

LUIS RENATO WOISKI GUILHERME

**UMA ABORDAGEM DE
AFINAMENTO POR
APRENDIZAGEM ATRAVÉS DE
EXEMPLOS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Curitiba
2007

LUIS RENATO WOISKI GUILHERME

**UMA ABORDAGEM DE
AFINAMENTO POR
APRENDIZAGEM ATRAVÉS
DE EXEMPLOS**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: Visão, Imagem e Robótica

Orientador: Prof. Dr. Jacques Facon

Curitiba
2007

Guilherme, Luis Renato Woiski
UMA ABORDAGEM DE AFINAMENTO POR APRENDIZAGEM
ATRAVÉS DE EXEMPLOS. Curitiba, 2007.

Dissertação de Mestrado - Pontifícia Universidade Católica do Paraná Pro-
grama de Pós-Graduação em Informática Aplicada.

1. Afinamento 2. Aprendizagem através de exemplos 3. Avaliação de carac-
terísticas I. Pontifícia Universidade Católica do Paraná. Centro de Ciências
Exatas e Tecnologia. Programa de Pós-Graduação em Informática Apli-
cada.

Para meu pai e minha mãe. Sem essa dupla fantástica, não seria nem sombra do que sou hoje.

Agradecimentos

Novamente, e em especial, ao meu pai, sempre em meus pensamentos, e a minha mãe, fonte inesgotável de amor e sabedoria. Sem eles não chegaria onde estou hoje.

Ao meu grande amigo e mestre, o professor Jacques Facon, pelas incontáveis horas de paciência, companheirismo e dedicação. Mais que um professor, um grande amigo que terei por toda a vida.

Ao meu irmão Luis Daniel, por ser a pessoa que é, por pegar no meu pé, mas sempre me apoiar (do jeito dele, é claro) quando eu preciso. Sem esquecer da Adriana e da Aninha.

Aos meus colegas de faculdade e mestrado, os quais prefiro chamar de amigos: Andre Murbach Maidl e Rafael Sandim Kretschmar (nosso trio apavora!); Silvana Alves Cim (tirando as brigas por causa do futebol, é claro); Adriano Ricardo Duma (e seus milhões de e-mails com piadinhas sem graça); Andre Gustavo Hochuli (sempre contabilizando risadas gerais simplesmente por ser quem ele é); Jaime Dalla Valle Junior (fonte inesgotável de informações sobre as provas que estavam por vir).

Em especial, ao meu grande amigo Ricardo de Oliveira Abu Hana, pelas incontáveis horas de companheirismo, seja em relação a assuntos sérios, ou no Couto Pereira torcendo (é, voltamos!), ou simplesmente jogando conversa fora. Ele e sua namorada, Pollyana Almeida, sempre soube me ajudar nos momentos ruins, além de farrear comigo nos momentos bons. São um exemplo de amizade verdadeira.

Em especial, a minha grande amiga Lucianne Elise Takamura. Amiga não. O mais apropriado seria dizer que ela é a irmã que eu não tive. Amiga para todas as horas, que me aguentou durante todos esses anos, nos momentos de risadas e de reflexões. Prova viva que amizade verdadeira pode existir e que sinceridade sempre faz bem as pessoas. Só tenho a agradecer por tudo que passamos e por tudo que com certeza ainda passaremos juntos.

Aos meus amigos da iGoo, Rafael Kaminenko (meu irmão, amigo e camarada desde o começo da faculdade) e Daniel Bitencourt, e da Voxelz, Paulo Henrique Zung de Andrade (meu irmão, amigo e camarada desde a infância) e Carlos Eduardo Reikdal,

pela amizade e pelos momentos juntos, de apoio e de compreensão. Sem a ajuda desses caras, não conseguiria conciliar o trabalho com o mestrado. Sem esquecer, é claro, de suas namoradas e suas famílias. Obrigado!

A minha amiga Caroline Pompemaier, pela amizade, pelos conselhos, pelo apoio, e por mais um monte de coisas que provavelmente ela vai achar besteira, mas que para mim foram muito importantes.

A todo pessoal da minha família que reside em Londrina e Maringá. Essa família é fantástica e merece todo meu amor e admiração. Em especial, é claro, a minha tia Rosi e minha prima Rosemari.

Não posso esquecer outro grande professor, Gil Marcos Jess, que no período mais complicado de minha vida acadêmica me ajudou muito, me colocando no caminho certo, para eu alcançar tudo que alcancei até hoje.

Por fim, é claro que não posso deixar de agradecer à duas pessoas que conviveram muito comigo e me ajudaram a ser a pessoa que sou: Odette Duarte dos Santos Silva e Leci Aparecida Teixeira. A presença de vocês é marcante em minha vida. Vocês me ensinaram tantas coisas que seria impossível escrever tudo aqui sem duplicar o tamanho desta dissertação. Muito obrigado!

A todas as pessoas que de alguma forma cruzaram meu caminho e me ajudaram em minha caminhada, como o Sérgio Wagenheimer Lima, o André Gustavo Degraf Uchôa, o Horacio Andrés Legal Ayala, o Denis Hattori Matias, o Daniel Martineschen, entre muitos outros.

Ao glorioso Coritiba Football Club, por proporcionar tantos momentos de alegria.

Por fim, a Deus, por tudo que Ele fez, faz e sei que irá fazer, por mim e por todas as pessoas que amo.

Sumário

Agradecimentos	ii
Sumário	iv
Lista de Figuras	viii
Lista de Tabelas	xii
Lista de Abreviações	xiv
Resumo	xv
Abstract	xvi
Capítulo 1	
Introdução	1
1.1 Problemática	1
1.2 Objetivos	1
1.3 Desafio	3
1.4 Contribuição	5
1.5 Estrutura do documento	5
Capítulo 2	
Estado da arte	6
2.1 Histórico	6
2.2 Considerações iniciais	7
2.2.1 Imagem binária	7
2.2.2 Imagem em níveis de cinza	7
2.2.3 Vizinhaça	7
2.2.3.1 4-Vizinhaça	8
2.2.3.2 8-Vizinhaça	8
2.2.4 Conectividade	9
2.2.4.1 Número de Conectividade	9

2.3	Classificação dos algoritmos de afinamento	10
2.3.1	Algoritmos Seqüenciais	11
2.3.2	Algoritmos Paralelos	11
2.4	Algoritmos de afinamento	12
2.4.1	Algoritmo de Stentiford	13
2.4.1.1	Máscaras utilizadas	13
2.4.1.2	Passos do algoritmo	14
2.4.2	Algoritmo de Zhang-Suen	14
2.4.2.1	Definições teóricas	15
2.4.3	Algoritmo de Zhang-Wang	17
2.4.3.1	Definições teóricas	17
2.4.3.2	Funcionamento	20
2.4.4	Algoritmo de Holt	20
2.4.4.1	Definições teóricas	20
2.4.4.2	Staircase removal	22
2.4.4.3	Comparação de métodos	23
2.4.5	Algoritmo de Petrosino e Salvi	23
2.4.5.1	Definições teóricas	24
2.4.5.2	Comparação de métodos	27
2.4.6	Algoritmo utilizando morfologia matemática	27
2.4.6.1	Definições Básicas	28
2.5	Estudo comparativo	29
2.6	Comentários finais	31

Capítulo 3

Fundamentação teórica		33
3.1	Função distância	33
3.1.1	Definição teórica	33
3.1.1.1	Funcionamento	34
3.1.1.2	Algoritmo base	37
3.1.1.3	Abordagens	38
3.1.1.4	Função distância por morfologia matemática	38
3.2	Aprendizagem baseada em exemplos	40
3.2.1	Uma abordagem de segmentação de imagens por aprendizagem . . .	40
3.2.1.1	Mecanismo	40
3.2.1.2	Vantagens e desvantagens	40

3.2.2	Construção rápida de operadores morfológicos eficientes através de aprendizagem	41
3.2.2.1	Mecanismo	42
3.2.2.2	Vantagens e desvantagens	42
3.2.3	Redução de dimensionalidade utilizando entropia condicional média: aplicações em filtragem de imagens e em reconhecimento de texturas	43
3.2.3.1	Mecanismo	43
3.2.3.2	Vantagens e desvantagens	44
3.3	Classificador	45
3.3.1	Vizinho mais próximo - 1-NN	45
3.3.1.1	Distância Euclidiana	46
3.3.2	k vizinhos mais próximos - k -NN	46
3.3.2.1	Valor ideal de k	46
3.4	Comentários finais	47

Capítulo 4

Metodologia proposta	48	
4.1	Afinamento por aprendizagem através de exemplos	48
4.2	Metodologia	48
4.2.1	Fundamentos da metodologia proposta	49
4.2.2	Janelas	50
4.2.3	Extração das características	50
4.2.4	Aprendizagem	52
4.2.5	Tamanho ideal da janela	53
4.2.6	Matriz de classificação	55
4.2.7	Afinamento de novas imagens por aprendizagem	56
4.3	Comentários finais	57

Capítulo 5

Testes e resultados	58	
5.1	Base contendo as 26 letras do alfabeto	58
5.1.1	Metodologia de avaliação de resultados	63
5.1.2	Avaliação de características	63
5.1.3	Avaliação dos resultados	65
5.1.3.1	Função distância	71
5.1.3.2	Comparações com outros métodos	75

5.2	Base de imagens da Bíblia	80
5.2.1	Avaliação dos resultados	86
5.2.1.1	Função distância	91
5.2.1.2	Comparações com outros métodos	93
5.3	Comentários finais	93
Capítulo 6		
Conclusões		96
Referências Bibliográficas		100

Lista de Figuras

Figura 1.1	Exemplos de métodos de afinamento	2
Figura 1.2	Exemplo de erosão excessiva	3
Figura 1.3	Letra T pelo algoritmo de Stentiford	4
Figura 1.4	Letra X pelo algoritmo de Stentiford: perda de conectividade	4
Figura 1.5	Algoritmo de Zhang e Suen: número 8 com linhas destoantes (re- barbas)	4
Figura 1.6	Algoritmo de Zhang e Suen: letra V obtém aparência de Y	4
Figura 2.1	Vizinhança: $p_2, p_3, p_4, p_5, p_6, p_7, p_8$ e p_9 constituem a vizinhança de p_1	7
Figura 2.2	Vizinhança: 4-vizinhança	8
Figura 2.3	Vizinhança: 8-vizinhança	8
Figura 2.4	Número de Conectividade é igual 2	9
Figura 2.5	Máscaras usadas pelo algoritmo de Stentiford	13
Figura 2.6	Método de Zhang-Suen: máscara utilizada	15
Figura 2.7	Método de Zhang-Suen: Ou P_2 , ou P_4 ou P_8 deve ser um pixel branco	16
Figura 2.8	Método de Zhang-Suen: Ou P_2 , ou P_6 ou P_8 deve ser um pixel branco	16
Figura 2.9	Método de Zhang-Suen: Ou P_2 , ou P_4 ou P_6 deve ser um pixel branco	16
Figura 2.10	Método de Zhang-Suen: Ou P_4 , ou P_6 ou P_8 deve ser um pixel branco	17
Figura 2.11	Janelas do método de Zhang-Wang	18
Figura 2.12	Método de Holt: janela utilizada para vizinhança	21
Figura 2.13	Método de Holt: exemplo do número de vizinhos	22
Figura 2.14	Máscaras do método de Holt com staircase removal	22

Figura 2.15	Método de Holt: exemplo	23
Figura 2.16	Método de Petrosino e Salvi: máscara utilizada	24
Figura 2.17	Máscaras utilizadas no método de Petrosino e Salvi	26
Figura 2.18	Método de Petrosino: máscara 5×5	27
Figura 2.19	Comparação entre métodos de afinamento	28
Figura 2.20	Máscaras do método da Morfologia Matemática	28
Figura 2.21	Máscaras do método da Morfologia Matemática	29
Figura 2.22	Problemas ocorridos com o algoritmo de Stentiford	30
Figura 2.23	Imagem afinada através da morfologia matemática	31
Figura 2.24	Exemplos de um selo de envelope postal afinado através das várias abordagens	32
Figura 3.1	Exemplo da imagem resultante de uma função distância	34
Figura 3.2	Imagem exemplo submetida às técnicas de cálculo da função distância	34
Figura 3.3	Resultado do cálculo da função distância na imagem 3.2	35
Figura 3.4	Máscara do modo forward das abordagens Chamfer 3-4, Chessboard e City-block	36
Figura 3.5	Máscara do modo backward das abordagens Chamfer 3-4, Chess- board e City-block	36
Figura 3.6	Máscara do modo forward da abordagem Chamfer 5-7-11	36
Figura 3.7	Máscara do modo backward da abordagem Chamfer 5-7-11	37
Figura 3.8	Exemplo de imagem binária	37
Figura 3.9	Exemplo de máscara utilizada para a abordagem Chamfer 3-4, City- Block e Chessboard	38
Figura 3.10	Máscara da abordagem Chamfer 3-4	39
Figura 3.11	Máscara da abordagem Chamfer 5-7-11	39
Figura 3.12	Imagem amostra e solução ideal	41
Figura 3.13	Nova imagem amostra e solução ideal	41
Figura 3.14	Binarização obtida por aprendizagem	41
Figura 3.15	Exemplo de imagem de treinamento	42
Figura 3.16	Exemplo de imagem original, ideal e o resultado obtido	43
Figura 3.17	(a) baixa entropia (b) alta entropia	44
Figura 3.18	Imagem amostra para treinamento e sua solução ideal	44
Figura 3.19	Nova imagem; solução ideal; resultado utilizando 40% para apren- dizagem	45
Figura 3.20	Exemplo de k vizinhos mais próximos	46

Figura 4.1	Esquema geral da abordagem proposta de afinamento por aprendizagem	49
Figura 4.2	Exemplo de varredura para um pixel	51
Figura 4.3	Vetor de características [AYA04]	52
Figura 4.4	A etapa de aprendizagem e definição automática do tamanho da janela [AYA04]	53
Figura 4.5	Gráfico de total de erros $(\Sigma \varepsilon) \times k$ [AYA04]	55
Figura 4.6	Exemplo de matriz de classificação	56
Figura 4.7	Processo de afinamento de imagens	56
Figura 5.1	Exemplos de letras da base utilizada	58
Figura 5.2	Imagens ideais das letras mostradas na figura 5.1	59
Figura 5.3	Função distância das letras da figura 5.1 através de várias abordagens	60
Figura 5.4	Exemplo de letras afinadas com o par amostra/ideal sendo a letra A	61
Figura 5.5	Imagens submetidas para extração de características	61
Figura 5.6	Nova imagem submetida para afinar com o que foi aprendido com a imagem 5.5	62
Figura 5.7	Resultado ideal do afinamento da imagem 5.6	62
Figura 5.8	Afinamento pelo método de aprendizagem	62
Figura 5.9	Afinamento da base de letras para cada característica isolada e o valor de k vizinhos mais próximos	66
Figura 5.10	Ruído da base de letras para cada característica isolada e o valor de k vizinhos mais próximos	67
Figura 5.11	Taxa de afinamento da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$. .	68
Figura 5.12	Taxa de ruído da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$. .	69
Figura 5.13	Taxa de afinamento da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 11$. .	70
Figura 5.14	Taxa de ruído da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 11$. .	71
Figura 5.15	Taxa de afinamento para o melhor conjunto de características (mvs) de acordo com o valor de k	72
Figura 5.16	Taxa de ruído para o melhor conjunto de características (mvs) de acordo com o valor de k	72
Figura 5.17	Taxa de afinamento para cada função distância	73

Figura 5.18	Taxa de ruído para cada função distância	74
Figura 5.19	Comparação entre as taxas de afinamento dos métodos testados . .	75
Figura 5.20	Comparação entre as taxas de ruído dos métodos testados	76
Figura 5.21	Afinamento pelo método de Holt com staircase removal	76
Figura 5.22	Afinamento pelo método de Holt sem staircase removal	76
Figura 5.23	Afinamento pelo método de Mandeville	76
Figura 5.24	Afinamento pelo método Morfológico	77
Figura 5.25	Afinamento pelo método de Stentiford	77
Figura 5.26	Afinamento pelo método de Zhang-Suen	77
Figura 5.27	Afinamento pelo método de Zhang-Wang	77
Figura 5.28	Detalhes da comparação entre os métodos testados	78
Figura 5.29	Exemplo de uma página da Bíblia de Gutenberg	80
Figura 5.30	Exemplo de imagem a ser usada nesta abordagem	81
Figura 5.31	Função distância da figura 5.30(b) através de várias abordagens . .	82
Figura 5.32	Função distância da figura 5.30(b) através de várias abordagens . .	83
Figura 5.33	Função distância da figura 5.30(b) através de várias abordagens . .	84
Figura 5.34	Exemplo de imagens afinadas através da Aprendizagem	85
Figura 5.35	Afinamento da base imagens para cada característica isolada e o valor de k vizinhos mais próximos	87
Figura 5.36	Ruído da base de imagens cada característica isolada e o valor de k vizinhos mais próximos	88
Figura 5.37	Taxa de afinamento da base de imagens com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$	89
Figura 5.38	Taxa de ruído da base de imagens com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$. .	89
Figura 5.39	Taxa de afinamento para o melhor conjunto de características (mc) de acordo com o valor de k	90
Figura 5.40	Taxa de ruído para o melhor conjunto de características (mc) de acordo com o valor de k	91
Figura 5.41	Taxa de afinamento para cada função distância	92
Figura 5.42	Taxa de ruído para cada função distância	92
Figura 5.43	Comparação entre as taxas de afinamento dos métodos testados . .	94
Figura 5.44	Comparação entre as taxas de ruído dos métodos testados	94
Figura 5.45	Exemplo de imagens afinadas através dos métodos analisados nesta abordagem	95

Lista de Tabelas

Tabela 2.1	Estudo comparativo entre os métodos apresentados	30
Tabela 5.1	Resultados do afinamento em letras para a média de acordo com o valor de k	65
Tabela 5.2	Resultados do afinamento em letras para a variância de acordo com o valor de k	65
Tabela 5.3	Resultados do afinamento em letras para a obliquidade de acordo com o valor de k	65
Tabela 5.4	Resultados do afinamento em letras para a curtose de acordo com o valor de k	66
Tabela 5.5	Taxas de afinamento e ruído para as 15 combinações possíveis das quatro características utilizadas nesta abordagem, para o valor $k = 1$. . .	68
Tabela 5.6	Taxas de afinamento e ruído para as 15 combinações possíveis das quatro características utilizadas nesta abordagem, para o valor $k = 11$. . .	69
Tabela 5.7	Melhor combinação de características de acordo com o valor de k .	71
Tabela 5.8	Comparação entre as funções distância com as características mvs e $k = 1$	73
Tabela 5.9	Comparação entre os métodos de afinamento analisados nesta abordagem	75
Tabela 5.10	Resultados do afinamento das imagens para a média de acordo com o valor de k	86
Tabela 5.11	Resultados do afinamento das imagens para a variância de acordo com o valor de k	86
Tabela 5.12	Resultados do afinamento das imagens para a obliquidade de acordo com o valor de k	87
Tabela 5.13	Resultados do afinamento das imagens para a curtose de acordo com o valor de k	87

Tabela 5.14	Taxas de afinamento e ruído para as 15 combinações possíveis das quatro características utilizadas nesta abordagem, para o valor $k = 1$. . .	88
Tabela 5.15	Melhor combinação de características de acordo com o valor de k . . .	90
Tabela 5.16	Comparação entre as funções distância com as características mc e $k = 1$	91
Tabela 5.17	Comparação entre os métodos de afinamento analisados nesta abordagem	93

Lista de Abreviações

k -NN	<i>k nearest neighbor (k vizinhos mais próximos)</i>
μ	<i>média</i>
σ^2	<i>variância</i>
c	<i>curtose</i>
s	<i>obliquidade</i>
sc	<i>obliquidade e curtose</i>
v	<i>variância</i>
vc	<i>variância e curtose</i>
vs	<i>variância e obliquidade</i>
vsc	<i>variância, obliquidade e curtose</i>
m	<i>média</i>
mc	<i>média e curtose</i>
ms	<i>média e obliquidade</i>
msc	<i>média, obliquidade e curtose</i>
mv	<i>média e variância</i>
mvc	<i>média, variância e curtose</i>
mvs	<i>média, variância e obliquidade</i>
$mvsc$	<i>média, variância, obliquidade e curtose</i>

Resumo

A proposta deste trabalho é a criação de uma abordagem de afinamento de imagens através de métodos de aprendizagem. O processo de aprendizagem utiliza-se de duas imagens, a imagem original e sua versão ideal afinada. A partir destas imagens, é gerada uma matriz de classificação que contempla cada pixel da imagem. Para afinar outra imagem similar, é reutilizado o conhecimento adquirido que é armazenado na matriz de classificação. A nova imagem é afinada com o auxílio do classificador k -NN. O classificador busca, para cada pixel desta nova imagem, a melhor solução na matriz de classificação. As principais vantagens desta abordagem são: a reprodução fiel dos objetivos do usuário; o uso de somente duas imagens para a aprendizagem, independente de parâmetros heurísticos e interação com um usuário experiente após o processo de aprendizagem. Já as desvantagens são: a dificuldade em encontrar o melhor conjunto de características, o tempo de processamento variável e a qualidade do resultado ser dependente do cálculo da função distância empregado.

Palavras-chave: Afinamento, aprendizagem baseada em exemplos

Abstract

The proposal of this work is an image through learning methods. The learning process works with two images, the original and its thinned ideal version. From these images, a classifier matrix is generated to contemplate each pixel of the image. To thin another similar image, the acquired knowledge that is stored in the classifier matrix is reused. The new image is thinned through a classifier called k -NN. The classifier searches, for each pixel of this new image, the best solution in the classifier matrix. The main advantages of this work are: the faithful reproduction of the user objectives; the use of only two images for the learning; the unnecessary use of heuristical parameters and interaction with an experienced user after the learning process. The disadvantages are: the difficulty in finding the best set of features, the variable time of processing and the fact that the result is dependent of distance function.

Keywords: Thinning, learning by example

Capítulo 1

Introdução

O uso dos computadores desencadeou uma revolução em todo o âmbito científico no que se diz respeito às novas tecnologias que poderiam ser geradas com o auxílio destas máquinas. Desde o início desta revolução tecnológica, os computadores têm sido empregados na tentativa de reproduzir o comportamento humano em relação à visão. Junto a isso, houve a crescente necessidade de reduzir o volume de informações a serem processadas. Com isso, surgiu uma área na computação chamada de *Processamento Digital de Imagens* ou, como é mais conhecido, *Processamento de Imagens*, que visa interpretar as inúmeras informações recebidas na forma de imagens.

1.1 Problemática

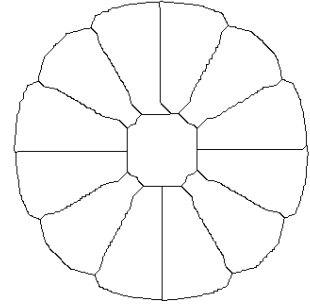
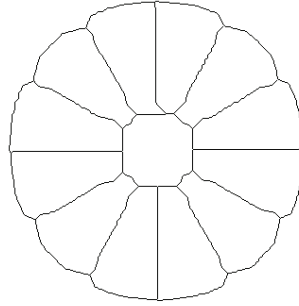
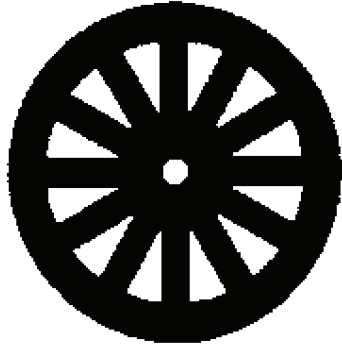
O *afinamento*, objeto de estudo desta pesquisa, pode ser enquadrado antes da etapa de *interpretação* de dados em um reconhecimento. Como tal, deve fornecer informações suficientes e corretas para que o reconhecimento seja feito da melhor maneira possível.

Os métodos de afinamento atuais muitas vezes são incapazes de fornecer um resultado que reproduza fielmente a forma da imagem original, como pode ser visto na figura 1.1.

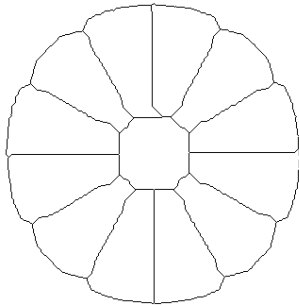
1.2 Objetivos

Esta pesquisa objetiva propor um novo mecanismo de afinamento por aprendizagem baseada em exemplos capaz de:

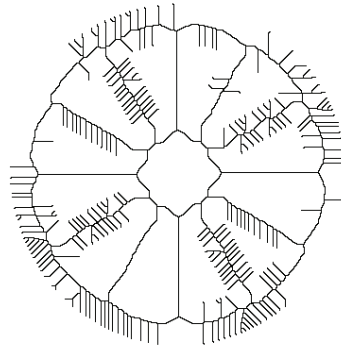
- gerar um esqueleto de espessura de um pixel (problema encontrado no resultado das figuras 1.1(c), onde o algoritmo de Holt gera um esqueleto com dois pixels de largura em alguns pontos);



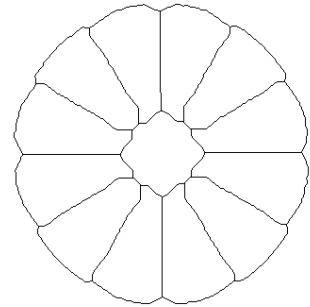
(a) figura geométrica complexa e (b) Holt com staircase removal (c) Holt sem staircase removal simétrica



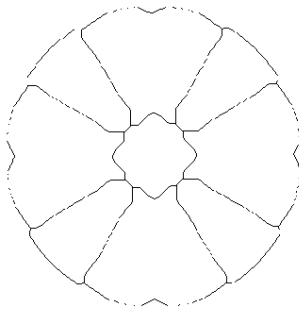
(d) Mandeville



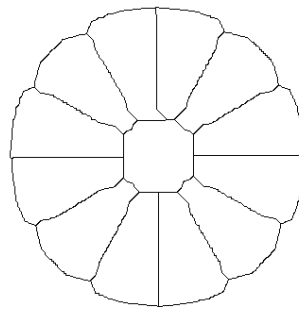
(e) Morfológico



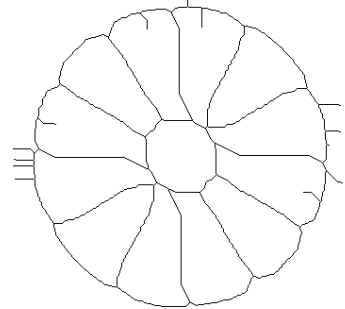
(f) Petrosino



(g) Stentiford



(h) Zhang e Suen



(i) Zhang e Wang

Figura 1.1: Exemplos de métodos de afinamento

- preservar a simetria dos objetos independentemente de sua posição (como Holt na figura 1.1(b), Mandeville na figura 1.1(d) e Zhang e Suen na figura 1.1(h), em que a simetria das regiões do objeto não são preservadas em relação à figura original 1.1(a));
- gerar um esqueleto homotópico sem pixels redundantes;
- minimizar rebarbas (problema encontrado no resultado da figura 1.1(e), onde o processo Morfológico faz com que o resultado final tenha muitas rebarbas);
- produzir linhas perfeitamente conectadas (problema encontrado na figura 1.1(g), a qual mostra o resultado do algoritmo de Stentiford com perda de conectividade);
- ser tão rápida quanto os métodos de sub-iterações hoje existentes;
- prevenir a erosão excessiva (como na figura 1.2, onde o resultado do algoritmo de esqueletização está provocando uma erosão excessiva);
- atingir os objetivos do usuário;

Afim de explicar os objetivos desta pesquisa, é usada uma figura complexa e simétrica (figura 1.1(a)) para ilustrar os diversos problemas e distorções geradas pelos algoritmos atuais.

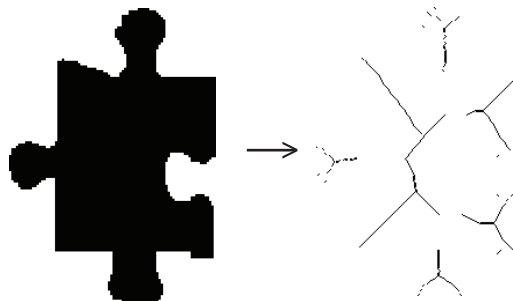


Figura 1.2: Exemplo de erosão excessiva

1.3 Desafio

O processo de afinamento de imagens é um processo custoso computacionalmente e sujeito a deformações. Alguns algoritmos, como o de *Stentiford* [SM83], funcionam apenas para algumas imagens, como visto na imagem 1.3. Para outras, os objetos afinados



Figura 1.3: Letra T pelo algoritmo de Stentiford

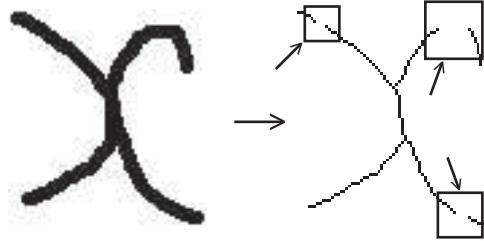


Figura 1.4: Letra X pelo algoritmo de Stentiford: perda de conectividade

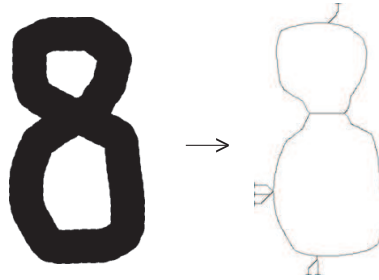


Figura 1.5: Algoritmo de Zhang e Suen: número 8 com linhas destoantes (rebarbas)

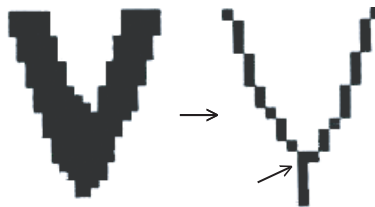


Figura 1.6: Algoritmo de Zhang e Suen: letra V obtém aparência de Y

apresentam uma perda de conectividade entre seus pixels (figura 1.4), tornando a imagem resultante inutilizável, devido à perda de informação.

Em outros algoritmos, como o de *Zhang e Suen* [ZS84], a imagem resultante mostra uma série de distorções, como linhas destoantes (figura 1.5) e perda de vértices (figura 1.6) (neste caso, a letra V toma a forma da letra Y), sendo o resultado totalmente dependente de uma imagem original de boa qualidade. Há ainda algoritmos [PS00] que exigem muito processamento devido ao excesso de formulações matemáticas utilizadas ao longo

do processo, tornando-os mais lentos em comparação a outros citados anteriormente.

1.4 Contribuição

Tendo como base todos os problemas apresentados na seção 1.3, a grande contribuição que este trabalho pode fornecer à área de processamento de imagens será um método de afinamento baseado em exemplos capaz de gerar um esqueleto homotópico de espessura de um pixel, sem rebarbas e sem a perda de conectividade, preservando a simetria dos objetos. Este método deve ser rápido tanto no processamento quanto na aprendizagem, afim de gerar resultados o mais rapidamente possível.

Com esta técnica, espera-se aprimorar o processo de afinamento de imagens. Conseqüentemente, todo o processamento a ser realizado após afinamento deverá fornecer resultados mais precisos. Além disso, espera-se que toda a análise de métodos de afinamento e dos resultados realizada neste documento torne-se um exemplo para futuros trabalhos, servindo como base e ajudando a comunidade científica a produzir documentos cada vez mais claros e completos.

1.5 Estrutura do documento

Este documento está dividido da seguinte forma: o capítulo 2 aborda o estado da arte, mostrando as técnicas mais conhecidas de afinamento; o capítulo 3 mostra alguns métodos de aprendizagem através de exemplos e também um pouco sobre o k -NN e o cálculo da função distância; o capítulo 4 mostra a metodologia proposta neste trabalho; por fim, o capítulo 5 mostra os testes realizados com a abordagem proposta.

Capítulo 2

Estado da arte

Uma importante etapa de processamento de imagens é a determinação do esqueleto por afinamento da imagem a ser processada. De modo geral, este processo consiste em realizar um “desgaste” dos objetos que formam a imagem, até que cada objeto seja representado apenas por uma linha simples, de largura de um pixel¹. A imagem resultante deste processo contém apenas os esqueletos dos objetos contidos na imagem.

2.1 Histórico

O processo de afinamento começou a ser mais utilizado na década de 50, quando foi usado para a criação de algoritmos de compressão. Já nos anos 60, a *biomedicina*² descobriu que as técnicas de afinamento seriam muito úteis para contar e medir as células brancas do sangue. O algoritmo de *Shrink* foi utilizado para identificar células anormais [PRE61]. A partir daí, houve uma explosão de aplicações nesta área, desde a análise de cromossomos, a análise de imagens de raio-x [PD79] e de artérias coronárias [NS86].

Já em outras áreas, o afinamento vem sendo utilizado em sistemas de reconhecimento de retina, classificação de impressões digitais [MF75], análise visual de peças industrializadas [MJ77], confecção de placas de circuito impresso [YD88], entre outras. A grande vantagem da utilização do processo de afinamento é a redução que ocorre na quantidade de dados a serem processados. Utilizando este processo, os padrões utilizados para a análise se aproximam da concepção humana, permitindo uma análise estrutural simplificada e, conseqüentemente, apropriada para os algoritmos de reconhecimento.

¹O pixel é o menor elemento em um dispositivo de exibição (como por exemplo um monitor), ao qual é possível atribuir-se uma cor. De uma forma mais simples, um pixel é o menor ponto que forma uma imagem digital, sendo que o conjunto de milhares de pixels formam a imagem inteira.

