

Leoncio Claro de Barros Neto

**MODELAGEM EM GEOMETRIA DIGITAL
APRIMORADA POR TÉCNICAS
ADAPTATIVAS DE SEGMENTOS DE
RETAS**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Engenharia Elé-
trica.

São Paulo
2011

Leoncio Claro de Barros Neto

**MODELAGEM EM GEOMETRIA DIGITAL
APRIMORADA POR TÉCNICAS
ADAPTATIVAS DE SEGMENTOS DE
RETAS**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Engenharia Elé-
trica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Antônio Marcos de
Aguirra Massola

São Paulo
2011

Dedico este trabalho ao meu pai (*in memoriam*) e à minha mãe, como uma singela homenagem por seus esforços, dedicação e amor para minha formação.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Antonio Marcos de Aguirra Massola, pelo apoio, pela confiança, pelos conselhos e opiniões ao longo do desenvolvimento desta pesquisa.

Ao Prof. Dr. André Riyuiti Hirakawa por ter acompanhado as diversas fases do desenvolvimento deste trabalho, o qual não seria possível sem a sua ajuda.

Ao Prof. Dr. João José Neto, cujas idéias e entusiasmo foram fundamentais para a concepção e aperfeiçoamento desta pesquisa.

Aos Prof. Dr. Antonio Mauro Saraiva e Prof. Dr. Carlos Eduardo Cugnasca do Laboratório de Automação Agrícola.

Ao Prof. Dr. Ivan Ramos Pagnossin pelo apoio técnico relativo ao compilador LaTeX.

Ao Prof. Dr. Reginaldo Arakaki pelo incentivo em determinadas etapas da pesquisa.

À equipe da Biblioteca do Instituto de Matemática e Estatística da USP, pelo apoio em diversos levantamentos na literatura técnica.

À minha família, pela compreensão e apoio dado durante esta aventura na área do conhecimento.

Aos servidores e técnicos das secretarias da POLI, do PCS e da Biblioteca da POLI pelo atendimento às diversas solicitações.

A todas as pessoas que direta ou indiretamente contribuíram para a realização deste trabalho, em particular aos amigos Mario Carraza, Ramona Straube, Nelson André de Barros Gama do Nascimento, Marcelina Hollanda De Requena, Gláucio Diré e aos colegas do Laboratório de Automação Agrícola.

- O todo é igual à soma de suas partes.

EUCLIDES

- O todo é maior do que a soma de suas partes.

MAX WERTHEIMER

- To see the World in a Grain of Sand
And a Heaven in a Wild Flower
Hold Infinity in the palm of your hand
And Eternity in a hour.

WILLIAN BLAKE

Resumo

Visando representar linhas retas digitais, segmentos digitalizados e arcos, cada uma das linhas de pesquisa disponíveis apresenta suas vantagens e aplicações apropriadas. No entanto, considerando as complexidades de cenários do mundo real, o uso dessas representações não é tão popular em situações que requerem modelos flexíveis ou envolvendo interferências espúrias. As tecnologias adaptativas são formalismos da ciência da computação capazes de alterar seu comportamento dinamicamente, sem a interferência de agentes externos, em resposta a estímulos de entrada. Ao serem capazes de responder às mencionadas condições variáveis do ambiente, os dispositivos adaptativos naturalmente tendem a apresentar a flexibilidade requerida para atuarem em cenários dinâmicos. Assim, este trabalho investiga uma alternativa fundamentada no autômato finito adaptativo por meio do dispositivo denominado segmento digitalizado adaptativo, que incorpore o poder expressivo de representar parâmetros desses segmentos. Dentre esses parâmetros destacam-se a capacidade de representar as tolerâncias, a escalabilidade, os erros causados por desvios em ângulo ou em comprimento dos segmentos mencionados, resultando em estruturas mais flexíveis. Considerando que os métodos sintáticos são estruturais, os segmentos digitalizados adaptativos são modelados por conjuntos de regras, partindo-se de primitivas, concebendo-se as funções adaptativas correspondentes para alteração dos estados e de regras de transição. Posteriormente, estruturas mais elaboradas são concebidas relacionadas a arcos digitais pelos quais cadeias (*strings*) estimulam, em um passo único, autômatos finitos adaptativos que implementam segmentos digitalizados adaptativos. As implementações utilizam uma ferramenta cujo núcleo é um simulador para edição dos arquivos que compõem os autômatos. Conseqüentemente, o método proposto torna-se uma alternativa relativamente simples e intuitiva comparando-se com as abordagens existentes, apresentando capacidade de aprendizagem, além de ser computacionalmente poderosa.

Palavras-Chave: Computação Reconfigurável, Geometria e Modelagem Computacional, Reconhecimento de Padrões, Teoria dos Autômatos, Erros (Recuperação).

Abstract

For the representation of digital straight lines, digitized straight line segments and arcs, each of the available research approaches has its advantages and suitable applications. However, taking into account the complexities of real-world scenarios, the use of these representations is not so popular in situations that require flexible models or involving spurious interferences. Adaptive technologies are computer science formalisms able to change their behavior dynamically, without the interference of external agents, in response to incoming stimuli. By being able to respond to changing environmental conditions, adaptive devices naturally tend to have the required flexibility to work in dynamic scenarios. Thus, the purpose of this study is to investigate an alternative based on adaptive finite automaton through the device called adaptive digitized straight line segment, incorporating the expressive power to represent parameters of these segments. Among these parameters, emphasis is given to the ability to represent tolerances, scalability or errors caused by deviations in angle or length of the mentioned segments, resulting in more flexible structures. Whereas syntactic methods are structural, adaptive digitized straight line segments are modeled by sets of rules, starting from primitives, conceiving the corresponding adaptive functions to amend the set of states and transition rules. Later, more elaborate structures are designed related to digital arcs the corresponding strings of which stimulate, in just a single step, adaptive finite automata that implement adaptive digitized straight line segments. The implementations use a simulator for editing the files that compose the automata. Consequently, the proposed method reveals to be a simple and intuitive alternative capable of learning, besides being computationally powerful.

Keywords: Reconfigurable Computing, Geometry and Computational Modeling, Pattern Recognition, Automaton Theory, Errors (Recovering).

Sumário

Lista de Figuras

1	INTRODUÇÃO	20
1.1	Princípios fundamentais	22
1.1.1	Reticulado	22
1.1.2	Grade e pontos digitais	22
1.1.3	Arcos	23
1.1.4	Atributos	24
1.1.5	Primitivas	24
1.1.6	Equações diofantinas	25
1.1.7	Heurística	26
1.1.8	Linhas retas digitais	27
1.1.9	Vizinhanças principais	28
1.1.10	O <i>chain code</i>	30
1.1.11	Propriedades dos segmentos de retas digitais	31
1.1.12	Codificação	32
1.1.13	A propriedade da corda	34
1.1.14	Parametrização	35

1.1.15	Pré-imagem	35
1.1.16	Distância entre segmentos digitais	37
1.1.17	Retitude	37
1.1.18	Hierarquia das linguagens	38
1.2	O segmento de linha reta digitalizada adaptativo	39
1.3	Breve histórico	41
1.4	Objetivo	43
1.5	Justificativa	43
1.6	Organização da tese	45
2	CONCEITOS	47
2.1	Autômatos finitos adaptativos	47
2.1.1	Sumário sobre o autômato finito	49
2.1.2	Sumário sobre o formalismo do autômato finito adaptativo	50
2.1.3	Técnicas adaptativas básicas	56
2.1.4	Considerações sobre os autômatos finitos adaptativos	63
2.2	Características da representação digital	65
2.2.1	Transformações entre espaços	65
2.2.2	Conceitos complementares sobre o <i>chain code</i>	68
2.2.3	Métodos de quantização	68
2.2.4	Sensibilidade do chain code	71
2.3	As retas digitais clássica e analítica	72
2.3.1	O conceito clássico	72

2.3.2	O conceito analítico pelas linhas discretas aritméticas	77
2.3.3	Definições	80
2.4	Soluções pela geometria discreta aritmética	84
2.5	Métricas	86
2.6	Considerações finais	88
2.6.1	Segmento de linha reta digital	88
2.6.2	O poder computacional requerido para a análise sintática de segmentos digitalizados	89
2.6.3	Considerações sobre o estado-da-arte	90
2.6.4	Vantagens e desvantagens da representação adaptativa	92
3	ANÁLISE ESTRUTURAL	94
3.1	Ordem dos modelos	97
3.1.1	Modelo de primeira ordem	99
3.1.2	Modelo de segunda ordem	99
3.1.3	O efeito do ruído	100
3.2	Comprimento	102
4	ESTRUTURAS ADAPTATIVAS DA TESE	104
4.1	Prolegômenos	105
4.1.1	Recuperação de erros	105
4.1.2	Técnica utilizada em recuperação de erros	107
4.2	Variações em ângulo	108
4.2.1	Implementação de segmento digitalizado adaptativo	109

4.3	Erros em comprimento	112
4.4	Integração de diversos quadrantes	114
4.5	Expressividade do modelo proposto	115
4.6	Determinismo	116
5	SÍNTESE DA TESE	117
5.1	Introdução	117
5.1.1	Escalas adaptáveis	117
5.1.2	Atributos de arcos digitais	118
5.1.3	Métrica para segmentos de retas	120
5.2	A tese da vizinhança adaptativa	122
5.2.1	A vizinhança adaptativa	122
5.2.2	Implementação e testes de segmentos digitalizados adaptativos quanto à vizinhança adaptativa	124
5.2.3	Comparação com o modelo tradicional	133
5.2.4	Análise dos resultados	134
5.3	Sugestões de trabalhos futuros	139
5.3.1	Características da escala adaptável	140
5.3.2	Inferência de segmentos digitalizados	141
5.3.3	Função de vizinhança	149
5.4	Posicionamento desta pesquisa quanto ao estado-da-arte	149
6	SIMULAÇÕES COMPLEMENTARES	151
6.1	Implementações realizadas	151

6.1.1	Similaridade de formas quanto aos erros em comprimento sem considerar parâmetros de correção	153
6.1.2	Similaridade de formas quanto aos erros em ângulo sem considerar parâmetros de correção	154
6.1.3	Similaridade de formas quanto ao escalonamento	157
6.1.4	Similaridade de formas considerando erros em comprimento e ângulo	159
6.2	Conclusões sobre os estudos de caso	163
7	CONSIDERAÇÕES FINAIS	164
7.1	Sumário geral	164
7.2	Contribuições	166
7.3	Propostas para trabalhos futuros	170
7.4	Conclusão	175
	Referências	177
	Apêndice A – metodologia	183
A.1	Metodologia para fundamentar a tese	184
A.1.1	Ferramental de software e infra estrutura	185
	Apêndice B – código do autômato da Figura 6.10	187
	Apêndice C – lista de definições	190

Lista de Figuras

1.1	Dois reticulados equivalentes representados por linhas paralelas formando arranjos regulares que determinam o mesmo sistema de pontos (adaptado de G. Hardy (1979)).	23
1.2	Um círculo em uma grade ou reticulado (adaptado de Pierce (2010)). .	26
1.3	Um segmento de linha reta genérico L e o correspondente segmento digitalizado obtido por amostragem em uma grade. O conjunto de <i>pixels</i> esquematizado à direita da figura é obtido por amostragem do segmento de reta à esquerda relativamente à grade por algum processo pré-definido (adaptado de Gao e Leung (2002)).	27
1.4	À esquerda está representado o conjunto de pontos vizinhos de um ponto P em <i>vizinhança-4</i> . À direita está representado o conjunto de pontos vizinhos de um ponto P em <i>vizinhança-8</i> (adaptado de Rosenfeld (1998)).	29
1.5	À esquerda está a representação gráfica dos símbolos de 0 a 3 do <i>chain code</i> para <i>vizinhança-4</i> . À direita está a representação gráfica dos símbolos de 0 a 7 do <i>chain code</i> para <i>vizinhança-8</i> . Esses símbolos denotam o vetor unindo dois <i>pixels</i> vizinhos de um arco (adaptado de Aghito e Forchhammer (2006)).	30
1.6	À esquerda está a representação gráfica dos símbolos de a a d do <i>chain code</i> para <i>vizinhança-4</i> . À direita está a representação gráfica dos símbolos de a a h do <i>chain code</i> para <i>vizinhança-8</i> . Esses símbolos denotam o vetor unindo dois <i>pixels</i> vizinhos de um arco (adaptado de Aghito e Forchhammer (2006)).	31
1.7	Modelo de segmento digitalizado genérico no primeiro quadrante para a <i>vizinhança-4</i>	32

1.8	Primitivas em conectividade de vizinhança-4 utilizada nas implementações deste trabalho (adaptado de Aghito e Forchhammer (2006)) . . .	33
1.9	Segmento de linha reta digital em laranja com o código 0101 (<i>chain code</i> na vizinhança-8) e sua pré-imagem no plano (x,y) (área definida pelos pontos A, B, C e D da região em amarelo) (adaptado de Aghito e Forchhammer (2006)).	36
1.10	Hierarquia das linguagens (adaptado de Lewis e Papadimitriou (1981))	39
1.11	Interpretação geométrica da definição geral de linha discreta aritmética (adaptado de Figueiredo (1999)).	44
2.1	Dois exemplos de sequências conexas de estados: uma com transições que consomem o símbolo a e outra com transições em vazio. O último estado da sequência com transições em vazio é indeterminado.	48
2.2	Uma transição adaptativa genérica, onde R e S são funções adaptativas paramétricas opcionais dos tipos <i>anterior</i> e <i>posterior</i> respectivamente (adaptado de Neto (2001)).	52
2.3	Configuração inicial de autômato finito adaptativo.	57
2.4	Três primeiras configurações de autômato finito adaptativo	60
2.5	Configurações de autômato finito adaptativo após a configuração da Figura 2.4(c).	61
2.6	Configurações de autômato finito adaptativo após a configuração da Figura 2.5(c).	62
2.7	Configurações de autômato finito adaptativo após a configuração da Figura 2.6(b).	63
2.8	Representação de <i>chain code</i> de 4, 6 e 8 direções (adaptado de Tang (2001)).	70
2.9	Um mesmo contorno em duas grades diferentes (adaptado de (TANG, 2001)).	71

2.10	Segmento de linha reta digitalizado com o código 0101 e sua pré- imagem no plano (x,y) (área em amarelo) (adaptado de Aghito e Forchhammer (2006)).	74
2.11	Cinco tipos de domínios que são possíveis no espaço dos parâmetros (adaptado de Dorst e Smeulder (1984)).	75
2.12	Espaço de parâmetros, com a pré-imagem de dois segmentos com códigos de cadeia 0101 e 0100. A união das regiões compõem a pré- imagem do código da cadeia 010 (adaptado de Aghito e Forchhammer (2006)).	77
2.13	Interpretação geométrica da definição geral de linha discreta aritmética no plano $x y$ (adaptado de Figueiredo (1999)).	78
2.14	Uma linha digital representada por linha discreta aritmética (abaixo) e na representação clássica (acima) (adaptado de Figueiredo (1999)). . .	79
2.15	Fecho convexo: o polígono indicado que contém o conjunto de pontos representados, passando por alguns desses pontos, é o fecho convexo do conjunto de pontos planares (adaptado de Feschet (2006)).	81
2.16	Discretização de um segmento. Os quadrados com os pontos no centro são os <i>pixels</i> . Os quadrados sombreados são os <i>pixels</i> interceptados pelo segmento s (adaptado de Gioi (2008)).	82
2.17	Um conjunto de três pontos (abaixo) e o correspondente fecho convexo (acima) (adaptado de Feschet (2006)).	82
2.18	Dois segmentos de reta para a exemplificação de métricas (adaptado de Park (2009)).	87
2.19	Modelo de segmento digitalizado genérico no primeiro quadrante. . .	88
3.1	Exemplo de processo de modelagem. Nas três figuras da parte supe- rior são dados N pontos: (a) À esquerda, os pontos são aproximados por uma reta; (b) No meio, são aproximados por uma curva; (c) À direita, são aproximados por uma segunda curva. As três figuras da parte inferior mostram a reta e as curvas resultantes das aproximações (adaptado de Corona (2008))	96

3.2	Exemplo de um modelo contendo apenas o valor P da Expressão 3.1.	98
3.3	Modelo de primeira ordem, com P constante.	99
3.4	Modelo de segunda ordem, com P e Q constantes.	100
4.1	Esquemática de <i>Spyrograph</i> correspondente a uma reta que passa pela origem (adaptado de Klette e Rosenfeld (2004a)).	108
4.2	Autômato finito adaptativo para modelar a primeira unidade de segmento de um segmento digitalizado adaptativo	109
4.3	Configuração inicial de segmento digitalizado adaptativo, considerando variações em ângulo	110
4.4	Configuração do autômato finito adaptativo da Figura 4.3 após ativação da função adaptativa B	111
4.5	Exemplos de segmentos digitalizados modelados pelo autômato finito adaptativo das Figuras 4.3 e 4.4	112
4.6	Um <i>loop</i> adaptativo genérico, interpretado como caso particular de sequência conexa da Definição 32 cujos estados inicial e final são idênticos.	113
4.7	Variante determinística do autômato finito adaptativo da Figura 4.3.	116
5.1	Dois segmentos de reta representativos de um arco digital (adaptado de Park (2009)).	122
5.2	Modelos de ordens superiores (ordem n), com P variável.	123
5.3	Configurações de segmento digitalizado adaptativo cuja vizinhança varia adaptativamente em função do comprimento da cadeia de entrada.	125
5.4	Exemplos de arcos para teste do segmento digitalizado adaptativo da Figura 5.3(a) cuja vizinhança varia adaptativamente em função do comprimento da cadeia de entrada.	129

5.5	Configurações inicial de segmento digitalizado adaptativo cuja vizinhança varia adaptativamente em função do comprimento da cadeia de entrada.	130
5.6	Configurações de segmento digitalizado adaptativo da Figura 5.5(a) cuja vizinhança varia adaptativamente em função do comprimento da cadeia de entrada.	132
5.7	Exemplos de segmentos digitalizados reconhecidos pelo segmento digitalizado adaptativo da Figura 5.3(a); porém, não reconhecidos pelo método tradicional, sem adaptatividade.	133
5.8	Efeito dos atributos ângulo e comprimento locais na determinação de cantos (adaptado de Liu e Srinath (1990)).	136
5.9	Um retângulo digitalizado entre dois contornos delimitadores, numerados 1 e 2, e a zona de insensibilidade (adaptado de Lebedev (2004)).	138
5.10	Os erros nos vértices (cantos) de um retângulo como uma função do declive, ou ângulo de orientação. (adaptado de Lebedev (2004)). . . .	139
5.11	Um arco com irregularidades, passível de ser representado por um único segmento digitalizado. (adaptado de You e Fu (1979)).	140
5.12	Exemplo de região de tolerância (vizinhança adaptativa) para o quadrado representado no centro da região.	144
5.13	Uma célula no espaço tridimensional correspondente a uma região de tolerância especificada por vizinhança adaptativa.	145
5.14	Diagrama IDEF0 do sistema.	146
5.15	Diagrama IDEF0 detalhado da unidade de Análise e Classificação. . .	147
5.16	Esquematização do fluxo de dados sugerido em que o sistema gerente é a camada adaptativa mestre.	148
6.1	Um modelo simplificado de segmento digitalizado. Exemplo com 6 unidades de segmento.	152

6.2	Configuração inicial de autômato finito adaptativo que implementa segmento digitalizado adaptativo	153
6.3	Representação gráfica de cadeia aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8.	155
6.4	Representação gráfica de cadeia aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8	156
6.5	Representação gráfica de cadeia aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8	156
6.6	Representação gráfica de cadeia rejeitada pelo autômato finito adaptativo da Figura 6.2 e Quadro 8	157
6.7	Configuração inicial de autômato finito adaptativo que implementa segmento digitalizado adaptativo	158
6.8	Exemplo de triângulo classificado corretamente pelo autômato finito adaptativo da Figura 6.7.	158
6.9	A mesma forma do triângulo da Figura 6.8, em maior escala, classificada corretamente pelo autômato finito adaptativo da Figura 6.7. . . .	159
6.10	Configuração inicial de autômato finito adaptativo que implementa segmento digitalizado adaptativo, incorporando parâmetro de correção de erros em comprimento	160
6.11	Formas classificadas corretamente com 20% de tolerância em comprimento.	162
7.1	Panorama geral apresentando trabalhos importantes e época de publicação. Obs: SLRD é a abreviatura de segmentos de linhas retas digitalizadas, abreviados “segmentos digitalizados”.	165
7.2	Resumo das contribuições desta tese.	167
7.3	Segmento digitalizado adaptativo para classificação de formas.	171
7.4	A imagem à esquerda mostra a folha de uma samambaia. À direita, a imagem segmentada (adaptado de (GAUVIN, 2002)).	172

7.5	A imagem à esquerda mostra um campo de futebol. À direita, a imagem segmentada em linhas retas apresentando a região de tolerância em vermelho (adaptado de Szenberg (2001)).	173
7.6	A figura esquematiza um robô e uma porta com respectivas dimensões para identificação de estados. (adaptado de Aguiar (2005)).	175

1 INTRODUÇÃO

Diz-se que um dispositivo é adaptativo sempre que seu comportamento se altera dinamicamente, em resposta a estímulos de entrada, sem interferência de outros agentes externos, inclusive de operadores ou usuários, executando as correspondentes automodificações (NETO, 2001). Por serem capazes de responder às variações nas condições ou situação momentânea do ambiente, os dispositivos adaptativos apresentam naturalmente a flexibilidade requerida para atuarem em cenários dinâmicos. Normalmente, são compostos de duas camadas descritas pela Definição ¹ 1.

Definição 1 *Dispositivo adaptativo*: um dispositivo adaptativo AD é da forma $AD = (ND_0, AM)$ tal que ND_0 é o dispositivo subjacente e AM é o mecanismo ou camada adaptativa associada, formalizada nos mesmos moldes que ND_0 . O dispositivo subjacente é normalmente não-adaptativo (NETO, 2001).

A camada adaptativa da Definição 1 integra o conjunto de ações adaptativas, responsáveis por alterar a estrutura do dispositivo não adaptativo subjacente dinamicamente em resposta aos estímulos.

Definição 2 *Estímulo*: é toda ocorrência de evento ou situação especial capaz de ativar e provocar reações de dispositivos adaptativos formais (“[...] adaptive actions, which are self-modification procedures intended to be activated as reactions of adaptive formal devices to the occurrence of special events or situations” (NETO, 2000)).

Em outras palavras, a essência dos conceitos apresentados pelas Definições 1 e 2 está na capacidade de os dispositivos adaptativos executarem ações adaptativas, que são procedimentos de automodificação destinados a serem ativados como reações de dispositivos adaptativos formais para as ocorrências de eventos ou situações especiais,

¹As definições são apresentadas em três formatos, descritos a seguir. Definições correspondendo à transcrição exata de trechos de autores são apresentadas no formato de citação direta, entre aspas. Definições correspondendo à transcrição adaptada de trechos de autores são apresentadas no formato normal, sem espaçamento das margens do texto. Definições do autor deste trabalho apresentam espaçamento das margens esquerda e direita do texto.

denominados estímulos. A adaptatividade tem como uma de suas finalidades estender formalismos consolidados, aumentando seu poder de expressão (PISTORI, 2003). O acréscimo em complexidade na inclusão de ações adaptativas a formalismos clássicos pela Definição 1, não só resulta em notável incremento de poder expressivo, mas também em flexibilidade, tendo em vista a possibilidade de escolher qualquer mecanismo consolidado como dispositivo não adaptativo subjacente.

Segundo Visnevski (2005), é vantajoso modelar padrões complexos por uma representação compacta usando os benefícios da teoria já estabelecida dos autômatos finitos. Devido a esse fato, associado à sua relativa simplicidade, selecionou-se o autômato finito neste trabalho como dispositivo não adaptativo subjacente, obtendo-se o autômato finito adaptativo representado pela Expressão 1.1.

$$AFA = (ND_0, AM). \quad (1.1)$$

Esse acréscimo em complexidade na integração de ações adaptativas ao formalismo do autômato finito, resulta em um mecanismo adaptativo com poder computacional equivalente ao da máquina de Turing (ROCHA; NETO, 2001).

Partindo dos estudos de Neto (1993) na área de construção de compiladores utilizando autômatos de pilha estruturados como dispositivo não adaptativo subjacente, surgiram aplicações nas mais diversas áreas, abrangendo: aprendizagem computacional (ROCHA; NETO, 2000), processamento de linguagens naturais (MENEZES, 2000.), ambientes multilinguagens (FREITAS; NETO, 2000.), robótica (SOUSA, 2006), e outras.

O acervo de recursos teóricos, conceituais e de ferramental nas mais diversas aplicações que foram sendo acumulados em adaptatividade pode ser avaliado em Neto (2007). No entanto, a despeito da relevância da representação computacional de linhas, linhas retas e arcos, inclusive sendo uma área ativa de pesquisas há quase meio século conforme os levantamentos de Klette e Rosenfeld (2004b) e Bhowmick e Bhattacharya (2007), esse assunto ainda não foi estudado sob o enfoque adaptativo com os recursos do mencionado acervo em adaptatividade, até o momento.

A hipótese inicial deste trabalho é a viabilidade em modelar propriedades das retas digitais por um conjunto de regras a fim de aplicar a adaptatividade. Dar-se-á ênfase a retas no plano. A pesquisa se aplica também à representação de arcos quaisquer, considerando que os mesmos podem ser representados pela concatenação de segmentos.

1.1 Princípios fundamentais

As definições apresentadas neste item visam introduzir conceitos iniciais.

1.1.1 Reticulado

Os reticulados (*lattices*) apresentam todo um formalismo que não faz parte do objetivo principal desta pesquisa. Entretanto, é conveniente entender a sua conceituação, mesmo que de forma simplificada. G. Hardy (1979) apresenta detalhes sobre a fundamentação matemática dos reticulados.

Definição 3 Reticulados: *são representados por figuras de linhas definindo arranjos regulares de pontos no espaço euclidiano (G. HARDY, 1979). O conjunto de pontos definidos, ou seja, o sistema de pontos, correspondem às intersecções das linhas de acordo com a Definição 4.*

Definição 4 Reticulado-pontos: *é um conjunto de pontos cujas coordenadas $(x_0, x_1, x_2, \dots, x_{n-1}) \in \mathbb{R}^n$ são números inteiros em \mathbb{Z}^n . Assim, uma maneira de especificar reticulados é pelas soluções inteiras de uma dada equação ou sistema de equações (G. HARDY, 1979).*

Dois reticulados diferentes são denominados equivalentes quando determinam o mesmo conjunto reticulado-pontos, conforme a Figura 1.1. Nessa figura um reticulado tem o seu arranjo de pontos associados aos segmentos \overline{OP} e \overline{OQ} ; o outro reticulado, nos segmentos \overline{OP} e \overline{OR} .

1.1.2 Grade e pontos digitais

Existem vários tipos de grade, sendo que se utiliza a mais comum nesta pesquisa, cuja definição é apresentada a seguir.

