


Universidade Federal de Pernambuco  
Graduação em Ciência da Computação

Centro de Informática

---



CÓDIGO DE VIZINHANÇA APLICADO EM  
OPERADORES DE MORFOLOGIA MATEMÁTICA

---

Trabalho de Graduação

Aluno: Paulo Roberto Figueirôa Amorim (prfa@cin.ufpe.br)

Orientador: Tsang Ing Ren (tir@cin.ufpe.br)

Dezembro de 2010

## Resumo

Dentro da área de processamento de imagens, uma imagem normalmente é transformada em outra de melhor forma de representação para extração de componentes úteis na sua descrição e representação. Tal procedimento pode ser realizado através de várias abordagens de operação com pixel, locais (vizinhança), geométricas e globais. A morfologia matemática é uma das áreas que realiza tal representação de imagens, através da Teoria dos Conjuntos, aplicando conceitos básicos através de operadores lógicos (AND, OR, NOT...), em que cada conjunto de uma morfologia representa objetos em uma imagem.

Uma das várias abordagens na operação de processamento de imagens, o código de vizinhança é definido como uma operação local (vizinhança), em que a saída de um pixel depende não apenas do seu próprio valor na imagem de entrada, mas também dos pixels de entrada imediatamente na janela ao seu redor, ou seja, os seus vizinhos. Esse trabalho tem como proposta apresentar, de uma forma geral, o conceito de morfologia matemática. Assim como o estudo da abordagem de código de vizinhança dentro dos operadores morfológicos, sua implementação, e também, uma extensão de sua aplicação, no âmbito binário. Além disso, oferecer uma análise comparativa do estado da arte dos operadores morfológicos existentes, com os operadores de vizinhança aqui implementados, com intuito de no final do trabalho avaliar a eficácia do método apresentado através de um conjunto de parâmetros.

## Agradecimentos

Agradeço a todos que contribuíram direta ou indiretamente com o desenvolvimento e conclusão deste trabalho.

# Índice

1. Introdução .....	8
2. Morfologia Matemática .....	10
2.1 Histórico.....	10
2.2 Conceitos.....	11
2.3 Operadores Morfológicos.....	13
2.3.1 Dilatação.....	13
2.3.2 Erosão.....	14
2.3.3 Abertura e Fechamento.....	15
2.4 Transformada Hit-or-miss.....	18
2.5 Algoritmos Morfológicos Básicos.....	20
2.5.1 Extração de Fronteiras.....	20
2.5.2 Afinamento.....	21
2.5.3 Espessamento.....	23
3. Código de Vizinhaça.....	25
3.1 Conceito.....	25
3.2 Metodologia.....	26
3.3 Operadores de Vizinhaça.....	28
3.3.1 Processamento Prévio.....	28
3.3.2 Dilatação.....	28
3.3.3 Erosão.....	29
3.3.4 Abertura e Fechamento.....	30
3.3.5 Transformada Hit-or-miss.....	30
3.3.6 Extração de Fronteiras.....	31
3.3.7 Afinamento.....	31
3.3.8 Espessamento.....	32
4. Avaliação e Resultados .....	33
4.1 Base de Imagens.....	33
4.2 Resultados.....	33
4.3 Avaliação.....	37
5. Conclusão e Trabalhos Futuros .....	39
Referências .....	40

## Índice de Equações

Equação 2.3-1: Dilatação de "A" por "B".....	13
Equação 2.3-2: Erosão de "A" por "B".....	14
Equação 2.3-3: Abertura de "A" por "B".....	16
Equação 2.3-4: Fechamento de "A" por "B".....	16
Equação 2.4-1: Formalização da transformada hit-or-miss.....	19
Equação 2.5-1: Processo de Extração de Fronteiras.....	20
Equação 2.5-2: Afinamento definido em termos da transformada hit-or-miss.....	21
Equação 2.5-3: Processo de Afinamento com uma seqüência de elementos estruturantes.....	22
Equação 2.5-4: Espessamento definido em termos da transformada hit-or-miss.....	23
Equação 2.5-5: Processo de Afinamento com uma seqüência de elementos estruturantes. ....	24

## Índice de Figuras

Figura 1: Operadores Lógicos Básicos.....	12
Figura 2: Exemplo do processo de dilatação.....	14
Figura 3: Exemplo do processo de erosão.....	15
Figura 4: Processo de abertura de uma imagem.....	17
Figura 5: Processo de fechamento de uma imagem.....	17
Figura 6: Exemplo de aplicação da transformada hit-or-miss.....	19
Figura 7: Exemplo de aplicação da Extração de Fronteiras.....	21
Figura 8: Exemplo do processo de Afinamento.....	23
Figura 9: Exemplo do processo de Espessamento.....	24
Figura 10: Imagens utilizadas como base.....	33
Figura 11: Resultado comparativo do processo de erosão.....	34
Figura 12: Resultado comparativo do processo de dilatação.....	34
Figura 13: Resultado comparativo do processo de abertura.....	35
Figura 14: Resultado comparativo do processo de fechamento.....	35
Figura 15: Resultado comparativo do processo de extração de fronteiras..	35
Figura 16: Resultado comparativo do processo da transformada hit-or-miss.....	36
Figura 17: Resultado comparativo do processo de afinamento.....	36
Figura 18: Resultado comparativo do processo de espessamento.....	36

## Índice de Tabelas

Tabela 1: Visão geral da análise técnica dos operadores.....	37
Tabela 2: Análise do tempo de execução dos operadores e redução percentual do tempo.....	38

# 1. Introdução

Dentro da análise de imagens e reconhecimento de padrões, para extração de componentes úteis na descrição e representação, uma imagem normalmente é transformada em outra de melhor forma de representação. Técnicas de processamento de imagem têm sido desenvolvidas em grande escala nos últimos cinquenta anos. Uma delas, em particular, a de morfologia matemática, que provê uma descrição quantitativa da estrutura geométrica, da forma, através de uma descrição matemática de álgebra, topologia e geometria integral da imagem, tem recebido bastante atenção, provando ser extremamente eficiente em várias aplicações de análise e processamento de imagem [2].

Em operações de processamento de imagem, várias abordagens podem ser definidas, como operações com pixel, em que a saída de um pixel depende exclusivamente da entrada desse pixel, independente dos outros pixels da imagem e operações locais (vizinhança), em que a saída de um pixel depende não apenas do pixel de mesma localização na imagem de entrada, mas também dos pixels de entrada imediatamente ao seu redor, ou seja, os seus vizinhos. Outras abordagens incluem operações geométricas, em que a saída de um pixel depende apenas do nível de entrada de alguns outros pixels definido por transformações geométricas e operações globais, em que a saída de um pixel depende de todos os outros pixels da imagem de entrada.

Dentro do contexto da morfologia matemática, o conceito morfológico é comumente utilizado para denotar noções de forma e estrutura na representação de uma ferramenta, de pré e pós-processamento, para extração de componentes de uma imagem úteis na sua descrição e representação. São utilizados dois conjuntos de vetores, um deles representando a imagem de entrada inicial, e outro um elemento estruturante auxiliar pré-definido, de forma a serem combinados para realizar determinada operação e obter um resultado final.



Um contrastante a essa idéia anterior, que depende basicamente de estruturas locais externas, é a codificação de vizinhança dentro dos operadores de morfologia matemática, assim como em seus algoritmos de aplicação. Em tal abordagem, os operadores de imagens são obtidos transformando-se os pixels da imagem de acordo com seus códigos de vizinhos nas quatro direções (norte, oeste, sul, leste), utilizando-se também os outros quatro componentes da janela de vizinhança (nordeste, noroeste, sudoeste, sudeste) quando necessário.

## 2. Morfologia Matemática

### 2.1 Histórico

A morfologia matemática teve início no ano de 1964 pelo trabalho colaborativo de Georges Matheron e Jean Serra na École des Mines localizada em Paris na França. Em 1968, foi fundado o Centre de Morphologie Mathématique liderado por ambos. Tal conceito, formalizado nos anos 60, foi utilizado inicialmente para estudar as propriedades geométricas e de trituração dos minérios.

