Para as provas posteriores, são requeridas as seguintes definições:

- Inteiro $x \geq 0$
- Elemento estruturante binário L_x
- Imagem binária I_x
- Conjunto $U = \bigcup_{i=1}^k L_i$
- Lista \mathcal{F}_{x+1}^u
- Imagem $D_{x+1} = \bigcup_{u \in U} \mathcal{F}_{x+1}^u$

Tais definições são relacionadas ao corretismo do Algoritmo 21 (TRANSFORMADAPROPOSTA). O valor inteiro não-negativo x é utilizado como índice para representar uma dada combinação de *iteração* e *índiceElementoEstruturante*, conforme sua definição na Equação 2.17, exceto na inicialização, caso em que

$$x = \textit{iteração} = 0 \ .$$

O elemento estruturante L_x foi definido na Equação 2.18. De modo equivalente, a imagem I_x foi definida na Tabela 2.5 e no algoritmo é representado pela variável *imagemTrabalho*.

A lista \mathcal{F}_{x+1}^u está atrelada a um elemento u pertencente à união U dos elementos estruturantes de $\mathcal{L} \left(U = \bigcup_{i=1}^k L_i \right)$ e está relacionada à invariante do algoritmo proposto. No caso da Transformada da Erosão, a lista armazena os dados utilizados para o cálculo da Erosão posterior à x -ésima Erosão sequencial efetuada pelo algoritmo. De modo análogo, no caso da Transformada da dilatação, a lista armazena os dados utilizados para o cálculo da Dilatação posterior à x -ésima Dilatação sequencial efetuada pelo algoritmo. Tal lista é representada no algoritmo através da posição relativa a u em *vetorListasUnião*. A definição formal, de acordo com a operação desejada, é apresentada através da Tabela 5.1.

Tabela 5.1: Definição de \mathcal{F}_{x+1}^u de acordo com a operação desejada.

Operação	Definição da Lista \mathcal{F}_{x+1}^u
Transformada da Erosão	$\mathcal{F}_{x+1}^u = \{p \in I_x : p + u \notin I_x\}$
Transformada da Dilatação	$\mathcal{F}_{x+1}^u = \{p \in I_x : p - u \notin I_x\}$

O conjunto D_{x+1} é a união das listas, de modo que é representado por três variáveis no algoritmo: *matrizNósBorda*, a qual permite o acesso direto de seus dados através do uso de coordenadas; *listaNósBorda*, a qual é uma versão linearizada do conjunto, fato que permite analisar em tempo $O(1)$ se o conjunto está vazio; e *vetorListasUnião*, o qual armazena as listas \mathcal{F}_{x+1}^u em cada uma das suas posições referentes a $u \in U$.

Como essas definições se relacionam? De modo a adiantar parte dos resultados obtidos pelas provas demonstradas neste capítulo, a borda da imagem I_x de acordo com o elemento estruturante L_{x+1} e com a operação desejada, representada formalmente por $\textit{borda}_{L_{x+1}}(I_x)$, está armazenada em D_{x+1} , ou seja,

$$\textit{borda}_{L_{x+1}}(I_x) \subseteq D_{x+1} \ .$$

Conforme prova posterior, podemos calcular a imagem resultante posterior à combinação representada por x através da relação

$$I_{x+1} = I_x \setminus \text{borda}_{L_{x+1}}(I_x)$$

de modo a consumir somente as listas \mathcal{F}_{x+1}^u em que $u \in L_{x+1}$.

5.2 Algoritmo de Transformada da Erosão

A prova de corretismo para o algoritmo proposto está dividida em duas partes: (i) avaliamos como a distribuição dos elementos contidos na união dos elementos estruturantes de \mathcal{L} em listas individuais nos auxilia a efetuar a operação de Erosão, e (ii) os invariantes do algoritmo são apresentados e provados de modo a calcular corretamente a saída.

Teorema 5.3. *Seja A uma imagem binária em \mathbb{Z}^d . Seja P um elemento estruturante de modo que a origem $o \in P$. A seguinte relação é válida:*

$$\underbrace{\bigcup_{\Delta \in P} (A \setminus \varepsilon_{\{\Delta, o\}}(A))}_R = \underbrace{A \setminus \varepsilon_P(A)}_S . \quad (5.1)$$

Demonstração. Para provar a relação, são necessários dois passos:

1. $R \supseteq S : a \in S \Rightarrow a \in R$

$$\begin{aligned} a \in S &\xrightarrow{\text{definição de } S} a \in A \setminus \varepsilon_P(A) \\ &\xrightarrow{\text{definição de diferença de conjuntos}} a \in A \text{ e } a \notin \varepsilon_P(A) \\ &\xrightarrow{\text{definição de Erosão}} a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : \forall p \in P, p+h \in A \right\} \\ &\xrightarrow{\text{generalização existencial}} a \in A \text{ e } \exists p \in P : p+a \notin A \\ &\xrightarrow{\text{conjunção}} \exists p \in P : a \in A \text{ e } p+a \notin A \\ &\xrightarrow{\text{reinterpretação}} \exists p \in P : a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : p+h \in A \right\} \\ &\xrightarrow{x \text{ deve pertencer a } A} \exists p \in P : a \in A \text{ e } a \notin \left(\left\{ h \in \mathbb{Z}^d : p+h \in A \right\} \cap A \right) \\ &\xrightarrow{\text{definição de } A} \exists p \in P : a \in A \text{ e } a \notin \left(\left\{ h \in \mathbb{Z}^d : p+h \in A \right\} \cap \left\{ h \in \mathbb{Z}^d : h \in A \right\} \right) \\ &\xrightarrow{\text{definição de interseção}} \exists p \in P : a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : p+h \in A \text{ e } h \in A \right\} \\ &\xrightarrow{\text{definição de conjunto}} \exists p \in P : a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : \{p+h, h\} \subseteq A \right\} \\ &\xrightarrow{\text{definição de translação}} \exists p \in P : a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : h + \{p, o\} \subseteq A \right\} \\ &\xrightarrow{\text{definição de Erosão}} \exists p \in P : a \in A \text{ e } a \notin \varepsilon_{\{p, o\}}(A) \\ &\xrightarrow{\text{definição de diferença de conjuntos}} \exists p \in P : a \in A \setminus \varepsilon_{\{p, o\}}(A) \\ &\xrightarrow{\text{definição de união}} a \in \bigcup_{p \in P} (A \setminus \varepsilon_{\{p, o\}}(A)) \\ &\xrightarrow{\text{definição de } R} a \in R . \end{aligned}$$

2. $R \subseteq S : a \in R \Rightarrow a \in S$

$$\begin{aligned}
a \in R &\xrightarrow{\text{definição de } R} \exists \Delta \in P : a \in A \setminus \varepsilon_{\{\Delta, o\}}(A) \\
&\xrightarrow{\text{definição de diferença de conjuntos}} \exists \Delta \in P : a \in A \text{ e } a \notin \varepsilon_{\{\Delta, o\}}(A) \\
&\xrightarrow{\text{definição de Erosão}} \exists \Delta \in P : a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : \{\Delta + h, h\} \subseteq A \right\} \\
&\xrightarrow{\text{definição de conjunto}} \exists \Delta \in P : a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : \Delta + h \in A \text{ e } h \in A \right\} \\
&\xrightarrow{\text{restrição de domínio}} \exists \Delta \in P : a \in A \text{ e } a \notin \{h \in A : \Delta + h \in A\} \\
&\xrightarrow{\text{substituição direta}} \exists \Delta \in P : a \in A \text{ e } \Delta + a \notin A \\
&\xrightarrow{\text{generalização universal}} a \in A \text{ e } a \notin \left\{ h \in \mathbb{Z}^d : \forall p \in P, p + h \in A \right\} \\
&\xrightarrow{\text{definição de Erosão}} a \in A \text{ e } a \notin \varepsilon_P(A) \\
&\xrightarrow{\text{definição de diferença de conjuntos}} a \in A \setminus \varepsilon_P(A) \\
&\xrightarrow{\text{definição de } S} a \in S .
\end{aligned}$$

□

Corolário 5.4. Para $x > 0$, as listas individuais para cada um dos elementos contidos na união dos elementos estruturantes de \mathcal{L} são utilizadas no Algoritmo 21 (TRANSFORMADAPROPOSTA) para o cálculo de

$$\varepsilon_{L_x}(I_{x-1}) .$$

Demonstração. Para uma dada imagem binária A e um elemento estruturante P , o qual possui sua origem, o Teorema 5.3 prova que é possível utilizar a união de bordas de A de acordo com cada elemento de P $\left(\bigcup_{\Delta \in P} (A \setminus \varepsilon_{\{\Delta, o\}}(A)) \right)$ para determinar a borda de A de acordo com o próprio P ($A \setminus \varepsilon_P(A)$).

Sejam a imagem I_{x-1} , o elemento estruturante L_x e o conjunto U de união dos elementos de \mathcal{L} definidos conforme a Seção 5.1. O Algoritmo 21 (TRANSFORMADAPROPOSTA) utiliza o Teorema 5.3, de modo que a imagem I_{x-1} e a união U substituem respectivamente A e P . A variável *vetorListasUnião*, calculada no algoritmo, armazena em cada uma de suas posições uma lista com os elementos que compõe a borda da imagem I_{x-1} de acordo com uma coordenada de U através dos nós de ligação, ou seja,

$$I_{x-1} \setminus \varepsilon_{\{\text{elementosUnião}[i], o\}}(I_{x-1}) = \text{vetorListasUnião}[i] = \mathcal{F}_x^{\text{elementosUnião}[i]} \vee \text{elementosUnião}[i] \in U .$$

Pela definição de U e L_x , temos que L_x é um subconjunto de U , ou seja, $L_x \subseteq U$. Tal fato em conjunto com o Corolário 5.2 e o Teorema 5.3, resulta no cálculo da erosão de I_{x-1} pelo elemento estruturante L_x como

$$\varepsilon_{L_x}(I_{x-1}) = I_{x-1} \setminus \bigcup_{\text{elementosUnião}[i] \in L_x} \mathcal{F}_x^{\text{elementosUnião}[i]} .$$

Dessa forma, as listas individuais para cada um dos elementos contidos na união dos elementos estruturantes de \mathcal{L} são utilizadas no Algoritmo 21 (TRANSFORMADAPROPOSTA) para o cálculo de $\varepsilon_{L_x}(I_{x-1})$. □

Podemos verificar que o Corolário 5.4 soluciona o item i apresentado no começo da seção. Ao longo do restante da seção, algumas propriedades são apresentadas para demonstrar a questão abordada pelo

item **ii** do começo da seção. A seguinte proposição apresenta as invariantes utilizadas no algoritmo.

Lema 5.5. *Sejam $x \geq 0$ um valor inteiro e B_x o conjunto composto por todos os elementos de I_{x-1} que estejam nas listas \mathcal{F}_x^u , de modo que u seja um elemento de L_x , ou seja,*

$$B_x = \{p \in I_{x-1} : p \in \mathcal{F}_x^u, u \in L_x\} .$$

No começo do loop determinado pelas instruções contidas da linha 8 à 25 do Algoritmo 24 (ALGORITMOPROPOSTO), afirma-se o seguinte como suas invariantes:

(A) a imagem de entrada para o valor x equivale a

$$\text{imagemTrabalho} = I_{x-1} = \varepsilon_{L_{x-1}}(\dots(\varepsilon_{L_0}(X))\dots) ;$$

(B) a dimensão da matriz esparsa bidimensional relacionada às coordenadas de U é composta por cabeças de listas duplamente ligadas, de modo que a composição dessas listas é definida como

$$\text{vetorListasUnião}[i] = \mathcal{F}_x^{\text{elementosUnião}[i]} \forall \text{elementosUnião}[i] \in U ;$$

(C) a dimensão da matriz esparsa bidimensional relacionado às coordenadas de interesse equivale ao conjunto de elementos determinado pela relação

$$\text{listaNósBorda} = \text{matrizNósBorda} = D_x = \bigcup_{\text{elementosUnião}[i] \in U} \mathcal{F}_x^{\text{elementosUnião}[i]} = I_{x-1} \setminus \varepsilon_U(I_{x-1}) ;$$

(D) a imagem binária representada pela variável *imagemCandidato* equivale ao conjunto de elementos que já foram inseridos em *listaNósBorda*, ou seja,

$$\text{imagemCandidato} = \bigcup_{w=1}^x D_w ; e$$

(E) a imagem em níveis de cinza resultado armazena corretamente os valores obtidos através das bordas B_1 a B_{x-1} para a Transformada da Erosão, conforme a relação

$$\text{TRANSFORMADAEROSÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \max\{w \mid c \in \varepsilon_{wL_k}(\dots(\varepsilon_{wL_2}(\varepsilon_{wL_1}(X)))\dots)\} & : c \in \bigcup_{y=1}^{x-1} B_y \\ 0 & : c \in X \setminus \bigcup_{y=1}^{x-1} B_y \\ -1 & : c \notin X \end{cases}$$

Demonstração. Deve-se demonstrar que todas as invariantes são inicializadas, mantidas e finalizadas corretamente.

Inicialização: A etapa de inicialização prepara as estruturas para a primeira execução do *loop*. Durante esse preparo, vale a relação

$$x = \text{iteração} = 0 .$$

Os dados da matriz esparsa bidimensional são inicializados pelo Algoritmo 22

(INICIALIZEMATRIZESPARGA), de modo que seu resultado é utilizado como parâmetro de entrada do

Algoritmo 24 (ALGORITMOPROPOSTO). A inicialização das invariantes **B**, **C** e **D** é demonstrada através do Teorema 5.6. Para o cálculo da Equação 2.14, utilizaremos duas imagens: *imagemTrabalho* e *resultado*. A inicialização de *imagemTrabalho* é efetuada na linha 2 de ALGORITMOPROPOSTO e seu valor inicial é

$$imagemTrabalho = I_0 = \varepsilon_{L_0}(X) = X ,$$

trivialmente validando a invariante **A**. Já a imagem *resultado* é inicializada na linha 1 de ALGORITMOPROPOSTO com o valor -1 em todas as coordenadas de sua caixa delimitadora e as instruções contidas nas linhas 4 e 5 atribuem o valor 0 às coordenadas $c \in X$. Essas instruções estão trivialmente corretas, uma vez que, para $x = 0$, a invariante **E** referente aos valores de *resultado* vale:

$$TRANSFORMADAEROSÃO(X, \mathcal{L}, k)[c] = \begin{cases} 0 & : c \in X \\ -1 & : c \notin X \end{cases} .$$

Manutenção: O conjunto B_x definido no começo do enunciado é utilizado pela etapa de manutenção de todas as invariantes. O Lema 5.7 prova que B_x é o gradiente morfológico interno de I_{x-1} de acordo com L_x enquanto que o Lema 5.8 apresenta como B_x é utilizado para calcular $\varepsilon_{L_x}(I_{x-1})$. Durante o *loop* descrito no enunciado, B_x é mapeado no algoritmo como a lista *aAtualizar*, a qual é calculada como o resultado das instruções contidas no intervalo da linha 9 à 24. As instruções da linha 12 à 24 percorrem as listas contidas em *vetorListasUnião* de modo que

$$aAtualizar = \bigcup_{u \in L_x} \mathcal{F}_x^u = B_x .$$

No início da etapa de manutenção, conforme a hipótese de indução, *listaNósBorda* e *matrizNósBorda* contêm os elementos de D_x , através dos nós da borda. Em paralelo à inserção das coordenadas de interesse c em *aAtualizar*, conforme definido na linha 17, as linhas 18 e 19 removem o nó da borda referente a c , respectivamente, de *listaNósBorda* e de *matrizNósBorda*, enquanto que as linhas 20 a 23 removem todos os nós de ligação referentes a c das listas cujas cabeças estão contidas em *vetorListasUnião*.

Após as instruções contidas no intervalo da linha 9 à 24, as seguintes relações são válidas:

$$listaNósBorda = matrizNósBorda = D_x \setminus B_x \quad (5.2)$$

$$vetorListasUnião[índicePosiçãoUnião] = \begin{cases} \emptyset & : elementosUnião[índicePosiçãoUnião] \in L_x \\ \mathcal{F}_x^{elementosUnião[índicePosiçãoUnião]} & \\ & : elementosUnião[índicePosiçãoUnião] \notin L_x \end{cases} \quad (5.3)$$

O Teorema 5.10 demonstra que são efetuados o registro correto dos valores relacionados aos elementos de B_x em *imagemTrabalho* e *resultado*; a inserção de novos nós da borda em *listaNósBorda* e *matrizNósBorda* com registro em *imagemCandidato*; e a inserção de novos nós de ligação em *vetorListasUnião*, de modo a finalizar a manutenção das invariantes propostas através da instrução contida na linha 25.

Término: O término do Algoritmo 24 (ALGORITMOPROPOSTO) é determinado trivial e exclusivamente através da instrução contida na linha 6. Essa instrução verifica se D_x , representado por *listaNósBorda* e *matrizNósBorda*, não possui mais coordenadas de interesse. Caso não exista mais coordenadas de interesse a serem analisadas, a imagem em níveis de cinza *resultado* é entregue como resultado do algoritmo. □

Para a demonstração do Lema 5.5, são utilizados diversos teoremas e lemas. O Teorema 5.6 efetua uma demonstração relacionadas à etapa de inicialização das invariantes B a D.

Teorema 5.6. *O Algoritmo 22 (INICIALIZEMATRIZESPARGA) inicializa corretamente as invariantes B, C e D do Lema 5.5, de modo que ao término de sua execução valem as seguintes relações:*

$$\begin{aligned} \text{vetorListasUnião}[i] &= \mathcal{F}_1^{\text{elementosUnião}[i]} \quad \forall \text{elementosUnião}[i] \in U ; \\ \text{listaNósBorda} = \text{matrizNósBorda} = D_1 &= \bigcup_{\text{elementosUnião}[i] \in U} \mathcal{F}_1^{\text{elementosUnião}[i]} = I_0 \setminus \varepsilon_U(I_0) ; e \\ \text{imagemCandidato} &= D_1 . \end{aligned}$$

Demonstração. Os parâmetros de entrada *cardinalidadeUnião* e *elementosUnião* de INICIALIZEMATRIZESPARGA são definidos como os valores de saída homônimos calculados pelo Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE). Para essa etapa de inicialização, temos que

$$x = \text{iteração} = 0 .$$

Como o próprio nome do algoritmo indica, INICIALIZEMATRIZESPARGA possui como objetivo preencher corretamente a matriz esparsa, de modo a preencher *listaNósBorda* e *matrizNósBorda* com nós da borda; inserir os nós de ligação pertinentes em cada uma das listas de *vetorListasUnião*; e registrar quais coordenadas já foram inseridas em *listaNósBorda* através da imagem *imagemCandidato*.

As instruções das linhas 6 e 7 percorrem a caixa delimitadora de X de modo a verificar quais são as coordenadas de interesse da imagem, sendo nesse caso as próprias coordenadas dos elementos de X . A coordenada c é definida pela instrução da linha 6 e no caso em que ela seja classificada como uma coordenada de interesse, as instruções contidas entre a linha 8 e 15 serão executadas. Esse trecho efetua *cardinalidadeUnião* verificações: uma verificação para cada elemento u da união U de elementos estruturantes de \mathcal{L} , de modo que *ligaçãoEncontrada* recebe **verdadeiro** no caso em que $c + u \notin X$.

No caso em que *ligaçãoEncontrada* seja validada pela instrução da linha 15, as instruções contidas entre a linha 16 e 20 efetuam as inserções relacionadas às invariantes citadas no enunciado.

A linha 16 verifica se será efetuada a inserção do primeiro nó de ligação: em caso positivo, a linha 17 atualiza *imagemCandidato* de modo a registrar as inserções do nó de borda relacionado a c em *listasNósBorda* e em *matrizNósBorda*. A instanciação do nó de borda e as inserções ocorrem na linha seguinte (18). Desse modo, *listasNósBorda*, *matrizNósBorda* e *imagemCandidato* são inicializados exclusivamente por dados relacionados aos elementos de D_1 , através da validação da condição exigida $\left(c \in \bigcup_{u \in U} \mathcal{F}_1^u \right)$.

A linha 19 recupera o nó da borda relacionado à coordenada c enquanto a linha 20 é responsável por instanciar o nó de ligação que relaciona a coordenada de interesse c à coordenada u da união U , ao ligar

o nó da borda com a lista em *vetorListasUnião* referente a u . Desse modo, *vetorListasUnião* é inicializado conforme a relação $\text{vetorListasUnião}[i] = \mathcal{F}_1^{\text{elementosUnião}[i]} \forall \text{elementosUnião}[i] \in U$. \square

Os Lemas 5.7 ao 5.9 e o Teorema 5.10 demonstram propriedades relacionadas à manutenção das invariantes descritas no enunciado do Lema 5.5.

Lema 5.7. *Sejam B_x , I_{x-1} e L_x definidos conforme o enunciado do Lema 5.5. B_x é o gradiente morfológico interno de I_{x-1} de acordo com L_x .*

Demonstração.

$$\begin{aligned} a \in B_x &\xleftrightarrow{\text{definição de } B_x} a \in I_{x-1} \text{ e } \exists u \in L_x : a \in \mathcal{F}_x^u \\ &\xleftrightarrow{\text{definição de } \mathcal{F}_x^u} a \in I_{x-1} \text{ e } \exists u \in L_x : a + u \notin I_{x-1} \\ &\xleftrightarrow{\text{definição de gradiente morfológico interno}} a \in I_{x-1} \setminus \varepsilon_{L_x}(I_{x-1}) . \end{aligned}$$

\square

Lema 5.8. *Sejam B_x , I_{x-1} e L_x definidos conforme o enunciado do Lema 5.5 e seja Y_x uma imagem tal que*

$$Y_x = I_{x-1} \setminus B_x .$$

Demonstraremos que a imagem Y_x equivale à imagem I_x .

Demonstração.

$$\begin{aligned} Y_x = I_{x-1} \setminus B_x &\xleftrightarrow{\text{definição de } B_x} Y_x = I_{x-1} \setminus (I_{x-1} \setminus \varepsilon_{L_x}(I_{x-1})) \\ &\xleftrightarrow{\text{Corolário 5.2}} Y_x = \varepsilon_{L_x}(I_{x-1}) \\ &\xleftrightarrow{\text{hipótese de indução}} Y_x = \varepsilon_{L_x}(\varepsilon_{L_{x-1}}(\dots(\varepsilon_{L_0}(X))\dots)) \\ &\xleftrightarrow{\text{definição de } I_x} Y_x = I_x . \end{aligned}$$

\square

Lema 5.9. *Considere o loop determinado pelas instruções contidas da linha 8 à 25 do Algoritmo 24 (ALGORITMOPROPOSTO), conforme o Lema 5.5, e seja resultado definido conforme a invariante E do mesmo lema. Demonstraremos que uma dada coordenada $c \in \mathbb{Z}^d$ será marcada não mais que uma vez em resultado durante o loop em questão.*

Demonstração. Para a demonstração, apresentamos três fatos:

1. Por inspeção, nenhuma instrução no *loop* efetua a inserção de elementos em *imagemTrabalho*, incluindo as instruções do Algoritmo 23 (ATUALIZAÇÃOALGORITMOPROPOSTO).
2. Novamente por inspeção, juntamente com o Corolário 5.4, temos que todas as coordenadas de interesse inseridas na matriz esparsa devem estar em *imagemTrabalho*.
3. Caso $c \in B_x$, c é simultaneamente (i) removido da matriz esparsa, conforme as linhas 18 a 23 de ALGORITMOPROPOSTO; (ii) removido de *imagemTrabalho*, conforme a linha 3 de ATUALIZAÇÃOALGORITMOPROPOSTO; e (iii) marcado em *resultado*, conforme a linha 5 de ATUALIZAÇÃOALGORITMOPROPOSTO.

Analisando os três fatos em conjunto, temos que $c \in \mathbb{Z}^d$ será marcado não mais que uma vez em *resultado*. \square

Teorema 5.10. *O Algoritmo 23 (ATUALIZAÇÃOALGORITMOPROPOSTO) efetua a manutenção correta das invariantes apresentadas no enunciado do Lema 5.5, de modo que ao término de sua execução valem as seguintes relações:*

(A) a variável *imagemTrabalho* equivale à relação

$$\text{imagemTrabalho} = I_x = \varepsilon_{L_x}(\dots(\varepsilon_{L_0}(X))\dots) ;$$

(B) a dimensão da matriz esparsa bidimensional relacionada às coordenadas de U é composta por cabeças de listas duplamente ligadas, de modo que a composição dessas listas é definida como

$$\text{vetorListasUnião}[i] = \mathcal{F}_{x+1}^{\text{elementosUnião}[i]} \forall \text{elementosUnião}[i] \in U ;$$

(C) a dimensão da matriz esparsa bidimensional relacionado às coordenadas de interesse equivale ao conjunto de elementos determinado pela relação

$$\text{listaNósBorda} = \text{matrizNósBorda} = D_{x+1} = \bigcup_{\text{elementosUnião}[i] \in U} \mathcal{F}_{x+1}^{\text{elementosUnião}[i]} = I_x \setminus \varepsilon_U(I_x) ;$$

(D) a imagem binária representada pela variável *imagemCandidato* equivale ao conjunto de elementos que já foram inseridos em *listaNósBorda*, ou seja,

$$\text{imagemCandidato} = \bigcup_{w=1}^{x+1} D_w ; e$$

(E) a imagem em níveis de cinza resultado atualiza corretamente os valores obtidos através da borda B_x para a Transformada da Erosão, conforme a relação

$$\text{TRANSFORMADAEROSÃO}(A, \mathcal{L}, k)[c] = \text{iteração} : c \in B_x .$$

Demonstração. Inicialmente, deve-se avaliar os valores armazenados nos parâmetros de entrada do algoritmo.

imagemTrabalho, *imagemCandidato* e *resultado* representam respectivamente as imagens I_{x-1} , $\bigcup_{w=1}^x D_w$ e a imagem em níveis de cinza que atende a invariante E do Lema 5.5, uma vez que as instruções contidas nas linhas 8 a 24 do Algoritmo 24 (ALGORITMOPROPOSTO) não alteraram tais variáveis. *listaNósBorda* e *matrizNósBorda* possuem seus nós de borda definidos de acordo com a Equação 5.2, enquanto que *vetorListasUnião* possui seus nós de ligação definidos de acordo com a Equação 5.3. *aAtualizar* contém as coordenadas da borda B_x , conforme definido pelo Lema 5.5. *cardinalidadeUnião* e *elementosUnião* são definidos como os valores de saída homônimos calculados pelo Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE). *iteração* possui o valor atualizado pela instrução da linha 7 do Algoritmo 24 (ALGORITMOPROPOSTO).

Através da linha 3, *imagemTrabalho* é atualizada para atender a relação A do enunciado, conforme demonstrado no Lema 5.8. Desse modo, temos que tal atualização garante a invariante A do Lema 5.5 para o próximo valor de x .

Através da linha 6, *resultado* é atualizado de modo a atender a relação E do enunciado. De acordo com o Lema 5.9, cada coordenada $c \in \mathbb{Z}^d$ será marcado não mais que uma vez. Particularmente, temos o caso em que $c \in B_x$: como a linha 6 efetua a marcação descrita pela relação E do enunciado, então temos que a marcação realizada determina *iteração* como o valor máximo descrito pela invariante E do Lema 5.5, de modo a garanti-la para o próximo valor de x .

Para atender as relações B, C e D, deve-se atualizar a matriz esparsa. Como tal atualização é efetuada? Através da inserção de novos nós de ligação, uma vez que as coordenadas dos elementos de B_x não possuem nenhum elemento em I_x . Essa atualização é realizada pelas instruções das linhas 7 a 26.

A coordenada c é definida pela instrução da linha 7 e as instruções contidas entre a linha 8 e 21 serão executadas. Esse trecho efetua *cardinalidadeUnião* verificações: uma verificação para cada elemento u da união U de elementos estruturantes de \mathcal{L} , de modo que *ligaçãoEncontrada* recebe **verdadeiro** no caso em que $c - u \in I_x$. Note que essa verificação é um pouco diferente quando comparado à verificação efetuada no Algoritmo 22 (INICIALIZEMATRIZESPARSA), pois no caso da atualização sabe-se que $c \notin I_x$, mas não se sabe de antemão a informação a ser verificada.

No caso em que *ligaçãoEncontrada* seja validada pela instrução da linha 21, as instruções contidas entre a linha 22 e 26 efetuam as atualizações de modo a atender às relações citadas no enunciado.

A linha 22 verifica se será efetuada a inserção do primeiro nó de ligação referente à coordenada de interesse $c - u$: em caso positivo, a linha 23 atualiza *imagemCandidato* de modo a registrar as inserções do nó de borda relacionado a $c - u$ em *listasNósBorda* e em *matrizNósBorda*. A instanciação do nó de borda e as inserções ocorrem na linha seguinte (24). Considerando o estado como parâmetro de entrada para as variáveis *imagemCandidato*, *listasNósBorda* e *matrizNósBorda*, temos que *imagemCandidato* é alterada de modo a atender a relação D do enunciado e as variáveis *listasNósBorda* e *matrizNósBorda* são modificadas para atender a relação C do enunciado. Através dessas atualizações, temos que as invariantes C e D do Lema 5.5 são garantidas para o próximo valor de x .

A linha 25 recupera o nó da borda relacionado à coordenada $c - u$ enquanto a linha 26 é responsável por instanciar o nó de ligação que relaciona a coordenada de interesse $c - u$ à coordenada u da união U , ao ligar o nó da borda com a lista em *vetorListasUnião* referente a u . Desse modo, *vetorListasUnião* é atualizado conforme a relação B do enunciado. Portanto, temos que tal atualização garante a invariante B do Lema 5.5 para o próximo valor de x . \square

Para concluir a demonstração referente ao algoritmo de Transformada da Erosão, apresentamos o Teorema 5.11.

Teorema 5.11. *O Algoritmo 20 (TRANSFORMADAEROSÃOPROPOSTA) devolve a saída da Equação 2.14.*

Demonstração. O Lema 5.5 demonstrou com o auxílio de outros lemas e teoremas as etapas de inicialização, manutenção e término de suas cinco invariantes utilizadas ao longo da execução do Algoritmo 21 (TRANSFORMADAPROPOSTA).