²Disciplina combinada de biologia e medicina para análise das tolerâncias humanas e proteção contra variações de ambiente.

2.2 Considerações iniciais

Algumas considerações se fazem necessárias para o pleno entendimento das próximas seções deste capítulo, sendo estas mostradas a seguir.

2.2.1 Imagem binária

Uma imagem nada mais é do que um conjunto de todos os pontos no plano euclidiano. Um conjunto é finito se ele contém apenas um número finito de elementos. Por exemplo, S é uma imagem finita e $B = \{0, 1\}$. Uma imagem binária finita é uma função V mapeada em S dentro de B . Então a imagem binária é unicamente definida no primeiro plano no conjunto P (ou conjunto dos pixels pretos). Enquanto todos os conjuntos complementares serão denominados de segundo plano (ou conjunto dos pixels brancos).

$$P = \{p \mid p \in S \wedge V(p) = 1\} \quad (2.1)$$

2.2.2 Imagem em níveis de cinza

Enquanto o conjunto de pontos possíveis em uma imagem binária é $\{0, 1\}$, em uma imagem em níveis de cinza este conjunto pode possuir 256 variações de cor, ou seja, S é uma imagem finita e $B = \{0, \dots, 255\}$. A imagem em níveis de cinza é uma função mapeada em S dentro de B .

2.2.3 Vizinhaça

Os pixels $p_2, p_3, p_4, p_5, p_6, p_7, p_8$ e p_9 mostrados na figura 2.1 representam a vizinhaça do pixel central p_1 [PR99]. O pixel p_1 é o pixel central em análise, então ele é preto.

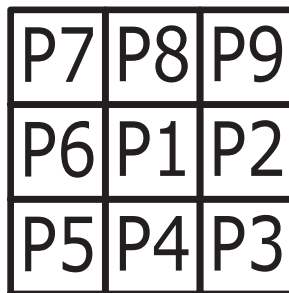


Figura 2.1: Vizinhaça: $p_2, p_3, p_4, p_5, p_6, p_7, p_8$ e p_9 constituem a vizinhaça de p_1

2.2.3.1 4-Vizinhança

A 4-vizinhança de um pixel p é constituída de seus dois vizinhos horizontais e os dois vizinhos verticais. Considerando que o pixel p possui as coordenadas cartesianas (x, y) , então seus dois vizinhos horizontais são os pixels de coordenadas $(x+1, y)$ e $(x-1, y)$. Já os pixels verticais possuem as coordenadas $(x, y+1)$ e $(x, y-1)$. A figura 2.2 mostra a 4-vizinhança.

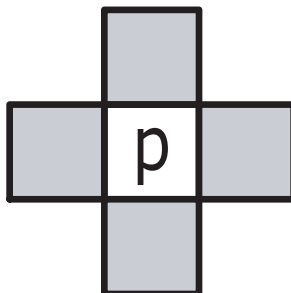


Figura 2.2: Vizinhança: 4-vizinhança

2.2.3.2 8-Vizinhança

Junto aos quatro vizinhos horizontais e verticais, os quatro vizinhos em diagonal ao pixel p constituem a 8-vizinhança. Usando novamente o sistema de coordenadas cartesianas, a 8-vizinhança engloba, além das coordenadas dos pixels da 4-vizinhança, as coordenadas $(x-1, y-1)$, $(x-1, y+1)$, $(x+1, y-1)$ e $(x+1, y+1)$. A figura 2.3 mostra a 8-vizinhança.

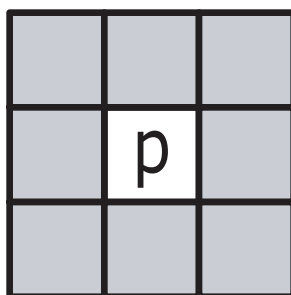


Figura 2.3: Vizinhança: 8-vizinhança

Tomando como base a figura 2.1, os pixels p_2, p_4, p_6 e p_8 formam a 4-vizinhança de p_1 e são 4-adjacentes a p_1 . Já os pixels $p_2, p_3, p_4, p_5, p_6, p_7, p_8$ e p_9 formam a 8-vizinhança do pixel p_1 , e são 8-adjacentes a p_1 .

2.2.4 Conectividade

Conectividade [ZS84] é a característica mais importante em um processo de afinamento. Dois pixels são ditos conexos se existe uma seqüência de pixels que os liga, tal que dois pixels consecutivos dessa seqüência satisfaçam a condição de *conectividade imediata*. Esta conectividade traduz o fato que os pixels têm características comuns e que são vizinhos imediatos. Os pixels conexos de uma região devem satisfazer o predicado associado que traduz o critério segundo o qual a região é construída. Esse predicado pode ser definido como uma função $F_c(\bullet)$ cujo domínio de definição pode variar muito e onde os valores dependem:

- do pixel $p(xp, yp)$ onde a função é calculada;
- de uma vizinhança desse pixel p , $V(p)$ (em geral 8-vizinhança);
- do valor do pixel p definindo uma região em formação $G(p)$;

No caso de imagens binárias, o valor da função $G(\bullet)$ do pixel p é simplesmente o valor binário deste pixel, ou seja, 0 ou 1. Já nas imagens em nível de cinza, o valor da função $G(\bullet)$ do pixel p está contido no conjunto $\{0, \dots, 255\}$.

2.2.4.1 Número de Conectividade

O número de conectividade [ZS84] é definido como sendo o número de transições de branco para preto entre os pixels, na região de pixels que circunda o pixel central p .

$$CN(p) = \sum_{k=0,2,4,6} \overline{p[k]} * (p[k+1] \cup p[k+2]) \quad (2.2)$$

No exemplo mostrado na figura 2.4, as setas mostram duas transições de branco para preto, fazendo com que o número de conectividade desta janela seja igual a 2.

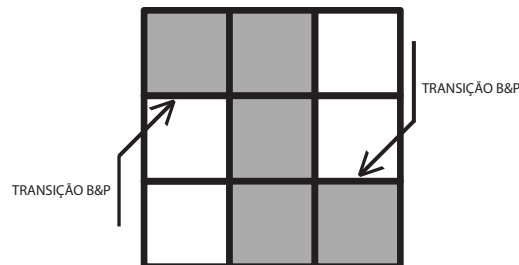


Figura 2.4: Número de Conectividade é igual 2

2.3 Classificação dos algoritmos de afinamento

Com o crescente estudo sobre métodos de afinamento de imagens, surgiram maneiras de se classificar os algoritmos de afinamento. Uma destas maneiras, criada em 1966 por Rosenfeld e Pfaltz [RP66], classifica os algoritmos de afinamento em dois grupos principais:

- Paralelos;
- Seqüenciais.

De acordo com [LLS92], um algoritmo seqüencial realiza uma análise dos pixels para verificar se estes podem ser excluídos. Esta análise possui uma seqüência fixa de operações a cada iteração. O pixel analisado somente será excluído na n -ésima iteração caso os resultados de todas as operações que foram realizadas até aquele momento tenham sido satisfatórias.

Já nos algoritmos paralelos, excluir um pixel na n -ésima iteração depende apenas dos pixels da iteração $(n - 1)$. Assim sendo, todos os pixels podem ser analisados independentemente e com este tipo de algoritmo pode ser utilizado em sistemas com mais de um processador.

Ambos os tipos de algoritmos possuem vantagens e desvantagens, sendo que, antes de se utilizar algum, deve ser realizada uma análise para descobrir o que melhor se adapta ao problema em questão. Os algoritmos seqüências produzem um esqueleto mais fiel no que diz respeito à conectividade e conservação da topologia dos objetos [SR71], mas, como dito anteriormente, com um esforço e um custo de processamento maior. Os algoritmos paralelos são mais rápidos que os seqüenciais, mas possuem o agravante que muitas vezes produzem um resultado de menor qualidade, em relação ao que seria produzido através do processo seqüencial.

Independente do processo a ser utilizado, é necessário que o algoritmo tenha a capacidade de:

- gerar o esqueleto de um pixel de largura;
- prevenir a erosão excessiva, que pode causar perda de conectividade;
- preservar a topologia dos objetos;
- gerar um esqueleto homotópico;
- minimizar rebarbas;
- não remover pontos finais;

2.3.1 Algoritmos Seqüenciais

Como dito na seção 2.3, os algoritmos que excluem sucessivamente as camadas de pixels dos objetos da imagem até que eles se encontrem com apenas um pixel de largura, ou seja, afinados, são chamados de *algoritmos seqüenciais*.

A exclusão de um pixel p depende única e exclusivamente da configuração dos pixels ao redor de p ou, como é mais conhecida, da janela de p . Através de um algoritmo seqüencial, os pixels da borda da janela são analisados em uma ordem específica, que pode variar em função do algoritmo. O resultado desta análise vai resultar na exclusão ou não do pixel. Esta análise pode ser feita de duas formas [LLS92]:

- Varredura (*raster scans*): que varre a imagem, por exemplo, da esquerda para direita e de cima para baixo;
- Verificação do contorno (*contour following*): que segue o contorno dos objetos.

Um pixel, quando passível de exclusão, é marcado para que seja excluído apenas no final da iteração [LLS92]. Este procedimento evita com que haja a remoção indevida dos pixels. Para que um pixel p possa ser removido, ele deve satisfazer as seguintes propriedades:

- p é preto;
- p não está isolado nem constitui um ponto final;
- p é um pixel de borda, ou seja, tem como vizinhança um ponto branco.

2.3.2 Algoritmos Paralelos

Diferentemente do que acontece nos algoritmos seqüenciais (seção 2.3.1), nos algoritmos paralelos os pixels são excluídos independentemente dos resultados obtidos na iteração anterior. Esta característica permite que o algoritmo seja implementado em um sistema de processamento paralelo, onde os pixels que satisfaçam as condições de exclusão podem ser removidos simultaneamente.

Em contrapartida, alguns algoritmos paralelos têm dificuldade em manter a conectividade (ver seção 2.2.4). Um exemplo clássico seria tentar afinar uma linha contendo dois pixels de largura. Após o processo de afinamento, esta linha poderia vir a desaparecer, como acontece com o algoritmo de Stentiford [SM83]. Por este motivo, utiliza-se vizinhanças 3×3 (ver seção 2.2.3), dividindo cada iteração em sub-iterações (sub-ciclos),

nas quais apenas um subconjunto de pixels de borda são considerados na remoção. Ao final de cada iteração, a imagem é atualizada com as remoções efetuadas durante o ciclo.

Existem várias maneiras de remover um pixel utilizando sub-iterações. Em uma delas, são usados quatro sub-ciclos para cada pixel, como em [SR71] e [RD76]. Outra forma de realizar este processo é através de 2 sub-iterações apenas, como em [SR71] e [ZS84], onde uma das iterações os pixels de contorno situados acima e à direita do pixel testado, e outra iteração exclui os pixels situados abaixo e à esquerda do pixel testado.

Uma outra maneira de processar imagens, através de algoritmos de sub-ciclos, é mapeá-la em forma de sub-áreas³. Recentemente, algoritmos com uma sub-iteração foram implementados, mas estes utilizam uma grande quantidade de informações para preservar a conectividade dos objetos [LLS92], tornando-os lentos.

2.4 Algoritmos de afinamento

A geração de um esqueleto digital é, normalmente, um dos primeiros passos em sistemas de visão computacional, quando o objetivo é extrair características de um objeto em uma imagem. Um esqueleto de um objeto visa representar a forma do objeto em um número menor de pixels no qual todos eles são necessários. Com essa idéia, o esqueleto deve ter todas as informações contidas na imagem original, como a posição, a orientação e o comprimento dos segmentos.

Os algoritmos de afinamento têm como característica a repetição do processo de eliminação de todos os pontos de contorno que foram identificados como sendo passíveis de eliminação, ou seja, efetuar a remoção de todos os pixels redundantes de uma imagem produzindo uma simplificação do objeto, com a largura de um único pixel. Para o algoritmo ser eficiente, o número de repetições ou iterações deve ser igual à metade da largura máxima dos objetos presentes na imagem, sobre a restrição de evitar erosão excessiva e manter as curvas as mais perfeitas possíveis [PS00].

O maior problema dos algoritmos de afinamento é determinar, com exatidão, quais os pixels redundantes na imagem. Para isso, o objeto resultante do afinamento deve respeitar as seguintes propriedades:

- As regiões afinadas precisam ter um pixel de largura;
- Os pixels que formam o esqueleto precisam permanecer próximos do centro da região de cruzamento de regiões;

³Sub-área ou zoneamento (do inglês “zoning”) é o método que separa a imagem em n zonas de tamanhos iguais, para serem tratadas individualmente.

- É necessário que os pixels do esqueleto formem o mesmo número de regiões que a imagem original apresentava (homotopia).

Portanto, o algoritmo de afinamento deve respeitar as seguintes propriedades:

- Deve manter a conectividade dos objetos;
- Não deve remover pontos extremos;
- Não deve causar erosão excessiva.

Nas seções a seguir, serão apresentadas as descrições dos principais métodos de afinamento, tais como Stentiford (seção 2.4.1), Zhang-Suen (seção 2.4.2), Zhang-Wang (seção 2.4.3), Holt (seção 2.4.4), Petrosino e Salvi (seção 2.4.5) e Morfológico (seção 2.4.6). Os algoritmos comparados são seqüenciais, de acordo com a classificação mostrada na seção 2.3

2.4.1 Algoritmo de Stentiford

Em 1983, Stentiford e Mortimer [SM83] introduziram uma nova abordagem para os algoritmos de afinamento: o conceito de máscaras.

2.4.1.1 Máscaras utilizadas

Para realizar o afinamento de uma imagem binária, o algoritmo utiliza-se de quatro máscaras (figura 2.5), que devem ser aplicadas sucessivamente e de forma ordenada na imagem.

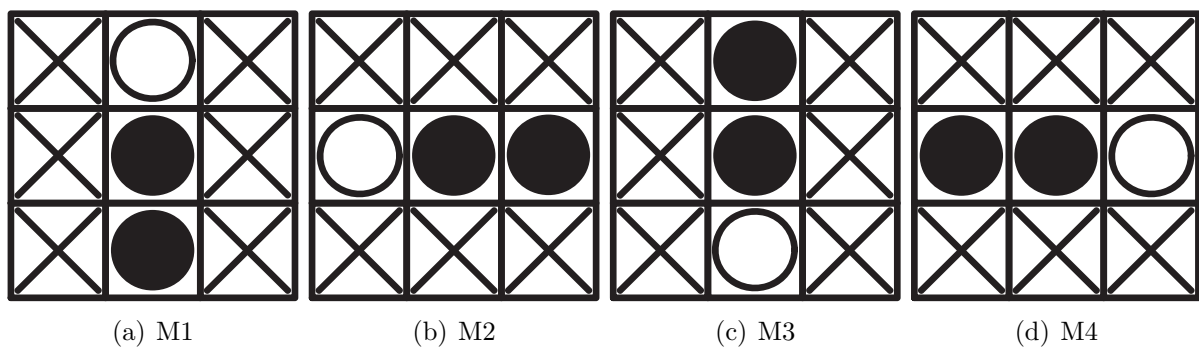


Figura 2.5: Máscaras usadas pelo algoritmo de Stentiford

As máscaras da figura 2.5 seguem a seguinte representação:

- Os \circ (círculos brancos) representam os pixels de valor 0;

- Os • (círculos pretos) representam os pixels de valor 1;
- Os × representam os pixels os quais não importam seu valor (*don't care*).

Cada máscara percorre a imagem em uma determinada ordem:

- **M1** esquerda para direita e cima para baixo (figura 2.5(a));
- **M2** esquerda para direita e baixo para cima (figura 2.5(b));
- **M3** direita para esquerda e baixo para cima (figura 2.5(c));
- **M4** direita para esquerda e cima para baixo (figura 2.5(d)).

2.4.1.2 Passos do algoritmo

O algoritmo de Stentiford é basicamente composto de seis passos, vistos a seguir:

1. Percorrer a imagem até encontrar um pixel que se encaixe na máscara $M1$;
2. Se este pixel não for um ponto final e se o seu número de conectividade = 1, marcar este ponto para que seja apagado mais tarde.
3. Repetir os passos 1 e 2 para todos os pixels que se encaixem na máscara $M1$.
4. Repetir os passo 1, 2 e 3 para cada uma das máscaras $M2$, $M3$ e $M4$, nesta ordem.
5. Se algum ponto estiver marcado para ser apagado, ele deve ser apagado mudando-o para a cor branca.
6. Se algum ponto foi apagado no passo 5, repetir todos os passos a partir do passo 1. Caso contrário, o processo termina.

2.4.2 Algoritmo de Zhang-Suen

Em 1984, Zhang e Suen publicaram um trabalho [ZS84] no qual propuseram um novo algoritmo paralelo de afinamento. Este trabalho trouxe resultados muito superiores quando comparados a outros da sua época. Mesmo com o decorrer dos anos, muitos pesquisadores, incluindo o próprio Zhang, propuseram novos testes e metodologias que melhoraram ainda mais o algoritmo de Zhang e Suen.

O afinamento pelo método de Zhang-Suen tem como base a comparação do pixel que está sendo testado em relação a sua 8-vizinhança. Há quatro regras básicas a serem

aplicadas na vizinhança, e o pixel só será excluído se satisfizer estas quatro regras. Estas regras têm como objetivo assegurar que o pixel, se eliminado, não fará com que diferentes regiões interligadas percam a conectividade, além de assegurar que a eliminação de pixels sempre ocorrerá nas bordas do objeto.

2.4.2.1 Definições teóricas

O algoritmo de Zhang-Suen consiste em sucessivas aplicações ao contorno da imagem de duas regras, sendo que os pontos do contorno são quaisquer pontos com valor 1 (preto, ou objeto) que tenham ao menos um dos seus oito vizinhos iguais a 0 (branco, ou fundo).

O algoritmo é dividido em duas sub-iterações. Em uma delas, o pixel $I(i, j)$ (pixel testado) é marcado para exclusão dependendo das seguintes condições:

- 1) O número de conectividade (explicado na seção 2.2.4) ser igual a 1;

Tomando a definição, que diz que conectividade é o número de transições de branco para preto em um determinado intervalo de pixels, os pixels que circundam o pixel central $P1$ na figura 2.6, ou seja, os pixels de vão de intervalo de $P2$ até $P9$, devem possuir apenas uma transição ou em outras palavras, o número de conectividade deve ser igual a 1.

- 2) Existir ao menos dois e no máximo seis pixels pretos, que equivalem ao objeto;

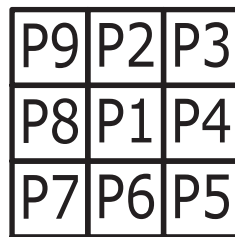


Figura 2.6: Método de Zhang-Suen: máscara utilizada

- 3) Ao menos um dos pixels $I(i, j + 1)$, $I(i - 1, j)$ ou $I(i, j - 1)$ devem ser brancos, como mostrado na figura 2.7;
- 4) Ao menos um dos pixels $I(i - 1, j)$, $I(i + 1, j)$ ou $I(i, j - 1)$ devem ser brancos, como mostrado na figura 2.8;

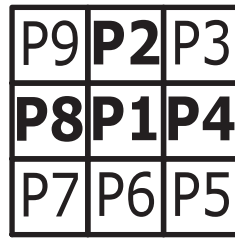


Figura 2.7: Método de Zhang-Suen: Ou $P2$, ou $P4$ ou $P8$ deve ser um pixel branco

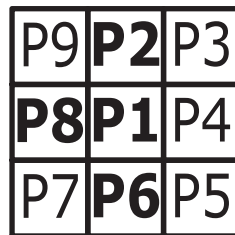


Figura 2.8: Método de Zhang-Suen: Ou $P2$, ou $P6$ ou $P8$ deve ser um pixel branco

Após estes quatro primeiros passos, encerra-se a primeira sub-iteração. Os pixels marcados para exclusão são eliminados. A próxima sub-iteração é a mesma, exceto para os passos 3 e 4:

- 1) O número de conectividade (seção 2.2.4) ser igual a 1;
- 2) Existir ao menos dois e no máximo seis pixels pretos, que equivalem ao objeto;
- 3) Ao menos um dos pixels $I(i-1, j)$, $I(i, j+1)$ ou $I(i+1, j)$ devem ser brancos, como mostrado na figura 2.9;

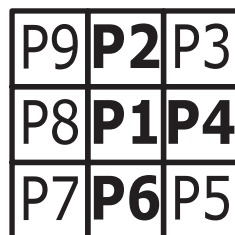


Figura 2.9: Método de Zhang-Suen: Ou $P2$, ou $P4$ ou $P6$ deve ser um pixel branco

- 4) Ao menos um dos pixels $I(i, j+1)$, $I(i+1, j)$ ou $I(i, j-1)$ devem ser brancos, como mostrado na figura 2.10;

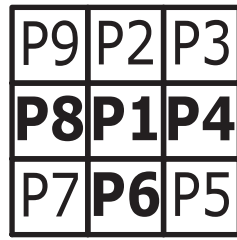


Figura 2.10: Método de Zhang-Suen: Ou $P4$, ou $P6$ ou $P8$ deve ser um pixel branco

Ao fim da iteração, os pixels marcados são excluídos. Em ambas as iterações, os pixels só devem ser eliminados no final das mesmas. Ao final da segunda iteração, senão houver mais nenhum pixel a ser eliminado, então o processo de Zhang-Suen está completo e a imagem encontra-se afinada.

2.4.3 Algoritmo de Zhang-Wang

Em 1996, Zhang e Wang [ZW96] aperfeiçoaram o método proposto anteriormente pelo próprio Zhang e Suen [ZS84].

2.4.3.1 Definições teóricas

Como visto na seção 2.2.1, uma imagem nada mais é do que um conjunto de pontos no plano. Uma imagem binária pode ser definida como sendo um conjunto P no primeiro plano (pixels pretos) e os pixels restantes (pixels brancos) pertencentes ao segundo plano, conforme a equação 2.1. A seguir são descritas as definições do algoritmo de Zhang-Wang, segundo [ZW96]:

- **Definição 1** - Janelas

Como visto na seção 2.2.3, $N(p)$ é o conjunto de todos os vizinhos de p , também denominado vizinhança de p , como mostrado na figura 2.11. Os vizinhos que têm como característica serem pares ($p[0]$, $p[2]$, $p[4]$ e $p[6]$ ou norte(p), leste(p), sul(p) e oeste(p)) são chamados de 4-vizinhos de p . Os vizinhos que têm como característica serem ímpares ($p[1]$, $p[3]$, $p[5]$ e $p[7]$ ou nordeste(p), sudeste(p), sudoeste(p) e noroeste(p)), junto com os 4-vizinhos de p , são chamados de 8-vizinhos de p .

Ao contrário do algoritmo de Zhang-Suen (seção 2.4.2), o algoritmo de Zhang-Wang utiliza duas janelas (figura 2.11) para realizar o processamento de um pixel. Cada uma das janelas é usada em uma das sub-iterações do algoritmo.

- **Definição 2** - Peso do pixel

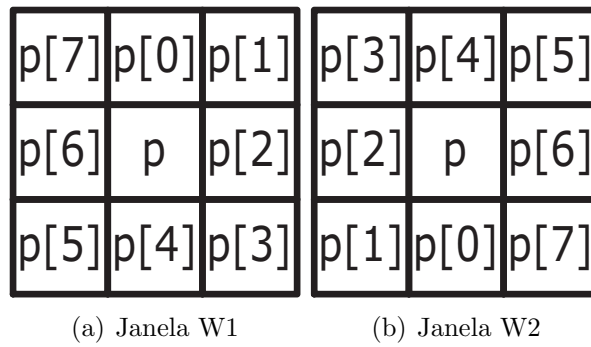


Figura 2.11: Janelas do método de Zhang-Wang

O valor do peso (*weight number*) de um pixel p , $WN(p)$, é definido como:

$$WN(p) = \sum_{k=0}^7 p[k] * 2^k \quad (2.3)$$

Considerando uma janela 3×3 , o valor de $WN(p)$ está sempre entre 0 e 255. O conjunto de todos os pesos é chamado de WS :

$$WS = \{0, 1, \dots, 255\}$$

- **Definição 3** - Número de vizinhos de p

O número de vizinhos (*neighbor number*) de p , $NN(p)$, é o número de vizinhos diferentes de zero que p possui:

$$NN(p) = \sum_{k=0}^7 p[k] \quad (2.4)$$

- **Definição 4** - Pontos finais

Um ponto é denominado ponto final (*end point*) se $NN(p) = 1$. Estes pontos constituem um conjunto denominado ES :

$$ES = \{1, 2, 4, 8, 16, 32, 64, 128\}$$

- **Definição 5** - Número de conectividade

O número de conectividade (*connection number*) de p , $CN(p)$, é definido, como já visto na seção 2.2.4, como:

$$CN(p) = \sum_{k=0,2,4,6} \overline{p[k]} * (p[k+1] \cup p[k+2]) \quad (2.5)$$

- **Definição 6** - Globalmente removível

Um ponto, p em P , é denominado globalmente removível (*globally removable*) se $CN(p) = 1$ e $NN(p) > 1$. Todos os pontos globalmente removíveis constituem um conjunto denominado GRS :

$$GRS = \{3, 5, 6, 7, 12, \dots, 248, 249, 251, 252, 254\} - 108$$

- **Definição 7** - Irredutível

Um ponto, p em P , é denominado irredutível (*irreducible*) se p não é um ponto globalmente removível. Todos os pontos irredutíveis formam um conjunto denominado IPS :

$$IPS = WS - GRS$$

- **Definição 8** - Localmente removível

O conjunto localmente removível (*locally removable set*), LRS , é um subconjunto de GRS e é definido por sub-iteração. O elemento em LRS é chamado de ponto removível e os vizinhos de p que estão no LRS são chamados de vizinhos removíveis de p . Para o algoritmo, define-se conjuntos localmente removíveis. Por exemplo, em 4-sub-iterações, define-se 4 conjuntos localmente removíveis:

$$LRS4[1] = \{p \mid p \text{ em } GRS \text{ e } norte(p) = 0\}$$

$$LRS4[2] = \{p \mid p \text{ em } GRS \text{ e } sul(p) = 0\}$$

$$LRS4[3] = \{p \mid p \text{ em } GRS \text{ e } leste(p) = 0\}$$

$$LRS4[4] = \{p \mid p \text{ em } GRS \text{ e } oeste(p) = 0\}$$

- **Definição 9**

O esqueleto SS é um conjunto de pontos pretos que não podem ser apagados pelo algoritmo de afinamento.

- **Definição 10**

Um esqueleto SS tem um pixel de largura se todos os pontos em SS encontram-se em IPS .

- **Definição 11**

Afinar uma imagem binária significa aplicar um procedimento iterativo para remoção dos pontos localmente removíveis.

- **Definição 12**

Um algoritmo de afinamento é perfeito se ele pode gerar esqueletos de largura de um pixel.

2.4.3.2 Funcionamento

No algoritmo de Zhang-Wang, a exclusão dos pixels é baseada apenas na iteração anterior. Isso o torna adequado para ser implementado de forma paralela, já que os pixels que satisfazem as condições de exclusão podem ser removidos simultaneamente. Entretanto, neste caso podem existir dificuldades para manter a conectividade. Para contornar este problema, cada iteração é dividida em duas sub-iterações, onde somente um subconjunto de pixels de contorno são analisados para possível remoção. Estas sub-iterações utilizam-se das janelas $W1$ e $W2$ definidas na seção 2.4.3.1.

A primeira sub-iteração varre a imagem com a janela $W1$ removendo os pixels de contorno acima e a direita do ponto analisado. A segunda sub-iteração, utilizando a janela $W2$, remove os pixels de contorno abaixo e a esquerda do ponto analisado. No fim de cada iteração, a imagem restante é atualizada para a próxima iteração.

2.4.4 Algoritmo de Holt

Em 1997, Holt [HSCP97] sugeriu um algoritmo rápido em que as iterações utilizam expressões lógicas em um ponto p e sua vizinhança 3×3 para verificar se o mesmo pode ser removido ou não.

2.4.4.1 Definições teóricas

A expressão lógica usada na primeira iteração pode ser descrita como:

$$v(C) \wedge (\sim edge(C) \vee (v(L) \vee v(S) \vee (v(N) \vee v(O)))) \quad (2.6)$$

A segunda iteração usa a seguinte expressão lógica:

$$v(C) \wedge (\sim edge(C) \vee (v(O) \vee v(N) \vee (v(S) \vee v(L)))) \quad (2.7)$$

Um ponto só pode ser removido se o resultado das expressões lógicas for verdadeiro. Para tal, é necessário conhecer a nomenclatura usada nas expressões:

- **Função $v()$:** representa o valor do ponto. O valor retornado será *verdadeiro* se o ponto for igual ao ponto de um objeto (ponto preto) e *falso* se ponto for de fundo (ponto branco).
- **Função $edge()$:** o valor retornado será *verdadeiro* se o ponto analisado estiver na borda do objeto e *falso* caso contrário.
- **Letras N, S, L e O :** representam os vizinhos do ponto analisado, como mostrado na figura 2.12.

NO	N	NE
O	C	L
SO	S	SE

Figura 2.12: Método de Holt: janela utilizada para vizinhança

Para que a função $edge()$ retorne *verdadeiro*, é necessário que ela atenda as seguintes condições:

- Número de conectividade igual a 1 (ver seção 2.2.4);
- Possuir de dois a seis pontos pretos (objeto) nas oito direções possíveis a partir do ponto analisado, como pode ser visto na figura 2.13.

Com base nestas definições e regras, as expressões 2.6 e 2.7 podem ser mescladas, formando a seguinte expressão:

$$v(C) \wedge (\sim edge(C) \vee (edge(L) \wedge v(N) \wedge v(S)) \vee (edge(S) \wedge v(O) \wedge v(L)) \vee (edge(L) \wedge edge(SE) \wedge edge(S))) \quad (2.8)$$

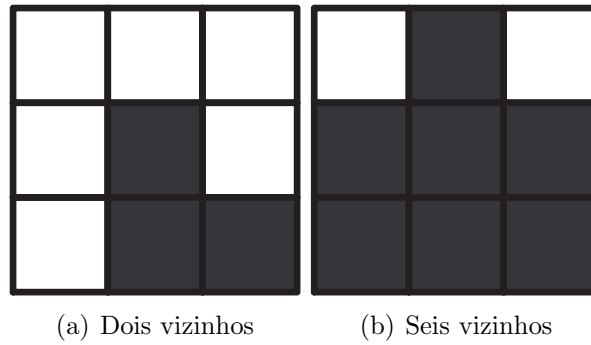


Figura 2.13: Método de Holt: exemplo do número de vizinhos

2.4.4.2 Staircase removal

O método de Holt, representado pela expressão 2.8, pode ser aperfeiçoado agregando a ele o método de remoção de degraus (*staircase removal*).

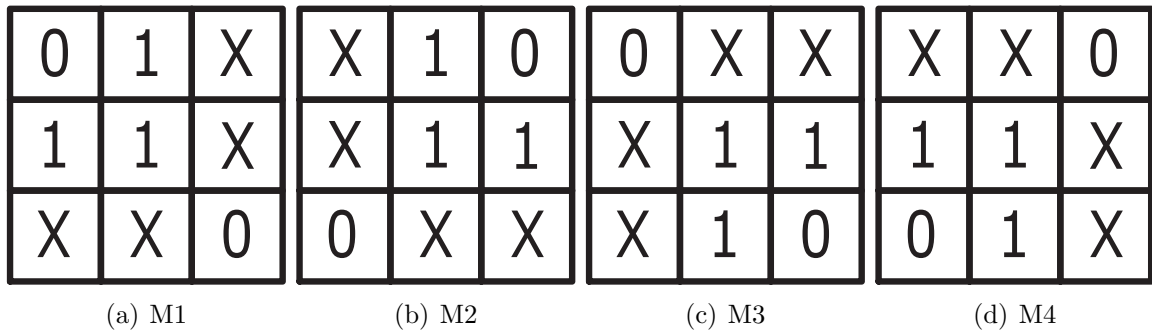


Figura 2.14: Máscaras do método de Holt com staircase removal

O processo de *staircase removal* consiste em remover os pixels que apresentam forma semelhante a uma escada e que não afetam o formato e a conectividade do objeto. Para esta operação são utilizadas quatro máscaras (figura 2.14), que permitem que o pixel analisado seja removido caso alguns dos valores de X representem o fundo da imagem (branco).

As expressões lógicas que descrevem as máscaras mostradas na figura 2.14 são:

$$\begin{aligned}
 v(C) \wedge \sim (v(N) \wedge ((v(L) \wedge \sim v(NE) \wedge \sim v(SO) \wedge (\sim v(O) \wedge \sim v(S)) \vee \\
 (v(O) \wedge \sim v(NO) \wedge \sim v(SE) \wedge (\sim v(L) \wedge \sim v(S))))) \quad (2.9)
 \end{aligned}$$

$$v(C) \wedge \sim (v(S) \wedge ((v(L) \wedge \sim v(NE) \wedge \sim v(SO) \wedge (\sim v(O) \wedge \sim v(N)) \vee (v(O) \wedge \sim v(NO) \wedge \sim v(SE) \wedge (\sim v(L) \wedge \sim v(N))))))) \quad (2.10)$$

2.4.4.3 Comparação de métodos

As imagens geradas pelo método de Holt com *staircase removal* apresentam melhores resultados, como pode ser visto na figura 2.15.