Durante o resto dos anos 60 e boa parte dos 70, morfologia matemática foi aplicada essencialmente com imagens binárias, tratadas com o uso da teoria dos conjuntos, gerando um grande número de técnicas e operadores binários, como: a transformada hit-or-miss, dilatação, erosão, abertura, fechamento, granulometria, afinamento, esqueletização, entre outros. Entre o meio dos anos 70 e 80, morfologia matemática foi generalizada para funções e imagens em escala de cinza. Além de estender os principais conceitos (como dilatação, e erosão) para funções, essa generalização possibilitou o desenvolvimento de novos operadores, como gradientes morfológicos, a transformada top-hat e o watershed, principal abordagem de segmentação na morfologia matemática.

Nos anos 80 e 90, morfologia matemática ganhou um maior reconhecimento, ao passo que centros de pesquisa em vários países começaram a adotar e investigar tal método. Morfologia Matemática começou a ser utilizada em um grande número de aplicações e problemas de imagens. Em 1989 e 1991, respectivamente, F.Y. Shih e, O.R. Mitchell criaram a técnica de decomposição de limiar na morfologia de tons de cinza, que resultou numa nova visão do processamento de morfologia matemática em escala de cinza.

Morfologia matemática se tornou um campo de pesquisa bastante ativo desde 1985. A literatura teórica e prática apareceram como o tema central em várias revistas profissionais, e em várias conferências profissionais, como o International Symposium on Mathematical Morphology, e em sociedades profissionais como a Internacional Society for Mathematical Morphology (ISMM), fundada em 1993, em Barcelona.

A literatura em morfologia matemática se estendeu durante todo o séc. XX através de inúmeras publicações, tendo alguns destaques: o livro *Random Sets and Integral Geometry*, escrito por Matheron em 1975; dois volumes contendo teorias de morfologia matemática por Serra em 1982 e 1988; e desde 1993, *The Morphology Digest*, editado por Henk Heijmans e Pierre Soille tem como objetivo promover um canal de comunicação entre pessoas interessadas em morfologia matemática e análise morfológica de imagens.

## 2.2 Conceitos

Morfologia matemática é uma área do processamento de imagens, que tem como linguagem Teoria dos Conjuntos, em que cada conjunto de uma morfologia representa objetos em uma imagem. É uma técnica de análise e processamento de estruturas geométricas, baseada na teoria de conjuntos, teoria de redes, topologia e funções randômicas. Normalmente aplicada a imagens digitais, também pode ser aplicada em grafos, malhas de superfícies, sólidos e muitas outras estruturas espaciais.

Conceitos de topologia e geometria do espaço-contínuo como tamanho, formato, convexidade, conectividade e distância geométrica podem ser caracterizados pela morfologia matemática em ambos os espaços contínuo e discreto. A morfologia matemática também é a fundação do processamento morfológico de imagens, que consiste em um conjunto de operações que transforma imagens de acordo com a sua caracterização.

Em imagens binárias, os conjuntos em questão são membros do espaço bi-dimensional. Cada elemento de um conjunto é uma tupla (vetor bi-dimensional) cujos componentes fazem referência aos eixos "x" e "y", dos pixels pretos (ou brancos, dependendo da convenção utilizada) da imagem. Imagens em escala de cinza também podem ser representadas como conjuntos, cujos componentes fazem parte do espaço tri-dimensional. Dois componentes de cada elemento do conjunto fazem referência à coordenada do pixel e o terceiro corresponde ao seu valor de cinza discreto.

Como mostrado em [1], tal modelo bi-dimensional aplica conceitos básicos da teoria de conjuntos através de operadores lógicos (AND, OR, NOT,...) (Fig.1). Através dessa abordagem, operadores morfológicos fundamentais, como exemplo, dilatação, erosão, abertura e fechamento, são aplicados em uma imagem inicial "A" através de um segundo elemento "B", denominado de elemento estruturante.

<i>p</i>	<i>q</i>	<i>p AND q (also <math>p \cdot q</math>)</i>	<i>p OR q (also <math>p + q</math>)</i>	<i>NOT (p) (also <math>\bar{p}</math>)</i>
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Figura 1: Operadores Lógicos Básicos. (extraído de [1])

A idéia básica em morfologia binária é explorar uma imagem com uma simples, pré-determinada forma, com o intuito de determinar se essa forma existe ou não dentro dos formatos da imagem. Tal componente é denominado elemento estruturante, e ele mesmo é uma imagem binária, ou seja, um subconjunto do espaço. A transformada Hit-or-miss e outros algoritmos, como o de extração de fronteiras, afinamento e espessamento são algumas das funcionalidades que podem ser alcançadas através do uso de tais operadores.

## 2.3 Operadores Morfológicos

### 2.3.1 Dilatação

Uma das operações elementares da morfologia matemática, o processo de dilatação consiste na modificação de uma imagem inicial a partir de uma análise com elementos estruturantes de diferentes formatos e tamanho. A imagem sendo processada é denominada de imagem ativa, e a outra imagem, auxiliar ao processo, denominada como elemento estruturante. Cada elemento estruturante tem um formato específico desejado e pode ser visto como uma sonda ou filtro da imagem ativa.

Em termos mais técnicos, a dilatação de uma imagem ativa "A", por um elemento estruturante "B" é o conjunto de todos os deslocamentos "x" tais que a reflexão de "B" transladado em "x" e "A" sobreponham-se em pelo menos um elemento não-nulo. Dentro da teoria de conjuntos, a dilatação de "A" por "B" é definida como o conjunto, não-nulo, contido ou igual a "A", formado pela intersecção da reflexão do conjunto "B" em torno de sua origem, transladado em "x", com o conjunto "A", definido matematicamente pela equação 2.3-1 abaixo:

$$A \oplus B = \{x \mid [(\hat{B})_x \cap A] \neq \emptyset\} \quad (2.3-1)$$

A dilatação binária de "A" por "B" é o conjunto de todos os possíveis vetores de pares de elementos, um vindo de "A" e outro de "B". Já que ambos "A" e "B" são inicialmente imagens no âmbito bi-dimensional, tal operação morfológica aplicada nesses dois conjuntos é chama de morfologia binária.

Um exemplo de tal procedimento, conhecido como "preenchimento", "expansão" ou "crescimento" é a utilização de um elemento estruturante, representado por um pequeno quadrado (3x3) em que é realizada uma simples operação com os 8

componentes vizinhos da janela, com um centro definido, através de uma arquitetura adjacente de arrays conectados (Fig.2).

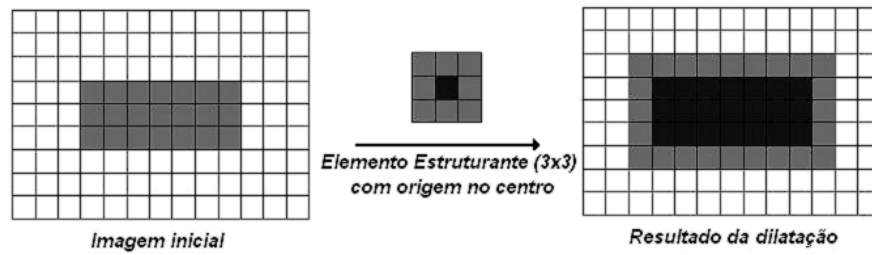


Figura 2: Exemplo do processo de dilatação. (extraído de [1])

Resumindo, pode-se definir o processo de dilatação como a “expansão” dos limites de fronteira de “A”. Como exemplo de aplicação, podemos identificar o preenchimento de espaços (gap filling) em que textos com uma pobre resolução com caracteres quebrados podem ser melhorados através de tal processo que junta tais segmentos.