Através da etapa de inicialização, *resultado* é atribuído inicialmente com os valores conforme a relação

$$\text{TRANSFORMADAEROSÃO}(X, \mathcal{L}, k)[c] = \begin{cases} 0 & : c \in X \\ -1 & : c \notin X \end{cases}.$$

Enquanto se realizam as $k \cdot m$ erosões sucessivas para a determinação do resultado da Transformada

de Erosão, a relação seguinte foi validada como a invariante E do lema em questão:

$$\text{TRANSFORMADAEROSÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \max\{w \mid c \in \varepsilon_{wL_k}(\dots(\varepsilon_{wL_2}(\varepsilon_{wL_1}(X)))\dots)\} & : c \in \bigcup_{y=1}^{x-1} B_y \\ 0 & : c \in X \setminus \bigcup_{y=1}^{x-1} B_y \\ -1 & : c \notin X \end{cases}$$

O Lema 5.9 é utilizado para verificar o corretismo da marcação da coordenada c como o valor máximo apresentado. Por fim, dado que a condição de parada é verificada quando D_x não possui mais coordenadas de interesse, ou seja, $B_x = \emptyset$, então *resultado* armazena os valores estabelecidos pela Equação 2.14:

$$\text{TRANSFORMADAEROSÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \max\{w \mid c \in \varepsilon_{wL_k}(\dots(\varepsilon_{wL_2}(\varepsilon_{wL_1}(X)))\dots)\} & : c \in X \\ -1 & : c \notin X \end{cases}$$

□

5.3 Algoritmo de Transformada da Dilatação

A prova do corretismo para o algoritmo proposto em relação à Transformada da Dilatação utiliza a dualidade entre as operações de Erosão e Dilatação representada através da Equação 2.6

$$\delta_W(G) = (\varepsilon_{W^r}(G^c))^c ,$$

de modo a utilizar as demonstrações efetuadas na Seção 5.2 para provar o cálculo correta da Equação 2.15, armazenado em *resultado*.

A primeira etapa consiste em demonstrar que o valor de I_x conforme definido na Tabela 2.5 é apropriado para o cálculo da Transformada da Dilatação.

Teorema 5.12. *Seja $x \geq 0$ um inteiro. Dada a seguinte relação:*

$$I_x = \underbrace{\delta_{L_x}(\dots(\delta_{L_1}(\delta_{L_0}(X)))\dots)}_P = \underbrace{(\varepsilon_{(L_x)^r}(\dots(\varepsilon_{(L_1)^r}(\varepsilon_{(L_0)^r}(X^c)))\dots))}_Q^c ,$$

demonstraremos que o termo P equivale ao termo Q.

Demonstração. O termo P foi apresentado na Tabela 2.4, como a definição de I_x para o Algoritmo 3 (TRANSFORMADADILATAÇÃOSIMPLES), enquanto que o termo Q foi apresentado na Tabela 2.5, de modo a definir I_x para os algoritmos que utilizam a propriedade de dualidade através da Equação 2.6:

$$\delta_W(G) = (\varepsilon_{W^r}(G^c))^c .$$

Inicialmente, devemos provar a relação para $x = 0$. Sua demonstração é trivial, uma vez que

$$I_0 \xleftrightarrow{\text{Definição de } P} \delta_{L_0}(X) \xleftrightarrow{\text{Equação 2.6}} (\varepsilon_{(L_0)^r}(X^c))^c .$$

Para $x > 0$, estabelecemos a seguinte relação com base no termo P apresentado para calcular um dado

I_x a partir de I_{x-1} :

$$I_x = \underbrace{\delta_{L_x}(I_{x-1})}_R .$$

Para a prova por indução, utilizaremos a seguinte invariante:

$$I_{x-1} = (\varepsilon_{(L_{x-1})^r}(\dots(\varepsilon_{(L_1)^r}(\varepsilon_{(L_0)^r}(X^c))\dots)))^c .$$

$$\begin{aligned} I_x &\xleftrightarrow{\text{Definição de } P} \delta_{L_x}(\dots(\delta_{L_1}(\delta_{L_0}(X)))\dots) \\ &\xleftrightarrow{\text{Definição de } R} \delta_{L_x}(I_{x-1}) \\ &\xleftrightarrow{\text{Equação 2.6}} (\varepsilon_{(L_x)^r}((I_{x-1})^c))^c \\ &\xleftrightarrow{\text{Hipótese de indução}} (\varepsilon_{(L_x)^r}(((\varepsilon_{(L_{x-1})^r}(\dots(\varepsilon_{(L_1)^r}(\varepsilon_{(L_0)^r}(X^c))\dots)))^c)^c))^c \\ &\xleftrightarrow{(Y^c)^c = Y} (\varepsilon_{(L_x)^r}(\varepsilon_{(L_{x-1})^r}(\dots(\varepsilon_{(L_1)^r}(\varepsilon_{(L_0)^r}(X^c))\dots))))^c . \end{aligned}$$

□

Como resultado do Teorema 5.12, temos que as dilatações sequenciais requeridas pela Transformada da Dilatação podem ser calculadas através de erosões sequenciais. Dessa forma, apresentamos a seguir como o resultado obtido na Seção 5.2 permite calcular corretamente a operação através da Transformada da Erosão.

Corolário 5.13. *Para $x > 0$, as listas individuais para cada um dos elementos contidos na união dos elementos estruturantes de \mathcal{L} são utilizadas no Algoritmo 21 (TRANSFORMADAPROPOSTA) para o cálculo de*

$$\delta_{L_x}(I_{x-1}) .$$

Demonstração. Aplicando a Equação 2.6 na operação apresentada no enunciado, podemos reescrever a operação de Dilatação em função da operação de Erosão como

$$(\varepsilon_{(L_x)^r}((I_{x-1})^c))^c .$$

Quando comparado a operação obtida com a estabelecida no enunciado do Corolário 5.4, podemos verificar que o que difere são três operações: (i) o complemento da imagem I_{x-1} ; (ii) a reflexão de L_x ; e (iii) o complemento ao final da operação de Erosão.

O complemento da imagem I_{x-1} é obtido através das linhas 7 e 14 do Algoritmo 22 (INICIALIZEMATRIZESPARGA) e das linhas 5 e 20 do Algoritmo 23 (ATUALIZAÇÃOALGORITMOPROPOSTO).

A reflexão de L_x é obtida através da linha 14 de INICIALIZEMATRIZESPARGA e da linha 14 de ATUALIZAÇÃOALGORITMOPROPOSTO.

O complemento ao final da operação de Erosão é efetuado através da linha 5 de ATUALIZAÇÃOALGORITMOPROPOSTO. □

Teorema 5.14. *O Algoritmo 19 (TRANSFORMADADILATAÇÃOPROPOSTA) devolve a saída da Equação 2.15.*

Demonstração. Através de alguns ajustes no Algoritmo 21 (TRANSFORMADAPROPOSTA), temos que a Transformada da Dilatação pode ser calculada através da Transformada da Erosão, conforme apresentado pelo

Corolário 5.13. Dessa forma, são apresentados algumas diferenças nesta demonstração quando comparado ao do Teorema 5.11.

O Lema 5.5 demonstrou com o auxílio de outros lemas e teoremas as etapas de inicialização, manutenção e término de suas cinco invariantes utilizadas ao longo da execução do Algoritmo 21

(TRANSFORMADAPROPOSTA).

Através da etapa de inicialização, *resultado* é atribuído inicialmente com os valores conforme a relação

$$\text{TRANSFORMADADILATAÇÃO}(X, \mathcal{L}, k)[c] = \begin{cases} 0 & : c \in X \\ -1 & : c \notin X \end{cases}.$$

Enquanto se realizam as $k \cdot m$ erosões sucessivas em X^c para a determinação do resultado da Transformada de Dilatação, a relação seguinte é derivada da invariante E do lema em questão:

$$\text{TRANSFORMADADILATAÇÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \min\{w \mid c \in \delta_{wL_k}(\dots(\delta_{wL_2}(\delta_{wL_1}(X)))\dots)\} \\ \quad : c \in X \cup \bigcup_{y=1}^{x-1} B_y \\ -1 : c \in \mathbb{Z}^d \setminus \left(X \cup \bigcup_{y=1}^{x-1} B_y \right) \end{cases}$$

O Lema 5.9 é utilizado para verificar o corretismo da marcação da coordenada c como o valor máximo apresentado para o registro relacionado à Erosão. De modo equivalente, como o registro relacionado a c em *resultado* é efetuado no máximo uma vez, o valor registrado será o mínimo. Por fim, dado que a condição de parada é verificada quando D_x não possui mais coordenadas de interesse, ou seja, $B_x = \emptyset$, então *resultado* armazena os valores estabelecidos pela Equação 2.15:

$$\text{TRANSFORMADADILATAÇÃO}(X, \mathcal{L}, k)[c] = \begin{cases} \min\{w \mid c \in \delta_{wL_k}(\dots(\delta_{wL_2}(\delta_{wL_1}(X)))\dots)\} \\ \quad : \mathbf{se} \exists w, c \in \delta_{wL_k}(\dots(\delta_{wL_2}(\delta_{wL_1}(X)))\dots) \\ -1 : \forall w, c \notin \delta_{wL_k}(\dots(\delta_{wL_2}(\delta_{wL_1}(X)))\dots) \end{cases}$$

Cabe ressaltar que o algoritmo determina os valores para as coordenadas c que se encontram dentro da caixa delimitadora de X . □

Capítulo 6

Experimento

6.1 Procedimento experimental

O experimento foi projetado para medir o tempo de execução da implementação* de alguns algoritmos apresentados neste trabalho para a Transformada da Erosão e Transformada da Dilatação para o caso em que $d = 2$.

Para validarmos o cálculo teórico de tempo computacional com as implementações desenvolvidas, variamos os seguintes parâmetros:

Tabela 6.1: Lista de parâmetros analisados no procedimento experimental.

Parâmetro	Valores
Operação	<ul style="list-style-type: none"> • Transformada da Erosão • Transformada da Dilatação
Implementação	<ul style="list-style-type: none"> • Algoritmo 9 (TRANSFORMADADUALLINEAR) • Algoritmo 13 (TRANSFORMADADUALLINEARDEBORDA) • Algoritmo 21 (TRANSFORMADAPROPOSTA)
Imagem de Entrada (X)	<ul style="list-style-type: none"> • Figura 6.1 • Figura 6.2a • Figura 6.2b • Figura 6.2c • Figura 6.2d
Tamanho dos Elementos Estruturantes em \mathcal{L}	<ul style="list-style-type: none"> • 3×3 • 5×5
Número k de Elementos Estruturantes em \mathcal{L}	<ul style="list-style-type: none"> • 1 a 15 (para elementos estruturantes 3×3) • 1 a 15 (para elementos estruturantes 5×5)

Para cada uma dessas 900 combinações, foram efetuadas 1001 execuções com o intuito de extrair medidas relacionadas ao tempo de execução de uma determinada combinação.

Inicialmente foi selecionada uma imagem de 2500×2500 pixels referenciada pelo rótulo Figura 5.3 em (Machado, 2008), apresentada aqui como Figura 6.1. Tal imagem é composta por quatro quadrantes: (i) quatro objetos naturais, extraídas de (Latecki *et al.*, 2000) (Figura 6.2a – 1268×1142 pixels); (ii) linhas igualmente espaçadas orientadas a 45° (Figura 6.2b – 1232×1142 pixels); (iii) ruído sal e pimenta distribuído uniformemente (Figura 6.2c – 1253×1358 pixels); e (iv) outros quatro objetos naturais, também extraídas de (Latecki *et al.*, 2000) (Figura 6.2d – 1247×1358 pixels). Dada a natureza de cada um desses

*O código-fonte foi disponibilizado em <https://github.com/ayharano/proposed-dual-linear-mm-transform> e em <http://www.ime.usp.br/~ayharano/>.



Figura 6.1: Imagem de teste oriunda de (Machado, 2008).

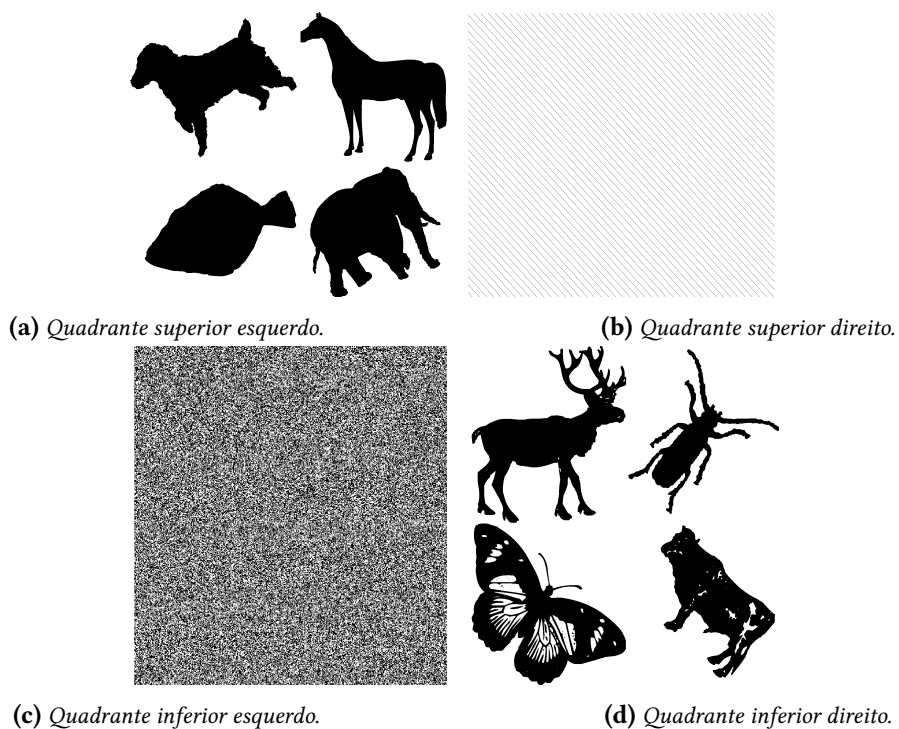


Figura 6.2: Divisão da Figura 6.1 de acordo com seus quadrantes.

quadrantes, optamos por extraí-las para a utilização como imagens de entrada individuais no procedimento experimental.

Para o tamanho de elemento estruturante de \mathcal{L} de valor 3×3 , foram calculadas 15 listas, $\mathcal{L}_{1,3 \times 3}$ a $\mathcal{L}_{15,3 \times 3}$, de modo que:

$$\begin{aligned}
\mathcal{L}_{1,3 \times 3} &= \langle L_{1,3 \times 3} \rangle & (k = 1) \\
\mathcal{L}_{2,3 \times 3} &= \langle L_{1,3 \times 3}, L_{2,3 \times 3} \rangle & (k = 2) \\
&\dots = \dots & \dots \\
\mathcal{L}_{15,3 \times 3} &= \langle L_{1,3 \times 3}, L_{2,3 \times 3}, \dots, L_{15,3 \times 3} \rangle & (k = 15)
\end{aligned}$$

De modo análogo, para o tamanho de elemento estruturante de \mathcal{L} de valor 5×5 , foram calculadas 15 listas, $\mathcal{L}_{1,5 \times 5}$ a $\mathcal{L}_{15,5 \times 5}$, de modo que:

$$\begin{aligned}
\mathcal{L}_{1,5 \times 5} &= \langle L_{1,5 \times 5} \rangle & (k = 1) \\
\mathcal{L}_{2,5 \times 5} &= \langle L_{1,5 \times 5}, L_{2,5 \times 5} \rangle & (k = 2) \\
&\dots = \dots & \dots \\
\mathcal{L}_{15,5 \times 5} &= \langle L_{1,5 \times 5}, L_{2,5 \times 5}, \dots, L_{15,5 \times 5} \rangle & (k = 15)
\end{aligned}$$

Como de costume na área de morfologia matemática, os elementos estruturantes 3×3 e 5×5 possuem suas origens no centro de matrizes com os respectivos tamanhos. As Figuras 6.3 e 6.4 apresentam os elementos estruturantes 3×3 e 5×5 utilizados, respectivamente.

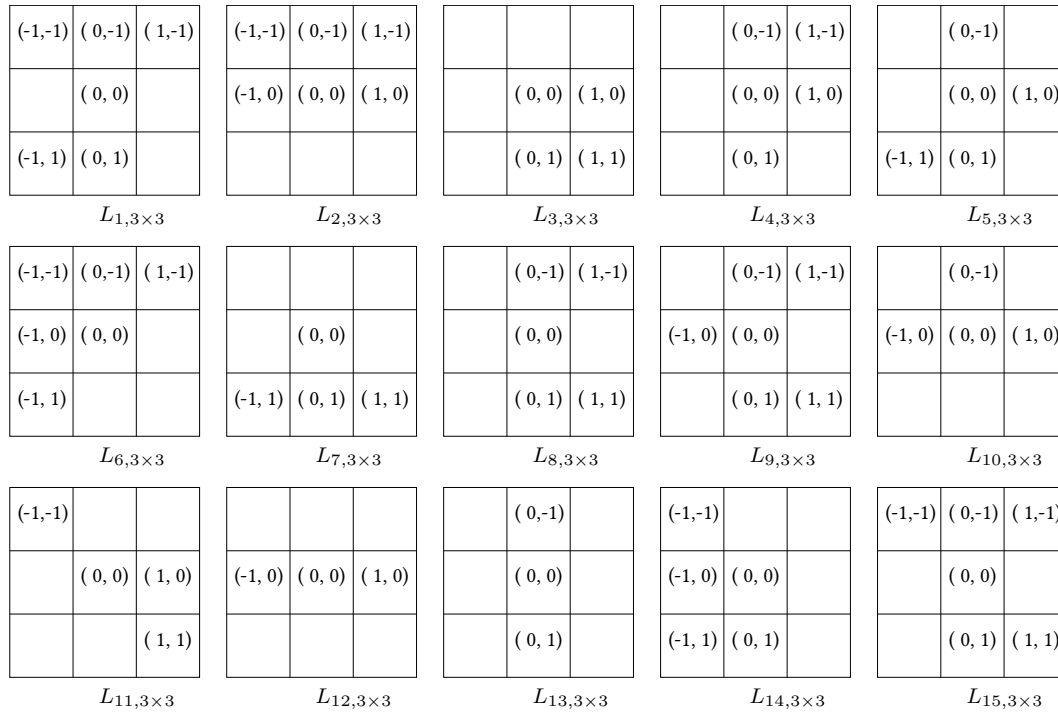


Figura 6.3: Elementos estruturantes 3×3 utilizados no procedimento experimental.

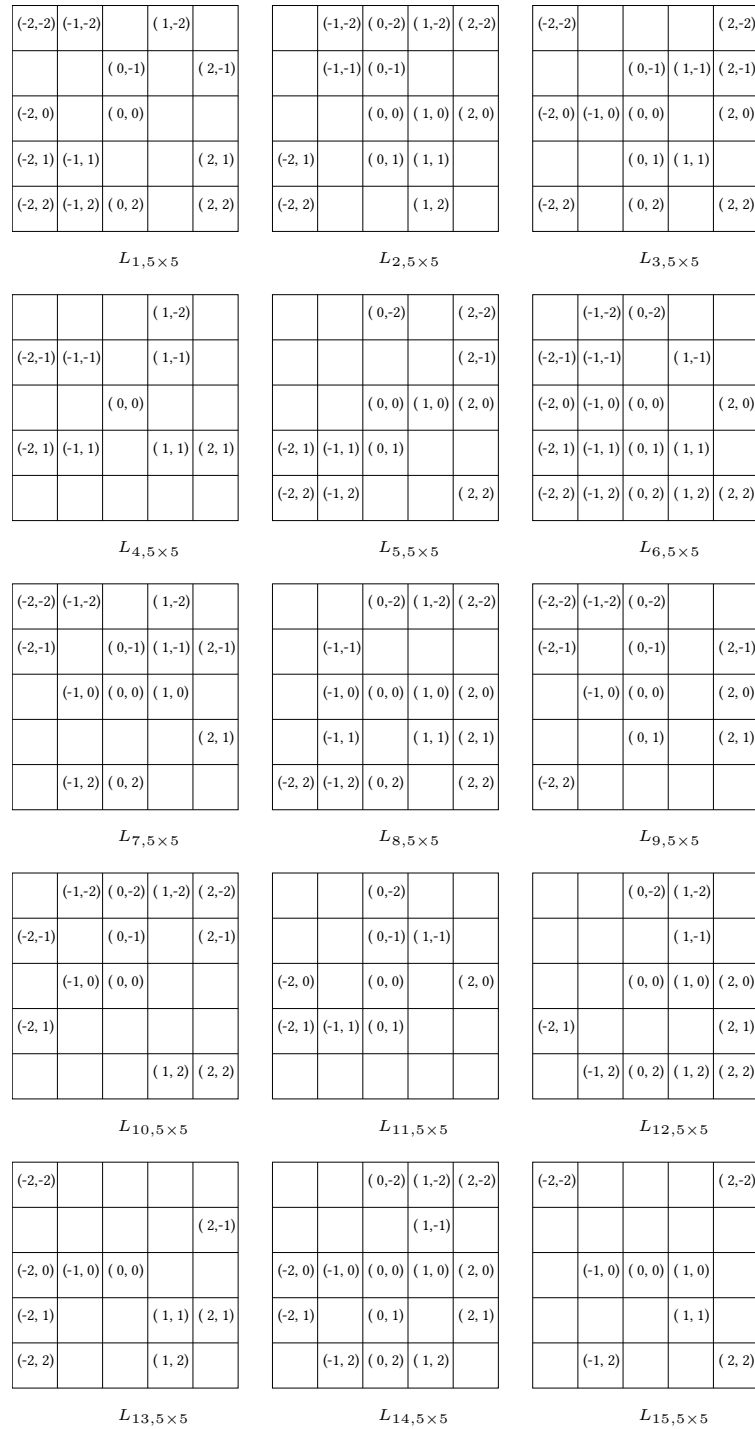
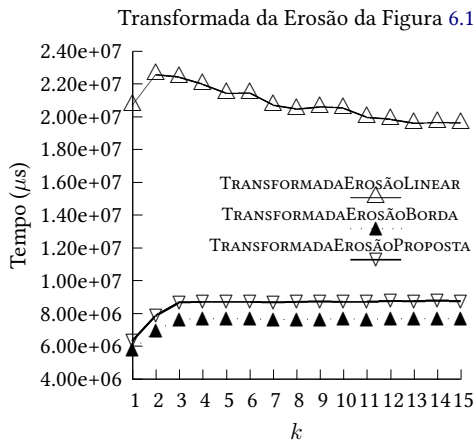


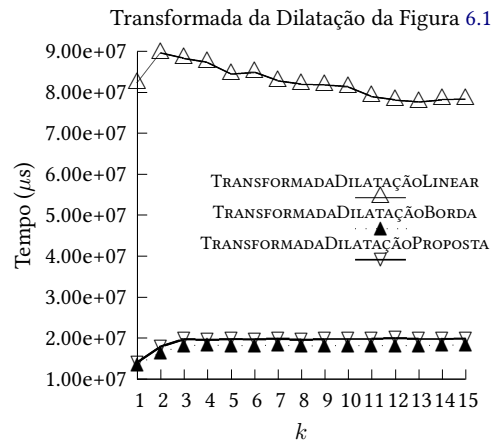
Figura 6.4: Elementos estruturantes 5×5 utilizados no procedimento experimental.

6.2 Resultados

Conforme citado anteriormente, foram efetuadas 1001 execuções para cada uma das 900 combinações de parâmetros de entrada. Os dados de valores mínimos, máximos, médias, medianas e desvios-padrão calculados com base nessas 1001 execuções de cada combinação estão detalhados na Seção B.1 do Apêndice B. A partir dos dados de tempo médio obtidos para cada combinação, geramos gráficos apresentados nas Figuras 6.5 a 6.14 para efetuar análises qualitativas de acordo com a natureza das imagens e algoritmos utilizados.

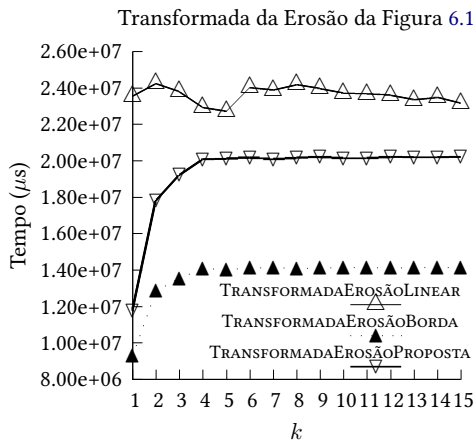


(a) Transformada da Erosão.

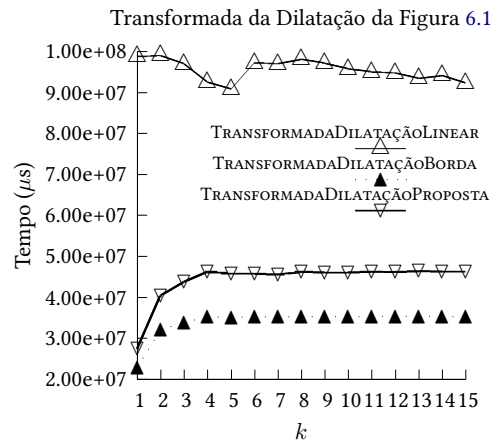


(b) Transformada da Dilatação.

Figura 6.5: Gráficos do tempo médio para a Figura 6.1 utilizando elementos estruturantes 3×3 .

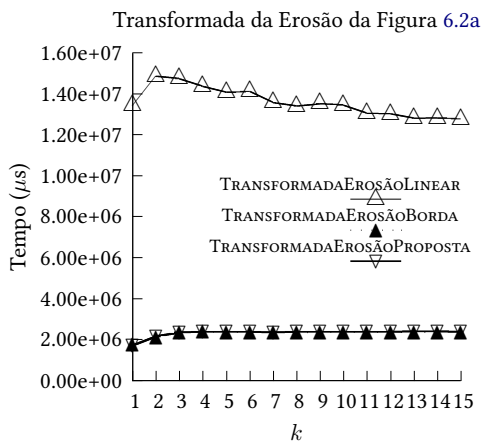


(a) Transformada da Erosão.

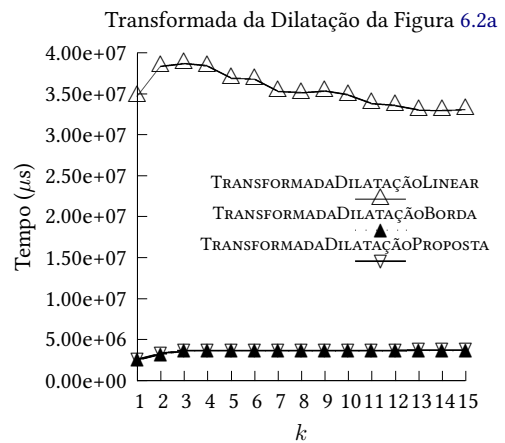


(b) Transformada da Dilatação.

Figura 6.6: Gráficos do tempo médio para a Figura 6.1 utilizando elementos estruturantes 5×5 .



(a) Transformada da Erosão.



(b) Transformada da Dilatação.

Figura 6.7: Gráficos do tempo médio para a Figura 6.2a utilizando elementos estruturantes 3×3 .

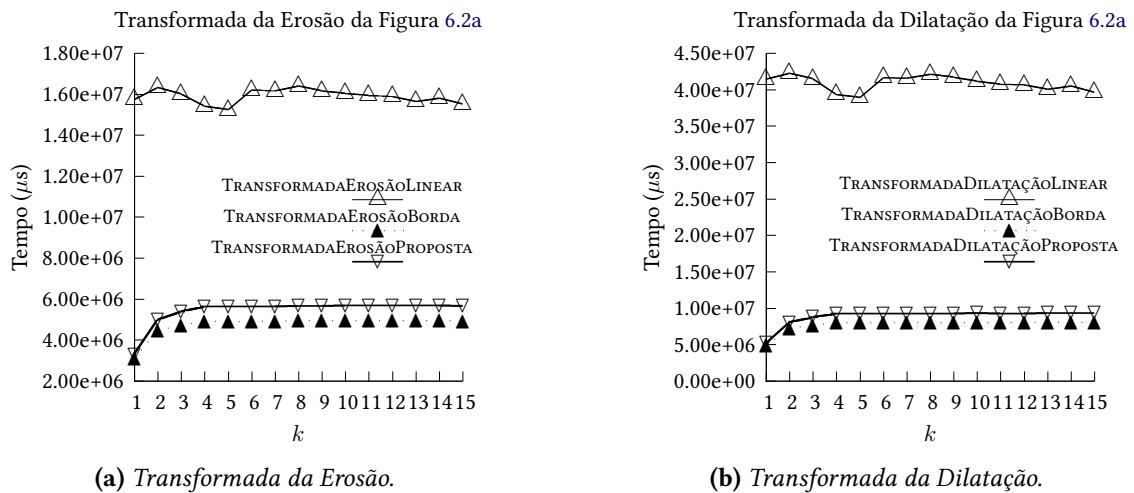


Figura 6.8: Gráficos do tempo médio para a Figura 6.2a utilizando elementos estruturantes 5×5 .

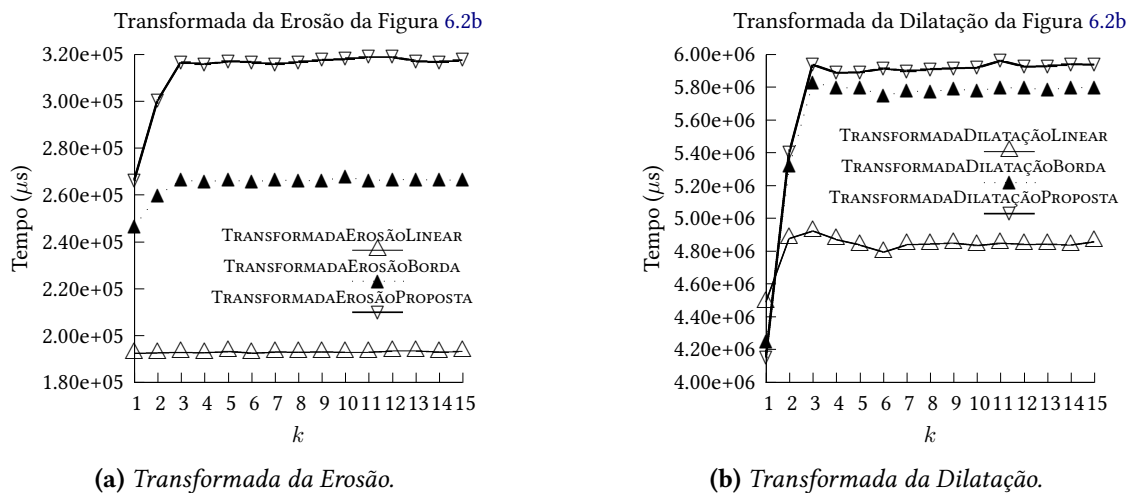


Figura 6.9: Gráficos do tempo médio para a Figura 6.2b utilizando elementos estruturantes 3×3 .