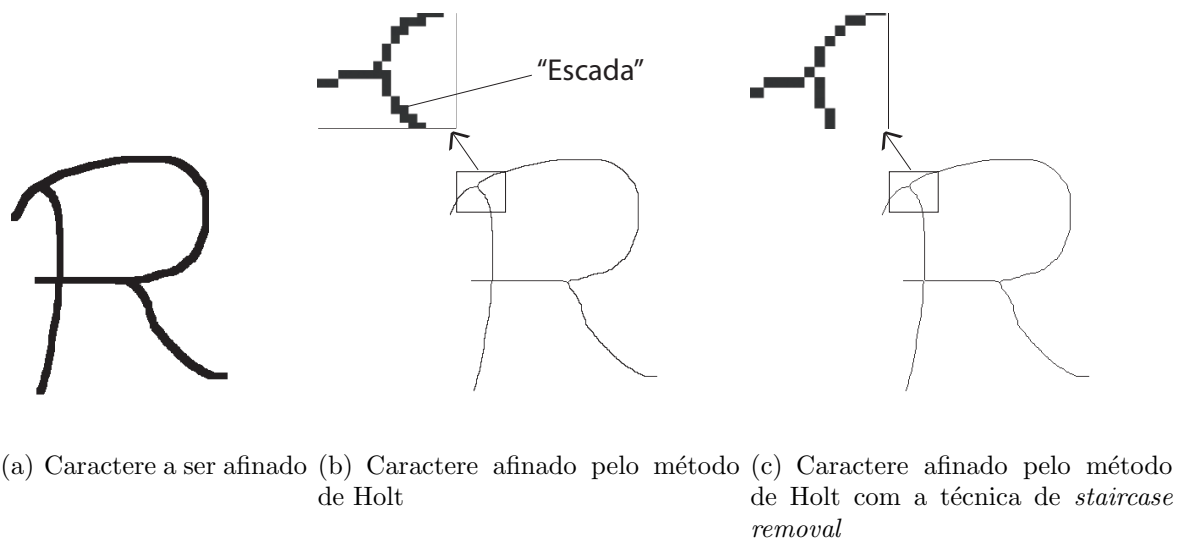


Figura 2.15: Método de Holt: exemplo

Os destaques apresentados nas figuras 2.15(b) e 2.15(c) mostram que o *staircase removal* realiza com sucesso a remoção de pontos redundantes formadores de degraus. Com isso, o método de Holt com *staircase removal* constitui uma poderosa ferramenta para afinamento, sendo mais rápido e simples de implementar que métodos como o de Zhang-Suen [ZS84].

2.4.5 Algoritmo de Petrosino e Salvi

No ano 2000, Alfredo Petrosino e Giuseppe Salvi [PS00] propuseram um novo algoritmo de afinamento paralelo de duas sub-iterações que também utiliza o conceito de janelas para processar um pixel. A diferença entre este algoritmo e os outros que usam o mesmo sistema, como Zhang-Suen [ZS84] e Zhang-Wang [ZW96], é o formato *assimétrico* 3×4 da janela utilizada na primeira iteração.

2.4.5.1 Definições teóricas

O algoritmo de Petrosino e Salvi elimina os pixels a partir da análise dos pixels vizinhos, utilizando uma máscara 3×4 na primeira etapa e uma máscara 3×3 na segunda etapa.

Dado um pixel em estudo p , p é um pixel preto e os seus vizinhos na janela 3×3 são rotulados como x_1, x_2, \dots, x_8 conforme a figura 2.16.

x4	x3	x2	y4
x5	p	x1	y5
x6	x7	x8	y6
y1	y2	y3	

Figura 2.16: Método de Petrosino e Salvi: máscara utilizada

- **Definição 1:** Os pixels (x_1, x_2, \dots, x_8) são os 8-vizinhos do pixel p , denotado por $N(p)$ e são ditos 8-adjacentes vizinhos a p ;
- **Definição 2:** Os pixels $(x_1, x_3, x_5$ e $x_7)$ são os 4-vizinhos do pixel p , e são ditos 4-adjacentes vizinhos a p ;
- **Definição 3:** Uma seqüência de pontos Z_1, Z_2, \dots, Z_n é chamada de 8- ou 4-caminhos em 8- ou 4-vizinhos de Z_i , para $i = 1, 2, \dots, n - 1$;
- **Definição 4:** Um subconjunto C de uma imagem P é denominado 8- ou 4-conectado se para todo par de pontos (x, y) em C existir 8- ou 4-caminhos de x para y , e que sejam pontos pertencentes a C ;
- **Definição 5:** Definição 5: O número de transições entre pixels pretos e brancos, e vice-versa, é denominado número de *Rutowitz*, é definido na equação 2.11.

$$X_R(p) = \sum_{i=1}^8 |x_{i+1} - x_i| \quad \text{onde } x_9 = x_1 \quad (2.11)$$

Um algoritmo paralelo de duas sub-iterações necessita de dois conjuntos de pontos removíveis em paralelo, cuja união contém todo os pixels de bordas removíveis. Isto

significa que todos os pontos p do conjunto podem ser simultaneamente apagados sem causar erosão excessiva e ainda manter a topologia de qualquer padrão binário.

- **Definição 6:** Se um pixel preto p possui pelo menos um pixel branco como vizinho em sua 4-vizinhança, é chamado ponto de borda (*edge point*);
- **Definição 7:** Um pixel preto que tem no máximo um vizinho como pixel preto em sua 8-vizinhança é chamado de ponto final (*end point*);
- **Definição 8:** A quebra de continuidade na imagem padrão com a exclusão de pixel p torna este ponto um ponto de quebra (*break point*).

O algoritmo de Petrosino e Salvi executa as iterações até que haja um iteração onde nenhum pixel seja removido. Em cada uma das iterações, o pixel analisado é marcado para remoção se ele for um ponto de borda e não for um ponto final, sendo que remoção não deve causar a erosão excessiva. Para que os critérios acima possam ser seguidos, algumas regras devem ser estipuladas. Eles tratam da possibilidade de remoção de dois conjuntos de pixels, I_1 para a primeira iteração e I_2 para a segunda.

Lema 1 *O conjunto I_1 de pixels pretos p devem satisfazer as seguintes condições:*

$$X_r(p) = 2 \tag{2.12}$$

$$2 \leq b(p) \leq 6 \tag{2.13}$$

$$R_0 = 0 \quad \text{onde} \quad R_0 = x_1 x_7 x_8 ((\overline{y_5} x_2 x_3 \overline{x_5}) \vee (\overline{y_2} x_3 x_5 x_6)) \tag{2.14}$$

A equação 2.13 mostra que existem de dois a seis pixels vizinhos de p que são candidatos a serem removidos. A equação 2.12 mostra que existem apenas duas transições entre pixels brancos e pretos (1-0 ou 0-1) na mesma vizinhança, e isso corresponde a ter dois pixels pretos (1) entre x_1, x_2, \dots, x_8 , enquanto os seis pixels restantes podem ser todos brancos (0) ou quatro deles serem iguais a 1 (preto). O conjunto $N(p)$ contém entre dois e seis pixels que estão 4-conectados entre eles possuem possíveis pares de pixels que são conectados por um 4-caminho. A expressão 2.13 garante que p tem, pelo menos, um pixel branco em sua 4-vizinhança e que p não é um ponto final.

Se um pixel p satisfazer as condições das equações 2.12 e 2.13, eles podem ser removidos em paralelo. Porém, objetos que possuem largura de dois pixels desaparecem por completo. Este problema não pode ser solucionado usando a janela de 3×3 . Para

solucionar este problema, faz-se necessário o uso de uma janela 3×4 e 4×3 , mostrados nas figuras 2.17(a) e 2.17(b). A expressão booleana R_0 , representada em 2.14, recupera as linhas horizontais e verticais que possuam a largura de dois pixels. Em cada padrão que R_0 é verdadeiro, o pixel analisado é salvo ou, em outras palavras, todos os pixels p que satisfaçam a condição de remoção da equação 2.14 podem ser removidos em paralelo.

Lema 2 *O conjunto I_2 de pixels pretos p devem satisfazer as seguintes condições:*

$$S_0 = x_3x_7 \vee x_5x_1$$

$$S_1 = x_1\overline{x_6}(\overline{x_4} \vee x_3) \vee x_3\overline{x_8}(\overline{x_6} \vee x_5) \vee x_7\overline{x_4}(\overline{x_2} \vee x_1) \vee x_5\overline{x_2}(\overline{x_8} \vee x_7)$$

$$R_1 = \overline{x_3}(x_1\overline{x_8} \vee x_5\overline{x_6})x_7(\overline{x_5x_8} \vee \overline{x_1x_6})$$

$$\overline{S_0} \wedge S_1 \wedge (b(p) \geq 2) \quad (2.15)$$

$$(R_1 = 0) \wedge (b(p) \geq 3) \quad (2.16)$$

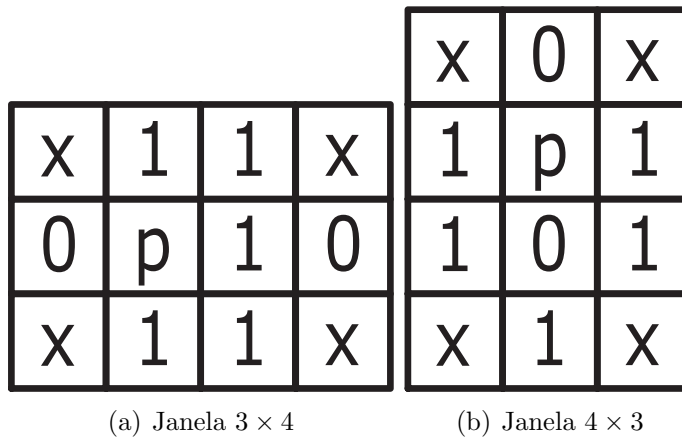


Figura 2.17: Máscaras utilizadas no método de Petrosino e Salvi

Se a condição imposta pela equação 2.15 for satisfeita, os pixels podem ser removidos em paralelo. Caso contrário, o pixel não poderá ser removido até a aplicação de outra expressão booleana (expressão 2.16). A expressão 2.16, associada com a expressão 2.15, permite que o pixel seja removido em paralelo sem ocasionar nenhum problema.

Muitos algoritmos existentes produzem esqueletos com muito ruído se as bordas das imagens originais possuírem muito ruído também. Isto tem sido causa de inúmeros estudos. Alguns destes estudos sugerem que sejam aplicados métodos de suavização de bordas antes do método de afinamento. Outros eliminam os pontos ruidosos a cada iteração utilizando uma máscara 3×3 .

Em alguns casos, o uso da máscara 3×3 não retorna bons resultados, sendo necessário aplicar uma máscara 5×5 (figura 2.18) em alguns pixels vizinhos ao ponto analisado.

x	y	x	x	x
x	1	0	0	x
x	0	p	0	x
x	0	0	0	x
x	x	x	x	x

Figura 2.18: Método de Petrosino: máscara 5×5

A máscara da figura 2.18 e suas reflexões horizontais e verticais são utilizadas para limpeza de pixels através da operação booleana *OR*.

2.4.5.2 Comparação de métodos

Segundo [PS00], os resultados obtidos foram muito bons, ou seja, o algoritmo de Petrosino e Salvi funciona satisfatoriamente. Por exemplo, com relação ao algoritmo de Zhang-Suen, o algoritmo de Petrosino e Salvi obteve um esqueleto melhor definido em termos do número de pixels (menor quantidade), um desempenho muito bom e resultados visíveis interessantes, como pode ser visto na figura 2.19.

2.4.6 Algoritmo utilizando morfologia matemática

O método da morfologia matemática segue a aplicação do operador morfológico de afinamento, que implica primeiro em passar uma série de “elementos estruturantes” pela imagem, transformando o pixel analisado em preto, se o padrão do elemento foi

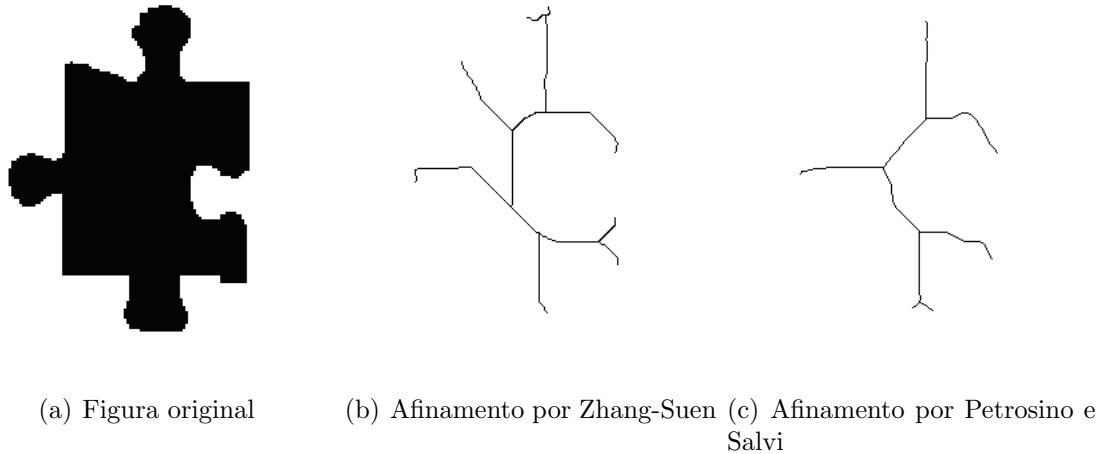


Figura 2.19: Comparação entre métodos de afinamento

encontrado, ou em branco, caso não o seja. Este processo também é conhecido como *Transformada Hit or Miss*. Após este processo, é aplicado o operador diferença entre a imagem original e a imagem transformada.

2.4.6.1 Definições Básicas

A equação 2.17 mostra o raciocínio utilizado no algoritmo da morfologia.

$$A \odot (B1, B2) = A - (A \otimes (B1, B2)) = A \cap (A \otimes B)^c \quad (2.17)$$

Esta equação pode ser dividida nos seguintes passos:

1. Oito elementos estruturantes (figuras 2.20 e 2.21) percorrem a imagem;

0	0	0	X	0	0	1	X	0	X	1	X
X	1	X	1	1	0	1	1	0	1	1	0
1	1	1	X	1	X	1	X	0	X	0	0
(a) M1	(b) M2	(c) M3	(d) M4								

Figura 2.20: Máscaras do método da Morfologia Matemática

2. Cada máscara que percorre a imagem faz com que o pixel analisado torne-se 1 (preto) caso haja uma coincidência. Caso contrário, torna-se 0 (branco). Lembrando que o

X das máscaras indicam que não importa o valor daquele pixel;

1	1	1	X	1	X	0	X	1	0	0	X
X	1	X	0	1	1	0	1	1	0	1	1
0	0	0	0	0	X	0	X	1	X	1	X
(a) M5	(b) M6	(c) M7	(d) M8								

Figura 2.21: Máscaras do método da Morfologia Matemática

3. Realiza-se a operação lógica *AND* entre o pixel original e o pixel transformado;
4. Os dois primeiros passos se repetem para os pixels restantes da imagem;
5. A segunda máscara será aplicada sobre o resultado da aplicação da primeira;
6. A terceira máscara será aplicada sobre o resultado da aplicação da segunda e assim sucessivamente até a oitava máscara;
7. Os pixels alterados são modificados na imagem original;
8. Se houveram alterações, recomeça o algoritmo. Caso contrário, foi alcançado o esqueleto da imagem.

2.5 Estudo comparativo

Existe uma série de problemas relacionados aos algoritmos de afinamento mostrados nas seções anteriores. A tabela 2.1 mostra um comparativo entre os métodos, mostrando as vantagens e os defeitos que cada um pode ter.

O método de Stentiford funciona somente para imagens contendo objetos de geometria simples. Dependendo da imagem, o objeto resultante pode conter pixels desconexos (figura 2.22(d)), sendo uma característica ruim para um algoritmo de afinamento. A causa provável desta descontinuidade é alguma falha no processo que verifica o número de conectividade de um pixel ou na forma com que as máscaras são aplicadas na imagem.

Como pode ser observado na figura 2.22, existem outros problemas que a aplicação do algoritmo de Stentiford pode causar. Na figura 2.22(a), ocorre uma intersecção entre duas linhas, que ao ser afinada produz um segmento relativamente longo. Isto é chamado

Tabela 2.1: Estudo comparativo entre os métodos apresentados

	Gera pixels desconexos	Necking	Tailing	Rebarbas	Não mantém a forma	Não gera apenas 1px de largura
Stentiford	X	X	X	X		
Zhang e Suen				X	X	
Zhang e Wang				X	X	
Holt		X	X			X
Petrosino e Salvi		X	X			
Morfologia matemática				X		

de “*necking*”. A figura 2.22(b) mostra outro problema, ocasionado quando temos duas linhas que possuem um ângulo relativamente pequeno se unem, fazendo com que apareça um segmento reto que não corresponde à nenhuma parte da imagem. Este problema é chamado de “*tailing*”.

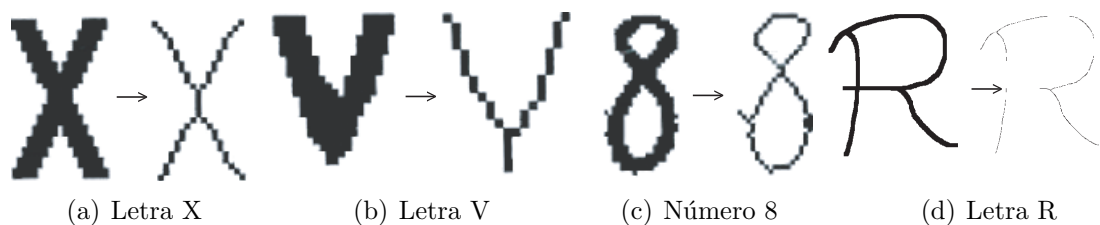


Figura 2.22: Problemas ocorridos com o algoritmo de Stentiford

Outro problema que pode ocorrer, mostrado na figura 2.22(c), é o “*line fuzz*”. Qualquer pixel adicional que esteja conectado à borda do objeto sendo afinado, pode criar um elemento que não deveria pertencer ao esqueleto. Este segmento também é chamado de “*pés-de-galinha*”, ou “*rebarba*”. Neste caso, não só o algoritmo de Stentiford que apresenta este problema. Zhang-Suen, Zhang-Wang e o Morfológico também podem resultar em objetos afinados contendo rebarbas.

O método de Zhang-Suen não mantém a morfologia dos objetos. Em outras palavras, a forma do objeto não é reproduzida fielmente na imagem afinada, como pode ser visto na figura 1.6. A grande diferença entre o algoritmo de Zhang-Suen e Zhang-Wang

é que este último é capaz de manter a geometria dos objetos. Mesmo assim, o método de Zhang-Wang pode, ocasionalmente, dependendo da topologia dos objetos, alterá-los.

Por fim, como dito anteriormente, o algoritmo da Morfologia Matemática tem uma série de problemas em relação ao aparecimento de rebarbas no objeto afinado, como pode ser visto na figura 2.23.

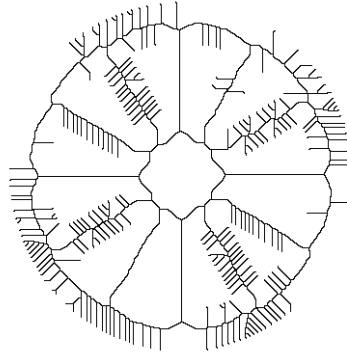


Figura 2.23: Imagem afinada através da morfologia matemática

2.6 Comentários finais

Nenhum dos algoritmos descritos neste capítulo é perfeito. A disposição do objeto influencia no resultado final do afinamento, assim como os ruídos existentes na imagem. A figura 2.24 ilustra este problema.

A figura 2.24 mostra um selo de envelope postal afinado através de diversos métodos. O grande problema é na maioria das vezes é impossível adquirir uma imagem de uma forma perfeita. Normalmente, ela é digitalizada com deformações e ruídos, que faz com que, submetida aos processos de afinamento, apresente resultados diferentes e insatisfatórios.

O grande desafio desta pesquisa é, através do método a ser proposto de afinamento baseado em exemplos, fazer com que o objeto, mesmo com deformações e ruídos, possa apresentar um resultado satisfatório, que seja capaz de ser melhor interpretado do que aqueles conseguidos com os algoritmos hoje existentes. Em outras palavras, o desafio é tentar reproduzir aquilo que o usuário quer, visto que ele é quem passa a imagem ideal a ser treinada para o algoritmo.

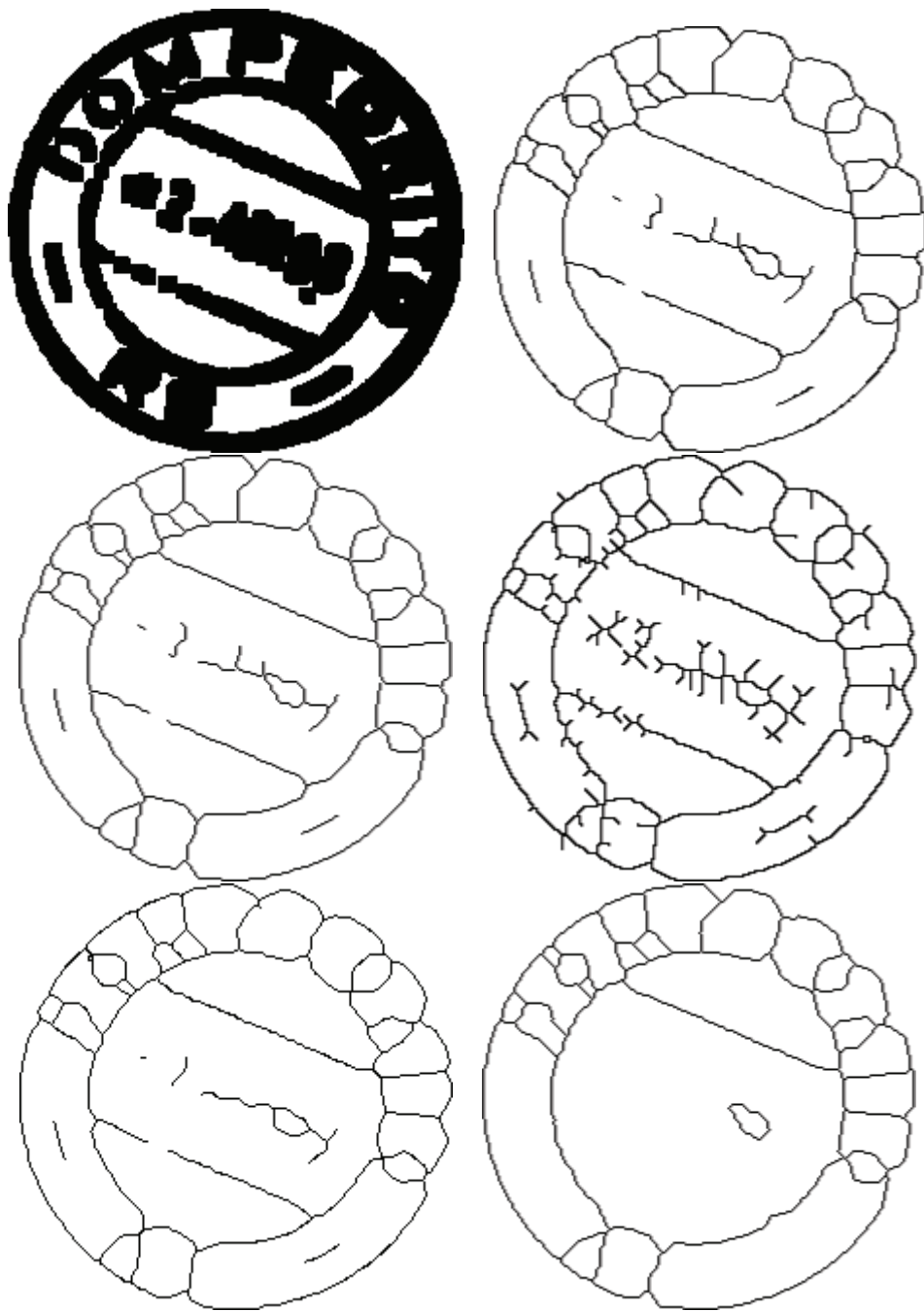


Figura 2.24: Exemplos de um selo de envelope postal afinado através das várias abordagens

Capítulo 3

Fundamentação teórica

Neste capítulo serão apresentados alguns métodos que usam aprendizagem como parte de sua solução, além de informações sobre a função distância e o classificador utilizado.

3.1 Função distância

A função distância é uma operação que calcula, em uma imagem binária, as distâncias dos pixels relevantes, ou seja, os objetos em relação ao fundo da imagem. Isto resulta uma imagem de distância cujos os valores dos pixels corresponderão à distância destes pixels em relação ao pixel do fundo mais próximo.

A função distância é um instrumento importante em processamento de imagens pois pode ser usada em muitas outras transformações, como: dilatação, erosão, caminho mínimo entre dois pixels, entre outras.

A maneira mais conhecida de calcular a distância é através da *Distância Euclidiana*, por ser invariante à rotação. Entretanto, por serem algoritmos computacionalmente custosos, a construção de algoritmos eficientes para a Distância Euclidiana é uma tarefa difícil. Para contornar este problema, foram desenvolvidos outros algoritmos, que consideram pequenas vizinhanças de um ponto ao invés de toda a imagem, fornecendo uma distância aproximada. Alguns destes métodos são mostrados na figura 3.2, destacando-se entre eles *Chamfer* [BOR86], *City-block*, *Chessboard* e a *Morfologia Matemática*.

3.1.1 Definição teórica

A partir de uma imagem binária, uma função f mapeia seus pixels no intervalo discreto entre 0 (fundo) e 1 (objeto). A função distância atribui aos pixels de um objeto

na imagem analisada a distância mínima entre estes pixels e o fundo da imagem, como mostrado na figura 3.1.

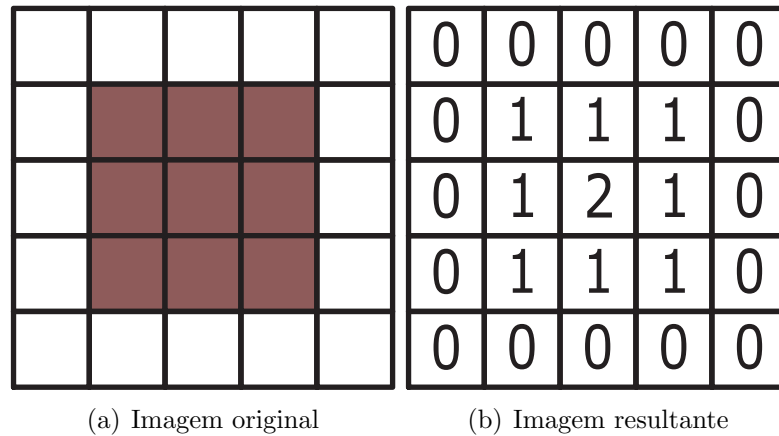


Figura 3.1: Exemplo da imagem resultante de uma função distância

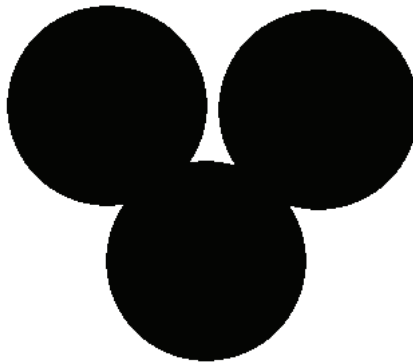


Figura 3.2: Imagem exemplo submetida às técnicas de cálculo da função distância

3.1.1.1 Funcionamento

O cálculo da função distância consiste em passar uma máscara na imagem binária a ser processada. O formato e os valores desta máscara dependem da abordagem de cálculo escolhida, como por exemplo City-block, Chamfer ou Chessboard. A figura 3.1 mostra o resultado do cálculo da função distância.

Pode-se perceber que, quanto mais interno for o pixel do objeto (pixel preto), maior o valor da distância. O cálculo é realizado em dois passos chamados *forward* e *backward*.

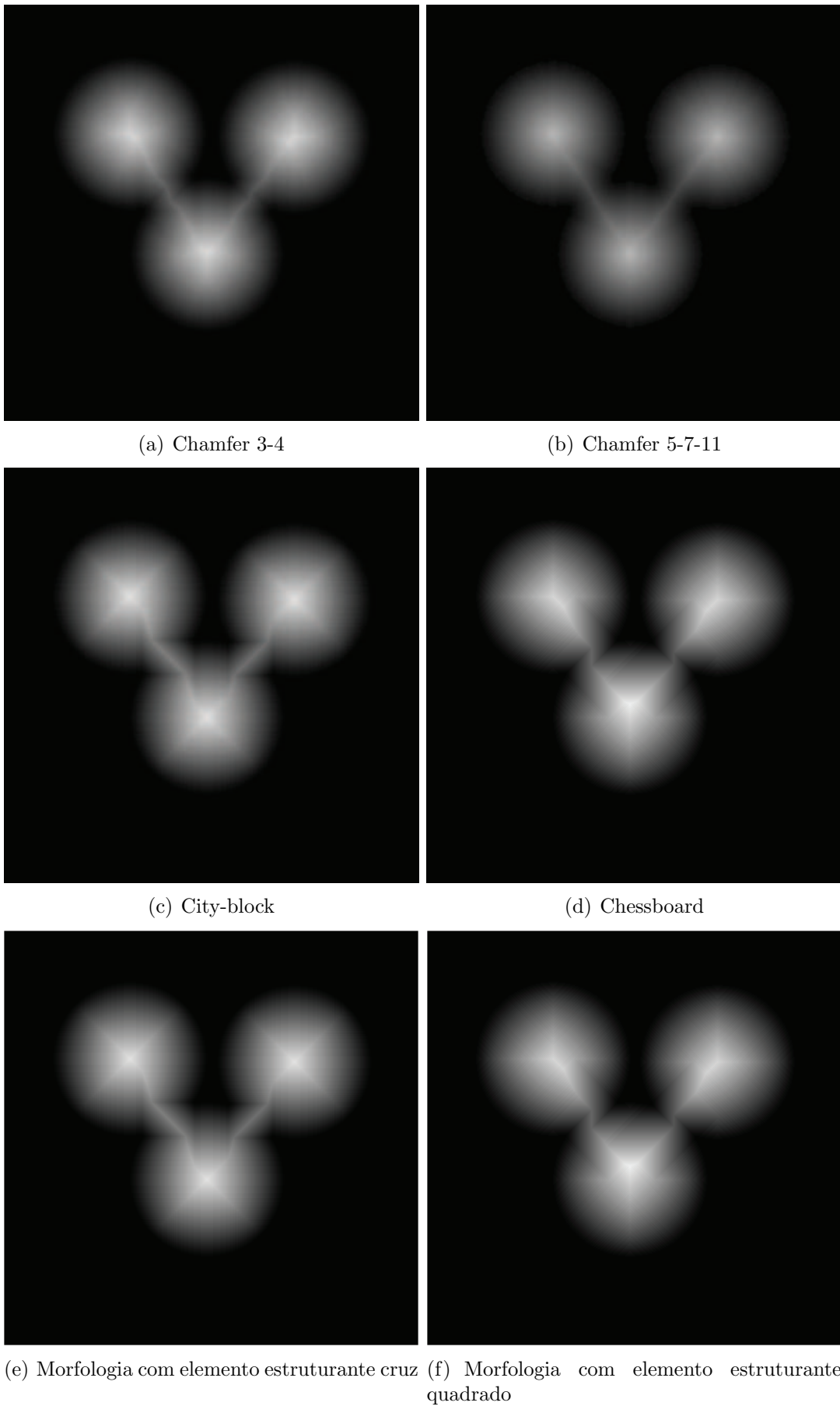


Figura 3.3: Resultado do cálculo da função distância na imagem 3.2

No modo *forward*, a imagem é percorrida da esquerda para direita e de cima para baixo. Já o modo *backward* é o oposto, partindo da direita para a esquerda e de baixo para cima.

As abordagens Chamfer 3-4, Chessboard e City-block utilizam, no modo *forward*, uma máscara 3×3 , como mostrado na figura 3.4.

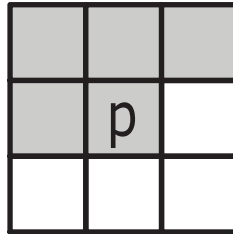


Figura 3.4: Máscara do modo forward das abordagens Chamfer 3-4, Chessboard e City-block

Já o modo *backward* utiliza uma máscara como a apresentada na figura 3.5.

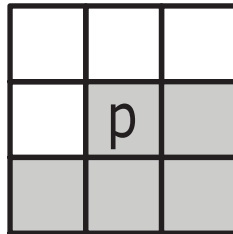


Figura 3.5: Máscara do modo backward das abordagens Chamfer 3-4, Chessboard e City-block

A abordagem Chamfer 5-7-11 utiliza máscaras diferentes das apresentadas nas figuras 3.4 e 3.5. O modo *forward* utiliza a máscara apresentada na figura 3.6, enquanto o modo *backward* utiliza a máscara apresentada na figura 3.7.

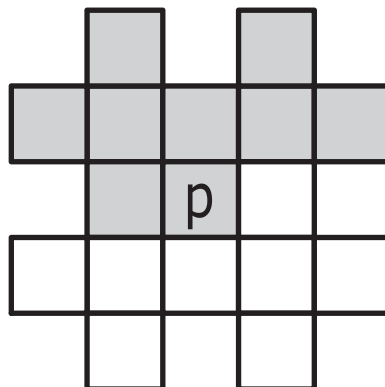


Figura 3.6: Máscara do modo forward da abordagem Chamfer 5-7-11

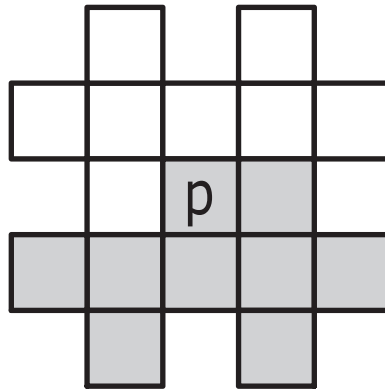


Figura 3.7: Máscara do modo backward da abordagem Chamfer 5-7-11

3.1.1.2 Algoritmo base

A função distância é aplicada em imagens binárias, onde a matriz de pixels contém apenas pixels pretos e brancos, como na figura 3.8. O processo inicia gerando uma nova imagem, em níveis de cinza, onde um valor infinito é atribuído aos pixels pretos e onde os pixels brancos recebem o valor zero. Na prática, este valor tido como infinito é qualquer valor suficientemente maior que qualquer distância entre os pixels internos da figura. Um exemplo é este valor ser inicializado como sendo a altura multiplicada pela largura da imagem.