### 2.3.2 Erosão

Considerado o complemento morfológico da dilatação, o processo de erosão tem como característica a combinação de dois conjuntos usando vetores para subtrair um conjunto de elementos da imagem. Em termos técnicos, a erosão de “A” por “B” é o conjunto de todos os pontos “x” tais que “B”, quando translado por “x” fique contido em “A”, demonstrado na equação 2.3-2 abaixo:

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad (2.3-2)$$

A erosão binária de "A" por "B" é o conjunto de todos os elementos "x", para os quais "x+b" pertencente a "A", para todo elemento "b" pertencente a "B". Tal noção indica que a implementação da erosão binária é similar a da dilatação, exceto pela troca de uma função OR para uma função AND e o uso da imagem transladada pela negação dos pontos de "B".

Semelhante à dilatação, um exemplo de tal procedimento, conhecido como "encolhimento", ou "redução" é a utilização de um elemento estruturante, representado por um pequeno quadrado (3x3) em que, da mesma forma, é realizada uma simples operação com os 8 componentes vizinhos da janela, com um centro definido, através de uma arquitetura adjacente de arrays conectados (Fig.3).

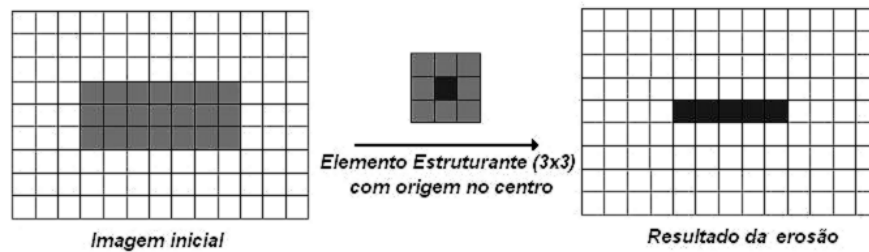


Figura 3: Exemplo do processo de erosão. (extraído de [1])

Resumindo, pode-se definir o processo de erosão como a "contração" dos limites de fronteira de "A". Como exemplo de aplicação, podemos identificar remoção de componentes, em que de acordo com a estrutura do elemento estruturante auxiliar, determinados componentes podem ser retirados de uma imagem.

### 2.3.3 Abertura e Fechamento

Como visto anteriormente, a dilatação expande uma imagem, enquanto que a erosão a encolhe. Em aplicações práticas, os processos de dilatação e erosão são combinados em seqüência para

obtenção de resultados, dilatação seguida de erosão, ou vice-versa. Em qualquer caso, o resultado desse procedimento, aplicando dilatação e erosão, é a eliminação de detalhes específicos da imagem cujo tamanho são menores do que o elemento estruturante sem nenhuma distorção global ou perda de informação essencial.

A abertura, que tem como efeito a suavização do contorno da imagem, a quebra de istmos estreitos e eliminação de protruções finas, é realizada através da erosão de "A" por "B" seguido da dilatação do resultado por "B", como demonstrado na equação 2.3-3 abaixo:

$$A \circ B = (A \ominus B) \oplus B \quad (2.3-3)$$

O fechamento, que tem como efeito a suavização do contorno da imagem, a fusão de quebras em golfos finos, a eliminação de pequenos buracos e o preenchimento de fendas em um contorno, é realizada pela dilatação de "A" por "B" seguida da erosão do resultado por "B", como demonstrado na equação 2.3-4 abaixo:

$$A \bullet B = (A \oplus B) \ominus B \quad (2.3-4)$$

Um exemplo de tais procedimentos dentro da morfologia binária com a utilização de um elemento estruturante, representado por um pequeno componente quadrado de tamanho 3x3, mostra que o processo de abertura, vai remover todos os pixels das regiões que são muito pequenas para conter o elemento estruturante, tendendo a suavizar defeitos externos, quebrar seções estreitas e eliminar protruções finas (Fig. 4). De forma semelhante, a seqüência oposta, o processo de fechamento, vai preencher buracos e concavidades menores do que o elemento auxiliar, o que tende a eliminar pequenos buracos e defeitos internos (Fig. 5).



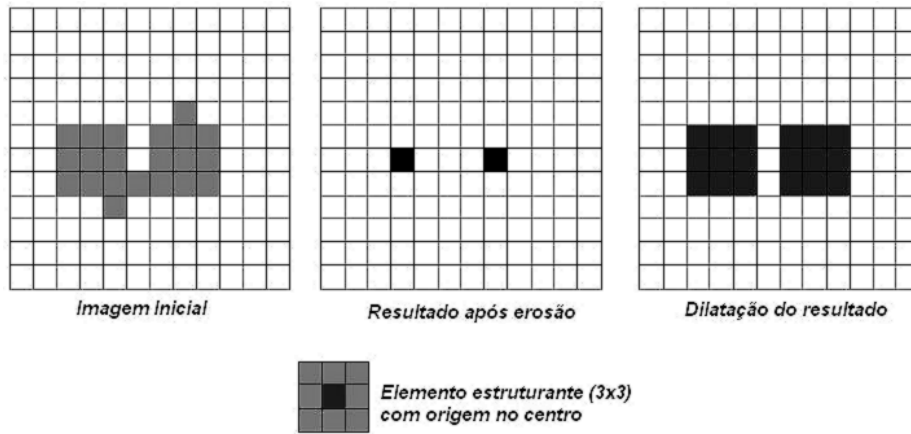


Figura 4: Processo de abertura de uma imagem. (extraído de [1])

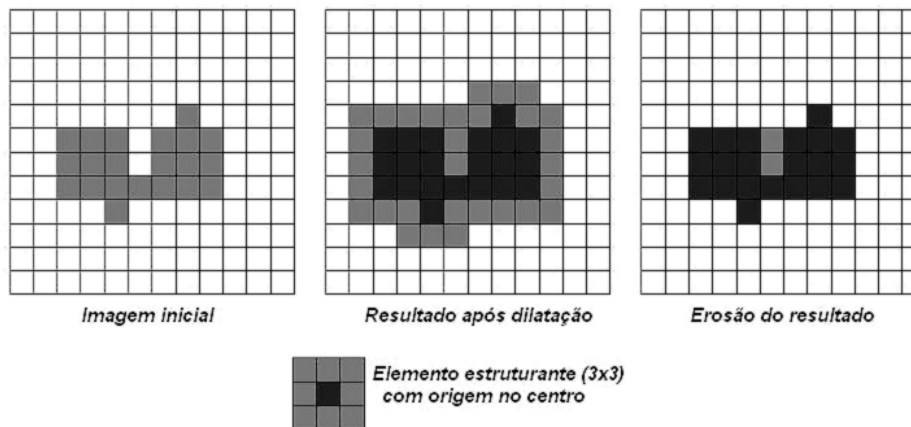


Figura 5: Processo de fechamento de uma imagem. (extraído de [1])

Como exemplo de aplicação de tais procedimentos, podemos exemplificar a redução de ruídos, assim como suprimir recursos espaciais e discriminar objetos com base na sua distribuição de tamanho.

## 2.4 Transformada Hit-or-miss

Na morfologia matemática, a transformada hit-or-miss é uma operação que detecta uma dada configuração (ou padrão) em uma imagem binária, usando o operador morfológico da erosão e um par de elementos estruturantes disjuntos. O resultado da transformada hit-or-miss é um conjunto de posições, em que o primeiro elemento estruturante se encaixa no fundo da imagem de entrada, e o segundo elemento estruturante não faz parte.

Um exemplo de tal conceito, pode ser demonstrando utilizando-se uma imagem binária "A" contendo três formas (subconjuntos) denominadas "X", "Y" e "Z" (Fig. 6[a]). O objetivo é achar a localização de uma das formas, digamos, "X". Sendo a origem de cada forma determinada pelo seu centro de gravidade, o primeiro elemento estruturante "B<sub>1</sub>" (Fig. 6[b]) é definido pela forma "X" e o segundo elemento estruturante "B<sub>2</sub>" (Fig. 6[c]) é definido pelo conjunto formado por uma pequena janela "W", menos o conteúdo do primeiro elemento, ou seja, "W-X".