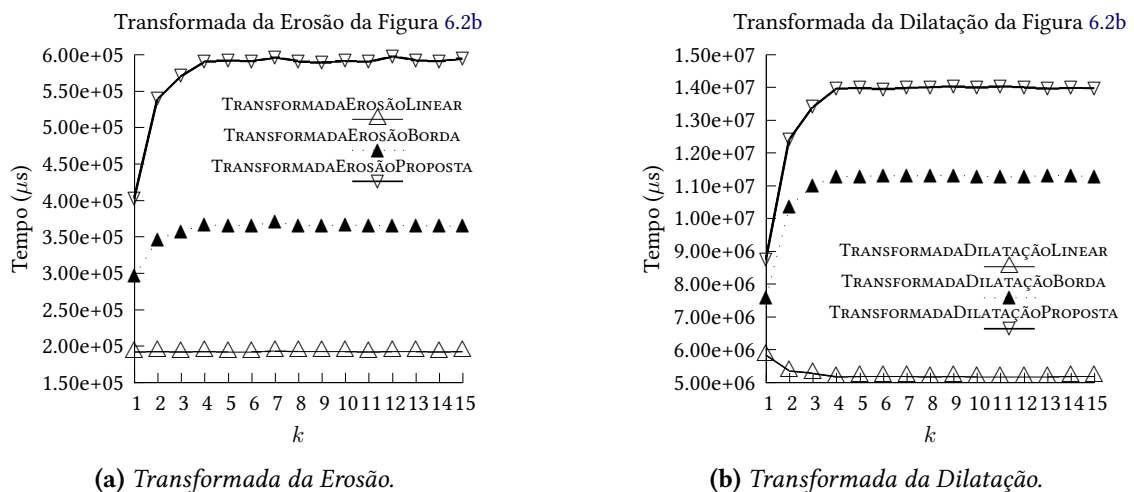


Figura 6.10: Gráficos do tempo médio para a Figura 6.2b utilizando elementos estruturantes 5×5 .

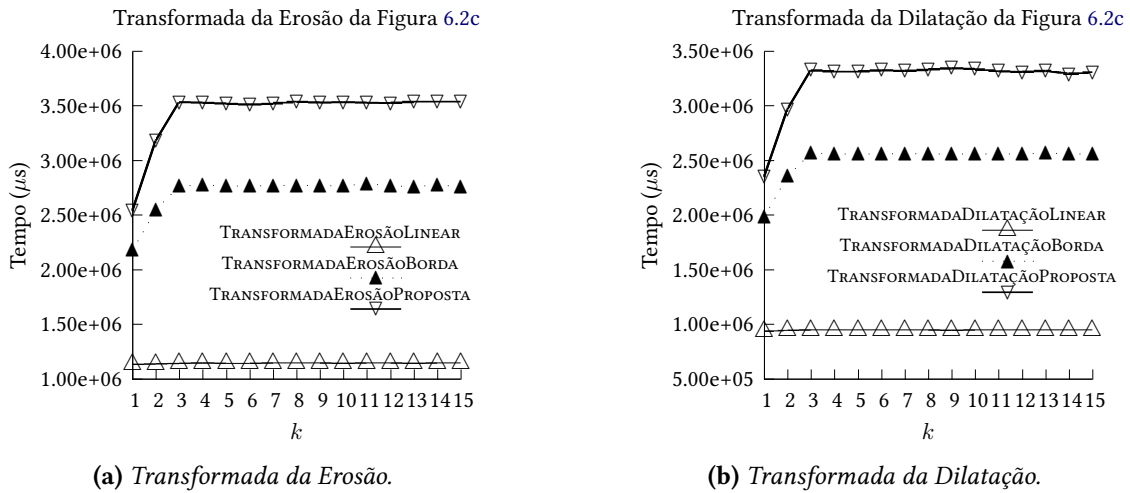


Figura 6.11: Gráficos do tempo médio para a Figura 6.2c utilizando elementos estruturantes 3 × 3.

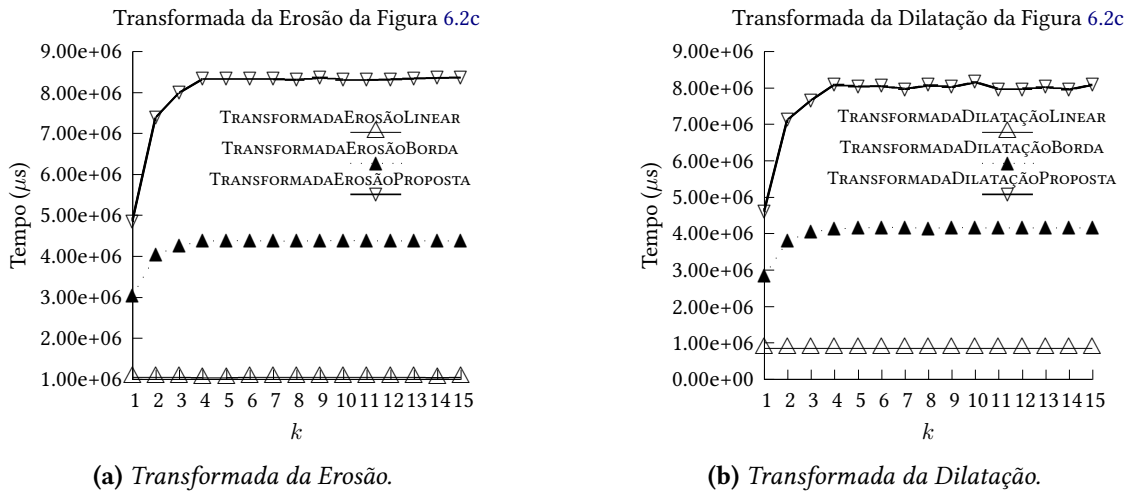


Figura 6.12: Gráficos do tempo médio para a Figura 6.2c utilizando elementos estruturantes 5 × 5.

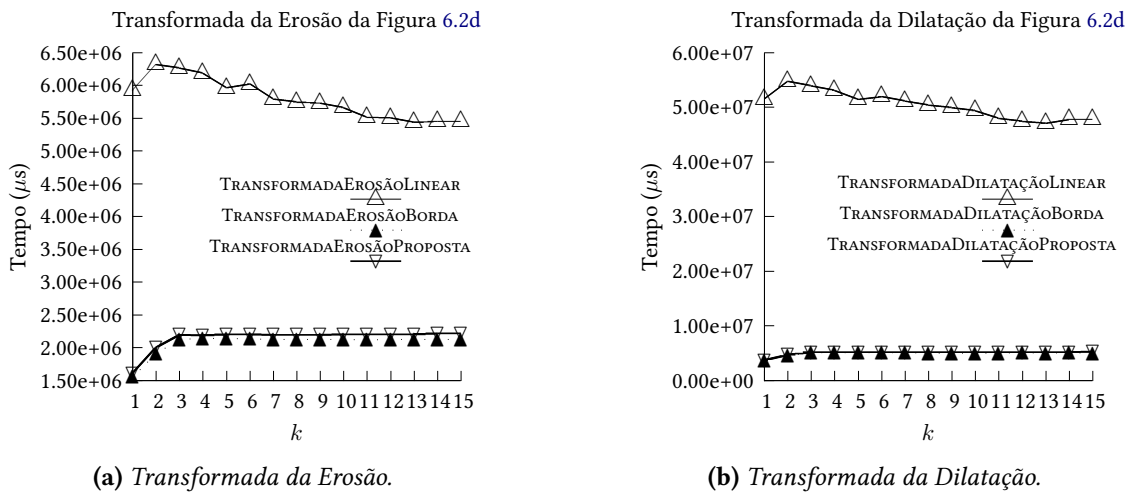


Figura 6.13: Gráficos do tempo médio para a Figura 6.2d utilizando elementos estruturantes 3 × 3.

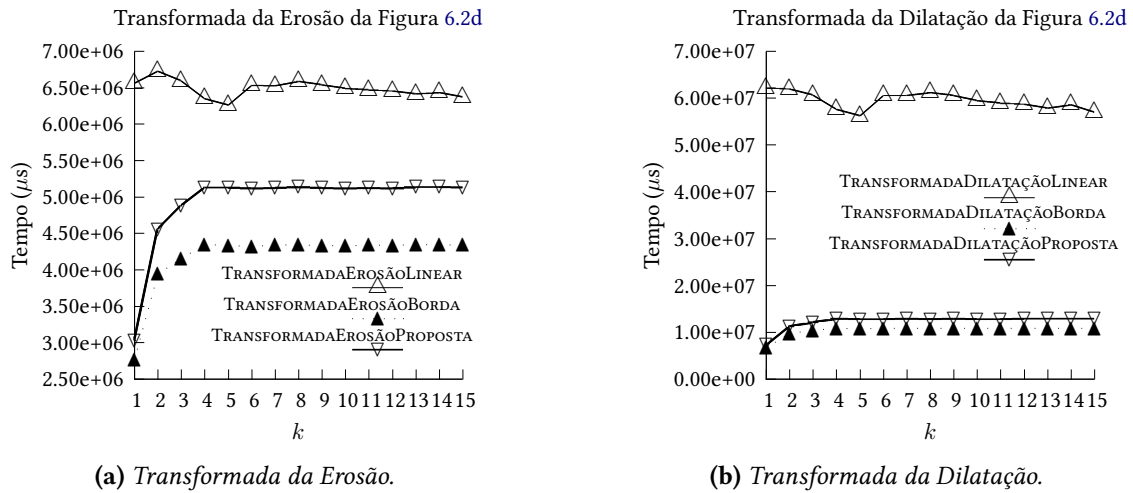


Figura 6.14: Gráficos do tempo médio para a Figura 6.2d utilizando elementos estruturantes 5×5 .

6.3 Análise do Dados Experimentais

6.3.1 Análise do Tempo Computacional Teórico

Tabela 6.2: Análise de Complexidade das implementações utilizadas.

	Tempo Computacional	Espaço Computacional
<p>Algoritmo 9</p> <p>(TRANSFORMADA DUAL LINEAR)</p>	$O\left([X] + \left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k [L_i] + k \cdot \left \bigcup_{i=1}^k L_i\right + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot L_x \right)$	$O\left([X] + \left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k L_i \right)$
<p>Algoritmo 15</p> <p>(TRANSFORMADA DUAL LINEAR DE BORDA)</p>	$O\left([X] + \left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k [L_i] + (k + I_0) \cdot \left \bigcup_{i=1}^k L_i\right + \sum_{x=1}^{m \cdot k} \left borda_{\bigcup_{i=1}^k L_i}(I_{x-1})\right \cdot L_x + \sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left([X] + \left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k L_i \right)$
<p>Algoritmo 21</p> <p>(TRANSFORMADA PROPOSTA)</p>	$O\left([X] + \left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k [L_i] + (k + I_0) \cdot \left \bigcup_{i=1}^k L_i\right + \left borda_{\bigcup_{i=1}^k L_i}(I_0)\right \cdot \left \bigcup_{i=1}^k L_i\right + \sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left([X] + \max_x \left\{ \left borda_{\bigcup_{i=1}^k L_i}(I_x)\right \cdot \left \bigcup_{i=1}^k L_i\right \right\}\right)$

Como base para a análise do resultado experimental, é apresentada uma breve análise do tempo computacional teórico dos algoritmos utilizados. O Apêndice A apresenta a complexidade computacional calculada linha a linha para os algoritmos apresentados nos Capítulos 3 e 4, enquanto que a Tabela 6.2 resume

os valores teóricos calculados para os algoritmos utilizados como base das implementações.

Podemos verificar que os quatro primeiros termos do tempo computacional apresentado para cada um dos algoritmos do experimento são os mesmos, uma vez que é necessário percorrer toda a caixa delimitadora da imagem de entrada pelo menos uma vez ($O([X])$) e todos os operadores executam o Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE) para linearizar os dados relacionados à lista \mathcal{L} de k elementos estruturantes $\left(O \left(\left[\bigcup_{i=1}^k L_i \right] + \sum_{i=1}^k [L_i] + k \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$.

O Algoritmo 9 (TRANSFORMADADUALLINEAR) lineariza as coordenadas de interesse da imagem de entrada, de modo que, para cada valor de x , todas as coordenadas de interesse da imagem I_{x-1} são verificadas de acordo com cada elemento do L_x analisado $\left(O \left(\sum_{x=1}^{m \cdot k} |I_{x-1}| \cdot |L_x| \right) \right)$.

Os Algoritmos 15 (TRANSFORMADADUALLINEARDEBORDA) e 21 (TRANSFORMADAPROPOSTA) utilizam um critério como base para o armazenamento dos dados relacionados às coordenadas de interesse: para um dado valor de x , as coordenadas armazenadas devem pertencer à borda de I_{x-1} de acordo com a união dos elementos estruturantes de \mathcal{L} . Relacionado a esse critério, os termos dos tempos computacionais comuns a ambos os algoritmos $\left(O \left(|I_0| \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$ e $\left(O \left(\sum_{x=1}^{m \cdot k} |borda_{L_x}(I_{x-1})| \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$ são responsáveis respectivamente pela validação inicial e a manutenção desse critério ao longo da execução dos mesmos.

Dado que ambos os algoritmos utilizam o mesmo critério, o que os difere?

TRANSFORMADADUALLINEARDEBORDA armazena as coordenadas de interesse que cumprem o critério e, para os valores de x , verifica cada coordenada armazenada da borda com as coordenadas do L_x

$\left(O \left(\sum_{x=1}^{m \cdot k} \left| borda_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right| \cdot |L_x| \right) \right)$. A matriz esparsa bidimensional da TRANSFORMADAPROPOSTA foi desenhada para que as verificações descritas para TRANSFORMADADUALLINEARDEBORDA não sejam efetuadas. Tal fato se deve ao uso dos nós de ligação, os quais, para os valores de x , atrelam uma coordenada de interesse de I_{x-1} que cumpre o critério descrito no parágrafo anterior a uma coordenada da união

dos elementos estruturantes de \mathcal{L} . A inicialização da matriz requer $\left(O \left(\left| borda_{\bigcup_{i=1}^k L_i}(I_0) \right| \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$

e a utilização da matriz ao longo do algoritmo consome $\left(O \left(\sum_{x=1}^{m \cdot k} |borda_{L_x}(I_{x-1})| \right) \right)$, o qual é englobado pelo termo previamente descrito referente à manutenção do critério discutido. Dessa forma, ao efetuarmos uma comparação termo a termo dos tempos computacionais calculados para os dois algoritmos, TRANSFORMADAPROPOSTA requer menor tempo.

6.3.2 Análise do Tempo Médio de Execução

Efetuada uma análise qualitativa referente às curvas dos gráficos contidos nas Figuras 6.5 a 6.14 apresentados na Seção 6.2 e apresentamos algumas averiguações sobre os dados obtidos.

1. Para as Figuras 6.2b e 6.2c, o Algoritmo 9 (TRANSFORMADADUALLINEAR) obteve tempos médios de execução menores do que os obtidos pelos outros dois algoritmos, conforme os gráficos apresentados nas Figuras 6.9, 6.10, 6.11 e 6.12.
2. Para cada uma das 900 combinações, o tempo médio de execução obtido para o Algoritmo 15 (TRANSFORMADADUALLINEARDEBORDA) foi menor que do que o obtido pelo Algoritmo 21 (TRANSFORMADAPROPOSTA).

Para a discussão sobre a primeira averiguação, retomamos a descrição das Figuras 6.2b e 6.2c respectivamente como linhas igualmente espaçadas orientadas a 45° e ruído sal e pimenta distribuído uniformemente. O custo de manutenção das premissas adotadas para o desenvolvimento dos Algoritmos 15 (TRANSFORMADADUALLINEARDEBORDA) e 21 (TRANSFORMADAPROPOSTA) está diretamente relacionado à quantidade de elementos nas bordas calculadas. Tal dependência agregada ao fato de que as bordas das imagens em questão serem qualitativamente mais irregulares quando comparadas às bordas obtidas a partir das imagens naturais resultam na primeira averiguação listada.

Como base para a discussão relativa à segunda averiguação, inicialmente verificamos a diferença dos termos apresentados para os tempos computacionais de TRANSFORMADADUALLINEARDEBORDA e TRANSFORMADAPROPOSTA através da Tabela 6.2 e o primeiro algoritmo aparentemente requer maior tempo computacional. Essa comparação não condiz com a averiguação efetuada. Como se justifica essa divergência?

O procedimento experimental estabeleceu, em sua implementação, o uso de estruturas de dados para registrar a contagem de (i) comparações para a determinação de $borda_{L_x}(I_{x-1})$ para todos os valores de x calculados; (ii) comparações para a inserção dos dados referentes às coordenadas de interesse; (iii) acessos à memória para a inserção dos dados referentes às coordenadas de interesse; e (iv) acessos à memória para a remoção dos dados referentes às coordenadas de interesse. Esses dados foram coletados para cada uma das 900 combinações, sendo apresentados na Seção B.2 do Apêndice B. Conforme explicitado na Seção 1.2, TRANSFORMADAPROPOSTA possui como premissa básica a redução do número de comparações para determinar as coordenadas da borda. Tal objetivo foi alcançado e pode ser constatado através do fato de que, para qualquer uma das combinações, o número de comparação para a determinação da borda é zero. Entretanto, os outros 3 quesitos avaliados através das contagens efetuadas para TRANSFORMADAPROPOSTA obtiveram desempenhos piores quando comparados aos resultados obtidos para TRANSFORMADADUALLINEARDEBORDA em cada uma das combinações analisadas.

O tempo computacional calculado para os algoritmos apresentados indica o tempo requerido para o pior caso de execução, ou seja, quando o número de comparações efetuadas é o maior possível. No caso de TRANSFORMADADUALLINEARDEBORDA, foi verificado que o tempo de execução para caso médio é aquém do tempo requerido pelo pior caso, enquanto que a análise de tempo computacional de TRANSFORMADAPROPOSTA reflete também o tempo gasto no caso médio.

Uma das instruções em TRANSFORMADADUALLINEARDEBORDA que suporta essa afirmação é a verificação contida na linha 16 do Algoritmo 18 (ALGORITMODUALLINEARDEBORDA), o qual pode encerrar previamente um bloco de repetição de instruções, caso *removaCandidato* seja validado como **verdadeiro**.

Capítulo 7

Conclusões

7.1 Considerações Finais

Foi apresentado um algoritmo para as operações de Transformada da Erosão e Transformada da Dilatação para imagens binárias em \mathbb{Z}^d , o qual possui como parâmetro de entrada uma decomposição prévia de elementos estruturantes, de modo que cada elemento possua a origem.

A proposta inicial de minimizar o número de comparações necessárias para a determinação das bordas utilizadas ao longo do algoritmo foi efetivada. Dentre os algoritmos duais que utilizam a borda calculada através da união dos elementos da lista \mathcal{L} como parte das invariantes adotadas, pode-se verificar comparativamente que o Algoritmo 21 (TRANSFORMADAPROPOSTA) possui tempo computacional teórico menor que o Algoritmo 15 (TRANSFORMADADUALLINEARDEBORDA).

Entretanto, ao efetuarmos o procedimento experimental, verificamos que TRANSFORMADADUALLINEARDEBORDA obteve tempos médios de execução menores que os obtidos por TRANSFORMADAPROPOSTA para todas as combinações de parâmetros testadas. A justificativa verificada para a divergência observada consiste no fato que, enquanto o tempo computacional calculado indica o tempo requerido para o pior caso de execução, o caso médio de TRANSFORMADADUALLINEARDEBORDA consome um tempo aquém do tempo teórico calculado, enquanto que o caso médio de TRANSFORMADAPROPOSTA requer o tempo calculado como seu pior caso.

Podemos observar também que os dois algoritmos possuem bom desempenho no caso em que o custo de manutenção do algoritmo— i.e., a determinação de novas coordenadas de interesse que pertençam à borda da imagem I_x de acordo com a união dos elementos de \mathcal{L} — seja menor que verificar cada coordenada de interesse de I_x por cada elemento da união dos elementos de \mathcal{L} , para todos os valores de x .

7.2 Sugestões para Pesquisas Futuras

Um fato que intrigou o autor, mas que não pôde ser verificado devido à falha dos testes iniciais e a ausência de tempo hábil para efetuar novos testes relacionados, consiste na maneira como os dados das imagens foram verificados nos algoritmos apresentados.

Pegemos como exemplo a linha 12 do Algoritmo 24 (ALGORITMOPROPOSTO): veja que a instrução contida nessa linha estabelece uma sequência para verificar os elementos de $L_{\text{indiceElementoEstruturante}}$. Entretanto, conceitualmente, as operações da morfologia matemática aplicada a imagens binárias trabalham com *conjuntos* e não *sequências*. Esse assunto foi levado em consideração para a elaboração do procedimento experimental, pois, alguma das implementações testadas pode ser mais vantajosa devido a tal ausência de aleatoriedade para a determinação da próxima coordenada a ser analisada, acarretando em tempos de execução tendenciosos para uma determinada combinação de imagem X e elementos de \mathcal{L} .

Foram realizados testes iniciais, de modo a determinar todas as sequências utilizadas de maneira alea-

tória. Entretanto, a execução desses testes foi verificada como errônea devido ao fato das implementações de cada um dos algoritmos, para os mesmos parâmetros de entrada, devolver imagens diferentes como resultado da operação desejada.

Apesar do autor acreditar que não haja diferenças significativas para o comportamento obtido para as curvas de tempo médio de execução, as implementações dos algoritmos estão preparadas para o caso em que o número d de dimensões das imagens sejam maiores que 2.

Apêndice A

Tabelas de Complexidade Computacional dos Algoritmos Apresentados

Neste apêndice são apresentados os valores de complexidade computacional calculados para uma execução completa de cada um dos algoritmos apresentados nos Capítulos 3 e 4. Para as análises de complexidade efetuadas, considere os símbolos definidos na Seção 2.5.

A.1 Complexidade Computacional dos Algoritmos da Seção 3.1

A Tabela A.1 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 2 (TRANSFORMADAEROSÃOSIMPLES). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X para cada elemento estruturante de \mathcal{L} e para cada iteração até o término do algoritmo, realizada na linha 12 ($O(m \cdot k \cdot [X])$) e a verificação de cada coordenada de I_{x-1} por todas as coordenadas do elemento estruturante de \mathcal{L} analisado, realizada na linha 13 ($O\left(\sum_{x=1}^{m \cdot k} |I_{x-1}| \cdot [L_x]\right)$). Quanto ao espaço computacional, as instruções que mais o consomem são as instanciações de imagens com a mesma caixa delimitadora de X , realizadas nas linhas 1 e 2 ($O([X])$).

Tabela A.1: Análise de Complexidade de TRANSFORMADAEROSÃOSIMPLES.

Linha	Tempo Computacional	Espaço Computacional
1–2	$O([X])$	$O([X])$
3	$O(1)$	$O(1)$
4	$O([X])$	
5	$O(X)$	
6–8	$O(m)$	
9–10	$O(m \cdot k)$	
11	$O(m \cdot k)$	$O(1)$
12	$O(m \cdot k \cdot [X])$	
13	$O\left(\sum_{x=1}^{m \cdot k} I_{x-1} \cdot [L_x]\right)$	
14	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	$O\left(\max_x \{ borda_{L_x}(I_{x-1}) \}\right)$
15–17	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	
18	$O(m \cdot k)$	
19	$O(1)$	
Total	$O\left(m \cdot k \cdot [X] + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot [L_x]\right)$	$O([X])$

A Tabela A.2 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 3 (TRANSFORMADADILATAÇÃOSIMPLES). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X para cada elemento estruturante de \mathcal{L} e para cada iteração até o término do algoritmo, realizada na linha 12 ($O(m \cdot k \cdot [X])$) e a verificação de cada coordenada de I_{x-1} por todas as coordenadas do elemento estruturante de \mathcal{L} analisado, realizada nas linhas 13 e 14 ($O\left(\sum_{x=1}^{m \cdot k} |I_{x-1}| \cdot [L_x]\right)$). Quanto ao espaço computacional, as instruções que mais o consomem são as instanciações de imagens com a mesma caixa delimitadora de X , realizadas nas linhas 1 e 2 ($O([X])$).

Cabe ressaltar que na Tabela A.2, a imagem I_0 equivale à imagem X e as imagens I_x equivalem às respectivas imagens $\delta_{L_x}(I_{x-1})$.

Tabela A.2: Análise de Complexidade de TRANSFORMADADILATAÇÃOSIMPLES.

Linha	Tempo Computacional	Espaço Computacional
1–2	$O([X])$	$O([X])$
3	$O(1)$	$O(1)$
4	$O([X])$	
5	$O(X)$	
6–8	$O(m)$	
9–10	$O(m \cdot k)$	
11	$O(m \cdot k)$	$O(1)$
12	$O(m \cdot k \cdot [X])$	
13–14	$O\left(\sum_{x=1}^{m \cdot k} I_{x-1} \cdot [L_x]\right)$	
15	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	$O\left(\max_x \{ borda_{L_x}(I_{x-1}) \cdot [L_x]\}\right)$
16–18	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	
19	$O(m \cdot k)$	
20	$O(1)$	
Total	$O\left(m \cdot k \cdot [X] + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot [L_x]\right)$	$O([X])$

A.2 Complexidade Computacional dos Algoritmos da Seção 3.2

Note que a complexidade computacional, apresentada através da Tabela A.3 tanto no quesito tempo quanto no quesito espaço do Algoritmo 6 (TRANSFORMADADUAL), é a mesma que a calculada nos Algoritmos 2 (TRANSFORMADAEROSÃOSIMPLES) e 3 (TRANSFORMADADILATAÇÃOSIMPLES), com a vantagem de efetuar ambos os operadores com um único algoritmo, determinando a operação desejada através de uma variável lógica.

Tabela A.3: Análise de Complexidade de TRANSFORMADADUAL.

Linha	Tempo Computacional	Espaço Computacional
1–2	$O([X])$	$O([X])$
3–4	$O(1)$	$O(1)$
5	$O([X])$	
6	$O(X)$	
7–9	$O(m)$	
10–11	$O(m \cdot k)$	
12	$O(m \cdot k)$	$O(1)$
13	$O(m \cdot k \cdot [X])$	
14, 16, 18	$O\left(\sum_{x=1}^{m \cdot k} I_{x-1} \right)$	
15, 17	$O\left(\sum_{x=1}^{m \cdot k} I_{x-1} \cdot [L_x]\right)$	
19	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	$O\left(\max_x \{ borda_{L_x}(I_{x-1}) \}\right)$
20–25	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	
26	$O(m \cdot k)$	
27	$O(1)$	
Total	$O\left(m \cdot k \cdot [X] + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot [L_x]\right)$	$O([X])$

A.3 Complexidade Computacional dos Algoritmos da Seção 3.3

A Tabela A.4 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 10 (LINEARIZEELEMENTOESTRUTURANTE). O tempo computacional é composto por múltiplos termos pelo fato de que a determinação de qual desses termos ser o maior dependerá dos elementos estruturantes fornecidos em \mathcal{L} . As instruções que mais consomem tempo são as relacionadas à instanciação da imagem *união*, realizada na linha 8 ($O\left(\left[\bigcup_{i=1}^k L_i\right]\right)$); à verificação de cada coordenada nas caixas delimitadoras dos elementos estruturantes de \mathcal{L} , realizada na linha 11 ($O\left(\sum_{i=1}^k [L_i]\right)$); e ao cálculo dos vetores relacionados a cada elemento estruturante de \mathcal{L} , realizado entre as linhas 26 e 28 ($O\left(k \cdot \left|\bigcup_{i=1}^k L_i\right|\right)$). Quanto ao espaço computacional, as instruções que mais o consomem são a instanciação da imagem *união*, realizada na linha 8 ($O\left(\left[\bigcup_{i=1}^k L_i\right]\right)$), e o armazenamento das coordenadas dos elementos de cada elemento estruturante de \mathcal{L} , realizado na linha 23 ($O\left(\sum_{i=1}^k |L_i|\right)$).

Tabela A.4: Análise de Complexidade de LINEARIZEELEMENTOESTRUTURANTE.

Linha	Tempo Computacional	Espaço Computacional
1 – 2	$O(k)$	$O(k)$
3 – 5	$O(1)$	$O(1)$
6 – 7	$O(k)$	
8	$O\left(\left[\bigcup_{i=1}^k L_i\right]\right)$	$O\left(\left[\bigcup_{i=1}^k L_i\right]\right)$
9 – 10	$O(k)$	
11	$O\left(\sum_{i=1}^k [L_i]\right)$	
12 – 13	$O\left(\sum_{i=1}^k L_i \right)$	
14 – 15	$O\left(\left \bigcup_{i=1}^k L_i\right \right)$	
16 – 17	$O\left(\left \bigcup_{i=1}^k L_i\right \right)$	$O\left(\left \bigcup_{i=1}^k L_i\right \right)$
18 – 21	$O\left(\left \bigcup_{i=1}^k L_i\right \right)$	
22	$O(k)$	
23	$O\left(\sum_{i=1}^k L_i \right)$	$O\left(\sum_{i=1}^k L_i \right)$
24 – 25	$O(k)$	$O(1)$
26 – 28	$O\left(k \cdot \left \bigcup_{i=1}^k L_i\right \right)$	
29 – 30	$O\left(\sum_{i=1}^k L_i \right)$	
31	$O(k)$	
32	$O(1)$	
Total	$O\left(\left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k [L_i] + k \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left(\left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k L_i \right)$

A Tabela A.5 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 11 (LINEARIZEIMAGEMLINEAR). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X , realizada nas linhas 2 e 3 ($O([X])$). Quanto ao espaço computacional, a instrução que mais o consome é o armazenamento das coordenadas de interesse de X , realizado na linha 4 ($O(|I_0|)$).