Definição 5 Grade: *uma grade bi-dimensional quadrada é uma rede de linhas horizontais e verticais que fornecem as coordenadas para a localização de pontos em uma imagem, por meio dos pontos de grade relativos à intersecção dessas linhas. Esses pontos (x, y) de intersecção apresentam coordenadas inteiras (ROSENFELD, 1998).*

As imagens, antes de serem submetidas a processos computacionais, são codificadas por amostragens dos seus valores de brilho relativamente a uma grade discreta de pon-

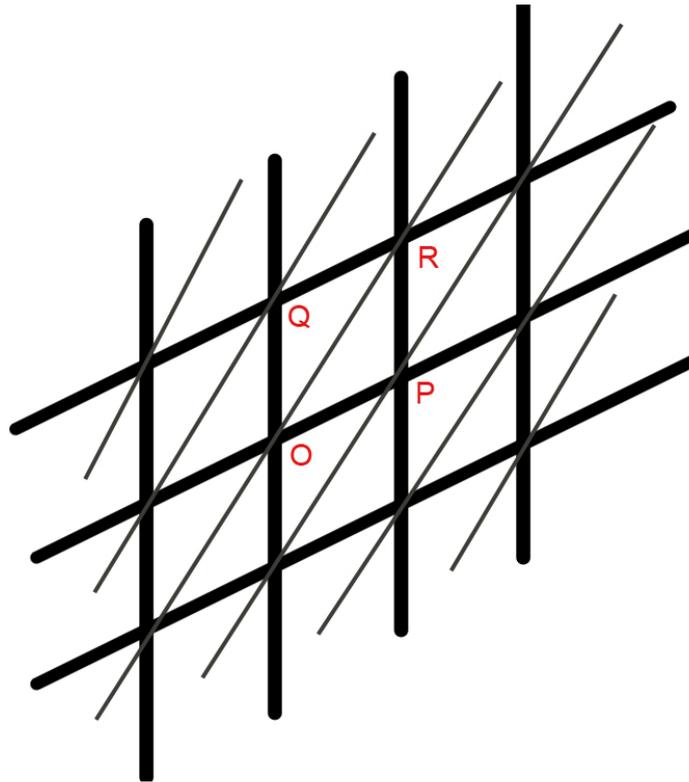


Figura 1.1: Dois reticulados equivalentes representados por linhas paralelas formando arranjos regulares que determinam o mesmo sistema de pontos (adaptado de G. Hardy (1979)).

tos, nos moldes da Definição 5; seguida pela digitalização desses valores para um número finito de valores binários. O resultado deste processo resulta em uma imagem digital na forma de uma matriz retangular de valores discretos. Os pontos digitais dessa matriz são os denominados *pixels* (abreviatura de *picture elements*). O valor de um *pixel* é chamado de nível de cinza.

Assim, a conversão de uma imagem em números costuma ser efetuada por uma amostragem uniformemente espaçada da mesma, gerando uma grade (*grid*) regular composta normalmente por quadrados de medidas iguais. Cada um desses quadrados é uma célula denominada *pixel*, aos quais são associados a informação sobre o nível de cinza que cada quadrado representa. Portanto, supondo uma grade discreta relativa a um sistema de coordenadas, atribui-se um *pixel* a cada quadrado ou célula da grade.

1.1.3 Arcos

Definição 6 *Arco:* "um arco $\widehat{X_1X_2}$ é uma linha direcionada com ponto de partida X_1 e ponto final X_2 , que apresenta uma função de curvatura $f(l)$ ao longo da linha direcionada, com $0 < l < L$, onde L é o comprimento total do segmento" (YOU; FU, 1979).

A Definição 6 pode ser aplicada tanto para o caso contínuo quanto para o digital, incluindo-se o termo “digital” ou “contínuo” para esclarecer o texto. $f(l)$ corresponde à derivada da direção do segmento ao longo do segmento, relativamente ao comprimento.

Definição 7 *Segmento de arco digital:* o segmento de arco digital S é entendido como um conjunto de pontos de grade interligados, “em que cada ponto desse conjunto tem exatamente apenas dois vizinhos (considerando vizinhança-4 ou vizinhança-8); exceto dois desses pontos, denominados extremos, que possuem apenas um vizinho em S ” (ROSENFELD, 1974). Segmento de arco digital é o resultado da digitalização de um arco ou reta contínuos, e que apresenta atributos de comprimento, de orientação e de informações estruturais possível de localizá-lo relativamente a outros segmentos (GAO; LEUNG, 2002)

No caso mais geral, um segmento de arco pode ser decomposto em segmentos concatenados.

1.1.4 Atributos

Na Definição 6 a função de curvatura $f(l)$ é um dos atributos do segmento de curva. Tais atributos são comentados em maiores detalhes no decorrer deste texto, em especial no capítulo 5. Em termos gerais, atributos são valores indicativos das condições, propriedades ou qualidades peculiares de alguém ou de alguma coisa. A Definição 8 apresenta alguns atributos de segmentos.

Definição 8 *Atributos de segmentos:* são valores associados a segmentos específicos, tal como os correspondentes comprimentos, ângulos de orientação ou outras informações estruturais.

1.1.5 Primitivas

Primitivas representam atributos atômicos ou elementares, no sentido de não serem decompostos em atributos mais simples, segundo o esclarecimento na página 328 de Bell (2006). A Definição 9 é relativa ao caso geométrico.

Definição 9 *Primitiva:* é um objeto geométrico elementar tal como uma linha, um círculo, ou um polígono (ANDRES; ACHARYA; SIBATA, 1997).

Entretanto, como a Definição 9 não especifica claramente o seu contexto de aplicação, a Definição 10 particulariza a primitiva digital.

Definição 10 *Primitiva digital:* *é o resultado de um processo de aproximação local de uma primitiva contínua. Tal processo de aproximação local é denominado digitalização (ANDRES; ACHARYA; SIBATA, 1997).*

1.1.6 Equações diofantinas

A denominação diofantina caracteriza principalmente os tipos de soluções desejadas para tais equações, e não tanto as propriedades das mesmas. Denominadas diofantinas em homenagem a Diofanto de Alexandria, que se supõe tenha vivido no século terceiro D.C., essas equações podem não ter solução, ou ter número finito ou infinito de soluções.

Definição 11 *Equação diofantina:* *uma equação diofantina é uma expressão algébrica com uma ou mais incógnitas contendo coeficientes inteiros, para a qual são buscadas soluções de números inteiros ou números racionais. É uma equação da forma $N = M$, em que N e M são polinômios com coeficientes inteiros. No caso mais geral, podem compor um sistema de equações diofantinas (PIERCE, 2010).*

Para se entender a importância das equações diofantinas, o texto básico de Pierce (2010) mostra o “problema do círculo” que o matemático Carl Friedrich Gauss propôs em 1801.

Pierce (2010) supõe a grade da Figura 1.2 composta de quadrados no plano de lados de comprimento unitário. Escolhe-se nessa grade um ponto denominado origem, de coordenada $(0,0)$. As coordenadas dos demais pontos são números inteiros pertencendo ao conjunto $\{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$. Tais pontos são os reticulado-pontos.

Dado um círculo de raio r nessa grade, a área do círculo é sabidamente dada pela equação $A(r) = \pi r^2$. Entretanto, o problema de Gauss é a determinação da quantidade de pontos da grade que estão no interior do círculo.

Uma aplicação das aproximações diofantinas é na aproximação de formas lineares por pontos numa grade. Em ciência da computação, os denominados “problemas da grade” são uma classe de problemas de otimização de grades que empregam essas equações. A intratabilidade conjecturada desses problemas tem sido a base para a construção de sistemas seguros de criptografia baseada em grades. Simplificadamente, o processo é

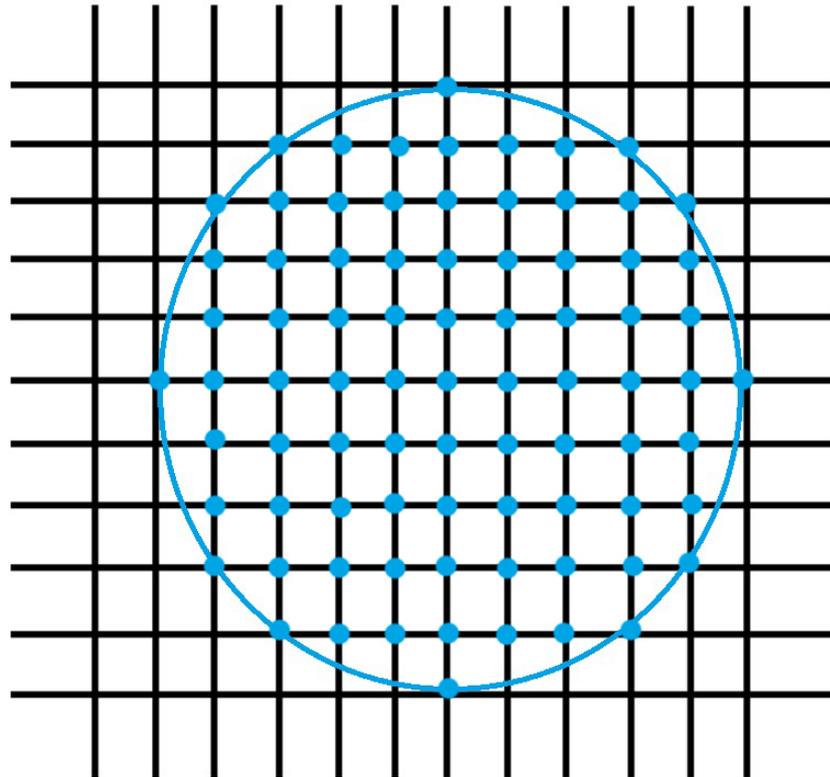


Figura 1.2: Um círculo em uma grade ou reticulado (adaptado de Pierce (2010)).

análogo a aproximar a circunferência do círculo do problema de Gauss pelos pontos de grade mais próximos.

O assunto da aproximação diofantina se relaciona com a teoria dos números, mais especificamente com a aproximação de números reais por números racionais. Por exemplo, a denominada aproximação homogênea diofantina de um número real β envolve encontrar números inteiros diferentes de zero r e s que resultem em um valor mínimo de $r\beta - s$ ou em $\beta - \frac{r}{s}$ pequeno em valor absoluto. Esses inteiros obtidos, diferentes de zero, que resultem em um erro de aproximação menor que um determinado valor, seriam a melhor aproximação.

1.1.7 Heurística

Em Computação, o termo heurística não é preciso (*a rule of thumb*), no sentido de que pode apresentar mais de um significado de acordo com o entendimento do utilizador. Porém, na maioria das vezes, está associado ao uso de regras da experiência ou da intuição, portanto não demonstradas, na pesquisa de algoritmos e de métodos. Heurística e método heurístico são denominações para o algoritmo que fornece soluções

sem um limite formal de qualidade, comumente avaliado empiricamente em termos de complexidade e qualidade das soluções, portanto, sem garantias de soluções ótimas, de acordo com o descrito na página 127 e capítulo 12 de Harel (1987).

A metodologia heurística é viável principalmente em situações complexas, ainda não totalmente dominadas pela comunidade científica. Há casos em que o termo se aplica a apenas uma classe de problemas, de acordo com a Definição 12.

Definição 12 *Heurística direcionada:* considera-se, nesta pesquisa, o termo “heurística direcionada” significando uma heurística restrita a um único ou a determinados problemas, e que não leva em conta o conjunto de situações e parâmetros envolvidos em diferentes cenários. Portanto, a aplicação da heurística direcionada não resulta na solução ótima ou provavelmente boa para o problema em todos os casos quanto, por exemplo, à eficácia da aplicação de recursos computacionais.

1.1.8 Linhas retas digitais

A Figura 1.3 indica o processo de digitalização de um segmento de reta euclidiano genérico e o correspondente segmento digitalizado obtido por amostragem numa grade de espaçamento Δ .

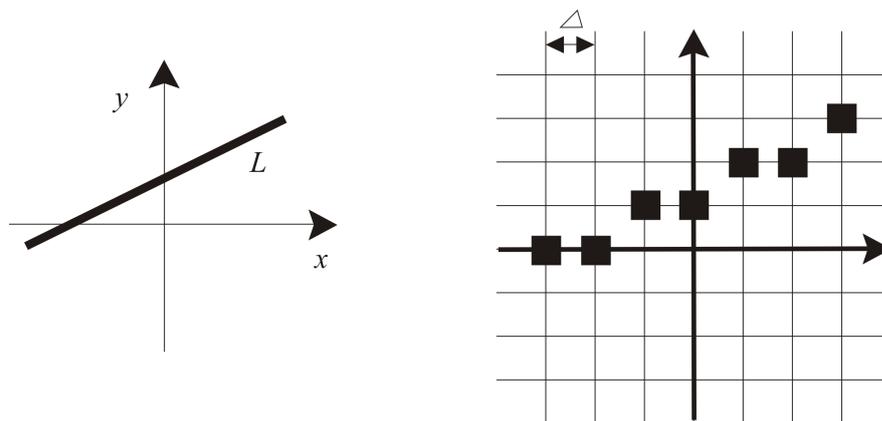


Figura 1.3: Um segmento de linha reta genérico L e o correspondente segmento digitalizado obtido por amostragem em uma grade. O conjunto de *pixels* esquematizado à direita da figura é obtido por amostragem do segmento de reta à esquerda relativamente à grade por algum processo pré-definido (adaptado de Gao e Leung (2002)).

As Definições 13 e 14 são pertinentes, tendo em vista que, pelo conjunto de *pixels* da Figura 1.3, podem ser traçadas um número infinito de retas euclidianas que resultariam no mesmo conjunto.

Definição 13 *Linha reta digital:* é uma reta especificada por um conjunto de pixels relativos a uma grade bi-dimensional que corresponde à digitalização de uma linha reta euclidiana qualquer por um processo de amostragem pré-definido, representado na Figura 1.3. Note que essa linha reta euclidiana pode ser qualquer uma, dentre as infinitas retas que resultariam no mesmo conjunto de pixels pelo processo de amostragem (DORST; SMEULDER, 1987).

Definição 14 *Linha reta digitalizada:* é uma reta especificada por um conjunto de pixels relativos à grade bi-dimensional que corresponde à digitalização de uma linha reta euclidiana particular por um processo de amostragem representado na Figura 1.3. Note que trata-se de uma linha reta euclidiana particular, específica dentre o número infinito de retas que resultariam no mesmo conjunto de pixels pela digitalização (DORST; SMEULDER, 1987).

Conforme Dorst e Smeulder (1987), a diferença entre linha reta digitalizada da Definição 14 e linha reta digital da Definição 13 é sutil pois a última é um conjunto de pixels correspondente à digitalização de uma reta euclidiana qualquer; enquanto a primeira é resultado da digitalização de uma reta euclidiana particular.

Definição 15 *Segmento de linha reta digital:* é um conjunto finito de pixels relativos a uma grade, pelo qual se pode caracterizar um conjunto de segmentos de linhas retas euclidianas cuja digitalização resulta no mesmo conjunto de pixels (DORST; SMEULDER, 1987).

Definição 16 *Segmento de linha reta digitalizado:* (abreviado segmento digitalizado) é um conjunto finito de pixels relativos a uma grade, resultado da digitalização de um segmento de linha reta euclidiana particular (DORST; SMEULDER, 1987).

1.1.9 Vizinhanças principais

Rosenfeld (1998) apresentar os dois tipos principais de vizinhança de um determinado pixel definidos por uma grade regular sobreposta a uma imagem. Para simplificar, Rosenfeld (1998) se restringe às imagens digitais de apenas dois valores 0 e 1 de intensidade (nível de cinza), definindo-as da seguinte maneira: imagem digital é o mapeamento dos pontos do plano que têm coordenadas inteiras, associados às intensidades 0 ou 1, tal que, nesse mapeamento, cada célula da grade regular corresponde a um pixel.

A Figura 1.4 indica um ponto P da grade em questão como referência. Este ponto P tem duas classes principais de pontos de grade vizinhos. Na primeira, à esquerda

da Figura 1.4 estão indicados quatro pontos vizinhos nas direções norte, sul, leste e oeste, considerados em vizinhança-4: um diretamente acima do ponto de referência, o segundo abaixo do ponto, e os dois últimos à sua direita e à sua esquerda com relação à grade. A segunda classe são os quatro vizinhos diagonais nas direções noroeste, nordeste, sudoeste, e sudeste, à esquerda (ROSENFELD, 1998). Os pontos em vizinhança-4 adicionados aos pontos vizinhos diagonais resultam no conjunto de pontos em vizinhança-8 de P , à direita da figura.

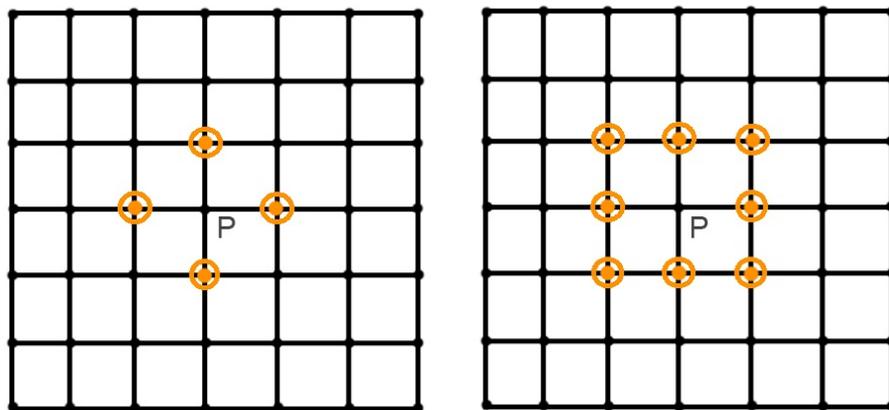


Figura 1.4: À esquerda está representado o conjunto de pontos vizinhos de um ponto P em vizinhança-4. À direita está representado o conjunto de pontos vizinhos de um ponto P em vizinhança-8 (adaptado de Rosenfeld (1998)).

A seguir, a adjacência-4 e a adjacência-8 são entendidas da seguinte maneira por Rosenfeld (1998).

Definição 17 Adjacência-8: dois pontos de grade estão em adjacência-8 num determinado plano caso atendam às seguintes condições:

- São diferentes entre si;
- Cada coordenada de um difere da coordenada correspondente do outro por, no máximo, uma unidade (ROSENFELD, 1998).

Definição 18 Adjacência-4: dois pontos de grade no plano estão em adjacência-4 se atenderem às duas condições seguintes:

- São pontos adjacentes-8;

- *Diferem, no máximo, em apenas uma de suas coordenadas correspondentes (RO-SENFELD, 1998).*

1.1.10 O chain code

O *chain code* foi desenvolvido por Freeman (1970) como um descritor de contornos de apenas um *pixel* de espessura.

Definição 19 *Chain code*: o *chain code*, ou o código da cadeia, é uma sequência de elementos onde cada elemento é um símbolo que representa o vetor unindo dois *pixels* vizinhos de um arco digital. Dentre suas finalidades, está a de representar o arco digital em questão e seus atributos, tal como a função de curvatura mencionada na Definição 6 (DORST; SMEULDER, 1987) (FREEMAN, 1970).

A Figura 1.5 mostra um exemplo do relacionamento dos símbolos do *chain code* com a vizinhança-4 e vizinhança-8. As vizinhanças da Figura 1.6 são as mesmas da Figura 1.5, apenas com símbolos diferentes.

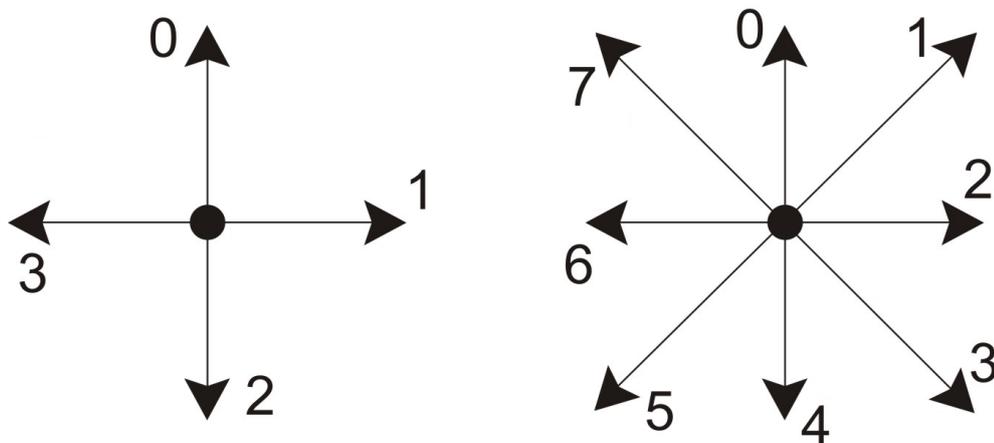


Figura 1.5: À esquerda está a representação gráfica dos símbolos de 0 a 3 do *chain code* para vizinhança-4. À direita está a representação gráfica dos símbolos de 0 a 7 do *chain code* para vizinhança-8. Esses símbolos denotam o vetor unindo dois *pixels* vizinhos de um arco (adaptado de Aghito e Forchhammer (2006)).

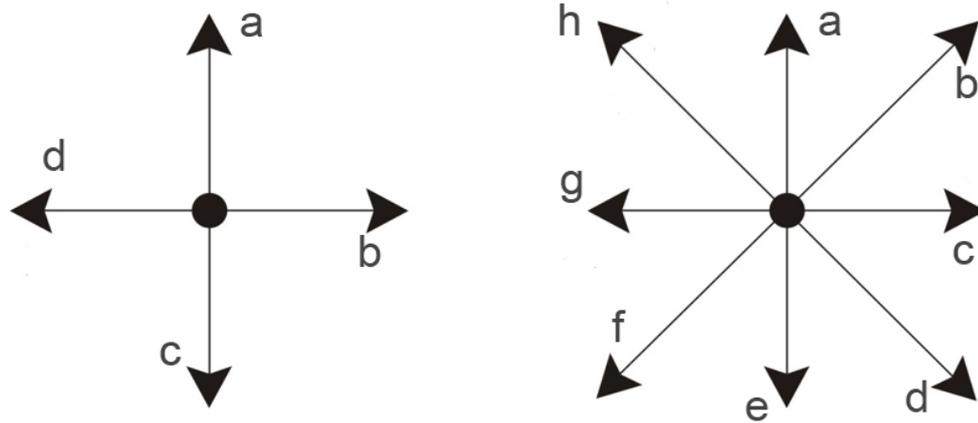


Figura 1.6: À esquerda está a representação gráfica dos símbolos de *a* a *d* do *chain code* para vizinhança-4. À direita está a representação gráfica dos símbolos de *a* a *h* do *chain code* para vizinhança-8. Esses símbolos denotam o vetor unindo dois *pixels* vizinhos de um arco (adaptado de Aghito e Forchhammer (2006)).

A escolha de uma vizinhança específica nos trabalhos não reduz a generalização das soluções (AGHITO; FORCHHAMMER, 2006) (PROFFITT; ROSEN, 1979). Comparando-se a vizinhança-4 com a vizinhança-8, a primeira apresenta simplicidade nas implementações, além dos algoritmos de processamento digital considerarem normalmente apenas 2 eixos x e y , porém tem envolver maior perda de informação, relativamente à vizinhança-8. A escolha da vizinhança nos trabalhos pode ser também determinada pela notação requerida.

1.1.11 Propriedades dos segmentos de retas digitais

Juntamente com o *chain code*, Freeman (1970) conjecturou um modelo definidor de retas digitais de acordo com a Definição 20. A partir dessa conjectura, as pesquisas sobre segmentos de linhas retas digitais e segmentos digitalizados mantêm até hoje a sua natureza desafiadora (KLETTE; ROSENFELD, 2004b), incluindo a representação de arcos genéricos pela concatenação de segmentos digitalizados (YOU; FU, 1979).

Definição 20 *Propriedades dos segmentos de retas digitais:* no modelo de Freeman, as cadeias que representam linhas retas obedecem a três propriedades, em uma codificação utilizando símbolos consecutivos de 0 até 7, em vizinhança-8, ou de 0 até 3 em vizinhança-4:

Prop1: No máximo dois tipos de símbolos, representando direções distintas no código do chain, podem estar presentes, e estes são símbolos consecutivos corres-

pondentes do chain, módulo oito para a vizinhança-8 ou módulo quatro para a vizinhança-4;

Prop2: Um dos dois símbolos sempre ocorre isoladamente, solitário;

Prop3: As ocorrências sucessivas do símbolo isolado são tão uniformemente espaçadas quanto possível entre códigos do outro valor, que ocorrem em grupos ou corridas (runs) (adaptado de (FREEMAN, 1970)).

Caso não sejam usados dígitos para representar as direções, alterar os caracteres relacionados às direções para dígitos a fim de atender Prop1. A título de exemplo para a vizinhança-4 da Figura 1.6: alterar a para 0, b para 1, c para 2 e d para 3.

Na vizinhança-8, o significado de *Prop1* a *Prop3* é representar a linha por uma sequência de vetores com ângulos de inclinação múltiplos de 45° e cujos comprimentos são 1 (se horizontal ou vertical) ou $\sqrt{2}$ (se diagonal), exigindo corridas de *pixels* de, no máximo, duas direções, as quais diferem de 45° .

Na vizinhança-4, o significado de *Prop1* a *Prop3* é representar a linha por uma sequência de vetores de comprimento unitários que são sempre na horizontal ou vertical, exigindo corridas de *pixels* de, no máximo, duas direções, as quais diferem de 90° .

A título de exemplo para a vizinhança-4, a Figura 1.7 procura representar corridas de *P* e *Q* símbolos *b*, tão uniformemente espaçadas quanto possível entre códigos do outro valor *a*, que ocorre isolado.

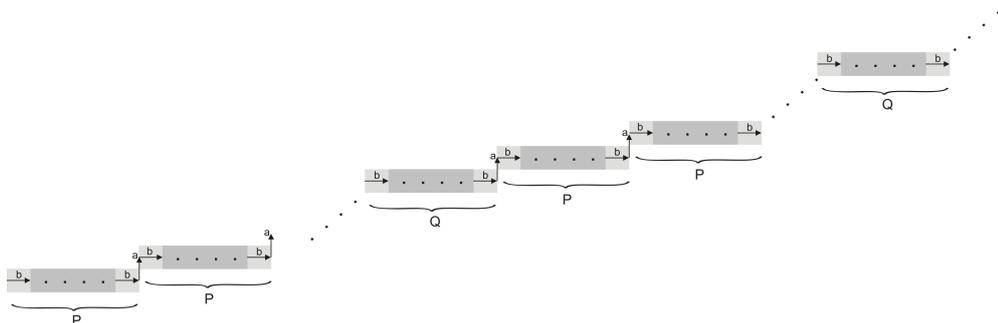


Figura 1.7: Modelo de segmento digitalizado genérico no primeiro quadrante para a vizinhança-4.

1.1.12 Codificação

Caso nada em contrário seja especificado, sem qualquer redução em generalidade, utiliza-se neste trabalho a vizinhança-4 como padrão: símbolos da propriedade *Prop1*

da Definição 20 devem ser consecutivos, módulo quatro.

Uma cadeia é uma sequência de zero ou mais símbolos do alfabeto Σ . Na vizinhança-4, os símbolos de $\Sigma = \{a, b, c, d\}$ da Figura 1.8 estão associados a direções com propósito de representar os arcos por uma sequência de vetores na horizontal ou vertical de comprimentos unitários. Mais precisamente, os símbolos que compõem as cadeias pertencem a $\Sigma = \{a, b, c, d\}$ como mostrado na Figura 1.8.

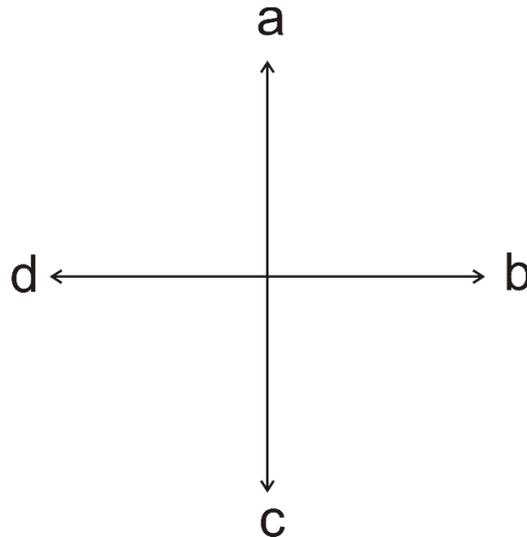


Figura 1.8: Primitivas em conectividade de vizinhança-4 utilizada nas implementações deste trabalho (adaptado de Aghito e Forchhammer (2006))

Os símbolos de Σ indicados na Figura 1.8 são também denominados como primitivas, *tokens*, elementos do *chain code* ou simplesmente estímulos. A Figura 1.7 mostra um exemplo de segmento digitalizado nessa codificação com símbolos *a* e *b*.