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Figura 3.8: Exemplo de imagem binária

As máscaras correspondentes a cada algoritmo varrem a imagem original, como mostrado na figura 3.9. Esta varredura é comandada por laços, onde os valores de a e b são somados aos valores da imagem dependendo da máscara utilizada.

Cada pixel é analisado através da equação 3.1 a seguir:

$$v_{i,j} = \min(v_{i+kj+l} + c(k, l)) \quad (3.1)$$

b	a	b
a	p	a
b	a	b

Figura 3.9: Exemplo de máscara utilizada para a abordagem Chamfer 3-4, City-Block e Chessboard

O elemento $v(i, j)$ é o pixel da posição (i, j) na imagem em nível de cinza criada. Este pixel armazena o valor mínimo de todas as somas realizadas por $v_{i+kj+l} + c(k, l)$. Esta equação representa a soma entre os valores do pixel da imagem binária e de uma posição da máscara que esteja exatamente sobre este pixel durante um determinado momento da varredura. Atribui-se a $v_{i,j}$ o menor valor da soma, tanto entre o modo *forward* quanto o modo *backward*.

3.1.1.3 Abordagens

Dentre as abordagens existentes para o cálculo da função distância, as quatro descritas por Borgefors [BOR86] são as estudadas, como já mostrado nas máscaras da seção anterior.

A abordagem City-block é a mais simples e rápida de todas as transformações. Utiliza a máscara mostrada na figura 3.9 com os valores $a = 1$ e $b = \infty$ (normalmente altura \times largura). O valor ∞ significa que as distâncias diagonais são ignoradas pelo método. Parecida com a abordagem City-block, o algoritmo Chessboard utiliza $b = 1$, fazendo com que as distâncias diagonais não sejam ignoradas pelo método.

A máscara utilizada na abordagem Chamfer 3-4 é a mesma que os dois exemplos anteriores. Porém, os valores de a e b são, respectivamente 3 e 4, como na figura 3.10. Por fim, a abordagem Chamfer 5-7-11 é a única que não apresenta uma máscara 3×3 . Sua máscara, com valores 5, 7 e 11 usa uma janela 5×5 , como mostrado na figura 3.11.

3.1.1.4 Função distância por morfologia matemática

Com a Morfologia Matemática também é possível calcular a função distância em uma imagem binária. Para tanto, é necessário aplicar a equação

$$\forall P(x, y) \in S, D(S)(P(x, y)) = \min n \in N : P(x, y) \notin \text{ero}^{nB}(X), \quad (3.2)$$

4	3	4
3	p	3
4	3	4

Figura 3.10: Máscara da abordagem Chamfer 3-4

	11		11	
11	7	5	7	11
	5	0	5	
11	7	5	7	11
	11		11	

Figura 3.11: Máscara da abordagem Chamfer 5-7-11

onde para cada pixel $P(x, y)$ pertencente a S , $D(S(P(x, y)))$ representa a distância entre $P(x, y)$ e o fundo da imagem. De certa forma, o funcionamento do algoritmo é parecido com as abordagens mostradas na seção anterior, mudando apenas na forma em que a nova imagem é gerada.

Primeiramente, uma nova imagem em níveis de cinza é criada, onde os pixels pretos da imagem são marcados como ∞ (como no exemplo da seção anterior, qualquer valor maior que a maior distância possível) e os pixels brancos (fundo) são marcados como 0. A nova imagem também é varrida pelos métodos *forward* e *backward*, entretanto, ao invés de uma máscara como nas abordagens anteriores, a distância é determinada através de uma erosão em níveis de cinza, como mostrado na equação 3.3, definida em [FAC96].

$$ero^g(f(x)) = \min[f(y) - g(x - y)] : y \in E \quad (3.3)$$

A erosão pode ser realizada com qualquer elemento estruturante. Este será dividido em duas “metades”, para acelerar o processo. A primeira “metade” $E1$ do elemento estruturante vai de seu primeiro ponto superior esquerdo até o centro da imagem. A segunda “metade” $E2$ começa imediatamente depois do centro do elemento estruturante até o último ponto inferior direito. Deste modo, a primeira parte do rastreamento é realizar a erosão na imagem com o $E1$ e depois com o $E2$.

3.2 Aprendizagem baseada em exemplos

Apesar da imensa gama de informações existentes relacionadas a métodos de aprendizagem, os métodos baseados em exemplos ainda constituem um desafio. Existem poucas referências bibliográficas a respeito, algumas mostradas a seguir.

3.2.1 Uma abordagem de segmentação de imagens por aprendizagem

Nesta abordagem, [AYA04] realiza um processo de segmentação de imagens baseada em exemplos.

3.2.1.1 Mecanismo

O mecanismo usado neste processo é bastante parecido com o que será implantado nesta pesquisa. A grande diferença está na profundidade de cores utilizada (256 cores).

A partir de um par de imagens amostra/ideal, o algoritmo é treinado, mapeando cada pixel de acordo com suas características, através de janelas. A cada cor $(0, \dots, 255)$ é atribuída uma saída $(0, 1)$, indicando se o resultado é fundo ou objeto. No caso de redundância entre os dados, o tamanho da janela é aumentado, até que não haja mais dados redundantes (o mesmo valor de cor resultando em fundo e objeto).

Com uma matriz contendo todos os valores do treinamento, novas imagens são submetidas para segmentação. Para determinar o resultado do pixel em análise na nova imagem, são extraídas suas características. Estas são comparadas com as características previamente extraídas no treinamento. A comparação é realizada com o classificador k -NN, e o resultado determina se o pixel vai ser fundo ou objeto.

3.2.1.2 Vantagens e desvantagens

Segundo os resultados mostrados por [AYA04], este método mostra-se muito eficiente. A única desvantagem real dele é o tempo de treinamento que, dependendo do tamanho da imagem, pode ser bastante longo, dependendo do equipamento utilizado. Conseqüentemente, a classificação para imagens novas também torna-se demorada, visto que o k -NN, segundo o autor, não é muito rápido neste caso. Um exemplo de treinamento e imagem segmentada [AYA04] pode ser visto nas figuras 3.12 (imagem de treinamento e sua solução ideal), 3.13 (nova imagem e sua solução ideal) e 3.14 (solução obtida através do método de segmentação por aprendizagem). É importante ressaltar que as imagens ideais são obtidas através de um especialista na área.

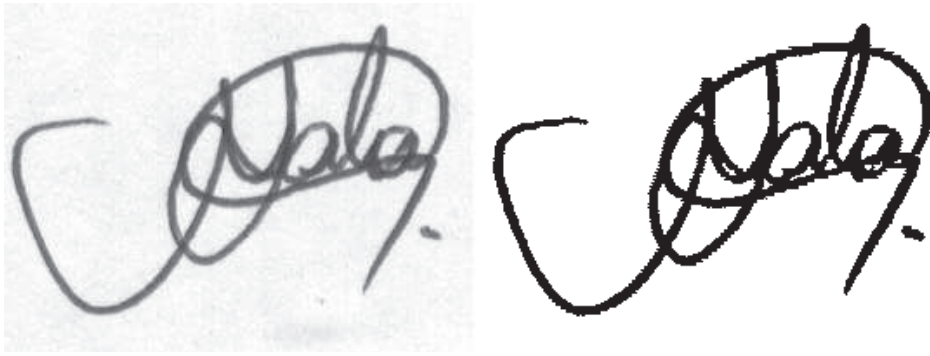


Figura 3.12: Imagem amostra e solução ideal



Figura 3.13: Nova imagem amostra e solução ideal

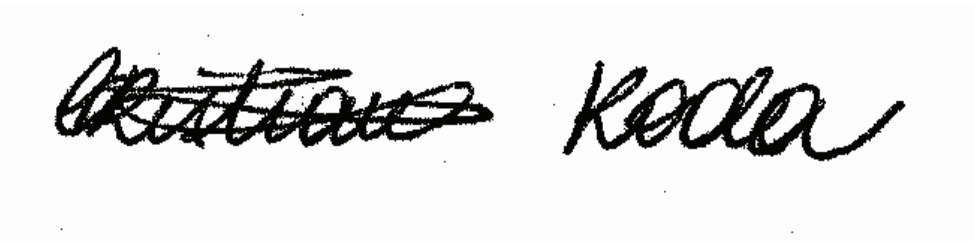


Figura 3.14: Binarização obtida por aprendizagem

3.2.2 Construção rápida de operadores morfológicos eficientes através de aprendizagem

Nesta abordagem, [KIM97a] realiza um processo de construção de operadores morfológicos genéricos, usando pares de imagens como exemplos de treinamento.

3.2.2.1 Mecanismo

O operador morfológico é representado como uma árvore binária, chamada árvore de corte. Qualquer operador morfológico pode ser representado por uma árvore de corte.

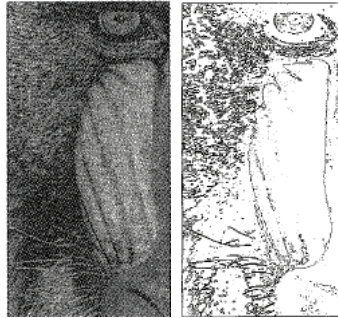


Figura 3.15: Exemplo de imagem de treinamento

Um hiperplano paralelo divide o sistema de coordenadas L . O hiperplano divide os pontos conhecidos em dois conjuntos de mesma cardinalidade. Para cada semi-espaço obtido, o sistema de divisão continua recursivamente, gerando hiper-paralelepípedos. O processo pára quando cada hiper-paralelepípedo contém apenas pontos que contenham o mesmo valor de saída. Cada hiperplano corresponde a um nó da árvore de corte e cada hiper-paralelepípedo corresponde a uma folha.

Uma árvore de corte pode ser construída através de imagens-exemplo. Supondo que um exemplo consiste de uma par de imagens, os pixels da imagem amostra, baseados em sua janela, são convertidos em um ponto no espaço L . Para cada ponto neste espaço, há uma saída relacionada. Esta sequência pode ser usada como exemplo de treinamento do algoritmo. Através dela, novas imagens podem ser submetidas para serem processadas.

3.2.2.2 Vantagens e desvantagens

Este algoritmo pode ser usado tanto em imagens binárias, em tons de cinza e coloridas. Pode ser aplicado para eliminação de ruídos, reconhecimento de padrões e texturas. Enquanto um algoritmo elementar de morfologia aplica, por exemplo, 100000 erosões em uma imagem, levando até 1 dia para completar a tarefa, este algoritmo pode realizar o mesmo processo em 15 segundos, como mostrado em [AYA04].



Figura 3.16: Exemplo de imagem original, ideal e o resultado obtido

3.2.3 Redução de dimensionalidade utilizando entropia condicional média: aplicações em filtragem de imagens e em reconhecimento de texturas

Este mecanismo de [MJCJB] apresenta uma forma de redução de dimensionalidade utilizando a entropia condicional média. Em outras palavras, procura-se identificar um subespaço suficientemente reduzido de características que seja capaz de representar qualquer padrão conhecido de acordo com determinado critério.

3.2.3.1 Mecanismo

Para entender a entropia como critério de seleção de características, é necessário considerar o gráfico onde os rótulos Y são representados pela abscissa e a probabilidade de um padrão ser rotulado como $Y = y$ dada a ocorrência de x , representada pela ordenada, como na figura 3.17. Se ele apresentar um pico saliente (massa de probabilidade bem concentrada), isso significa que a entropia condicional é pequena, isto é, prediz os rótulos de Y com boa confiança. Por outro lado, o gráfico apresenta-se achatado (massa de probabilidades bem distribuída), a entropia é alta, significando que não prediz Y . Assim, a entropia condicional pode ser usada como um critério bastante apropriado para realizar a seleção de características.

Para decidir se X prediz Y , basta calcular a entropia condicional média de todas as instâncias, ponderada pelo número de ocorrências de cada uma das instâncias no conjunto de treinamento.

São construídos W -operadores através de conjuntos de treinamento de pares de imagens ideal/observada, nas quais os vetores de características são obtidos através de translações de uma janela de dimensões fixas sobre as imagens observadas e seus respectivos rótulos são obtidos através do pixel central da janela correspondente. Essas

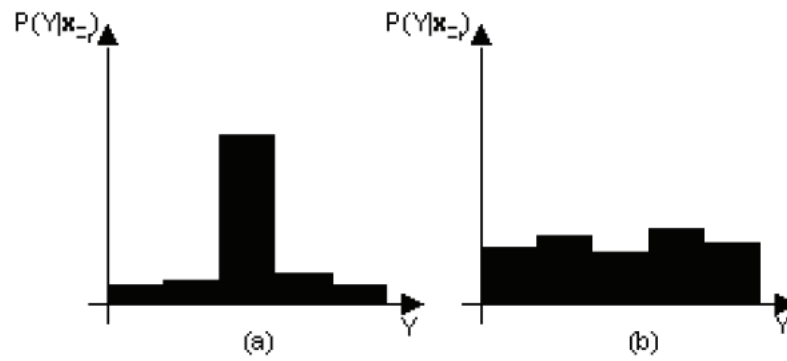


Figura 3.17: (a) baixa entropia (b) alta entropia

características são parciais, ou seja, há a possibilidade de obter características através de apenas um determinado espaço da imagem. Essa porcentagem pode ser definida, como por exemplo 10%, 20% ou 40%.

3.2.3.2 Vantagens e desvantagens

O método apresentado em 3.2.3 foi aplicado para processar dois tipos de imagens: filtragem de imagens ruidosas binárias e reconhecimento de texturas em imagens em níveis de cinza.

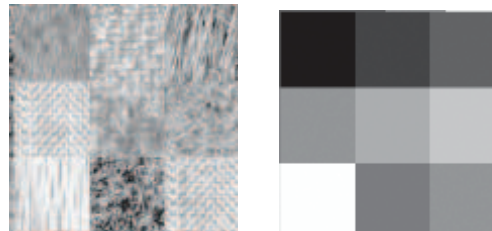


Figura 3.18: Imagem amostra para treinamento e sua solução ideal

No caso das imagens ruidosas, o método proposto se mostrou superior, principalmente nos casos em que as imagens apresentam formas bem finas. Já no reconhecimento de texturas, cada região rotulada recebeu o rótulo correto com maioria significativa, como pode ser visto na figura 3.18 (imagens de treinamento) e 3.19 (nova imagem, sua solução ideal e resultado final).

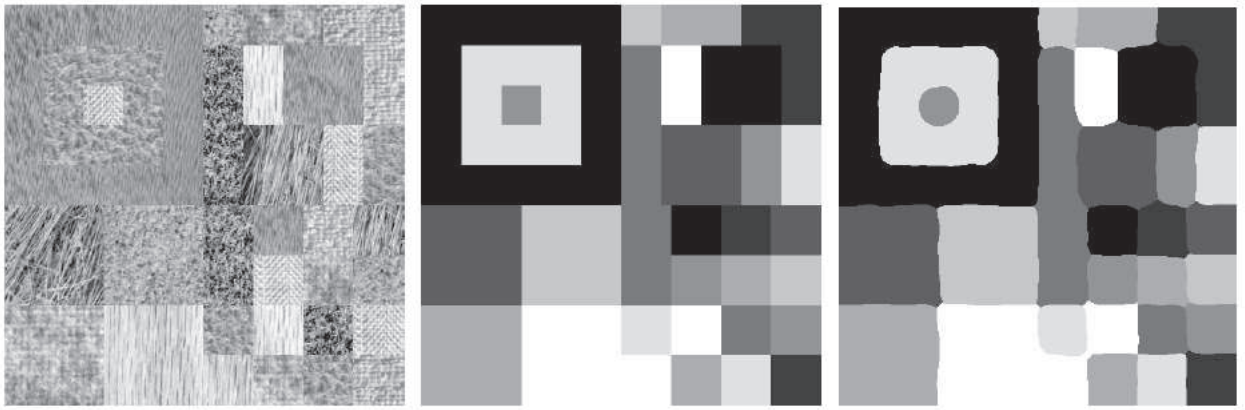


Figura 3.19: Nova imagem; solução ideal; resultado utilizando 40% para aprendizagem

3.3 Classificador

Para correta interpretação dos dados gerados pela aprendizagem e armazenados no formato de uma matriz (ou um vetor de vetores), faz-se necessário um classificador supervisionado, onde o usuário pode interagir com o algoritmo, fornecendo a quantidade de classes que o classificador irá dividir os dados armazenados. O classificador escolhido para este caso é o k -NN (*k nearest neighbor (k vizinhos mais próximos)*), por tratar-se de um algoritmo baseado em instâncias para aproximar funções alvo, de valor real ou de valor discreto, assumindo que as instâncias correspondem a pontos no espaço d -dimensional [AYA04].

3.3.1 Vizinho mais próximo - 1-NN

O vizinho mais próximo é o modelo de aprendizagem baseada em instâncias utilizada nesta abordagem. Este modelo tem sido amplamente estudado e aplicado em diversas áreas, e pode ser definido da seguinte forma [AYA04]: dado um novo vetor para classificação, denotado por x , e um conjunto de treinamento na forma de amostras identificadas:

$$H_n = \{x_1, x_2, \dots, x_n\} \quad (3.4)$$

sendo cada vetor x_i identificado como pertencendo a uma classe. Se x' é a amostra mais próxima a x , x recebe a mesma identificação que a classe x' .

3.3.1.1 Distância Euclidiana

Para que se possa utilizar a regra do vizinho mais próximo, é necessária uma regra para definir quem é o vizinho mais próximo. Nesta abordagem, a distância euclidiana é utilizada.

O valor da distância euclidiana pode ser definido como:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.5)$$

onde n é a quantidade de características utilizadas.

3.3.2 k vizinhos mais próximos - k -NN

A abordagem 1-NN pode ser modificada para, ao invés de se analisar apenas o vizinho mais próximo, analisar as k -amostras mais próximas em H . A figura 3.20 mostra um exemplo utilizando 3-NN e 4-NN.

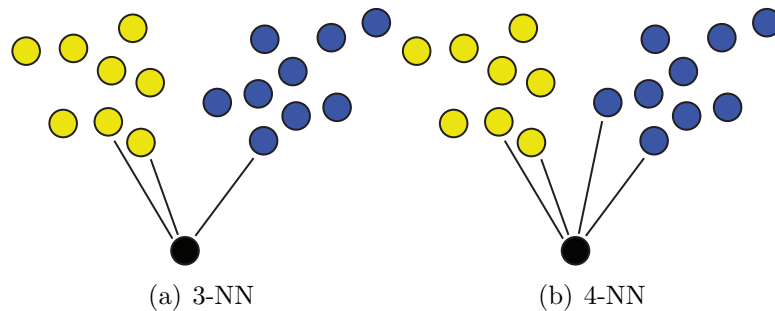


Figura 3.20: Exemplo de k vizinhos mais próximos

3.3.2.1 Valor ideal de k

Assim como em [AYA04], a abordagem de afinamento aqui apresentada utiliza-se de imagens em níveis de cinza (geradas pela função distância sobre a imagem binária) para extrair as características. Pode-se considerar que o tamanho da população é 256, referente aos 256 níveis de cinza existentes. [CM86] propõe a fórmula mostrada na equação 3.6 para cálculo do valor ideal de k .

$$k \approx \sqrt{\frac{m}{C}} \quad (3.6)$$

Na equação 3.6, k é o número de vizinhos, m é o tamanho da população e C é o

número de classes que se deseja obter. Levando-se em consideração que esta abordagem deseja separar imagens em níveis de cinza ($m = 256$) em duas classes (fundo e objeto afinado, logo $C = 2$), chega-se à seguinte equação:

$$k \approx \sqrt{\frac{256}{2}} \therefore k \approx \sqrt{128} \therefore k \approx 11, 31 \quad (3.7)$$

Segundo a equação 3.7, o valor ideal de k seria 11, valor que será analisado no capítulo 5. Entretanto, na literatura existem divergências entre vários autores sobre um valor ideal de k que minimize erros de classificação. Segundo [CE67], a regra do 1-NN é a melhor. Entretanto, há autores (como [KIM97b]) que consideram que o valor de k depende da aplicação.

3.4 Comentários finais

Neste capítulo foram mostrados alguns mecanismos que se utilizam de aprendizagem para realizar suas funções. Além disso, foi introduzido o conceito de função distância e o k -NN, o classificador utilizado nesta abordagem.

Capítulo 4

Metodologia proposta

Os métodos de afinamento mais tradicionais têm como característica a varredura da imagem pixel a pixel, aplicando a cada um destes pixels uma máscara, que irá decidir se o pixel será excluído ou não. Como visto anteriormente no capítulo 3, estes métodos possuem falhas que podem vir a prejudicar o resultado final do processamento da imagem.

4.1 Afinamento por aprendizagem através de exemplos

Este trabalho propõe uma abordagem que, ao invés de varrer a imagem aplicando máscaras pré-definidas, recebe duas imagens - a imagem original e afinada - e é treinada a partir delas, extraíndo características que vão compor um conjunto que irá ser usado para o afinamento de novas imagens. Submetendo novas imagens ao algoritmo treinado, ele será capaz de afiná-las baseado em seu modelo de treinamento. A idéia básica da abordagem proposta é utilizar o conhecimento do usuário, em transformar uma imagem binária em sua respectiva imagem afinada. Se o usuário sabe como deve ficar a imagem final, porque não compartilhar este conhecimento com o computador?

A abordagem apresentada consiste em ensinar o computador a afinar, empregando para isto imagens cujas soluções são conhecidas. Diferentemente das técnicas que utilizam redes neurais ou algoritmos genéticos na aprendizagem, a presente abordagem aprende a partir de um único par de imagens amostra/ideal.

4.2 Metodologia

Como mostrado anteriormente, o processo de afinamento por aprendizagem através de exemplos consiste em duas etapas principais: aprendizagem e afinamento (figura 4.1).

Na primeira fase, antes de submeter a imagem para aprendizagem, faz-se necessário

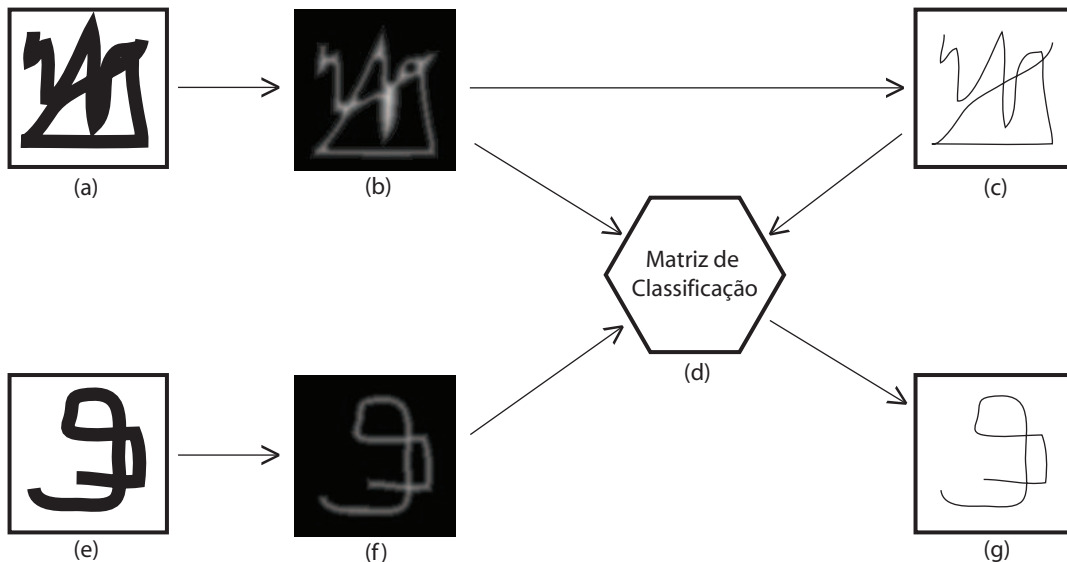


Figura 4.1: Esquema geral da abordagem proposta de afinamento por aprendizagem (a) imagem amostra, (b) função distância da imagem amostra, (c) solução ideal da imagem amostra, (d) matriz de classificação, (e) nova imagem submetida, (f) função distância da imagem submetida e (g) imagem afinada

extrair sua função distância (figuras 4.1(a) e 4.1(b)). As características então extraídas são armazenadas em uma matriz chamada “*matriz de classificação*” (figura 4.1(d)). Estas características caracterizam o pixel em análise, em relação a sua vizinhança.

O próximo passo é afinar novas imagens, empregando a matriz criada para classificar cada pixel analisado na função distância da imagem submetida (figura 4.1(e) e 4.1(f)). O resultado disso (figura 4.1(g)) é o a imagem afinada segundo as características aprendidas pelo algoritmo em sua fase de treinamento.

4.2.1 Fundamentos da metodologia proposta

Esta metodologia propõe o aproveitamento do conhecimento e a experiência do usuário, para afinar uma imagem binária. O usuário que conhece exatamente o que deseja afinar da imagem, interage com o computador através da transmissão deste conhecimento.

Como dito anteriormente, o método pode ser dividido em duas etapas principais, a aprendizagem e o afinamento. O processo de aprendizagem utiliza três imagens, a original, sua função distância e a ideal. A partir destas imagens é montada uma matriz de classificação para cada um dos pixels, do par de imagens analisado. No processo de afinamento de outras imagens é utilizado o conhecimento adquirido e armazenado na matriz de classificação. Cada uma das novas imagens é afinada através da estratégia dos vizinhos mais próximos, que procura, para cada pixel a ser processado, a melhor solução

na matriz de classificação.

4.2.2 Janelas

Cada pixel em análise possui uma janela ao seu redor. Essa janela fornece dados para que sejam estabelecidas as características deste pixel.

O tamanho da janela pode variar, de acordo com a necessidade de extrair características mais precisas. Isso será visto com mais detalhes na seção 4.2.5.

4.2.3 Extração das características

O objetivo em extrair as características é obter aquelas que melhor definam os relacionamentos entre o pixel amostra e seu correspondente pixel ideal. A disposição dos pixels vizinhos, em relação ao ponto em estudo, é de fundamental importância, principalmente considerando-se que a informação binária não é suficiente para um efetivo afinamento [AYA04]. Deve ser considerado, além dos níveis binários, a proximidade dos pontos, continuidade de áreas, etc. Esta noção subjetiva deve ser levada em consideração no momento da aprendizagem.

A princípio, as seguintes características são capazes de definir o pixel em análise.

1. **Nível de cinza do pixel:** é o valor do pixel na imagem. Neste caso corresponde ao valor da distância gerada pela função distância. É o marcador que indica com qual pixel está sendo feita a análise.

$$N(c) = x_i \text{ sendo } x_i \in [0, 1, \dots, 255] \quad (4.1)$$

2. **Média (μ):** caracteriza o comportamento do pixel e a influência do ruído em relação à janela utilizada [AYA04]. μ (*média*)

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i, \quad (4.2)$$

onde N corresponde ao número de pixels da janela utilizada.

3. **Variância (σ^2):** mostra o comportamento dos níveis de cinza na janela utilizada, caracterizando o aspecto homogêneo e a influência do ruído [AYA04]. σ^2 (*variância*)

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2 \quad (4.3)$$

4. **Obliquidade**: quociente do momento de terceira ordem e a terceira potência do desvio padrão (σ) [AYA04].

$$Ske = \frac{1}{\sigma^3} \sum_{i=1}^N (x_i - \mu)^3 \quad (4.4)$$

5. **Curtose**: quociente do momento de quarta ordem em torno da média μ e a quarta potência do desvio padrão (σ) [AYA04].

$$Cur = \frac{1}{\sigma^4} \sum_{i=1}^N (x_i - \mu)^4 \quad (4.5)$$

6. **Número de conectividade (NC)**: o número de conectividade, neste caso, é definido como sendo o número de transições de uma cor para outra entre os pixels, na região da janela que circunda o pixel central p [ZS84].

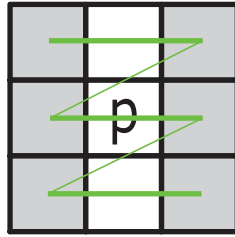


Figura 4.2: Exemplo de varredura para um pixel

$$CN(p) = \sum_{k=0,2,4,6} \overline{p[k]} * (p[k+1] \cup p[k+2]) \quad (4.6)$$

Levando em consideração a janela apresentada na figura 4.2, o número de conectividade resultante é 6.

7. **Valores máximos e mínimos**: dentro da janela analisada, o maior e o menor valor de um pixel é armazenado. Se o maior valor coletado for igual ao pixel analisado, há uma grande possibilidade que este pixel seja parte do esqueleto da imagem. Já se o valor mínimo for igual ao valor de p , o pixel pode corresponder ao fundo da imagem.

$$m(p) = p \in P : max(p) \quad (4.7)$$

$$m(p) = p \in P : \min(p) \quad (4.8)$$

As equações 4.7 e 4.8 representam estes valores, onde P é conjunto de pixels da janela analisada.

4.2.4 Aprendizagem

A imagem digital pode ser definida como sendo o conjunto discreto de números correspondentes a cada pixel que conforma esta imagem. Neste caso, a imagem binária é composta por um conjunto discreto de números no intervalo de 0 até 1, e a imagem em níveis de cinza por um conjunto discreto no intervalo de 0 até 255.

Seja uma imagem binária digital, e x_i o valor correspondente a cada pixel, no domínio discreto, definem-se que uma imagem binarizada $x_i \in [0, 1]$, podendo assumir 2 valores, onde $x_i = 0$ é o nível preto e $x_i = 1$ é o nível branco.

Seja uma imagem em níveis de cinza digital, e x_i o valor correspondente a cada pixel, no domínio discreto, definem-se que uma imagem binarizada $x_i \in [0, \dots, 255]$, podendo assumir 255 valores, onde $x_i = 0$ é o nível preto e $x_i = 255$ é o nível branco.

Na etapa de aprendizagem são processadas duas imagens [AYA04]: uma em níveis de cinza (gerada pela função distância aplicada em uma imagem binária), chamada amostra, e sua correspondente solução ideal afinada. O termo “solução ideal” ou ainda, imagem ideal significa o resultado esperado pelo usuário.

São analisadas as vizinhanças de cada par de pixels da imagem amostra e a correspondente imagem ideal, extraindo-se um conjunto de características. Nesta etapa é gerado um vetor de características, como mostrado na figura 4.3, para cada pixel e seu pixel associado, junto a seu valor do conceito alvo, o qual será reutilizado no momento do afinamento de novas imagens. O conjunto de características é normalizado no intervalo $[0, \dots, 1]$.

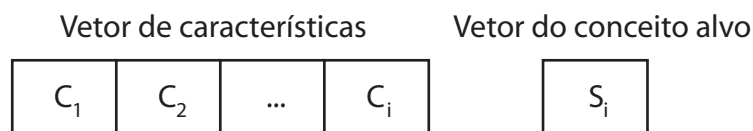


Figura 4.3: Vetor de características [AYA04] onde C_i representa a i -ésima característica; e S_i representa a saída ideal do i -ésimo vetor de característica

A figura 4.3 também mostra que cada vetor possui um respectivo valor de saída ideal. O conjunto obtido de todos os vetores de características, ordenados de acordo com

um critério, é representada pela “*matriz de classificação*”. Esta matriz abrange todos os pixels da imagem amostra. Os passos da etapa de aprendizagem podem resumidos da seguinte forma:

- O cálculo do vetor de características, normalizado, para cada pixel, e sua correspondente vizinhança (de janela centralizada), na imagem amostra;
- A definição automática do tamanho de janela (seção 4.2.5), a partir da análise de cada pixel na imagem amostra e sua saída ideal. No caso de existirem dois pixels de entrada apresentando o mesmo valor binário, com vetores de características iguais, porém com saídas ideais diferentes, o tamanho da janela é aumentado até que estas ambigüidades sejam eliminadas.

Um esquema mais detalhado dos passos da etapa de aprendizagem é apresentado na figura 4.4.