Tabela A.5: Análise de Complexidade de LINEARIZEIMAGEMLINEAR.

Linha	Tempo Computacional	Espaço Computacional
1	$O(1)$	$O(1)$
2 – 3	$O([X])$	
4	$O(I_0)$	$O(I_0)$
5	$O(1)$	
Total	$O([X])$	$O(I_0)$

A Tabela A.6 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 12 (ALGORITMO DUAL LINEAR). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X , realizada nas linhas 1, 2 e 5 ($O([X])$), e a verificação de cada

elemento de I_{x-1} pelos elementos do L_x analisado, realizada entre as linhas 17 e 25 $\left(O \left(\sum_{x=1}^{m \cdot k} |I_{x-1}| \cdot |L_x| \right) \right)$. Quanto ao espaço computacional, as instruções que mais o consomem são as instanciações de imagens com a mesma caixa delimitadora de X , realizadas nas linhas 1 e 2 $(O([X]))$.

Tabela A.6: *Análise de Complexidade de ALGORITMODUALINEAR.*

Linha	Tempo Computacional	Espaço Computacional
1–2	$O([X])$	$O([X])$
3–4	$O(1)$	$O(1)$
5	$O([X])$	
6	$O(X)$	
7–9	$O(m)$	
10–11	$O(m \cdot k)$	
12	$O(m \cdot k)$	$O(1)$
13–16	$O \left(\sum_{x=1}^{m \cdot k} I_{x-1} \right)$	
17–25	$O \left(\sum_{x=1}^{m \cdot k} I_{x-1} \cdot L_x \right)$	
26	$O \left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	$O \left(\max_x \{ borda_{L_x}(I_{x-1}) \} \right)$
27–33	$O \left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	
34	$O(m \cdot k)$	
35	$O(1)$	
Total	$O \left([X] + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot L_x \right)$	$O([X])$

Tabela A.7: *Análise de Complexidade de TRANSFORMADADUALINEAR.*

Linha	Tempo Computacional	Espaço Computacional
1	$O \left(\left[\bigcup_{i=1}^k L_i \right] + \sum_{i=1}^k [L_i] + k \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O \left(\left[\bigcup_{i=1}^k L_i \right] + \sum_{i=1}^k L_i \right)$
2	$O([X])$	$O(I_0)$
3	$O \left([X] + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot L_x \right)$	$O([X])$
Total	$O \left([X] + \left[\bigcup_{i=1}^k L_i \right] + \sum_{i=1}^k [L_i] + k \cdot \left \bigcup_{i=1}^k L_i \right + \sum_{x=1}^{m \cdot k} I_{x-1} \cdot L_x \right)$	$O \left([X] + \left[\bigcup_{i=1}^k L_i \right] + \sum_{i=1}^k L_i \right)$

Comparando os termos do tempo computacional de TRANSFORMADADUAL, apresentado na Tabela A.3, com os valores obtidos para TRANSFORMADADUALINEAR, apresentado na Tabela A.7, podemos verificar que a versão linear possui um custo atrelado à linearização dos dados

$\left(O \left([X] + \left[\bigcup_{i=1}^k L_i \right] + \sum_{i=1}^k [L_i] + k \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$, o qual é esperado que seja menor que o custo relacionado à determinação das coordenadas de interesse do primeiro algoritmo $(O(m \cdot k \cdot [X]))$. Já o custo de determinação da borda, como consequência do próprio processo de linearização, é menor em

TRANSFORMADADUALINEAR do que em TRANSFORMADADUAL

$$\left(O \left(\sum_{x=1}^{m \cdot k} |I_{x-1}| \cdot |L_x| \right) \right) < O \left(\sum_{x=1}^{m \cdot k} |I_{x-1}| \cdot [L_x] \right).$$

A.4 Complexidade Computacional dos Algoritmos da Seção 3.4

A Tabela A.8 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 16 (LINEARIZEBORDAIMAGEM). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X , realizada entre as linhas 4 e 6 ($O(\lceil X \rceil)$), e à verificação das coordenadas de interesse que pertencem à borda de I_0 de acordo com a união dos elementos de \mathcal{L} , realizada entre as linhas 9 e 16 ($O\left(|I_0| \cdot \left| \bigcup_{i=1}^k L_i \right|\right)$). Note que, como o algoritmo analisado possui um critério mais estrito que o aplicado em Algoritmo 11 (LINEARIZEIMAGEMLINEAR) para a inserção em *listaCoordenadas*, o tempo computacional para a execução do Algoritmo 16 (LINEARIZEBORDAIMAGEM) é maior. Quanto ao espaço computacional, a instrução que mais o consome é a instanciação da imagem binária para auxiliar o controle de quais coordenadas de interesse já entraram em *listaCoordenadas*, realizada na linha 4 ($O(\lceil X \rceil)$).

Tabela A.8: Análise de Complexidade de LINEARIZEBORDAIMAGEM.

Linha	Tempo Computacional	Espaço Computacional
1–3	$O(1)$	$O(1)$
4	$O(\lceil X \rceil)$	$O(\lceil X \rceil)$
5–6	$O(\lceil X \rceil)$	
7–8	$O(I_0)$	
9–16	$O\left(I_0 \cdot \left \bigcup_{i=1}^k L_i \right \right)$	
17	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i} (I_0) \right \right)$	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i} (I_0) \right \right)$
18	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i} (I_0) \right \right)$	
19	$O(1)$	
Total	$O\left(\lceil X \rceil + I_0 \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O(\lceil X \rceil)$

A Tabela A.9 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 17 (ATUALIZAÇÃOODUALLINEARDEBORDA). As instruções que mais consomem tempo são as relacionadas à fase de manutenção da invariante adotada para este operador, no qual deve-se inserir as coordenadas da borda de I_x de acordo com a união dos elementos de \mathcal{L} , realizada entre as linhas 8 e 21 ($O\left(\left| \text{borda}_{L_x}(I_{x-1}) \right| \cdot \left| \bigcup_{i=1}^k L_i \right|\right)$). Quanto ao espaço computacional, a instrução que mais o consome é o armazenamento das coordenadas reconhecidas como resultado da borda, as quais ainda não estavam em *listaCoordenadas*, realizado na linha 22 ($O\left(\left| \text{borda}_{\bigcup_{i=1}^k L_i} (I_x) \setminus \text{borda}_{\bigcup_{i=1}^k L_i} (I_{x-1}) \right|\right)$).

Tabela A.9: *Análise de Complexidade de ATUALIZAÇÃO DUAL LINEAR DE BORDA.*

Linha	Tempo Computacional	Espaço Computacional
1 – 7	$O(borda_{L_x}(I_{x-1}))$	
8 – 21	$O\left(borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	
22	$O\left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1})\right \right)$	$O\left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1})\right \right)$
23	$O\left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1})\right \right)$	
24	$O(1)$	
Total	$O\left(borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O\left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1})\right \right)$

A Tabela A.10 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 18 (ALGORITMO DUAL LINEAR DE BORDA). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X , realizada nas linhas 1, 2 e 5 ($O([X])$); à verificação dos elementos da borda de I_{x-1} pela união dos elementos de \mathcal{L} para determinar a borda de I_{x-1} pelo L_x analisado, realizada entre as linhas 16 e 24 $\left(O\left(\sum_{x=1}^{m \cdot k} \left|borda_{\bigcup_{i=1}^k L_i}(I_{x-1})\right| \cdot |L_x|\right)\right)$; e à manutenção da invariante aplicada nesse operador, a qual requer a inserção das coordenadas da borda de I_x de acordo com a união dos elementos de \mathcal{L} que não pertenciam previamente a *listaCoordenadas*, realizada na linha 27 $\left(O\left(\sum_{x=1}^{m \cdot k} |borda_{L_x}(I_{x-1})| \cdot \left| \bigcup_{i=1}^k L_i \right|\right)\right)$. Quanto ao espaço computacional, as instruções que mais o consomem são as instanciações de imagens com a mesma caixa delimitadora de X , realizadas nas linhas 1 e 2 ($O([X])$).

Tabela A.10: Análise de Complexidade de ALGORITMO DUAL LINEAR DE BORDA.

Linha	Tempo Computacional	Espaço Computacional
1 – 2	$O([X])$	$O([X])$
3 – 4	$O(1)$	$O(1)$
5	$O([X])$	
6	$O(X)$	
7 – 8	$O(m)$	
9 – 10	$O(m \cdot k)$	
11	$O(m \cdot k)$	$O(1)$
12 – 15	$O\left(\sum_{x=1}^{m \cdot k} \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \right)$	
16 – 24	$O\left(\sum_{x=1}^{m \cdot k} \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \cdot L_x \right)$	
25	$O\left(\sum_{x=1}^{m \cdot k} \text{borda}_{L_x}(I_{x-1}) \right)$	$O\left(\max_x \{ \text{borda}_{L_x}(I_{x-1}) \}\right)$
26	$O\left(\sum_{x=1}^{m \cdot k} \text{borda}_{L_x}(I_{x-1}) \right)$	
27	$O\left(\sum_{x=1}^{m \cdot k} \text{borda}_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O\left(\max_x \left\{ \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_x) \right \right\} \right)$
28	$O(1)$	
Total	$O\left([X] + \sum_{x=1}^{m \cdot k} \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \cdot L_x + \sum_{x=1}^{m \cdot k} \text{borda}_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O([X])$

Tabela A.11: Análise de Complexidade de TRANSFORMADA DUAL LINEAR DE BORDA.

Linha	Tempo Computacional	Espaço Computacional
1	$O\left(\left \bigcup_{i=1}^k L_i \right + \sum_{i=1}^k [L_i] + k \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O\left(\left \bigcup_{i=1}^k L_i \right + \sum_{i=1}^k L_i \right)$
2	$O\left([X] + I_0 \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O([X])$
3	$O\left([X] + \sum_{x=1}^{m \cdot k} \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \cdot L_x + \sum_{x=1}^{m \cdot k} \text{borda}_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O([X])$
Total	$O\left([X] + \left \bigcup_{i=1}^k L_i \right + \sum_{i=1}^k [L_i] + (k + I_0) \cdot \left \bigcup_{i=1}^k L_i \right + \sum_{x=1}^{m \cdot k} \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \cdot L_x + \sum_{x=1}^{m \cdot k} \text{borda}_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O\left([X] + \left \bigcup_{i=1}^k L_i \right + \sum_{i=1}^k L_i \right)$

Comparando o tempo computacional apresentado para o Algoritmo 9 (TRANSFORMADA DUAL LINEAR)

através da Tabela A.7 com o tempo computacional apresentado para o Algoritmo 15 (TRANSFORMADADUALLINEARDEBORDA) através da Tabela A.11, cabe ressaltar que o operador apresentado nesta seção é vantajoso se a quantidade de comparações relacionadas às bordas for menor que a quantidade de comparações relacionadas às coordenadas de interesse pertencentes a uma imagem binária I_{x-1} .

Apesar do desempenho do algoritmo estar atrelado à natureza da imagem binária, tal condição também foi utilizada como premissa para o desenvolvimento do algoritmo proposto, apresentado no Capítulo 4.

A.5 Complexidade Computacional dos Algoritmos do Capítulo 4

A Tabela A.12 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 22 (INICIALIZEMATRIZESPARGA). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X , realizada entre as linhas 4 e 7 ($O([X])$); à verificação das coordenadas de interesse que pertencem à borda de I_0 de acordo com a união dos elementos de \mathcal{L} , realizada entre as linhas 8 e 15 ($O(|I_0| \cdot \left| \bigcup_{i=1}^k L_i \right|)$); e à alocação dos nós de ligação, realizada na linha 20 ($O\left(\left| \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right| \cdot \left| \bigcup_{i=1}^k L_i \right|\right)$). Quanto ao espaço computacional, as instruções que mais consomem são as instanciações de imagem de mesmo tamanho que a caixa delimitadora de X , efetuadas nas linhas 4 e 5 ($O([X])$).

Tabela A.12: Análise de Complexidade de INICIALIZEMATRIZESPARGA.

Linha	Tempo Computacional	Espaço Computacional
1	$O\left(\left \bigcup_{i=1}^k L_i \right \right)$	$O\left(\left \bigcup_{i=1}^k L_i \right \right)$
2–3	$O(1)$	$O(1)$
4–5	$O([X])$	$O([X])$
6–7	$O([X])$	
8–15	$O\left(I_0 \cdot \left \bigcup_{i=1}^k L_i \right \right)$	
16–17, 19	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \right)$	
18	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \right)$	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \right)$
20	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O\left(\left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \cdot \left \bigcup_{i=1}^k L_i \right \right)$
21	$O(1)$	
Total	$O\left([X] + I_0 \cdot \left \bigcup_{i=1}^k L_i \right + \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O\left([X] + \left \text{borda}_{\bigcup_{i=1}^k L_i}(I_0) \right \cdot \left \bigcup_{i=1}^k L_i \right \right)$

A Tabela A.13 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 23 (ATUALIZAÇÃOALGORITMOPROPOSTO). As instruções que mais consomem tempo são as relacionadas à fase de manutenção da invariante adotada para este operador, no qual deve-se inserir as coordenadas da borda de I_x de acordo com a união dos elementos de \mathcal{L} , realizada entre as linhas 8 e 22 e na linha 25

$\left(O \left(|borda_{L_x}(I_{x-1})| \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$. Quanto ao espaço computacional, as instruções que mais o consomem são referentes ao armazenamento das coordenadas reconhecidas como resultado da borda, as quais ainda não estavam em *listaNósBorda*, realizado na linha 24 $\left(O \left(\left| borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right| \right) \right)$ e ao armazenamento dos novos nós de ligação, realizado na linha 26 $\left(O \left(|borda_{L_x}(I_{x-1})| \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$.

Tabela A.13: Análise de Complexidade de ATUALIZAÇÃOALGORITMOPROPOSTO.

Linha	Tempo Computacional	Espaço Computacional
1 –7	$O(borda_{L_x}(I_{x-1}))$	
8 –22, 25	$O \left(borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	
23	$O \left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \right)$	
24	$O \left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \right)$	$O \left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right \right)$
26	$O \left(borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O \left(borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$
27	$O(1)$	
Total	$O \left(borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$	$O \left(\left borda_{\bigcup_{i=1}^k L_i}(I_x) \setminus borda_{\bigcup_{i=1}^k L_i}(I_{x-1}) \right + borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i \right \right)$

A Tabela A.14 apresenta o tempo e o espaço computacionais requeridos pelo Algoritmo 24 (ALGORITMOPROPOSTO). As instruções que mais consomem tempo são as relacionadas à verificação de cada coordenada da caixa delimitadora de X , realizada nas linhas 1, 2 e 4 ($O([X])$), e à manutenção da invariante aplicada nesse operador, a qual requer a inserção das coordenadas da borda de I_x de acordo com a união dos elementos de \mathcal{L} que não pertenciam previamente a *listaCoordenadas* e a inserção de novos nós de ligação, realizadas entre as linhas 21 e 23 e na linha 25 $\left(O \left(\sum_{x=1}^{m \cdot k} |borda_{L_x}(I_{x-1})| \cdot \left| \bigcup_{i=1}^k L_i \right| \right) \right)$. Quanto ao espaço computacional, as instruções que mais o consomem são as instanciações de imagens com a mesma caixa delimitadora de X , realizadas nas linhas 1 e 2 ($O([X])$), e as alocações de nós de ligação, realizadas na linha 25 $\left(O \left(\max_x \left\{ \left| borda_{\bigcup_{i=1}^k L_i}(I_x) \right| \cdot \left| \bigcup_{i=1}^k L_i \right| \right\} \right) \right)$.

Tabela A.14: Análise de Complexidade de ALGORITMOPROPOSTO.

Linha	Tempo Computacional	Espaço Computacional
1 – 2	$O([X])$	$O([X])$
3	$O(1)$	$O(1)$
4	$O([X])$	
5	$O(X)$	
6 – 7	$O(m)$	
8 – 10	$O(m \cdot k)$	
11	$O(m \cdot k)$	$O(1)$
12 – 13	$O\left(\sum_{x=1}^{m \cdot k} L_x \right)$	
14 – 20	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	
21 – 23	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	
24	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \right)$	$O\left(\max_x \{ borda_{L_x}(I_{x-1}) \}\right)$
25	$O\left(\sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left(\max_x \left\{ \left borda_{\bigcup_{i=1}^k L_i}(I_x)\right \cdot \left \bigcup_{i=1}^k L_i\right \right\}\right)$
26	$O(1)$	
Total	$O\left([X] + \sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left([X] + \max_x \left\{ \left borda_{\bigcup_{i=1}^k L_i}(I_x)\right \cdot \left \bigcup_{i=1}^k L_i\right \right\}\right)$

Tabela A.15: Análise de Complexidade de TRANSFORMADAPROPOSTA.

Linha	Tempo Computacional	Espaço Computacional
1	$O\left(\left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k [L_i] + k \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left(\left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k L_i \right)$
2	$O\left([X] + I_0 \cdot \left \bigcup_{i=1}^k L_i\right + \left borda_{\bigcup_{i=1}^k L_i}(I_0)\right \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left([X] + \left borda_{\bigcup_{i=1}^k L_i}(I_0)\right \cdot \left \bigcup_{i=1}^k L_i\right \right)$
3	$O\left([X] + \sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left([X] + \max_x \left\{ \left borda_{\bigcup_{i=1}^k L_i}(I_x)\right \cdot \left \bigcup_{i=1}^k L_i\right \right\}\right)$
Total	$O\left([X] + \left[\bigcup_{i=1}^k L_i\right] + \sum_{i=1}^k [L_i] + (k + I_0) \cdot \left \bigcup_{i=1}^k L_i\right + \left borda_{\bigcup_{i=1}^k L_i}(I_0)\right \cdot \left \bigcup_{i=1}^k L_i\right + \sum_{x=1}^{m \cdot k} borda_{L_x}(I_{x-1}) \cdot \left \bigcup_{i=1}^k L_i\right \right)$	$O\left([X] + \max_x \left\{ \left borda_{\bigcup_{i=1}^k L_i}(I_x)\right \cdot \left \bigcup_{i=1}^k L_i\right \right\}\right)$

Para maiores detalhes da comparação de tempo computacional obtido para o Algoritmo 15 (TRANSFORMADADUALLINEARDEBORDA) e TRANSFORMADAPROPOSTA, veja a Subseção 6.3.1.

Apêndice B

Dados experimentais

Neste apêndice é apresentado um resumo dos dados obtidos para o procedimento experimental descrito no Capítulo 6.

B.1 Tempos de Execução Obtidos

B.1.1 Elementos estruturantes 3×3

Referente à Figura 6.1

Tabela B.1: Tempo de execução do Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.05e+07	4.10e+07	2.07e+07	2.06e+07	8.43e+05	8.19e+07	1.06e+08	8.23e+07	8.21e+07	1.63e+06
2	2.23e+07	3.94e+07	2.26e+07	2.25e+07	9.58e+05	8.91e+07	1.12e+08	8.96e+07	8.93e+07	1.46e+06
3	2.22e+07	4.43e+07	2.24e+07	2.23e+07	8.91e+05	8.78e+07	1.21e+08	8.82e+07	8.80e+07	1.67e+06
4	2.18e+07	4.34e+07	2.20e+07	2.18e+07	9.42e+05	8.67e+07	1.20e+08	8.73e+07	8.70e+07	1.70e+06
5	2.12e+07	3.36e+07	2.14e+07	2.13e+07	8.63e+05	8.40e+07	9.96e+07	8.45e+07	8.42e+07	1.30e+06
6	2.13e+07	3.34e+07	2.15e+07	2.14e+07	6.15e+05	8.44e+07	1.02e+08	8.49e+07	8.46e+07	1.29e+06
7	2.06e+07	3.03e+07	2.07e+07	2.07e+07	4.42e+05	8.23e+07	1.21e+08	8.28e+07	8.25e+07	2.05e+06
8	2.03e+07	3.01e+07	2.05e+07	2.04e+07	5.46e+05	8.15e+07	9.41e+07	8.19e+07	8.18e+07	9.39e+05
9	2.04e+07	4.01e+07	2.06e+07	2.05e+07	8.93e+05	8.13e+07	1.16e+08	8.18e+07	8.14e+07	2.13e+06
10	2.03e+07	4.05e+07	2.05e+07	2.04e+07	1.01e+06	8.08e+07	1.11e+08	8.13e+07	8.10e+07	1.74e+06
11	1.98e+07	2.77e+07	2.00e+07	1.99e+07	4.58e+05	7.86e+07	1.16e+08	7.90e+07	7.88e+07	1.48e+06
12	1.97e+07	2.91e+07	1.98e+07	1.98e+07	3.74e+05	7.78e+07	9.16e+07	7.81e+07	7.80e+07	6.14e+05
13	1.94e+07	2.97e+07	1.96e+07	1.95e+07	6.71e+05	7.70e+07	1.16e+08	7.76e+07	7.73e+07	2.10e+06
14	1.95e+07	3.51e+07	1.97e+07	1.96e+07	7.95e+05	7.77e+07	1.04e+08	7.82e+07	7.79e+07	1.68e+06
15	1.95e+07	3.11e+07	1.96e+07	1.96e+07	6.17e+05	7.77e+07	1.07e+08	7.83e+07	7.80e+07	1.98e+06

Tabela B.2: Tempo de execução do Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR DE BORDA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	5.70e+06	1.10e+07	5.81e+06	5.78e+06	2.16e+05	1.31e+07	2.46e+07	1.34e+07	1.34e+07	6.17e+05
2	6.85e+06	1.33e+07	6.98e+06	6.94e+06	3.44e+05	1.62e+07	2.27e+07	1.66e+07	1.65e+07	4.23e+05
3	7.51e+06	1.45e+07	7.65e+06	7.60e+06	3.31e+05	1.78e+07	3.36e+07	1.82e+07	1.81e+07	7.75e+05
4	7.58e+06	1.45e+07	7.70e+06	7.65e+06	3.28e+05	1.79e+07	2.82e+07	1.83e+07	1.82e+07	5.66e+05
5	7.56e+06	1.45e+07	7.70e+06	7.67e+06	2.69e+05	1.79e+07	2.70e+07	1.82e+07	1.82e+07	4.89e+05
6	7.54e+06	1.45e+07	7.70e+06	7.65e+06	4.49e+05	1.79e+07	3.13e+07	1.82e+07	1.81e+07	6.29e+05
7	7.55e+06	9.30e+06	7.64e+06	7.62e+06	8.76e+04	1.78e+07	3.37e+07	1.83e+07	1.82e+07	1.15e+06
8	7.52e+06	1.27e+07	7.67e+06	7.63e+06	3.04e+05	1.78e+07	2.54e+07	1.81e+07	1.81e+07	3.50e+05
9	7.54e+06	1.43e+07	7.67e+06	7.62e+06	4.32e+05	1.78e+07	3.36e+07	1.82e+07	1.81e+07	8.42e+05
10	7.53e+06	1.45e+07	7.68e+06	7.63e+06	4.45e+05	1.78e+07	3.35e+07	1.81e+07	1.81e+07	6.97e+05
11	7.53e+06	1.34e+07	7.67e+06	7.62e+06	2.89e+05	1.78e+07	3.35e+07	1.82e+07	1.81e+07	6.38e+05
12	7.53e+06	1.30e+07	7.68e+06	7.63e+06	3.58e+05	1.78e+07	2.37e+07	1.82e+07	1.82e+07	3.17e+05
13	7.54e+06	1.44e+07	7.68e+06	7.64e+06	2.88e+05	1.79e+07	2.59e+07	1.82e+07	1.82e+07	3.66e+05
14	7.53e+06	1.45e+07	7.69e+06	7.64e+06	4.14e+05	1.78e+07	2.58e+07	1.83e+07	1.82e+07	4.21e+05
15	7.50e+06	1.45e+07	7.68e+06	7.65e+06	3.18e+05	1.78e+07	3.36e+07	1.83e+07	1.82e+07	9.53e+05

Tabela B.3: Tempo de execução do Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	6.20e+06	1.12e+07	6.33e+06	6.29e+06	2.89e+05	1.38e+07	2.36e+07	1.42e+07	1.41e+07	4.40e+05
2	7.72e+06	1.31e+07	7.87e+06	7.83e+06	2.87e+05	1.75e+07	2.84e+07	1.79e+07	1.79e+07	6.34e+05
3	8.53e+06	1.54e+07	8.68e+06	8.64e+06	2.86e+05	1.93e+07	3.46e+07	1.98e+07	1.97e+07	7.48e+05
4	8.54e+06	1.55e+07	8.71e+06	8.67e+06	3.74e+05	1.92e+07	3.22e+07	1.96e+07	1.95e+07	5.23e+05
5	8.57e+06	1.56e+07	8.72e+06	8.69e+06	3.83e+05	1.94e+07	3.00e+07	1.98e+07	1.97e+07	6.52e+05
6	8.56e+06	1.53e+07	8.70e+06	8.67e+06	3.03e+05	1.94e+07	2.79e+07	1.96e+07	1.96e+07	3.15e+05
7	8.53e+06	1.14e+07	8.68e+06	8.66e+06	1.44e+05	1.93e+07	3.26e+07	1.99e+07	1.98e+07	9.67e+05
8	8.57e+06	1.15e+07	8.72e+06	8.69e+06	1.77e+05	1.92e+07	2.82e+07	1.96e+07	1.96e+07	4.71e+05
9	8.57e+06	1.54e+07	8.74e+06	8.68e+06	5.02e+05	1.94e+07	2.88e+07	1.98e+07	1.98e+07	3.86e+05
10	8.57e+06	1.54e+07	8.72e+06	8.69e+06	3.30e+05	1.95e+07	3.01e+07	1.98e+07	1.97e+07	5.98e+05
11	8.58e+06	1.38e+07	8.73e+06	8.68e+06	3.02e+05	1.93e+07	2.94e+07	1.98e+07	1.97e+07	5.99e+05
12	8.60e+06	1.55e+07	8.79e+06	8.77e+06	3.22e+05	1.94e+07	2.86e+07	2.00e+07	2.00e+07	4.53e+05
13	8.58e+06	1.48e+07	8.76e+06	8.73e+06	2.45e+05	1.94e+07	2.85e+07	1.98e+07	1.98e+07	3.74e+05
14	8.63e+06	1.38e+07	8.80e+06	8.74e+06	3.88e+05	1.94e+07	2.65e+07	1.99e+07	1.98e+07	4.17e+05
15	8.58e+06	1.54e+07	8.74e+06	8.70e+06	3.61e+05	1.94e+07	2.94e+07	1.99e+07	1.98e+07	6.18e+05

Referente à Figura 6.2a

Tabela B.4: Tempo de execução do Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEAR										
Transformada da Erosão						Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.34e+07	1.76e+07	1.34e+07	1.34e+07	2.36e+05	3.45e+07	4.38e+07	3.47e+07	3.46e+07	5.20e+05
2	1.47e+07	2.94e+07	1.49e+07	1.48e+07	7.82e+05	3.82e+07	5.04e+07	3.84e+07	3.82e+07	6.44e+05
3	1.46e+07	2.21e+07	1.47e+07	1.47e+07	4.85e+05	3.84e+07	5.81e+07	3.87e+07	3.85e+07	9.84e+05
4	1.43e+07	1.91e+07	1.44e+07	1.43e+07	3.04e+05	3.82e+07	5.51e+07	3.84e+07	3.83e+07	9.97e+05
5	1.40e+07	2.54e+07	1.41e+07	1.40e+07	5.49e+05	3.67e+07	4.92e+07	3.69e+07	3.67e+07	7.96e+05
6	1.40e+07	2.74e+07	1.41e+07	1.40e+07	6.15e+05	3.65e+07	4.71e+07	3.68e+07	3.66e+07	8.67e+05
7	1.35e+07	2.40e+07	1.36e+07	1.35e+07	4.44e+05	3.50e+07	5.19e+07	3.53e+07	3.51e+07	1.15e+06
8	1.33e+07	1.90e+07	1.34e+07	1.33e+07	3.06e+05	3.49e+07	4.18e+07	3.51e+07	3.50e+07	4.61e+05
9	1.34e+07	2.36e+07	1.35e+07	1.34e+07	7.19e+05	3.50e+07	5.59e+07	3.53e+07	3.51e+07	1.50e+06
10	1.34e+07	1.97e+07	1.35e+07	1.34e+07	3.95e+05	3.46e+07	5.01e+07	3.48e+07	3.47e+07	9.61e+05
11	1.30e+07	2.28e+07	1.31e+07	1.30e+07	4.52e+05	3.36e+07	4.45e+07	3.38e+07	3.37e+07	6.09e+05
12	1.29e+07	2.13e+07	1.30e+07	1.30e+07	4.48e+05	3.34e+07	4.52e+07	3.35e+07	3.34e+07	5.75e+05
13	1.27e+07	1.98e+07	1.28e+07	1.27e+07	5.26e+05	3.28e+07	4.04e+07	3.30e+07	3.29e+07	5.21e+05
14	1.27e+07	2.54e+07	1.28e+07	1.27e+07	8.26e+05	3.28e+07	4.80e+07	3.29e+07	3.29e+07	6.58e+05
15	1.27e+07	1.98e+07	1.28e+07	1.27e+07	3.49e+05	3.29e+07	4.90e+07	3.31e+07	3.29e+07	8.84e+05

Tabela B.5: Tempo de execução do Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEARDEBORDA										
Transformada da Erosão						Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.70e+06	2.94e+06	1.73e+06	1.72e+06	4.96e+04	2.51e+06	4.31e+06	2.55e+06	2.54e+06	9.30e+04
2	2.09e+06	4.04e+06	2.12e+06	2.11e+06	9.18e+04	3.20e+06	6.19e+06	3.25e+06	3.23e+06	1.67e+05
3	2.32e+06	4.42e+06	2.35e+06	2.34e+06	8.26e+04	3.57e+06	6.30e+06	3.61e+06	3.59e+06	1.42e+05
4	2.34e+06	4.54e+06	2.38e+06	2.37e+06	1.47e+05	3.62e+06	7.03e+06	3.67e+06	3.65e+06	1.45e+05
5	2.33e+06	4.48e+06	2.37e+06	2.36e+06	1.12e+05	3.61e+06	6.91e+06	3.65e+06	3.64e+06	1.52e+05
6	2.33e+06	4.48e+06	2.37e+06	2.35e+06	1.45e+05	3.59e+06	5.26e+06	3.64e+06	3.62e+06	1.03e+05
7	2.32e+06	4.44e+06	2.35e+06	2.34e+06	1.01e+05	3.56e+06	6.85e+06	3.62e+06	3.60e+06	2.10e+05
8	2.32e+06	4.40e+06	2.34e+06	2.34e+06	7.77e+04	3.57e+06	6.89e+06	3.62e+06	3.60e+06	1.64e+05
9	2.31e+06	4.51e+06	2.35e+06	2.34e+06	1.43e+05	3.56e+06	6.82e+06	3.62e+06	3.59e+06	2.13e+05
10	2.32e+06	4.23e+06	2.35e+06	2.34e+06	8.81e+04	3.57e+06	6.84e+06	3.62e+06	3.60e+06	1.62e+05
11	2.32e+06	4.48e+06	2.35e+06	2.34e+06	1.09e+05	3.57e+06	5.73e+06	3.61e+06	3.60e+06	7.05e+04
12	2.32e+06	4.54e+06	2.36e+06	2.34e+06	1.74e+05	3.58e+06	6.86e+06	3.62e+06	3.61e+06	1.47e+05
13	2.32e+06	4.47e+06	2.36e+06	2.35e+06	1.14e+05	3.58e+06	6.56e+06	3.63e+06	3.61e+06	1.56e+05
14	2.32e+06	4.49e+06	2.36e+06	2.35e+06	1.36e+05	3.58e+06	6.93e+06	3.64e+06	3.61e+06	2.36e+05
15	2.32e+06	4.45e+06	2.35e+06	2.34e+06	1.21e+05	3.57e+06	6.89e+06	3.62e+06	3.60e+06	2.10e+05

Tabela B.6: Tempo de execução do Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 3×3 .