O primeiro passo do procedimento é dado pelo processo de erosão da imagem "A" pelo elemento estruturante "B<sub>1</sub>", ou seja, o conjunto de localizações da origem de "X", de forma que "X" esteja contido em "A" (Fig. 6[d]). Outra interpretação, é que o processo de erosão de "A" por "B<sub>1</sub>" pode ser visto geometricamente como o conjunto de todas as localizações da origem de "X" em que "X" acha uma combinação (hit) em "A". O segundo passo do procedimento é dado pelo processo de erosão do complemento de "A" pelo elemento estruturante "B<sub>2</sub>" (Fig. 6[e]). O resultado é dado pelo conjunto de localizações em que "X" cabe exatamente dentro de "A", ou seja, a intersecção do resultado do primeiro passo com o resultado do segundo (Fig. 6[f]).

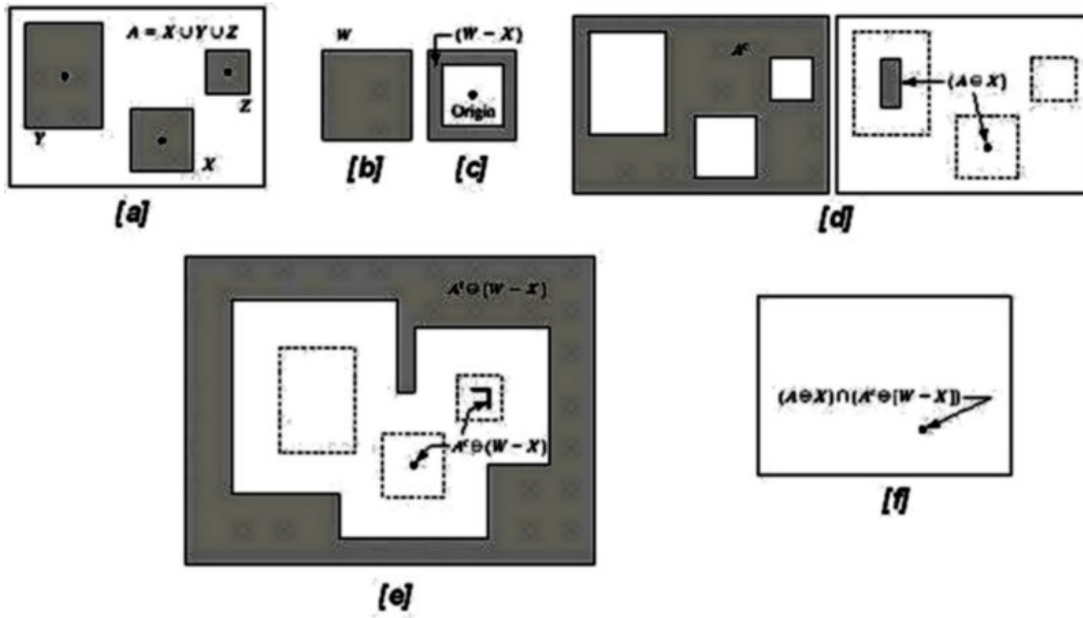


Figura 6: Exemplo de aplicação da transformada *hit-or-miss*.(extraído de [1])

A razão para se usar um elemento estruturante “ $B_1$ ” associado a objetos e um elemento “ $B_2$ ” associado ao fundo da imagem é baseado na definição assumida que dois ou mais objetos são distintos apenas se estes formam conjuntos disjuntos (desconexos). Tal noção é garantida requerendo que cada objeto tenha pelo menos uma camada de fundo da imagem ao seu redor.

De forma geral a transformada hit-or-miss pode ser apresentada através de uma equação formal pela intersecção da erosão de “ $A$ ” por “ $B_1$ ” e da erosão do complemento de “ $A$ ” por “ $B_2$ ” ou devido à relação dual entre erosão e dilatação, a subtração da erosão de “ $A$ ” por “ $B_1$ ” pela dilatação de “ $A$ ” pela reflexão de “ $B_2$ ”, como mostrado na equação 2.4-1 abaixo:

$$A * B = (A \ominus B_1) \cap (A^c \ominus B_2) \quad (2.4-1)$$

$$A * B = (A \ominus B_1) - (A \oplus \hat{B}_2)$$

## 2.5 Algoritmos Morfológicos Básicos

### 2.5.1 Extração de Fronteiras

A extração de fronteiras é uma das principais aplicações na parte de extração de componentes úteis a representação e descrição de formas de uma imagem. Tal procedimento é definido pela extração dos pixels da imagem que não possuem nenhum componente com valor referente ao fundo da imagem.

Na morfologia matemática, o processo de extração de fronteiras, em imagens binárias, é realizado através de duas etapas. Primeiramente, pela erosão da imagem inicial "A" por um elemento estruturante "B", e por final pela diferença da imagem inicial "A" com o resultado do processo de erosão, como demonstrado pela equação 2.5-1 abaixo:

$$\beta(A) = A - (A \ominus B) \quad (2.5-1)$$

Um exemplo de tal procedimento apresentado é a utilização de um simples objeto binário e um elemento estruturante de tamanho 3x3 (Fig. 7). Em tal procedimento, o tamanho do elemento estruturante define a espessura do contorno do objeto. Nesse caso, com um elemento estruturante 3x3 vai ser gerado um objeto com espessura de contorno de um pixel. No caso do elemento estruturante ser maior, por exemplo, de tamanho 5x5, a espessura do contorno do objeto gerado seria algo em torno de dois ou três pixels.

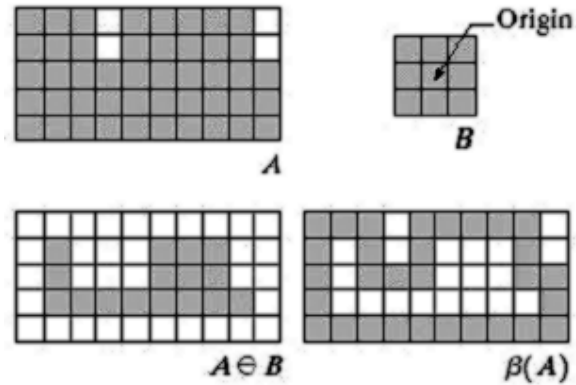


Figura 7: Exemplo de aplicação da Extração de Fronteiras. (Extraído de [1])

### 2.5.2 Afinamento

O afinamento é um processamento fundamental na representação do formato estrutural de um padrão como um grafo. Pode ser aplicado na inspeção de partes industriais, reconhecimento de digitais, reconhecimento de caracteres e diagnósticos biomédicos. Semelhante ao processo de erosão, o processo de afinamento não causa o desaparecimento dos componentes de um objeto. Tem como objetivo reduzir a espessura do objeto para um pixel, gerando um eixo minimamente conectado equidistante às bordas do objeto.

O processo de afinamento de uma imagem inicial "A" por um elemento estruturante "B" pode ser definido em termos da transformada hit-or-miss, como mostrado na equação 2.5-2. Tal procedimento pode ser visto como um processo de "procura-e-deleta", em que são localizadas todas as ocorrências de "B" em "A", e uma subtração de conjuntos remove tais pixels localizados em "A".

$$\begin{aligned}
 A \otimes B &= A - (A \circledast B) \\
 &= A \cap (A \circledast B)^c
 \end{aligned}
 \tag{2.5-2}$$

Outra forma mais utilizada de realizar o afinamento de "A" simetricamente é através da utilização de uma seqüência elementos estruturantes, em que "B<sup>i</sup>" é versão de rotação de "B<sup>i-1</sup>". O processo consiste em afinar "A" em um passo com "B<sup>1</sup>", então afinar o resultado em um passo com "B<sup>2</sup>", e assim consecutivamente, até que "A" tenha sido afinado em um passo de "B<sup>n</sup>", como mostrado na equação 2.5-3. Tal procedimento é repetido até que nenhuma mudança ocorra em relação ao resultado obtido anteriormente.

$$\begin{aligned} \{B\} &= \{B^1, B^2, B^3, \dots, B^n\} \\ A \otimes \{B\} &= (((\dots((A \otimes B^1) \otimes B^2) \dots) \otimes B^n) \end{aligned} \quad (2.5-3)$$

Um exemplo de tal procedimento, em uma imagem binária inicial "A" é a utilização de uma seqüência de oito elementos estruturantes "B<sub>1</sub>, B<sub>2</sub>, ..., B<sub>8</sub>" rotacionados de tamanho 3x3, compostos de quatro componentes preenchidos, três representando o fundo da imagem e os outros dois podendo ser tanto um quanto outro (Fig. 8). O resultado de tal procedimento descrito é apresentado seqüencialmente com a aplicação de cada um dos oito elementos e por final uma convergência a n-conectividade que representa uma seqüência continua do procedimento até que nenhuma mudança de pixel seja realizada de uma fase para a outra.