O conjunto de todas as cadeias possíveis com o alfabeto Σ é denotado por Σ^* . O comprimento de uma cadeia qualquer S é denotado por $|S|$. A cadeia vazia, de comprimento zero, é representada por ε . O i -ésimo símbolo de uma cadeia S é representado por s_i . Uma cadeia qualquer $S = s_1 \dots s_n$ pode ser também representada pelo seu símbolo, seguido pela descrição do seu i -ésimo elemento s_i tal como:

$$S : s_i; i = 1, 2, \dots, n. \quad (1.2)$$

Caso todos os elementos de S sejam idênticos $s = s_1 = s_2 = \dots = s_i = \dots = s_{n-1} = s_n$ uma representação compacta é $S = s^n$.

Dada a cadeia $S = s_1 \dots s_n$ e a cadeia $V = v_1 \dots v_n$, a operação que justapõe o conteúdo dessas duas cadeias $SV = s_1 \dots s_n v_1 \dots v_n$ é denominada concatenação: SV é a concate-

nação de S e V . Similarmente, $S_1S_2\dots S_n$ é a concatenação das cadeias S_1, S_2 , até, por último, S_n .

Na cadeia $S_1S_2\dots S_n$: cada cadeia S_1, S_2 e, sequencialmente, até S_n são também denominadas sub-cadeias.

A título ilustrativo, em vizinhança-8, os seguintes segmentos digitalizados S_1 e S_2 obedecem a uma mesma estrutura, apesar de terem comprimentos diferentes:

$$S_1 = (a^2b)^4(ab)(a^2b)^4(ab)$$

$S_2 = [(a^2b)^4(ab)]^2(a^2b)^3(ab)[(a^2b)^4(ab)]^2(a^2b)^3(ab)$. Em ambos, S_1 e S_2 , seguindo as propriedades *Prop1* e *Prop2*, o símbolo b ocorre solitário quando junto com o símbolo a , o qual ocorre em grupos ou corridas (*runs*) de períodos variáveis (a^2 e a^1). As ocorrências sucessivas de b são tão uniformemente espaçadas quanto possível entre códigos de a atendendo *Prop3*. Uma maneira de identificar a existência de uma única estrutura nos dois segmentos digitalizados acima é pela comparação das sequências de símbolos com os modelos apresentados por Li e Loew (1988), identificando o modelo que melhor se ajusta aos segmentos. Com base na propriedade da corda, Li e Loew (1988) apresentaram padrões mostrando como o arranjo de símbolos dos segmentos digitalizados se apresenta graficamente.

1.1.13 A propriedade da corda

Contudo, como a terceira propriedade *Prop3* da Definição 20 não é muito clara, foi posteriormente comprovado por Rosenfeld (1974), para a vizinhança-8, que a condição necessária e suficiente para um arco digital C ser uma reta digital, é atender à propriedade da corda. Também Hung (1985) demonstrou que a inexistência de irregularidades (*unevenness*) na estrutura hierárquica básica do código, atende à mencionada propriedade.

Definição 21 *Propriedade da corda: os seguinte aspectos são pertinentes:*

- *A propriedade da corda foi definida inicialmente para a vizinhança-8 da seguinte maneira: Diz-se que um arco digital C , representando objetos sólidos delgados em uma imagem digitalizada apresenta a propriedade da corda “se, para cada dois pontos digitais c e d pertencentes a C , e para cada ponto $p = (x, y)$ em \overline{cd} , existe um ponto $e = (h, k)$ pertencente a C tal que $\max\{|x - h|, |y - k|\} < 1$ onde \overline{cd} é o segmento de reta entre c e d ” (ROSENFELD, 1974).*
- *A propriedade da corda foi estendida para a vizinhança-4, tal que a expressão*

$\max \{|x - h|, |y - k|\} < 1$ é válida apenas para a vizinhança-8, pois na vizinhança-4 não existem primitivas na diagonal (RONSE, 1986).

- *Prop4: A propriedade da corda implicou na propriedade Prop4, adicional às mencionadas na Definição 20, com a descrição seguinte: quanto à direção referente ao símbolo que ocorre em grupos (não isolado), as corridas correspondentes podem ocorrer com apenas dois valores, os quais diferem de uma unidade (por exemplo, P e $P + 1$).*

Na Definição 21, a Expressão $\max \{|x - h|, |y - k|\} < 1$ restringe a espessura dos arcos digitais a apenas um *pixel*.

1.1.14 Parametrização

Entende-se por parâmetro todo elemento cuja variação de valor altera a solução de um problema, sem alterar-lhe a natureza, da mesma maneira que determinada quantidade entra em uma equação representativa de um arco, a qual se pode fazer variar sem alterar a natureza desse arco, como, por exemplo: o raio R da circunferência em $x^2 + y^2 = R^2$. Daí, pode-se estudar o problema num espaço de parâmetros, fazendo-se variar o valor do parâmetro e obtendo-se diferentes soluções ((adaptado de Weiszflog (2010)). Conforme a introdução apresentada por Dorst e Smeulder (1984), qualquer segmento de reta euclidiano pode ser representado por quatro parâmetros, considerando-se tal segmento como parte de uma reta. Para o sistema cartesiano ortogonal com eixo das abscissas x e das ordenadas y , uma linha reta pode ser expressa por $y = \alpha x + e$, onde α indica a inclinação (o ângulo da linha com respeito ao eixo das abscissas x , e e indica o deslocamento (*offset*) vertical para $x = 0$. As constantes α e e são utilizadas na parametrização dos segmentos. Os dois parâmetros restantes são obtidos pelos pontos extremos do segmento. Desse modo, pode-se estudar os segmentos num espaço de parâmetros (α, e) . A representação dos segmentos digitalizados por meio de cadeias nem sempre é a mais conveniente para os cálculos e, portanto, em alguns trabalhos utilizam-se elementos que compõem parametrizações concisas de uma cadeia arbitrária. Entende-se tal descrição paramétrica como uma contrapartida discreta do caso contínuo Dorst e Smeulders (1991).

1.1.15 Pré-imagem

A Definição 22 apresenta o conceito de pré-imagem.

Definição 22 *Pré-imagem:* dado um conjunto de pixels componentes de um segmento digital em linha reta (ou seja, não está sendo enfatizado se pertence à digitalização de uma reta euclidiana específica), pré-imagem é o conjunto de linhas retas euclidianas, cuja digitalização resulta no mesmo segmento inicial dado. Demonstra-se que a pré-imagem define uma região na forma de um polígono que contém o conjunto de linhas retas euclidianas mencionadas (DORST; SMEULDER, 1987).

A fim de esclarecer a Definição 22, a Figura 1.9 mostra um segmento digital e os vértices A, B, C e D do polígono da pré-imagem. As equações $y = \frac{2}{3}x$, $y = \frac{1}{2}x + \frac{1}{2}$, $y = \frac{1}{2}x$ e $y = \frac{1}{3}x + \frac{2}{3}$ definem retas delimitadoras da pré-imagem.

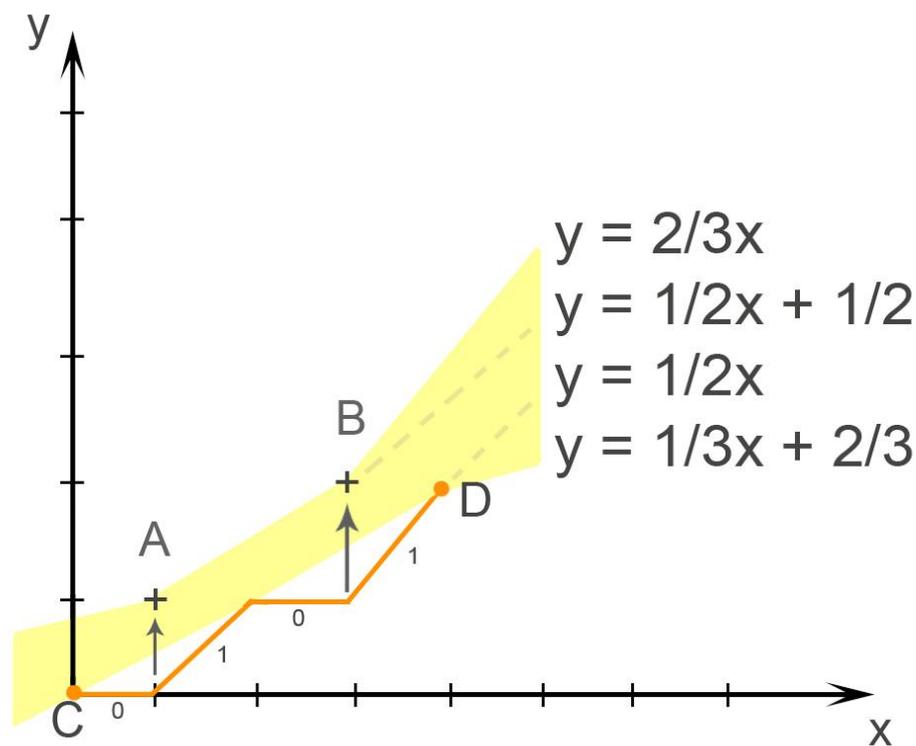


Figura 1.9: Segmento de linha reta digital em laranja com o código 0101 (*chain code* na vizinhança-8) e sua pré-imagem no plano (x,y) (área definida pelos pontos A, B, C e D da região em amarelo) (adaptado de Aghito e Forchhammer (2006)).

Definição 23 *Ruído:* dado um segmento de reta no espaço euclidiano e os correspondentes segmento digitalizado e pré-imagem, o ruído de interesse para esta pesquisa é toda influência espúria que, ao alterar o segmento digitalizado mencionado, afeta também a sua pré-imagem.

1.1.16 Distância entre segmentos digitais

Uma função de vizinhança (*neighborhood*), escolhida de modo a expressar como agrupar segmentos digitalizados em uma única estrutura linear por meio de uma métrica, viabiliza a classificação de segmentos digitalizados em próximos ou distantes. A Definição 24 apresenta um exemplo de distância entre segmentos necessária para conceituar essa função.

Definição 24 *Distância entre segmentos digitais: pode ser conceituada da seguinte maneira:*

1. Dado um segmento digitalizado, encontra-se a pré-imagem do mesmo;
2. Escolhe-se uma distância considerada mínima, por exemplo uma distância euclidiana mínima;
3. Encontra-se duas pré-imagens denominadas “pré-imagens delimitadoras” paralelas e separadas pela distância mínima da pré-imagem inicial obtida no item 1;
4. A região definida pelos contorno mais externos das pré-imagens delimitadoras definem uma região de aceitação;
5. Relativamente ao segmento digitalizado inicial, demais segmentos digitalizados serão classificados como próximos se as correspondentes pré-imagens se encontrarem no interior da região de aceitação. Serão classificados como distantes se estiverem fora total ou parcialmente da mencionada região.

1.1.17 Retitude

Pela propriedade da corda, pode-se definir uma linha reta digitalizada ideal conforme a seguir.

Definição 25 *Segmento de linha reta digitalizada ideal: é um conjunto de pixels, resultado da digitalização de um segmento de reta euclidiano particular, que satisfaz à propriedade da corda. Note que a propriedade da corda foi originariamente definida para a vizinhança-8, existindo especializações para a vizinhança-4 (RONSE, 1986). Todo segmento de reta digitalizada ideal é também um segmento de reta digital ideal.*

Definição 26 *Segmento de linha reta digital ideal: é um conjunto de pixels, não necessariamente o resultado da digitalização de um segmento de reta euclidiano particular, que satisfaz à propriedade da corda (RONSE, 1986).*

Ocorre que o processo de digitalização de uma linha reta euclidiana específica introduz o ruído, causando que o correspondente segmento de linha reta digitalizada não atenda à propriedade da corda, por resultarem em muitos segmentos curtos nas proximidades de um *pixel*, seguindo-se o especificada pela mencionada propriedade. Portanto, atendendo à Definição 24, é necessário algum tipo de medida, denominada métrica, apropriada para avaliar se dois segmentos de reta digitalizados pertencem a uma única estrutura linear (NACKEN, 1993). Por essa métrica, define-se uma função de vizinhança (*neighborhood*) escolhida de modo a expressar como agrupar segmentos de linha reta digitalizados em uma única estrutura.

Definição 27 *Retitude:* é uma possível tradução da palavra inglesa *straightness*, no sentido de indicar o grau em que um arco digital se aproxima de uma linha reta digitalizada ideal da Definição 25. A retitude de um arco digital é dada por uma métrica estabelecida, representada por tolerâncias com possíveis interpretações geométricas no plano. As tolerâncias controlam o quanto determinado parâmetro ou atributo pode se desviar daquele de uma linha reta digitalizada ideal.

Note que, pela Definição 7, um segmento digitalizado é um caso particular de arco digital, mas nem todo arco digital é um segmento digitalizado.

1.1.18 Hierarquia das linguagens

O entendimento do problema, sob o ponto de vista sintático, envolve os conceitos de linguagem, gramática e tipos de gramáticas. Segundo a hierarquia de Noam Chomsky datada de 1956, descrita em Lewis e Papadimitriou (1981), as linguagens são classificadas em quatro classes diferentes representadas na Figura 1.10, que são: Linguagens Enumeráveis Recursivamente (ou Tipo 0), Linguagens Sensíveis ao Contexto (ou Tipo 1), Linguagens Livres de Contexto (ou Tipo 2) e Linguagens Regulares (ou Tipo 3). Associa-se um grau de complexidade entre as classes mencionadas, em que a classe Tipo 3 é um subconjunto da classe Tipo 2, a classe Tipo 2 é um subconjunto da classe Tipo 1, e a classe Tipo 1 é um subconjunto da classe Tipo 0.

Um formalismo com poder para representar uma linguagem enumerável recursivamente pode representar qualquer linguagem menos complexa. Também, quanto mais abrangente a classe de linguagens, maior será a complexidade do formalismo necessário para representá-la, seja por uma gramática seja por um reconhecedor.

Essa hierarquia é importante para o problema desta pesquisa pois permite avaliar o poder computacional requerido para as modelagens.

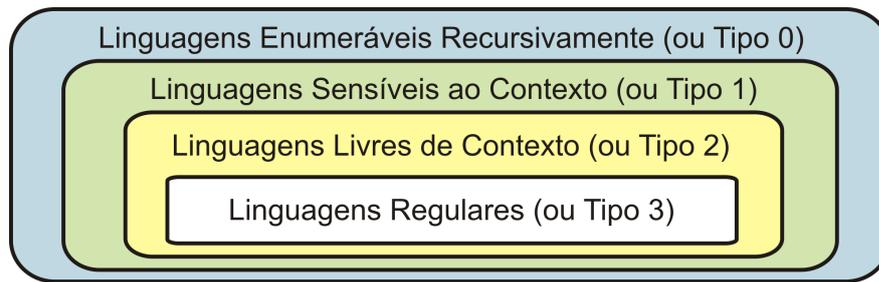


Figura 1.10: Hierarquia das linguagens (adaptado de Lewis e Papadimitriou (1981))

1.2 O segmento de linha reta digitalizada adaptativo

A hipótese inicial da abordagem deste trabalho é quanto à viabilidade da modelagem das propriedades das retas digitais por um conjunto de regras. Caso as interferências espúrias possam também ser modeladas, mesmo que indiretamente por intermédio de tolerâncias permitidas, a abordagem torna-se uma alternativa para que um dispositivo adaptativo represente as diferentes instâncias possíveis das estruturas. Este estudo se concentra nos segmentos digitalizados, os quais podem ser considerados como um caso particular de arco digital. Evidentemente, arcos quaisquer podem ser representados pela concatenação de segmentos digitalizados (YOU; FU, 1979).

Portanto, nesta tese, as interferências espúrias são modeladas indiretamente por intermédio de tolerâncias permitidas, utilizando uma propriedade da corda modificada:

Definição 28 *Propriedade da corda modificada:* diz-se que um arco digital C apresenta a propriedade da corda modificada se, para cada dois pontos digitais c e d pertencentes a C , e para cada ponto $p = (x, y)$ em \overline{cd} , existe um ponto $e = (h, k)$ pertencente a C tal que

$$\max\{|x - h|, |y - k|\} < n \text{ com } n \geq 1. \overline{cd} \text{ é o segmento de reta entre } c \text{ e } d;$$

enquanto n leva em conta o comprimento do segmento digitalizado.

Definição 29 *Segmento de linha reta digitalizada adaptativo:* (resumido segmento digitalizado adaptativo) pela propriedade da corda modificada da Definição 28, tendo em vista a vizinhança $\max\{|x - h|, |y - k|\} < n$ com $n \geq 1$, o segmento digitalizado adaptativo desta tese tem as seguintes características principais:

1. *Apresenta o poder expressivo de representar um conjunto de segmentos de arcos digitais que podem ser um segmento digitalizado ideal ou arco aproximadamente reto comparativamente a uma métrica indicadora de retitude da Definição 27, atendendo à propriedade da corda modificada;*
2. *Para um segmento digitalizado ideal ou um arco aproximadamente reto do item 1, representa os diferentes comprimentos desses segmentos ou arcos, teoricamente até infinito;*
3. *Além disso, representa um número ilimitado de retas euclidianas possíveis de serem digitalizadas para o conjunto de pontos digitais especificados pelo segmento digitalizado adaptativo.*

Contudo, os termos “arcos aproximadamente retos” ou “quase” retos são entendidos pela Definição 29 como linhas retas visualmente corretas dentro de uma tolerância, não necessariamente no tocante à propriedade da corda, mas sim dessa propriedade modificada. Portanto, no texto que se segue, a menos que de outra maneira seja especificado, a nomenclatura deste texto não faz nenhuma distinção entre um segmento digitalizado ideal e as linhas “quase” retas nas proximidades da primeira, considerando que todo o conjunto é reconhecido pelo autômato finito adaptativo que implementa o segmento digitalizado adaptativo.

Definição 30 *Vizinhança adaptativa: é uma região digital, representada por um segmento digitalizado adaptativo e variável com os estímulos, correspondente a arcos “aproximadamente retos” ou “quase” retos, entendidos como segmentos digitalizados dentro de uma tolerância, não necessariamente no tocante à propriedade da corda, mas sim dessa propriedade modificada.*

Definição 31 *Condições de contorno adaptativas: são condições relacionadas aos limites máximos das vizinhanças adaptativas da Definição 30 visando especificar as cadeias aceitas pelos autômatos finitos adaptativos que implementam os segmentos digitalizados adaptativos. Em outras palavras, as condições de contorno especificam a envoltória de uma região digital, variável conforme os estímulos.*

1.3 Breve histórico

Após a conjectura inicial de Freeman, a caracterização formal de segmentos digitalizados foi efetuada por Rosenfeld (1974), demonstrando que um arco digital é um segmento digitalizado se, e somente se, atender à propriedade da corda. Baseado nesse fato, provaram-se os dois primeiros critérios de Freeman e algumas propriedades relativas à regularidade na estrutura de um segmento digitalizado.

Ao considerar os possíveis comprimentos das corridas (*run lengths*) bem como das corridas de corridas (*runs of runs*) ao longo da direção principal (a tangente do ângulo principal do segmento com o eixo x), foi provado por Rosenfeld (1974) que só pode haver dois comprimentos possíveis diferentes dessas corridas, que são dois números inteiros consecutivos, e um deles sempre ocorre isoladamente. Este padrão de números consecutivos foi a primeira expressão matemática da terceira propriedade (*Prop3*) do modelo de Freeman.

Dentre os estudos subsequentes de outros autores (adaptado de Hung (1985)) destacam-se:

- A geração de segmentos digitalizados quando a sua inclinação, entendida como a tangente do ângulo principal do segmento com o eixo x , é um número racional, provando a existência de uma estrutura hierárquica nas retas digitais;
- A extensão do padrão de números consecutivos para inclinações (a tangente do ângulo do segmento com o eixo x) de números irracionais por Rosenfeld (1974), em todos os níveis da estrutura hierárquica de uma linha reta digital. Esse resultado foi considerado uma expressão mais definida do terceiro critério;
- A generalização do algoritmo do primeiro item ao incluir no formalismo inclinações de números irracionais;
- Hung (1985) demonstrou que a existência do padrão de números consecutivos, em todos os níveis da hierarquia, é, em geral, suficiente, mas não uma condição necessária para atender à propriedade da corda.

Avaliando-se os trabalhos que utilizam o método sintático, destacam-se:

- Com a especificação de linguagem sensível a contexto para as linhas retas numa grade, Rothstein e Weiman (1976) propõem um *polyautomaton* que reconhece

linhas retas visando computação paralela. A idéia era que cada célula da grade fosse representada por um autômato finito em condições de ser ativado ao ser atravessado por uma linha, e também com a possibilidade de troca de informações com as células vizinhas mais próximas. Ao se executar os algoritmos de cada célula de uma forma altamente paralela, a grade tornava-se um modelo de uma “retina”, as células se tornavam modelos de “neurônios” e as linhas se tornavam modelos de excitação visual. Com muitas linhas sendo manipulados ao mesmo tempo, obtém-se um tipo de autômato celular habilitado também a efetuar análises geométricas em diferentes topologias. Além disso, esse trabalho de Rothstein e Weiman (1976) apresentou uma correspondência entre a série de Farey e a tangente do ângulo de inclinação referente ao eixo x de retas passando pela origem;

- Brons (1974) efetuou uma avaliação de diferentes métodos linguísticos para descrição de uma linha reta em uma grade, enfatizando a comparação de gramáticas que geram as cadeias;
- Shlien (1983) apresentou um esquema de reconhecimento sintático de padrões, sem considerar o ruído, para segmentação de uma curva digital em linhas retas.

Tomando por base trabalhos como os de Chaudhuri e Pal (1997) e de Lee e Fu (1982), violou-se a propriedade da corda, introduzindo-se tolerâncias, tendo em vista a constatação da inviabilidade de satisfazê-la nos casos práticos, o que conduziu aos aprimoramentos mais recentes de Bhowmick e Bhattacharya (2007) e Bhowmick, Biswas e Bhattacharya (2008).

Entretanto, autores como Feschet (2008) comentam sobre as restrições existentes com segmentos digitalizados e linhas retas digitais, mesmo após os aprimoramentos de Bhowmick e Bhattacharya (2007), destacando-se as seguintes:

1. Tratam-se de estruturas demasiadamente rígidas, que não toleram o ruído;
2. Devido a essa rigidez, somente segmentos curtos podem ser construídos na prática;
3. Uma das consequências é a geração de grande quantidade de segmentos em arcos irregulares;
4. Outras consequências de tal rigidez são: não incorporam a capacidade de representar a escalabilidade, os desvios em ângulo ou os desvios em comprimento dos segmentos bem como as respectivas tolerâncias.

Em complemento, Feschet (2008) segue uma abordagem com base na geometria discreta aritmética, visando flexibilizar as estruturas, solucionando as restrições mencionadas nos itens 1, 2 e 3 acima.

Reveillès (1991) apud Fiorio, Jamet e Toutant (2006) fundamentou a geometria discreta aritmética ao apresentar a linha discreta como um subconjunto de Z^2 que satisfaz duas desigualdades diofantinas.

Considerando que as imagens digitais podem ser vistas como matrizes de pixels, ou seja, subconjuntos de Z^2 , a geometria discreta aritmética foi desenvolvida como uma geometria análoga à geometria euclidiana, direcionada para o espaço discreto Z^n (FIORIO; JAMET; TOUTANT, 2006) (ROUSSILLON; SIVIGNON; TOUGNE, 2007).

1.4 **Objetivo**

O objetivo deste trabalho é propor uma alternativa de representação de segmentos de linhas retas digitalizadas sob o enfoque adaptativo, utilizando o autômato finito adaptativo.

1.5 **Justificativa**

Andres, Acharya e Sibata (1997) enfatizam que há normalmente duas maneiras principais de representar objetos do mundo real em computação. A primeira é a representação analítica em que o objeto é representado por combinações de primitivas contínuas euclidianas, no sentido de não envolverem o processo de digitalização, e definidas analiticamente.

A segunda maneira é pelo processo de digitalização em que o objeto é representado por um conjunto de pontos discretos. Tais pontos são interligados por primitivas, entendidas como aproximações locais de primitivas contínuas; aproximação essa efetuada por meio do processo de digitalização. Este é o método de Freeman, considerado a abordagem clássica de representação digital de linhas retas.

A representação de linhas retas utilizando o processo de digitalização tem sido tradicionalmente utilizada com sucesso em diversas aplicações, apesar de dificuldades constatadas pela literatura em controlar propriedades das primitivas envolvidas, espe-

cialmente propriedades globais (ANDRES; ACHARYA; SIBATA, 1997).

Por outro lado, é intrinsecamente impossível obter uma medida precisa de determinadas propriedades globais, por exemplo, que envolvam o comprimento do segmento digitalizado. O melhor que se pode fazer é obter uma boa estimativa do comprimento correspondente, minimizando algum critério de erro especificado (DORST; SMEULDER, 1987).

A principal alternativa existente na atualidade ao método clássico (o modelo de Freeman) de representação é uma abordagem intermediária entre as duas maneiras de representação, a analítica e a discreta, denominada representação discreta-analítica com base na geometria discreta aritmética. Essa alternativa híbrida se fundamenta principalmente na representação analítica dos pontos discretos utilizando equações diofantinas com propósito de obter conceituações de retitude de caráter mais global, com maior controle das propriedades das primitivas do que o método clássico.

Enquanto o modelo de Freeman é especificado pelas propriedades *Prop1* a *Prop3* da Definição 20 juntamente com a propriedade da corda; a alternativa discreta-aritmética é centrada em equações diofantinas que definem uma espessura w denominada espessura aritmética. Pela Figura 1.11 a espessura aritmética w define uma região digital delimitada pelas retas $ax + by + \mu = 0$ e $ax + by + \mu = \omega$.

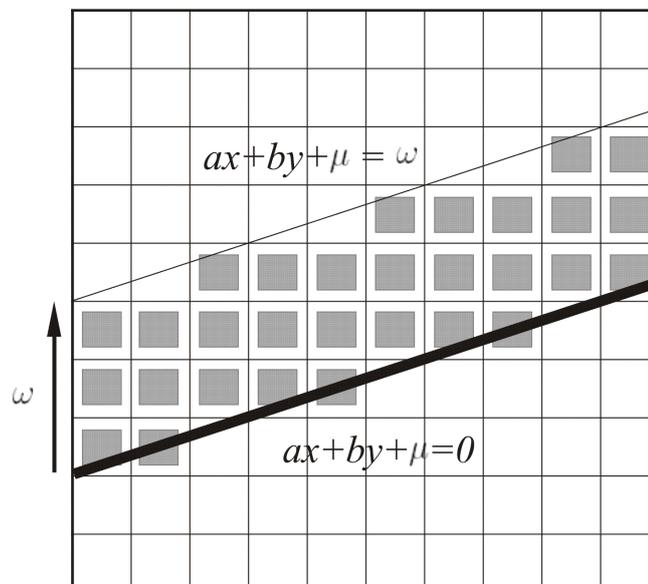


Figura 1.11: Interpretação geométrica da definição geral de linha discreta aritmética (adaptado de Figueiredo (1999)).