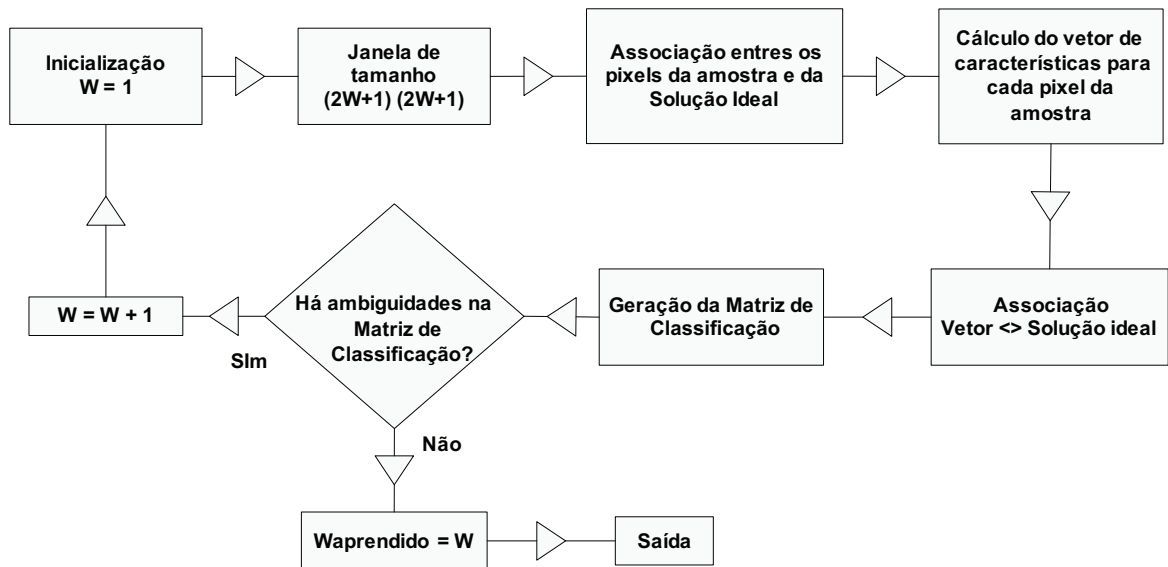


Figura 4.4: A etapa de aprendizagem e definição automática do tamanho da janela [AYA04]

4.2.5 Tamanho ideal da janela

A escolha do tamanho de janela apropriado é fundamental para o desempenho geral do processo de aprendizagem. Além disso, faz parte da estratégia para solucionar as confusões na aprendizagem.

A janela pode ser definida como um conjunto de pixels vizinhos ao pixel em análise. Possui forma quadrada e os lados possuem número ímpar para manter o pixel em análise

centralizado. Para evitar inconsistências na decisão, deve-se garantir que não existam pixels com saídas ideais diferentes das que tenham o mesmo valor binário e vetores de características iguais.

O cálculo dos vetores de características depende diretamente do tamanho da janela [AYA04]. Portanto, é necessário descobrir o tamanho de janela adequado, que soluciona essas inconsistências. Porém, deve exigir o menor tempo computacional possível. Ou seja:

- Quanto maior a janela, maior o tempo de processamento. Isso ocorre porque, para o cálculo das características, utiliza-se um maior número de pontos ou pixels. Portanto, é desejável utilizar sempre a menor janela possível;
- Janelas muito pequenas, em alguns casos, podem gerar vetores de características similares, porém com saídas ideais diferentes, criando ambigüidades;
- Como o tamanho da janela apropriada varia com o tipo de imagem utilizado, é recomendável que este tamanho seja ajustado, sem que haja intervenção manual do operador.

Devido às considerações anteriores, um algoritmo de seleção automática de janela se faz necessário [AYA04], no qual é processada a imagem inteira, iniciando-se num tamanho 3×3 . No final do processamento da imagem, verifica-se a existência de inconsistências na matriz de classificação. Caso exista alguma, reinicia-se o processamento da imagem, com o tamanho de janela superior (no caso, 5×5 ; logo 7×7 ; e assim por diante). Este algoritmo encontra-se esquematizado na figura 4.4. Por motivos práticos, o algoritmo de ajuste de janela deve ser limitado a um tamanho máximo (por exemplo 21×21).

Cada inconsistência achada é interpretada como a presença de um erro (ε) na aprendizagem. A inconsistência, ou ambigüidade, é definida como a presença de dois vetores de característica idênticos, porém com saídas diferentes. O algoritmo de aprendizagem e ajuste automático do tamanho de janela é projetado para admitir três condições de parada: No final do processo de aprendizagem, com um dado tamanho de janela, é computado o total de inconsistências achadas, ou seja o total de erros. Este total de erros $(\Sigma\varepsilon)_k$, é comparado com o valor total da iteração anterior $(\Sigma\varepsilon)_{k-1}$. Assim, são estabelecidas as três seguintes condições de parada:

- Total de erros na aprendizagem da iteração atual igual a zero;

$$(\Sigma\varepsilon)_k = 0 \tag{4.9}$$

- Total de erros na aprendizagem da iteração atual maior ao total calculado na iteração anterior;

$$(\Sigma\varepsilon)_k > (\Sigma\varepsilon)_{k-1} \quad (4.10)$$

- Número máximo de iterações na aprendizagem alcançado (por exemplo, na equação 4.11, 17;

$$(\Sigma\varepsilon)_{17} \quad (4.11)$$

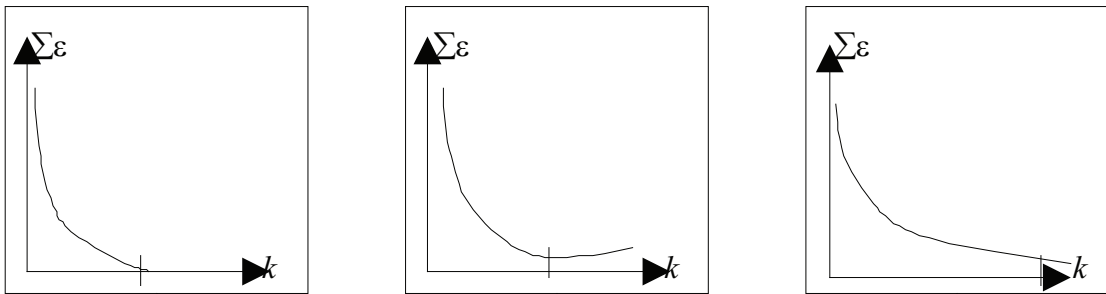


Figura 4.5: Gráfico de total de erros $(\Sigma\varepsilon) \times k$ [AYA04] onde $k = \text{número de iterações}$. (a) $(\Sigma\varepsilon)_k = 0$; (b) $(\Sigma\varepsilon)_k > (\Sigma\varepsilon)_{k-1}$ e (c) $(\Sigma\varepsilon)_{17}$.

Estes três critérios podem ser melhor compreendidos de forma gráfica, como mostrado na figura 4.5.

4.2.6 Matriz de classificação

A matriz de classificação armazena, de acordo com os valores binários da imagem amostra e com uma ordenação particular, o conjunto de vetores de características obtidos para todos os pares de pixels do par de imagens amostra/ideal [AYA04]. O número de vetores de características na matriz de classificação é igual ao número de pares de pixels aprendido da imagem submetida da amostra e sua correspondente imagem esperada ideal.

A matriz de classificação é construída com todos os vetores de características ordenados seqüencialmente. Na etapa de afinamento de novas imagens, a estratégia k -NN seleciona o vetor de características mais próximo ao pixel em análise. Nos casos de valores não contemplados, são incluídos os vetores de características, com valores imediatamente anterior e posterior, como sendo pertencentes à faixa válida.

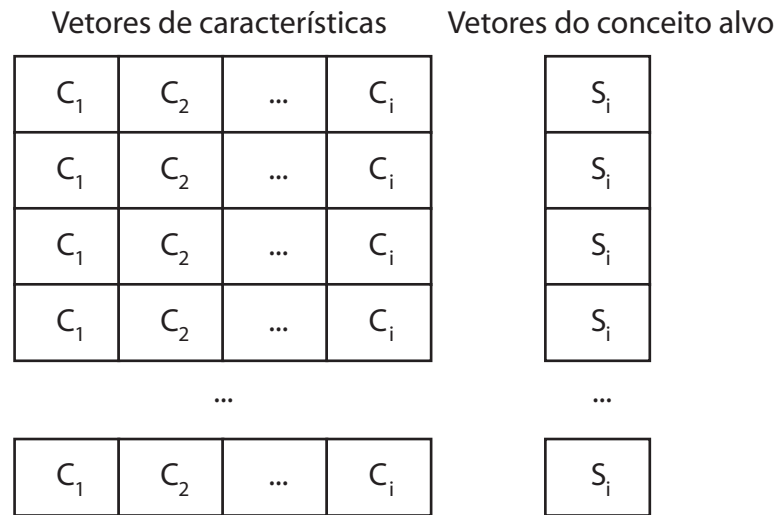


Figura 4.6: Exemplo de matriz de classificação

4.2.7 Afinamento de novas imagens por aprendizagem

Depois de concluída a etapa de aprendizagem, novas imagens do mesmo tipo podem ser afinadas. Os passos da etapa de afinamento por aprendizagem são [AYA04] (figura 4.7):

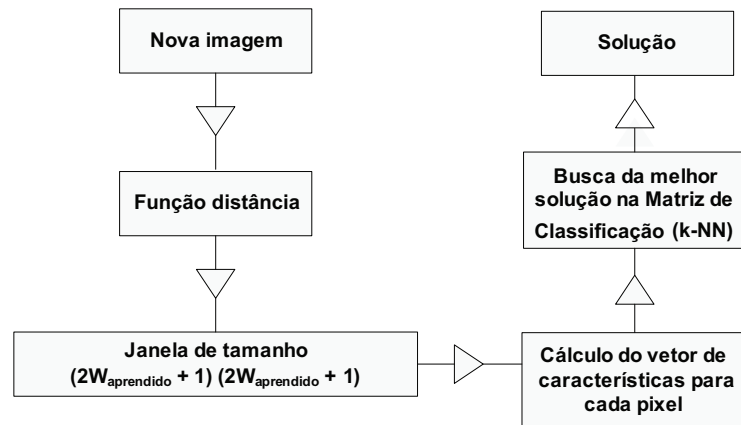


Figura 4.7: Processo de afinamento de imagens

1. Aplicação da função distância na imagem binária submetida, gerando uma imagem em níveis de cinza;
2. Cálculo de vetor de características, normalizado, para cada pixel na nova imagem. Para isto, utiliza-se a informação do tamanho da janela, obtida durante o processo de aprendizagem;
3. Escolha do vetor de características mais similar na matriz de decisão;

4. Atribuição do valor de saída do pixel armazenado desse vetor na matriz de classificação. A estratégia k -NN é utilizada para selecionar o vetor de características “mais similar” dentro da matriz de classificação, atribuindo-se assim o valor do pixel de saída armazenado.

4.3 Comentários finais

Neste capítulo foi apresentada a abordagem proposta de Afinamento por Aprendizagem. Foram detalhadas as etapas de aprendizagem e afinamento, incluindo a seleção das características, as considerações sobre o tamanho da janela e sobre o k -NN.

Capítulo 5

Testes e resultados

Este capítulo mostra os resultados de testes realizados em duas bases de imagens distintas, de acordo com cada combinação de características utilizada, o método utilizado na função distância e o valor de k no classificador k -NN.

A primeira base a ser utilizada consiste em um conjunto das 26 letras do alfabeto. Esta base servirá para analisar cada uma das características descritas nesta abordagem, assim como a melhor função distância e também o valor de k que obtém o melhor aproveitamento em termos de afinamento. Além disso, a partir dela serão feitas comparações com outros métodos de afinamento.

A segunda base, contendo imagens mais complexas (retiradas da Bíblia de Gutenberg [GUT03]), passará pelos mesmos testes da base de letras, mas com o objetivo principal de analisar como a abordagem proposta se comporta em um caso de imagens com uma complexidade mais elevada.

5.1 Base contendo as 26 letras do alfabeto

A base de letras utilizada contém letras consideradas comuns em documentos gerados através de um computador. Alguns exemplos são mostrados na figura 5.1.



Figura 5.1: Exemplos de letras da base utilizada

Vários testes foram realizados com esta base, todas com o objetivo de afiná-las da forma mais parecida possível com as imagens consideradas ideais usadas no treinamento. Como exemplo, as imagens ideais da figura 5.1 podem ser vistas na figura 5.2. Estes testes realizados analisaram cada uma das 26 letras e o resultado de sua aprendizagem nas 25 letras restantes.

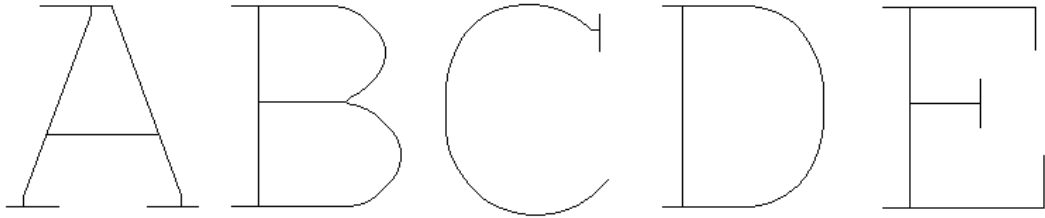


Figura 5.2: Imagens ideais das letras mostradas na figura 5.1

Para a extração das características, se faz necessário calcular a função distância de cada imagem binária. As imagens contendo a função distância de cada uma das letras da figura 5.1 podem ser vistas nas figuras 5.3(a) a 5.3(f).

Tomando como base a aprendizagem realizada com o par de imagens amostra/ideal representadas apenas pela letra “A”, por exemplo, alguns resultados são mostrados na figura 5.4.

As imagens afinadas da figura 5.4 mostram que, se a aprendizagem for insuficiente, o resultado pode não ser tão bom. Como a letra “A” não possui curvas em sua forma, essa característica não conseguiu ser reproduzida nas novas imagens submetidas ao método de afinamento, como pode ser visto nas imagens 5.4(a), 5.4(c), 5.4(e), 5.4(f), 5.4(g) e 5.4(h). Por isso é necessário realizar uma aprendizagem capaz de reproduzir todas as características do conjunto de imagens a ser submetido para afinamento.

Um outro exemplo é a figura 5.5(a), onde temos as letras “XMR” e a respectiva imagem ideal (figura 5.5(b)). Como é uma imagem mais completa, ela é capaz de reproduzir uma quantidade maior de características e, conseqüentemente, é capaz de afinar melhor outras letras. Isto pode ser claramente notado na figura 5.8, resultado do afinamento da imagem 5.6, baseado no que foi aprendido com a aprendizagem na imagem 5.5.



(a) através de Chamfer 3-4



(b) através de Chamfer 5-7-11



(c) através de Chessboard



(d) através de City-block



(e) através da Morfologia (cruz)



(f) através da Morfologia (quadrado)

Figura 5.3: Função distância das letras da figura 5.1 através de várias abordagens

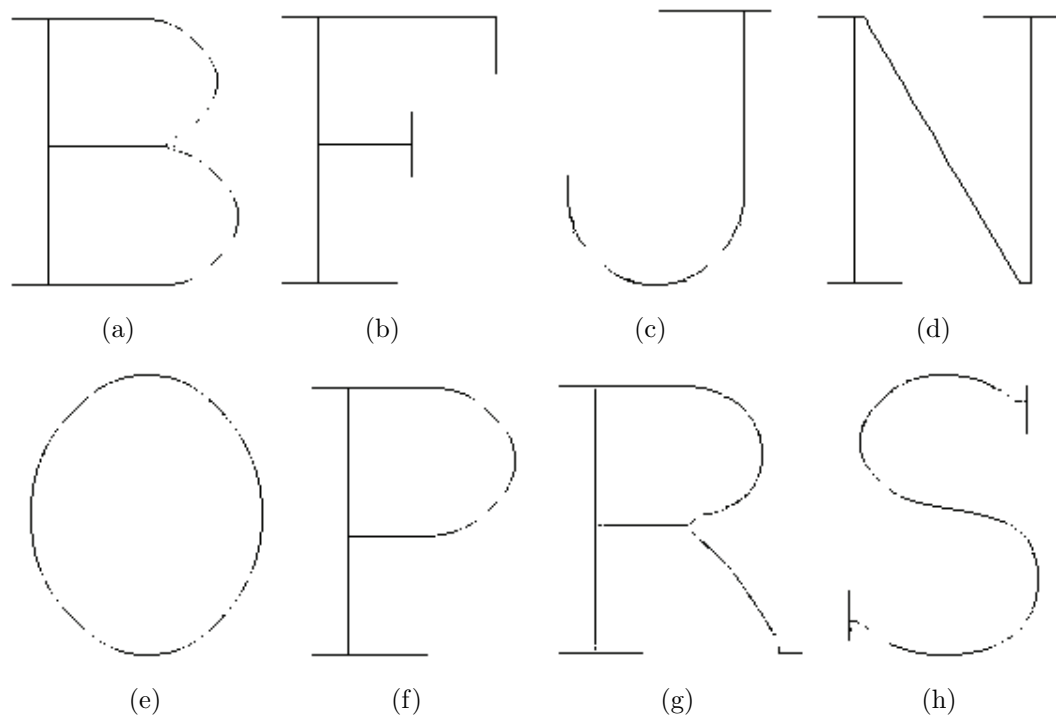


Figura 5.4: Exemplo de letras afinadas com o par amostra/ideal sendo a letra A



Figura 5.5: Imagens submetidas para extração de características

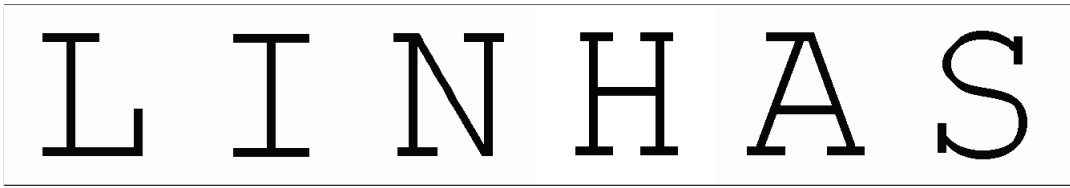


Figura 5.6: Nova imagem submetida para afinar com o que foi aprendido com a imagem 5.5

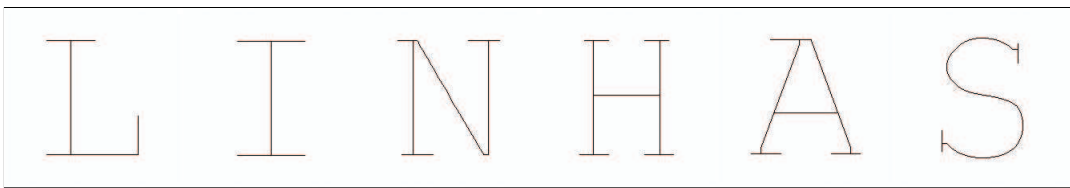


Figura 5.7: Resultado ideal do afinamento da imagem 5.6

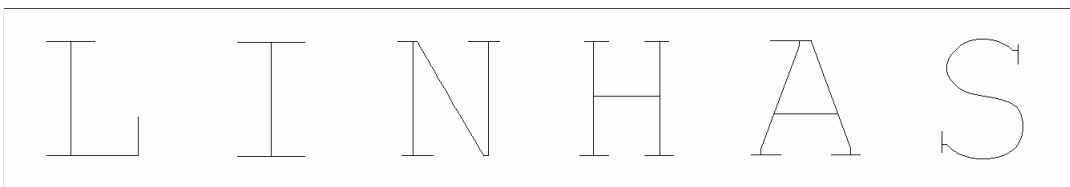


Figura 5.8: Afinamento pelo método de aprendizagem

5.1.1 Metodologia de avaliação de resultados

Afim de analisar os resultados obtidos pela abordagem proposta, uma metodologia de avaliação de resultados se faz necessária, como a apresentada em [AYA04]. Como o objetivo desta abordagem é gerar uma imagem com duas classes (objeto afinado e fundo), podemos considerar esta imagem como sendo I . O processo de avaliação nada mais é do que a comparação da imagem ideal I_I com a imagem afinada I_A , sendo que a classe objeto desta imagem é chamada de I_{AO} e o fundo de I_{AF} , da mesma forma que na imagem ideal temos I_{IO} e I_{IF} . O resultado destas relações pode ser visto nas equações 5.1 e 5.2.

$$I_A = I_{AO} \cup I_{AF} \quad (5.1)$$

$$I_I = I_{IO} \cup I_{IF} \quad (5.2)$$

A avaliação ocorre da seguinte forma: cada pixel em I_A encontrado em I_I é classificado como objeto, ou seja, pertencente à imagem afinada. Conseqüentemente, o oposto disto faz com que o pixel seja classificado como ruído, pertencente ao fundo da imagem. A taxa de afinamento é a razão entre o número de pixels existente na imagem afinada I_{AO} e na imagem ideal I_{IO} . Já o ruído é razão entre o número de pixels que foram segmentados em I_{AO} e não pertencem a I_{IO} . Desta forma, as taxas de afinamento e ruído podem ser definidas como nas equações 5.3 e 5.4, onde p representa o pixel analisado na imagem.

$$TX_{afinamento} = \frac{p \in (I_{AO} \cup I_{IO})}{p \in (I_{IO})} \quad (5.3)$$

$$TX_{ruído} = \frac{p \in (I_{AO} \cup I_{IF})}{p \in (I_{IF})} \quad (5.4)$$

5.1.2 Avaliação de características

Para obter o melhor conjunto de resultados e determinar quais características têm maior efeito no resultado final do afinamento, uma pré-análise foi realizada, testando cada característica individualmente. Para esta análise, as imagens da base de imagens utilizada foram submetidas ao processo de aprendizagem. Desta forma, temos uma matriz de classificação para cada imagem da base utilizada, cada uma contendo todas as características da respectiva imagem.

A seguir, cada matriz de classificação foi utilizada para afinar cada uma das imagens restantes da base. Cada característica foi usada individualmente com cada um dos

valores de k contidos no conjunto $\{1, 3, 5, 7, 9, 11, 13\}$. Este conjunto de valores de k tem como objetivo testar qual valor fornece o melhor resultado. Como há autores (capítulo 3.3.2.1) que enfatizam que os melhores resultados são com $k = 1$, há outros quem afirmam que a melhor forma é através do k obtido pela equação 3.6, e por fim há os que digam que o valor de k depende da aplicação, foi criado este conjunto, contendo todos os valores de k entre 1 e 11, ou seja, o valor único e o valor gerado pela equação. Adicionalmente, foi introduzido o valor $k = 13$ para analisar o que acontece com um valor de k acima do valor resultante da equação 3.6.

O resultado destes testes mostrou dois casos bem distintos. Enquanto a média, a variância, a obliquidade e a curtose forneceram resultados bastante interessantes, o número de conectividade e os valores máximos e mínimos pouco ajudaram na obtenção de um resultado satisfatório, ficando com taxas inferiores a 35% no percentual de pixels iguais entre a imagem afinada e a ideal. Com base nisto, estas duas características foram descartadas dos outros testes, sendo utilizadas apenas as quatro primeiras.

Tendo este conjunto de quatro características, o próximo passo é encontrar a combinação entre elas que forneça o melhor resultado em termos de afinamento. Como um conjunto de quatro características pode fornecer 15 combinações diferentes, cada uma destas 15 combinações foi usada para gerar novas matrizes de classificação para cada uma das imagens do subgrupo selecionado. Utilizando-se do algoritmo k -NN com o valor de $k = 1$, as matrizes de classificação geradas foram utilizadas para afinar cada uma das imagens restantes neste subgrupo da base, obtendo-se o percentual de afinamento e ruído. Desta forma, a combinação de características que melhor afina uma imagem pode ser encontrada.

Por fim, a melhor combinação encontrada foi submetida à uma nova análise, com o objetivo de achar o k que apresenta o melhor resultado de afinamento.

Como a imagem binária submetida à aprendizagem passa pela função distância para extração de características, faz-se necessário analisar qual função distância oferece o melhor resultado. Para esta análise, o mesmo subgrupo de imagens dos testes anteriores é usado. É gerada a função distância de cada uma das imagens deste subgrupo, através de cada um dos algoritmos apresentados na seção 3.1 (Chamfer 3-4, Chamfer 5-7-11, City-block, Chessboard e Morfologia Matemática). A melhor combinação de características e o k que apresentou o melhor resultado são utilizados.

5.1.3 Avaliação dos resultados

Utilizando-se de uma base contendo as 26 letras do alfabeto, os testes descritos na seção 5.1.2 foram realizados. O primeiro teste (figuras 5.9 e 5.10, tabelas 5.1, 5.2, 5.3 e 5.4) qual identifica a melhor característica isolada, analisando-as juntamente com os valores de k previamente selecionados na seção 5.1.2, ou seja, com k variando entre os valores do conjunto $\{1,3,5,7,9,11,13\}$.

Tabela 5.1: Resultados do afinamento em letras para a média de acordo com o valor de k

Média		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	95.95 \pm 09.90	0.08 \pm 0.20
3	92.89 \pm 13.99	0.26 \pm 0.40
5	92.32 \pm 17.03	0.29 \pm 0.36
7	92.68 \pm 17.35	0.31 \pm 0.37
9	92.90 \pm 17.66	0.35 \pm 0.38
11	93.02 \pm 17.66	0.38 \pm 0.39
13	93.04 \pm 17.56	0.40 \pm 0.39

Tabela 5.2: Resultados do afinamento em letras para a variância de acordo com o valor de k

Variância		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	96.42 \pm 10.60	0.02 \pm 0.04
3	92.36 \pm 15.61	0.21 \pm 0.34
5	92.64 \pm 16.15	0.29 \pm 0.36
7	93.05 \pm 16.50	0.35 \pm 0.38
9	93.06 \pm 16.30	0.39 \pm 0.39
11	93.42 \pm 16.15	0.41 \pm 0.40
13	93.45 \pm 16.31	0.43 \pm 0.41

Tabela 5.3: Resultados do afinamento em letras para a obliquidade de acordo com o valor de k

Obliquidade		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	89.67 \pm 19.57	0.40 \pm 0.39
3	94.10 \pm 09.80	0.27 \pm 0.35
5	93.51 \pm 15.26	0.40 \pm 0.40
7	94.00 \pm 15.19	0.44 \pm 0.42
9	94.33 \pm 15.18	0.47 \pm 0.43
11	94.44 \pm 15.23	0.47 \pm 0.43
13	94.45 \pm 15.24	0.48 \pm 0.43

Tabela 5.4: Resultados do afinamento em letras para a curtose de acordo com o valor de k

Curtose		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	95.28 ± 15.24	0.08 ± 0.27
3	93.13 ± 15.77	0.28 ± 0.37
5	92.91 ± 16.47	0.37 ± 0.39
7	92.36 ± 17.80	0.42 ± 0.40
9	92.62 ± 17.91	0.44 ± 0.41
11	92.73 ± 17.95	0.45 ± 0.41
13	92.76 ± 17.96	0.46 ± 0.41

A figura 5.9 mostra a taxa de afinamento para cada uma das características selecionadas com quantidades diferentes de vizinhos mais próximos (k), onde se pode notar uma diferença significativa no que se diz respeito ao valor de k . Quando $k = 1$, temos os melhores resultados para três das quatro características, ou seja, média, variância e curtose. Porém, quando $k > 1$, acontece o inverso. A melhor característica passa a ser a obliquidade, com uma taxa de afinamento um pouco melhor que as outras três características, que praticamente obtêm o mesmo resultado.

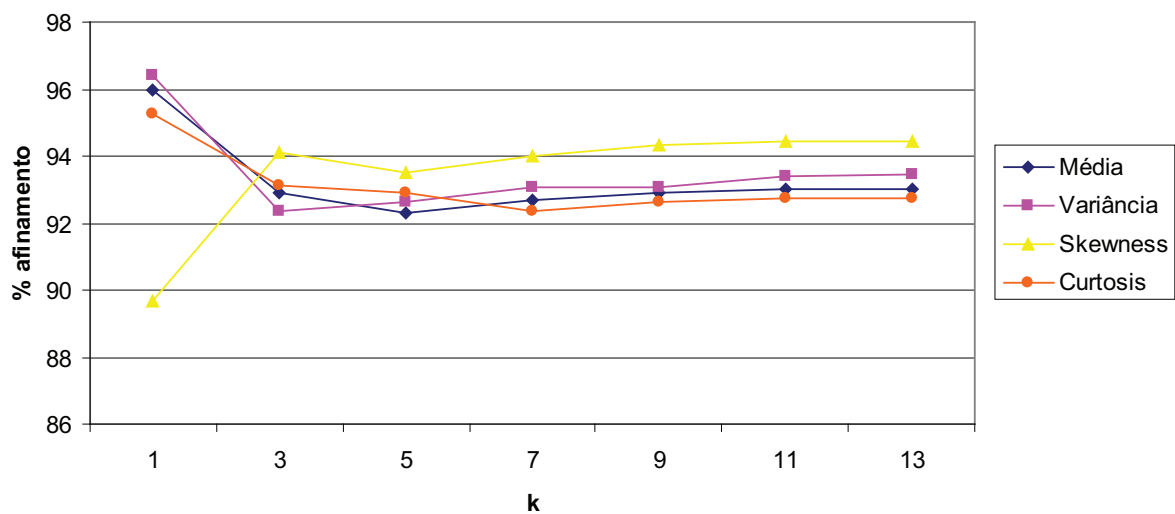


Figura 5.9: Afinamento da base de letras para cada característica isolada e o valor de k vizinhos mais próximos

Por outro lado, o ruído apresentado no afinamento da base de letras (figura 5.10) indica que, mesmo a obliquidade sendo uma característica melhor quando $k > 1$, sua taxa de ruído é sempre maior que as outras três características, principalmente quando $k = 1$. Aliás, quando $k = 1$ temos os menores valores de ruído para as outras três características.

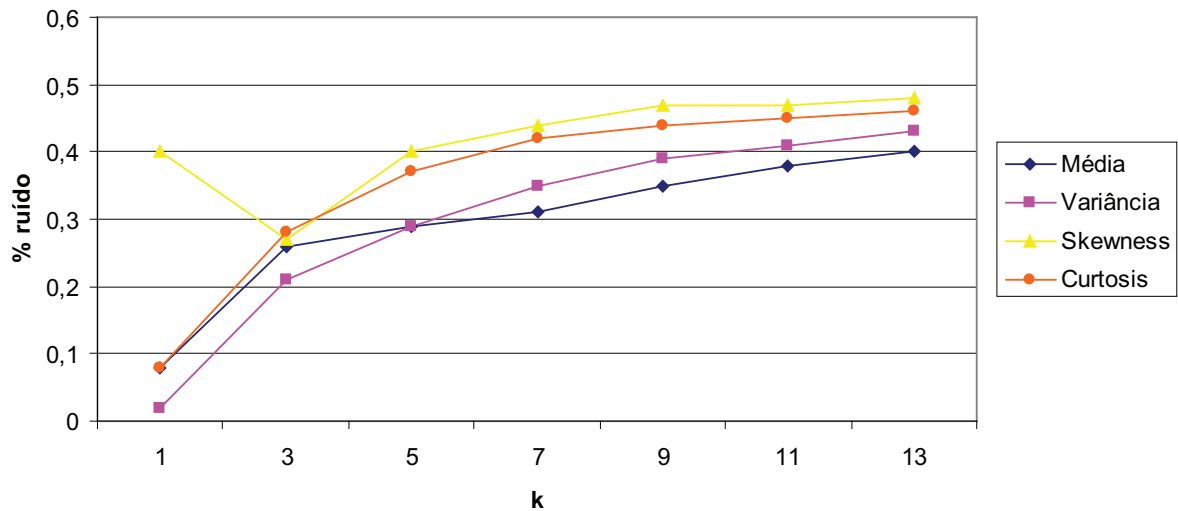


Figura 5.10: Ruído da base de letras para cada característica isolada e o valor de k vizinhos mais próximos

Se compararmos os dois gráficos nas figuras 5.9 e 5.10, a característica que obteve o melhor resultado para esta base é a variância, com a taxa de afinamento e ruído ($\mu \pm \sigma$) sendo, respectivamente, $96,42 \pm 10,60$ e $0,02 \pm 0,04$ para o parâmetro $k = 1$.

O próximo passo, de acordo com o que foi descrito na seção 5.1.2, é determinar qual combinação de características é a mais eficiente. Para isso, o processo de afinamento foi realizado na base de letras utilizando cada uma das 15 combinações possíveis. O resultado pode ser visto na tabela 5.5, onde m é a média, v é a variância, s é a obliquidade e c é a curtose.

De acordo com a tabela 5.5 e a figura 5.11, a melhor taxa de afinamento para cada combinação de características acontece com as características *média*, *variância* e *obliquidade* combinadas (mvs), com a taxa de afinamento sendo $97,35 \pm 6,05$. Curiosamente, a característica que apresenta a pior taxa de afinamento neste caso é a obliquidade ($89,67 \pm 19,57$), mostrando que isoladamente esta característica não é capaz de definir tão bem o ponto analisado na imagem.

Em relação ao ruído (tabela 5.5 e a figura 5.12), a situação é bastante parecida. Novamente a obliquidade se apresenta como a pior alternativa, com uma taxa de ruído quase três vezes maior que as outras combinações ($0,40 \pm 0,39$). Em contrapartida, a variância apresenta a mais baixa taxa de ruído ($0,02 \pm 0,04$), a mesma que obtém quando combinada com a média. Logo a seguir vem a combinação entre média, variância e obliquidade (mvs), com $0,05 \pm 0,11$, comprovando que realmente a melhor combinação de características para esta base são essas três.