TRANSFORMADAPROPOSTA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.69e+06	2.36e+06	1.72e+06	1.71e+06	2.94e+04	2.51e+06	4.35e+06	2.56e+06	2.55e+06	1.14e+05
2	2.11e+06	3.91e+06	2.15e+06	2.14e+06	5.72e+04	3.21e+06	5.96e+06	3.28e+06	3.26e+06	1.71e+05
3	2.33e+06	4.32e+06	2.37e+06	2.36e+06	1.11e+05	3.58e+06	6.68e+06	3.65e+06	3.64e+06	1.41e+05
4	2.33e+06	4.37e+06	2.37e+06	2.36e+06	1.25e+05	3.59e+06	5.66e+06	3.66e+06	3.65e+06	1.06e+05
5	2.33e+06	4.00e+06	2.38e+06	2.37e+06	1.16e+05	3.60e+06	6.65e+06	3.67e+06	3.65e+06	1.39e+05
6	2.33e+06	4.32e+06	2.37e+06	2.36e+06	1.01e+05	3.59e+06	5.44e+06	3.65e+06	3.64e+06	9.86e+04
7	2.33e+06	4.30e+06	2.37e+06	2.36e+06	8.84e+04	3.59e+06	6.64e+06	3.65e+06	3.63e+06	1.96e+05
8	2.32e+06	4.37e+06	2.37e+06	2.36e+06	9.36e+04	3.59e+06	6.66e+06	3.65e+06	3.64e+06	1.46e+05
9	2.33e+06	4.35e+06	2.37e+06	2.36e+06	1.16e+05	3.57e+06	6.62e+06	3.66e+06	3.64e+06	2.12e+05
10	2.33e+06	4.30e+06	2.37e+06	2.36e+06	8.18e+04	3.60e+06	6.65e+06	3.67e+06	3.65e+06	1.85e+05
11	2.34e+06	4.36e+06	2.38e+06	2.38e+06	9.62e+04	3.60e+06	4.49e+06	3.67e+06	3.66e+06	5.51e+04
12	2.33e+06	4.32e+06	2.39e+06	2.38e+06	1.45e+05	3.62e+06	6.66e+06	3.69e+06	3.67e+06	1.81e+05
13	2.36e+06	4.15e+06	2.40e+06	2.39e+06	9.37e+04	3.62e+06	6.69e+06	3.72e+06	3.69e+06	2.30e+05
14	2.36e+06	4.38e+06	2.41e+06	2.40e+06	1.24e+05	3.61e+06	6.65e+06	3.71e+06	3.68e+06	2.44e+05
15	2.35e+06	4.40e+06	2.39e+06	2.38e+06	1.11e+05	3.63e+06	6.77e+06	3.70e+06	3.68e+06	1.71e+05

Referente à Figura 6.2b

Tabela B.7: Tempo de execução do Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEAR										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.91e+05	2.04e+05	1.92e+05	1.92e+05	7.65e+02	4.46e+06	6.53e+06	4.49e+06	4.49e+06	7.27e+04
2	1.91e+05	3.80e+05	1.93e+05	1.92e+05	7.94e+03	4.83e+06	8.65e+06	4.88e+06	4.86e+06	1.86e+05
3	1.91e+05	3.77e+05	1.93e+05	1.92e+05	7.79e+03	4.88e+06	8.83e+06	4.92e+06	4.91e+06	1.47e+05
4	1.91e+05	3.72e+05	1.93e+05	1.92e+05	5.71e+03	4.82e+06	5.96e+06	4.87e+06	4.86e+06	5.89e+04
5	1.91e+05	3.84e+05	1.93e+05	1.92e+05	1.08e+04	4.77e+06	8.99e+06	4.84e+06	4.82e+06	1.90e+05
6	1.91e+05	3.79e+05	1.92e+05	1.92e+05	5.94e+03	4.75e+06	6.98e+06	4.79e+06	4.79e+06	9.92e+04
7	1.84e+05	3.84e+05	1.93e+05	1.92e+05	1.01e+04	4.81e+06	6.13e+06	4.84e+06	4.84e+06	5.04e+04
8	1.91e+05	3.84e+05	1.93e+05	1.92e+05	8.52e+03	4.80e+06	8.75e+06	4.84e+06	4.83e+06	1.85e+05
9	1.91e+05	3.79e+05	1.93e+05	1.93e+05	8.41e+03	4.82e+06	6.93e+06	4.85e+06	4.84e+06	1.07e+05
10	1.91e+05	3.80e+05	1.93e+05	1.92e+05	7.15e+03	4.79e+06	5.94e+06	4.84e+06	4.83e+06	3.78e+04
11	1.91e+05	3.83e+05	1.93e+05	1.92e+05	8.55e+03	4.78e+06	8.95e+06	4.85e+06	4.83e+06	2.27e+05
12	1.91e+05	3.83e+05	1.93e+05	1.93e+05	1.16e+04	4.80e+06	7.28e+06	4.84e+06	4.83e+06	1.12e+05
13	1.91e+05	3.83e+05	1.93e+05	1.92e+05	1.32e+04	4.79e+06	7.70e+06	4.84e+06	4.83e+06	1.40e+05
14	1.91e+05	3.87e+05	1.93e+05	1.92e+05	8.11e+03	4.78e+06	6.57e+06	4.84e+06	4.83e+06	9.66e+04
15	1.91e+05	3.59e+05	1.93e+05	1.92e+05	9.05e+03	4.79e+06	9.08e+06	4.86e+06	4.83e+06	2.33e+05

Tabela B.8: Tempo de execução do Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR DE BORDA										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.45e+05	2.56e+05	2.47e+05	2.47e+05	9.16e+02	4.21e+06	4.53e+06	4.25e+06	4.25e+06	2.25e+04
2	2.57e+05	5.08e+05	2.60e+05	2.59e+05	1.04e+04	5.27e+06	7.20e+06	5.32e+06	5.31e+06	9.23e+04
3	2.64e+05	5.16e+05	2.66e+05	2.66e+05	1.32e+04	5.76e+06	9.49e+06	5.83e+06	5.81e+06	1.89e+05
4	2.64e+05	5.12e+05	2.66e+05	2.65e+05	7.84e+03	5.72e+06	9.93e+06	5.80e+06	5.78e+06	1.96e+05
5	2.64e+05	5.32e+05	2.66e+05	2.66e+05	1.16e+04	5.73e+06	9.18e+06	5.80e+06	5.78e+06	1.74e+05
6	2.64e+05	3.67e+05	2.66e+05	2.65e+05	3.32e+03	5.69e+06	7.78e+06	5.75e+06	5.74e+06	9.55e+04
7	2.54e+05	5.34e+05	2.66e+05	2.66e+05	1.37e+04	5.70e+06	8.89e+06	5.78e+06	5.76e+06	1.62e+05
8	2.64e+05	5.32e+05	2.66e+05	2.66e+05	1.11e+04	5.70e+06	9.22e+06	5.77e+06	5.76e+06	1.76e+05
9	2.64e+05	5.06e+05	2.66e+05	2.66e+05	9.77e+03	5.73e+06	7.42e+06	5.79e+06	5.78e+06	8.69e+04
10	2.64e+05	5.30e+05	2.68e+05	2.67e+05	8.36e+03	5.73e+06	9.81e+06	5.78e+06	5.77e+06	1.32e+05
11	2.64e+05	5.29e+05	2.66e+05	2.65e+05	1.15e+04	5.70e+06	1.04e+07	5.80e+06	5.78e+06	2.53e+05
12	2.64e+05	5.34e+05	2.67e+05	2.66e+05	1.59e+04	5.72e+06	1.01e+07	5.79e+06	5.78e+06	2.16e+05
13	2.64e+05	5.17e+05	2.67e+05	2.66e+05	1.48e+04	5.72e+06	9.03e+06	5.79e+06	5.78e+06	1.45e+05
14	2.64e+05	5.34e+05	2.66e+05	2.66e+05	1.11e+04	5.73e+06	1.04e+07	5.80e+06	5.78e+06	2.12e+05
15	2.64e+05	5.35e+05	2.66e+05	2.66e+05	1.17e+04	5.73e+06	1.02e+07	5.80e+06	5.78e+06	2.38e+05

Tabela B.9: Tempo de execução do Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.64e+05	5.31e+05	2.66e+05	2.66e+05	8.43e+03	4.10e+06	6.49e+06	4.15e+06	4.15e+06	8.14e+04
2	2.97e+05	5.72e+05	3.00e+05	2.99e+05	1.35e+04	5.32e+06	7.77e+06	5.40e+06	5.39e+06	1.49e+05
3	3.12e+05	5.90e+05	3.17e+05	3.16e+05	1.43e+04	5.85e+06	9.20e+06	5.94e+06	5.92e+06	1.76e+05
4	3.13e+05	3.27e+05	3.16e+05	3.16e+05	1.42e+03	5.82e+06	9.80e+06	5.89e+06	5.88e+06	1.39e+05
5	3.13e+05	6.24e+05	3.17e+05	3.16e+05	1.85e+04	5.82e+06	8.31e+06	5.89e+06	5.87e+06	1.50e+05
6	3.14e+05	6.09e+05	3.17e+05	3.16e+05	1.09e+04	5.82e+06	9.92e+06	5.92e+06	5.91e+06	1.64e+05
7	3.12e+05	5.83e+05	3.16e+05	3.15e+05	1.41e+04	5.83e+06	1.04e+07	5.90e+06	5.89e+06	1.90e+05
8	3.13e+05	6.23e+05	3.17e+05	3.16e+05	1.68e+04	5.84e+06	1.04e+07	5.91e+06	5.90e+06	1.93e+05
9	3.14e+05	5.58e+05	3.18e+05	3.17e+05	1.04e+04	5.87e+06	7.27e+06	5.92e+06	5.91e+06	5.63e+04
10	3.15e+05	6.22e+05	3.18e+05	3.17e+05	1.50e+04	5.84e+06	9.75e+06	5.92e+06	5.91e+06	1.61e+05
11	3.15e+05	6.24e+05	3.19e+05	3.17e+05	1.96e+04	5.88e+06	1.03e+07	5.96e+06	5.95e+06	1.79e+05
12	3.13e+05	6.23e+05	3.19e+05	3.17e+05	2.16e+04	5.86e+06	9.95e+06	5.93e+06	5.90e+06	2.33e+05
13	3.13e+05	6.06e+05	3.17e+05	3.16e+05	1.46e+04	5.86e+06	8.42e+06	5.93e+06	5.92e+06	1.15e+05
14	3.13e+05	6.23e+05	3.17e+05	3.16e+05	1.34e+04	5.86e+06	1.03e+07	5.94e+06	5.92e+06	2.10e+05
15	3.13e+05	6.24e+05	3.18e+05	3.16e+05	2.24e+04	5.87e+06	8.41e+06	5.94e+06	5.93e+06	1.23e+05

Referente à Figura 6.2c

Tabela B.10: Tempo de execução do Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.12e+06	2.16e+06	1.13e+06	1.13e+06	5.43e+04	9.28e+05	1.84e+06	9.40e+05	9.35e+05	5.68e+04
2	1.12e+06	2.17e+06	1.14e+06	1.14e+06	6.51e+04	9.35e+05	1.84e+06	9.46e+05	9.42e+05	5.05e+04
3	1.11e+06	1.89e+06	1.14e+06	1.14e+06	3.20e+04	9.38e+05	1.76e+06	9.52e+05	9.46e+05	6.07e+04
4	1.13e+06	2.21e+06	1.15e+06	1.14e+06	7.47e+04	9.32e+05	1.83e+06	9.51e+05	9.46e+05	5.22e+04
5	1.11e+06	2.20e+06	1.14e+06	1.14e+06	4.67e+04	9.28e+05	1.80e+06	9.50e+05	9.47e+05	4.37e+04
6	1.13e+06	2.21e+06	1.15e+06	1.14e+06	4.99e+04	9.17e+05	1.86e+06	9.52e+05	9.47e+05	5.67e+04
7	1.11e+06	2.20e+06	1.15e+06	1.14e+06	6.07e+04	9.20e+05	1.86e+06	9.52e+05	9.46e+05	5.87e+04
8	1.12e+06	2.20e+06	1.15e+06	1.14e+06	7.49e+04	9.40e+05	1.87e+06	9.53e+05	9.46e+05	6.80e+04
9	1.11e+06	2.20e+06	1.15e+06	1.14e+06	7.55e+04	9.32e+05	1.73e+06	9.49e+05	9.47e+05	3.24e+04
10	1.13e+06	1.78e+06	1.14e+06	1.14e+06	2.54e+04	9.24e+05	1.69e+06	9.52e+05	9.47e+05	5.25e+04
11	1.13e+06	2.16e+06	1.15e+06	1.14e+06	7.92e+04	9.26e+05	1.86e+06	9.51e+05	9.47e+05	5.30e+04
12	1.12e+06	2.20e+06	1.15e+06	1.14e+06	6.80e+04	9.28e+05	1.84e+06	9.51e+05	9.47e+05	5.03e+04
13	1.12e+06	2.09e+06	1.15e+06	1.14e+06	5.09e+04	9.40e+05	1.80e+06	9.52e+05	9.47e+05	6.07e+04
14	1.13e+06	2.19e+06	1.15e+06	1.14e+06	7.48e+04	9.26e+05	1.85e+06	9.52e+05	9.47e+05	5.86e+04
15	1.12e+06	2.13e+06	1.15e+06	1.14e+06	6.42e+04	9.24e+05	1.77e+06	9.53e+05	9.47e+05	5.97e+04

Tabela B.11: Tempo de execução do Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR DE BORDA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.15e+06	4.03e+06	2.18e+06	2.17e+06	9.40e+04	1.97e+06	3.32e+06	1.99e+06	1.98e+06	6.99e+04
2	2.49e+06	4.82e+06	2.55e+06	2.54e+06	9.10e+04	2.33e+06	4.59e+06	2.36e+06	2.35e+06	1.20e+05
3	2.73e+06	5.36e+06	2.77e+06	2.75e+06	1.41e+05	2.51e+06	4.96e+06	2.57e+06	2.55e+06	1.85e+05
4	2.72e+06	5.37e+06	2.78e+06	2.76e+06	1.89e+05	2.53e+06	4.94e+06	2.56e+06	2.55e+06	1.31e+05
5	2.74e+06	5.20e+06	2.77e+06	2.76e+06	1.20e+05	2.53e+06	4.86e+06	2.56e+06	2.55e+06	1.19e+05
6	2.72e+06	5.33e+06	2.77e+06	2.75e+06	1.33e+05	2.51e+06	5.03e+06	2.56e+06	2.55e+06	1.60e+05
7	2.70e+06	5.40e+06	2.77e+06	2.76e+06	1.81e+05	2.52e+06	5.00e+06	2.57e+06	2.55e+06	1.91e+05
8	2.71e+06	5.30e+06	2.77e+06	2.75e+06	1.59e+05	2.50e+06	4.95e+06	2.56e+06	2.55e+06	1.45e+05
9	2.74e+06	5.27e+06	2.77e+06	2.76e+06	1.44e+05	2.52e+06	4.96e+06	2.56e+06	2.55e+06	1.26e+05
10	2.74e+06	5.36e+06	2.77e+06	2.76e+06	1.47e+05	2.51e+06	5.01e+06	2.56e+06	2.55e+06	1.36e+05
11	2.72e+06	5.37e+06	2.79e+06	2.76e+06	2.42e+05	2.51e+06	5.03e+06	2.56e+06	2.55e+06	1.20e+05
12	2.74e+06	4.54e+06	2.77e+06	2.76e+06	1.12e+05	2.51e+06	4.94e+06	2.56e+06	2.55e+06	1.30e+05
13	2.71e+06	5.33e+06	2.76e+06	2.76e+06	1.12e+05	2.50e+06	4.98e+06	2.57e+06	2.55e+06	1.90e+05
14	2.73e+06	5.31e+06	2.77e+06	2.76e+06	1.96e+05	2.50e+06	4.86e+06	2.56e+06	2.55e+06	1.35e+05
15	2.73e+06	4.18e+06	2.76e+06	2.76e+06	7.27e+04	2.52e+06	4.99e+06	2.56e+06	2.55e+06	1.29e+05

Tabela B.12: Tempo de execução do Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.50e+06	4.15e+06	2.54e+06	2.53e+06	9.16e+04	2.30e+06	4.34e+06	2.35e+06	2.34e+06	9.79e+04
2	3.14e+06	5.82e+06	3.18e+06	3.16e+06	1.65e+05	2.91e+06	5.16e+06	2.97e+06	2.96e+06	9.18e+04
3	3.46e+06	6.37e+06	3.54e+06	3.52e+06	1.53e+05	3.28e+06	5.80e+06	3.33e+06	3.31e+06	1.61e+05
4	3.46e+06	6.37e+06	3.53e+06	3.52e+06	1.40e+05	3.27e+06	5.07e+06	3.32e+06	3.30e+06	1.25e+05
5	3.48e+06	6.43e+06	3.52e+06	3.51e+06	1.41e+05	3.25e+06	5.90e+06	3.31e+06	3.30e+06	1.54e+05
6	3.46e+06	6.35e+06	3.51e+06	3.50e+06	1.36e+05	3.28e+06	6.00e+06	3.33e+06	3.31e+06	1.36e+05
7	3.45e+06	6.34e+06	3.52e+06	3.50e+06	1.75e+05	3.27e+06	6.04e+06	3.32e+06	3.31e+06	1.45e+05
8	3.48e+06	6.44e+06	3.54e+06	3.51e+06	2.37e+05	3.25e+06	5.98e+06	3.33e+06	3.32e+06	1.49e+05
9	3.48e+06	5.12e+06	3.53e+06	3.52e+06	7.49e+04	3.30e+06	6.07e+06	3.35e+06	3.32e+06	1.89e+05
10	3.48e+06	6.33e+06	3.54e+06	3.52e+06	1.46e+05	3.28e+06	6.02e+06	3.34e+06	3.32e+06	1.71e+05
11	3.48e+06	6.38e+06	3.53e+06	3.51e+06	1.91e+05	3.27e+06	6.01e+06	3.32e+06	3.30e+06	1.59e+05
12	3.46e+06	6.12e+06	3.53e+06	3.52e+06	1.19e+05	3.26e+06	5.98e+06	3.31e+06	3.30e+06	1.36e+05
13	3.49e+06	6.39e+06	3.54e+06	3.52e+06	1.92e+05	3.26e+06	5.89e+06	3.32e+06	3.30e+06	1.74e+05
14	3.48e+06	5.50e+06	3.54e+06	3.53e+06	9.53e+04	3.22e+06	6.00e+06	3.29e+06	3.28e+06	1.25e+05
15	3.48e+06	5.57e+06	3.54e+06	3.53e+06	1.16e+05	3.27e+06	6.05e+06	3.31e+06	3.30e+06	1.39e+05

Referente à Figura 6.2d**Tabela B.13:** Tempo de execução do Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	5.87e+06	1.13e+07	5.92e+06	5.89e+06	2.90e+05	5.12e+07	6.37e+07	5.15e+07	5.13e+07	1.10e+06
2	6.28e+06	9.27e+06	6.32e+06	6.31e+06	1.40e+05	5.45e+07	6.43e+07	5.48e+07	5.46e+07	7.58e+05
3	6.23e+06	9.94e+06	6.27e+06	6.26e+06	1.63e+05	5.36e+07	7.19e+07	5.39e+07	5.37e+07	1.09e+06
4	6.13e+06	1.19e+07	6.19e+06	6.15e+06	3.32e+05	5.29e+07	7.27e+07	5.31e+07	5.30e+07	1.03e+06
5	5.92e+06	1.17e+07	5.96e+06	5.94e+06	2.53e+05	5.12e+07	7.33e+07	5.15e+07	5.13e+07	1.21e+06
6	5.97e+06	1.08e+07	6.03e+06	6.00e+06	2.97e+05	5.17e+07	5.80e+07	5.20e+07	5.18e+07	8.47e+05
7	5.75e+06	1.11e+07	5.80e+06	5.76e+06	2.88e+05	5.09e+07	6.55e+07	5.12e+07	5.10e+07	8.66e+05
8	5.69e+06	1.12e+07	5.75e+06	5.71e+06	3.25e+05	5.01e+07	6.73e+07	5.04e+07	5.02e+07	1.16e+06
9	5.67e+06	1.08e+07	5.73e+06	5.69e+06	3.40e+05	4.97e+07	5.60e+07	5.00e+07	4.98e+07	5.24e+05
10	5.61e+06	9.60e+06	5.66e+06	5.65e+06	1.70e+05	4.91e+07	7.51e+07	4.94e+07	4.92e+07	1.51e+06
11	5.49e+06	7.60e+06	5.51e+06	5.51e+06	9.39e+04	4.77e+07	5.87e+07	4.80e+07	4.78e+07	6.56e+05
12	5.44e+06	8.59e+06	5.51e+06	5.49e+06	1.78e+05	4.72e+07	6.00e+07	4.74e+07	4.73e+07	9.13e+05
13	5.39e+06	1.03e+07	5.44e+06	5.42e+06	2.12e+05	4.67e+07	6.55e+07	4.71e+07	4.68e+07	1.38e+06
14	5.40e+06	1.08e+07	5.45e+06	5.42e+06	3.08e+05	4.76e+07	6.80e+07	4.78e+07	4.76e+07	1.06e+06
15	5.40e+06	1.08e+07	5.45e+06	5.42e+06	2.89e+05	4.75e+07	7.08e+07	4.78e+07	4.76e+07	1.16e+06

Tabela B.14: Tempo de execução do Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR DE BORDA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.52e+06	2.99e+06	1.56e+06	1.55e+06	9.57e+04	3.58e+06	6.89e+06	3.63e+06	3.61e+06	1.58e+05
2	1.88e+06	3.54e+06	1.91e+06	1.91e+06	7.78e+04	4.46e+06	8.57e+06	4.56e+06	4.55e+06	1.54e+05
3	2.08e+06	4.07e+06	2.13e+06	2.12e+06	9.84e+04	4.96e+06	9.25e+06	5.07e+06	5.05e+06	1.85e+05
4	2.10e+06	4.06e+06	2.14e+06	2.13e+06	7.58e+04	5.01e+06	6.86e+06	5.08e+06	5.07e+06	7.21e+04
5	2.10e+06	4.07e+06	2.14e+06	2.13e+06	1.12e+05	5.02e+06	9.62e+06	5.11e+06	5.08e+06	2.76e+05
6	2.10e+06	4.03e+06	2.14e+06	2.13e+06	8.61e+04	5.00e+06	9.49e+06	5.10e+06	5.07e+06	2.05e+05
7	2.09e+06	4.02e+06	2.12e+06	2.11e+06	1.08e+05	5.00e+06	9.55e+06	5.09e+06	5.05e+06	2.77e+05
8	2.08e+06	4.01e+06	2.12e+06	2.12e+06	9.34e+04	4.99e+06	9.49e+06	5.07e+06	5.05e+06	1.72e+05
9	2.07e+06	4.03e+06	2.13e+06	2.11e+06	1.23e+05	4.97e+06	7.25e+06	5.04e+06	5.02e+06	1.10e+05
10	2.08e+06	4.01e+06	2.13e+06	2.11e+06	1.07e+05	4.97e+06	7.52e+06	5.07e+06	5.05e+06	1.51e+05
11	2.08e+06	3.56e+06	2.12e+06	2.11e+06	6.59e+04	4.98e+06	8.26e+06	5.05e+06	5.03e+06	1.71e+05
12	2.06e+06	4.04e+06	2.13e+06	2.12e+06	8.95e+04	5.00e+06	8.73e+06	5.08e+06	5.06e+06	1.78e+05
13	2.08e+06	4.01e+06	2.13e+06	2.12e+06	9.47e+04	4.98e+06	9.03e+06	5.07e+06	5.05e+06	1.38e+05
14	2.08e+06	4.02e+06	2.13e+06	2.12e+06	1.29e+05	4.99e+06	9.51e+06	5.07e+06	5.06e+06	1.83e+05
15	2.08e+06	4.02e+06	2.13e+06	2.11e+06	1.33e+05	4.99e+06	8.40e+06	5.06e+06	5.05e+06	1.46e+05

Tabela B.15: Tempo de execução do Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.57e+06	2.92e+06	1.61e+06	1.60e+06	9.02e+04	3.64e+06	6.72e+06	3.70e+06	3.69e+06	1.43e+05
2	1.96e+06	3.60e+06	2.00e+06	1.99e+06	6.97e+04	4.60e+06	7.66e+06	4.73e+06	4.71e+06	1.70e+05
3	2.16e+06	3.94e+06	2.20e+06	2.19e+06	8.90e+04	5.06e+06	8.11e+06	5.20e+06	5.18e+06	1.75e+05
4	2.15e+06	3.23e+06	2.19e+06	2.19e+06	4.95e+04	5.08e+06	9.00e+06	5.21e+06	5.20e+06	1.62e+05
5	2.16e+06	3.96e+06	2.21e+06	2.19e+06	1.29e+05	5.11e+06	9.31e+06	5.24e+06	5.21e+06	2.03e+05
6	2.16e+06	3.92e+06	2.21e+06	2.20e+06	9.31e+04	5.10e+06	9.24e+06	5.24e+06	5.22e+06	1.63e+05
7	2.15e+06	3.92e+06	2.19e+06	2.19e+06	7.41e+04	5.08e+06	9.27e+06	5.19e+06	5.17e+06	2.09e+05
8	2.16e+06	3.93e+06	2.20e+06	2.19e+06	7.25e+04	5.11e+06	9.25e+06	5.22e+06	5.20e+06	1.87e+05
9	2.15e+06	3.96e+06	2.20e+06	2.19e+06	8.89e+04	5.11e+06	7.87e+06	5.21e+06	5.19e+06	1.11e+05
10	2.16e+06	3.93e+06	2.21e+06	2.19e+06	1.25e+05	5.11e+06	9.38e+06	5.25e+06	5.22e+06	2.45e+05
11	2.16e+06	2.79e+06	2.20e+06	2.20e+06	3.29e+04	5.09e+06	9.31e+06	5.22e+06	5.19e+06	2.03e+05
12	2.16e+06	3.94e+06	2.21e+06	2.20e+06	1.00e+05	5.14e+06	6.32e+06	5.25e+06	5.24e+06	7.67e+04
13	2.16e+06	3.99e+06	2.21e+06	2.20e+06	8.33e+04	5.11e+06	9.30e+06	5.23e+06	5.21e+06	1.80e+05
14	2.17e+06	4.06e+06	2.22e+06	2.21e+06	1.34e+05	5.15e+06	9.33e+06	5.25e+06	5.22e+06	1.89e+05
15	2.17e+06	4.03e+06	2.22e+06	2.21e+06	1.15e+05	5.15e+06	9.32e+06	5.26e+06	5.24e+06	1.82e+05

B.1.2 Elementos estruturantes 5×5

Referente à Figura 6.1

Tabela B.16: Tempo de execução do Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR										
Transformada da Erosão						Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.34e+07	3.89e+07	2.35e+07	2.34e+07	7.58e+05	9.82e+07	1.33e+08	9.88e+07	9.84e+07	1.81e+06
2	2.41e+07	3.39e+07	2.42e+07	2.41e+07	7.42e+05	9.83e+07	1.13e+08	9.90e+07	9.86e+07	1.38e+06
3	2.36e+07	3.96e+07	2.38e+07	2.37e+07	9.59e+05	9.65e+07	1.23e+08	9.71e+07	9.68e+07	1.51e+06
4	2.28e+07	3.72e+07	2.29e+07	2.28e+07	6.23e+05	9.19e+07	1.22e+08	9.25e+07	9.22e+07	1.85e+06
5	2.25e+07	2.90e+07	2.27e+07	2.26e+07	3.98e+05	9.04e+07	1.07e+08	9.08e+07	9.06e+07	1.02e+06
6	2.38e+07	3.76e+07	2.40e+07	2.39e+07	7.93e+05	9.66e+07	1.31e+08	9.73e+07	9.68e+07	2.04e+06
7	2.37e+07	3.86e+07	2.39e+07	2.38e+07	7.56e+05	9.65e+07	1.18e+08	9.70e+07	9.67e+07	1.33e+06
8	2.40e+07	4.64e+07	2.42e+07	2.41e+07	8.03e+05	9.74e+07	1.51e+08	9.81e+07	9.76e+07	3.14e+06
9	2.37e+07	3.99e+07	2.40e+07	2.38e+07	8.32e+05	9.64e+07	1.27e+08	9.71e+07	9.67e+07	1.81e+06
10	2.36e+07	3.22e+07	2.37e+07	2.36e+07	4.42e+05	9.15e+07	1.24e+08	9.57e+07	9.54e+07	1.67e+06
11	2.35e+07	4.03e+07	2.37e+07	2.35e+07	9.27e+05	9.44e+07	1.23e+08	9.51e+07	9.47e+07	2.25e+06
12	2.34e+07	4.05e+07	2.36e+07	2.35e+07	8.84e+05	9.43e+07	1.19e+08	9.48e+07	9.45e+07	1.75e+06
13	2.31e+07	4.50e+07	2.34e+07	2.32e+07	1.15e+06	9.29e+07	1.26e+08	9.34e+07	9.30e+07	2.16e+06
14	2.33e+07	3.75e+07	2.35e+07	2.34e+07	6.01e+05	9.37e+07	1.16e+08	9.42e+07	9.39e+07	1.55e+06
15	2.29e+07	4.12e+07	2.32e+07	2.30e+07	1.16e+06	8.79e+07	1.24e+08	9.23e+07	9.19e+07	2.27e+06

Tabela B.17: Tempo de execução do Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR DE BORDA										
Transformada da Erosão						Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	9.19e+06	1.82e+07	9.32e+06	9.25e+06	5.63e+05	2.27e+07	4.44e+07	2.30e+07	2.29e+07	1.27e+06
2	1.27e+07	2.44e+07	1.29e+07	1.28e+07	6.33e+05	3.18e+07	4.83e+07	3.21e+07	3.20e+07	9.60e+05
3	1.34e+07	1.73e+07	1.35e+07	1.35e+07	2.49e+05	3.36e+07	4.33e+07	3.39e+07	3.38e+07	6.58e+05
4	1.39e+07	2.67e+07	1.41e+07	1.40e+07	5.49e+05	3.47e+07	6.25e+07	3.52e+07	3.51e+07	1.26e+06
5	1.39e+07	1.68e+07	1.40e+07	1.40e+07	2.29e+05	3.47e+07	4.24e+07	3.51e+07	3.50e+07	5.65e+05
6	1.39e+07	2.27e+07	1.41e+07	1.40e+07	4.69e+05	3.48e+07	5.24e+07	3.53e+07	3.51e+07	1.27e+06
7	1.39e+07	2.74e+07	1.41e+07	1.41e+07	9.48e+05	3.48e+07	4.06e+07	3.52e+07	3.51e+07	4.63e+05
8	1.39e+07	2.42e+07	1.41e+07	1.40e+07	4.85e+05	3.48e+07	6.83e+07	3.53e+07	3.51e+07	1.52e+06
9	1.39e+07	2.48e+07	1.41e+07	1.40e+07	6.80e+05	3.48e+07	5.19e+07	3.54e+07	3.51e+07	1.57e+06
10	1.39e+07	2.74e+07	1.41e+07	1.41e+07	7.08e+05	3.33e+07	4.77e+07	3.52e+07	3.50e+07	7.41e+05
11	1.39e+07	2.69e+07	1.41e+07	1.41e+07	5.17e+05	3.49e+07	5.24e+07	3.53e+07	3.52e+07	9.26e+05
12	1.39e+07	2.55e+07	1.41e+07	1.41e+07	5.51e+05	3.49e+07	6.40e+07	3.54e+07	3.52e+07	1.76e+06
13	1.39e+07	2.75e+07	1.41e+07	1.41e+07	5.38e+05	3.48e+07	5.19e+07	3.53e+07	3.52e+07	1.29e+06
14	1.40e+07	2.74e+07	1.41e+07	1.41e+07	5.82e+05	3.49e+07	5.78e+07	3.54e+07	3.52e+07	1.50e+06
15	1.39e+07	2.14e+07	1.41e+07	1.41e+07	5.08e+05	3.33e+07	6.13e+07	3.53e+07	3.51e+07	1.58e+06

Tabela B.18: Tempo de execução do Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 5×5 .