A conexão da geometria discreta aritmética com a geometria euclidiana ocorre no limite tendendo a infinito, da mesma maneira que uma grade discreta sendo observada

de um ponto suficientemente distante aparenta ser contínua (FIORIO; JAMET; TOUTANT, 2006). Ou seja, arcos irregulares observados de pontos suficientemente distantes podem se revelar, na verdade, como retas.

A representação adaptativa possibilita a alteração das escalas dos segmentos, de tal maneira que a retitude de arcos irregulares possa ser analisada em escalas compatíveis utilizando vizinhanças adaptativas definidoras de regiões digitais variáveis com os estímulos. A adaptatividade surge, portanto, como uma alternativa para incorporar os fundamentos da geometria discreta aritmética ao modelo de Freeman.

1.6 Organização da tese

No capítulo 1, apresentam-se conceitos iniciais, a motivação, os objetivos e a visão geral da tese, inclusive a sua organização.

No capítulo 2, apresentam-se conceitos adicionais que embasam esta pesquisa, compreendendo uma breve revisão sobre a tecnologia adaptativa, uma descrição de particularidades do processo de digitalização tal como a perda de informação que ocorre no processo, uma exposição do poder computacional requerido para a análise sintática de segmentos digitalizados, e um panorama do estado-da-arte.

No capítulo 3, avaliam-se os segmentos digitalizados partindo-se de suas propriedades de ângulo de orientação e comprimento de segmentos.

No capítulo 4, descrevem-se as estruturas adaptativas propostas por este trabalho, bem como os algoritmos adotados e detalhes de implementação dos segmentos digitalizados adaptativos.

No capítulo 5, descrevem-se implementações e testes referentes à tese de vizinhança adaptativa, apresentando uma síntese dos conceitos principais envolvidos, bem como possíveis restrições. Propõe-se também a arquitetura de um sistema adaptativo incorporando escalas adaptáveis para atuação em cenários que envolvam arcos complexos, ou seja, arcos passíveis de serem segmentados em vários segmentos digitalizados.

No capítulo 6, descrevem-se as simulações complementares realizadas, separadas em estudos de caso, com os detalhes de implementação apresentados gradativamente, do mais simples ao mais complexo.

No capítulo 7, apresentam-se as considerações finais deste texto, contribuições da tese e sugestões de trabalhos futuros.

Finalmente, apresentam-se, no final deste texto, as referências bibliográficas empregadas para o desenvolvimento da pesquisa e os apêndices A, que apresenta a metodologia adotada nesta pesquisa, B contendo o código de um autômato finito adaptativo e C com a lista de definições da tese.

2 CONCEITOS

Este capítulo apresenta fundamentos adicionais necessários para o entendimento do assunto estudado por esta tese. Inicialmente, neste capítulo descrevem-se fundamentos sobre os autômatos finitos adaptativos. Posteriormente, a partir de trabalhos obtidos na literatura específica, comenta-se sobre o processo de digitalização nos seus aspectos mais característicos, abrangendo os conceitos principais de retas. Finalmente, este capítulo apresenta um tópico conclusivo sobre as justificativas e os esclarecimentos requeridos para o desenvolvimento lógico do trabalho.

2.1 Autômatos finitos adaptativos

A Definição 32, afeta às sequências de estados conexos (abreviado sequências conexas), facilita as descrições dos autômatos desta pesquisa. Outras definições particulares aos demais assuntos, precedem cada tópico. Na terminologia sintática as sequências conexas são denominadas pelo termo *head-to-tail concatenations*, de acordo com a página 3 de Fu (1974).

Definição 32 *Sequência de estados conexos: (abreviado sequência conexa), é um conjunto de estados em sequência concatenada e interligados por transições que consomem o mesmo símbolo em toda a sequência. Apenas dois estados, denominados extremos, se conectam a um único estado do conjunto, enquanto os demais estados se conectam a dois estados. Para identificar uma determinada sequência conexa nas descrições basta utilizar o seu primeiro estado. Por exemplo, na Figura 2.1 há duas sequências conexas: uma partindo do estado l e outra do estado k .*

As sequências conexas permitem uma notação representativa dos estados da mesma da seguinte maneira: o símbolo indicativo da sequência indexado pela ordem do es-

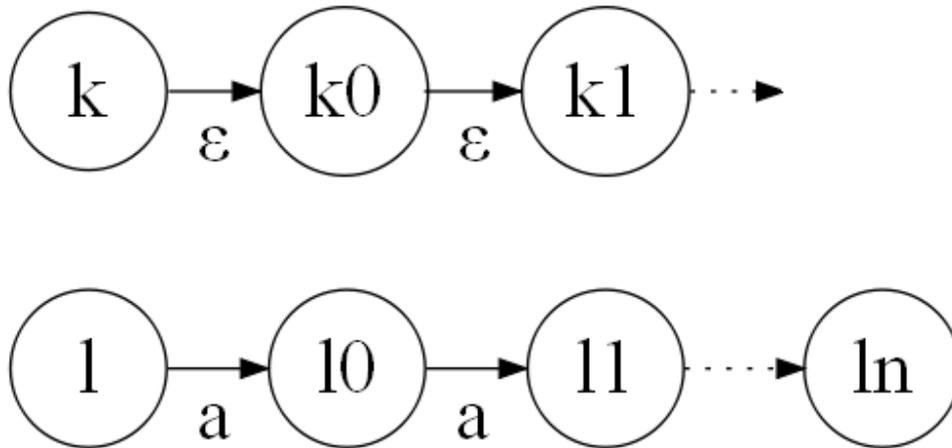


Figura 2.1: Dois exemplos de seqüências conexas de estados: uma com transições que consomem o símbolo a e outra com transições em vazio. O último estado da seqüência com transições em vazio é indeterminado.

tado na seqüência. Por exemplo, uma seqüência conexa com símbolo indicativo r é concatenada na seqüência $r, r_1, r_2, r_3, \dots, r_i$.

Os dispositivos tratados nesta seção são formalizados por conjunto de regras, sendo que a característica de um dispositivo adaptativo é a capacidade de alterar o seu conjunto de regras no decorrer da leitura dos estímulos de entrada, partindo de um conjunto de regras inicial.

Conforme a Expressão $AFA = (ND_0, AM)$, o autômato finito adaptativo é um dispositivo formalizado por conjunto de regras composto de duas camadas. A primeira camada ND_0 , denominada camada subjacente, é um autômato finito, ou seja o dispositivo guiado por regras e não-adaptativo conforme Neto (2001): “A (non-adaptive) rule-driven device is any formal machine whose behavior depends exclusively on a finite set of rules which map each possible configuration of the device into a corresponding next configuration”.

A segunda camada, AM , é definida por um conjunto de “ações adaptativas”, que modificam o conjunto de regras inicial por meio de remoções de regras e inserções de novas regras.

ND_0 caracteriza a configuração inicial do autômato finito adaptativo. Dependendo do estímulo s_k da Expressão $S: s_i; i = 1, 2, \dots, n$, a configuração ND_{k-1} do autômato finito adaptativo é alterada pelas ações adaptativas, resultando que o autômato finito ND_{k-1} é modificado para outro autômato finito ND_k pertencente ao conjunto da Expressão 2.1

de todas as possíveis configurações do autômato:

$$\{ND_0, ND_1, ND_2, ND_3, \dots, ND_k : k \geq 0\}. \quad (2.1)$$

2.1.1 Sumário sobre o autômato finito

Os autômatos finitos classificam-se em:

1. Autômatos Finitos Determinísticos: Quando cada passo da computação segue um caminho que é único, no sentido de que para cada configuração e estímulo há um única configuração seguinte possível;
2. Autômatos Finitos Não-Determinísticos: Quando há mais de um caminho possível na passagem de uma configuração para a seguinte.

Apesar das classes do autômato finito descritas acima nos itens 1 e 2, todas essas reconhecem a mesma classe de linguagens regulares, sendo que todo autômato finito não-determinístico pode ser convertido no seu equivalente determinístico. Existe também um subconjunto dos autômatos finitos não determinísticos, denominada autômatos finitos com movimentos vazios, incluindo transições em vazio ϵ .

De acordo com Carroll e Long (1989), o autômato finito é um formalismo matemático de uma máquina que aceita determinado conjunto de palavras sobre um alfabeto Σ . Os autômatos finitos são apropriados sempre que houver um número finito de entradas e apenas um número finito de situações a serem distinguidas pela máquina, tal que, para um dado estado corrente do autômato finito e uma dada sequência de estímulos, a função de transição determina o estado seguinte do autômato.

Formalmente, um autômato finito não determinístico ou reconhecedor finito não determinístico é uma quintupla (CARROLL; LONG, 1989):

$$AF = (Q, \Sigma, \delta, q_o, F). \quad (2.2)$$

Onde:

- Q é um conjunto finito não vazio de estados;
- Σ é o alfabeto de entrada (um conjunto finito não vazio de símbolos);
- q_o é o estado de partida, (ou inicial), um elemento pertencente a Q ;

- δ é a relação de transição entre estados $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow Q$;
- F é o conjunto de estados finais, um subconjunto de Q , $F \subseteq Q$.

Em um conjunto X qualquer, $\|X\|$ denota a cardinalidade do conjunto, ou seja, o seu tamanho entendido como o número de elementos em X . No conjunto total de estados do autômato finito denominado Q , $\|Q\|$ é o seu tamanho, em que se destacam um único estado inicial, e um, ou vários estados finais. O consumo pelo autômato finito dos estímulos da cadeia de entrada resulta em duas possibilidades: *i*) Conduzir o autômato finito a um dos estados finais (neste caso a cadeia de entrada foi “aceita”); *ii*) Ou “rejeitar” a cadeia, se um estado final não for atingido, ou não ocorrer o consumo de toda a cadeia de entrada.

A relação de transição δ é composta de regras do tipo $(x, i) \rightarrow y$ onde x é o estado atual do autômato antes da transição entre estados; y é o estado de destino do autômato após a transição; i é o estímulo de entrada pertencente a Σ .

2.1.2 Sumário sobre o formalismo do autômato finito adaptativo

Este tópico apresenta um resumo de Neto (1993), Neto (2000) e Neto (2001) adaptado ao autômato finito adaptativo. Com propósito de manter fidelidade ao formalismo original, alguns parágrafos são apresentados em inglês por ter sido o idioma de Neto (2000), e determinadas palavras traduzidas para o português apresentam a original correspondente em inglês entre parêntesis.

2.1.2.1 Ação adaptativa

Definição 33 *Ação adaptativa:* ações adaptativas são responsáveis por alterar a estrutura do dispositivo não adaptativo subjacente dinamicamente em resposta aos estímulos:

Uma ação adaptativa corresponde à chamada de uma função adaptativa, e é dada por um par ordenado (\mathcal{F}, π) onde:

- \mathcal{F} representa o nome da função adaptativa correspondente à ação adaptativa em questão;
- π simboliza uma sequência, eventualmente vazia, de argumentos ρ_1, ρ_2, \dots onde os ρ_i correspondem posicionalmente aos parâmetros formais ϕ_1, ϕ_2, \dots da função adaptativa \mathcal{F} , e designam valores a serem utilizados pela função adaptativa em substituição aos correspondentes parâmetros formais para a composição de transições adaptativas a serem inseridas, consultadas ou eliminadas do autômato. Os argumentos ρ_i

podem assumir quaisquer valores compatíveis com o papel que representam nas produções adaptativas em que são referenciados (NETO, 1993).

Assim, as ações adaptativas são implementadas por meio de funções adaptativas, as quais determinam as alterações a serem realizadas na camada subjacente, quando do acionamento das mencionadas ações. Pode-se interpretar as ações adaptativas como chamadas de função, a função adaptativa, em que esta pode ser paramétrica.

2.1.2.2 Função adaptativa

Na Definição 33, \mathcal{F} é o nome da função adaptativa, a qual pode ser paramétrica com parâmetros formais $\phi_1, \phi_2 \dots \phi_i$. Apesar de opcionais, se parâmetros forem especificados, terão de ser informados para ativar a correspondente função adaptativa F .

Definição 34 *Função adaptativa:* em termos aproximados, funções adaptativas são coleções de ações adaptativas elementares a serem aplicadas ao conjunto de transições do autômato ("[...] adaptive functions, which may be roughly regarded as collections of elementary adaptive actions to be applied to the transition set of the automaton" (NETO, 2000).

Ou seja, pela Definição 34, o cerne de uma função adaptativa é composto de listas de ações adaptativas elementares aplicadas ao conjunto de transições do autômato.

2.1.2.3 Transição adaptativa genérica

A Expressão 2.3 representa a forma geral de uma regra no autômato finito adaptativo

$$(q_1, i) : R \rightarrow q_2 : S \quad (2.3)$$

onde q_1 é o estado de origem do autômato antes da transição entre estados, q_2 é o estado de destino do autômato após a transição, i é o estímulo de entrada, R e S são funções adaptativas.

O lado esquerdo da Expressão 2.3 refere-se à configuração do autômato adaptativo antes da transição de estados, enquanto o seu lado direito codifica a sua configuração após a transição de estados.

Numa transição adaptativa, as ações adaptativas são especificados opcionalmente, tal que o lado esquerdo da Expressão 2.3 representa uma modificação a ser aplicado (*to be applied before*) antes da transição de estados, enquanto o lado direito especifica as mudanças a serem impostas ao autômato após a transição (*changes to be imposed to the automaton after the transition*).

A Figura 2.2 mostra uma representação gráfica estática de uma transição genérica do autômato finito adaptativo.

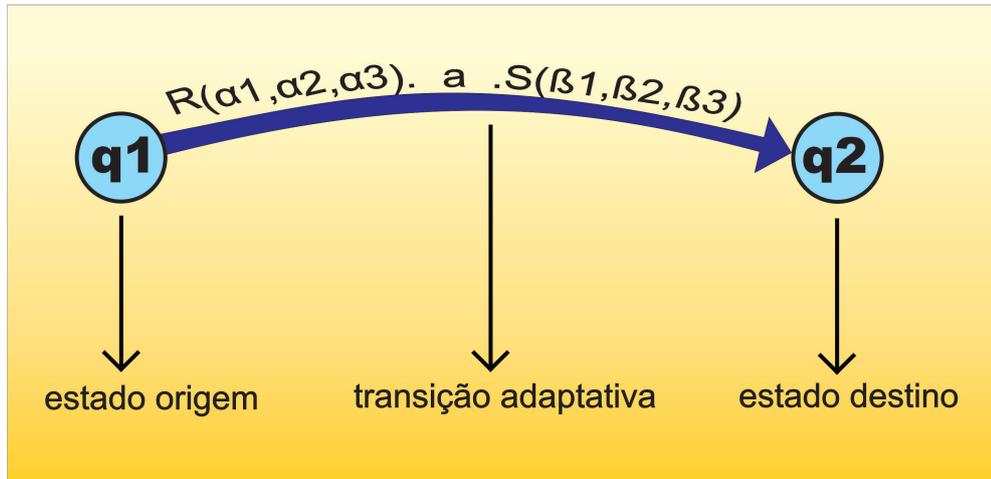


Figura 2.2: Uma transição adaptativa genérica, onde R e S são funções adaptativas paramétricas opcionais dos tipos *anterior* e *posterior* respectivamente (adaptado de Neto (2001)).

Assim, nos moldes da Figura 2.2, uma determinada função adaptativa R é representada graficamente por $R\bullet$, caso seja do tipo *anterior*, a ser ativada antes da transição. Similarmente, uma função adaptativa S é representada graficamente por $\bullet S$, caso seja do tipo *posterior*, a ser ativada após a transição de estados.

2.1.2.4 Ação adaptativa elementar

Definição 35 *Ação adaptativa elementar:* é a única operação de edição do formalismo, aplicada ao conjunto de transições do autômato. Três operações diferentes são permitidas com as ações adaptativas elementares: inspeção, remoção e inserção de transições (“[...] elementary adaptive actions to be applied to the transition set of the automaton are the only editing operations supplied by the formalism. Three different operations are allowed: inspection, deletion and insertion of transitions” (NETO, 2000)).

Pela Definição 35, existem três modalidades de ações adaptativas elementares que se-

guem o formato da Expressão 2.4

$$\otimes[(q_1, i) : R \rightarrow q_2 : S]. \quad (2.4)$$

Nesse formato da Expressão 2.4, a regra entre colchetes $(q_1, i) : R \rightarrow q_2 : S$ é denominada gabarito. Note que este texto utiliza o termo “gabarito”, evitando-se os termos “padrão ou “pattern” (constantes de Neto (2000)) os quais se referem nesta Tese a menções à área de Reconhecimento de Padrões (*Pattern Recognition*).

Na Expressão 2.4, as três modalidades de ações adaptativas elementares ocorrem pela substituição do operador \otimes por $?$, $-$ ou $+$, indicativos das modalidades de inspeção, remoção e inserção, respectivamente. Essas modalidades estão apresentadas no Quadro 1.

Significado	Simbologia
Consulta	$?[(q_1, i) : R \rightarrow q_2 : S]$
Remoção	$-[(q_1, i) : R \rightarrow q_2 : S]$
Inserção	$+[(q_1, i) : R \rightarrow q_2 : S]$

Quadro 1: Simbologia para as três modalidades de ações adaptativas elementares com o gabarito de cada ação elementar especificado entre colchetes (adaptado de Neto (2000))

Para as ações adaptativas elementares de consulta e de remoção do Quadro 1 ocorre um processo de busca apresentado na Definição 36 (“*Note that both inspection and deletion elementary adaptive actions search the current set of transitions for any transition matching the given pattern.*” (NETO, 2000)).

Definição 36 *Processo de busca:* o processo de busca afeto às ações adaptativas elementares é discriminado da seguinte forma:

- para ações de consulta: ocorre uma busca (“search”) no conjunto corrente de transições (*current set of transitions*) por qualquer regra que se iguale (“any transition matching the given pattern”) com o dado gabarito (“a transition having the shape enclosed in brackets”) entre colchetes, sem alterar o conjunto de regras;
- para ações de remoção: ocorre uma busca (“search”) no conjunto corrente de transições (“current set of transitions”) por qualquer transição que se iguale (“match”) com o dado gabarito (“a transition having the shape enclosed in brac-

kets”) entre colchetes, removendo as transições que se igualem do conjunto de regras;

- *para ações de inserção: inserem uma nova transição no conjunto de regras, aquela especificada pelo gabarito (adaptado de Neto (2000)).*

Observe-se que, havendo mais de uma ação adaptativa elementar a ser aplicada, independentemente da ordem em que foram declaradas, as consultas apresentam a prioridade máxima. Em seguida são aplicadas as remoções, e posteriormente as inserções. Possíveis transições em vazio são aplicadas por último, após as inserções.

O formalismo apresenta também os conceitos (*the concept of generator*) de variáveis e geradores.

2.1.2.5 Formato das funções adaptativas

As funções adaptativas podem ser especificadas por variáveis e geradores, também opcionais, com os significados seguintes:

Definição 37 *Nomes de variáveis:* “variables used in place of any of the components of the elementary adaptive action are assigned the actual corresponding values in the matching transition” (NETO, 2000). Variáveis são “símbolos associados a elementos cujos valores são desconhecidos no instante da chamada da função, e que são preenchidos, uma única vez durante cada execução da função adaptativa, como resultado de ações adaptativas de consulta” (ver página 63 de Neto (1993)).

Definição 38 *Nomes de geradores:* “generators are used to assign names to newly created states” (NETO, 2000). Os geradores fazem parte de um conjunto que “inclui estados que não são referenciados na configuração inicial do autômato mas que poderão ser eventualmente incluídos posteriormente” (PISTORI, 2003). São “variáveis especiais que são automaticamente preenchidas a cada chamada da função adaptativa, com valores novos, não utilizados pelo autômato até esta ocasião, permanecendo com este valor durante toda a execução da função” (ver página 63 de Neto (1993)).

Pelas Definições 37 e 38 um entendimento sobre variáveis e geradores é o seguinte:

- Variáveis são identificadores que recebem um valor de acordo com o resultado de ações adaptativas elementares de consulta ou exclusão. As variáveis são utilizadas para representar entidades existentes em transições do autômato finito adaptativo tais como estados ou estímulos

$$\begin{aligned} &\eta(\alpha, \beta) \{ ger_1^*, var_1 : \\ &?[(r_{i-1}, b) \rightarrow \beta] \\ &+[(ger_1, a) \rightarrow \alpha] \\ &-[(var_1, \varepsilon) : A \rightarrow r_{i+1}] \} \end{aligned}$$

Por exemplo, η pode ser ativada por uma transição $(1, a) : \eta(2, 6) \rightarrow 2$. Nesse caso, a função adaptativa η é ativada antes que o autômato finito adaptativo mude do estado 1 para o estado 2, desde que um *token* a seja recebido.

Da mesma forma como a função adaptativa η é ativada nesse exemplo, pela maneira que as funções adaptativas são conectadas às transições do autômato finito adaptativo define-se o comportamento do autômato resultando em aceitar ou rejeitar a cadeia de entrada.

2.1.2.6 Definição formal autômato finito adaptativo

Um autômato finito adaptativo é representado pela 7-tupla da Expressão 2.5

$$AFA = (Q, AR_0, \Sigma, q_0, F, \mathcal{B}, \mathcal{A}); \quad (2.5)$$

tal que:

Q é um conjunto não vazio e possivelmente infinito, de estados do autômato;

AR_0 é um conjunto de ações adaptativas: $AR_0 \subseteq \mathcal{B} \times (Q \times (\Sigma \cup \{\varepsilon\})) \times \mathcal{A}$;

Σ é o alfabeto de entrada do autômato. É um conjunto não vazio contendo número finito de símbolos;

q_0 é o estado inicial do autômato;

F é o conjunto de estados finais;

\mathcal{B} e \mathcal{A} são conjuntos de compostos de funções adaptativas anteriores e posteriores, respectivamente.

2.1.3 Técnicas adaptativas básicas

Um exemplo recorrente nos trabalhos sobre adaptatividade é o autômato que aceita cadeias pertencentes à linguagem $L_1 = \{a^n b^n c^n d^n : n \geq 1\}$ porque L_1 é sensível ao contexto, fora do poder computacional do autômato finito.

O autômato finito adaptativo da Figura 2.3 e Quadro 2 aceita uma linguagem C definida pela composição de duas linguagens L_1 e L_2 :

- $L_1 = \{a^n b^n c^n d^n : n \geq 1\}$;
- $L_2 = \{a^n b^m c^n d^m : n, m \geq 1, n \neq m\}$.

L_1 e L_2 são discriminadas pelos estados finais do autômato. As cadeias pertencentes a C são do tipo $a^i b^j c^v d^z$ definidas como segue:

$$C(a^i b^j c^v d^z) = \begin{cases} \forall i, j, v, z \geq 1 : \\ i = j = v = z : \text{aceitar, estado final SQ;} \\ i = v, j = z, i \neq j, v \neq z : \text{aceitar, estado final RT;} \\ \text{se não: rejeitar a cadeia.} \end{cases}$$

As cadeias de C são compostas pela concatenação de quatro sub-cadeias, uma para cada símbolo a , b , c e d respectivamente. Essa linguagem é reconhecida em quatro etapas pelo autômato finito adaptativo da Figura 2.3; a primeira etapa para *tokens* a , a seguinte para b , e assim sucessivamente. O autômato finito adaptativo compara os comprimentos de cada sub-cadeia, atingindo estados final SQ ou RT , e rejeitando as demais cadeias.

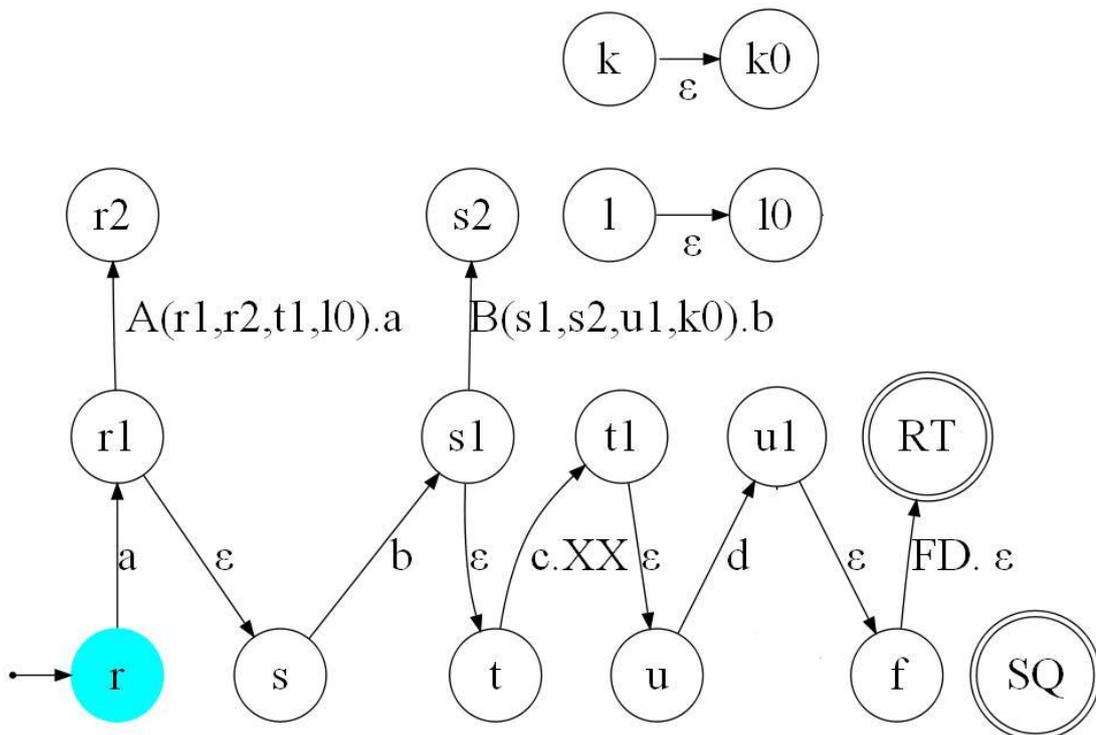


Figura 2.3: Configuração inicial de autômato finito adaptativo.

$ \begin{aligned} &A(\phi, \varphi, \gamma, \eta)\{g1^*, g2^*, g3^* : \\ &- [(\phi, \varepsilon) \rightarrow s] \\ &+ [(\varphi, \varepsilon) \rightarrow s] \\ &+ [(\varphi, a) : A(\varphi, g1, g2, g3) \rightarrow g1] \\ & \\ &- [(\gamma, \varepsilon) \rightarrow u] \\ &+ [(g2, \varepsilon) \rightarrow u] \\ &+ [(\gamma, c) \rightarrow g2 : XX] \\ & \\ &+ [(\eta, \varepsilon) \rightarrow g3] \} \\ & \\ &XX\{var1, var2 : \\ &- [(k, \varepsilon) \rightarrow var1] \\ &- [(var1, \varepsilon) \rightarrow var2] \\ &+ [(k, \varepsilon) \rightarrow var2] \\ &\} \end{aligned} $	$ \begin{aligned} &B(l, \kappa, \lambda, \mu)\{v1, v2, \\ &g1^*, g2^*, g3^* : \\ &- [(l, \varepsilon) \rightarrow t] \\ &+ [(\kappa, \varepsilon) \rightarrow t] \\ &+ [(\kappa, b) : B(\kappa, g1, g2, g3) \rightarrow g1] \\ & \\ &- [(\lambda, \varepsilon) \rightarrow f] \\ &+ [(g2, \varepsilon) \rightarrow f] \\ &+ [(\lambda, d) \rightarrow g2] \\ & \\ &- [(l, \varepsilon) \rightarrow v1] \\ &- [(v1, \varepsilon) \rightarrow v2] \\ &+ [(l, \varepsilon) \rightarrow v2] \\ & \\ &+ [(\mu, \varepsilon) \rightarrow g3] \} \\ & \\ &FD(l_0, k_0)\{ : \\ &- [(f, \varepsilon) : FD \rightarrow RT] \\ &+ [(f, \varepsilon) : FD \rightarrow SQ] \\ &\} \end{aligned} $
---	--

Quadro 2: Funções Adaptativas do autômato finito adaptativo da Figura 2.3

O Quadro 2 apresenta o algoritmo correspondente ao autômato finito adaptativo da Figura 2.3 por meio das funções adaptativas. A configuração inicial do autômato finito adaptativo da Figura 2.3 está preparada para consumir a cadeia $abcd$, sem ativar funções adaptativas. Para outras cadeias de $a^i b^j c^v d^z$, com a ativação da função adaptativa A, o autômato consome os símbolos a ; enquanto o autômato passa a consumir os símbolos b com a ativação da função adaptativa B.