O processo de afinamento remove todos os pixels da borda externa de um objeto. Os elementos estruturantes são desenhados para achar tais pixels de borda, cuja remoção não irá mudar a conectividade do objeto.

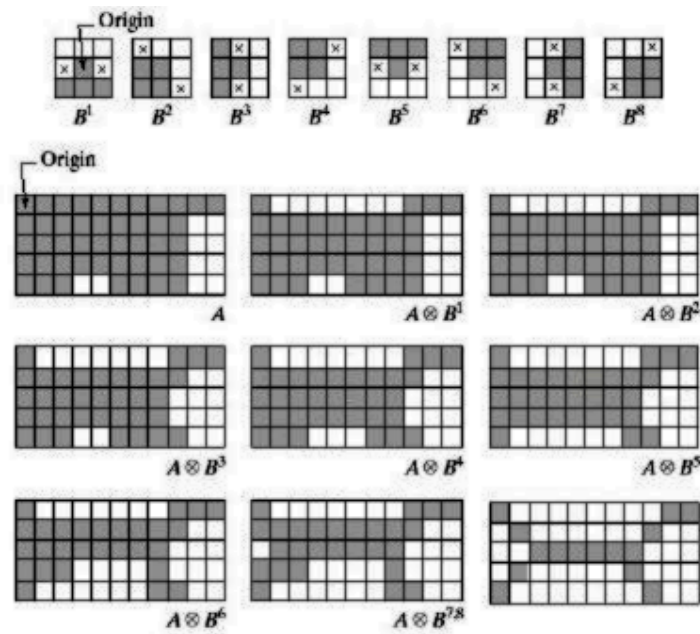


Figura 8: Exemplo do processo de Afinamento. (extraído de [1])

### 2.5.3 Espessamento

Processo morfológico dual ao afinamento, o espessamento é similar à dilatação, no entanto não causa a fusão de objetos desconectados. É utilizado para crescer algumas concavidades e também pode ser representado em termos da transformada hit-or-miss, composto por uma imagem inicial "A" e por um elemento estruturante "B", como mostrado na equação 2.5-4. A imagem final após o processo de afinamento contém o conjunto de pixels original mais alguns pixels preenchidos determinados pela transformada hit-or-miss.

$$A \odot B = A \cup (A \otimes B) \quad (2.5-4)$$

Analogamente ao afinamento, outra forma de realizar o espessamento de "A" simetricamente é através da utilização de uma seqüência de elementos estruturantes, em que "B<sup>i</sup>" é versão de rotação de "B<sup>i-1</sup>". O processo consiste em afinar "A" em um passo com "B<sup>1</sup>", então afinar o resultado em um passo com "B<sup>2</sup>", e assim

consecutivamente, até que "A" tenha sido afinado em um passo de "B<sup>n</sup>", como mostrado na equação 2.5-5 abaixo:

$$\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$$

$$A \odot \{B\} = (((\dots((A \odot B^1) \odot B^2) \dots) \odot B^n) \quad (2.5-5)$$

Um exemplo de tal procedimento, em uma imagem binária "A", bastante utilizado na prática, é um procedimento que utiliza o processo de afinamento com os mesmo oito elementos estruturantes, antes apresentados, para obter o resultado do espessamento. Tal procedimento é realizado em quatro etapas. Primeiramente, é obtido o complemento de "A". Posteriormente é feito o afinamento do complemento de "A", seguido pela obtenção do complemento do resultado obtido. Por fim, devido à natureza da imagem inicial "A", esse procedimento pode resultar em alguns pontos desconectados, por isso tal método de espessamento possui um simples passo de processamento posterior para remover tais pontos (Fig. 9).

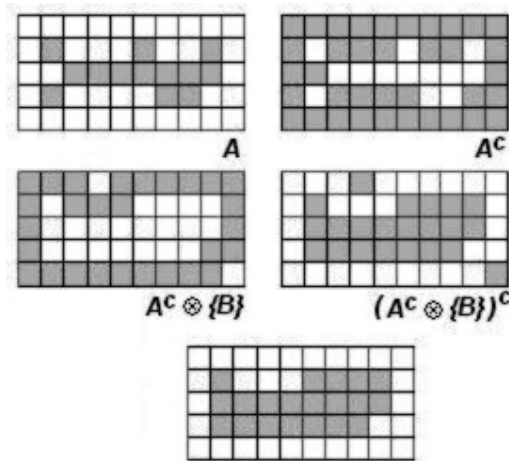


Figura 9: Exemplo do processo de Espessamento. (extraído de [1])



## 3. Código de Vizinhança

### 3.1 Conceito

O código de vizinhança representa uma das várias abordagens na operação de processamento de imagens. Definida como uma operação local (vizinhança), em que a saída de um pixel depende não apenas do seu próprio valor na imagem de entrada, mas também dos pixels de entrada imediatamente na janela ao seu redor, ou seja, os seus vizinhos.

A operação de vizinhança relaciona o pixel com os seus vizinhos. A propriedade de cada célula é mudada baseada em algum tipo de relação com suas células vizinhas ou o fluxo material das células vizinhas. Em outras palavras, para cada célula um novo valor é calculado (e armazenado em uma nova camada) com base nos valores do valor associado a células que tem algum tipo de associação espacial com essa respectiva célula. Várias categorias de associação espacial podem ser representadas por operadores de vizinhança. Em um delas, a operação de vizinhança dentro de uma janela, um novo valor é calculado para cada célula com base nos valores das células dentro de uma janela quadrada ou circular, em que a célula considerada em questão é definida como centro.

Em visão computacional e processamento de imagem, a operação de vizinhança é comumente utilizada através de um conjunto de computações em uma dada imagem. O procedimento é realizado de forma que é definida uma heurística, em que cada ponto "p" da imagem é alterado através do resultado de uma dada função "f" de acordo com um parâmetro de entrada "n" representando a vizinhança ou região de dados ao redor do ponto "p". Tal procedimento pode ser aplicado a uma imagem de dimensão arbitrária. Da mesma forma, o dado de imagem em que a operação é aplicada não precisa necessariamente ser definido em termos de intensidade ou cor, pode ser qualquer tipo de informação

que é organizada a partir de uma função espacial (e possivelmente temporal) variável em "p".

O resultado de aplicar um operador de vizinhança em uma imagem também pode ser interpretado como uma imagem, que possui a mesma dimensão da original. O valor de cada ponto, no entanto, não precisa estar diretamente relacionado a intensidade ou cor. Ao invés disso, representa um elemento dentro da função "f", que pode ser de tipo arbitrário.

Normalmente, o valor de vizinhança "n" é de tamanho fixo e é um quadrado (ou cubo, dependendo da dimensionalidade do dado que representa a imagem) centrado no ponto "p". Também, a função "f" é fixada, mas pode em alguns casos ter parâmetros que podem variar de acordo com o ponto "p". No caso mais simples, a vizinhança "n" pode ser definida por um simples ponto.

Tal heurística sugere que a operação de vizinhança pode ser implementada em termos de um loop externo, através da análise de todos os pontos da imagem. No entanto, caso os resultados sejam independentes, os pontos da imagem podem ser visitados em uma ordem arbitrária, ou até mesmo em paralelo.

## 3.2 Metodologia

Operadores de imagem são conjuntos de transformações que podem eliminar irrelevâncias, e simplificar dados de imagem, enquanto preservam suas características essenciais de forma. Várias aplicações em processamento de imagem requerem um pré-processamento para normalizar, reduzir a quantidade excessiva de dados, ou remover propriedades indesejáveis na imagem, como ruído.