TRANSFORMADAPROPOSTA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.16e+07	2.09e+07	1.18e+07	1.17e+07	4.16e+05	2.71e+07	3.98e+07	2.75e+07	2.74e+07	7.34e+05
2	1.75e+07	3.00e+07	1.78e+07	1.77e+07	6.04e+05	3.99e+07	5.54e+07	4.05e+07	4.04e+07	9.68e+05
3	1.89e+07	3.40e+07	1.92e+07	1.91e+07	7.75e+05	4.33e+07	6.07e+07	4.38e+07	4.37e+07	7.60e+05
4	1.98e+07	3.50e+07	2.01e+07	2.00e+07	8.44e+05	4.54e+07	6.52e+07	4.63e+07	4.61e+07	1.25e+06
5	1.98e+07	3.42e+07	2.01e+07	2.00e+07	6.73e+05	4.49e+07	6.21e+07	4.58e+07	4.57e+07	1.11e+06
6	1.98e+07	2.83e+07	2.02e+07	2.01e+07	6.06e+05	4.50e+07	6.15e+07	4.58e+07	4.57e+07	8.76e+05
7	1.98e+07	2.44e+07	2.01e+07	2.01e+07	3.08e+05	4.48e+07	6.90e+07	4.56e+07	4.54e+07	1.10e+06
8	1.98e+07	3.55e+07	2.02e+07	2.01e+07	5.75e+05	4.54e+07	7.00e+07	4.63e+07	4.61e+07	1.38e+06
9	1.98e+07	2.78e+07	2.02e+07	2.01e+07	4.82e+05	4.53e+07	5.20e+07	4.61e+07	4.60e+07	6.32e+05
10	1.92e+07	3.09e+07	2.01e+07	2.00e+07	5.37e+05	4.36e+07	6.11e+07	4.60e+07	4.59e+07	9.34e+05
11	1.98e+07	3.55e+07	2.02e+07	2.01e+07	9.35e+05	4.57e+07	6.53e+07	4.63e+07	4.62e+07	9.89e+05
12	1.99e+07	3.05e+07	2.02e+07	2.02e+07	5.42e+05	4.55e+07	6.46e+07	4.62e+07	4.60e+07	1.16e+06
13	1.99e+07	3.14e+07	2.02e+07	2.01e+07	7.69e+05	4.56e+07	6.14e+07	4.64e+07	4.62e+07	1.16e+06
14	1.98e+07	3.55e+07	2.02e+07	2.01e+07	6.51e+05	4.55e+07	8.09e+07	4.63e+07	4.61e+07	1.54e+06
15	1.93e+07	3.58e+07	2.02e+07	2.01e+07	9.32e+05	4.39e+07	7.40e+07	4.63e+07	4.61e+07	1.63e+06

Referente à Figura 6.2a

Tabela B.19: Tempo de execução do Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEAR										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.56e+07	2.39e+07	1.57e+07	1.57e+07	4.65e+05	4.11e+07	5.58e+07	4.14e+07	4.12e+07	1.05e+06
2	1.62e+07	2.77e+07	1.63e+07	1.62e+07	6.50e+05	4.20e+07	6.45e+07	4.23e+07	4.20e+07	1.30e+06
3	1.58e+07	2.83e+07	1.60e+07	1.59e+07	8.04e+05	4.12e+07	6.38e+07	4.15e+07	4.13e+07	1.56e+06
4	1.53e+07	2.40e+07	1.54e+07	1.53e+07	5.61e+05	3.92e+07	4.98e+07	3.93e+07	3.92e+07	6.69e+05
5	1.51e+07	2.31e+07	1.52e+07	1.52e+07	5.47e+05	3.87e+07	5.03e+07	3.89e+07	3.88e+07	7.42e+05
6	1.61e+07	2.68e+07	1.62e+07	1.61e+07	4.93e+05	4.15e+07	5.36e+07	4.17e+07	4.15e+07	6.65e+05
7	1.60e+07	2.59e+07	1.62e+07	1.61e+07	7.30e+05	4.14e+07	5.23e+07	4.16e+07	4.14e+07	7.69e+05
8	1.63e+07	3.23e+07	1.64e+07	1.63e+07	9.27e+05	4.20e+07	6.83e+07	4.22e+07	4.20e+07	1.27e+06
9	1.60e+07	2.58e+07	1.62e+07	1.61e+07	5.54e+05	4.14e+07	7.28e+07	4.17e+07	4.14e+07	1.48e+06
10	1.59e+07	2.61e+07	1.61e+07	1.59e+07	6.61e+05	4.10e+07	5.29e+07	4.11e+07	4.10e+07	6.94e+05
11	1.58e+07	2.61e+07	1.59e+07	1.59e+07	6.02e+05	4.05e+07	5.29e+07	4.08e+07	4.06e+07	9.49e+05
12	1.58e+07	2.11e+07	1.59e+07	1.58e+07	3.30e+05	4.05e+07	5.42e+07	4.07e+07	4.06e+07	8.29e+05
13	1.56e+07	2.50e+07	1.56e+07	1.56e+07	4.98e+05	3.98e+07	5.66e+07	4.00e+07	3.99e+07	9.74e+05
14	1.57e+07	2.66e+07	1.58e+07	1.58e+07	4.95e+05	4.03e+07	5.48e+07	4.05e+07	4.03e+07	9.54e+05
15	1.54e+07	2.23e+07	1.55e+07	1.54e+07	6.41e+05	3.94e+07	5.01e+07	3.96e+07	3.95e+07	7.17e+05

Tabela B.20: Tempo de execução do Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEARDEBORDA										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	3.08e+06	4.32e+06	3.12e+06	3.11e+06	8.88e+04	4.87e+06	9.62e+06	4.92e+06	4.89e+06	2.48e+05
2	4.42e+06	8.74e+06	4.48e+06	4.45e+06	2.78e+05	7.15e+06	1.42e+07	7.24e+06	7.18e+06	4.34e+05
3	4.66e+06	9.26e+06	4.73e+06	4.70e+06	2.66e+05	7.58e+06	1.50e+07	7.66e+06	7.62e+06	3.51e+05
4	4.87e+06	9.57e+06	4.92e+06	4.89e+06	2.53e+05	7.92e+06	1.56e+07	8.00e+06	7.96e+06	3.45e+05
5	4.87e+06	9.41e+06	4.91e+06	4.89e+06	2.03e+05	7.92e+06	1.25e+07	8.00e+06	7.97e+06	2.40e+05
6	4.88e+06	9.41e+06	4.93e+06	4.91e+06	1.83e+05	7.95e+06	1.30e+07	8.02e+06	7.99e+06	2.61e+05
7	4.88e+06	6.94e+06	4.93e+06	4.92e+06	1.09e+05	7.96e+06	1.07e+07	8.02e+06	8.00e+06	1.83e+05
8	4.88e+06	9.65e+06	4.95e+06	4.91e+06	3.31e+05	7.95e+06	1.57e+07	8.04e+06	7.99e+06	4.01e+05
9	4.88e+06	9.60e+06	4.95e+06	4.91e+06	3.21e+05	7.95e+06	1.57e+07	8.05e+06	7.99e+06	5.15e+05
10	4.87e+06	9.49e+06	4.95e+06	4.91e+06	2.89e+05	7.95e+06	1.42e+07	8.05e+06	7.99e+06	4.02e+05
11	4.90e+06	9.66e+06	4.95e+06	4.93e+06	2.45e+05	7.99e+06	1.01e+07	8.05e+06	8.03e+06	1.34e+05
12	4.89e+06	8.78e+06	4.94e+06	4.93e+06	1.73e+05	7.99e+06	1.32e+07	8.07e+06	8.03e+06	2.74e+05
13	4.89e+06	9.66e+06	4.96e+06	4.93e+06	3.04e+05	7.97e+06	1.32e+07	8.06e+06	8.01e+06	3.81e+05
14	4.90e+06	9.59e+06	4.97e+06	4.93e+06	3.07e+05	7.98e+06	1.58e+07	8.08e+06	8.02e+06	4.97e+05
15	4.89e+06	9.47e+06	4.94e+06	4.92e+06	1.87e+05	7.96e+06	1.02e+07	8.03e+06	8.01e+06	1.28e+05

Tabela B.21: Tempo de execução do Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 5×5 .

TRANSFORMADAPROPOSTA										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	3.26e+06	6.10e+06	3.32e+06	3.30e+06	1.66e+05	5.15e+06	8.90e+06	5.23e+06	5.21e+06	2.03e+05
2	4.92e+06	9.06e+06	4.99e+06	4.97e+06	2.15e+05	7.95e+06	1.47e+07	8.10e+06	8.04e+06	4.03e+05
3	5.30e+06	9.71e+06	5.39e+06	5.37e+06	2.31e+05	8.61e+06	1.59e+07	8.80e+06	8.73e+06	5.71e+05
4	5.56e+06	1.02e+07	5.64e+06	5.62e+06	2.43e+05	9.08e+06	1.68e+07	9.26e+06	9.22e+06	3.33e+05
5	5.57e+06	9.01e+06	5.65e+06	5.64e+06	1.57e+05	9.08e+06	1.61e+07	9.26e+06	9.21e+06	3.83e+05
6	5.57e+06	8.89e+06	5.66e+06	5.64e+06	1.58e+05	9.09e+06	1.68e+07	9.25e+06	9.22e+06	3.57e+05
7	5.58e+06	1.02e+07	5.66e+06	5.64e+06	2.21e+05	9.11e+06	1.66e+07	9.25e+06	9.21e+06	4.09e+05
8	5.57e+06	1.02e+07	5.68e+06	5.65e+06	2.98e+05	9.10e+06	1.66e+07	9.28e+06	9.24e+06	4.47e+05
9	5.57e+06	1.03e+07	5.68e+06	5.64e+06	3.06e+05	9.09e+06	1.69e+07	9.29e+06	9.22e+06	5.40e+05
10	5.58e+06	1.03e+07	5.71e+06	5.66e+06	3.30e+05	9.14e+06	1.69e+07	9.31e+06	9.29e+06	3.08e+05
11	5.60e+06	1.03e+07	5.69e+06	5.66e+06	2.91e+05	9.12e+06	1.34e+07	9.28e+06	9.25e+06	2.36e+05
12	5.61e+06	8.42e+06	5.69e+06	5.68e+06	1.26e+05	9.11e+06	1.66e+07	9.28e+06	9.22e+06	4.62e+05
13	5.59e+06	1.03e+07	5.70e+06	5.65e+06	3.76e+05	9.14e+06	1.56e+07	9.32e+06	9.28e+06	3.19e+05
14	5.60e+06	1.03e+07	5.70e+06	5.66e+06	3.34e+05	9.15e+06	1.69e+07	9.32e+06	9.27e+06	4.69e+05
15	5.60e+06	9.92e+06	5.68e+06	5.66e+06	2.09e+05	9.16e+06	1.67e+07	9.34e+06	9.30e+06	3.94e+05

Referente à Figura 6.2b

Tabela B.22: Tempo de execução do Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.90e+05	3.82e+05	1.92e+05	1.91e+05	1.33e+04	5.79e+06	1.11e+07	5.84e+06	5.82e+06	2.72e+05
2	1.90e+05	3.77e+05	1.92e+05	1.91e+05	1.26e+04	5.29e+06	1.02e+07	5.35e+06	5.33e+06	2.57e+05
3	1.90e+05	2.68e+05	1.92e+05	1.91e+05	2.59e+03	5.24e+06	9.92e+06	5.28e+06	5.26e+06	2.31e+05
4	1.90e+05	3.78e+05	1.93e+05	1.91e+05	1.53e+04	5.13e+06	7.26e+06	5.18e+06	5.16e+06	1.29e+05
5	1.85e+05	3.77e+05	1.92e+05	1.91e+05	1.23e+04	5.13e+06	9.20e+06	5.18e+06	5.15e+06	2.84e+05
6	1.90e+05	3.83e+05	1.92e+05	1.91e+05	9.56e+03	5.14e+06	1.02e+07	5.18e+06	5.16e+06	2.54e+05
7	1.90e+05	3.80e+05	1.93e+05	1.91e+05	1.88e+04	5.13e+06	1.02e+07	5.19e+06	5.16e+06	3.16e+05
8	1.84e+05	3.78e+05	1.92e+05	1.91e+05	1.15e+04	5.13e+06	9.03e+06	5.18e+06	5.16e+06	2.28e+05
9	1.89e+05	3.80e+05	1.92e+05	1.91e+05	1.30e+04	5.11e+06	9.01e+06	5.18e+06	5.16e+06	2.26e+05
10	1.90e+05	3.80e+05	1.93e+05	1.91e+05	1.55e+04	5.13e+06	7.85e+06	5.17e+06	5.16e+06	1.66e+05
11	1.90e+05	3.78e+05	1.92e+05	1.91e+05	1.01e+04	5.13e+06	9.62e+06	5.17e+06	5.16e+06	1.97e+05
12	1.90e+05	3.78e+05	1.92e+05	1.91e+05	1.15e+04	5.14e+06	9.21e+06	5.18e+06	5.16e+06	1.79e+05
13	1.90e+05	3.77e+05	1.93e+05	1.91e+05	1.43e+04	5.14e+06	1.01e+07	5.18e+06	5.16e+06	2.37e+05
14	1.90e+05	3.67e+05	1.92e+05	1.91e+05	6.14e+03	5.13e+06	9.87e+06	5.18e+06	5.16e+06	2.62e+05
15	1.90e+05	3.85e+05	1.92e+05	1.91e+05	1.15e+04	5.13e+06	1.02e+07	5.18e+06	5.16e+06	2.61e+05

Tabela B.23: Tempo de execução do Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR DE BORDA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.93e+05	5.95e+05	2.97e+05	2.95e+05	2.05e+04	7.52e+06	1.01e+07	7.61e+06	7.59e+06	1.72e+05
2	3.43e+05	6.86e+05	3.47e+05	3.45e+05	1.86e+04	1.03e+07	1.78e+07	1.04e+07	1.03e+07	3.76e+05
3	3.56e+05	4.68e+05	3.58e+05	3.58e+05	4.55e+03	1.09e+07	1.36e+07	1.10e+07	1.10e+07	2.51e+05
4	3.63e+05	7.28e+05	3.66e+05	3.64e+05	2.34e+04	1.12e+07	1.40e+07	1.13e+07	1.13e+07	1.73e+05
5	3.51e+05	7.21e+05	3.65e+05	3.64e+05	1.89e+04	1.12e+07	1.91e+07	1.13e+07	1.12e+07	4.06e+05
6	3.62e+05	7.30e+05	3.65e+05	3.64e+05	1.94e+04	1.12e+07	1.82e+07	1.13e+07	1.13e+07	4.19e+05
7	3.62e+05	7.25e+05	3.70e+05	3.67e+05	3.33e+04	1.12e+07	2.00e+07	1.13e+07	1.12e+07	4.81e+05
8	3.49e+05	7.24e+05	3.65e+05	3.64e+05	1.79e+04	1.12e+07	1.72e+07	1.13e+07	1.13e+07	3.96e+05
9	3.49e+05	7.07e+05	3.66e+05	3.64e+05	1.83e+04	1.12e+07	1.88e+07	1.13e+07	1.12e+07	5.26e+05
10	3.62e+05	7.30e+05	3.66e+05	3.64e+05	2.41e+04	1.12e+07	1.74e+07	1.13e+07	1.12e+07	3.30e+05
11	3.62e+05	7.18e+05	3.65e+05	3.64e+05	1.55e+04	1.12e+07	1.49e+07	1.13e+07	1.12e+07	2.37e+05
12	3.63e+05	7.27e+05	3.66e+05	3.64e+05	2.16e+04	1.12e+07	1.99e+07	1.13e+07	1.12e+07	3.29e+05
13	3.62e+05	7.22e+05	3.66e+05	3.64e+05	2.27e+04	1.12e+07	2.13e+07	1.13e+07	1.13e+07	5.19e+05
14	3.62e+05	7.24e+05	3.65e+05	3.64e+05	1.18e+04	1.12e+07	1.78e+07	1.13e+07	1.12e+07	4.55e+05
15	3.62e+05	7.27e+05	3.65e+05	3.64e+05	1.26e+04	1.12e+07	1.92e+07	1.13e+07	1.12e+07	3.30e+05

Tabela B.24: Tempo de execução do Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	3.81e+05	7.78e+05	4.02e+05	4.00e+05	2.34e+04	8.62e+06	1.54e+07	8.75e+06	8.72e+06	3.31e+05
2	5.32e+05	9.85e+05	5.39e+05	5.37e+05	2.89e+04	1.22e+07	2.09e+07	1.24e+07	1.23e+07	4.72e+05
3	5.65e+05	1.07e+06	5.71e+05	5.69e+05	2.54e+04	1.32e+07	2.00e+07	1.34e+07	1.33e+07	4.46e+05
4	5.84e+05	1.11e+06	5.90e+05	5.88e+05	3.11e+04	1.38e+07	1.79e+07	1.40e+07	1.39e+07	2.90e+05
5	5.85e+05	1.10e+06	5.93e+05	5.91e+05	3.04e+04	1.38e+07	2.24e+07	1.40e+07	1.39e+07	5.88e+05
6	5.84e+05	1.11e+06	5.91e+05	5.88e+05	3.55e+04	1.38e+07	2.06e+07	1.40e+07	1.39e+07	3.83e+05
7	5.83e+05	1.12e+06	5.96e+05	5.92e+05	4.69e+04	1.38e+07	2.14e+07	1.40e+07	1.39e+07	5.31e+05
8	5.84e+05	1.12e+06	5.91e+05	5.89e+05	2.84e+04	1.38e+07	1.97e+07	1.40e+07	1.39e+07	4.48e+05
9	5.69e+05	1.11e+06	5.89e+05	5.86e+05	3.27e+04	1.38e+07	2.28e+07	1.40e+07	1.39e+07	5.73e+05
10	5.85e+05	1.10e+06	5.92e+05	5.90e+05	2.79e+04	1.38e+07	2.28e+07	1.40e+07	1.39e+07	4.77e+05
11	5.84e+05	1.09e+06	5.90e+05	5.90e+05	1.61e+04	1.38e+07	2.09e+07	1.40e+07	1.40e+07	3.75e+05
12	5.90e+05	1.10e+06	5.98e+05	5.96e+05	2.65e+04	1.38e+07	2.48e+07	1.40e+07	1.40e+07	4.96e+05
13	5.86e+05	1.11e+06	5.92e+05	5.90e+05	2.93e+04	1.38e+07	2.50e+07	1.40e+07	1.39e+07	4.87e+05
14	5.86e+05	1.11e+06	5.91e+05	5.90e+05	1.66e+04	1.38e+07	2.17e+07	1.40e+07	1.39e+07	3.96e+05
15	5.89e+05	1.11e+06	5.95e+05	5.93e+05	2.79e+04	1.38e+07	2.13e+07	1.40e+07	1.39e+07	4.46e+05

Referente à Figura 6.2c**Tabela B.25:** Tempo de execução do Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR										
k	Transformada da Erosão					Transformada da Dilatação				
	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	1.03e+06	2.01e+06	1.05e+06	1.04e+06	5.08e+04	8.22e+05	1.65e+06	8.52e+05	8.48e+05	4.21e+04
2	1.02e+06	2.03e+06	1.05e+06	1.04e+06	8.64e+04	8.30e+05	1.68e+06	8.56e+05	8.48e+05	7.57e+04
3	1.03e+06	2.03e+06	1.05e+06	1.04e+06	8.25e+04	8.41e+05	1.64e+06	8.55e+05	8.48e+05	6.09e+04
4	1.03e+06	1.67e+06	1.04e+06	1.04e+06	3.20e+04	8.41e+05	1.65e+06	8.50e+05	8.48e+05	3.03e+04
5	1.03e+06	1.98e+06	1.04e+06	1.04e+06	3.29e+04	8.39e+05	1.66e+06	8.54e+05	8.48e+05	6.15e+04
6	1.03e+06	2.01e+06	1.05e+06	1.04e+06	5.91e+04	8.40e+05	1.66e+06	8.57e+05	8.49e+05	7.01e+04
7	1.03e+06	2.00e+06	1.05e+06	1.04e+06	6.37e+04	8.31e+05	1.66e+06	8.54e+05	8.48e+05	6.17e+04
8	1.03e+06	2.04e+06	1.05e+06	1.04e+06	6.98e+04	8.41e+05	1.53e+06	8.50e+05	8.48e+05	3.52e+04
9	1.02e+06	1.88e+06	1.05e+06	1.04e+06	5.89e+04	8.41e+05	1.68e+06	8.54e+05	8.48e+05	5.99e+04
10	1.03e+06	1.98e+06	1.04e+06	1.04e+06	5.49e+04	8.32e+05	1.69e+06	8.53e+05	8.48e+05	5.98e+04
11	1.03e+06	2.03e+06	1.05e+06	1.04e+06	5.40e+04	8.41e+05	1.68e+06	8.52e+05	8.48e+05	5.49e+04
12	1.03e+06	2.02e+06	1.05e+06	1.04e+06	5.27e+04	8.27e+05	1.67e+06	8.55e+05	8.48e+05	6.64e+04
13	1.03e+06	2.00e+06	1.05e+06	1.04e+06	6.34e+04	8.33e+05	1.66e+06	8.55e+05	8.48e+05	7.21e+04
14	1.03e+06	1.98e+06	1.04e+06	1.04e+06	3.89e+04	8.41e+05	1.66e+06	8.56e+05	8.48e+05	7.42e+04
15	1.02e+06	2.03e+06	1.04e+06	1.04e+06	4.92e+04	8.41e+05	1.65e+06	8.53e+05	8.48e+05	5.64e+04

Tabela B.26: Tempo de execução do Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR DE BORDA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	3.00e+06	4.63e+06	3.05e+06	3.04e+06	8.83e+04	2.80e+06	5.60e+06	2.84e+06	2.83e+06	1.60e+05
2	3.96e+06	7.98e+06	4.04e+06	4.00e+06	3.27e+05	3.76e+06	7.34e+06	3.81e+06	3.77e+06	2.59e+05
3	4.20e+06	8.45e+06	4.26e+06	4.24e+06	2.34e+05	3.99e+06	7.08e+06	4.04e+06	4.01e+06	2.34e+05
4	4.34e+06	8.02e+06	4.37e+06	4.36e+06	1.48e+05	4.11e+06	6.27e+06	4.13e+06	4.13e+06	8.63e+04
5	4.34e+06	7.49e+06	4.38e+06	4.36e+06	1.92e+05	4.09e+06	7.94e+06	4.15e+06	4.13e+06	2.40e+05
6	4.32e+06	7.47e+06	4.38e+06	4.36e+06	1.88e+05	4.11e+06	8.27e+06	4.17e+06	4.13e+06	3.28e+05
7	4.33e+06	6.71e+06	4.38e+06	4.36e+06	1.48e+05	4.11e+06	8.21e+06	4.17e+06	4.13e+06	3.45e+05
8	4.34e+06	6.98e+06	4.37e+06	4.36e+06	1.41e+05	4.11e+06	5.92e+06	4.14e+06	4.13e+06	1.14e+05
9	4.33e+06	6.71e+06	4.39e+06	4.36e+06	2.03e+05	4.10e+06	8.27e+06	4.18e+06	4.13e+06	3.33e+05
10	4.34e+06	8.71e+06	4.38e+06	4.36e+06	2.28e+05	4.11e+06	8.25e+06	4.15e+06	4.13e+06	2.51e+05
11	4.33e+06	7.56e+06	4.38e+06	4.36e+06	1.94e+05	4.11e+06	7.94e+06	4.15e+06	4.13e+06	2.32e+05
12	4.32e+06	8.55e+06	4.38e+06	4.36e+06	2.07e+05	4.10e+06	8.16e+06	4.16e+06	4.13e+06	2.85e+05
13	4.33e+06	7.11e+06	4.39e+06	4.36e+06	2.03e+05	4.11e+06	8.23e+06	4.16e+06	4.13e+06	2.78e+05
14	4.33e+06	6.52e+06	4.38e+06	4.36e+06	1.62e+05	4.09e+06	8.12e+06	4.16e+06	4.13e+06	2.83e+05
15	4.34e+06	8.65e+06	4.39e+06	4.36e+06	2.48e+05	4.11e+06	7.57e+06	4.15e+06	4.13e+06	2.16e+05

Tabela B.27: Tempo de execução do Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	4.80e+06	8.00e+06	4.86e+06	4.84e+06	1.72e+05	4.52e+06	7.91e+06	4.60e+06	4.58e+06	1.82e+05
2	7.25e+06	1.31e+07	7.38e+06	7.32e+06	3.92e+05	7.02e+06	1.25e+07	7.13e+06	7.10e+06	2.42e+05
3	7.91e+06	1.07e+07	8.00e+06	7.97e+06	2.07e+05	7.56e+06	1.04e+07	7.66e+06	7.62e+06	2.40e+05
4	8.22e+06	1.49e+07	8.33e+06	8.30e+06	2.45e+05	7.86e+06	1.48e+07	8.10e+06	8.07e+06	2.97e+05
5	8.18e+06	1.34e+07	8.33e+06	8.28e+06	3.41e+05	7.88e+06	1.42e+07	8.04e+06	7.99e+06	4.15e+05
6	8.20e+06	1.46e+07	8.33e+06	8.28e+06	3.39e+05	7.84e+06	1.07e+07	8.05e+06	8.03e+06	1.84e+05
7	8.18e+06	1.34e+07	8.33e+06	8.30e+06	3.19e+05	7.85e+06	1.40e+07	7.97e+06	7.93e+06	3.42e+05
8	8.20e+06	1.40e+07	8.30e+06	8.26e+06	3.29e+05	7.93e+06	1.36e+07	8.09e+06	8.05e+06	2.38e+05
9	8.21e+06	1.35e+07	8.37e+06	8.30e+06	3.69e+05	7.89e+06	1.41e+07	8.02e+06	7.96e+06	3.79e+05
10	8.20e+06	1.36e+07	8.31e+06	8.27e+06	2.71e+05	8.02e+06	1.36e+07	8.17e+06	8.13e+06	3.13e+05
11	8.19e+06	1.47e+07	8.31e+06	8.25e+06	4.24e+05	7.84e+06	1.34e+07	7.96e+06	7.92e+06	2.86e+05
12	8.20e+06	1.46e+07	8.32e+06	8.28e+06	3.14e+05	7.85e+06	1.42e+07	7.96e+06	7.93e+06	2.91e+05
13	8.21e+06	1.21e+07	8.34e+06	8.29e+06	2.95e+05	7.88e+06	1.38e+07	8.03e+06	7.98e+06	3.79e+05
14	8.23e+06	1.47e+07	8.35e+06	8.29e+06	4.37e+05	7.85e+06	1.31e+07	7.95e+06	7.91e+06	3.04e+05
15	8.21e+06	1.45e+07	8.36e+06	8.30e+06	4.35e+05	7.93e+06	1.41e+07	8.09e+06	8.03e+06	4.48e+05