O autômato finito adaptativo constrói três sequências conexas por meio de ativações da função adaptativa A: a primeira iniciando no estado r_2 , a outra partindo do estado t_1 , e a última partindo do estado l . Por essa notação, os estados de cada sequência conexa apresentam um mesmo símbolo, indexado pela sua ordem na sequência. Por exemplo, a sequência conexa que inicia no estado r é concatenada na sequência $r, r_1, r_2, r_3, \dots, r_i$.

Portanto, à medida que os *token a* estão sendo consumidos, a concatenação de estados r_{i+1} gera uma sequência conexa. A outra sequência conexa, do estado t_1 , é construída pela ativação da função adaptativa A com a finalidade de ser um modelo (*template*) da sub-cadeia referente aos *tokens c*. Note que é introduzida na configuração do autômato a função adaptativa XX, a qual é ativada a cada token c consumido com a finalidade de reduzir a sequência conexa que inicia no estado k .

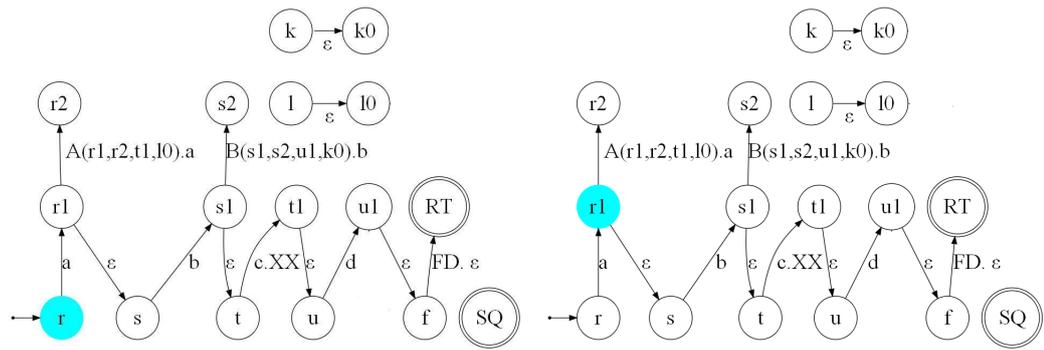
Da mesma forma, a cada token b consumido, três sequências conexas são construídas

pela ativação da função adaptativa B da Figura 2.3: uma partindo do estado s_2 , a outra do estado u_1 , e a seguinte do estado k . Após o autômato consumir algum token b , a sequência conexa do estado s_2 permite que o autômato finito adaptativo esteja preparado para potenciais estímulos b . A sequência conexa do estado u_1 é o *template* da sub-cadeia dos *tokens* d .

O autômato finito adaptativo compara os comprimentos das sub-cadeias dos *tokens* a e b por meio da sequência conexa do estado l ; enquanto a comparação dos comprimentos das sub-cadeias dos *tokens* b e c é efetuada por meio da sequência conexa do estado k . A sequência conexa do estado l , construída a cada ativação da função adaptativa A, é reduzida à medida que os *tokens* b estão sendo consumidos pelo autômato finito adaptativo. Analogamente, a sequência conexa de k é construída pela ativação da função adaptativa B, e reduzida pela ativação da função adaptativa XX. A função adaptativa XX é ativada por transições com estímulo c .

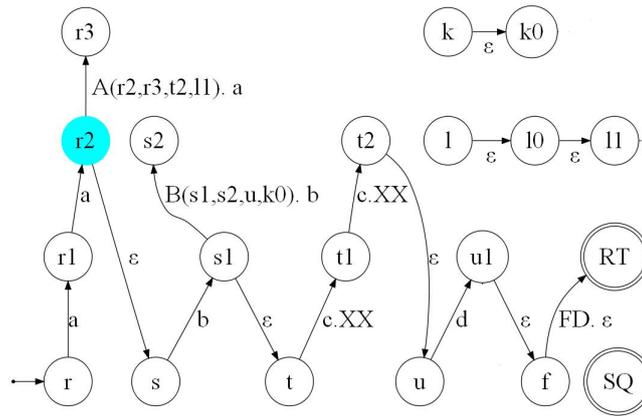
Na Figura 2.3, após o consumo dos *tokens* d , supondo que o autômato finito adaptativo alcance o estado f , duas situações podem ocorrer por existir apenas uma transição em vazio entre f e RT, ambas possibilitando que o autômato aceite a sequência. A primeira, na situação em que $i = j = v = z$, as transições $(l, \varepsilon) \rightarrow l_0$ e $(k, \varepsilon) \rightarrow k_0$ ainda permanecem na configuração, ativando a função adaptativa FD tendo em vista que seus parâmetros l_0 e k_0 existem. Pela ativação da função adaptativa FD, o estado final do autômato finito adaptativo é alterado para SQ. Não ocorrendo a primeira, a segunda situação é para o caso $i = v, j = z \neq i$ em que o autômato atinge o estado final RT.

A título ilustrativo, as Figuras 2.4(a) até 2.7(b) a seguir mostram as diferentes configurações do autômato descrito, que reconhece a linguagem C , para uma cadeia de entrada $aabbccdd$. O estado corrente aparece colorido em azul nessas figuras.



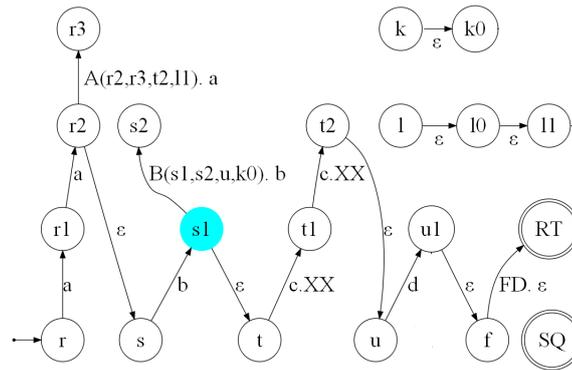
(a) Configuração inicial.

(b) Configuração após primeiro *token a* de entrada.

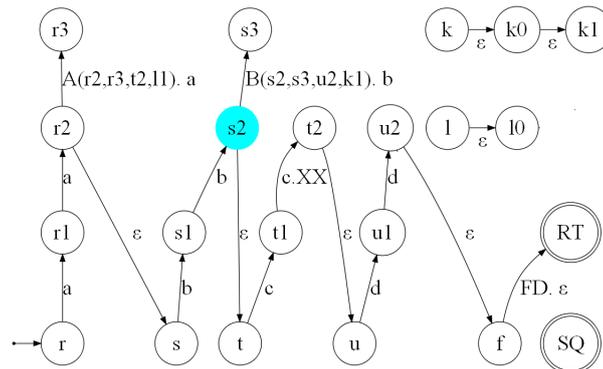


(c) Configuração após *tokens aa*.

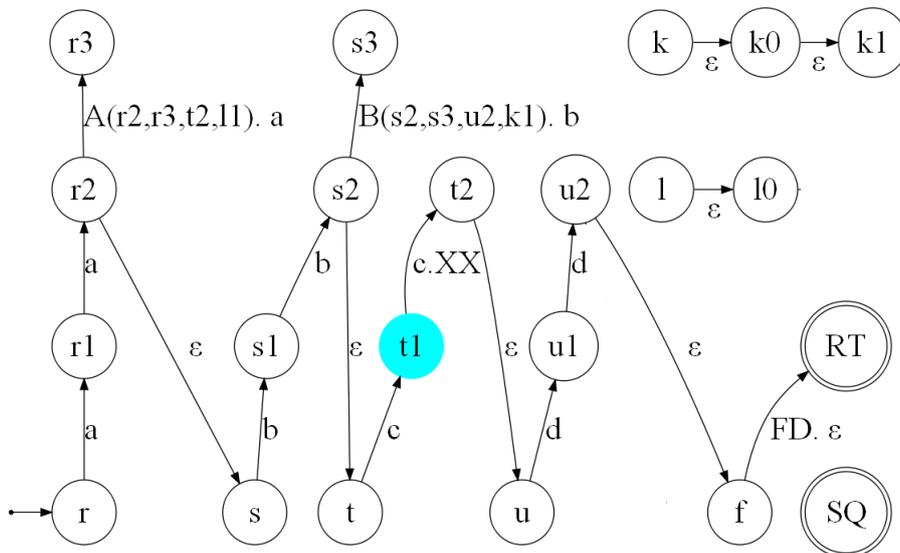
Figura 2.4: Três configurações de autômato finito adaptativo para cadeia de entrada *aabbccdd*.



(a) Configuração após *aab*

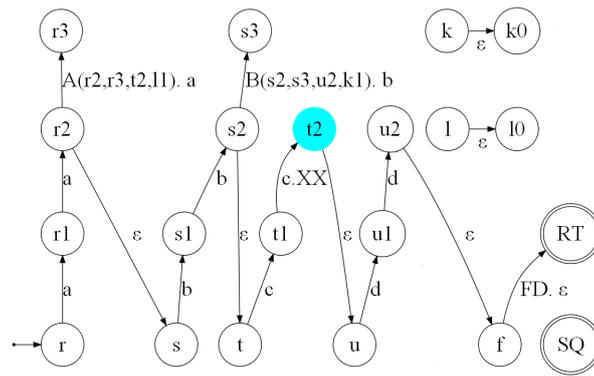


(b) Configuração após *aabb*.

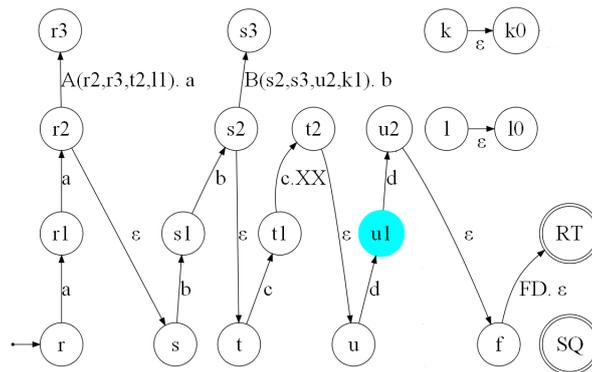


(c) Configuração após *aabbc*.

Figura 2.5: Apresentação de três configurações de autômato finito adaptativo, após a configuração da Figura 2.4(c).

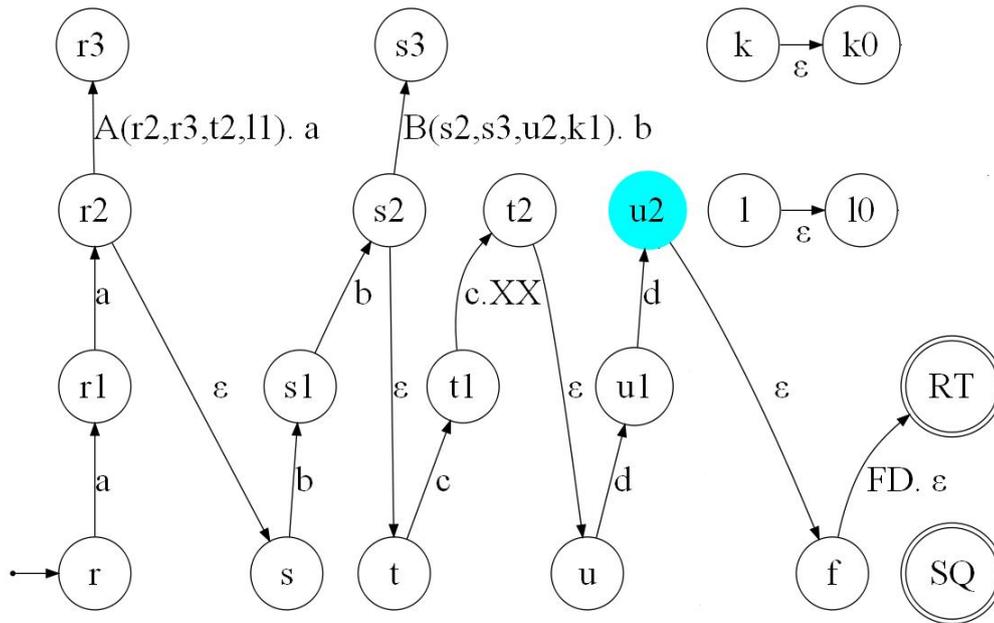
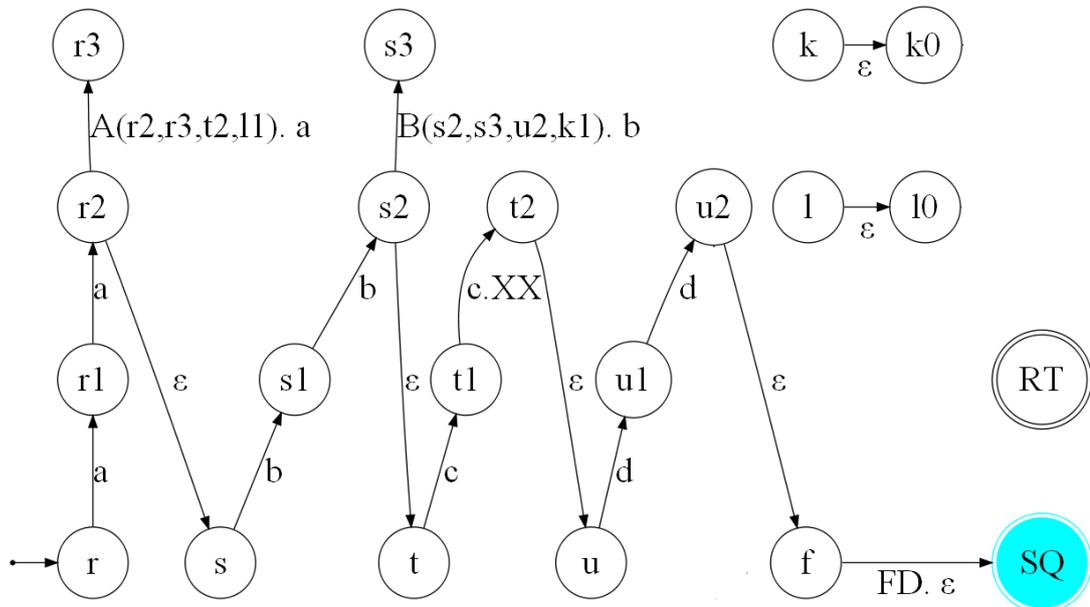


(a) Configuração após *aabbcc*.



(b) Configuração após *aabbccd*.

Figura 2.6: Apresentação de duas configurações de autômato finito adaptativo após configuração da Figura 2.5(c)

(a) Configuração após o $aabbccdd$.

(b) Configuração final.

Figura 2.7: Apresentação de duas configurações de autômato finito adaptativo após configuração da Figura 2.6(b). Note que a função adaptativa FD é ativada na passagem da configuração da Figura 2.7(a) para 2.7(b), alterando o estado final para SQ , pois as quantidades de símbolos a e b na cadeia de entrada são iguais.

2.1.4 Considerações sobre os autômatos finitos adaptativos

O conjunto de transições do autômato finito adaptativo é alterado dinamicamente conforme os estímulos de entrada (a variação no conjunto de estados é circunstancial de acordo com as diferentes topologias do autômato). A metodologia desta pesquisa pro-

cura interpretar esses estados e transições, por exemplo, por intermédio de grafos, resultando que os autômatos finitos adaptativos podem ser subdivididos em blocos especializados, sem que haja a necessidade de se provar o seu funcionamento repetidamente.

Esses blocos atuam como modelos funcionais de máquinas de estados elementares, que, uma vez validados, podem ser aplicados em qualquer reconhecedor desde que sejam introduzidos pequenos ajustes. O conjunto M , de estados e transições do autômato finito adaptativo, pode ser particionado em blocos M_i , $0 < i < n + 1$, pertencentes a $\{M_1, M_2, \dots, M_n\}$. Normalmente, o particionamento neste trabalho foi efetuado por critérios funcionais pelas aplicações controladas pelos blocos.

Por exemplo, no autômato finito adaptativo que reconhece a linguagem C apresentado no item 2.1.3, as sequências conexas dos estados k e l juntamente com as transições que ativam a função adaptativa FD atuam como um bloco contador que compara a quantidade de *tokens* a e b na cadeia de entrada, sendo que outros contadores serão especializações desse. Assim, um bloco contador é um dos blocos de M que compara quantidades de símbolos.

Uma sequência básica para a pesquisa e experimentação de autômatos finitos adaptativos é a seguinte:

- Identificar os algoritmos envolvidos.
- Dividir o projeto em blocos de acordo com as estruturas computacionais acessadas, tais como sequências conexas e *loops*.
- Modelar e codificar a máquina não-adaptativa de cada bloco.
- Codificar as funções adaptativas para acesso às estruturas, conforme o algoritmo.
- Integrar as funções adaptativas a uma ou mais cópias da máquina não-adaptativa para atuação com cada estrutura.
- Teste das unidades, avaliando os algoritmos.
- Integração geral dos blocos e teste global.
- Documentar o processo.

2.2 Características da representação digital

Em sistemas de reconhecimento de padrões, a primeira etapa é a digitalização abrangendo o processamento de imagens, ou seja, o pré-processamento, o tratamento das imagens, e segmentação, seguido pela extração de atributos. A etapa final é a de classificação.

Ao ser submetido ao mencionado processo de digitalização, é inevitável que determinado segmento de reta contínuo no espaço euclidiano seja afetado por ruído (abrangendo toda transformação, deformação ou influência espúria que afeta significativamente os dados) gerando um segmento digitalizado representado por uma cadeia com imperfeições (não ideal, afetada por erros) para a etapa de classificação.

Garitagoitia (2003) comenta que o processo de digitalização é afetado por erros na fonte de dados, de erros no processamento anterior à transformação para o formato de cadeia, ou até mesmo erros na transformação final para cadeia, e por ruído em todo o processamento.

É um fato que os sinais relacionados a alguma informação normalmente estão sob a interferência de um ambiente de sinais espúrios, que interferem nos dados úteis. Em muitos casos, tais como as usuais interferências estáticas da atmosfera, a influência é de natureza irregular, ou aleatória. Em outros, a informação tem que competir com sinais similares, porém indesejáveis, como a interferência de uma emissora de rádio em outra. Sendo impossível anular essa interferência, é necessário conviver com ela, mesmo desconhecendo-se o modelo probabilístico conveniente para o ruído.

No item a seguir, comentam-se as transformações envolvidas no processo de digitalização.

2.2.1 Transformações entre espaços

O processo de digitalização é interpretado como um processo de transformação entre três espaços: o espaço dos padrões, o espaço de atributos e o espaço decisório. O mundo físico, captado pelo sistema de transdução (sensor), por ter número de dimensões infinitas, deveria ser representado, na verdade, por um continuum de atributos. No caso do pressuposto por esta pesquisa, o transdutor representa o mundo por imagens compostas de grande quantidade, porém finita, de valores escalares. Esses valores são introduzidos como entrada no espaço de atributos. A seguir, a quantidade de valores

tende a ser reduzida ainda mais (também por questões, normalmente, de viabilidade econômica), mantendo o poder de discriminação para a etapa decisória. Sendo o espaço de atributos essencialmente um domínio digital, a essa redução dimensional, de um continuum de parâmetros correspondentes ao mundo físico para um conjunto limitado de valores discretos, se sobrepõem os erros de discretização, as distorções, em suma, o ruído.

Especificando melhor, o sensor poderia ser uma câmera de TV monocromática ou colorida juntamente com uma etapa de imageamento com a capacidade de digitalizar o sinal produzido pelo sensor.

Tipicamente, após a obtenção da imagem digital, o passo seguinte é o pré-processamento da mesma cuja importância reside em aprimorar a qualidade da imagem por meio de técnicas de realce de contrastes, remoção de ruído e isolamento de regiões. A etapa que se segue é a segmentação da imagem em objetos constituintes resultando como saída *pixels* correspondentes às regiões da imagem. De acordo com o método de segmentação, esses *pixels* podem indicar as fronteiras das regiões segmentadas ou as regiões como um todo.

A seguir, o processo de descrição extrai características que resultem em alguma informação requerida para discriminação entre classes de objeto. Os estágios finais são comumente denominados de reconhecimento e classificação. O processo de Reconhecimento atribui rótulos aos objetos baseado com base nas informações fornecidas pelo descritor. O estágio de classificação atribui significados aos objetos reconhecidos.

As implementações desta tese simulam a etapa de classificação com dados de entrada no formato de cadeias. Ruído é tudo que afeta as cadeias de entrada para classificação desde a etapa de captura da imagem pela câmera até a etapa de classificação, indicado pela Definição 23.

Várias decisões em todo o processo afetam o nível de ruído como uma das variáveis no processo da etapa de classificação. Por exemplo, se a digitalização é feita com número de bits suficiente, é esperado que o ruído, introduzido com translações e alterações na escala das retas, possa ser desprezado, desde que o reconhecedor tenha condições de identificar o segmento com pequenas deformações. Porém, distorções comumente ocorrem na formação das imagens devido às projeções geométricas entre os planos da câmera e os dos objetos. Algumas soluções utilizam processos diversos de aquisição (tais como diferença temporal, ou de ponto de vista) da mesma cena com o objetivo de obter um mapeamento entre as imagens correspondentes visando mini-

mizar erros. Imagens obtidas por câmeras de qualquer natureza estarão sempre sujeitas às distorções geométricas e erros residuais decorrentes de imperfeições na montagem das lentes, dos elementos sensores, ou mesmo, devido a características específicas de projeto (LOBÃO; BAUCHSPIESS, 2004).

Considerando o processo de digitalização do ponto de vista de Largeteau-Skapin e Andres (2006):

- Os formalismos matemáticos euclidianos e os formalismos digitais em computação (ou discretos) são geralmente considerados como antagônicos, no sentido de que não ocorrem operações comutativas de um formalismo para o outro: cada formalismo apresenta propriedades particulares que se refletem em suas operações. Existindo tal antagonismo, algumas operações, triviais em um formalismo, não são traduzidas de modo imediato para o outro;
- A correspondência entre conceitos do formalismo digital e do euclidiano costumam envolver considerações sobre erros e distorções, com relação às medidas na fonte de dados;
- A redução desses erros e distorções acarreta maior complexidade computacional dos algoritmos.

Linhas e segmentos de retas incorporam conceitos matemáticos referentes ao mundo euclidiano, os quais, a princípio, facilitariam a aplicação desses conceitos em retas digitalizadas. Caso não haja algoritmos eficientes e flexíveis aptos a levar em conta pequenas distorções nas retas digitalizadas, a primeira dificuldade com esses modelos é que geram uma grande quantidade de segmentos curtos, aumentando a complexidade computacional das soluções.

Constata-se que a etapa de classificação tem que incorporar técnicas de reparo de erros, causados ou não solucionado pelo processo de digitalização como um todo, em que pela Definição 23 o ruído é tudo que afeta a cadeia a ser classificada, tais como:

- Qualidade da imagem;
- Alterações do foco da câmera afetando as dimensões e ângulos de segmentos;
- Interferência de sinais espúrios;
- Pré-processamento imperfeito;

- Distorções afetando as escalas;
- A segmentação nunca é um processo ideal.

2.2.2 Conceitos complementares sobre o *chain code*

Na abordagem estrutural pertinente ao método sintático, parte-se de primitivas, construindo segmentos de retas (ou curvas) até estruturas mais complexas, utilizando normalmente o *chain code* ou alguma variante desse tipo de descritor.

Pela Definição 19, o *chain code*, ou o código da cadeia, é uma sequência de elementos onde cada elemento é um símbolo que representa o vetor unindo dois pixels vizinhos de um arco digital. Dentre suas finalidades, está a de representar o arco digital em questão (DORST; SMEULDER, 1987) (FREEMAN, 1970).

Desde o seu surgimento, várias pesquisas foram realizadas sobre o *chain code*, motivadas por sua simplicidade na tradução dos contornos das formas euclidianas para os formalismos digitais. Na versão básica, o contorno de uma curva arbitrária é aproximado por uma sequência de pequenos vetores de comprimento unitário, compondo um conjunto limitado de possíveis direções (denominado método do vetor unitário).

2.2.3 Métodos de quantização

Para a digitalização de imagens binárias bi-dimensionais, existem dois métodos principais de quantização, a “quantização de contornos de objetos” e a “quantização por interceptação da grade”.

Supondo um contorno de um objeto, em que os seus pontos mais afastados, ainda pertencentes ao objeto, são representados por um linha reta contínua dada pela Expressão 2.6, e o objeto se encontrando na direção de y negativo:

$$y = ax + b. \quad (2.6)$$

Na Expressão 2.6, sem qualquer perda em generalidade, é assumido o seguinte:

$$0 \leq a \leq 1, 0 \leq b < 1. \quad (2.7)$$

Supondo um contorno de um objeto, cujos pontos mais afastados são representados pela reta contínua da Expressão 2.6, e sobrepondo à reta uma grade paralela aos eixos,

em que a reta é submetida ao processo de digitalização, a quantização de contornos de objetos resulta nos seguinte pontos de contorno discretos:

$$(i, y_i) = (i, \lfloor ai + b \rfloor). \quad (2.8)$$

Considerando uma reta contínua da Expressão 2.6, e sobrepondo à reta uma grade paralela aos eixos, em que a reta é submetida ao processo de digitalização, a quantização por interceptação da grade resulta nos seguinte pontos discretos:

$$(i, y'_i) = (i, \lfloor ai + b \rfloor). \quad (2.9)$$

A Definição 39 resume o significado dos métodos mencionados.

Definição 39 *Métodos de quantização:* os principais são a quantização de contornos de objetos e a quantização por interceptação da grade:

- *Quantização de contornos de objetos, na qual os pontos mais externos dos conjunto de pontos ainda pertencentes a um objeto são digitalizados por uma cadeia, a qual pode ser codificada pelo modelo de Freeman da Definição 20.*
- *Quantização de interseção em grade, em que os pontos da grade mais próximos de um arco, sempre que este intercepta uma linha ou uma coluna da grade estão conectados por uma cadeia (adaptado de Dorst e Smeulder (1984)).*

O método por interceptação da grade consiste dos seguintes passos:

1. Sobreponha-se uma grade digital quadrada à imagem;
2. Determinam-se todas as intersecções entre a grade e o segmento da imagem;
3. Para todas as intersecções encontradas no passo 2, determina-se o ponto de grade mais próximo;
4. O segmento é representado pelo conjunto de pontos de grade encontrados no passo 3, tal que a cada um desses pontos seja associado um vetor do *chain code* com referência ao ponto anterior do seguinte modo: parte-se de um ponto inicial, e os pontos subsequentes são especificados com referência ao ponto anterior na codificação.

A partir da grade digital correspondente, que pode ser de conexão em vizinhança-4 (4 direções), em vizinhança-6 (6 direções) em vizinhança-8 (8 direções) ou ser generalizada, a codificação é efetuada por vetores (*links*) adjacentes sucessivos. Tal generalização significa que as vizinhanças mencionadas (ver Figura 2.8) fazem parte de uma família em que cada vizinhança é representada por um inteiro positivo m . Porém, na prática, m não excede a 10 e nem todos os valores de m entre 1 e 10 são de igual utilidade, por questões dependentes das propriedades dos arcos (tal como a função de curvatura) ou da velocidade de processamento requerida (SAGHRI; FREEMAN, 1981).