Nas operações de vizinhança, uma janela é utilizada para obter o resultado da operação, enquanto que na morfologia matemática dois conjuntos de vetores são combinados através de operações de vetores dentro do conjunto de elementos. Em contraste com essas duas abordagens que dependem basicamente de estruturas locais, em um recente método de representação do formato de imagens binárias, apresentado em [3], é possível obter um operador de imagem transformando os pixels de uma imagem de acordo com os códigos nas quatro ou oito direções.

Tal método é baseado na relação de vizinhança entre pixels. Em imagens binárias a representação dos vetores de vizinhança é utilizada como um parâmetro de formato. Cada pixel é transformado em um vetor, em que cada elemento do vetor representa o total do número de pixels vizinhos nas quatro respectivas direções (norte, sul, leste e oeste), assim como, oito (incluindo noroeste, nordeste, sudeste, sudoeste).

Usando os elementos dos vetores de vizinhança como base do operador leva-se em conta não apenas características locais, mas como também, características globais da imagem, já que a informação contida no vetor de vizinhança não depende exclusivamente de uma correlação local, mas também de uma correlação de longo alcance entre os pixels. Um objeto binário é representado, então, por um conjunto de vetores de vizinhança, nos quais as informações de estrutura e formato são retidas, e que pode ser usado em métodos de extração de características para soluções de problemas de processamento de imagem e reconhecimento de padrão.

### 3.3 Operadores de Vizinhança

#### 3.3.1 Processamento Prévio

Antes que qualquer tipo de operação possa ser realizada, é necessário um processamento prévio da imagem utilizada de forma a criar um novo objeto, que represente o vetor de vizinhança. No trabalho aqui realizado é feita uma varredura dinâmica na imagem. Para cada pixel, são calculados valores de vizinhança de acordo com a quantidade de pixels vizinhos consecutivos que fazem parte de imagem até que seja alcançado um representante do fundo da imagem. Tal procedimento é generalizado através do pseudocódigo a seguir:

Para todo pixel da imagem:

Para cada uma das oito direções:

Se, o pixel imediatamente anterior (ou posterior dependendo da direção) em uma direção específica fizer parte da imagem

Então, o valor que representa essa direção dentro do vetor de vizinhança é incrementado em um.

#### 3.3.2 Dilatação

Baseado no processo de dilatação da morfologia matemática, o operador de vizinhança de dilatação é realizado através da informação contida no vetor de vizinhança. No trabalho aqui realizado, é feita uma varredura dinâmica dentro de toda a imagem, em que para cada pixel, componente da imagem, que possui um componente no vetor de vizinhança que indica à borda, são preenchidos pixels num fator de distância dois, em uma das quatro direções (norte, sul, leste e oeste) específica. Tal procedimento é generalizado através do pseudocódigo a seguir:

Para todo pixel da imagem:

Se o pixel for um componente da imagem:

Então, para cada uma das quatro direções:

Se, na direção específica, o vetor de vizinhança indicar o componente que representa a borda

Então, cada pixel, localizado a um fator de distância dois na direção específica, é preenchido como componente da imagem.

### 3.3.3 Erosão

Baseado no processo de erosão da morfologia matemática, o operador de vizinhança de erosão é realizada através da informação contida no vetor de vizinhança. No trabalho aqui realizado é feita uma varredura dinâmica dentro de toda a imagem, em que é retirado cada pixel, componente da imagem, que possui pelo menos um componente no vetor de vizinhança que indica à borda, e que possui pelo menos um pixel em ambas as direções opostas, ou seja, norte e sul, leste e oeste. Tal procedimento é generalizado através do pseudocódigo a seguir:

Para todo pixel da imagem:

Se o pixel for um componente da imagem:

Então, Se, em pelo menos uma das quatro direções, o vetor de vizinhança indicar o componente que representa a borda e possuir um pixel em ambas as direções opostas (norte e sul, leste e oeste)

Então, o pixel é retirado como componente da imagem.

### 3.3.4 Abertura e Fechamento

Analogamente ao processo de abertura e fechamento na morfologia matemática, o operador de vizinhança de abertura, aqui implementado, é realizado através do processo de erosão, seguido pela atualização do vetor de vizinhança e, posteriormente, a realização do processo de dilatação. Da mesma forma, o operador de vizinhança de fechamento, aqui implementado, é realizado através do processo de dilatação, seguido pela atualização do vetor de vizinhança e, posteriormente, a realização do processo de erosão.

### 3.3.5 Transformada Hit-or-miss

Seguindo a analogia existente de centro de massa na transformada hit-or-miss da morfologia matemática, o operador de vizinhança da transformada hit-or-miss é realizado através da informação única de formato contida no vetor de vizinhança. No trabalho aqui realizado é obtida primeiramente a definição, através da obtenção do centro de massa para objetos de uma imagem binária através do somatório e divisão dos pixels componentes da imagem, do centro de massa do objeto que se deseja encontrar, e posteriormente, a localização dentro da imagem na qual se deseja encontrar o objeto por pontos equivalentes. Tal procedimento é generalizado através do pseudocódigo a seguir:

Para todo pixel da imagem:

Se o pixel for um componente da imagem:

Então, Se, todos os componentes do vetor de vizinhança desse respectivo pixel são exatamente iguais ao componente de vizinhança do ponto definido como centro de massa do objeto desejado.

Então, o pixel continua como componente da imagem. Caso contrário, o pixel é retirado da imagem.

### 3.3.6 Extração de Fronteiras

Diferentemente do processo de extração de fronteiras realizado na morfologia matemática, o operador de vizinhança de extração de fronteiras é realizado através da informação única de formato contida no vetor de vizinhança. No trabalho aqui realizado é feita uma varredura dinâmica dentro de toda a imagem, em que é retirado cada pixel, componente da imagem, que não possua, em qualquer direção, nenhum componente no vetor de vizinhança que indique a borda. Tal procedimento é generalizado através do pseudocódigo a seguir:

Para todo pixel da imagem:

Se o pixel for um componente da imagem:

Então, Se, o vetor de vizinhança do respectivo pixel não possuir, em qualquer uma das oito direções, o componente que representa a borda

Então, o pixel é retirado da imagem. Caso contrário, o pixel continua como componente da imagem.

### 3.3.7 Afinamento

Analogamente ao processo de afinamento na morfologia matemática, o operador de vizinhança de afinamento é realizado através da informação contida no vetor de vizinhança. No trabalho aqui realizado é feita uma varredura dinâmica dentro de toda a imagem, em que para cada pixel, componente da imagem, é realizada uma comparação dos componentes do vetor de vizinhança nas oito direções, recursivamente de acordo com oito determinados formatos emulando os elementos estruturantes encontrado na morfologia matemática.

Caso esse formato seja identificado o pixel em questão é então retirado da imagem, e ao final de cada procedimento com um respectivo elemento, o vetor de vizinhança é então atualizado. O

processo é realizado até que seja encontrada a condição de parada, seja ela apenas uma aplicação consecutiva dos oito elementos ou até que nenhuma mudança seja realizada na imagem. Tal procedimento é generalizado através do pseudocódigo a seguir:

Caso a condição de parada não seja encontrada:

Para todo pixel da imagem:

Se o pixel for um componente da imagem:

Então, Se, o componente do vetor de vizinhança do respectivo pixel segue o mesmo formato do respectivo elemento

Então, o pixel é retirado da imagem. Caso contrário, o pixel continua como componente da imagem

Por fim, o vetor de vizinhança é atualizado e o formato do elemento mudado para o próximo na seqüência.

### 3.3.8 Espessamento

Analogamente ao processo de espessamento da morfologia matemática, o operador de vizinhança de espessamento, aqui implementado, é realizado utilizando-se o operador de vizinhança de afinamento. Primeiramente é obtido o complemento da imagem inicial. Posteriormente, é utilizado o operador de vizinhança de afinamento no complemento da imagem inicial. E por final, é obtido o complemento do resultado alcançado pelo operador de vizinhança de afinamento.