Referente à Figura 6.2d

Tabela B.28: Tempo de execução do Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEAR										
Transformada da Erosão						Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	6.52e+06	9.15e+06	6.56e+06	6.53e+06	2.02e+05	6.17e+07	9.63e+07	6.22e+07	6.18e+07	2.10e+06
2	6.66e+06	1.29e+07	6.73e+06	6.68e+06	4.02e+05	6.16e+07	8.14e+07	6.19e+07	6.17e+07	1.34e+06
3	6.54e+06	1.11e+07	6.60e+06	6.56e+06	2.66e+05	6.04e+07	7.84e+07	6.07e+07	6.05e+07	1.16e+06
4	6.28e+06	1.25e+07	6.35e+06	6.30e+06	4.34e+05	5.72e+07	7.14e+07	5.76e+07	5.73e+07	1.24e+06
5	6.21e+06	1.25e+07	6.26e+06	6.22e+06	3.15e+05	5.59e+07	7.81e+07	5.62e+07	5.60e+07	1.03e+06
6	6.49e+06	1.18e+07	6.53e+06	6.51e+06	2.70e+05	6.03e+07	7.37e+07	6.06e+07	6.04e+07	7.44e+05
7	6.47e+06	1.04e+07	6.53e+06	6.50e+06	2.18e+05	6.02e+07	7.65e+07	6.06e+07	6.03e+07	1.12e+06
8	6.53e+06	1.31e+07	6.59e+06	6.55e+06	3.44e+05	6.07e+07	8.71e+07	6.12e+07	6.08e+07	1.85e+06
9	6.49e+06	9.47e+06	6.54e+06	6.51e+06	2.10e+05	6.01e+07	8.14e+07	6.06e+07	6.02e+07	1.72e+06
10	6.45e+06	9.57e+06	6.49e+06	6.47e+06	1.91e+05	5.93e+07	7.26e+07	5.95e+07	5.93e+07	8.91e+05
11	6.42e+06	1.28e+07	6.47e+06	6.44e+06	3.02e+05	5.87e+07	7.62e+07	5.89e+07	5.87e+07	8.81e+05
12	6.41e+06	1.24e+07	6.45e+06	6.43e+06	2.78e+05	5.84e+07	7.45e+07	5.87e+07	5.85e+07	9.64e+05
13	6.34e+06	1.25e+07	6.41e+06	6.37e+06	3.98e+05	5.76e+07	7.49e+07	5.78e+07	5.76e+07	1.03e+06
14	6.38e+06	1.24e+07	6.43e+06	6.40e+06	3.06e+05	5.81e+07	8.70e+07	5.86e+07	5.82e+07	2.06e+06
15	6.30e+06	1.27e+07	6.37e+06	6.33e+06	3.30e+05	5.66e+07	7.74e+07	5.70e+07	5.67e+07	1.56e+06

Tabela B.29: Tempo de execução do Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEARDEBORDA										
Transformada da Erosão						Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.74e+06	5.40e+06	2.77e+06	2.76e+06	1.10e+05	6.65e+06	1.31e+07	6.77e+06	6.71e+06	4.38e+05
2	3.90e+06	7.70e+06	3.95e+06	3.92e+06	2.10e+05	9.70e+06	1.65e+07	9.82e+06	9.77e+06	3.75e+05
3	4.11e+06	8.14e+06	4.15e+06	4.14e+06	1.80e+05	1.03e+07	2.02e+07	1.04e+07	1.03e+07	6.00e+05
4	4.28e+06	8.48e+06	4.35e+06	4.31e+06	3.31e+05	1.07e+07	1.89e+07	1.08e+07	1.08e+07	4.59e+05
5	4.27e+06	8.43e+06	4.33e+06	4.31e+06	2.60e+05	1.07e+07	1.73e+07	1.08e+07	1.07e+07	3.25e+05
6	4.28e+06	8.43e+06	4.32e+06	4.31e+06	1.68e+05	1.07e+07	1.51e+07	1.08e+07	1.08e+07	2.63e+05
7	4.29e+06	8.53e+06	4.34e+06	4.32e+06	2.70e+05	1.07e+07	2.11e+07	1.09e+07	1.08e+07	5.59e+05
8	4.28e+06	8.49e+06	4.34e+06	4.31e+06	3.02e+05	1.07e+07	2.11e+07	1.08e+07	1.08e+07	5.22e+05
9	4.28e+06	8.05e+06	4.33e+06	4.32e+06	1.44e+05	1.07e+07	2.12e+07	1.09e+07	1.08e+07	4.99e+05
10	4.28e+06	8.09e+06	4.33e+06	4.32e+06	1.50e+05	1.07e+07	1.51e+07	1.08e+07	1.08e+07	3.08e+05
11	4.29e+06	8.52e+06	4.34e+06	4.32e+06	1.99e+05	1.08e+07	1.67e+07	1.09e+07	1.08e+07	3.47e+05
12	4.27e+06	8.48e+06	4.33e+06	4.32e+06	1.55e+05	1.07e+07	2.04e+07	1.09e+07	1.08e+07	3.90e+05
13	4.29e+06	8.44e+06	4.35e+06	4.31e+06	2.59e+05	1.07e+07	1.64e+07	1.09e+07	1.08e+07	3.08e+05
14	4.29e+06	8.48e+06	4.34e+06	4.32e+06	2.28e+05	1.07e+07	2.11e+07	1.09e+07	1.08e+07	5.89e+05
15	4.28e+06	8.48e+06	4.34e+06	4.32e+06	2.32e+05	1.07e+07	1.76e+07	1.08e+07	1.08e+07	3.37e+05

Tabela B.30: Tempo de execução do Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA										
	Transformada da Erosão					Transformada da Dilatação				
k	Mínimo	Máximo	Média	Mediana	Desvio padrão	Mínimo	Máximo	Média	Mediana	Desvio padrão
1	2.99e+06	5.30e+06	3.03e+06	3.02e+06	1.12e+05	7.18e+06	1.33e+07	7.29e+06	7.27e+06	2.24e+05
2	4.49e+06	8.25e+06	4.56e+06	4.53e+06	2.31e+05	1.11e+07	1.70e+07	1.13e+07	1.12e+07	3.59e+05
3	4.82e+06	8.76e+06	4.88e+06	4.87e+06	1.76e+05	1.19e+07	2.19e+07	1.22e+07	1.21e+07	5.22e+05
4	5.04e+06	9.23e+06	5.13e+06	5.10e+06	2.51e+05	1.26e+07	2.13e+07	1.29e+07	1.28e+07	5.63e+05
5	5.05e+06	9.21e+06	5.13e+06	5.10e+06	2.53e+05	1.25e+07	2.28e+07	1.28e+07	1.27e+07	4.93e+05
6	5.04e+06	8.33e+06	5.12e+06	5.10e+06	1.87e+05	1.25e+07	2.03e+07	1.28e+07	1.27e+07	3.11e+05
7	5.03e+06	9.27e+06	5.12e+06	5.09e+06	2.99e+05	1.26e+07	1.99e+07	1.29e+07	1.28e+07	5.22e+05
8	5.05e+06	9.23e+06	5.14e+06	5.11e+06	2.68e+05	1.26e+07	1.93e+07	1.28e+07	1.28e+07	3.40e+05
9	5.04e+06	9.15e+06	5.12e+06	5.10e+06	1.84e+05	1.26e+07	2.27e+07	1.29e+07	1.29e+07	5.66e+05
10	5.05e+06	7.45e+06	5.12e+06	5.11e+06	1.26e+05	1.26e+07	1.82e+07	1.28e+07	1.28e+07	2.25e+05
11	5.05e+06	9.23e+06	5.12e+06	5.11e+06	1.95e+05	1.25e+07	1.91e+07	1.28e+07	1.28e+07	2.86e+05
12	5.05e+06	9.16e+06	5.12e+06	5.10e+06	1.83e+05	1.26e+07	2.01e+07	1.29e+07	1.28e+07	3.76e+05
13	5.06e+06	9.24e+06	5.13e+06	5.11e+06	2.62e+05	1.26e+07	2.28e+07	1.29e+07	1.28e+07	5.56e+05
14	5.05e+06	9.33e+06	5.14e+06	5.11e+06	2.86e+05	1.26e+07	2.27e+07	1.29e+07	1.28e+07	4.92e+05
15	5.06e+06	9.19e+06	5.13e+06	5.11e+06	2.06e+05	1.26e+07	2.31e+07	1.29e+07	1.29e+07	4.21e+05

B.2 Contadores de Comparações e Acessos à Memória

B.2.1 Elementos estruturantes 3×3

Referente à Figura 6.1

Tabela B.31: Contadores para o Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR								
	Transformada da Erosão				Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	1.42e+08	6.25e+06	4.09e+07	9.66e+06	5.93e+08	6.25e+06	5.28e+07	2.16e+07
2	1.55e+08	6.25e+06	4.09e+07	9.66e+06	6.47e+08	6.25e+06	5.28e+07	2.16e+07
3	1.54e+08	6.25e+06	4.09e+07	9.66e+06	6.36e+08	6.25e+06	5.28e+07	2.16e+07
4	1.50e+08	6.25e+06	4.09e+07	9.66e+06	6.28e+08	6.25e+06	5.28e+07	2.16e+07
5	1.46e+08	6.25e+06	4.09e+07	9.66e+06	6.07e+08	6.25e+06	5.28e+07	2.16e+07
6	1.47e+08	6.25e+06	4.09e+07	9.66e+06	6.09e+08	6.25e+06	5.28e+07	2.16e+07
7	1.41e+08	6.25e+06	4.09e+07	9.66e+06	5.95e+08	6.25e+06	5.28e+07	2.16e+07
8	1.40e+08	6.25e+06	4.09e+07	9.66e+06	5.88e+08	6.25e+06	5.28e+07	2.16e+07
9	1.40e+08	6.25e+06	4.09e+07	9.66e+06	5.87e+08	6.25e+06	5.28e+07	2.16e+07
10	1.40e+08	6.25e+06	4.09e+07	9.66e+06	5.82e+08	6.25e+06	5.28e+07	2.16e+07
11	1.35e+08	6.25e+06	4.09e+07	9.66e+06	5.66e+08	6.25e+06	5.28e+07	2.16e+07
12	1.35e+08	6.25e+06	4.09e+07	9.66e+06	5.60e+08	6.25e+06	5.28e+07	2.16e+07
13	1.32e+08	6.25e+06	4.09e+07	9.66e+06	5.53e+08	6.25e+06	5.28e+07	2.16e+07
14	1.33e+08	6.25e+06	4.09e+07	9.66e+06	5.58e+08	6.25e+06	5.28e+07	2.16e+07
15	1.33e+08	6.25e+06	4.09e+07	9.66e+06	5.59e+08	6.25e+06	5.28e+07	2.16e+07

Tabela B.32: Contadores para o Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR DE BORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	4.25e+06	2.37e+07	4.28e+07	9.66e+06	1.10e+07	4.75e+07	5.72e+07	2.16e+07
2	5.05e+06	3.02e+07	4.28e+07	9.66e+06	1.33e+07	6.31e+07	5.72e+07	2.16e+07
3	5.86e+06	3.33e+07	4.28e+07	9.66e+06	1.47e+07	7.10e+07	5.72e+07	2.16e+07
4	5.93e+06	3.33e+07	4.28e+07	9.66e+06	1.47e+07	7.10e+07	5.72e+07	2.16e+07
5	5.84e+06	3.33e+07	4.28e+07	9.66e+06	1.43e+07	7.10e+07	5.72e+07	2.16e+07
6	5.82e+06	3.33e+07	4.28e+07	9.66e+06	1.42e+07	7.10e+07	5.72e+07	2.16e+07
7	5.65e+06	3.33e+07	4.28e+07	9.66e+06	1.41e+07	7.10e+07	5.72e+07	2.16e+07
8	5.64e+06	3.33e+07	4.28e+07	9.66e+06	1.40e+07	7.10e+07	5.72e+07	2.16e+07
9	5.61e+06	3.33e+07	4.28e+07	9.66e+06	1.40e+07	7.10e+07	5.72e+07	2.16e+07
10	5.64e+06	3.33e+07	4.28e+07	9.66e+06	1.40e+07	7.10e+07	5.72e+07	2.16e+07
11	5.57e+06	3.33e+07	4.28e+07	9.66e+06	1.38e+07	7.10e+07	5.72e+07	2.16e+07
12	5.58e+06	3.33e+07	4.28e+07	9.66e+06	1.38e+07	7.10e+07	5.72e+07	2.16e+07
13	5.57e+06	3.33e+07	4.28e+07	9.66e+06	1.38e+07	7.10e+07	5.72e+07	2.16e+07
14	5.57e+06	3.33e+07	4.28e+07	9.66e+06	1.39e+07	7.10e+07	5.72e+07	2.16e+07
15	5.54e+06	3.33e+07	4.28e+07	9.66e+06	1.38e+07	7.10e+07	5.72e+07	2.16e+07

Tabela B.33: Contadores para o Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	2.56e+07	9.86e+07	1.79e+07	0.00e+00	4.94e+07	1.75e+08	3.78e+07
2	0.00e+00	3.33e+07	1.17e+08	2.03e+07	0.00e+00	6.67e+07	2.16e+08	4.29e+07
3	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.38e+08	4.58e+07
4	0.00e+00	3.72e+07	1.28e+08	2.20e+07	0.00e+00	7.53e+07	2.38e+08	4.58e+07
5	0.00e+00	3.72e+07	1.28e+08	2.20e+07	0.00e+00	7.53e+07	2.38e+08	4.59e+07
6	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.38e+08	4.57e+07
7	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.37e+08	4.54e+07
8	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.37e+08	4.54e+07
9	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.37e+08	4.55e+07
10	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.37e+08	4.56e+07
11	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.37e+08	4.56e+07
12	0.00e+00	3.72e+07	1.28e+08	2.19e+07	0.00e+00	7.53e+07	2.37e+08	4.57e+07
13	0.00e+00	3.72e+07	1.28e+08	2.20e+07	0.00e+00	7.53e+07	2.38e+08	4.58e+07
14	0.00e+00	3.72e+07	1.28e+08	2.20e+07	0.00e+00	7.53e+07	2.38e+08	4.58e+07
15	0.00e+00	3.72e+07	1.28e+08	2.20e+07	0.00e+00	7.53e+07	2.38e+08	4.57e+07

Referente à Figura 6.2a

Tabela B.34: Contadores para o Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEAR								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	9.73e+07	1.45e+06	9.93e+06	2.69e+06	2.58e+08	1.45e+06	1.18e+07	4.55e+06
2	1.08e+08	1.45e+06	9.93e+06	2.69e+06	2.86e+08	1.45e+06	1.18e+07	4.55e+06
3	1.07e+08	1.45e+06	9.93e+06	2.69e+06	2.87e+08	1.45e+06	1.18e+07	4.55e+06
4	1.04e+08	1.45e+06	9.93e+06	2.69e+06	2.85e+08	1.45e+06	1.18e+07	4.55e+06
5	1.02e+08	1.45e+06	9.93e+06	2.69e+06	2.73e+08	1.45e+06	1.18e+07	4.55e+06
6	1.02e+08	1.45e+06	9.93e+06	2.69e+06	2.73e+08	1.45e+06	1.18e+07	4.55e+06
7	9.80e+07	1.45e+06	9.93e+06	2.69e+06	2.61e+08	1.45e+06	1.18e+07	4.55e+06
8	9.67e+07	1.45e+06	9.93e+06	2.69e+06	2.61e+08	1.45e+06	1.18e+07	4.55e+06
9	9.72e+07	1.45e+06	9.93e+06	2.69e+06	2.61e+08	1.45e+06	1.18e+07	4.55e+06
10	9.70e+07	1.45e+06	9.93e+06	2.69e+06	2.58e+08	1.45e+06	1.18e+07	4.55e+06
11	9.40e+07	1.45e+06	9.93e+06	2.69e+06	2.51e+08	1.45e+06	1.18e+07	4.55e+06
12	9.36e+07	1.45e+06	9.93e+06	2.69e+06	2.48e+08	1.45e+06	1.18e+07	4.55e+06
13	9.17e+07	1.45e+06	9.93e+06	2.69e+06	2.44e+08	1.45e+06	1.18e+07	4.55e+06
14	9.18e+07	1.45e+06	9.93e+06	2.69e+06	2.44e+08	1.45e+06	1.18e+07	4.55e+06
15	9.19e+07	1.45e+06	9.93e+06	2.69e+06	2.44e+08	1.45e+06	1.18e+07	4.55e+06

Tabela B.35: Contadores para o Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEARDEBORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	1.30e+06	6.80e+06	1.05e+07	2.69e+06	2.12e+06	1.05e+07	1.27e+07	4.55e+06
2	1.63e+06	8.93e+06	1.05e+07	2.69e+06	2.84e+06	1.41e+07	1.27e+07	4.55e+06
3	1.95e+06	9.99e+06	1.05e+07	2.69e+06	3.25e+06	1.59e+07	1.27e+07	4.55e+06
4	1.97e+06	9.99e+06	1.05e+07	2.69e+06	3.38e+06	1.59e+07	1.27e+07	4.55e+06
5	1.92e+06	9.99e+06	1.05e+07	2.69e+06	3.26e+06	1.59e+07	1.27e+07	4.55e+06
6	1.90e+06	9.99e+06	1.05e+07	2.69e+06	3.22e+06	1.59e+07	1.27e+07	4.55e+06
7	1.82e+06	9.99e+06	1.05e+07	2.69e+06	3.03e+06	1.59e+07	1.27e+07	4.55e+06
8	1.81e+06	9.99e+06	1.05e+07	2.69e+06	3.05e+06	1.59e+07	1.27e+07	4.55e+06
9	1.80e+06	9.99e+06	1.05e+07	2.69e+06	3.04e+06	1.59e+07	1.27e+07	4.55e+06
10	1.82e+06	9.99e+06	1.05e+07	2.69e+06	3.07e+06	1.59e+07	1.27e+07	4.55e+06
11	1.79e+06	9.99e+06	1.05e+07	2.69e+06	3.00e+06	1.59e+07	1.27e+07	4.55e+06
12	1.80e+06	9.99e+06	1.05e+07	2.69e+06	3.00e+06	1.59e+07	1.27e+07	4.55e+06
13	1.79e+06	9.99e+06	1.05e+07	2.69e+06	3.00e+06	1.59e+07	1.27e+07	4.55e+06
14	1.79e+06	9.99e+06	1.05e+07	2.69e+06	2.98e+06	1.59e+07	1.27e+07	4.55e+06
15	1.77e+06	9.99e+06	1.05e+07	2.69e+06	2.95e+06	1.59e+07	1.27e+07	4.55e+06

Tabela B.36: Contadores para o Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	6.83e+06	2.50e+07	4.67e+06	0.00e+00	1.05e+07	3.76e+07	8.01e+06
2	0.00e+00	8.99e+06	2.99e+07	5.23e+06	0.00e+00	1.42e+07	4.59e+07	8.94e+06
3	0.00e+00	1.01e+07	3.27e+07	5.60e+06	0.00e+00	1.60e+07	5.03e+07	9.52e+06
4	0.00e+00	1.01e+07	3.28e+07	5.64e+06	0.00e+00	1.60e+07	5.05e+07	9.57e+06
5	0.00e+00	1.01e+07	3.28e+07	5.66e+06	0.00e+00	1.60e+07	5.06e+07	9.59e+06
6	0.00e+00	1.01e+07	3.27e+07	5.62e+06	0.00e+00	1.60e+07	5.04e+07	9.53e+06
7	0.00e+00	1.01e+07	3.26e+07	5.59e+06	0.00e+00	1.60e+07	5.02e+07	9.47e+06
8	0.00e+00	1.01e+07	3.27e+07	5.60e+06	0.00e+00	1.60e+07	5.02e+07	9.48e+06
9	0.00e+00	1.01e+07	3.26e+07	5.60e+06	0.00e+00	1.60e+07	5.02e+07	9.46e+06
10	0.00e+00	1.01e+07	3.27e+07	5.62e+06	0.00e+00	1.60e+07	5.03e+07	9.51e+06
11	0.00e+00	1.01e+07	3.27e+07	5.62e+06	0.00e+00	1.60e+07	5.03e+07	9.50e+06
12	0.00e+00	1.01e+07	3.28e+07	5.65e+06	0.00e+00	1.60e+07	5.04e+07	9.55e+06
13	0.00e+00	1.01e+07	3.29e+07	5.67e+06	0.00e+00	1.60e+07	5.05e+07	9.58e+06
14	0.00e+00	1.01e+07	3.28e+07	5.66e+06	0.00e+00	1.60e+07	5.05e+07	9.57e+06
15	0.00e+00	1.01e+07	3.28e+07	5.65e+06	0.00e+00	1.60e+07	5.05e+07	9.56e+06

Referente à Figura 6.2b

Tabela B.37: Contadores para o Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.60e+07	1.41e+06	1.38e+07	6.79e+06
2	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.87e+07	1.41e+06	1.38e+07	6.79e+06
3	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.89e+07	1.41e+06	1.38e+07	6.79e+06
4	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.86e+07	1.41e+06	1.38e+07	6.79e+06
5	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.82e+07	1.41e+06	1.38e+07	6.79e+06
6	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.80e+07	1.41e+06	1.38e+07	6.79e+06
7	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.84e+07	1.41e+06	1.38e+07	6.79e+06
8	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06
9	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.84e+07	1.41e+06	1.38e+07	6.79e+06
10	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06
11	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06
12	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06
13	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06
14	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06
15	9.25e+04	1.41e+06	7.28e+06	2.43e+05	2.83e+07	1.41e+06	1.38e+07	6.79e+06

Tabela B.38: Contadores para o Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 3×3 .

TRANSFORMADADUALINEARDEBORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	9.25e+04	1.74e+06	7.33e+06	2.43e+05	4.30e+06	1.47e+07	1.52e+07	6.79e+06
2	9.25e+04	1.84e+06	7.33e+06	2.43e+05	5.01e+06	1.98e+07	1.52e+07	6.79e+06
3	9.25e+04	1.89e+06	7.33e+06	2.43e+05	5.20e+06	2.23e+07	1.52e+07	6.79e+06
4	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.99e+06	2.23e+07	1.52e+07	6.79e+06
5	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.86e+06	2.23e+07	1.52e+07	6.79e+06
6	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.76e+06	2.23e+07	1.52e+07	6.79e+06
7	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.98e+06	2.23e+07	1.52e+07	6.79e+06
8	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.90e+06	2.23e+07	1.52e+07	6.79e+06
9	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.95e+06	2.23e+07	1.52e+07	6.79e+06
10	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.87e+06	2.23e+07	1.52e+07	6.79e+06
11	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.88e+06	2.23e+07	1.52e+07	6.79e+06
12	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.88e+06	2.23e+07	1.52e+07	6.79e+06
13	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.88e+06	2.23e+07	1.52e+07	6.79e+06
14	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.88e+06	2.23e+07	1.52e+07	6.79e+06
15	9.25e+04	1.89e+06	7.33e+06	2.43e+05	4.88e+06	2.23e+07	1.52e+07	6.79e+06

Tabela B.39: Contadores para o Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 3×3 .

TRANSFORMADAPROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	1.89e+06	9.24e+06	6.23e+05	0.00e+00	1.50e+07	4.93e+07	1.09e+07
2	0.00e+00	2.09e+06	1.01e+07	8.17e+05	0.00e+00	2.04e+07	6.31e+07	1.28e+07
3	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.89e+07	1.34e+07
4	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.86e+07	1.33e+07
5	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.89e+07	1.34e+07
6	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.86e+07	1.33e+07
7	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.81e+07	1.31e+07
8	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.81e+07	1.31e+07
9	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.84e+07	1.32e+07
10	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.85e+07	1.33e+07
11	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.85e+07	1.33e+07
12	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.85e+07	1.33e+07
13	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.85e+07	1.33e+07
14	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.85e+07	1.33e+07
15	0.00e+00	2.18e+06	1.03e+07	8.27e+05	0.00e+00	2.31e+07	6.85e+07	1.33e+07

Referente à Figura 6.2c

Tabela B.40: Contadores para o Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEAR								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	3.44e+06	1.70e+06	1.28e+07	4.29e+06	3.36e+06	1.70e+06	1.27e+07	4.22e+06
2	3.49e+06	1.70e+06	1.28e+07	4.29e+06	3.40e+06	1.70e+06	1.27e+07	4.22e+06
3	3.51e+06	1.70e+06	1.28e+07	4.29e+06	3.42e+06	1.70e+06	1.27e+07	4.22e+06
4	3.51e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
5	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
6	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
7	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
8	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
9	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
10	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
11	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
12	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
13	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
14	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06
15	3.52e+06	1.70e+06	1.28e+07	4.29e+06	3.43e+06	1.70e+06	1.27e+07	4.22e+06

Tabela B.41: Contadores para o Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 3×3 .

TRANSFORMADADUALLINEARDEBORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	1.74e+06	8.70e+06	1.37e+07	4.29e+06	1.72e+06	8.58e+06	1.36e+07	4.22e+06
2	1.91e+06	1.10e+07	1.37e+07	4.29e+06	1.87e+06	1.08e+07	1.36e+07	4.22e+06
3	2.05e+06	1.21e+07	1.37e+07	4.29e+06	2.01e+06	1.19e+07	1.36e+07	4.22e+06
4	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.01e+06	1.19e+07	1.36e+07	4.22e+06
5	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
6	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
7	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
8	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
9	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
10	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
11	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
12	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
13	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
14	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06
15	2.06e+06	1.21e+07	1.37e+07	4.29e+06	2.02e+06	1.19e+07	1.36e+07	4.22e+06

Tabela B.42: Contadores para o Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	1.03e+07	3.93e+07	8.27e+06	0.00e+00	1.01e+07	3.89e+07	8.16e+06
2	0.00e+00	1.37e+07	4.79e+07	9.41e+06	0.00e+00	1.35e+07	4.73e+07	9.28e+06
3	0.00e+00	1.54e+07	5.31e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
4	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
5	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
6	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
7	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
8	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
9	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
10	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
11	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
12	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
13	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
14	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07
15	0.00e+00	1.54e+07	5.32e+07	1.03e+07	0.00e+00	1.52e+07	5.25e+07	1.02e+07

Referente à Figura 6.2d

Tabela B.43: Contadores para o Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	4.09e+07	1.69e+06	1.09e+07	2.44e+06	3.81e+08	1.69e+06	1.45e+07	6.03e+06
2	4.40e+07	1.69e+06	1.09e+07	2.44e+06	4.07e+08	1.69e+06	1.45e+07	6.03e+06
3	4.35e+07	1.69e+06	1.09e+07	2.44e+06	3.99e+08	1.69e+06	1.45e+07	6.03e+06
4	4.27e+07	1.69e+06	1.09e+07	2.44e+06	3.93e+08	1.69e+06	1.45e+07	6.03e+06
5	4.11e+07	1.69e+06	1.09e+07	2.44e+06	3.80e+08	1.69e+06	1.45e+07	6.03e+06
6	4.16e+07	1.69e+06	1.09e+07	2.44e+06	3.84e+08	1.69e+06	1.45e+07	6.03e+06
7	3.98e+07	1.69e+06	1.09e+07	2.44e+06	3.78e+08	1.69e+06	1.45e+07	6.03e+06
8	3.94e+07	1.69e+06	1.09e+07	2.44e+06	3.72e+08	1.69e+06	1.45e+07	6.03e+06
9	3.93e+07	1.69e+06	1.09e+07	2.44e+06	3.69e+08	1.69e+06	1.45e+07	6.03e+06
10	3.89e+07	1.69e+06	1.09e+07	2.44e+06	3.65e+08	1.69e+06	1.45e+07	6.03e+06
11	3.79e+07	1.69e+06	1.09e+07	2.44e+06	3.54e+08	1.69e+06	1.45e+07	6.03e+06
12	3.77e+07	1.69e+06	1.09e+07	2.44e+06	3.49e+08	1.69e+06	1.45e+07	6.03e+06
13	3.71e+07	1.69e+06	1.09e+07	2.44e+06	3.46e+08	1.69e+06	1.45e+07	6.03e+06
14	3.71e+07	1.69e+06	1.09e+07	2.44e+06	3.52e+08	1.69e+06	1.45e+07	6.03e+06
15	3.72e+07	1.69e+06	1.09e+07	2.44e+06	3.52e+08	1.69e+06	1.45e+07	6.03e+06

Tabela B.44: Contadores para o Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 3×3 .

TRANSFORMADA DUAL LINEAR DE BORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	1.10e+06	6.50e+06	1.14e+07	2.44e+06	2.87e+06	1.37e+07	1.57e+07	6.03e+06
2	1.42e+06	8.40e+06	1.14e+07	2.44e+06	3.66e+06	1.84e+07	1.57e+07	6.03e+06
3	1.76e+06	9.35e+06	1.14e+07	2.44e+06	4.27e+06	2.08e+07	1.57e+07	6.03e+06
4	1.81e+06	9.35e+06	1.14e+07	2.44e+06	4.36e+06	2.08e+07	1.57e+07	6.03e+06
5	1.77e+06	9.35e+06	1.14e+07	2.44e+06	4.28e+06	2.08e+07	1.57e+07	6.03e+06
6	1.76e+06	9.35e+06	1.14e+07	2.44e+06	4.27e+06	2.08e+07	1.57e+07	6.03e+06
7	1.68e+06	9.35e+06	1.14e+07	2.44e+06	4.18e+06	2.08e+07	1.57e+07	6.03e+06
8	1.68e+06	9.35e+06	1.14e+07	2.44e+06	4.13e+06	2.08e+07	1.57e+07	6.03e+06
9	1.66e+06	9.35e+06	1.14e+07	2.44e+06	4.06e+06	2.08e+07	1.57e+07	6.03e+06
10	1.67e+06	9.35e+06	1.14e+07	2.44e+06	4.09e+06	2.08e+07	1.57e+07	6.03e+06
11	1.63e+06	9.35e+06	1.14e+07	2.44e+06	4.01e+06	2.08e+07	1.57e+07	6.03e+06
12	1.63e+06	9.35e+06	1.14e+07	2.44e+06	4.01e+06	2.08e+07	1.57e+07	6.03e+06
13	1.63e+06	9.35e+06	1.14e+07	2.44e+06	4.01e+06	2.08e+07	1.57e+07	6.03e+06
14	1.63e+06	9.35e+06	1.14e+07	2.44e+06	4.06e+06	2.08e+07	1.57e+07	6.03e+06
15	1.62e+06	9.35e+06	1.14e+07	2.44e+06	4.02e+06	2.08e+07	1.57e+07	6.03e+06

Tabela B.45: Contadores para o Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 3×3 .