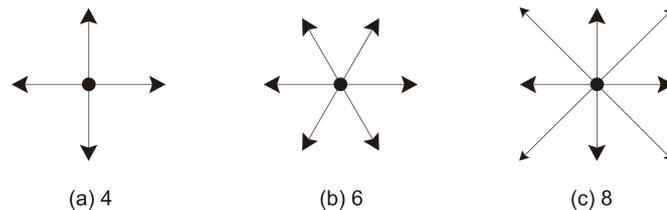


Figura 2.8: Representação de *chain code* de 4, 6 e 8 direções (adaptado de Tang (2001)).

Algumas vantagens e desvantagens do *chain code* estão relacionadas abaixo (TANG, 2001) (KOPLOWITZ, 1981):

1. Vantagens:

- Simplicidade: Apesar de existirem esquemas mais eficientes no critério de memória de armazenamento requerida, o *chain code* apresenta uma boa relação entre eficiência e facilidade de processamento;
- Possibilita avaliação de características da forma tais como área e perímetro;
- Por codificar a posição relativa do contorno, ao invés da absoluta, os dados são compactados, sendo a representação invariante a translações na forma correspondente.

2. Desvantagens

- É inerentemente sensível ao ruído e a erros de discretização;
- As operações subsequentes à codificação serão sensíveis a esses erros.

No que se refere a estudos sobre a perda de informação com o *chain code*, em Saghri e Freeman (1981) o código é inicialmente ampliado para um *chain code* generalizado. A seguir, o erro de quantização associado a um ponto num arco é definido como a menor distância entre o ponto e o *chain code*, viabilizando análises de precisão. Concluiu-se

que qualquer que seja o tipo de *chain code*, o erro de quantização é limitado a menos do que metade do espaçamento da grade de codificação. O aumento do número de direções do *chain code* apenas resulta em aumento do custo computacional referente aos aspectos de memória de armazenamento e tempo de processamento, não afetando o erro de quantização. É importante saber que, de acordo com a precisão requerida, a grade digital deve ser suficientemente fina para preservar o raio de curvatura mínimo, considerando também que erros de quantização máximos e médios constituem duas características de desempenho de qualquer processo de codificação.

2.2.4 Sensibilidade do *chain code*

A seleção de um esquema de codificação específico, dentre os abrangidos pelo conceito de *chain code* generalizado, deve levar em conta, principalmente, as propriedades das curvas a serem reconhecidas, mas não o erro de quantização. No caso de retas, dependendo da grade digital, Bongiovanni, Luccio e Zorat (1975) estudam uma equação para descrever o conjunto de retas contínuas associadas ao *chain code*. Ou seja, a perda de informação resultante do processo de digitalização se reflete em que a cada segmento digitalizado está associado um conjunto de retas contínuas, e não apenas à reta que deu origem ao código.

A sensibilidade do *chain code* pode ser avaliada introduzindo-se o mesmo contorno em grades diferentes de acordo com a Figura 2.9, e comparando-se os respectivos códigos com os detalhes preservados das retas (TANG, 2001).

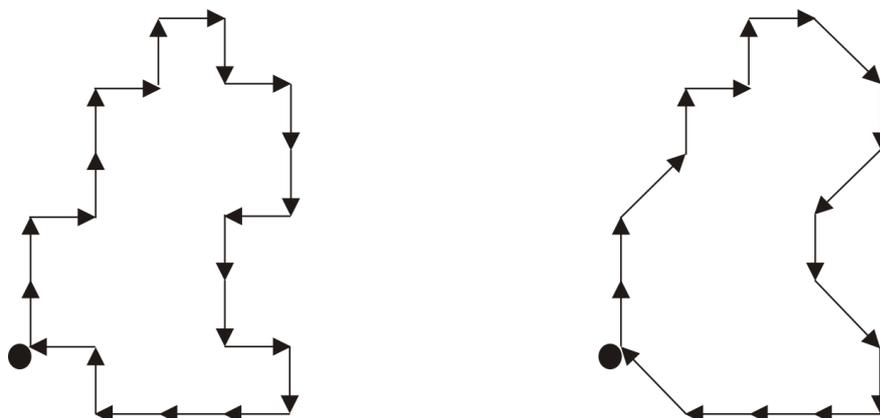


Figura 2.9: Um mesmo contorno em duas grades diferentes (adaptado de (TANG, 2001)).

2.3 As retas digitais clássica e analítica

2.3.1 O conceito clássico

Antes de considerar as influências espúrias, é necessário entender a informação contida no *chain code* sobre a reta contínua do espaço euclidiano que lhe deu origem. O estudo de Dorst e Smeulders (1991) comentado neste item não leva em conta essas influências, sendo apropriado para tal entendimento.

Para o sistema cartesiano ortogonal com eixos das abscissas x e das coordenadas y , considere-se uma semirreta contínua no primeiro quadrante conforme Expressão 2.10.

$$y = \alpha_0 x + e_0, 0 \leq \alpha_0 < 1, \quad (2.10)$$

onde α_0 indica a inclinação (o ângulo da linha com respeito ao eixo das abscissas x), e e_0 indica o deslocamento (*offset*) vertical para $x = 0$.

Existem duas representações clássicas para um segmento digital em uma grade. Inicialmente, considere um segmento digitalizado com n elementos denominado ζ^n definido pelo conjunto de pontos de grade (x_i, y_i) resultantes da digitalização da reta da Expressão 2.10. O conjunto de pontos é indicado pela Expressão 2.11 seguindo a notação de Dorst e Smeulders (1991) que utilizam a vizinhança-8 e “quantização de contornos de objetos” em uma grade quadrada unitária.

$$y_i = \lfloor \alpha_0 i + e_0 \rfloor, x_i = i, \forall i \in \{0, 1, \dots, n\}. \quad (2.11)$$

Pela Expressão 2.11 obtêm-se a primeira representação, em que se especifica o ponto mais à esquerda da grade $i = 0$ seguido pelo *chain code* dos vetores que descrevem as transições entre pontos digitais adjacentes do segmento digital.

A segunda representação clássica de segmentos digitais é mais concisa e informativa do que a lista de n vetores da primeira representação por utilizar apenas quatro números inteiros n, q, p, s definidores de uma pré-imagem, descrita no tópico a seguir.

2.3.1.1 Pré-imagem de segmentos digitalizados

A Definição 22 apresenta o conceito de pré-imagem como o conjunto de linhas retas euclidianas, cuja digitalização resulta no mesmo segmento inicial dado. Demonstra-se que a pré-imagem corresponde a um polígono convexo em um espaço de parâmetros

(DORST; SMEULDER, 1987).

Dorst e Smeulders (1991) provam que toda cadeia reta (gerada por uma reta contínua) pode ser representada por uma quádrupla (n, q, p, s) de números inteiros, nenhum dos quais é maior do que n , com q e p sendo números inteiros primos entre si. Essa quádrupla define a pré-imagem do segmento digital codificado pela cadeia.

O *chain code* correspondente à reta da Expressão 2.10 é representado em termos dos quatro parâmetros mencionados da seguinte maneira:

$$\zeta_i^n = \left\lfloor \frac{p}{q}(i-s) \right\rfloor - \left\lfloor \frac{p}{q}(i-s-1) \right\rfloor; i = 0, 1, \dots, n \quad (2.12)$$

A interpretação dos parâmetros (n, q, p, s) da Expressão 2.12 é a seguinte:

- n : é o comprimento da cadeia correspondente ao segmento;
- q : é o período da sequência dos n vetores;
- p : está relacionado à inclinação de reta definida pelo número racional $p \div q$ que é uma aproximação de α_0 . O significado de p é de um número inteiro tal que se a subsequência de período q se repetir indefinidamente, será criada uma linha digital infinita na qual estará incluído o dado segmento digital. A pré-imagem desta linha infinita incluirá somente as linhas cujas inclinações sejam $p \div q$;
- s : com $0 \leq s < q$, apresenta o significado de fase a fim de diferenciar entre as linhas digitais contendo os primeiros três parâmetros n, q, p idênticos.

Assim, pela Definição 22, dado um segmento digitalizado no plano (x, y) , define-se um conjunto denominado pré-imagem (ou domínio) como o conjunto contendo todas as linhas retas contínuas, cuja digitalização resulta no mesmo segmento digitalizado na grade. No tal domínio exemplificado na Figura 2.10 por Aghito e Forchhammer (2006), existem três ou quatro linhas, denominadas “linhas delimitadoras”, as quais especificam as fronteiras da região interior ao domínio no plano. Essa região tem a particularidade de conter o conjunto total de linhas da pré-imagem. Normalmente, usa-se nas pesquisas o caso com quatro linhas de delimitação da Figura 2.10, uma vez que o caso com três linhas pode ser tratado como um caso especial do primeiro.

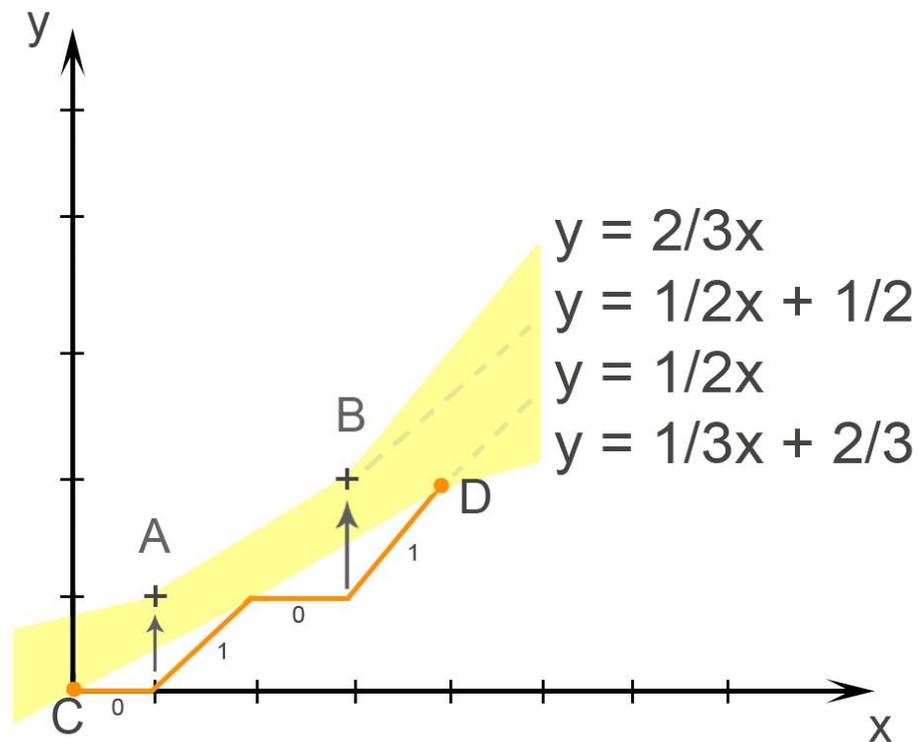


Figura 2.10: Segmento de linha reta digitalizado com o código 0101 e sua pré-imagem no plano (x,y) (área em amarelo) (adaptado de Aghito e Forchhammer (2006)).

Em especial, os vértices de intersecção A , B , C e D das linhas de delimitação da pré-imagem mostrada na Figura 2.10 têm, no máximo, três valores distintos de coordenadas, sendo que $p \div q$ é a coordenada intermediária. Portanto, se o domínio tiver quatro vértices, dois deles apresentarão coordenadas idênticas. Note que todas as informações sobre o segmento digitalizado em questão, inclusive o seu comprimento, são totalmente capturadas por esses quatro pontos.

Mais formalmente, a pré-imagem do segmento de linha reta digital ζ^n é o conjunto de todas as linhas contínuas representadas pela Expressão 2.13 que resultem em segmento digitalizado ζ^n representados pela Expressão 2.12, por incidirem na região correspondente às linhas delimitadoras.

$$y = \alpha x + e, \quad (2.13)$$

relativamente ao eixo das abscissas x e eixo das coordenadas y , onde α indica a inclinação (o ângulo da linha com respeito ao eixo x), e e indica o deslocamento (*offset*) vertical para $x = 0$.

Assim, a condição necessária e suficiente para que uma linha reta contínua representada pela Expressão 2.13 resulte em segmentos digitalizados no interior da pré-imagem

de ζ^n é atendida pela Expressão 2.14.

$$[\alpha_0 i + e_0] \leq \alpha i + e < [\alpha_0 i + e_0] + 1, i \in \{0, 1, \dots, n\} \quad (2.14)$$

Uma característica interessante é que cada linha delimitadora é univocamente mapeada a um ponto no plano (α, e) . Nesse plano, existem 5 tipos de domínios possíveis no espaço de parâmetros $\alpha \times e$ representados na Figura 2.11.

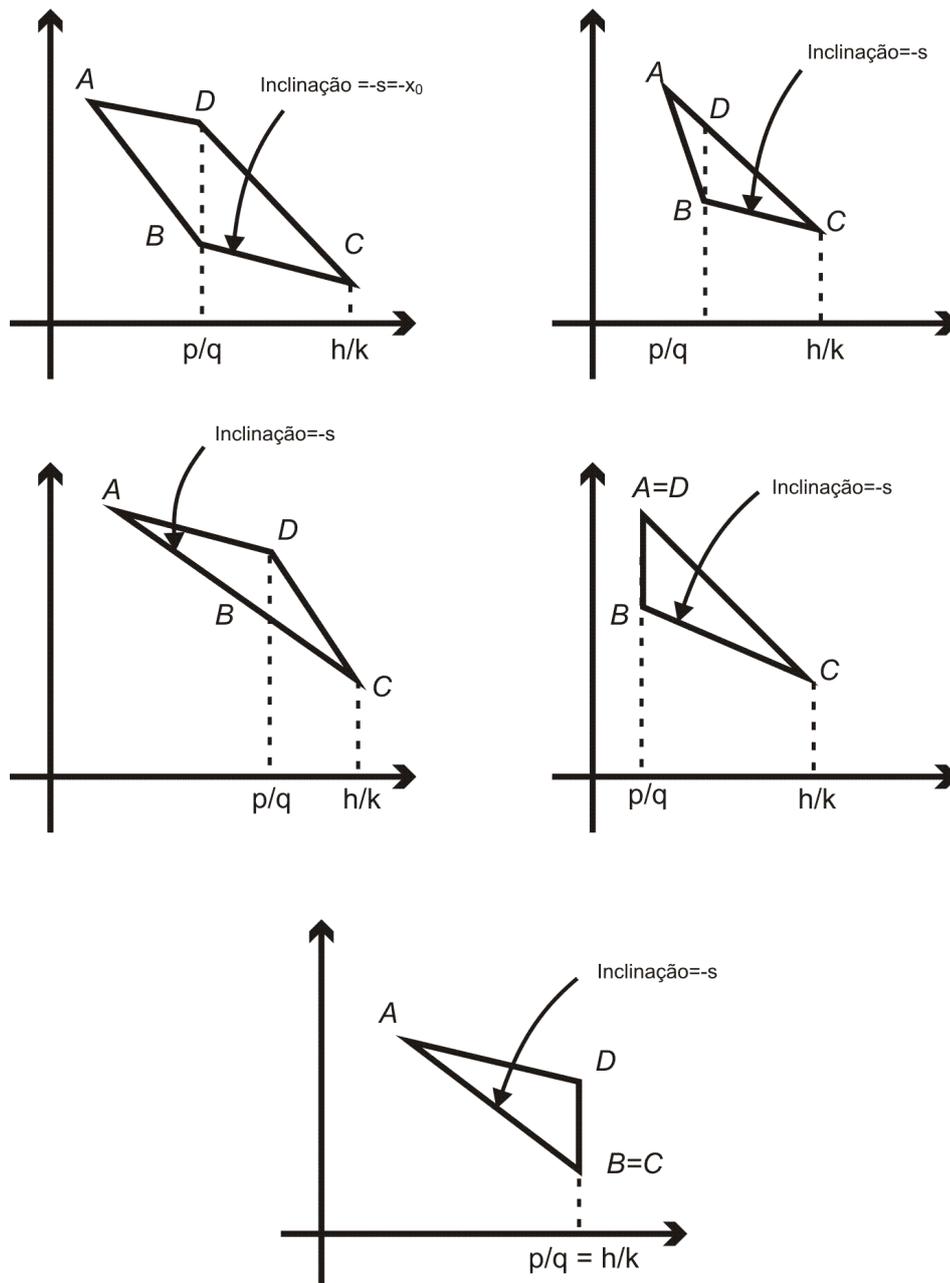


Figura 2.11: Cinco tipos de domínios que são possíveis no espaço dos parâmetros (adaptado de Dorst e Smeulder (1984)).

Os passos seguintes podem ser seguidos para se especificar os parâmetros (n, q, p, s) a

partir dos pontos A, B, C e D do plano (x, y) e dos pontos $\bar{A}, \bar{B}, \bar{C}$ e \bar{D} do plano (α, e) :

1. Inicialmente no plano (x, y) , identifica-se o valor da coordenada intermediária dos pontos, obtendo-se $p \div q$;
2. n é simplesmente o comprimento do segmento;
3. s pode ser definido através do seguinte argumento: No plano (x, y) , a linha com inclinação $\frac{p^-}{q^-}$, que é digitalizado para o segmento de linha digital ζ^n , atravessa um ponto de grade tal que $x_i = s$, sendo mapeada para o ponto \bar{B} no espaço de parâmetros da Figura 2.11. A linha com a inclinação máxima $\frac{p^+}{q^+}$ que é digitalizado para o segmento de linha digital ζ^n também atravessa esse mesmo ponto sendo mapeado para o ponto \bar{C} no espaço dos parâmetros;
4. O ponto $x_i = s$ é mapeado na linha $\overline{\bar{B}\bar{C}}$ no plano (αe) . s é a inclinação de $\overline{\bar{B}\bar{C}}$;
5. Portanto, conhecendo-se a região, definem-se todos os parâmetros.

O exemplo seguinte ilustra os procedimentos comentados.

Exemplo de espaço de parâmetros

Supondo a cadeia 0101 com $n = 3$ da Figura 2.10, os pontos extremos da sequência de vetores são os pontos C e D . Pela Expressão 2.14 definem-se os pontos A e B .

Pelos “pontos limitantes” A, B, C e D da Figura 2.10 definem-se as retas delimitadoras. A reta \overline{CB} (com inclinação $\frac{p^+}{q^+}$) é a reta delimitadora com maior inclinação. A reta \overline{AD} (com inclinação $\frac{p^-}{q^-}$) é a reta delimitadora com menor ângulo de inclinação.

Portanto, a reta $\overline{CB} y = 2/3x$ fornece os parâmetros $2/3$ (correspondente à maior inclinação $\frac{p^+}{q^+}$) e e mínimo igual a zero. Analogamente, a reta $\overline{AD} y = 1/3x + 2/3$ (correspondente a $\frac{p^-}{q^-}$) fornece os parâmetros $1/3$ e $2/3$.

Cada reta delimitadora é mapeada a um único ponto no espaço de parâmetros $\alpha \times e$ da Figura 2.12.

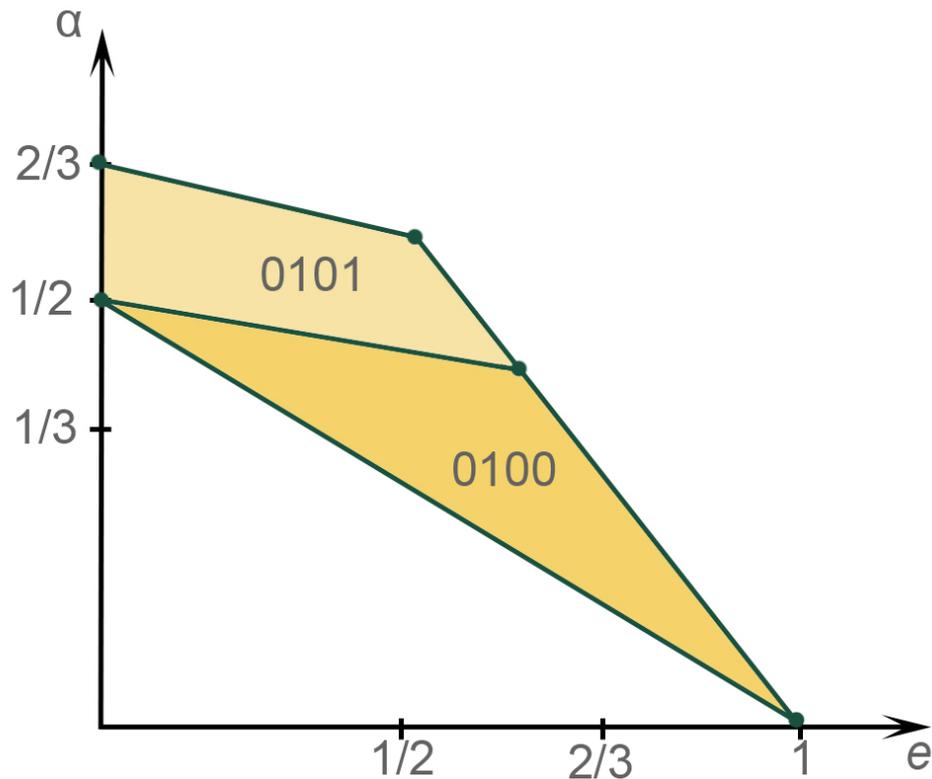


Figura 2.12: Espaço de parâmetros, com a pré-imagem de dois segmentos com códigos de cadeia 0101 e 0100. A união das regiões compõem a pré-imagem do código da cadeia 010 (adaptado de Aghito e Forchhammer (2006)).

O interior do quadrângulo ou triângulo com os vértices nestes pontos é a pré-imagem dos segmentos digitais no plano dos parâmetros como mostrado na Figura 2.12. Essa Figura indica duas pré-imagens: uma para a cadeia 0101 e outra para a cadeia 0100. A união dessas duas pré-imagens é a pré-imagem do segmento com *chain code* 010.

No item a seguir, apresentam-se conceitos iniciais para o entendimento dessa abordagem que busca integrar propriedades topológicas e geométricas das retas.

2.3.2 O conceito analítico pelas linhas discretas aritméticas

Freeman (1970) seguiu uma abordagem algorítmica, definindo linhas discretas como digitalizações de euclidianas. Por outro lado, Reveillès (1991) apud Fiorio, Jamet e Toutant (2006) deu início à geometria discreta aritmética apresentando as linhas discretas aritméticas como subconjuntos de Z^2 satisfazendo a uma dupla desigualdade diofantina. Devido ao caráter discreto de Z^n , um único objeto contínuo tem várias representações discretas distintas, cada uma herdando algumas de suas propriedades. Por exemplo, linhas discretas aritméticas são definidas como tiras de acordo com um parâmetro $w \in N$, a denominada espessura aritmética, conforme definido a seguir.

Definição 40 *Linha discreta aritmética:* uma linha discreta aritmética é um subconjunto de Z^2 definido por $D(a, b, \mu, \omega)$, apresentando vetor normal $v = (a, b) \in R^2$, parâmetro de translação $\mu \in R_+$ e espessura aritmética $w \in R_+$ conforme Expressão 2.15 e Figura 2.13 (FIORIO; JAMET; TOUTANT, 2006) (ROUSSILLON; SIVIGNON; TOUGNE, 2007).

$$D(a, b, \mu, \omega) = \{(i, j) \in Z^2 : \frac{\omega}{2} \leq ai + bj + \mu < \frac{\omega}{2}\} \quad (2.15)$$

A interpretação geométrica dos parâmetros da Expressão 2.15 é apresentada na Figura 2.13 no plano $x y$.

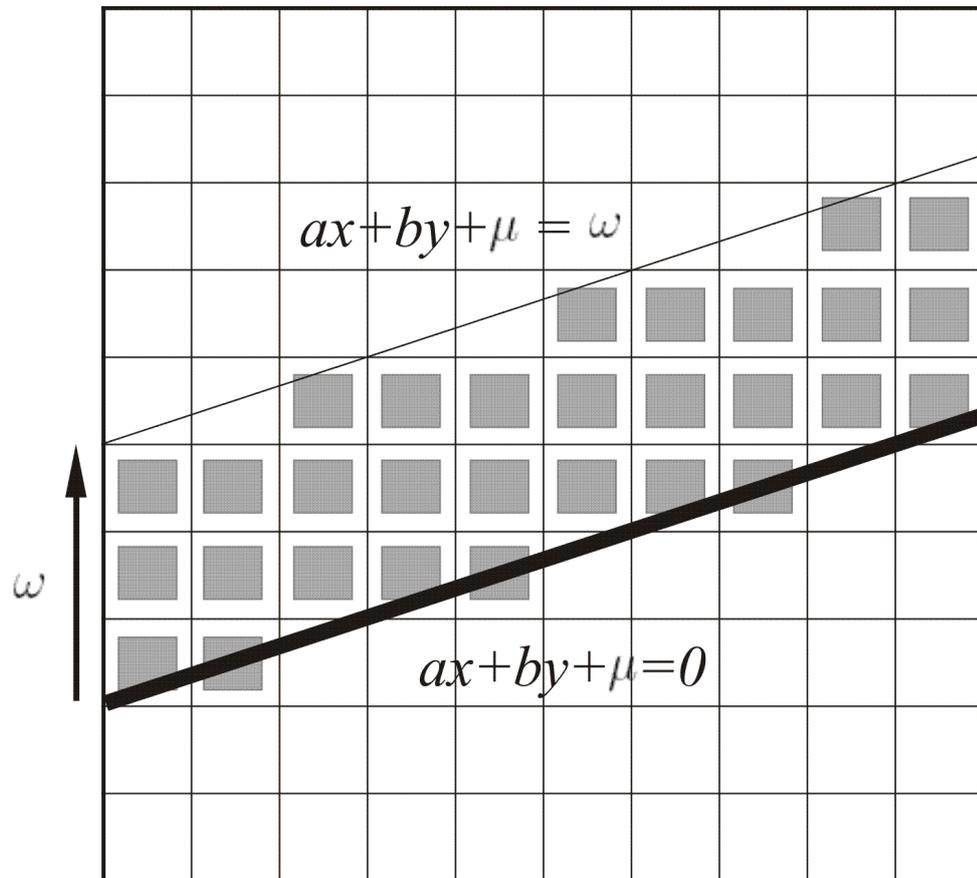


Figura 2.13: Interpretação geométrica da definição geral de linha discreta aritmética no plano $x y$ (adaptado de Figueiredo (1999)).

Observe-se que a escolha das desigualdades na Expressão 2.15 é arbitrária; assim, pode-se optar por definir a linha discreta aritmética $\tilde{D}(a, b, \mu, \omega)$ apresentando vetor normal $v = (a, b) \in R^2$, parâmetro de translação $\mu \in R_+$ e espessura aritmética $w \in R_+$ de acordo com a Expressão 2.16.

$$\tilde{D}(a, b, \mu, \omega) = \{(i, j) \in Z^2 : \frac{\omega}{2} < ai + bj + \mu \leq \frac{\omega}{2}\} \quad (2.16)$$

A diferença entre as Equações 2.15 e 2.16 está no posicionamento do símbolo de \leq nas inequações (à esquerda ou à direita) a fim de estabelecer um limite inferior e superior. Assim, as linhas discretas aritméticas $D(a, b, \mu, \omega)$ e $\tilde{D}(a, b, \mu, \omega)$ são denominadas linha discreta aritmética inferior e linha discreta aritmética superior respectivamente, apresentando vetor normal $v = (a, b) \in \mathbb{R}^2$, parâmetro de translação $\mu \in \mathbb{R}_+$ e espessura aritmética $w \in \mathbb{R}_+$.