## 4. Avaliação e Resultados

### 4.1 Base de Imagens

O conjunto de imagens utilizado para o teste de execução e posterior comparação dos operadores de vizinhança é composto por três imagens. Essas imagens (Fig. 10), comumente utilizadas, foram escolhidas de forma a melhor apresentar o tipo de operador utilizado e uma melhor comparação entre as duas técnicas apresentadas.

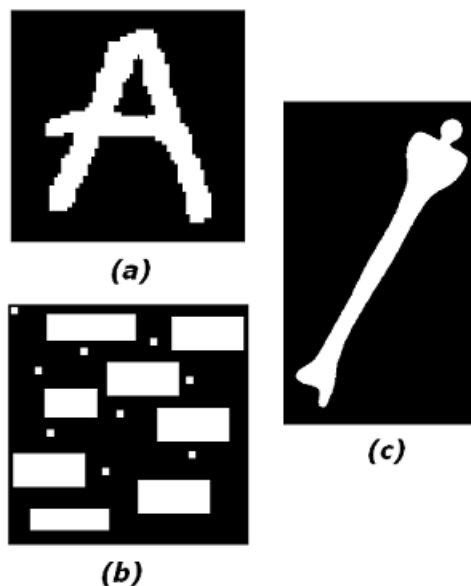


Figura 10: Imagens utilizadas como base.

### 4.2 Resultados

Os resultados aqui apresentados têm como objetivo a avaliação entre os operadores morfológicos existentes na morfologia matemática e os operadores de vizinhança aqui implementados. O processo de erosão, dilatação, assim como abertura e fechamento, e também de extração de fronteiras foram realizados tendo como imagem inicial a Fig.20(a), enquanto que na transformada hit-or-miss foi utilizada a Fig.20(b) na procura de um objeto quadrado de tamanho 3x3, e por fim no afinamento e espessamento, foi utilizada a Fig.20(c).

O resultado, aqui obtido, será apresentando visualmente através de um conjunto formado por uma seqüência de três imagens. A Figura 11(a) mostra a imagem inicial, posteriormente o resultado do operador morfológico específico na morfologia matemática é mostrado na Figura 11(b), e por final o resultado do correspondente operador de vizinhança é mostrando na Figura 11(c)

#### Erosão

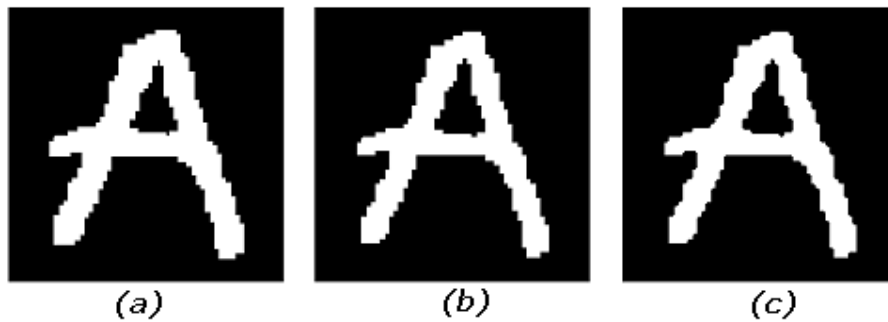


Figura 11: Resultado comparativo do processo de erosão.

#### Dilatação

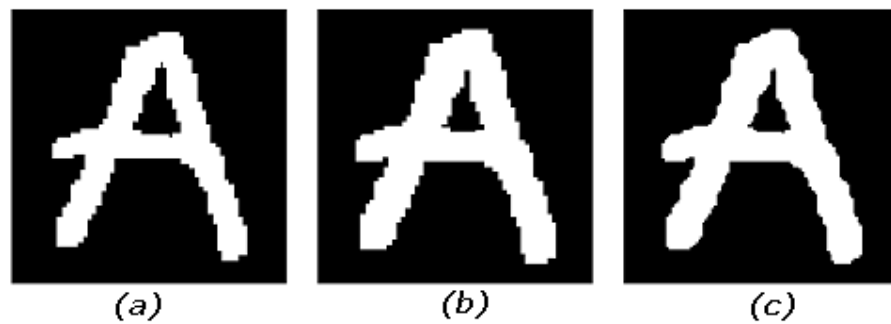


Figura 12: Resultado comparativo do processo de dilatação.

### Abertura

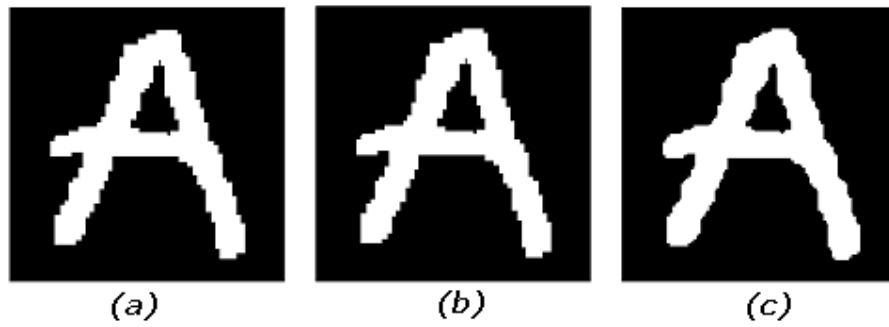


Figura 13: Resultado comparativo do processo de abertura.

### Fechamento

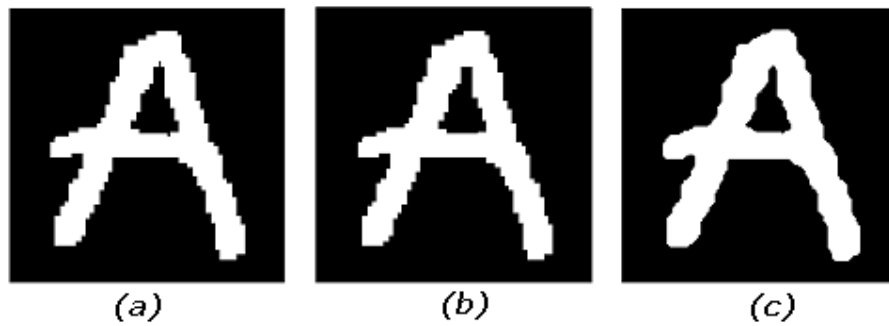


Figura 14: Resultado comparativo do processo de fechamento.

### Extração de Fronteiras

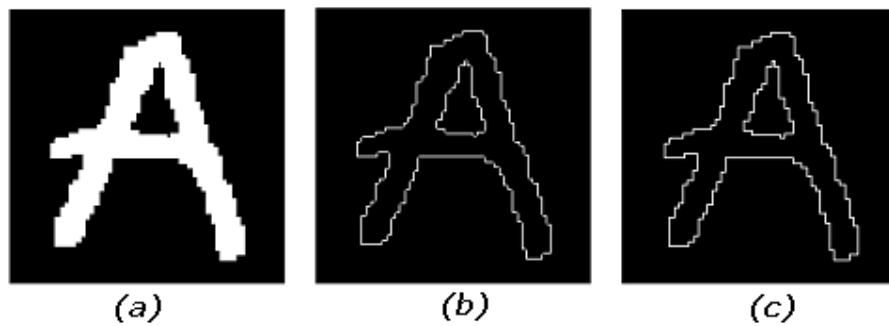


Figura 15: Resultado comparativo do processo de extração de fronteiras.

### Transformada Hit-or-miss

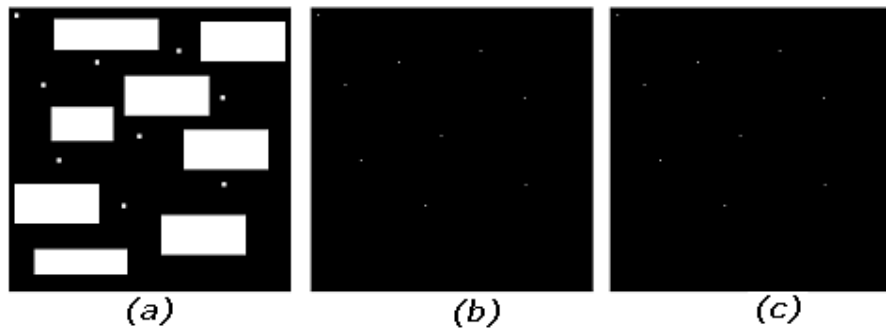


Figura 16: Resultado comparativo do processo da transformada *hit-or-miss*.