Tabela 5.5: Taxas de afinamento e ruído para as 15 combinações possíveis das quatro características utilizadas nesta abordagem, para o valor $k = 1$

Combinações de características para $k = 1$		
Característica	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
c (<i>curtose</i>)	95,28 \pm 15,24	0,08 \pm 0,27
s (<i>obliquidade</i>)	89,67 \pm 19,57	0,40 \pm 0,39
sc (<i>obliquidade e curtose</i>)	92,18 \pm 16,38	0,18 \pm 0,30
v (<i>variância</i>)	96,42 \pm 10,60	0,02 \pm 0,04
vc (<i>variância e curtose</i>)	95,37 \pm 15,08	0,08 \pm 0,27
vs (<i>variância e obliquidade</i>)	96,68 \pm 07,15	0,07 \pm 0,12
vsc (<i>variância, obliquidade e curtose</i>)	94,96 \pm 15,29	0,09 \pm 0,27
m (<i>média</i>)	95,95 \pm 09,90	0,08 \pm 0,02
mc (<i>média e curtose</i>)	95,28 \pm 15,24	0,08 \pm 0,27
ms (<i>média e obliquidade</i>)	94,33 \pm 15,12	0,15 \pm 0,34
msc (<i>média, obliquidade e curtose</i>)	94,70 \pm 15,57	0,11 \pm 0,28
mv (<i>média e variância</i>)	96,47 \pm 10,61	0,02 \pm 0,04
mvc (<i>média, variância e curtose</i>)	95,37 \pm 15,08	0,08 \pm 0,27
mvs (<i>média, variância e obliquidade</i>)	97,35 \pm 06,05	0,05 \pm 0,11
mvs (<i>média, variância, obliquidade e curtose</i>)	95,03 \pm 15,09	0,08 \pm 0,27

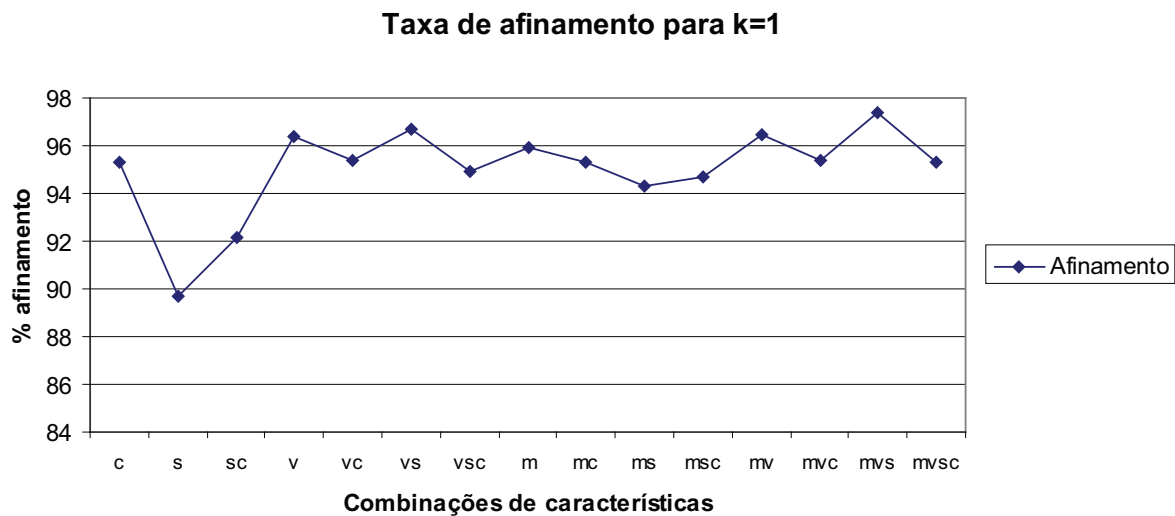


Figura 5.11: Taxa de afinamento da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$

Para efeito de comparação, o teste para achar a melhor combinação de características foi repetido, desta vez usando $k = 11$. Levando em consideração que há autores (como [CM86]) que afirmam que a melhor forma de determinar o valor de k é através da fórmula 3.6, será utilizado o valor encontrado por esta fórmula (mostrado em 3.7) para comparar com os resultados obtidos com $k = 1$.

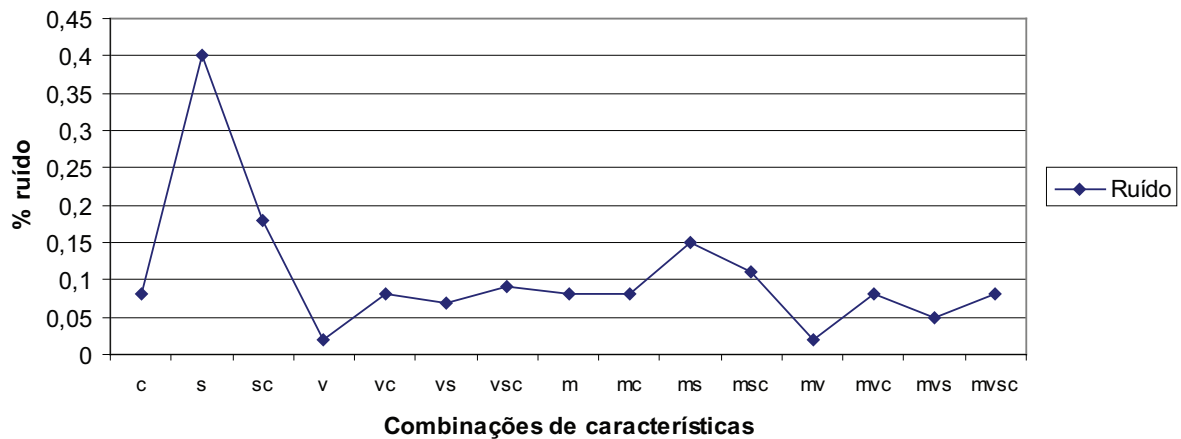
Taxa de ruído para $k=1$ 

Figura 5.12: Taxa de ruído da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$

Tabela 5.6: Taxas de afinamento e ruído para as 15 combinações possíveis das quatro características utilizadas nesta abordagem, para o valor $k = 11$

Combinações de características para $k = 11$		
Característica	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
<i>c</i>	92,73 \pm 17,75	0,45 \pm 0,41
<i>s</i>	94,44 \pm 15,23	0,47 \pm 0,43
<i>sc</i>	94,03 \pm 16,25	0,43 \pm 0,39
<i>v</i>	93,42 \pm 16,15	0,41 \pm 0,40
<i>vc</i>	93,71 \pm 16,32	0,40 \pm 0,41
<i>vs</i>	94,54 \pm 15,32	0,40 \pm 0,39
<i>vsc</i>	94,09 \pm 16,58	0,41 \pm 0,42
<i>m</i>	93,02 \pm 17,66	0,38 \pm 0,39
<i>mc</i>	94,56 \pm 16,41	0,33 \pm 0,38
<i>ms</i>	94,61 \pm 15,71	0,34 \pm 0,38
<i>mcs</i>	94,16 \pm 16,38	0,35 \pm 0,38
<i>mv</i>	93,83 \pm 16,48	0,32 \pm 0,38
<i>mvc</i>	94,11 \pm 16,45	0,33 \pm 0,39
<i>mvs</i>	94,72 \pm 15,41	0,34 \pm 0,38
<i>mvsc</i>	94,38 \pm 16,51	0,36 \pm 0,40

De acordo com os resultados mostrados na tabela 5.6, novamente temos a combinação de características entre a média, variância e obliquidade (*mvs*) como sendo aquela que obtém o melhor resultado (94,72 \pm 15,41). Entretanto, a média se mostrou como tendo a segunda pior taxa de afinamento (93,02 \pm 17,66), sendo melhor apenas que a curtose (92,73 \pm 17,75). Se compararmos com as taxas obtidas com $k = 1$, as obtidas

com $k = 11$ são mais compactas, estando mais próximas entre si, estando sempre em torno de 94%. As taxas mais baixas são maiores que as obtidas com $k = 1$. Em compensação, as taxas mais altas com $k = 11$ são menores que as obtidas com $k = 1$. Isto pode ser facilmente percebido quando comparamos as duas melhores características de cada k analisado. Para $k = 1$, o valor obtido foi $97,35 \pm 6,05$, enquanto para $k = 11$ o valor foi $94,72 \pm 15,41$, ou seja, uma diferença de 2,63 na média e 9,36 na variância, ambas a favor do valor de $k = 1$.

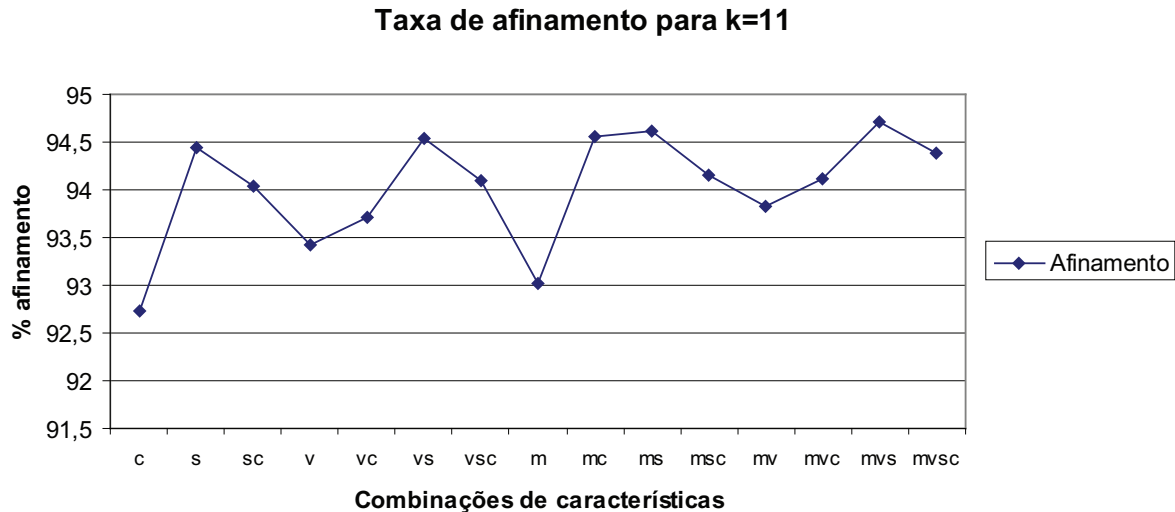


Figura 5.13: Taxa de afinamento da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 11$

A distribuição das taxas de ruído quando $k = 11$ (tabela 5.6 e a figura 5.14) é bastante parecida com as taxas ruído quando $k = 1$. A obliquidade novamente é a pior característica, com $0,47 \pm 0,43$. Entretanto, os valores obtidos para as outras combinações de características não são muito diferentes, sempre estando em torno de 0,4, mas nitidamente maiores que suas equivalentes quando $k = 1$.

Sabendo qual o melhor conjunto de características, é necessário encontrar o valor de k que obtenha o melhor resultado. Para tal, outra análise é realizada. Desta vez as imagens da base de letras são testadas para todos os valores de k definidos nesta abordagem, ou seja, aqueles contidos no conjunto $\{1,3,5,7,9,11,13\}$. Isto é processado usando sempre a mesma característica. Neste caso, como a característica que obteve o melhor resultado no teste anterior foi a combinação entre a média, variância e obliquidade (*mvs*), então esta será utilizada.

A tabela 5.7 mostra o resultado deste processamento. Analisando o gráfico da figura 5.15, o valor de k que obtém a melhor taxa de afinamento é 1, com $97,35 \pm 6,50$.

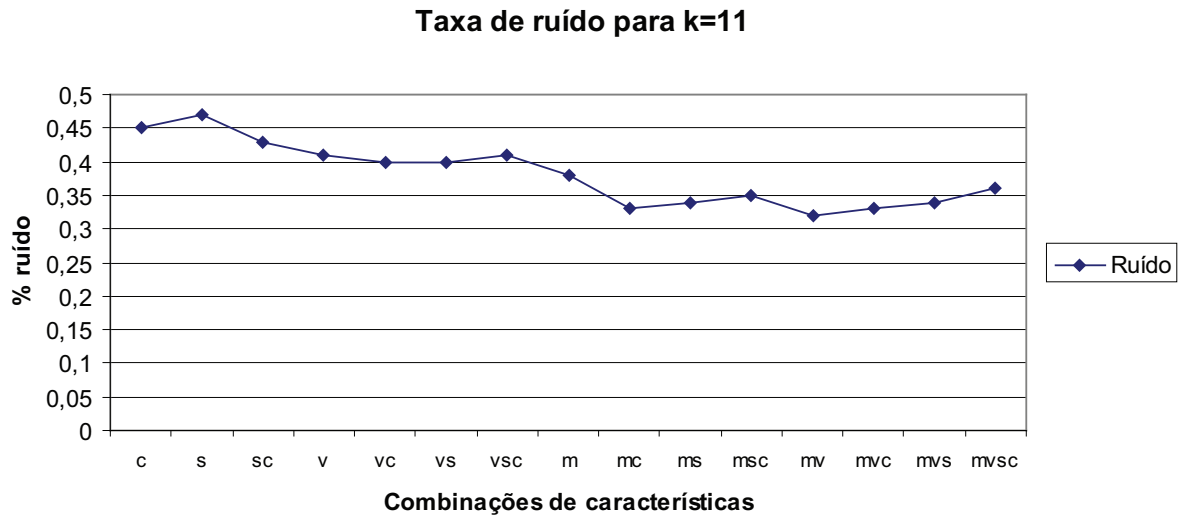


Figura 5.14: Taxa de ruído da base de letras com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 11$

Tabela 5.7: Melhor combinação de características de acordo com o valor de k

Melhor conjunto de características - mvs		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	97,35 \pm 06,50	0,05 \pm 0,11
3	94,65 \pm 13,66	0,17 \pm 0,32
5	93,05 \pm 16,27	0,22 \pm 0,36
7	93,67 \pm 16,43	0,27 \pm 0,38
9	93,87 \pm 16,46	0,31 \pm 0,37
11	94,72 \pm 15,41	0,34 \pm 0,38
13	94,81 \pm 15,39	0,37 \pm 0,38

Já a menor taxa de afinamento acontece quando $k = 5$, como o valor 93,05 \pm 16,27. O mesmo acontece com a taxa de ruído (gráfico da figura 5.16), onde o menor valor acontece quando $k = 1$. A medida que o k vai aumentando, a taxa de ruído também aumenta.

Com base em toda esta análise, pode-se afirmar que, com esta base de letras, as melhores taxas de afinamento e as menores taxas de ruído foram obtidas com $k = 1$, reforçando a tese dos autores (como [CE67]) que dizem que o melhor valor de k é 1.

5.1.3.1 Função distância

Para completar este estudo sobre a base de letras, a melhor combinação de características encontrada (*mvs*) e o valor de k que fornece o melhor resultado na classificação são usados para afinar as imagens da base processadas por diferentes funções distância.

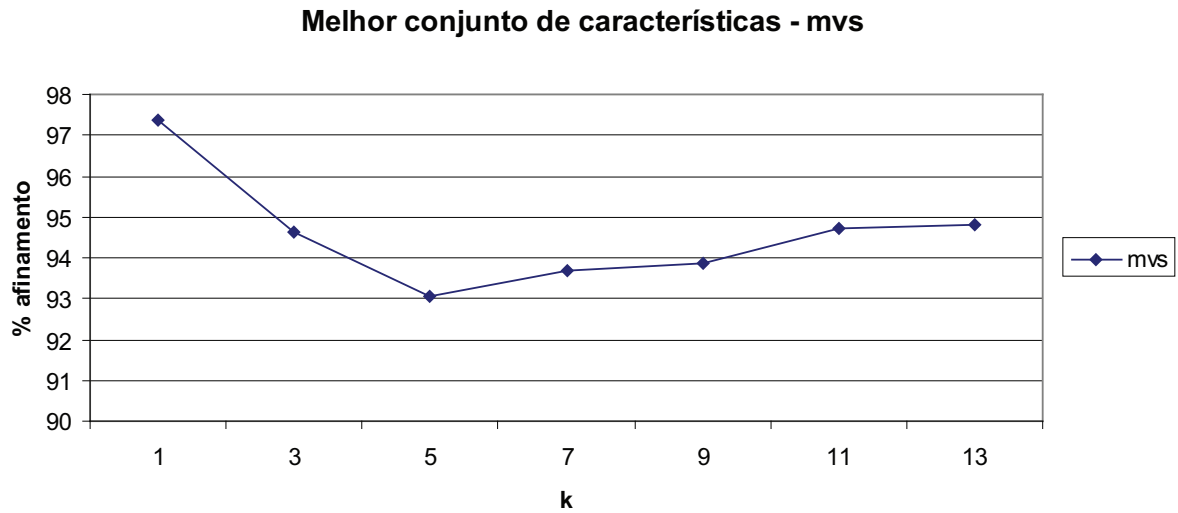


Figura 5.15: Taxa de afinamento para o melhor conjunto de características (*mvs*) de acordo com o valor de k

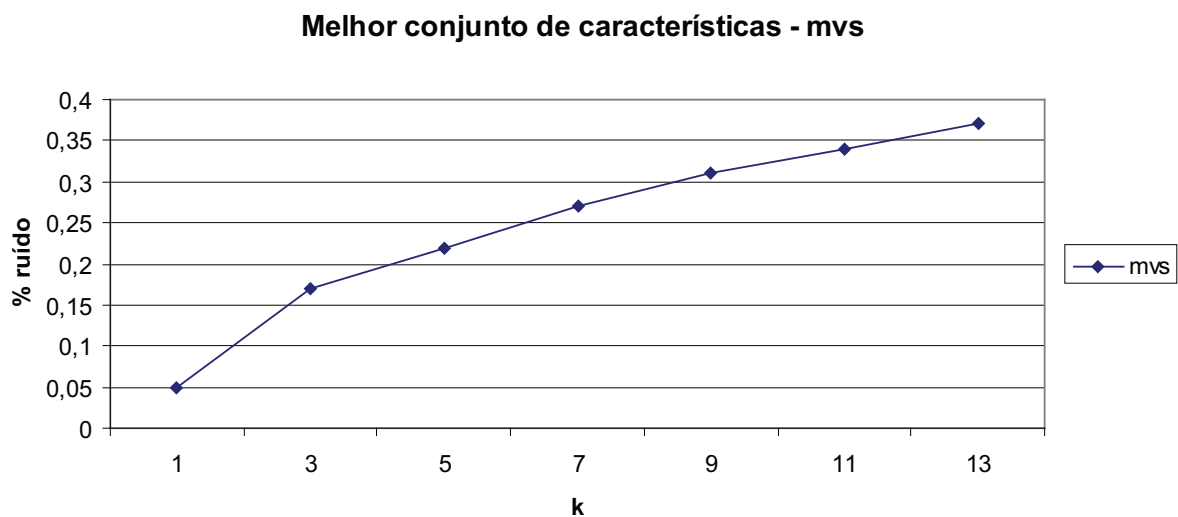


Figura 5.16: Taxa de ruído para o melhor conjunto de características (*mvs*) de acordo com o valor de k

A tabela 5.8 e o gráfico da figura 5.17 mostram que existem variações, dependendo da função distância utilizada. A abordagem Chamfer 5-7-11 mostrou-se superior às outras, com uma taxa de afinamento de $97,35 \pm 6,50$. Já a Morfologia Matemática, com o elemento estruturante *quadrado*, se mostrou a pior das cinco abordagens, com uma taxa de afinamento de $93,29 \pm 15,96$.

Já em relação à taxa de ruído (gráfico da figura 5.18), a abordagem que obteve

Tabela 5.8: Comparação entre as funções distância com as características mvs e $k = 1$

Comparação entre as funções distância		
Função distância	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
Chamfer 3-4	95,70 \pm 13,70	0,03 \pm 0,07
Chamfer 5-7-11	97,35 \pm 06,50	0,05 \pm 0,11
Chessboard	93,40 \pm 16,04	0,12 \pm 0,29
City-block	95,41 \pm 15,64	0,09 \pm 0,32
Morfologia - Quadrado	93,29 \pm 15,96	0,13 \pm 0,29
Morfologia - Cruz	95,44 \pm 15,64	0,09 \pm 0,32

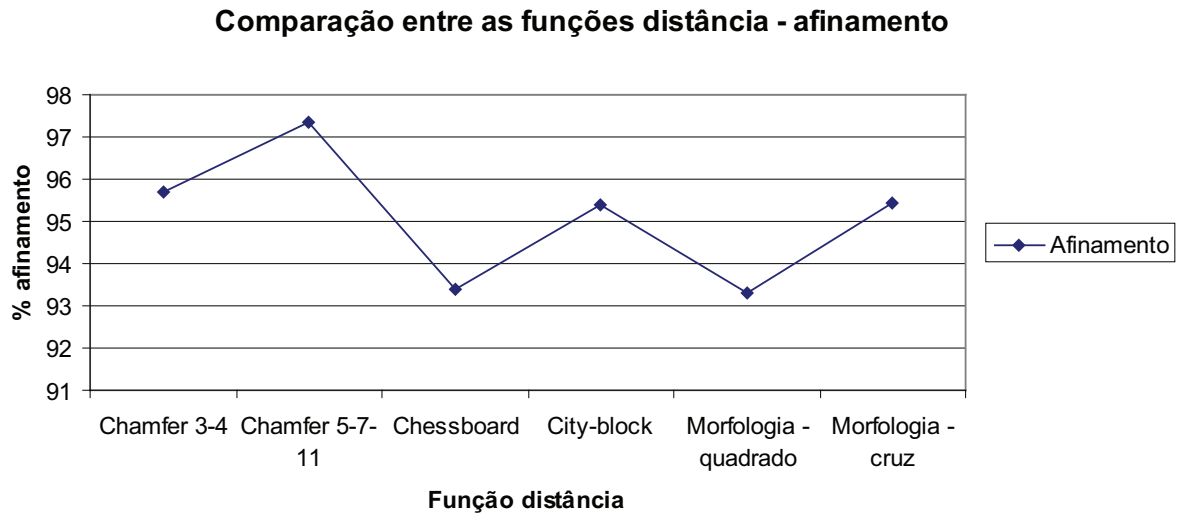


Figura 5.17: Taxa de afinamento para cada função distância onde a combinação de características é mvs (média, variância e obliquidade) e o $k = 1$

o melhor resultado foi Chamfer 3-4, com uma taxa de ruído de apenas $0,03 \pm 0,07$. Novamente, a abordagem com a pior taxa de ruído foi a Morfologia Matemática, com o elemento estruturante *quadrado*, com $0,13 \pm 0,29$.

Mesmo com uma taxa de ruído ligeiramente mais alta que o menor valor encontrado, a função distância que melhor caracteriza os pontos na base de letras é a abordagem Chamfer 5-7-11.

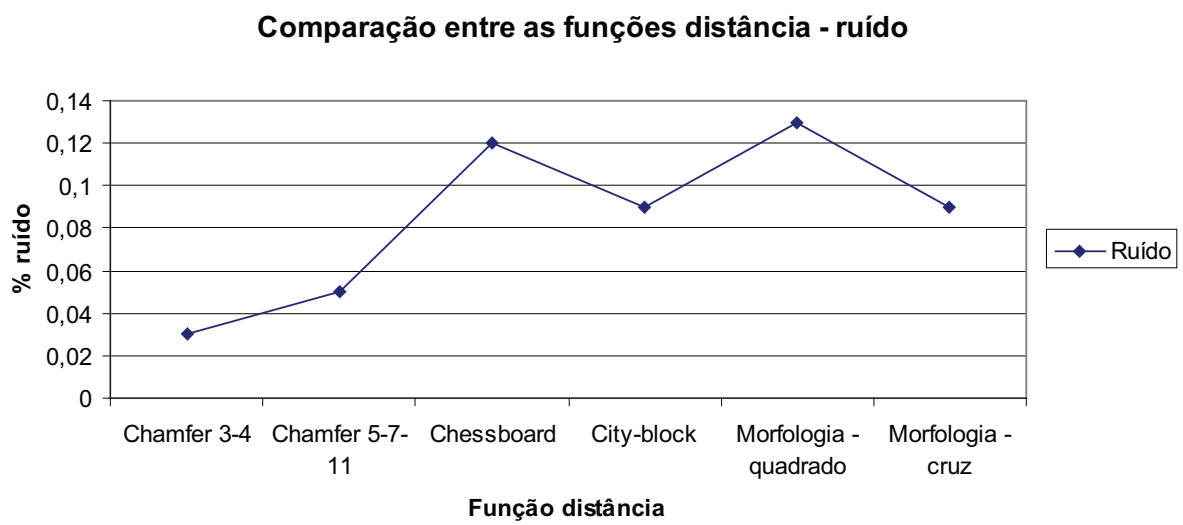


Figura 5.18: Taxa de ruído para cada função distância onde a combinação de características é mvs (média, variância e obliquidade) e o $k = 1$

5.1.3.2 Comparações com outros métodos

Afim de estabelecer uma comparação com outros métodos, uma imagem retirada da base de letras foi submetida ao afinamento por vários métodos. O resultado destes afinamentos pode ser visto na tabela 5.9.

Tabela 5.9: Comparação entre os métodos de afinamento analisados nesta abordagem

Comparação entre os métodos de afinamento		
Método	Afinamento (μ)	Ruído (μ)
Holt com staircase	82,53	0,17
Holt sem staircase	80,46	0,14
Mandeville	81,42	0,20
Morfológico	60,74	0,15
Stentiford	84,98	0,16
Zhang-Suen	80,93	0,14
Zhang-Wang	81,12	0,18
Aprendizagem	98,70	0,09

As figuras 5.5a e 5.5b mostram o par de imagens amostra e ideal submetidas à extração de características. A figura 5.6 representa a nova figura submetida ao método de aprendizagem e à todos os outros métodos de afinamento analisados nesta abordagem, para estabelecer uma comparação.

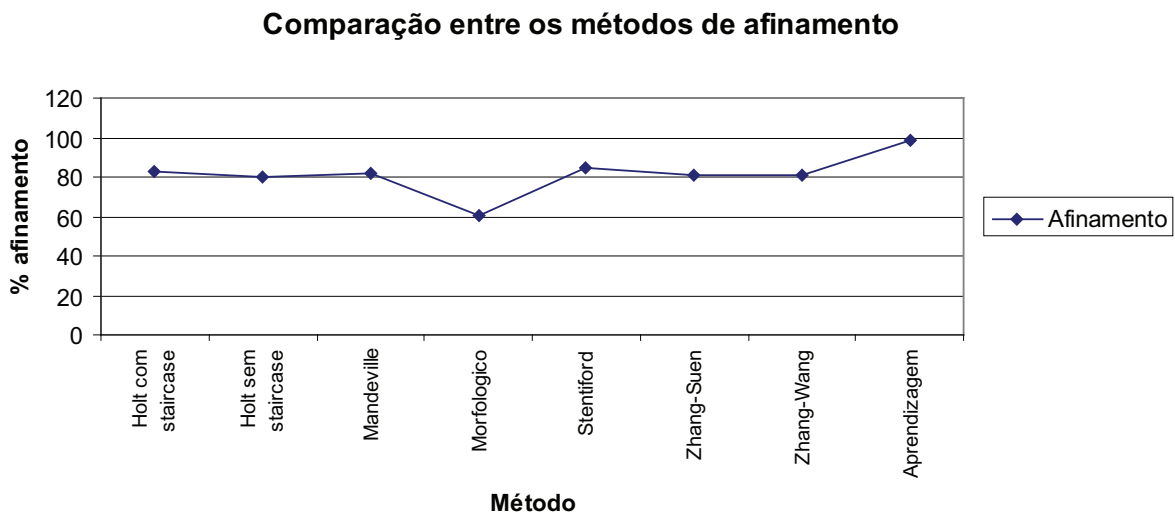


Figura 5.19: Comparação entre as taxas de afinamento dos métodos testados

A imagem 5.8 representa a imagem afinada pela metodologia de aprendizagem através de exemplos. É possível notar que, visualmente, não há diferenças em relação à imagem ideal (figura 5.7). Além disso, apresenta uma taxa de afinamento melhor que os

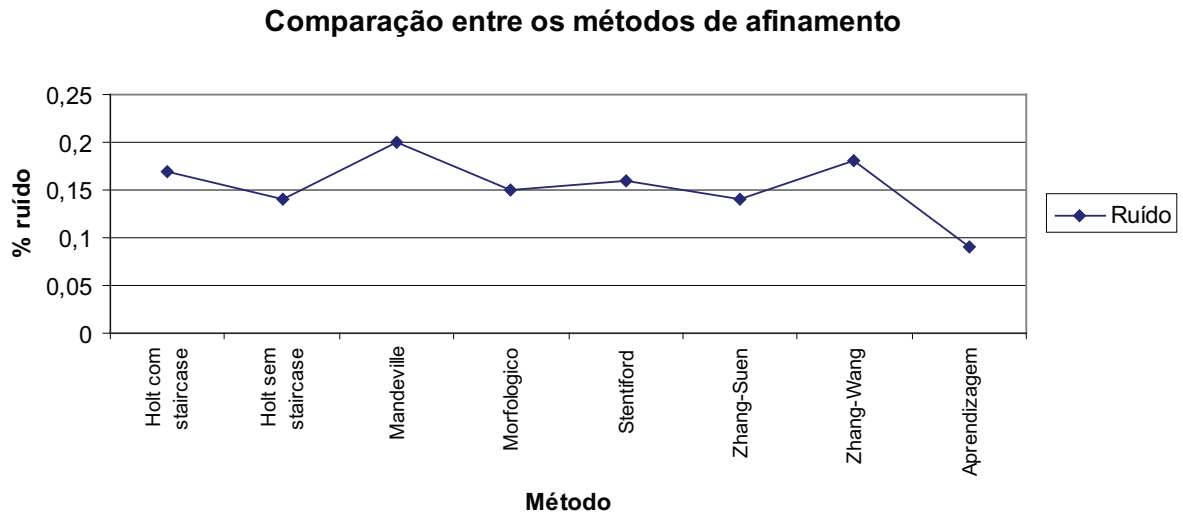


Figura 5.20: Comparação entre as taxas de ruído dos métodos testados

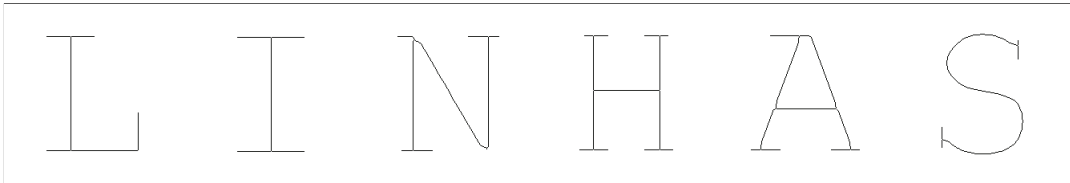


Figura 5.21: Afinamento pelo método de Holt com staircase removal
Taxa de afinamento e ruído: 82,53% e 0,17%

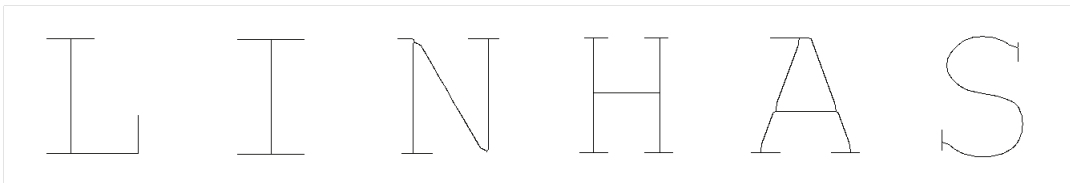


Figura 5.22: Afinamento pelo método de Holt sem staircase removal
Taxa de afinamento e ruído: 80,46% e 0,14%

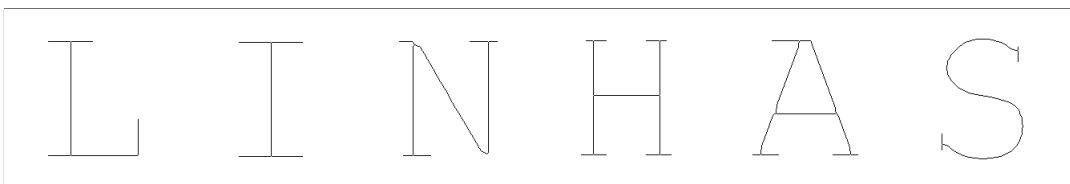


Figura 5.23: Afinamento pelo método de Mandeville
Taxa de afinamento e ruído: 81,42% e 0,20%

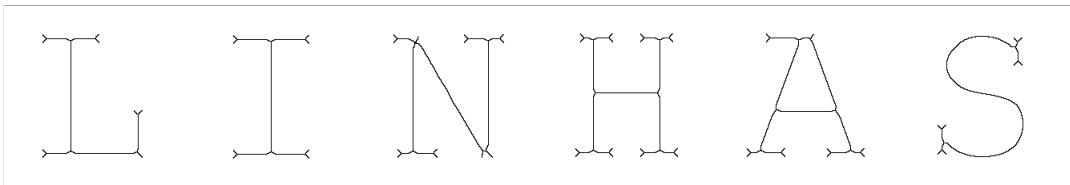


Figura 5.24: Afinamento pelo método Morfológico
Taxa de afinamento e ruído: 60,74% e 0,15%

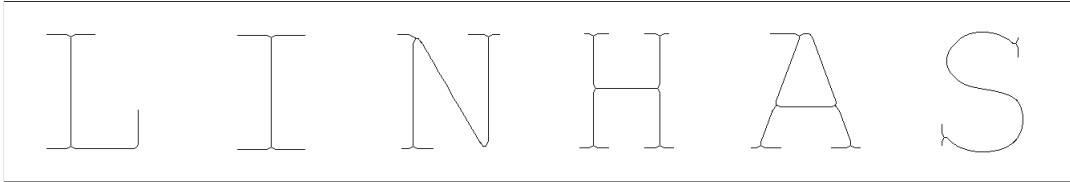


Figura 5.25: Afinamento pelo método de Stentiford
Taxa de afinamento e ruído: 84,98% e 0,16%

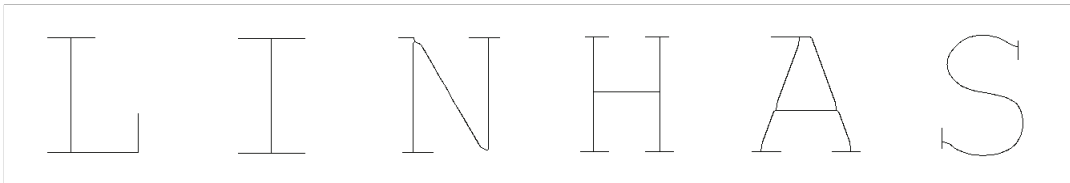


Figura 5.26: Afinamento pelo método de Zhang-Suen
Taxa de afinamento e ruído: 80,93% e 0,14%

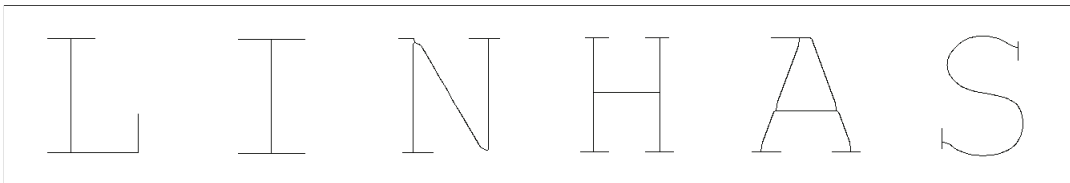


Figura 5.27: Afinamento pelo método de Zhang-Wang
Taxa de afinamento e ruído: 81,12% e 0,18%

outros métodos testados, chegando à 98,70% de acerto em relação à imagem ideal, como pode ser visto nos gráficos das imagens 5.19 e 5.20 (exemplos nas figuras 5.21 à 5.27). Os outros métodos ficam na faixa de 80% de acerto, com exceção do método Morfológico, com 60,74% de acerto.