Segue-se que uma linha discreta aritmética é o conjunto de soluções de equações diofantinas, ou seja, aquelas com soluções inteiras. Dada uma linha discreta aritmética, por conseguinte, $D(a, b, \mu, \omega)$ é o conjunto de soluções de no máximo w equações diofantinas. A espessura aritmética w está fortemente relacionada com a conectividade da respectiva linha discreta aritmética.

A Figura 2.14 mostra uma mesma linha $D(3, -5, -4, 5)$, pela representações clássica e pela representação pela geometria discreta aritmética. Uma linha digital pode ser vista tanto como um conjunto de pixels ou equivalentemente como um conjunto de pontos inteiros compreendidos entre duas linhas euclidianas.

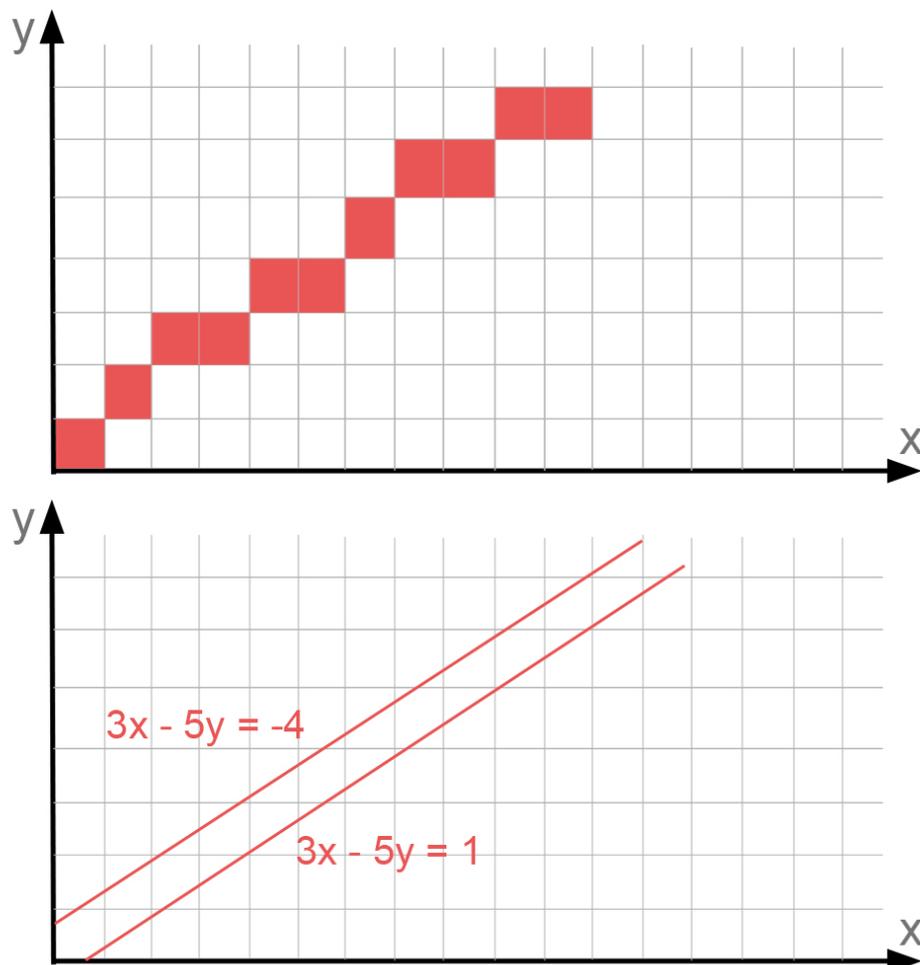


Figura 2.14: Uma linha digital representada por linha discreta aritmética (abaixo) e na representação clássica (acima) (adaptado de Figueiredo (1999)).

Pela Figura 2.14 é possível um outro ponto de vista alterando-se a Definição 2.15 para R^2 ao invés de Z^2 . Nesse caso, define-se uma faixa contínua de largura $w = \frac{\rho}{\sqrt{a^2+b^2}}$ medida ortogonalmente na direção da faixa. Essa faixa contínua pode ser interpretada como a contraparte contínua da linha discreta aritmética da Definição 2.15.

2.3.3 Definições

Nesta seção, apresentam-se definições relacionadas principalmente aos trabalhos com base na geometria discreta aritmética, possibilitando o entendimento do estado-da-arte.

Fecho convexo discreto

Os algoritmos em geometria discreta aritmética costumam partir da identificação do fecho convexo discreto do conjunto de pontos. A Definição 41 apresenta os conceitos de conjunto convexo e fecho convexo.

Definição 41 *Conjunto convexo e fecho convexo: um conjunto de pontos em geometria euclidiana é convexo se, e somente se, os segmentos de reta conectando quaisquer dois pares de pontos do conjunto estão incluídos no conjunto. Na geometria discreta, considerando que “[...] a intersecção de qualquer família de conjuntos convexos é também um conjunto convexo, existe o menor conjunto convexo contendo qualquer dos conjuntos dados. A esse menor conjunto convexo que contém o conjunto dado denomina-se o fecho convexo do conjunto”, adaptado de (SOILLE, 1998).*

A Figura 2.15 mostra um conjunto de pontos e o correspondente fecho convexo que contém os pontos.

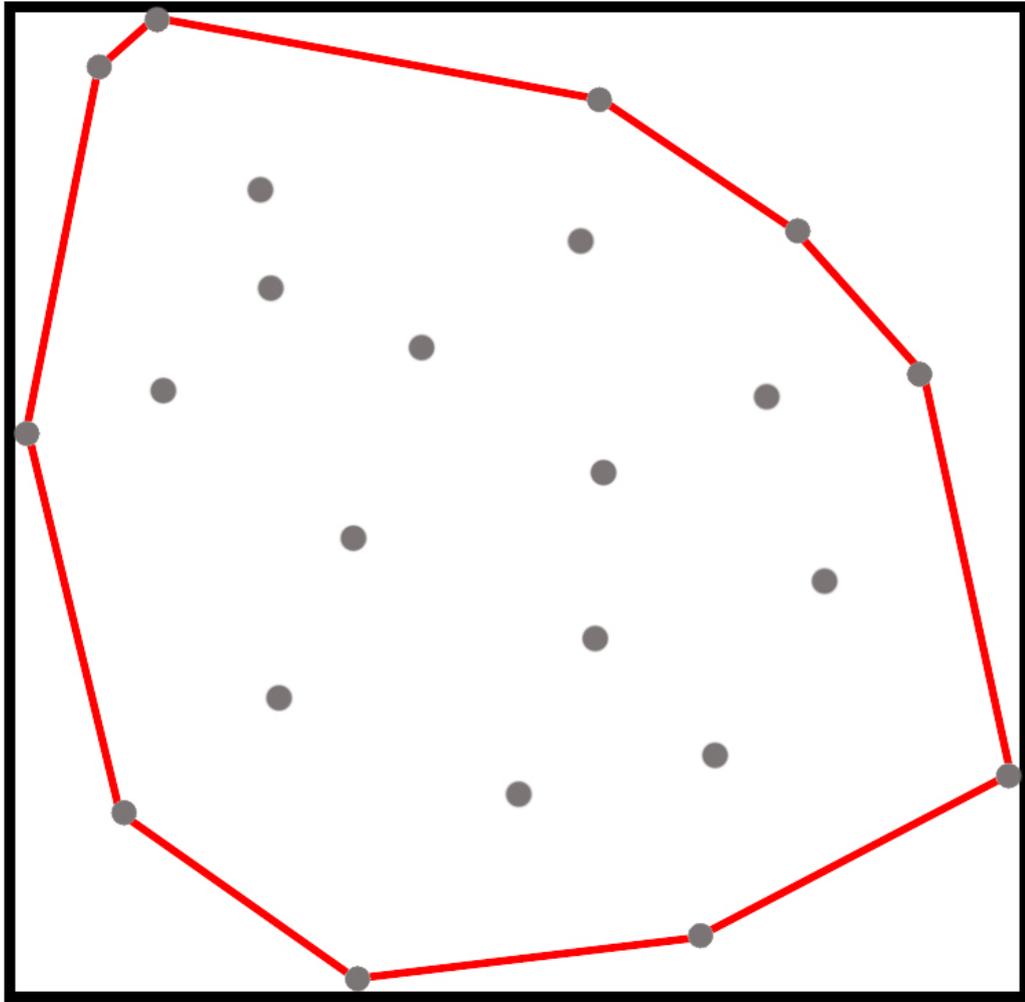


Figura 2.15: Fecho convexo: o polígono indicado que contém o conjunto de pontos representados, passando por alguns desses pontos, é o fecho convexo do conjunto de pontos planares (adaptado de Feschet (2006)).

No caso discreto, ocorre o problema de determinar se um conjunto de pontos é convexo. Isso porque a discretização de um segmento s qualquer conforme a Figura 2.16, gera a ambiguidade em definir retas euclidianas associadas aos *pixels* nos extremos do segmento digitalizado (SOILLE, 1998).

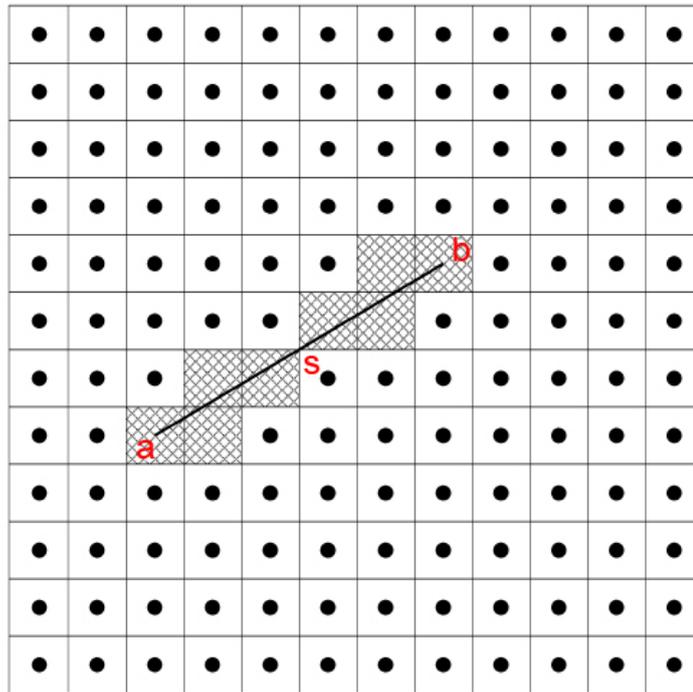


Figura 2.16: Discretização de um segmento. Os quadrados com os pontos no centro são os *pixels*. Os quadrados sombreados são os *pixels* interceptados pelo segmento s (adaptado de Gioi (2008)).

Ainda para o caso discreto, a Figura 2.17 mostra três pontos e o correspondente fecho convexo (2006).

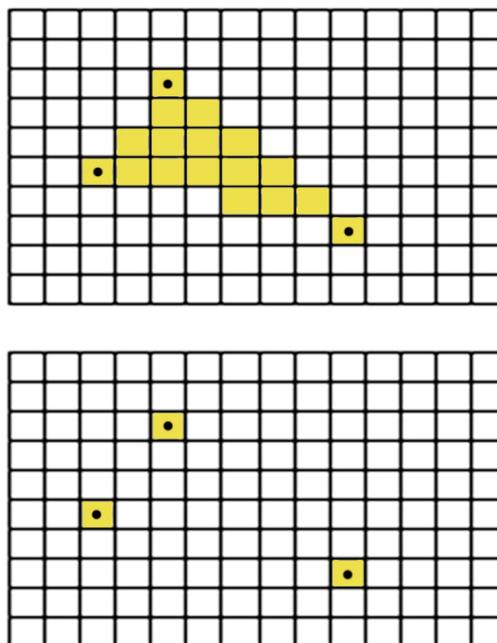


Figura 2.17: Um conjunto de três pontos (abaixo) e o correspondente fecho convexo (acima) (adaptado de Feschet (2006)).

Comprimento de Conjunto Fechado Convexo

Definição 42 *Comprimento de conjunto fechado convexo:* o comprimento $w_c(K)$ de um conjunto convexo fechado (abreviado comprimento de conjunto) K ao longo de uma direção $c \in R^n$, é definido pela Expressão 2.17:

$$w_c(K) := \max(cx : x \in K) - \min(cx : x \in K) \quad (2.17)$$

(FESCHET, 2006).

O comprimento de reticulado de conjuntos planares é definido a seguir.

Comprimento de reticulado de conjuntos planares

Definição 43 *Comprimento de reticulado:* a partir da Definição 42, o comprimento de reticulado $w(K)$ de K , com respeito ao reticulado padrão Z^n , é o mínimo de seus comprimentos tomados em todas as direções não nulas, conforme a Expressão 2.18:

$$w(K) := \min(w_c(K) : c \in Z^n) \quad (2.18)$$

(FESCHET, 2006).

Da Definição 43 e Expressão 2.18, conclui-se:

- O vetor-linha c que atingir o mínimo da Expressão 2.18 chama-se direção de espessura (*width direction*) do conjunto K ;
- Obviamente, $w(v + \alpha K) = \alpha w(K)$ para qualquer vetor v e qualquer número racional não negativo α ;
- Os conjuntos K e $(v + \alpha K)$ têm a mesma direção de espessura.

As definições seguintes são extensões das definições anteriores.

Razão de direção e segmento quase em linha reta

Tendo em vista que em R^2 uma linha reta apresenta uma única dimensão (não tem espessura), isto significa que a altura da ortogonal à linha é nula. Assim, no espaço digital, após o cálculo do comprimento de reticulado em todas as direções, calcula-se também o comprimento de reticulado do conjunto de pontos digitais na direção ortogonal.

Definição 44 *Razão de direção:* após o cálculo do comprimento de reticulado em todas as direções, calcula-se também o comprimento de reticulado do conjunto de pontos digitais na direção ortogonal ao conjunto. A seguir, para cada direção calcula-se um valor denominado razão de direção. Esse valor é a razão entre o menor dos comprimentos de reticulado do conjunto de pontos digitais e o maior dos comprimentos de reticulado na direção ortogonal, relativamente à direção específica (FESCHET, 2006).

Definição 45 *Segmento quase em linha reta pela geometria discreta aritmética:* para todas as direções possíveis de acordo com a Definição 44, escolhe-se aquela com a menor proporção, definindo-se antecipadamente um limite (threshold). Portanto o conjunto é considerado como um segmento quase em linha reta se, e somente se, a correspondente razão de direção for menor do que o limite (FESCHET, 2006).

Segmentos borrados

Definição 46 *Segmento borrado:* um conjunto \mathcal{S}_b de pontos consecutivos $|\mathcal{S}_b| \leq 2$ de uma curva em vizinhança-8 é um segmento borrado de ordem d no sentido de Reveillès (1991), se existe uma reta discreta $\mathcal{D}(a, b, \mu, \omega)$ denominada delimitadora, de tal forma que todos os pontos de \mathcal{S}_b pertençam a \mathcal{D} e $(\omega \div \max(|a| + |b|)) \leq d$ (DEBLED-RENESSON; FESCHET; ROUYER-DEGLI, 2006),.

2.4 Soluções pela geometria discreta aritmética

Esta seção apresenta conceitos para o entendimento dessa abordagem que busca integrar as propriedades topológicas e geométricas de retas.

Pela Expressão 2.15 uma linha discreta aritmética é definida como um conjunto de pontos $(x, y) \in \mathbb{Z}^2$. Valores especiais de ω conduzem às linhas clássicas da grade conexão-8 ou conexão-4:

- linhas da grade em conexão-8: $\omega = \max(|a|, |b|)$;
- linhas da grade em conexão-4: $(\omega = (|a| + |b|))$ (FESCHET, 2006).

Feschet (2006) comenta que o cálculo da espessura aritmética é exatamente o mesmo problema da determinação do comprimento de reticulado de pontos de uma linha discreta. Esse trabalho utiliza o comprimento de reticulado para extrair os sentidos de

linhas que minimizam a espessura aritmética. Isso permite listar todas as direções que são soluções.

A espessura aritmética pode ser minimizada enquanto que a espessura geométrica não pode. Com efeito, a equação $1 + d\sqrt{a^2 + b^2} = \omega$ é obtida por relações geométricas, sendo d a espessura geométrica de um conjunto ao longo da direção (a, b) e ω correspondendo ao comprimento de reticulado nessa direção.

O algoritmo proposto por Feschet (2006) se baseia no princípio de que o comprimento de reticulado de um conjunto de pontos K é necessariamente atingido por dois vértices opostos de K . Esse trabalho levanta o problema do cálculo exato do comprimento de reticulado de qualquer polígono inteiro K , por uma compartimentação geométrica do espaço de possíveis direções. Além disso, todas as direções que conduzem a comprimentos de reticulado devem ser calculadas.

Entretanto, o algoritmo de Feschet (2006) requer, como um dos dados de entrada, o polígono representativo do fecho convexo. Esse polígono é necessário para atender ao princípio de que o comprimento de reticulado de um conjunto de pontos é encontrado necessariamente entre dois vértices do polígono mencionado. Portanto, pela combinação de lados desse polígono, identifica-se a direção que minimiza o comprimento de reticulado.

O trabalho de Fiorio, Jamet e Toutant (2006) é um exemplo interessante da aplicação da espessura w . A fim de caracterizar círculos discretos, o ponto-chave da abordagem de Fiorio, Jamet e Toutant (2006) é uma função de espessura tal que w varie como uma função de derivadas locais da curva, pois w seria normalmente considerado nas Expressões 2.15 e 2.16 como uma constante. A mencionada função de espessura leva a um melhor conhecimento de linhas discretas aritméticas melhorando, parcialmente, o conhecimento de círculos discretos.

Simplificadamente, os círculos discretos são definidos como anéis, dependendo de um parâmetro $\omega \in \mathbb{R}_+$ chamado espessura aritmética, analogamente às linhas discretas aritméticas, em que $v = (a, b) \in \mathbb{R}^2$ é o vetor normal do plano tangente à linha euclidiana com equação $ax + by + \mu = 0$ em cada um dos seus pontos, com a e b sendo as derivadas da forma linear $ax + by + \mu$.

No item a seguir, comentam-se outros trabalhos importantes.

2.4.0.1 Um Conceito alternativo de reta

Neste item, resume-se o trabalho de Feschet (2008), o qual define um novo conceito de reta, comentado também em Feschet (2006), voltado à solução da restrição de perda de conexão com a aritmética dos segmentos borrados descritos em Debled-Rennesson, Feschet e Rouyer-Degli (2006).

Feschet (2008) avalia inicialmente a possível escolha de uma direção correspondente ao comprimento de reticulado no âmbito do conjunto de pontos digitais sob estudo. Contudo, isto nem sempre pode ser executado e a solução pode não ser única. Este foi o problema identificado correspondendo efetivamente à existência de simetrias no conjunto de pontos digitais.

Adicionalmente, mesmo que a direção correspondente ao comprimento de reticulado seja única, pode não ser uma boa solução. Na verdade, o problema torna-se crítico em conjuntos pequenos, os quais são frequentemente detectados como segmentos verticais ou horizontais. Portanto, a possível orientação visual do conjunto não é identificada pelo comprimento de reticulado mínimo.

Para se ter uma solução mais robusta, o autor preferiu usar outro critério. A fim de esclarecer tal critério, é conveniente retornar, provisoriamente, ao espaço em R^2 onde uma linha reta apresenta uma única dimensão (não tem espessura). Em R^2 , isto significa que a altura da ortogonal à linha é nula. Assim, voltando ao espaço digital, após o cálculo do comprimento de reticulado em todas as direções, calcula-se também o comprimento de reticulado do conjunto de pontos digitais na direção ortogonal.

Então, para cada direção, a razão entre o menor valor e o maior deles é calculada, resultando na “razão de direção” (*ratio of direction*) da Definição 44. Para todas as direções possíveis, escolhe-se aquela com a menor proporção, definindo-se antecipadamente um limite (*threshold*). Portanto o conjunto de pontos digitais é considerado como um segmento “quase em linha reta” se, e somente se, a correspondente razão de direção for menor do que o limite.

2.5 Métricas

Destacam-se neste item dois trabalhos que aplicam métricas para avaliar se dois segmentos de reta digitalizados pertencem a uma única estrutura linear. O primeiro é o

trabalho de Nacken (1993) que fundamenta o assunto de modo geral. Nacken (1993) introduz uma métrica concebida especialmente para segmentos de retas digitais. Park (2009) comenta que essa métrica tem sido usada com sucesso para medição de colinearidade e separação entre os segmentos de reta, levando em conta tanto a distância entre as posições dos dois segmentos, mas também as diferenças em suas características internas.

A Figura 2.18 apresenta dois segmentos com seus parâmetros.

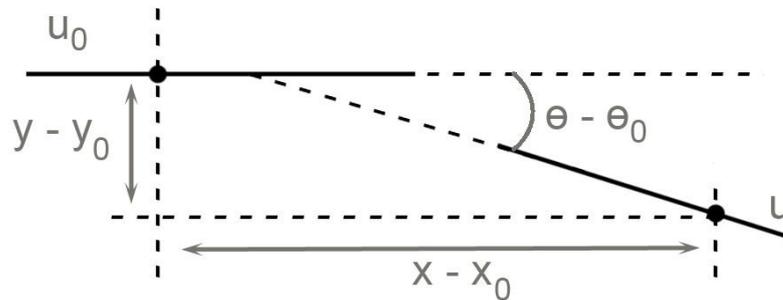


Figura 2.18: Dois segmentos de reta para a exemplificação de métricas (adaptado de Park (2009)).

A função de vizinhança para estimar a probabilidade de dois segmentos de reta, μ_0 e μ da Figura 2.18, pertencerem à mesma linha reta é apresentada na Expressão 2.19.

$$f(\mu_0, \mu) = G_{\sigma_{\text{ângulo}}}(\theta - \theta_0) \times G_{\sigma_{\text{comprimento}}} G_{\sigma_{\text{espessura}}}(R_{\theta_0}^{-1}(x - x_0, y - y_0)), \quad (2.19)$$

onde os seguintes elementos da Figura 2.18 são considerados:

- A diferença nas direções: $\theta - \theta_0$;
- A distância entre os pontos médios dos dois segmentos na direção da estrutura linear: $x - x_0$;
- O deslocamento de um segmento de reta com relação ao outro na direção ortogonal: $y - y_0$;
- A distribuição de Gauss para a componente de ângulo ou orientação é $G_{\sigma}(x)$ e a distribuição de Gauss para o componente de deslocamento é $G_{\sigma_x \sigma_y}(xy)$. R_{θ} está no denominador da Expressão 2.19, pois é um fator relacionado à rotação do segmento de um ângulo θ . Note que segmentos de reta são invariantes em translação e rotação, mas não são invariantes em escala. .

2.6 Considerações finais

Esta seção apresenta um resumo conclusivo sobre os tópicos anteriores.

2.6.1 Segmento de linha reta digital

Os segmentos digitalizados têm propriedades diferentes de uma linha reta contínua no espaço euclidiano (LARGETEAU-SKAPIN; ANDRES, 2006). Por exemplo, fundamentando no trabalho de Freeman (1970), Li e Loew (1988) demonstraram que um segmento digitalizado não pode ser subdividido infinitamente até um segmento arbitrariamente pequeno, mantendo o seu ângulo de orientação.

Complementarmente, Feschet (2008) citou que segmentos digitalizados são estruturas muito rígidas, cujo formalismo não contempla alterações nos ângulos e escalas dos segmentos digitalizados causadas pelas interferências e ruído que induzem inexatidões não previstas. A rigidez mencionada requer corridas (*runs*) fixas de valores P e Q , com $Q = P + 1$ ou $Q = P - 1$, a fim de atender à propriedade da corda, conforme o exemplo da Figura 2.19.

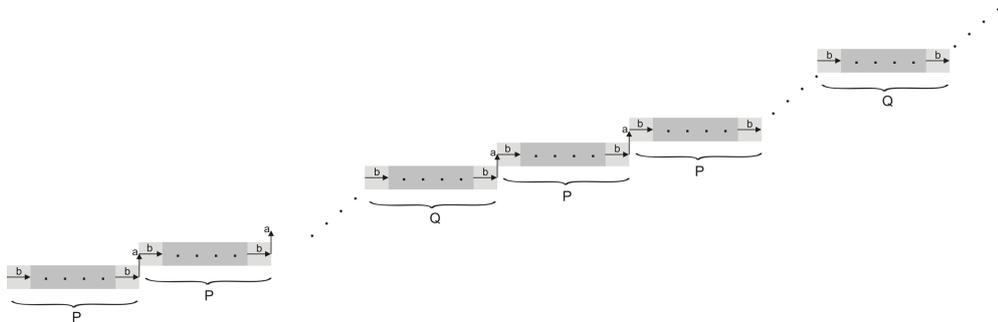


Figura 2.19: Modelo de segmento digitalizado genérico no primeiro quadrante.

A análise sintática de segmentos digitalizados requer poder computacional no nível de linguagens sensíveis ao contexto, inviabilizando a aplicação de formalismo simples, tais como autômatos finitos (FEDER, 1968) (KLETTE; ROSENFELD, 2004b), o que tem conduzido a poucos trabalhos com base em métodos sintáticos para tarefas envolvendo segmentos digitalizados. Os conceitos de Linguagem Regular e autômato finito são equivalentes no sentido que ambos estão associados exatamente a uma mesma classe; ou seja, para cada linguagem regular existe, pelo menos, um autômato finito que a reconhece e vice-versa (ver Lewis e Papadimitriou (1981) para o formalismo de gramáticas, linguagens e autômatos.).

2.6.2 O poder computacional requerido para a análise sintática de segmentos digitalizados

Os métodos sintáticos apresentam a particularidade de introduzir uma análise, a fim de reconhecer a reta original, de acordo com determinado modelo sintático. Pela visão de (BRONS, 1974), define-se uma gramática associada a algum tipo de dispositivo de reconhecimento. Esse reconhecedor é denominado *parser* por ser responsável pela decisão (*parsing*) se uma dada cadeia observada pertence ou não à classe representada pela gramática. Nessa técnica linguística, ou estrutural, os padrões são construídos a partir de blocos e de partículas elementares denominadas primitivas em diferentes composições, da mesma maneira que sentenças são construídas por palavras.

Nos tópicos anteriores foi comentado que o ruído e distorções complicam o processo computacional da análise sintática: além das distorções, influências espúrias são geradas afetando a informação. Porém, mesmo sem considerar o ruído, a própria natureza variável das corridas dos símbolos dos segmentos digitalizados, associada a comprimentos variáveis da escala das retas é um desafio para efetuar a análise sintática.

Por exemplo, Shlien (1983) especifica uma técnica linguística para segmentar arcos em segmentos digitalizados adaptativos apenas com operações de números inteiros seguindo as regras gramaticais sintáticas extraídas das propriedades dos segmentos digitalizados adaptativos. Um analisador verifica as regras relacionadas com uma camada k , e (eventualmente) ativa um parser para a próxima camada $k + 1$. Vários níveis de *parsers* podem ser ativados simultaneamente. Entretanto, dependências de contexto e alterações no ângulo de orientação em segmentos de comprimento arbitrário afetam a estrutura dos códigos das retas digitais, forçando os *parsers* a reverem a sua análise (SHLIEN, 1983). Como resultado, o algoritmo de Shlien (1983) é mais adequado para aplicações onde a representação do arco é livre de ruído.

O poder computacional requerido para a análise sintática de segmentos digitalizados está associado ao das linguagens sensíveis ao contexto, inviabilizando a aplicação de formalismo simples, tais como autômatos finitos (FEDER, 1968) (KLETTE; ROSENFELD, 2004b). A maneira que o autômato finito adaptativo aceita linguagens tipo 1 e 0 é apresentado por Rocha e Neto (2001).