TRANSFORMADA PROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	6.58e+06	2.50e+07	4.38e+06	0.00e+00	1.37e+07	4.88e+07	1.06e+07
2	0.00e+00	8.53e+06	2.94e+07	4.87e+06	0.00e+00	1.86e+07	5.97e+07	1.19e+07
3	0.00e+00	9.50e+06	3.18e+07	5.18e+06	0.00e+00	2.10e+07	6.59e+07	1.27e+07
4	0.00e+00	9.50e+06	3.19e+07	5.21e+06	0.00e+00	2.10e+07	6.59e+07	1.27e+07
5	0.00e+00	9.50e+06	3.19e+07	5.21e+06	0.00e+00	2.10e+07	6.59e+07	1.27e+07
6	0.00e+00	9.50e+06	3.18e+07	5.19e+06	0.00e+00	2.10e+07	6.58e+07	1.27e+07
7	0.00e+00	9.50e+06	3.17e+07	5.15e+06	0.00e+00	2.10e+07	6.56e+07	1.26e+07
8	0.00e+00	9.50e+06	3.17e+07	5.15e+06	0.00e+00	2.10e+07	6.55e+07	1.26e+07
9	0.00e+00	9.50e+06	3.17e+07	5.14e+06	0.00e+00	2.10e+07	6.54e+07	1.26e+07
10	0.00e+00	9.50e+06	3.17e+07	5.15e+06	0.00e+00	2.10e+07	6.56e+07	1.26e+07
11	0.00e+00	9.50e+06	3.17e+07	5.15e+06	0.00e+00	2.10e+07	6.56e+07	1.26e+07
12	0.00e+00	9.50e+06	3.18e+07	5.16e+06	0.00e+00	2.10e+07	6.57e+07	1.27e+07
13	0.00e+00	9.50e+06	3.18e+07	5.18e+06	0.00e+00	2.10e+07	6.59e+07	1.27e+07
14	0.00e+00	9.50e+06	3.18e+07	5.18e+06	0.00e+00	2.10e+07	6.59e+07	1.27e+07
15	0.00e+00	9.50e+06	3.18e+07	5.17e+06	0.00e+00	2.10e+07	6.58e+07	1.27e+07

B.2.2 Elementos estruturantes 5×5

Referente à Figura 6.1

Tabela B.46: Contadores para o Algoritmo 9 para a Figura 6.1, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	1.65e+08	6.25e+06	4.09e+07	9.66e+06	7.22e+08	6.25e+06	5.28e+07	2.16e+07
2	1.70e+08	6.25e+06	4.09e+07	9.66e+06	7.24e+08	6.25e+06	5.28e+07	2.16e+07
3	1.66e+08	6.25e+06	4.09e+07	9.66e+06	7.10e+08	6.25e+06	5.28e+07	2.16e+07
4	1.60e+08	6.25e+06	4.09e+07	9.66e+06	6.76e+08	6.25e+06	5.28e+07	2.16e+07
5	1.58e+08	6.25e+06	4.09e+07	9.66e+06	6.64e+08	6.25e+06	5.28e+07	2.16e+07
6	1.68e+08	6.25e+06	4.09e+07	9.66e+06	7.11e+08	6.25e+06	5.28e+07	2.16e+07
7	1.67e+08	6.25e+06	4.09e+07	9.66e+06	7.10e+08	6.25e+06	5.28e+07	2.16e+07
8	1.69e+08	6.25e+06	4.09e+07	9.66e+06	7.17e+08	6.25e+06	5.28e+07	2.16e+07
9	1.67e+08	6.25e+06	4.09e+07	9.66e+06	7.09e+08	6.25e+06	5.28e+07	2.16e+07
10	1.66e+08	6.25e+06	4.09e+07	9.66e+06	7.01e+08	6.25e+06	5.28e+07	2.16e+07
11	1.65e+08	6.25e+06	4.09e+07	9.66e+06	6.95e+08	6.25e+06	5.28e+07	2.16e+07
12	1.65e+08	6.25e+06	4.09e+07	9.66e+06	6.93e+08	6.25e+06	5.28e+07	2.16e+07
13	1.63e+08	6.25e+06	4.09e+07	9.66e+06	6.82e+08	6.25e+06	5.28e+07	2.16e+07
14	1.64e+08	6.25e+06	4.09e+07	9.66e+06	6.89e+08	6.25e+06	5.28e+07	2.16e+07
15	1.61e+08	6.25e+06	4.09e+07	9.66e+06	6.73e+08	6.25e+06	5.28e+07	2.16e+07

Tabela B.47: Contadores para o Algoritmo 13 para a Figura 6.1, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR DE BORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	6.95e+06	4.69e+07	4.28e+07	9.66e+06	1.79e+07	1.08e+08	5.72e+07	2.16e+07
2	7.54e+06	7.01e+07	4.28e+07	9.66e+06	2.02e+07	1.67e+08	5.72e+07	2.16e+07
3	6.96e+06	7.60e+07	4.28e+07	9.66e+06	1.85e+07	1.82e+08	5.72e+07	2.16e+07
4	7.02e+06	7.89e+07	4.28e+07	9.66e+06	1.78e+07	1.89e+08	5.72e+07	2.16e+07
5	6.97e+06	7.89e+07	4.28e+07	9.66e+06	1.76e+07	1.89e+08	5.72e+07	2.16e+07
6	7.16e+06	7.89e+07	4.28e+07	9.66e+06	1.82e+07	1.89e+08	5.72e+07	2.16e+07
7	7.20e+06	7.89e+07	4.28e+07	9.66e+06	1.83e+07	1.89e+08	5.72e+07	2.16e+07
8	7.17e+06	7.89e+07	4.28e+07	9.66e+06	1.82e+07	1.89e+08	5.72e+07	2.16e+07
9	7.15e+06	7.89e+07	4.28e+07	9.66e+06	1.81e+07	1.89e+08	5.72e+07	2.16e+07
10	7.14e+06	7.89e+07	4.28e+07	9.66e+06	1.81e+07	1.89e+08	5.72e+07	2.16e+07
11	7.26e+06	7.89e+07	4.28e+07	9.66e+06	1.84e+07	1.89e+08	5.72e+07	2.16e+07
12	7.25e+06	7.89e+07	4.28e+07	9.66e+06	1.83e+07	1.89e+08	5.72e+07	2.16e+07
13	7.18e+06	7.89e+07	4.28e+07	9.66e+06	1.82e+07	1.89e+08	5.72e+07	2.16e+07
14	7.20e+06	7.89e+07	4.28e+07	9.66e+06	1.82e+07	1.89e+08	5.72e+07	2.16e+07
15	7.10e+06	7.89e+07	4.28e+07	9.66e+06	1.79e+07	1.89e+08	5.72e+07	2.16e+07

Tabela B.48: Contadores para o Algoritmo 21 para a Figura 6.1, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	5.65e+07	1.77e+08	2.88e+07	0.00e+00	1.19e+08	3.42e+08	5.90e+07
2	0.00e+00	8.74e+07	2.53e+08	3.84e+07	0.00e+00	1.88e+08	4.92e+08	7.43e+07
3	0.00e+00	9.51e+07	2.69e+08	4.01e+07	0.00e+00	2.05e+08	5.28e+08	7.77e+07
4	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.52e+08	8.13e+07
5	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.51e+08	8.12e+07
6	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.51e+08	8.13e+07
7	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.52e+08	8.13e+07
8	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.51e+08	8.13e+07
9	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.51e+08	8.13e+07
10	0.00e+00	9.90e+07	2.81e+08	4.20e+07	0.00e+00	2.14e+08	5.51e+08	8.12e+07
11	0.00e+00	9.90e+07	2.81e+08	4.21e+07	0.00e+00	2.14e+08	5.52e+08	8.14e+07
12	0.00e+00	9.90e+07	2.81e+08	4.21e+07	0.00e+00	2.14e+08	5.52e+08	8.15e+07
13	0.00e+00	9.90e+07	2.81e+08	4.21e+07	0.00e+00	2.14e+08	5.52e+08	8.15e+07
14	0.00e+00	9.90e+07	2.81e+08	4.21e+07	0.00e+00	2.14e+08	5.52e+08	8.15e+07
15	0.00e+00	9.90e+07	2.81e+08	4.21e+07	0.00e+00	2.14e+08	5.52e+08	8.14e+07

Referente à Figura 6.2a

Tabela B.49: Contadores para o Algoritmo 9 para a Figura 6.2a, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	1.15e+08	1.45e+06	9.93e+06	2.69e+06	3.11e+08	1.45e+06	1.18e+07	4.55e+06
2	1.20e+08	1.45e+06	9.93e+06	2.69e+06	3.17e+08	1.45e+06	1.18e+07	4.55e+06
3	1.17e+08	1.45e+06	9.93e+06	2.69e+06	3.11e+08	1.45e+06	1.18e+07	4.55e+06
4	1.12e+08	1.45e+06	9.93e+06	2.69e+06	2.95e+08	1.45e+06	1.18e+07	4.55e+06
5	1.11e+08	1.45e+06	9.93e+06	2.69e+06	2.92e+08	1.45e+06	1.18e+07	4.55e+06
6	1.19e+08	1.45e+06	9.93e+06	2.69e+06	3.13e+08	1.45e+06	1.18e+07	4.55e+06
7	1.18e+08	1.45e+06	9.93e+06	2.69e+06	3.12e+08	1.45e+06	1.18e+07	4.55e+06
8	1.20e+08	1.45e+06	9.93e+06	2.69e+06	3.17e+08	1.45e+06	1.18e+07	4.55e+06
9	1.18e+08	1.45e+06	9.93e+06	2.69e+06	3.12e+08	1.45e+06	1.18e+07	4.55e+06
10	1.17e+08	1.45e+06	9.93e+06	2.69e+06	3.09e+08	1.45e+06	1.18e+07	4.55e+06
11	1.17e+08	1.45e+06	9.93e+06	2.69e+06	3.06e+08	1.45e+06	1.18e+07	4.55e+06
12	1.17e+08	1.45e+06	9.93e+06	2.69e+06	3.06e+08	1.45e+06	1.18e+07	4.55e+06
13	1.15e+08	1.45e+06	9.93e+06	2.69e+06	3.00e+08	1.45e+06	1.18e+07	4.55e+06
14	1.16e+08	1.45e+06	9.93e+06	2.69e+06	3.04e+08	1.45e+06	1.18e+07	4.55e+06
15	1.13e+08	1.45e+06	9.93e+06	2.69e+06	2.97e+08	1.45e+06	1.18e+07	4.55e+06

Tabela B.50: Contadores para o Algoritmo 13 para a Figura 6.2a, usando elementos estruturantes 5×5 .

TRANSFORMADADUALINEARDEBORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	2.35e+06	1.52e+07	1.05e+07	2.69e+06	3.72e+06	2.49e+07	1.27e+07	4.55e+06
2	2.63e+06	2.36e+07	1.05e+07	2.69e+06	4.21e+06	3.92e+07	1.27e+07	4.55e+06
3	2.32e+06	2.57e+07	1.05e+07	2.69e+06	3.71e+06	4.28e+07	1.27e+07	4.55e+06
4	2.38e+06	2.68e+07	1.05e+07	2.69e+06	3.80e+06	4.46e+07	1.27e+07	4.55e+06
5	2.37e+06	2.68e+07	1.05e+07	2.69e+06	3.78e+06	4.46e+07	1.27e+07	4.55e+06
6	2.49e+06	2.68e+07	1.05e+07	2.69e+06	3.99e+06	4.46e+07	1.27e+07	4.55e+06
7	2.51e+06	2.68e+07	1.05e+07	2.69e+06	4.04e+06	4.46e+07	1.27e+07	4.55e+06
8	2.50e+06	2.68e+07	1.05e+07	2.69e+06	4.03e+06	4.46e+07	1.27e+07	4.55e+06
9	2.49e+06	2.68e+07	1.05e+07	2.69e+06	3.99e+06	4.46e+07	1.27e+07	4.55e+06
10	2.48e+06	2.68e+07	1.05e+07	2.69e+06	3.98e+06	4.46e+07	1.27e+07	4.55e+06
11	2.56e+06	2.68e+07	1.05e+07	2.69e+06	4.13e+06	4.46e+07	1.27e+07	4.55e+06
12	2.55e+06	2.68e+07	1.05e+07	2.69e+06	4.13e+06	4.46e+07	1.27e+07	4.55e+06
13	2.50e+06	2.68e+07	1.05e+07	2.69e+06	4.04e+06	4.46e+07	1.27e+07	4.55e+06
14	2.52e+06	2.68e+07	1.05e+07	2.69e+06	4.06e+06	4.46e+07	1.27e+07	4.55e+06
15	2.45e+06	2.68e+07	1.05e+07	2.69e+06	3.94e+06	4.46e+07	1.27e+07	4.55e+06

Tabela B.51: Contadores para o Algoritmo 21 para a Figura 6.2a, usando elementos estruturantes 5×5 .

TRANSFORMADAPROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	1.54e+07	4.46e+07	6.89e+06	0.00e+00	2.51e+07	7.09e+07	1.18e+07
2	0.00e+00	2.41e+07	6.38e+07	8.98e+06	0.00e+00	3.97e+07	1.03e+08	1.52e+07
3	0.00e+00	2.62e+07	6.78e+07	9.24e+06	0.00e+00	4.33e+07	1.10e+08	1.57e+07
4	0.00e+00	2.73e+07	7.10e+07	9.77e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
5	0.00e+00	2.73e+07	7.11e+07	9.78e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
6	0.00e+00	2.73e+07	7.11e+07	9.80e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
7	0.00e+00	2.73e+07	7.11e+07	9.81e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
8	0.00e+00	2.73e+07	7.11e+07	9.81e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
9	0.00e+00	2.73e+07	7.11e+07	9.79e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
10	0.00e+00	2.73e+07	7.11e+07	9.80e+06	0.00e+00	4.51e+07	1.15e+08	1.66e+07
11	0.00e+00	2.73e+07	7.13e+07	9.87e+06	0.00e+00	4.51e+07	1.16e+08	1.67e+07
12	0.00e+00	2.73e+07	7.14e+07	9.89e+06	0.00e+00	4.51e+07	1.16e+08	1.67e+07
13	0.00e+00	2.73e+07	7.13e+07	9.87e+06	0.00e+00	4.51e+07	1.16e+08	1.67e+07
14	0.00e+00	2.73e+07	7.13e+07	9.88e+06	0.00e+00	4.51e+07	1.16e+08	1.67e+07
15	0.00e+00	2.73e+07	7.13e+07	9.86e+06	0.00e+00	4.51e+07	1.16e+08	1.67e+07

Referente à Figura 6.2b

Tabela B.52: Contadores para o Algoritmo 9 para a Figura 6.2b, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEAR								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.67e+07	1.41e+06	1.38e+07	6.79e+06
2	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.31e+07	1.41e+06	1.38e+07	6.79e+06
3	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.26e+07	1.41e+06	1.38e+07	6.79e+06
4	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
5	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
6	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
7	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
8	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
9	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
10	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
11	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
12	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
13	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
14	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06
15	8.77e+04	1.41e+06	7.28e+06	2.43e+05	3.18e+07	1.41e+06	1.38e+07	6.79e+06

Tabela B.53: Contadores para o Algoritmo 13 para a Figura 6.2b, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEARDEBORDA								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	8.77e+04	2.13e+06	7.33e+06	2.43e+05	6.67e+06	3.48e+07	1.52e+07	6.79e+06
2	8.77e+04	2.52e+06	7.33e+06	2.43e+05	7.82e+06	5.39e+07	1.52e+07	6.79e+06
3	8.77e+04	2.61e+06	7.33e+06	2.43e+05	7.30e+06	5.86e+07	1.52e+07	6.79e+06
4	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.56e+06	6.09e+07	1.52e+07	6.79e+06
5	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.54e+06	6.09e+07	1.52e+07	6.79e+06
6	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
7	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
8	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
9	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
10	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
11	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
12	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
13	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
14	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06
15	8.77e+04	2.66e+06	7.33e+06	2.43e+05	6.55e+06	6.09e+07	1.52e+07	6.79e+06

Tabela B.54: Contadores para o Algoritmo 21 para a Figura 6.2b, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	2.67e+06	1.25e+07	1.32e+06	0.00e+00	3.67e+07	1.01e+08	1.74e+07
2	0.00e+00	3.45e+06	1.54e+07	1.90e+06	0.00e+00	5.85e+07	1.40e+08	1.94e+07
3	0.00e+00	3.64e+06	1.63e+07	2.09e+06	0.00e+00	6.39e+07	1.52e+08	2.06e+07
4	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
5	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
6	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
7	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
8	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
9	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
10	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
11	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
12	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
13	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
14	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07
15	0.00e+00	3.74e+06	1.67e+07	2.19e+06	0.00e+00	6.66e+07	1.58e+08	2.15e+07

Referente à Figura 6.2c**Tabela B.55:** Contadores para o Algoritmo 9 para a Figura 6.2c, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
2	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
3	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
4	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
5	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
6	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
7	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
8	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
9	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
10	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
11	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
12	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
13	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
14	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06
15	2.88e+06	1.70e+06	1.28e+07	4.29e+06	2.80e+06	1.70e+06	1.27e+07	4.22e+06

Tabela B.56: Contadores para o Algoritmo 13 para a Figura 6.2c, usando elementos estruturantes 5×5 .

TRANSFORMADA DUAL LINEAR DE BORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	2.42e+06	1.57e+07	1.37e+07	4.29e+06	2.37e+06	1.54e+07	1.36e+07	4.22e+06
2	2.43e+06	2.28e+07	1.37e+07	4.29e+06	2.38e+06	2.25e+07	1.36e+07	4.22e+06
3	2.43e+06	2.46e+07	1.37e+07	4.29e+06	2.38e+06	2.42e+07	1.36e+07	4.22e+06
4	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
5	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
6	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
7	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
8	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
9	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
10	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
11	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
12	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
13	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
14	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06
15	2.43e+06	2.55e+07	1.37e+07	4.29e+06	2.38e+06	2.51e+07	1.36e+07	4.22e+06

Tabela B.57: Contadores para o Algoritmo 21 para a Figura 6.2c, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	2.40e+07	7.75e+07	1.41e+07	0.00e+00	2.37e+07	7.66e+07	1.40e+07
2	0.00e+00	3.77e+07	1.13e+08	1.92e+07	0.00e+00	3.72e+07	1.12e+08	1.90e+07
3	0.00e+00	4.11e+07	1.21e+08	2.01e+07	0.00e+00	4.05e+07	1.20e+08	1.99e+07
4	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
5	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
6	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
7	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
8	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
9	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
10	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
11	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
12	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
13	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
14	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07
15	0.00e+00	4.28e+07	1.26e+08	2.10e+07	0.00e+00	4.22e+07	1.25e+08	2.08e+07

Referente à Figura 6.2d

Tabela B.58: Contadores para o Algoritmo 9 para a Figura 6.2d, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEAR								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	4.63e+07	1.69e+06	1.09e+07	2.44e+06	4.65e+08	1.69e+06	1.45e+07	6.03e+06
2	4.74e+07	1.69e+06	1.09e+07	2.44e+06	4.64e+08	1.69e+06	1.45e+07	6.03e+06
3	4.65e+07	1.69e+06	1.09e+07	2.44e+06	4.55e+08	1.69e+06	1.45e+07	6.03e+06
4	4.45e+07	1.69e+06	1.09e+07	2.44e+06	4.30e+08	1.69e+06	1.45e+07	6.03e+06
5	4.38e+07	1.69e+06	1.09e+07	2.44e+06	4.20e+08	1.69e+06	1.45e+07	6.03e+06
6	4.60e+07	1.69e+06	1.09e+07	2.44e+06	4.54e+08	1.69e+06	1.45e+07	6.03e+06
7	4.60e+07	1.69e+06	1.09e+07	2.44e+06	4.54e+08	1.69e+06	1.45e+07	6.03e+06
8	4.64e+07	1.69e+06	1.09e+07	2.44e+06	4.57e+08	1.69e+06	1.45e+07	6.03e+06
9	4.60e+07	1.69e+06	1.09e+07	2.44e+06	4.53e+08	1.69e+06	1.45e+07	6.03e+06
10	4.57e+07	1.69e+06	1.09e+07	2.44e+06	4.46e+08	1.69e+06	1.45e+07	6.03e+06
11	4.55e+07	1.69e+06	1.09e+07	2.44e+06	4.41e+08	1.69e+06	1.45e+07	6.03e+06
12	4.54e+07	1.69e+06	1.09e+07	2.44e+06	4.40e+08	1.69e+06	1.45e+07	6.03e+06
13	4.50e+07	1.69e+06	1.09e+07	2.44e+06	4.33e+08	1.69e+06	1.45e+07	6.03e+06
14	4.52e+07	1.69e+06	1.09e+07	2.44e+06	4.37e+08	1.69e+06	1.45e+07	6.03e+06
15	4.47e+07	1.69e+06	1.09e+07	2.44e+06	4.26e+08	1.69e+06	1.45e+07	6.03e+06

Tabela B.59: Contadores para o Algoritmo 13 para a Figura 6.2d, usando elementos estruturantes 5×5 .

TRANSFORMADADUALLINEARDEBORDA								
Transformada da Erosão					Transformada da Dilatação			
k	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	2.09e+06	1.39e+07	1.14e+07	2.44e+06	5.03e+06	3.25e+07	1.57e+07	6.03e+06
2	2.39e+06	2.12e+07	1.14e+07	2.44e+06	5.84e+06	5.14e+07	1.57e+07	6.03e+06
3	2.12e+06	2.30e+07	1.14e+07	2.44e+06	5.14e+06	5.61e+07	1.57e+07	6.03e+06
4	2.11e+06	2.39e+07	1.14e+07	2.44e+06	5.16e+06	5.84e+07	1.57e+07	6.03e+06
5	2.08e+06	2.39e+07	1.14e+07	2.44e+06	4.95e+06	5.84e+07	1.57e+07	6.03e+06
6	2.15e+06	2.39e+07	1.14e+07	2.44e+06	5.36e+06	5.84e+07	1.57e+07	6.03e+06
7	2.17e+06	2.39e+07	1.14e+07	2.44e+06	5.46e+06	5.84e+07	1.57e+07	6.03e+06
8	2.15e+06	2.39e+07	1.14e+07	2.44e+06	5.32e+06	5.84e+07	1.57e+07	6.03e+06
9	2.15e+06	2.39e+07	1.14e+07	2.44e+06	5.33e+06	5.84e+07	1.57e+07	6.03e+06
10	2.14e+06	2.39e+07	1.14e+07	2.44e+06	5.29e+06	5.84e+07	1.57e+07	6.03e+06
11	2.19e+06	2.39e+07	1.14e+07	2.44e+06	5.48e+06	5.84e+07	1.57e+07	6.03e+06
12	2.18e+06	2.39e+07	1.14e+07	2.44e+06	5.43e+06	5.84e+07	1.57e+07	6.03e+06
13	2.16e+06	2.39e+07	1.14e+07	2.44e+06	5.34e+06	5.84e+07	1.57e+07	6.03e+06
14	2.16e+06	2.39e+07	1.14e+07	2.44e+06	5.33e+06	5.84e+07	1.57e+07	6.03e+06
15	2.13e+06	2.39e+07	1.14e+07	2.44e+06	5.20e+06	5.84e+07	1.57e+07	6.03e+06

Tabela B.60: Contadores para o Algoritmo 21 para a Figura 6.2d, usando elementos estruturantes 5×5 .

TRANSFORMADA PROPOSTA								
k	Transformada da Erosão				Transformada da Dilatação			
	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção	Comparações para Determinação da Borda	Comparações para Inserção	Acessos à Memória para Inserção	Acessos à Memória para Remoção
1	0.00e+00	1.44e+07	4.29e+07	6.42e+06	0.00e+00	3.30e+07	9.29e+07	1.57e+07
2	0.00e+00	2.22e+07	6.03e+07	8.34e+06	0.00e+00	5.23e+07	1.36e+08	2.04e+07
3	0.00e+00	2.42e+07	6.41e+07	8.62e+06	0.00e+00	5.71e+07	1.45e+08	2.11e+07
4	0.00e+00	2.51e+07	6.69e+07	9.07e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07
5	0.00e+00	2.51e+07	6.69e+07	9.06e+06	0.00e+00	5.95e+07	1.52e+08	2.21e+07
6	0.00e+00	2.51e+07	6.69e+07	9.06e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07
7	0.00e+00	2.51e+07	6.69e+07	9.07e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07
8	0.00e+00	2.51e+07	6.69e+07	9.06e+06	0.00e+00	5.95e+07	1.52e+08	2.21e+07
9	0.00e+00	2.51e+07	6.69e+07	9.06e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07
10	0.00e+00	2.51e+07	6.69e+07	9.06e+06	0.00e+00	5.95e+07	1.52e+08	2.21e+07
11	0.00e+00	2.51e+07	6.70e+07	9.09e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07
12	0.00e+00	2.51e+07	6.70e+07	9.09e+06	0.00e+00	5.95e+07	1.52e+08	2.23e+07
13	0.00e+00	2.51e+07	6.70e+07	9.09e+06	0.00e+00	5.95e+07	1.52e+08	2.23e+07
14	0.00e+00	2.51e+07	6.70e+07	9.09e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07
15	0.00e+00	2.51e+07	6.70e+07	9.09e+06	0.00e+00	5.95e+07	1.52e+08	2.22e+07

Bibliografia

- Chen e Haralick (1995)** Su Chen e R. M. Haralick. Recursive erosion, dilation, opening, and closing transforms. *Trans. Img. Proc.*, 4(3):335--345. ISSN 1057-7149. doi: 10.1109/83.366481. URL <http://dx.doi.org/10.1109/83.366481>. Citado na pág. 1, 7
- Cormen et al. (2009)** Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest e Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd ed. ISBN 0262033844, 9780262033848. Citado na pág. 11
- Hashimoto e Barrera (2001)** Ronaldo Fumio Hashimoto e Junior Barrera. From the sup-decomposition to sequential decompositions. *Journal of Mathematical Imaging and Vision*, 15:197--216. Citado na pág. 1
- Hashimoto e Barrera (2003)** Ronaldo Fumio Hashimoto e Junior Barrera. A greedy algorithm for decomposing convex structuring elements. *Journal of Mathematical Imaging and Vision*, 18:269--289. Citado na pág. 1
- Hashimoto et al. (2000)** Ronaldo Fumio Hashimoto, Junior Barrera e Carlos Eduardo Ferreira. A combinatorial optimization technique for the sequential decomposition of erosions and dilations. *Journal of Mathematical Imaging and Vision*, 13:17--33. Citado na pág. 1
- Hirata (1997)** Roberto Hirata, Jr. Segmentação de imagens por morfologia matemática. Dissertação de Mestrado, Instituto de Matemática e Estatística, Universidade de São Paulo, Brasil. Citado na pág. 2, 29
- Latecki et al. (2000)** Longin J. Latecki, Rolf Lakämper e Ulrich Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:1424. Citado na pág. 51
- Machado (2008)** Anderson F. Machado. Erosões e dilatações morfológicas binárias sequenciais rápidas. Dissertação de Mestrado, Instituto de Matemática e Estatística, Universidade de São Paulo, Brasil. Citado na pág. xv, 2, 51, 52
- Shih e Wu (2005)** Frank Y. Shih e Yi-Ta Wu. Decomposition of binary morphological structuring elements based on genetic algorithms. *Computer Vision and Image Understanding*, 99:291--302. Citado na pág. 1
- Xu (1991)** Jianning Xu. Decomposition of convex polygonal morphological structuring elements into neighborhood subsets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:153--162. Citado na pág. 1
- Yang e Lee (2005)** Hsin-Tai Yang e Shie-Jue Lee. Decomposition of morphological structuring elements with integer linear programming. *IEE Proceedings - Vision, Image, and Signal Processing*, 152:148--154. Citado na pág. 1
- Zampirolli (2003)** Francisco de Assis Zampirolli. *Transformada de Distância por Morfologia Matemática*. Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Brasil. Citado na pág. 2, 29
- Zhuang e Haralick (1986)** Xinhua Zhuang e Robert M. Haralick. Morphological structuring element decomposition. *Computer Vision, Graphics and Image Processing*, 35:370--382. Citado na pág. 1

Índice

- algoritmo proposto
 - nó da borda, 30
 - nó de ligação, 30
- algoritmos apresentados
 - borda, 8
 - condição de parada, 10
 - iteração, 10
- algoritmos duais
 - coordenada de interesse, 11
- estrutura de dados
 - fila, 5
 - imagem, 6
 - lista duplamente encadeada com cabeça, 5
 - matriz, 6
 - vetor, 5
- imagem
 - binária, 6
 - caixa delimitadora, 6
 - coordenada, 6
 - decomposição de elementos estruturantes, 7
 - elemento estruturante, 6
 - origem, 6
- operador morfológico
 - Dilatação, 6
 - Erosão, 6
 - gradiente morfológico externo, 7
 - gradiente morfológico interno, 7
 - raio de elemento estruturante, 7
 - Transformada da Dilatação, 7
 - Transformada da Erosão, 7
- operações em conjuntos
 - adição de Minkowski, 1
 - complemento, 6
 - reflexão, 6
 - translação, 6