Alguns dos problemas apresentados no capítulo 3 podem ser claramente vistos na figura 5.28, que apresenta um pedaço da letra *N* em detalhes, para melhor visualização. Todos os métodos, com exceção da Aprendizagem, perdem um pouco da simetria do objeto. Isto pode ser visto nas conexões entre duas linhas, que ao invés de terem a forma

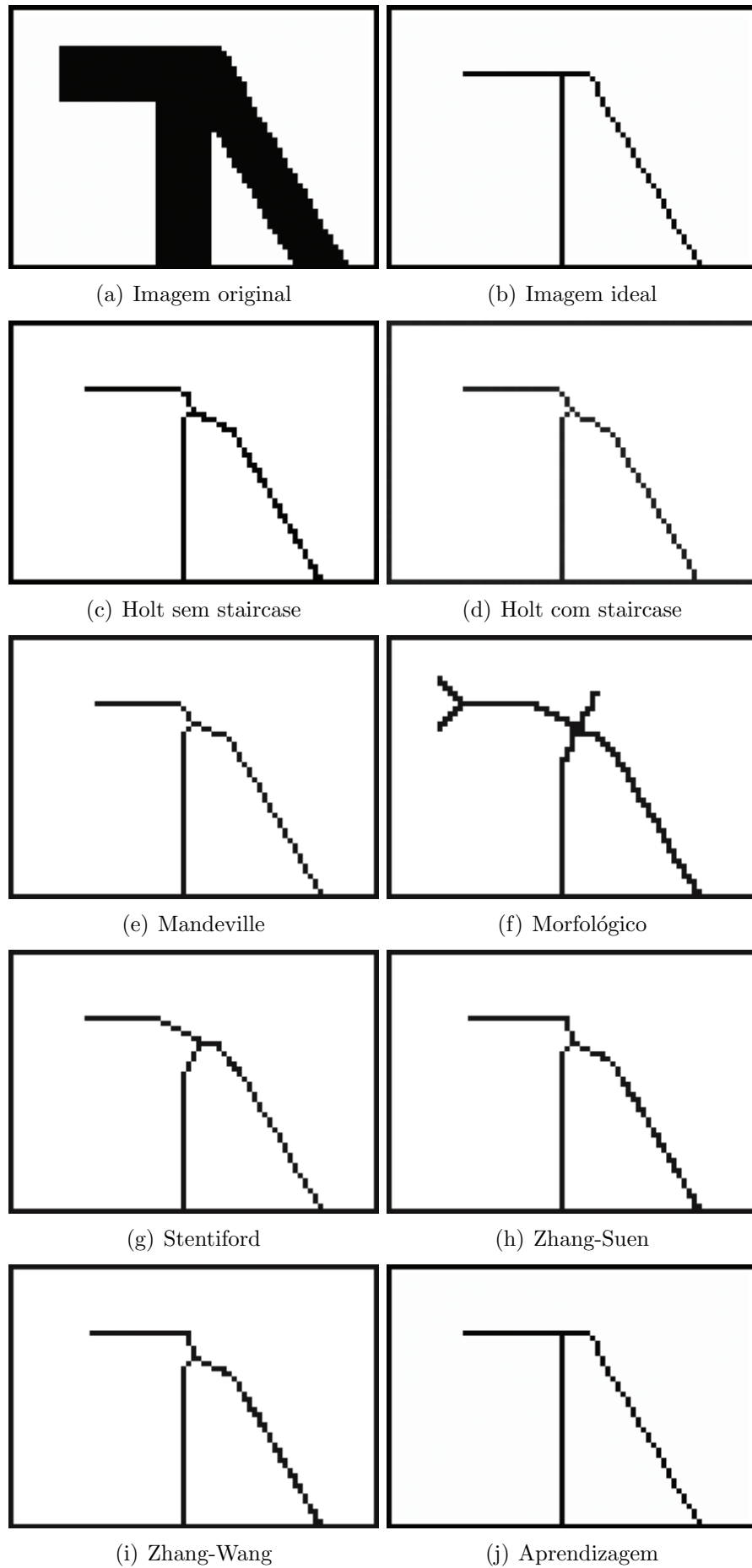


Figura 5.28: Detalhes da comparação entre os métodos testados

de um “T”, parecem mais um “Y” (figura 5.28(c), 5.28(d), 5.28(e), 5.28(g), 5.28(h) e 5.28(i)) ou um “X” (figura 5.28(f)). Adicionalmente, alguns métodos não respeitam o fato que um esqueleto deve conter apenas linhas com um pixel de espessura (figura 5.28(c), 5.28(f), 5.28(g), 5.28(h) e 5.28(i)).

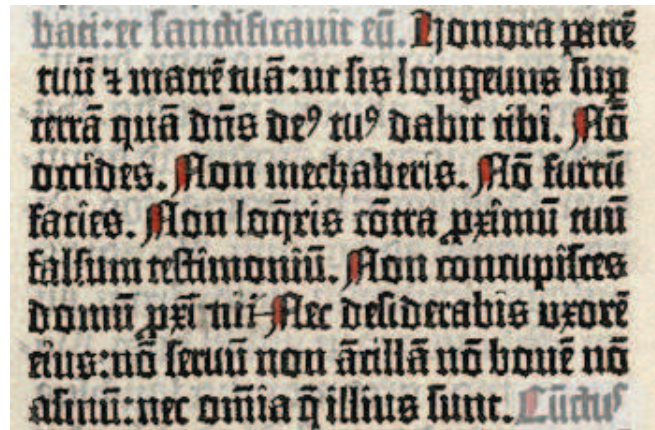
5.2 Base de imagens da Bíblia

A base de imagens utilizada nesta abordagem contém imagens digitais de passagens bíblicas, retiradas da Bíblia de Gutenberg [GUT03]. Como uma página inteira desta Bíblia contém muitos objetos além do texto (5.29), foram usados fragmentos destas páginas. Um exemplo pode ser visto na figura 5.30(a). Entretanto, a imagem que será passada para o cálculo da função distância e futura aprendizagem será a versão binarizada destas imagens, como pode ser visto na figura 5.30(b).



Figura 5.29: Exemplo de uma página da Bíblia de Gutenberg

Vários testes foram realizados com esta base, todas com o objetivo de afiná-las da forma mais parecida possível com as imagens consideradas ideais usadas no treinamento. Como exemplo, a imagem ideal da figura 5.30(b) pode ser vista na figura 5.30(c). Estes testes analisaram um conjunto de 10 imagens.



(a) Imagem original

Honora patrem
 tuum et matrem tuam: ut sis longeuus super
 terram quam dominus deus tuus dabit tibi. Non
 occides. Non mechaberis. Non furum
 facies. Non loquaris contra proximum tuum
 falsum testimonium. Non concupisces
 domum proximi tui. Nec desiderabis uxorem
 eius: ne seruum non ancillam non bouem non
 asinum: nec omnia que illius sunt.

(b) Imagem binarizada

Honora patrem
 tuum et matrem tuam: ut sis longeuus super
 terram quam dominus deus tuus dabit tibi. Non
 occides. Non mechaberis. Non furum
 facies. Non loquaris contra proximum tuum
 falsum testimonium. Non concupisces
 domum proximi tui. Nec desiderabis uxorem
 eius: ne seruum non ancillam non bouem non
 asinum: nec omnia que illius sunt.

(c) Imagem ideal

Figura 5.30: Exemplo de imagem a ser usada nesta abordagem

Non ora patre
 tuū ⁊ matrem tuā: ut sis longeuus sup
 terra quā dñs de⁹ tu⁹ dabit tibi. Non
 occides. Non mechaberis. Non furū
 facies. Non loq̄ris cōtra p̄mū nō
 falsum testimoniū. Non concupiscas
 domū p̄i nri ⁊ uxorem eius: nō seruū nō ancillā nō bouē nō
 asinū: nec om̄ia q̄ illius sunt.

(a) através de Chamfer 3-4

Non ora patre
 tuū ⁊ matrem tuā: ut sis longeuus sup
 terra quā dñs de⁹ tu⁹ dabit tibi. Non
 occides. Non mechaberis. Non furū
 facies. Non loq̄ris cōtra p̄mū nō
 falsum testimoniū. Non concupiscas
 domū p̄i nri ⁊ uxorem eius: nō seruū nō ancillā nō bouē nō
 asinū: nec om̄ia q̄ illius sunt.

(b) através de Chamfer 5-7-11

Figura 5.31: Função distância da figura 5.30(b) através de várias abordagens

Non ora paritē
 tuū ⁊ manū tuā: ut sis longeuus sup
 terra quā dñs de⁹ tu⁹ dabit tibi. Non
 occides. Non machaberis. Non furū
 facies. Non loq̄ris cōtra p̄ximū tuū
 falsum testimoniū. Non concupiscas
 domū p̄xi tui. Nec desiderabis uxore
 eius: nō seculū nō aīllā nō bouē nō
 asinū: nec om̄ia q̄ illius sunt.

(a) através de Chessboard

Non ora paritē
 tuū ⁊ manū tuā: ut sis longeuus sup
 terra quā dñs de⁹ tu⁹ dabit tibi. Non
 occides. Non machaberis. Non furū
 facies. Non loq̄ris cōtra p̄ximū tuū
 falsum testimoniū. Non concupiscas
 domū p̄xi tui. Nec desiderabis uxore
 eius: nō seculū nō aīllā nō bouē nō
 asinū: nec om̄ia q̄ illius sunt.

(b) através de City-block

Figura 5.32: Função distância da figura 5.30(b) através de várias abordagens

Honorata patre
 tuū ⁊ matrem tuā: ut sis longeuus sup
 terra quā dñs de⁹ tu⁹ dabit tibi. Non
 occides. Non mechaberis. Non furta
 facies. Non loq̄ris cōtra p̄mū tuū
 falsum testimoniū. Non concupiscas
 domū p̄m̄ tuū. Nec desiderabis uxore
 eius: nō seruū nō ancillā nō bouē nō
 asinū: nec om̄ia q̄ illius sunt.

(a) através da Morfologia (cruz)

Honorata patre
 tuū ⁊ matrem tuā: ut sis longeuus sup
 terra quā dñs de⁹ tu⁹ dabit tibi. Non
 occides. Non mechaberis. Non furta
 facies. Non loq̄ris cōtra p̄mū tuū
 falsum testimoniū. Non concupiscas
 domū p̄m̄ tuū. Nec desiderabis uxore
 eius: nō seruū nō ancillā nō bouē nō
 asinū: nec om̄ia q̄ illius sunt.

(b) através da Morfologia (quadrado)

Figura 5.33: Função distância da figura 5.30(b) através de várias abordagens

Novamente se faz necessário calcular a função distância de cada imagem binária para a extração das características. As imagens contendo a função distância da figura 5.30(b) podem ser vistas nas figuras 5.31(a), 5.31(b), 5.32(a), 5.32(b), 5.33(a) e 5.33(b).

Tomando como base a aprendizagem realizada com o par de imagens amostra/ideal representadas pelas figuras 5.30(b) e 5.30(c), por exemplo, alguns resultados são mostrados na figura 5.34.

Orantes aut nolite multum loqui: sicut et herici faciunt. parant enim in multiloquio suo reaudiantur. Nolite ergo affamiliari nis. Sicut enim patre vestre quid opus sit vobis: an equam peratis tum. Sic ergo vos orabit. Patre noster qui es in celis sanctificetur nome tuu: Aduentia regnu tuu: fiat voluntas tua sicut in celo et in terra: Panem nostru supsubstantiale da nobis hodie: Et dimitte nobis debita nostra. sicut et nos dimittimus debitoribus nostris. Et ne nos inducas in temptatione: sed libera nos a malo.

(a) Figura original limiarizada

Orantes aut nolite multum loqui: sicut et herici faciunt. parant enim in multiloquio suo reaudiantur. Nolite ergo affamiliari nis. Sicut enim patre vestre quid opus sit vobis: an equam peratis tum. Sic ergo vos orabit. Patre noster qui es in celis sanctificetur nome tuu: Aduentia regnu tuu: fiat voluntas tua sicut in celo et in terra: Panem nostru supsubstantiale da nobis hodie: Et dimitte nobis debita nostra. sicut et nos dimittimus debitoribus nostris. Et ne nos inducas in temptatione: sed libera nos a malo.

(b) Figura 5.34(a) afinada

Tunc oblatus tunc ei percussit manus eius in pectus et orare. Discipuli autem increpabant eum. Iesus vero ait eis. Quinque percussos et nolite eos prohibere ad intrare: talis est enim regnu celos. Et cum impulsisset eis manus: abiit inde.

(c) Figura original limiarizada

Tunc oblatus tunc ei percussit manus eius in pectus et orare. Discipuli autem increpabant eum. Iesus vero ait eis. Quinque percussos et nolite eos prohibere ad intrare: talis est enim regnu celos. Et cum impulsisset eis manus: abiit inde.

(d) Figura 5.34(c) afinada

Figura 5.34: Exemplo de imagens afinadas através da Aprendizagem

As imagens afinadas mostradas na figura 5.34 mostram que, com uma aprendizagem suficiente, o resultado apresentado pelo afinamento de novas imagens é promissor. Como a aprendizagem foi realizada com uma imagem complexa, não faltaram características para que as novas imagens fossem afinadas corretamente.

5.2.1 Avaliação dos resultados

Utilizando-se de uma base contendo 10 imagens da Bíblia de Gutenberg, foram repetidos os testes de avaliação usados para avaliar a base de letras, descritos na seção 5.1.2. O primeiro teste realizado (figuras 5.35 e 5.36, tabelas 5.10, 5.11, 5.12 e 5.13) mostra qual característica isolada obtém os melhores resultados, analisando-as juntamente com os valores de k previamente selecionados na seção 5.1.2, ou seja, novamente com k variando entre os valores do conjunto $\{1,3,5,7,9,11,13\}$.

Tabela 5.10: Resultados do afinamento das imagens para a média de acordo com o valor de k

Média		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	90,32 \pm 12,90	1,25 \pm 1,20
3	90,12 \pm 15,99	1,42 \pm 1,40
5	89,55 \pm 18,01	1,55 \pm 1,44
7	89,48 \pm 19,32	1,56 \pm 1,45
9	89,71 \pm 19,64	1,63 \pm 1,67
11	88,91 \pm 19,64	1,80 \pm 1,72
13	88,73 \pm 19,53	1,81 \pm 1,89

Tabela 5.11: Resultados do afinamento das imagens para a variância de acordo com o valor de k

Variância		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	88,78 \pm 12,60	1,36 \pm 1,54
3	88,90 \pm 17,34	1,45 \pm 1,64
5	88,92 \pm 19,85	1,49 \pm 1,86
7	88,56 \pm 19,12	1,58 \pm 1,88
9	88,95 \pm 19,32	1,69 \pm 1,89
11	88,99 \pm 19,13	1,81 \pm 1,80
13	89,30 \pm 19,41	1,89 \pm 1,91

A figura 5.35 mostra a taxa de afinamento para cada uma das características selecionadas, com a variação de k apresentada anteriormente. Pode-se notar uma diferença significativa no que se diz respeito aos resultados obtidos com cada valor de k . Assim como na base de letras, os melhores resultados para cada característica acontecem quando $k = 1$, exceto para a variância, que obtém seu melhor resultado quando $k = 13$.

Por outro lado, o ruído apresentado no afinamento da base de imagens (figura 5.36) se mostra crescente, à medida que o valor de k vai aumentando. Porém, não diferencia muito entre as características testadas.

Tabela 5.12: Resultados do afinamento das imagens para a obliquidade de acordo com o valor de k

Obliquidade		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	87,94 \pm 16,57	1,29 \pm 1,59
3	87,49 \pm 17,10	1,32 \pm 1,65
5	87,69 \pm 17,36	1,40 \pm 1,60
7	87,12 \pm 17,59	1,49 \pm 1,62
9	86,98 \pm 17,94	1,59 \pm 1,63
11	86,32 \pm 18,13	1,75 \pm 1,63
13	86,12 \pm 18,23	1,81 \pm 1,63

Tabela 5.13: Resultados do afinamento das imagens para a curtose de acordo com o valor de k

Curtose		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	89,25 \pm 17,24	1,31 \pm 1,63
3	88,65 \pm 18,01	1,33 \pm 1,78
5	88,64 \pm 18,10	1,42 \pm 1,90
7	88,21 \pm 18,12	1,56 \pm 1,91
9	88,02 \pm 18,78	1,78 \pm 1,94
11	87,85 \pm 18,91	1,79 \pm 1,94
13	87,59 \pm 18,91	1,85 \pm 1,95

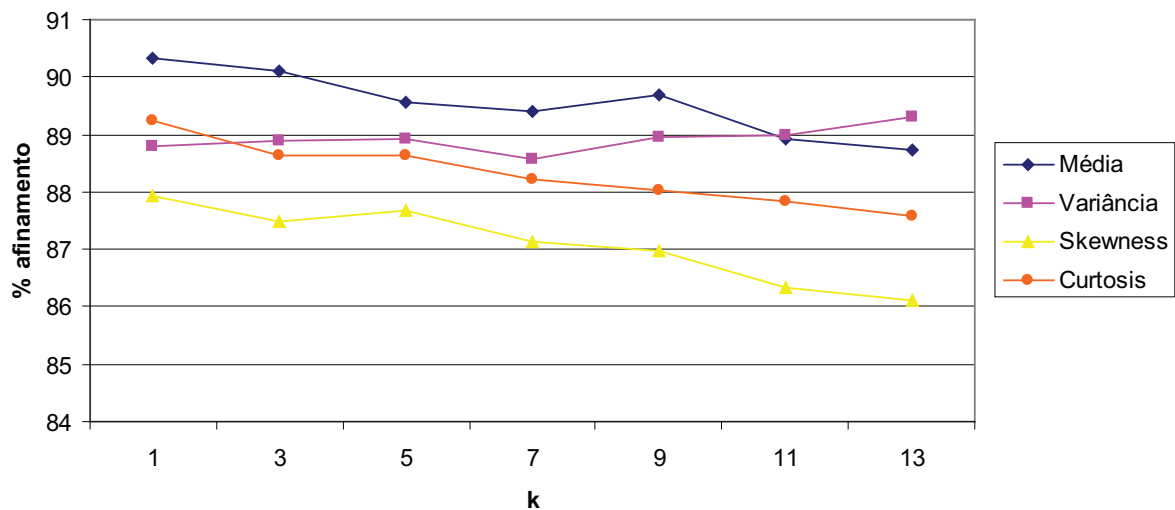


Figura 5.35: Afinamento da base imagens para cada característica isolada e o valor de k vizinhos mais próximos

Para determinar qual combinação de características é a mais eficiente, o processo de afinamento foi realizado na base de imagens utilizando cada uma das 15 combinações

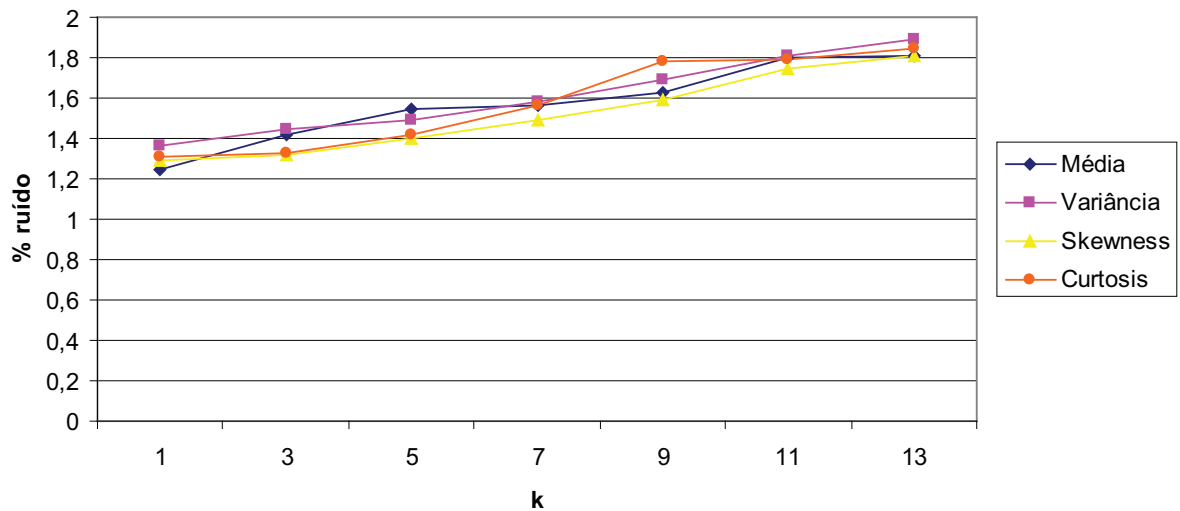


Figura 5.36: Ruído da base de imagens cada característica isolada e o valor de k vizinhos mais próximos

possíveis. O resultado pode ser visto na tabela 5.14.

Tabela 5.14: Taxas de afinamento e ruído para as 15 combinações possíveis das quatro características utilizadas nesta abordagem, para o valor $k = 1$

Combinações de características para $k = 1$		
Característica	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
c	$89,96 \pm 18,76$	$1,83 \pm 1,57$
s	$88,25 \pm 19,47$	$1,78 \pm 1,66$
sc	$89,29 \pm 17,38$	$1,62 \pm 1,60$
v	$87,45 \pm 15,20$	$1,78 \pm 1,54$
vc	$87,96 \pm 15,08$	$1,58 \pm 1,84$
vs	$86,63 \pm 14,15$	$1,65 \pm 1,62$
vsc	$87,46 \pm 15,55$	$1,52 \pm 1,86$
m	$90,12 \pm 13,90$	$1,45 \pm 1,52$
mc	$92,25 \pm 12,24$	$1,32 \pm 1,69$
ms	$85,26 \pm 15,42$	$1,53 \pm 1,91$
mcs	$86,13 \pm 15,87$	$1,63 \pm 1,68$
mv	$87,98 \pm 17,65$	$1,69 \pm 1,50$
mvc	$85,32 \pm 16,38$	$1,81 \pm 1,47$
mvs	$86,69 \pm 17,05$	$1,55 \pm 1,71$
$mvsc$	$87,01 \pm 16,29$	$1,61 \pm 1,97$

De acordo com a tabela 5.14 e a figura 5.37, a melhor taxa de afinamento acontece com as características média e curtose combinadas (mc), com a taxa de afinamento sendo $92,25 \pm 12,24$. Já a característica que apresenta a pior taxa de afinamento neste caso é a combinação média e obliquidade ($85,26 \pm 15,42$).

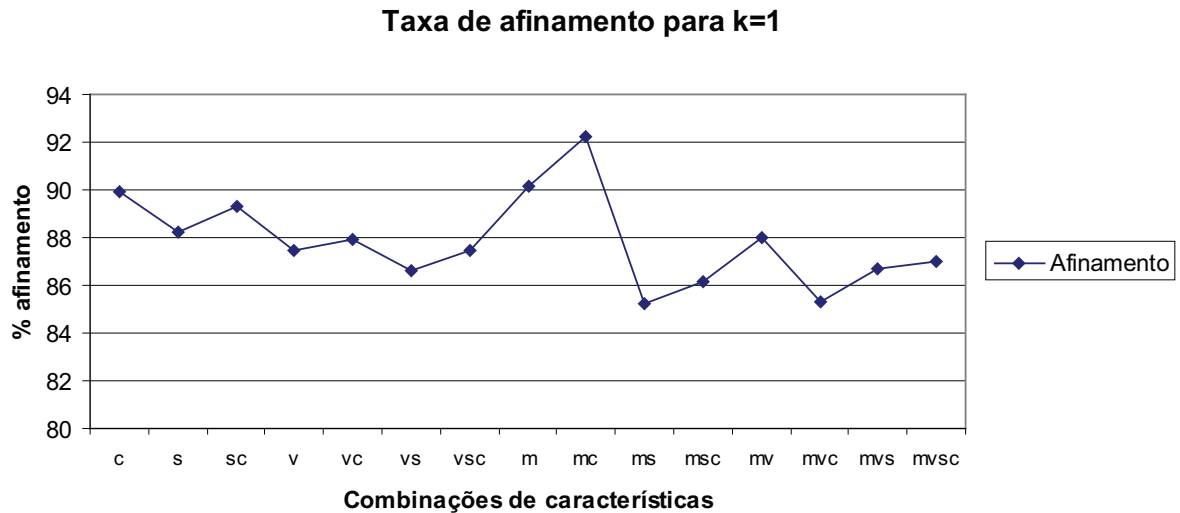


Figura 5.37: Taxa de afinamento da base de imagens com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$

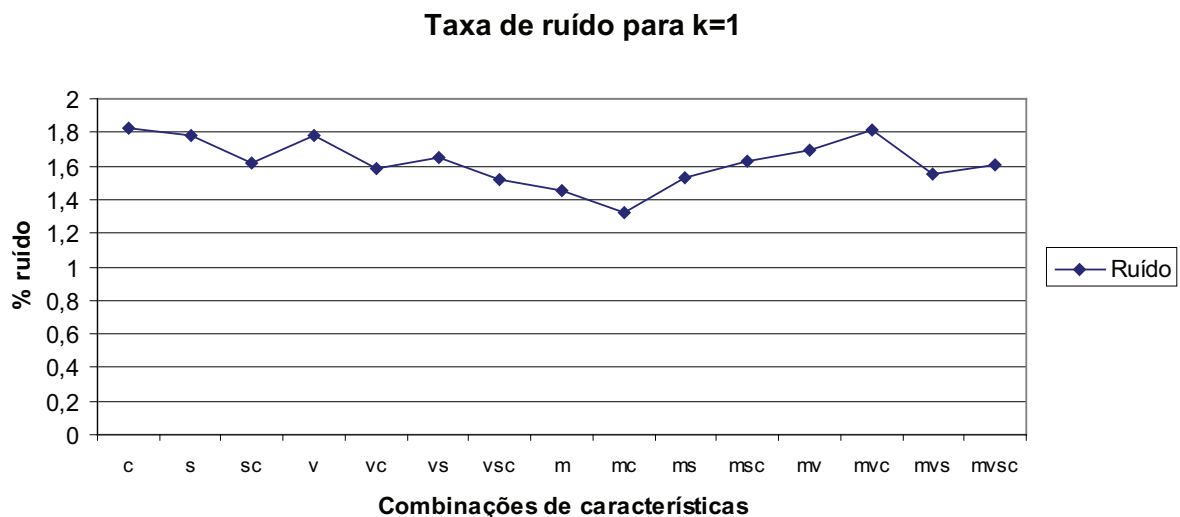


Figura 5.38: Taxa de ruído da base de imagens com todas as combinações das quatro características usadas nesta abordagem, para um valor de $k = 1$

Em relação ao ruído (tabela 5.14 e a figura 5.38), a situação é bastante parecida. Novamente a combinação entre a média e a obliquidade se apresenta como a pior alternativa, com uma taxa de ruído de $1,53 \pm 1,91$. Em contrapartida, a combinação entre a média e a curtose apresenta a mais baixa taxa de ruído ($1,32 \pm 1,69$), mostrando ser a combinação de características mais forte para esta base de imagens.

Sabendo que a média e a curtose representam o melhor conjunto de características

desta base de imagens, é necessário encontrar o valor de k que obtenha o melhor aproveitamento desta combinação.

Tabela 5.15: Melhor combinação de características de acordo com o valor de k

Melhor conjunto de características - mc		
k -NN	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
1	92,65 \pm 12,24	1,32 \pm 1,69
3	91,32 \pm 13,26	1,42 \pm 1,72
5	91,02 \pm 14,57	1,45 \pm 1,76
7	90,83 \pm 14,99	1,51 \pm 1,78
9	90,12 \pm 15,36	1,52 \pm 1,87
11	89,60 \pm 15,81	1,53 \pm 1,88
13	88,90 \pm 16,09	1,55 \pm 1,88

A tabela 5.15 mostra o resultado desta análise. Como pode ser visto no gráfico da figura 5.39, o valor de k que obtém a melhor taxa de afinamento é 1, com 92,65 \pm 12,24. Já a menor taxa de afinamento acontece quando $k = 13$, como o valor 88,90 \pm 16,09. O mesmo acontece com a taxa de ruído (gráfico da figura 5.40), onde o menor valor acontece quando $k = 1$. A medida que o k vai aumentando, a taxa de ruído também aumenta.

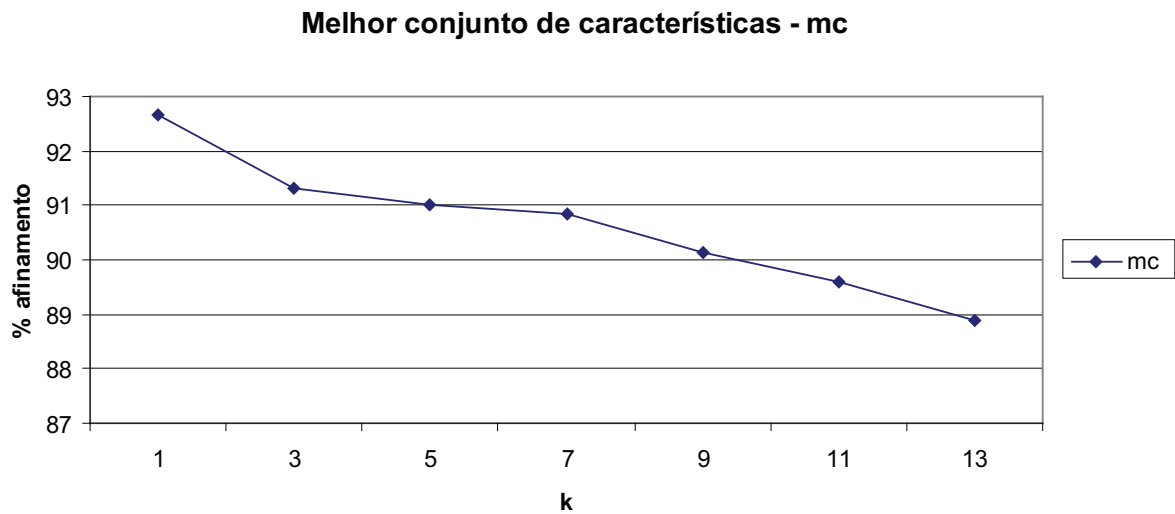


Figura 5.39: Taxa de afinamento para o melhor conjunto de características (mc) de acordo com o valor de k

As melhores taxas de afinamento e as menores taxas de ruído para esta base de imagens foram obtidas com $k = 1$, novamente reforçando a tese dos autores (como [CE67]) que afirmam que o valor de k que obtém os melhores resultados é 1.

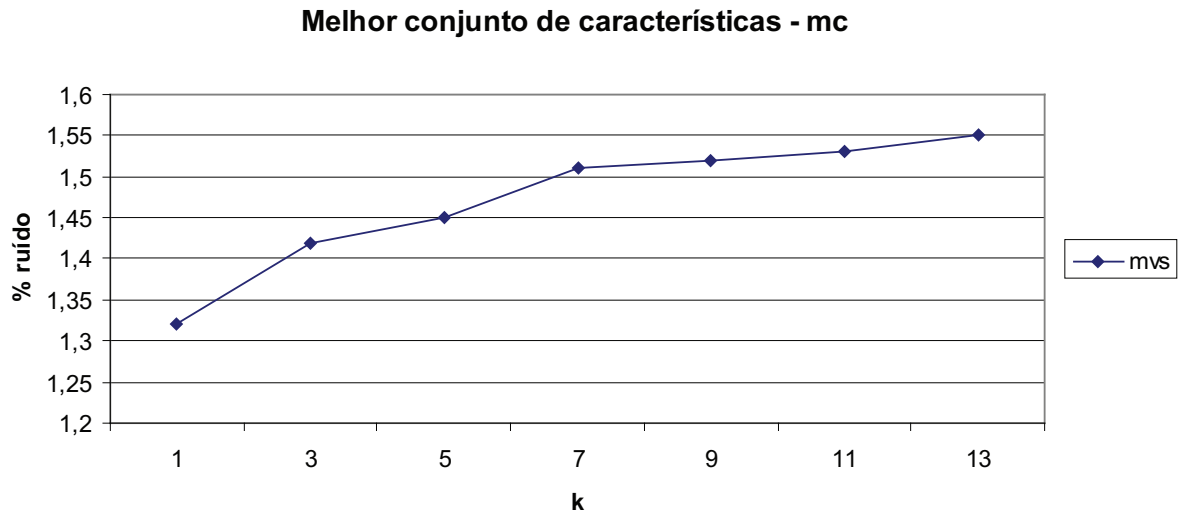


Figura 5.40: Taxa de ruído para o melhor conjunto de características (*mc*) de acordo com o valor de *k*

5.2.1.1 Função distância

Assim como os testes realizados com a base de letras, a melhor combinação de características encontrada (*mc*) e o valor de *k* que fornece o melhor resultado na classificação são usados para afinar novas imagens da base, geradas por diferentes funções distância.