### Afinamento

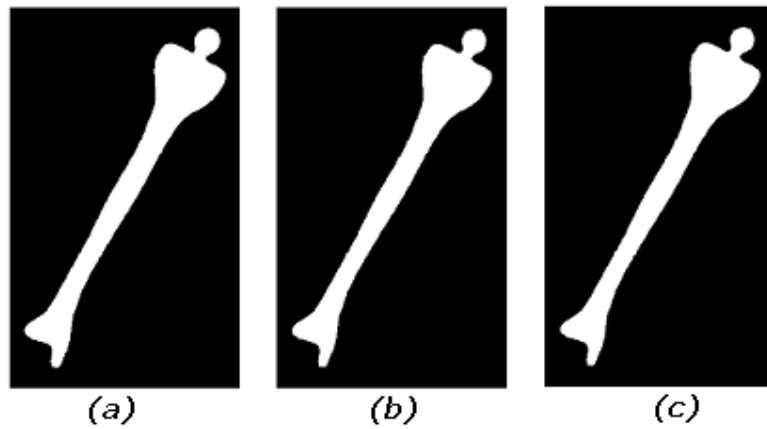


Figura 17: Resultado comparativo do processo de afinamento.

### Espessamento

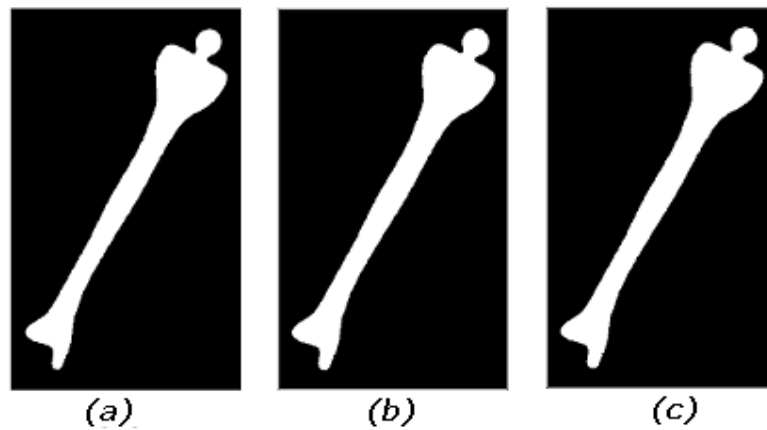


Figura 18: Resultado comparativo do processo de espessamento.

### 4.3 Avaliação

A avaliação entre essas duas abordagens será determinada através de parâmetros técnicos, como a necessidade de um processamento prévio, a preservação de informações, a utilização de um elemento estruturante externo, e a necessidade de um possível processamento adicional. Além disso, de um ponto de vista mais prático, também será analisado o tempo de execução de ambas as abordagens dentro de cada operador implementado.

Como foi apresentado antes, o operador de vizinhança demanda de um processamento prévio necessário para preenchimento do vetor de vizinhança para a representação da imagem. Tal procedimento se mostra bastante útil por preservar informações de formato da imagem. No entanto, em casos recursivos, como o afinamento e espessamento, se torna necessária também um processamento adicional de atualização do vetor de vizinhança ao passo que é realizada a seqüência do algoritmo.

De forma contrária, no caso do operador morfológico, não se faz necessário tal processamento prévio ou posterior processamento adicional por se utilizar de um elemento estruturante externo. No entanto tal procedimento se mostra preso a um elemento externo específico escolhido, com o objetivo de melhor resultado da operação. Tal análise pode ser vista de forma geral na tabela (Tab. 1) abaixo.

<b>Operadores</b> <b>Parâmetros</b>	<b>Morfológico</b>	<b>Vizinhança</b>
<i>Processamento Prévio</i>	✗	✓
<i>Preservar Informações</i>	✗	✓
<i>Elemento Externo</i>	✓	✗
<i>Processamento Adicional</i>	✗	✓

Tabela 1: Visão geral da análise técnica dos operadores.

Por fim, também foi realizada uma análise comparativa mais prática dos resultados, anteriormente apresentados, através do cálculo do tempo de execução, em milissegundos, de cada operação, para os operadores morfológico e de vizinhança. Através de tal análise, pode-se constatar a utilização do operador de vizinhança como de melhor escolha de uso, com reduções de tempo que vão desde pequenos valores de 18% até valores extremos de 92%. Tal informação, assim como o tempo de cada procedimento pode ser vista de forma completa na tabela (Tab. 2) abaixo.

<b>Operadores</b> <b>Processo</b>	<b>Vizinhança</b>	<b>Morfológico</b>	<b>Redução de Tempo</b>
<i>Erosão</i>	<b>15ms</b>	<b>62ms</b>	<b>76%</b>
<i>Dilatação</i>	<b>15ms</b>	<b>62ms</b>	<b>76%</b>
<i>Abertura</i>	<b>110ms</b>	<b>172ms</b>	<b>37%</b>
<i>Fechamento</i>	<b>125ms</b>	<b>188ms</b>	<b>34%</b>
<i>Extração de Fronteiras</i>	<b>32ms</b>	<b>93ms</b>	<b>66%</b>
<i>Hit-or-miss</i>	<b>47ms</b>	<b>547ms</b>	<b>92%</b>
<i>Afinamento</i>	<b>1359ms</b>	<b>2390ms</b>	<b>44%</b>
<i>Espessamento</i>	<b>2141ms</b>	<b>2594ms</b>	<b>18%</b>

Tabela 2: Análise do tempo de execução dos operadores e redução percentual do tempo.

## 5. Conclusão e Trabalhos Futuros

O presente trabalho apresentou uma extensão da utilização da abordagem do código de vizinhança usado na construção de um novo modelo de operadores de vizinhança na morfologia matemática, como apresentado em [3]. Através de uma breve avaliação comparativa entre tal abordagem, com os operadores de morfologia matemática usuais, pode-se constatar, através de uma análise técnica aqui definida e dados estatísticos do tempo de execução obtidos por esse trabalho, sua eficiência dentro do conjunto de imagens escolhidos.

No presente trabalho, novos desafios foram encontrados. Como exemplo, a atualização do vetor de vizinhança, ser realizada de forma local, ou seja, dentro apenas dos pontos associados pelo vetor de vizinhança ao ponto alterado, ao invés de global, com todos os pontos da imagem, como realizado. Além disso, dentro da aplicação da transformada hit-or-miss, serem utilizadas outras técnicas de escolha do ponto, ou eventuais pontos, de representação da imagem, utilizadas dentro da área de reconhecimento de padrão, de forma a estender tal função para um número de problemas mais abrangente. Tais mudanças podem ser levadas em conta em trabalhos futuros, de forma a melhorar ainda mais o desempenho de tal procedimento, através de parâmetros como o tempo de execução aqui utilizado.

## Referências

[1] GONZALEZ, R.C.; WOODS, R.E. **Digital Image Processing**. 2. ed. New York: Prentice Hall, 2002. 793 p.

[2] SHIH, F.K. **Image Processing and Mathematical Morphology Fundamentals and Applications**. CRC Press, 2009. 415 p.

[3] TSANG, I.R., TSANG, I.J., **Neighbourhood Vector as Shape Parameter for Pattern Recognition**. WCCI World Congress on Computational Intelligence – IJCNN 2006, pp.3204 – 3209, 2006.

[4] PCRaster Version 2 Manual Chapter 4. Cartographic Modelling. **Neighbourhood operations**. Em: <http://pcraster.geo.uu.nl/documentation/pcrman/x1280.htm>.



## Assinaturas

---

---

Tsang Ing Ren  
(Orientador)

---

Paulo Roberto Figueirôa Amorim  
(Aluno)