Tabela 5.16: Comparação entre as funções distância com as características *mc* e $k = 1$

Comparação entre as funções distância		
Função distância	Afinamento ($\mu \pm \sigma$)	Ruído ($\mu \pm \sigma$)
Chamfer 3-4	91,98 \pm 13,21	1,49 \pm 1,97
Chamfer 5-7-11	92,65 \pm 12,24	1,32 \pm 1,69
Chessboard	90,56 \pm 14,14	1,56 \pm 1,82
City-block	89,12 \pm 17,33	1,65 \pm 1,78
Morfologia - Quadrado	89,65 \pm 16,92	1,51 \pm 1,72
Morfologia - Cruz	88,98 \pm 15,34	1,60 \pm 1,82

A tabela 5.16 e o gráfico da figura 5.41 mostram que são grandes as variações, dependendo da função distância utilizada. Assim como na análise da base de letras, a abordagem Chamfer 5-7-11 mostrou-se superior as outras, com uma taxa de afinamento de 92,65 \pm 12,24. Já a Morfologia Matemática, com o elemento estruturante *cruz*, se mostrou a pior das cinco abordagens, com uma taxa de afinamento de 88,98 \pm 15,34. Em relação à taxa de ruído apresentada (gráfico da figura 5.42), a abordagem que obteve o melhor resultado foi Chamfer 5-7-11, com uma taxa de ruído de 1,32 \pm 1,69. A abordagem com a pior taxa de ruído foi a City-block, com 1,65 \pm 1,78.

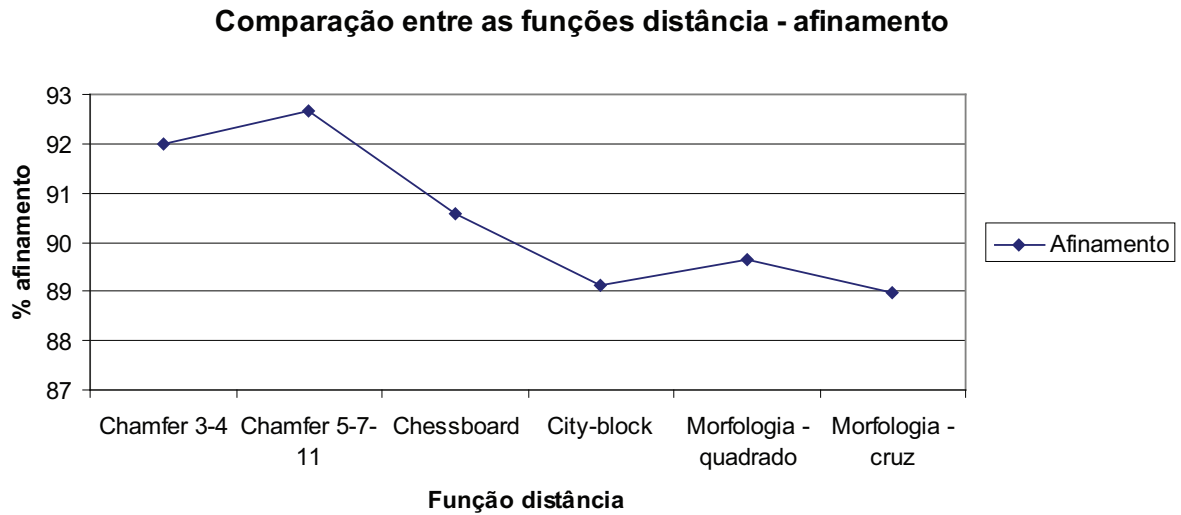


Figura 5.41: Taxa de afinamento para cada função distância onde a combinação de características é *mc* (média e curtose) e o $k = 1$

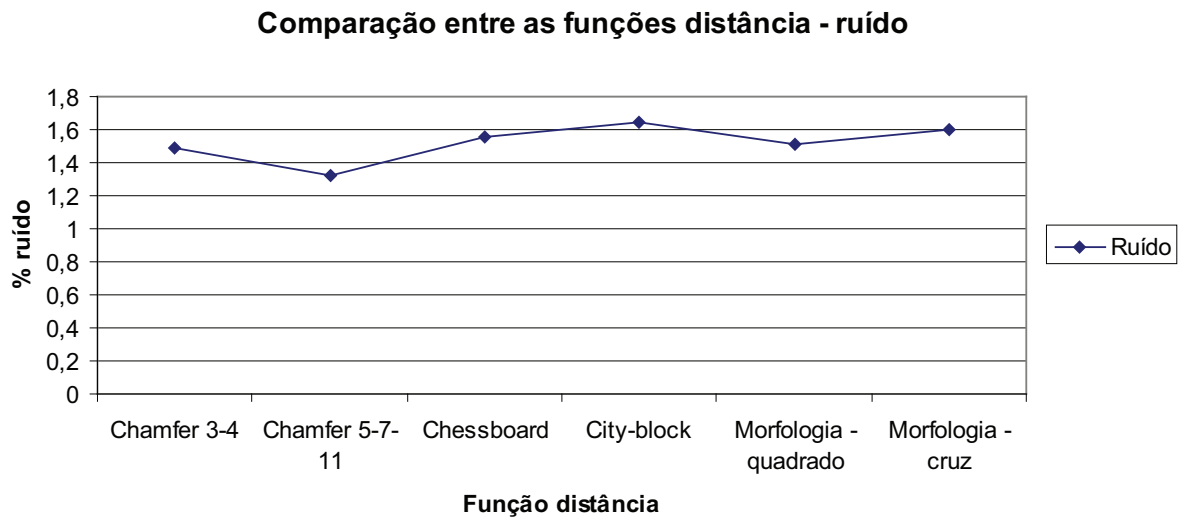


Figura 5.42: Taxa de ruído para cada função distância onde a combinação de características é *mc* (média e curtose) e o $k = 1$

Novamente é correto afirmar que, para a base de imagens analisada, a função distância que a melhor caracteriza é a abordagem Chamfer 5-7-11.

5.2.1.2 Comparações com outros métodos

Afim de estabelecer uma comparação com outros métodos de afinamento, uma imagem baseada na base de imagens da Bíblia de Gutenberg foi afinada pelos mesmos métodos aplicados à base de letras. O resultado destes afinamentos pode ser visto na tabela 5.17.

Tabela 5.17: Comparação entre os métodos de afinamento analisados nesta abordagem

Comparação entre os métodos de afinamento		
Método	Afinamento (μ)	Ruído (μ)
Holt com staircase	81, 27	3, 21
Holt sem staircase	80, 10	2, 95
Mandeville	69, 65	1, 54
Morfológico	79, 24	5, 24
Stentiford	72, 56	1, 15
Zhang-Suen	82, 36	2, 98
Zhang-Wang	84, 21	3, 12
Aprendizagem	90, 25	1, 98

As figuras 5.30(b) e 5.30(c) mostram o par de imagens amostra e ideal submetidas à extração de características. A figura 5.31(b) mostra a função distância gerada a partir da figura 5.30(b), pelo método Chamfer 5-7-11. A figura 5.45(a) é submetida ao método de aprendizagem e à todos os outros métodos de afinamento anteriormente descritos, afim de estabelecer uma comparação.

Pode ser notado que, em imagens mais complexas, o resultado de cada método de afinamento estudado varia bastante, sendo que os métodos de Mandeville, Morfológico e Stentiford (figuras 5.45(e), 5.45(f) e 5.45(g)) são incapazes de reproduzir grande parte dos objetos na imagem.

5.3 Comentários finais

Neste capítulo foram apresentados os testes e os resultados da abordagem proposta de Afinamento por Aprendizagem. Além disso, foi realizada uma comparação com outros métodos de afinamento, que mostrou que a abordagem proposta é bastante promissora.

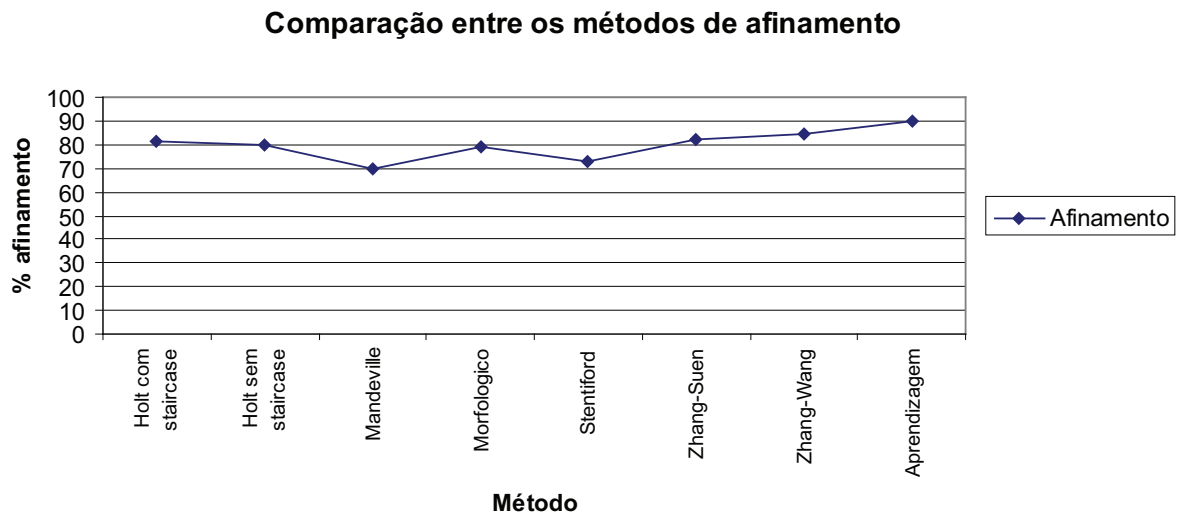


Figura 5.43: Comparação entre as taxas de afinamento dos métodos testados

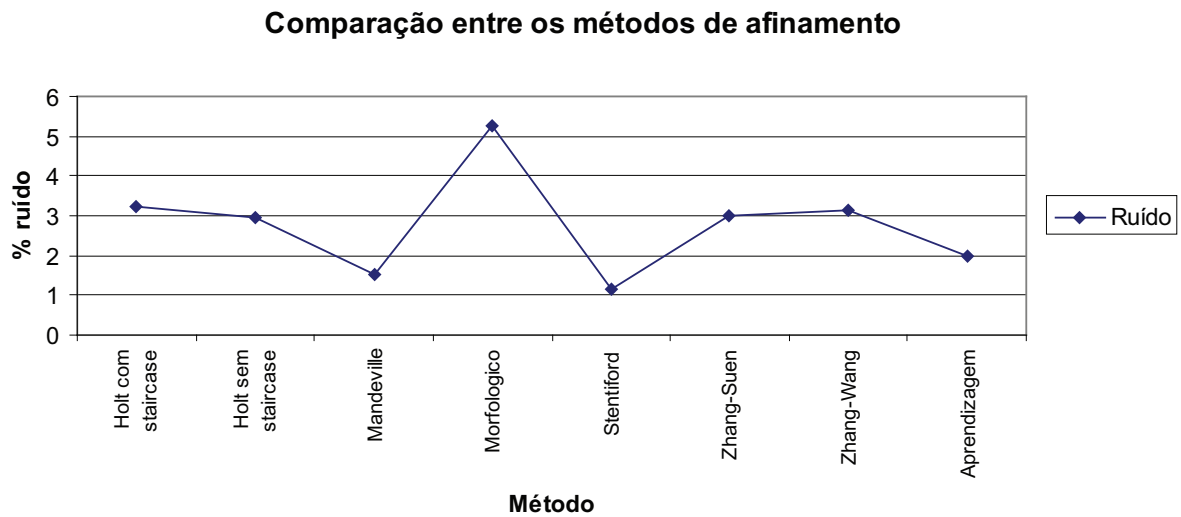


Figura 5.44: Comparação entre as taxas de ruído dos métodos testados

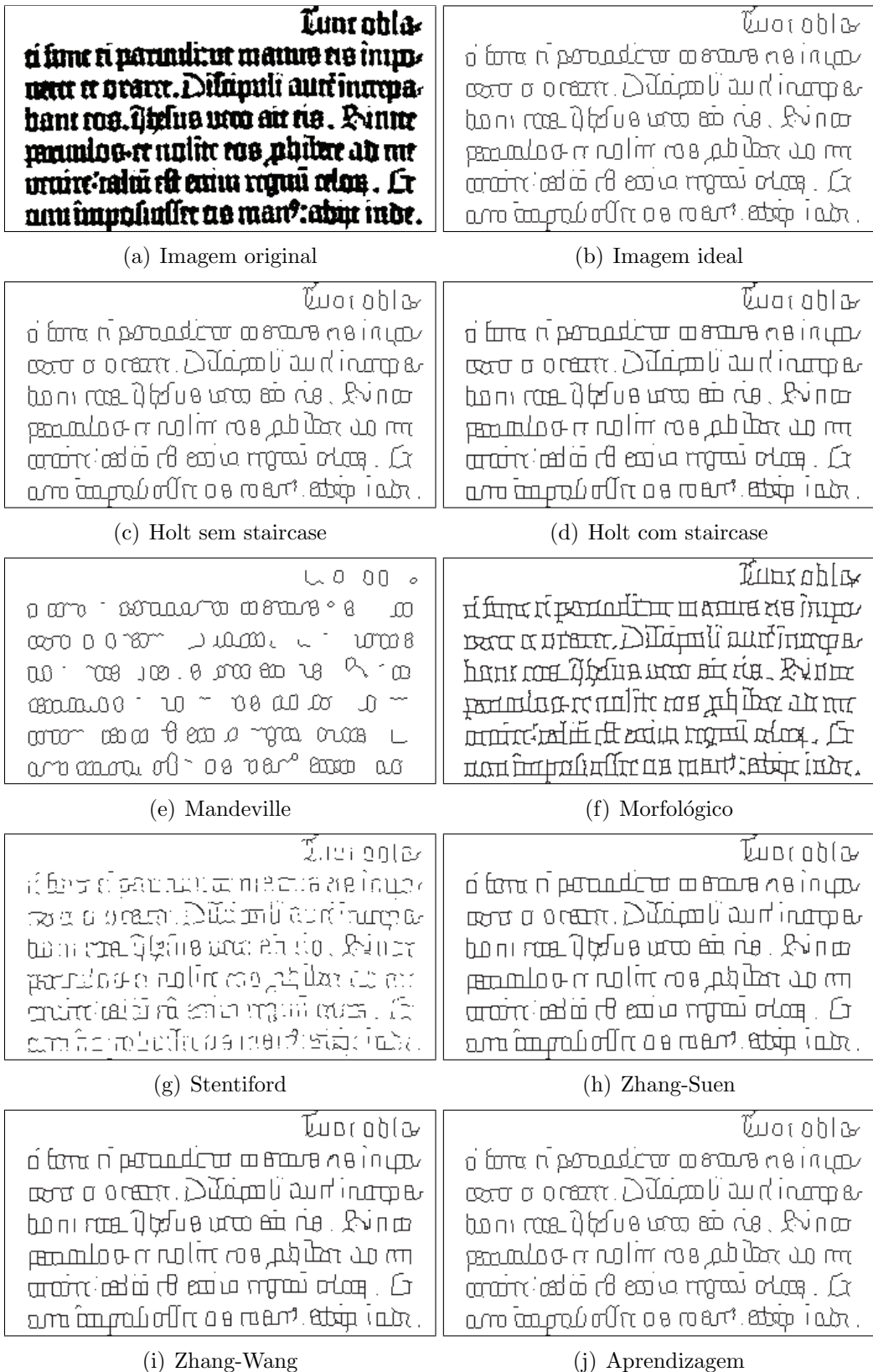


Figura 5.45: Exemplo de imagens afinadas através dos métodos analisados nesta abor-
dagem

Capítulo 6

Conclusões

Este trabalho apresentou uma nova abordagem automática de afinamento de imagens binárias empregando aprendizagem. O processo de aprendizagem consistiu em mapear matematicamente todos os pixels e suas respectivas vizinhanças na mesma posição em duas imagens, a primeira sendo a imagem binária original e a segunda sendo a versão ideal afinada. Um vetor de características para cada pixel em janelas de tamanho variável totalmente adaptativo foi criado. O fato do tamanho da janela ser variável foi proposto para eliminar possíveis redundâncias e conflitos na classificação.

O processo de aprendizagem foi aplicado a partir da função distância calculada da imagem binária original. A função distância atribui aos pixels pretos de um objeto de uma imagem binária a menor distância entre estes pixels e os pixels de fundo. Várias formas de calcular a função distância foram estudadas e usadas. A imagem contendo a função distância é uma imagem em níveis de cinza. O conjunto de características proposto foi o Nível de Cinza do pixel, a Média, a Variância, a Obliquidade, a Curtose, o Número de Conectividade, e os Valores Máximos e Mínimos. Com esses dados, uma matriz de classificação foi gerada, contemplando cada pixel da imagem original.

O processo de afinamento de novas imagens binárias fez-se reutilizando o conhecimento adquirido durante o processo de aprendizagem armazenado na matriz de classificação. A solução de cada pixel fez-se por auxílio do classificador supervisionado k -NN.

Duas bases de imagens foram usadas para avaliar a eficiência dessa nova abordagem automática de afinamento, uma primeira composta das 26 letras do alfabeto e de suas versões ideais afinadas geradas por um especialista da área. E uma segunda proveniente de imagens binarizadas da Primeira Bíblia impressa por Gutenberg, composta de 1282 páginas [GUT03].

Primeiramente, testes exaustivos foram realizados a partir da base composta das 26 letras do alfabeto testando:

- as diversas combinações possíveis de características dentro do conjunto proposto (Nível de cinza do pixel, Média, Variância, Obliquidade, Curtose, Número de Conectividade, Valores Máximos e Mínimos);
- as diversas formas de calcular a função distância estudadas 3.1 (Chamfer 3-4, Chamfer 5-7-11, City-block, Chessboard e Morfologia Matemática)
- as variações do valor de k no k -NN entre os valores do conjunto $\{1,3,5,7,9,11,13\}$.

Estes testes permitiram definir a função distância, a combinação de características e o k que apresentaram os melhores resultados. Esta variedade de combinações demonstrou através dos resultados obtidos a eficiência da abordagem proposta. De acordo com as tabelas 5.5, 5.7 e as figuras 5.11 e 5.15, destacou-se como a melhor combinação de características a combinação composta pela *média*, *variância* e *obliquidade* (*mvs*) e destacou-se $k = 1$ como sendo o parâmetro mais adequado do classificador supervisionado k -NN. A avaliação da menor taxa de ruído (gráfico da figura 5.16) também destacou $k = 1$ como sendo o parâmetro mais adequado do k -NN.

Os testes realizados em algumas imagens binarizadas da base da Primeira Bíblia impressa por Gutenberg demonstraram novamente que o valor de k sendo 1 e o cálculo da função distância sendo Chamfer 5-7-11 resulta nos melhores resultados. Em relação às características, neste caso a *média* e a *curtose* trouxeram os melhores resultados.

Não foi possível avaliar em tempo hábil numericamente os critérios inerentes ao processo de afinamento, que são:

- gerar o esqueleto de um pixel de largura;
- prevenir a erosão excessiva, que pode causar perda de conectividade;
- preservar a topologia dos objetos;
- gerar um esqueleto homotópico;
- minimizar rebarbas;
- não remover pontos finais;

Entretanto, uma avaliação visual permitiu perceber que a abordagem proposta de Afinamento através de Aprendizagem se mostrou eficiente (conforme mostrado no capítulo 5), gerando esqueletos de um pixel de largura, não erodindo excessivamente e, principalmente, mantendo a topologia do objeto afinado, resultando em uma imagem com maior fidelidade à imagem original.

As principais vantagens da abordagem proposta são:

- apenas duas imagens amostra/ideal foram usadas no processo de aprendizagem, o que demonstra a sua simplicidade;
- nenhuma heurística foi introduzida. E nem o tamanho da janela que, em outras abordagens precisa ser definido, foi fixado. Visto que a definição do tamanho das janelas é automática;
- o conhecimento de um especialista da área que apareceu na geração das imagens ground-truth pode ser reproduzido de forma totalmente automática a partir do ponto em que o conhecimento foi adquirido.

As desvantagens/limitações da abordagem proposta são:

- dificuldade em generalizar o melhor conjunto de características;
- o tempo de processamento é variável;
- a qualidade do resultado da função distância influencia no afinamento.

Trabalhos futuros podem consistir em:

- buscar outros conjuntos e outras combinações de características como, por exemplo, os momentos geométricos;
- aplicar outras transformadas de distância que permitem uma melhor aproximação da Distância Euclidiana;
- estudar e propor maneiras numéricas de avaliar o respeito aos critérios inerentes ao processo de afinamento;
- estudar a troca do k -NN por uma metodologia mais rápida, como o *SVM (Support Vector Machines)*;
- estudar e implementar outros métodos para calcular a função distância;
- avaliar a influência da isotropia dos objetos no afinamento;
- avaliar o comportamento do método em imagens com número par de pixels em suas linhas.

Por fim, é importante informar que o equipamento usado para realização dos testes apresentados neste documento consistem em um notebook HP Pavilion DV8333, com Ubuntu 6 instalado. Assim sendo, a reprodução destes testes em outros tipos de máquinas podem resultar em desempenhos diferentes.

Índice Remissivo

- Afinamento, 56
 - algoritmos, 12
 - paralelos, 11
 - seqüenciais, 11
 - aprendizagem, 59
 - através de exemplos, 48
 - classificação, 10
 - comparação, 75, 93
 - histórico, 6
 - problemática, 1, 3
- Algoritmos
 - Holt, 20, 30, 32
 - Staircase removal, 22
 - Morfológico, 27, 30, 32
 - Petrosino, 23, 30, 32
 - Stentiford, 13, 30, 32
 - Zhang-Suen, 14, 30, 32
 - Zhang-Wang, 17, 30, 32
- Aprendizagem, 40, 48, 52, 56
- Características
 - avaliação, 63
 - curtose, 51, 66, 87
 - extração, 50
 - média, 50, 65, 86
 - nível de cinza, 50
 - obliquidade, 51, 65, 87
 - variância, 50, 65, 86
- Classificador, 45
 - 1-NN, 45, 68, 88
 - k -NN, 46, 69, 71, 90
- Função distância, 33, 64
 - avaliação, 59, 71, 73, 80, 91
 - Chamfer, 34, 36
 - Chessboard, 34, 36
 - City-block, 34, 36
 - definição, 33
 - funcionamento, 34
 - Morfológica, 38
- Imagem
 - binária, 7
 - níveis de cinza, 7
- Janelas, 17, 50, 53
- Pixel
 - conectividade, 9, 15, 18, 21, 51
 - vizinhança, 7, 17
 - 4-vizinhança, 8
 - 8-vizinhança, 8
- Resultados
 - avaliação, 63, 65
- Vizinhança, 18

Referências Bibliográficas

- [AYA04] H. A. L. AYALA. *Uma abordagem de segmentação de imagens por aprendizagem*. PhD thesis, Pontifícia Universidade Católica do Paraná, 2004.
- [BER82] D. P. BERTSEKAS. *Constrained optimization and lagrange multiplier methods*. Academic Press, 1982.
- [BFLL80] B. G. BUCHANAN, E. A. FEIGENBAUM, J. LEDERBERG, and R. K. LINDSAY. *Applications of artificial intelligence for organic chemistry*. McGraw-Hill, New York, 1980.
- [BGV92] B. E. BOSER, I. GUYON, and V. N. VAPNIK. A training algorithm for optimal margin classifiers. *Computational Learning Theory*, pages 144–152, 1992.
- [BLU67] H. BLUM. A transformation for extracting new descriptors of shape. *In Wathen-Dunn. W., ed.. Models for the Perception of Speech and Visual Form*. M.I.T. Press, pages 362–380, 1967.
- [BOR86] G. BORGEFORS. Distance transformations in digital images. 34(3):344–371, Junho 1986.
- [BS79] M. BAZARA and C. M. SHETTY. *Nonlinear programming*. John Wiley and Sons, Inc, USA, 1979.
- [BSW⁺76] B. G. BUCHANAN, D. H. SMITH, W. C. WHITE, R. GRITTER, E. A. FEIGENBAUM, J. LEDERBERG, and C. DJERASSI. Applications of artificial intelligence for chemical inference, xxii: Automatic rule formation in mass spectrometry by means of the meta-dendral program. *Journal of the American Chemical Society*, 98:6168, 1976.
- [CE67] T. M. COVER and HART P. E. Nearest neighbor pattern classification. 13(1):21–27, Março 1967.

- [CM86] A. CORNUÉJOLS and L. MICLET. *Apprentissage Artificiel: Méthodes et Algorithmes*, volume 34. Computer Vision, Graphics and Image Processing, Junho 1986.
- [COT90] G. W. COTTRELL. Extracting features from faces using compression networks: Face, identity, emotion and gender recognition using holons. *Connection Models: Proceedings of the 1990 Summer School*, 1990.
- [DM92] B. DOLSAK and S. MUGGLETON. The application of inductive logic programming to finite element mesh design. 1992.
- [DOU92] E. R. DOUGHERTY. *An Introduction to Morphological Image Processing*, volume TT9. SPIE Press, 1992.
- [DSG93] K. A. DEJONG, W. M. SPEARS, and D. F. GORDON. Using genetic algorithms for concept learning. *Machine Learning*, 13:161–188, 1993.
- [FAC96] Jacques FACON. *Morfologia Matemática – Teoria e exemplos*. Jacques FACON, Curitiba-Brasil, 1996.
- [FAC02] Jacques FACON. *Processamento e Análise de Imagens*. Jacques FACON, Pontifícia Universidade Católica do Paraná – Curso de Mestrado em Informática Aplicada, Fevereiro 2002.
- [FLE87] R. FLETCHER. *Practical Methods of Optimization*. Wiley, 1987.
- [GBV93] I. GUYON, B. BOSER, and V. VAPNIK. *Automatic capacity tuning of very large VC-dimension classifiers*, volume 5 of *Advances in Neural Information Processing Systems*. Morgan Kaufmann, San Mateo, CA, 1993.
- [GUT03] J. GUTENBERG. Gutenberg bible, 2003.
- [HIR00] N. S. T. HIRATA. Projeto automático de operadores: Explorando conhecimentos a priori. *Tese de Doutorado, IME/USP, São Paulo-Brasil*, 2000.
- [HSCP97] C. M. HOLT, A. STEWART, M. CLINT, and R. H. PERROTT. A improved parallel thinning algorithm. *Algorithms for Image Processing and Computer Vision*, 30(2):156–160, Fevereiro 1997.
- [KIM97a] H. Y. KIM. Quick construction of efficient morphological operators by computational learning. 33(4):286–287, Fevereiro 1997.

- [KIM97b] Y. KIM, H. *Construção Automática de Operadores Morfológicos por Aprendizagem Computacional*. PhD thesis, Universidade de São Paulo, 1997.
- [KLV98] S. R. KULKARNI, G. LUGOSI, and S. S. VENJATESH. Learning pattern classification – a survey. *IEEE Transactions on Information Theory*, 44(6):2178–2206, Outubro 1998.
- [LBD⁺89] Y. LECUN, B. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. HUBBARD, and L. D. JACKEL. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 1989.
- [LHY⁺89] K. F. LEE, H. W. HON, HWANG M. Y., S. MAHAJAN, and R. REDDY. The sphinx speech recognition system. *1989 International Conference on Acoustics, Speech, and Signal Processing (ICASSP '89)*, 1:445–448, Maio 1989.
- [LLS92] L. LAM, S. W. LEE, and C. Y. SUEN. Thinning methodologies - a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869–885, Setembro 1992.
- [LUE73] D. G. LUENBERGER. *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, USA, 1973.
- [LUE86] D. G. LUENBERGER. *Linear and Nonlinear Programming*. Addison-Wesley, USA, 1986.
- [LWH90] K. J. LANG, A. H. WAIBEL, and G. E. HINTON. A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3:34–43, 1990.
- [MBPL99] A MANZANERA, T. M. BERNARD, F. J. PRÊTEUX, and B. LONGUET. Ultra-fast skeleton based on isotropic fully parallel algorithm. *Proceedings of the 8th International Conference on Discrete Geometry for Computer Imagery*, pages 313–324, Março 1999.
- [MER09] J MERCER. Functions of positive and negative types and their connection with the theory of integral equations. *Transactions of the London Philosophical Society*, 209:415–446, 1909.
- [MF75] B. MOAYER and K. S. FU. A syntactic approach to fingerprint pattern recognition. *Pattern Recognition*, 7:1–23, 1975.

- [MIT97] T. M. MITCHELL. *Machine Learning*, volume 1. McGraw-Hill Science/Engineering/Math, Março 1997.
- [MJ77] J. L. MUNDY and R. E. JOYNSON. Automatic visual inspection using syntatic analisys. *In Proceedings International Conference on Pattern Recognition and Image Processing*, pages 144–147, 1977.
- [MJCJB] D. C. MARTINS-Jr, R. M. CESAR-Jr, and J. BARRERA. Redução de dimensionalidade utilizando entropia condicional média: aplicações em filtragem de imagens e em reconhecimento de texturas.
- [NS86] T. V. NGUYEN and J. SKLANSKY. A fast skeleton-finder for coronary arteries. *In Proceedings 8th International Conference on Pattern Recognition (Paris, France)*, pages 481–483, 1986.
- [PB04] R. O. PLOTZE and O. M. BRUNO. Estudo e comparação de algoritmos de esqueletonização para imagens binárias. *IV Congresso Brasileiro de Computação – Computação Gráfica*, pages 59–64, 2004.
- [PD79] K. PRESTON and M. J. B. DUFF. Basies of cellular logic with some applications in medical image processing. *Proceeding of the IEEE*, 67(5):826–857, 1979.
- [POM89] D. A. POMERLEAU. Alvin: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, 1:305–318, 1989.
- [POM93] D. A. POMERLEAU. Knowledge-based training of artificial neural networks for autonomous robot driving. *Robot Learning*, pages 19–43, 1993.
- [PR99] T. POGGIO and SHELTON. C. R. Machine learning, machine vision, and the brain. *Artificial Intelligence Magazine*, 20:37–55, Setembro 1999.
- [PRE61] K. PRESTON. The cellscan system – a leucocyte pattern analyzer. *In Proceedings West Join Computer Conference Los Angeles CA*, 19:173–183, Maio 1961.
- [PS00] A. PETROSINO and G. SALVI. A two-subcycle thinning algorithm and its parallel implementation on simd machines. *IEEE Transactions on Image Processing*, 9(2):277–283, Fevereiro 2000.
- [RD76] A. ROSENFELD and L. S. DAVIS. A note on thinning. *IEEE Transactions on Systems, Man, and Cybernetics*, 3, Março 1976.

- [REE93] R REED. Pruning algorithms – a survey. *IEEE Transactions on Neural Networks*, 4:740–747, 1993.
- [RP66] A. ROSENFELD and J. L. PFALTZ. Sequential operations in a digital picture processing. *Journal of ACM*, 13(4):471–494, Outubro 1966.
- [RWL94] D. RUMELHART, B. Widrow, and M. LEHR. The basic ideas in neural networks. *Communications of the ACM*, 37(3):87–92, 1994.
- [SAG94] H SAGAN. Space-filling curves. 1994.
- [SHE68] D. SHEPARD. A two-dimensional interpolation function for irregularly spaced data. *Proceedings of the 23rd National Conference of the ACM*, pages 517–523, 1968.
- [SM83] F. W. M. STENTIFORD and R. G. MORTIMER. Some new heuristics for thinning binary handprinted characters for ocr. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:81–84, 1983.
- [SMB⁺99] B. SCHOLKOPF, S. MIKA, C. J. C. BURGES, P. KNIRSCH, K. R. MULLER, G. RATSCH, and A. J. SMOLA. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10:1000–1017, 1999.
- [SMKS94] A. SRINIVASAN, S. MUGGLETON, R. D. KING, and M. J. E. STEMBERG. Mutagenesis: Ilp experiments in a non-determinate biological domain. *Proceedings of the Fourth Inductive Logic Programming Workshop*, 1994.
- [SR71] R. STEFANELLI and A. ROSENFELD. Some parallel thinning algorithms for digital images. *Journal of ACM*, 18(2):225–264, Abril 1971.
- [TES94] G. TESAURO. Td-gammon, a self-teaching backgammon program achieves master-level play. *Neural Computation*, 6:215–219, 1994.
- [VAL84] L. G. VALIANT. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, Novembro 1984.
- [VAP98] V. N. VAPNIK. *Statistical Learning Theory*. John Wiley & Sons, 1998.
- [YD88] Q. Z. YE and P. E. DANIELSSON. Inspection of printed circuit boards by connectivity preserving shirinking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):737–742, Setembro 1988.

- [YGR⁺95] J. YOO, A. GRAY, J. RODEN, U. FAYYAD, R. DE CARVALHO, and S. DJORGOVSKI. Analysis of digital poss-ii catalogs using hierarchical unsupervised learning algorithms. *Astronomical Data Analysis Software and Systems V*, 101:41, 1995.
- [ZS84] Y. Y. ZHANG and C. Y. SUEN. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, Março 1984.
- [ZW96] Y. Y. ZHANG and P. S. P. WANG. A parallel thinning algorithm with two sub-iterations that generates one-pixel-wide skeleton. *13th International Conference on Pattern Recognition (ICPR'96)*, 4:457–461, 1996.