Na literatura, o modelo formal clássico utilizado para a aceitação de linguagens tipo 1 e 0 é a Máquina de Turing. As dependências de contexto encerradas nessas linguagens podem ser solucionadas pelo recurso do autômato finito adaptativo alterar seu próprio conjunto de estados e de regras de transição (ROCHA; NETO, 2001).

Ao delinear-se inicialmente esta pesquisa, a linguagem tipo 0 não faria parte do escopo deste trabalho. Entretanto, constatou-se por estudos subsequentes que o conjunto de estados do autômato finito adaptativo deve ser um conjunto Q não vazio e possivelmente tendendo a infinito, visando estar de acordo com os comprimentos das cadeias dos segmentos digitalizados, os quais podem estar em várias escalas de acordo com o comentado no tópico 2.1. Por esse fato, este trabalho relaciona-se com linguagens tipo 0.

Pela descrição do tópico 2.1, a maneira que o autômato finito adaptativo reconhece linguagens tipo 0 pode ser interpretada da seguinte modo:

- Supondo o autômato finito adaptativo numa determinada configuração k representada por um autômato finito ND_k ;
- Lembrando que, pela Definição 1 a camada AM de $AD = (ND_0, AM)$ é composta de ações adaptativas responsáveis por alterar a estrutura do dispositivo não adaptativo subjacente dinamicamente em resposta aos estímulos;
- Consequentemente, o autômato finito adaptativo altera a sua configuração daquela do passo k para uma nova configuração $k + 1$ (um outro autômato finito) dinamicamente. Tudo se passa como se o mecanismo selecionasse um autômato finito dentre todas as instâncias de autômatos finitos modelados.

2.6.3 Considerações sobre o estado-da-arte

Quanto ao processo de digitalização, a grade de digitalização quadrada é conveniente para demonstrações matemáticas, porém esse caso particular pode ser generalizado, sem dificuldade, a outras grades (por exemplo, grade hexagonal). Adicionalmente, a vizinhança em conectividade-4 pode ser generalizada. Tanto a “quantização de contornos de objetos” quanto o “método de interceptação da grade” se reduzem um ao outro por transformações (DORST; SMEULDER, 1987).

Dentre os principais trabalhos que norteiam esta tese, Bhowmick e Bhattacharya (2007) seguem uma abordagem algorítmica similar à de Freeman o qual define linhas discretas como digitalização de linhas euclidianas. No segundo trabalho, mais recente, Feschet (2008) fundamenta-se na noção de linhas da “geometria discreta aritmética” pesquisada por Reveillès (1991). No terceiro, Debled-Rennesson, Feschet e Rouyer-Degli (2006) utilizam a noção de “segmentos borrados”. A seguir, comentam-se os trabalhos mencionados.

- A solução de Bhowmick e Bhattacharya (2007) apresenta um algoritmo relacionado à abordagem de Freema, que tolera algum nível de ruído em segmentos digitalizados, porém gera grande quantidade de segmentos em arcos irregulares. Isso ocorre por várias razões, uma delas por utilizar heurísticas direcionadas, sem considerar o comprimento total do segmento. Assim, o processamento inicia-se pelo reconhecimento de um conjunto de segmentos digitalizados, para que estágio posterior efetue um refinamento dessa classificação inicial, reconhecendo segmentos digitalizados co-lineares, com base em algum critério. Outra razão é que a detecção de segmentos co-lineares depende do ponto inicial e da direção pré-definida do segmento.

A terceira razão é por considerar um critério adaptativo levando em consideração apenas o ângulo, e não o comprimento do segmento. Portanto, com alterações de escalas, a quantidade de segmentos detectados tenderia a reduzir.

Dentre as vantagens do trabalho, destaquem-se a sua fundamentação nos conceitos básicos da geometria euclidiana, e consequente robustez por se basear nas propriedades de segmentos digitalizados ideais, assim como não requer operações com números em ponto flutuante.

- A abordagem de Feschet (2008) se baseia na noção de espessura da grade (*lattice width*). Utiliza um critério para detectar curvas digitais aproximadamente retas as quais são denominadas segmentos “quase em linha reta” (*quasi-straight*) fundamentado em propriedades aritméticas de fechos convexos (*convex hulls*).

Uma linha discreta aritmética é definida como um conjunto de pontos $(x, y) \in \mathbb{Z}^2$. Com propósito de obter maior flexibilidade das estruturas, o trabalho de Feschet (2008) apresenta a desvantagem, relativamente à abordagem de Bhowmick e Bhattacharya (2007), na complexidade envolvida em calcular a espessura da grade em todas as direções e também a espessura da grade do conjunto de pontos digitais na direção ortogonal. Então, calcula-se, para cada direção, a denominada “razão da direção” (*ratio of direction*) entendida como a razão entre o menor e o maior valor de espessura de grade. Para todas as direções possíveis, escolhe-se aquela com a menor proporção, definindo-se antecipadamente um limite. Portanto o conjunto é considerado como um segmento quase em linha reta (*quasi straight*) se, e somente se, a menor razão da direção for menor do que o limite.

- O trabalho de Debled-Rennesson, Feschet e Rouyer-Degli (2006) utiliza a mesma base da geometria discreta aritmética introduzindo a noção de “segmentos borrados” (também denominados *fuzzy*) dependentes da definição aritmética de linhas discretas. Esse trabalho utiliza a noção de ordem de segmentos borrados a fim

de diferenciar as espessuras aritméticas das linhas delimitadoras uma vez que qualquer espessura aritmética suficientemente grande pode conter o conjunto de pontos digitais. Assim, linhas discretas que contenham o conjunto de pontos podem ser comparadas com relação às respectivas ordens conduzindo a uma classificação dessas linhas. A detecção de um segmento borrado com espessura v equivale, portanto, ao cálculo da distância vertical do conjunto convexo.

A partir do algoritmo inicial, um algoritmo para decomposição em polígonos borrados pode ser implementado onde a largura v é fixada antecipadamente. O princípio desse algoritmo é o seguinte: iniciar em algum ponto do arco e executar um reconhecimento de segmentos com espessura v . O último ponto reconhecido é considerado o primeiro ponto de computação para reconhecimento seguinte até que todos os pontos da curva tenham sido examinados.

O trabalho mostra que este conceito de ordem é suficientemente geral para garantir que os parâmetros de segmentos possam ser efetivamente computados, porém perde qualquer conexão com a aritmética, tendo levado a aprimoramentos descritos em Feschet (2006) e Feschet (2008).

2.6.4 Vantagens e desvantagens da representação adaptativa

Segue abaixo um breve resumo das principais vantagens e desvantagens da representação adaptativa de segmentos digitalizados:

- Vantagens:
 - Possibilita considerar, na representação, tanto o ângulo quanto o comprimento de segmentos. Consequentemente, um segmento digitalizado adaptativo conota um conjunto de segmentos de arcos digitais, os quais podem variar não apenas em comprimento, teoricamente até infinito; mas também podem ser arcos aproximadamente retos nas proximidades de uma cadeia específica;
 - O poder expressivo de representar atributos de segmentos digitalizados. Dentre esses, destacam-se a capacidade de representar as tolerâncias, a escalabilidade, os erros causados por desvios em ângulo ou em comprimento dos segmentos mencionados, resultando em estruturas mais flexíveis.
 - Mesmo que a aceitação de uma cadeia inicie por uma determinada configuração de autômato finito adaptativo definida por informações teóricas, pode ser alterada dinamicamente conforme os estímulos recebidos.

- Introduz o autômato finito adaptativo para representação de retas digitais, incluindo a armazenagem de arcos digitais, com possibilidade de compressão de dados. Uma das aplicações de segmentos digitalizados é a representação de um arco qualquer por segmentos de reta. A representação adaptativa de segmentos digitalizados permite que o arco seja representado em qualquer escala por um único autômato.
- É uma alternativa relativamente simples e intuitiva comparando-se com as abordagens existentes, apresentando capacidade de aprendizagem, além de ser computacionalmente poderosa.
- Desvantagens: Implicações a serem avaliadas pela complexidade resultante da inclusão de ações adaptativas a formalismos clássicos : possível incremento em custos, incremento em complexidade computacional, alteração de paradigmas computacionais.

3 ANÁLISE ESTRUTURAL

Nos capítulos anteriores tratou-se do processo de digitalização de um subconjunto pertencente a R^2 associado a uma reta. Esse processo envolve encontrar a correspondente representação do subconjunto em R^2 para uma representação Z^2 ; ou seja, aproximando o primeiro por um conjunto de coordenadas de pontos em uma grade associadas a números inteiros. Há diferentes maneiras de fazer isso, mas, em todas elas, o objetivo é sempre selecionar *pixels* que mais se aproximam do conjunto em R^2 . Também foi comentado sobre dois métodos principais para a representação de segmentos digitalizados: o método de Freeman e o método com base na geometria discreta aritmética.

Um outro método, aplicado neste item, é a representação das linhas digitais com base em frações contínuas, estudado por Brons (1974), que resultou num algoritmo válido apenas para retas com inclinações (a tangente do ângulo do segmento de reta com relação ao eixo x) de números racionais. O estudo da modelagem de retas por frações contínuas tem sido continuamente pesquisado, com vários desdobramentos após o trabalho de Brons (1974), descritos em Klette e Rosenfeld (2004b).

Um dos aspectos de interesse na representação por frações contínuas é o enfoque matemático, conveniente para propiciar modelos, avaliar erros e aproximações no processo de digitalização de retas partindo das correspondentes inclinações. A inclinação de um segmento digitalizado é dada pela tangente do ângulo do segmento com o eixo x . Por exemplo, existe uma correspondência entre inclinação e periodicidade do *chain code*, tal que retas com inclinações representadas por números racionais são periódicas; enquanto retas com inclinações representadas por irracionais não ocorre a periodicidade, comentado na página 312 de Klette e Rosenfeld (2004a). Entretanto, mesmo sem considerar distorções e ruído, os modelos digitais têm que atender às condições da grade, o que envolve procedimentos de amostragem e aproximações, tornando menos importante a questão de inclinações racionais ou irracionais para as situações desta pesquisa.

Como introdução ao assunto, apresenta-se a seguir a série de Farey.

A série de Farey F_N de ordem $N \geq 1$ é a seqüência ascendente de frações irredutíveis m/n entre 0 e 1, cujos denominadores não excedam N . A fração m/n pertence a F_N se, e somente se:

- $0 \leq m \leq n \leq N$
- O máximo divisor comum de m e n é igual a 1: $mdc(m, n) = 1$.

Um exemplo é a seguinte série:

$$F_5 = \{0/1, 1/5, 1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 1/1\}.$$

Uma característica interessante é uma correspondência biunívoca entre a série de Farey de ordem N e o segmento digital que passa pela origem numa grade $N \times N$, associando a série de Farey $F(n)$ e todos os segmentos digitalizados $y = \alpha x$ de comprimento n , passando pela origem no plano de coordenadas x e y , em que os pontos de intersecção na grade estão relacionados aos pontos de intersecção das retas com o eixo y (ROTHSTEIN; WEIMAN, 1976).

A seguir, apresenta-se uma breve análise estrutural dos segmentos digitalizados por meio dos formatos das respectivas cadeias, apresentadas como exemplos. Sem perda de generalidade, considera-se o *chain code* representando uma linha contínua de ângulo de orientação ϕ com o eixo positivo x tal que $\phi \in [0, \pi/4]$ (indica que ϕ pertence ao intervalo fechado entre 0 até $\pi/4$ radianos) (KIRYATI; KÜBLER, 1995). De acordo com o tópico 1.1.12, nas cadeias correspondentes aos segmentos digitalizados o símbolo a ocorre isoladamente enquanto o símbolo b ocorre em corridas ou agrupado, atendendo às propriedades *Prop1* e *Prop2*.

A fim de apresentar uma analogia com o caso contínuo, a Figura 3.1 mostra um conjunto de N pontos representados de três maneiras conforme a reta ou as curvas contínuas que passam aproximadamente pelos pontos. A reta e as curvas mencionadas correspondem a modelos, mais ou menos precisos, do conjunto de N pontos, representáveis por polinômios de determinada ordem. Tendo em vista essa analogia, a finalidade deste tópico é obter modelos de retas digitais levando em conta as aproximações envolvidas no processo de ajuste dos símbolos das cadeias, levando em conta as condições da grade de amostragem, a qual corresponde a números inteiros num reticulado.

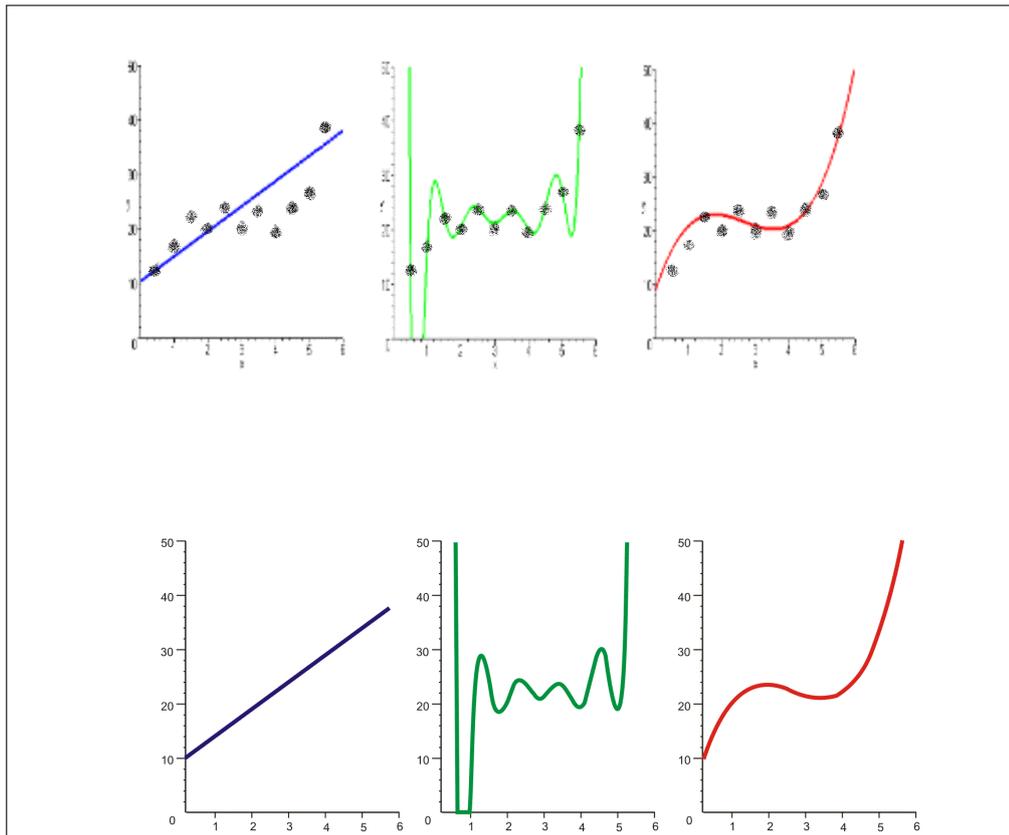


Figura 3.1: Exemplo de processo de modelagem. Nas três figuras da parte superior são dados N pontos: (a) À esquerda, os pontos são aproximados por uma reta; (b) No meio, são aproximados por uma curva; (c) À direita, são aproximados por uma segunda curva. As três figuras da parte inferior mostram a reta e as curvas resultantes das aproximações (adaptado de Corona (2008))

Na Figura 3.1 o modelo mais simples é uma linha reta. No entanto, a linha reta pode não expressar tão bem o posicionamento dos N pontos. Em contrapartida, os N pontos podem ser representados usando um polinômio de grau $N - 1$ que se ajustaria melhor aos dados, ao custo de maior complexidade nos processos envolvidos.

Além disso, ainda na Figura 3.1, se existirem outros grupos de N pontos, resulta que o polinômio de grau N previamente selecionado pode não ser o mais adequado para o novo conjunto. O problema torna-se mais complicado se a quantidade de pontos for variável. Por isso, ao invés de alterar o polinômio, normalmente seleciona-se um polinômio específico que possa resultar em um bom compromisso entre ajuste razoável aos vários dados e nível de complexidade. Essas mesmas questões fazem parte do caso digital.

3.1 Ordem dos modelos

Conforme Li e Loew (1988), dependendo do grau de precisão requerido, derivam-se fórmulas para o ângulo de orientação dos segmentos digitalizados bem como modelos matemáticos gerais ao se aproximar um segmento digitalizado a um segmento euclidiano. Os modelos são paramétricos, calculados diretamente em função do ângulo de orientação ϕ do segmento euclidiano. Para tais modelos, o conceito de unidade de segmento é importante:

Definição 47 *Unidade de segmento digital em linha reta:* (abreviado unidade de segmento) é o menor segmento possível em que um segmento digitalizado pode ser subdividido a fim de manter o correspondente ângulo de orientação (LI; LOEW, 1988).

As premissas de Li e Loew (1988) para especificação dos modelos foram:

- O segmento digitalizado deve atender à propriedade da corda;
- O segmento digitalizado não pode ser subdividido indefinidamente, de acordo com a Definição 47;
- A estrutura do arranjo de *pixels* do segmento digitalizado depende exclusivamente da correspondente inclinação.

A inclinação no plano, definida pela tangente do segmento euclidiano no primeiro quadrante, pode ser expressa pela fração contínua da Expressão 3.1 aproximada a um modelo nos moldes da Definição 48.

$$SL = \frac{B}{A} = \frac{1}{P \pm \frac{1}{M \pm \frac{1}{K \pm \dots}}}; \quad (3.1)$$

em que os sinais positivos e negativos são para acelerar a convergência da fração, reduzindo o número de termos da mesma. Para o entendimento da lei de formação da Expressão 3.1, primeiramente o segmento de reta implica no arranjo de $P + \frac{a}{B}$ *pixels* em cada linha, o que não pode ser feito em uma imagem digital se $a \neq 0$. Por isso, o ajuste do segmento de reta à grade se faz com a *pixels* nas B colunas, tão uniformemente quanto possível a fim de manter o ângulo de inclinação ao longo do segmento de reta o mais uniforme possível. O ajuste é feito através do aumento de um (se $a > 0$) ou pela diminuição de 1 (se $a < 0$).

(LI; LOEW, 1988). Por exemplo, a Figura 3.2 mostra um exemplo de segmento que se ajusta com exatidão à grade resultando apenas em P da Expressão 3.1. Tendo em

vista que estão envolvidos apenas números inteiros na maneira que as amostragens dos segmentos se ajustam à grade, surgem os diversos modelos indicados pelas operações de \pm da Expressão 3.1.

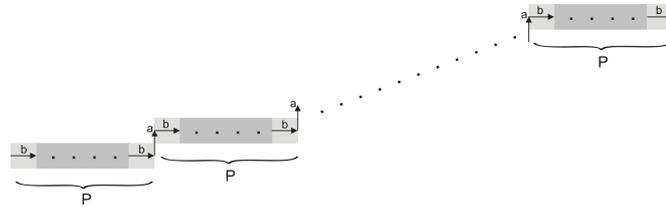


Figura 3.2: Exemplo de um modelo contendo apenas o valor P da Expressão 3.1.

O número de termos da Expressão indicam a ordem do modelo da Definição 48 em que A, B (com $B \leq A$) e P são números inteiros, enquanto M, K, \dots são inteiros não negativos.

Definição 48 *Ordem de modelo de reta digital:* (abreviado modelo de reta) a ordem de um modelo de reta digital corresponde ao número de termos da Expressão 3.1 (LI; LOEW, 1988).

Assim, a Expressão 3.1 pode se apresentar de várias formas atendendo à lei de formação dos termos pelas propriedades *Prop3* e *Prop4* ao se ajustar o segmento digitalizado à grade, tais como:

$$SL = \frac{B}{A} = \frac{1}{P}; \quad (3.2)$$

$$SL = \frac{B}{A} = \frac{1}{P \pm \frac{1}{M}}; \quad (3.3)$$

$$SL = \frac{B}{A} = \frac{1}{P \pm \frac{1}{M \pm \frac{1}{K}}}. \quad (3.4)$$

E assim por diante, dependendo dos valores de A, B , ou dos requisitos de precisão. As Expressões 3.2, 3.3 e 3.4 são denominadas frações contínuas de primeira ordem, de segunda ordem e terceira ordem, respectivamente, correspondendo à ordem do modelo.

Com o conceito de unidade de segmento, os dois modelos de primeira e segunda ordem são considerados principais para descrever o arranjo no ajuste dos padrões de *pixels* de segmentos digitalizados na grade. Note que os ângulos das unidades de segmento, denominados θ_U , podem variar no segmento digitalizado. A Definição 49 comenta sobre os ângulo de orientação principal do segmento digitalizado, que é o ângulo que se destaca no segmento dentre a distribuição de θ_U .

Definição 49 *Ângulo de orientação principal:* entende-se como ângulo de orientação principal do segmento digitalizado, denominado θ_S , como aquele ângulo que se destaca, por algum critério, dentre a distribuição de ângulos das unidades de segmento individuais do segmento. Por exemplo, Lee e Fu (1982) denominam a direção (o ângulo) correspondente ao símbolo que ocorre isoladamente de direção de transição, e a direção correspondente ao outro símbolo de direção principal (LEE; FU, 1982).

3.1.1 Modelo de primeira ordem

O primeiro modelo é denominado de primeira ordem, porque a inclinação é uma fração contínua de primeira ordem. Nesse modelo, o segmento digitalizado da Figura 3.3 atravessa $B = 1$ linhas e $A = P$ colunas, resultando $SL = B/A = 1/P$; onde P é um número inteiro denominado fator de inclinação de primeira ordem. P como número inteiro significa que os *pixels* do segmento de reta são uniformemente distribuídos, tal que cada linha é um USLS que contém P pixels consecutivos, com o arranjo dos padrões do segmento digitalizado do tipo $PPP...PP...P$.

Para o modelo de primeira ordem, as cadeias S_U das unidades de segmento seguem o formato da Expressão 3.5 em que m é o mesmo para toda unidade de segmento do segmento digitalizado.

$$S_U : b^m a; m \geq 1. \quad (3.5)$$

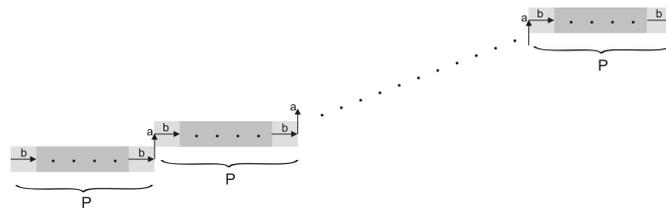


Figura 3.3: Modelo de primeira ordem, com P constante.

Com $SL = B/A = 1/P$ é evidente que no modelo de primeira ordem o ângulo de orientação principal do segmento digitalizado θ_S é idêntico ao ângulo de cada unidade de segmento θ_U , ou seja $\theta_S = \theta_U = \arctan(1/P)$.

3.1.2 Modelo de segunda ordem

No modelo de segunda ordem P é um número inteiro mais próximo de A/B , $P = \lfloor A/B \rfloor$. Nesse modelo, o segmento digitalizado tende a ajustar os *pixels* em B linhas,

tão uniformemente quanto possível, a fim de manter a inclinação global. No modelo de segunda ordem da Figura 3.4, o arranjo dos padrões do segmento digitalizado é do tipo $PPP\dots QPP\dots PQ$ em que o ajuste é efetuado por intermédio de Q , o qual pode ter apenas duas possibilidades: $Q = (P + 1)$ ou $Q = (P - 1)$. Ou seja, o ajuste é efetuado diminuindo-se P de uma unidade, ou aumentando-se P de uma unidade.

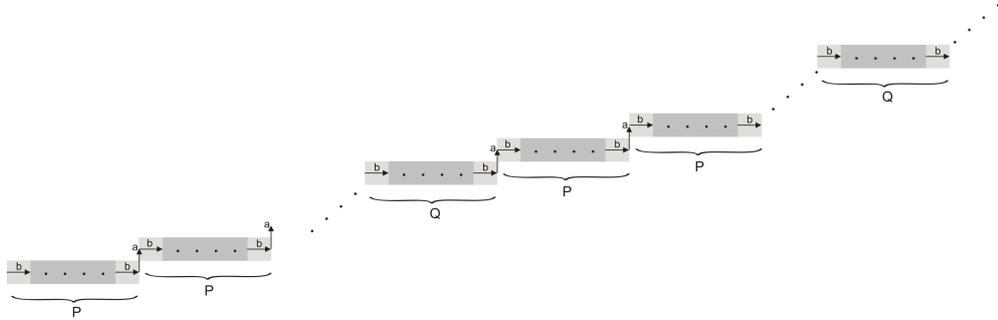


Figura 3.4: Modelo de segunda ordem, com P e Q constantes.

No modelo de segunda ordem, as cadeias S_U das unidades de segmento são de dois tipos:

$$S_U : \begin{cases} b^m a, & e; \\ b^{m+1} a; m \geq 1 & \text{ou } b^{m-1} a; m \geq 2. \end{cases}$$

Nesse modelo, o ângulo de orientação principal é $\theta_S = \arctan(1/P)$, aquele que se destaca, com inclinação $1/P$. Entretanto ocorrem unidades de segmento com inclinação $1/Q$ tão espaçadas quanto possível. Desta forma, o ângulo de orientação ϕ com o eixo positivo x da linha contínua, que deu origem à codificação, estará na faixa $\phi \in [\arctan(1/(P+1)), \arctan(1/P)]$ (supondo $Q = P+1$) (KIRYATI; KÜBLER, 1995).

Conforme comentado no item ?? sobre o levantamento do estado-da-arte, Rosenfeld (1974) estudou os possíveis comprimentos das corridas (*run lengths*) dos símbolos de um segmento digitalizado, bem como das corridas de corridas (*runs of runs*) ao longo da direção principal. Rosenfeld (1974) demonstrou que só pode haver dois comprimentos possíveis diferentes dessas corridas, que são dois números inteiros consecutivos, sendo que um deles sempre ocorre isoladamente. Este padrão de números consecutivos é considerado em soluções para a influência de ruído, descritas a seguir.

3.1.3 O efeito do ruído

Por outro lado, Lee e Fu (1982) mostraram exemplos de segmentos digitalizados que violam a regularidade implícita na propriedade da corda, comentando que, na prática, a propriedade *Prop3* não é atendida em arcos digitais. O mais razoável é esperar uma

pequena variação nas corridas, expressas em um nível de tolerância, porém sempre mantendo a inclinação global do segmento digitalizado fixada para definir uma reta aproximada. Portanto, o critério utilizado se concentrou em cadeias que satisfaçam as duas primeiras propriedades da conjectura, denominados “códigos monotônicos”, pois representam arcos digitais que, ou são ascendentes ou são descendentes, relativamente às coordenadas x e y .

Similarmente a Lee e Fu (1982), Chaudhuri e Pal (1997) relaxaram a propriedade da corda, considerando como segmento digitalizado válido qualquer arco que apresente corridas consecutivas nas direções principais, atendendo *Prop4*. Dado um arco que apresente corridas do símbolo que ocorre em grupos de valor P , Chaudhuri e Pal (1997) aceitam que esse arco é um segmento digitalizado desde que apresente as corridas variando, no máximo, entre P e Q , descritas a seguir:

- P e $Q = (P - 1)$, ou
- P e $Q = (P + 1)$.

A motivação Chaudhuri e Pal (1997) foi que as corridas consecutivas eram mais fáceis de detectar do que a propriedade da corda. Tendo em vista que cenários ruidosos tendem a distorcer os segmentos digitalizados, alterando a inclinação das unidades de segmento em determinadas posições, esse processo atua como uma espécie de filtragem de ruído ao rejeitar arcos fora da faixa de aceitação. Na vizinhança-4, as condições seriam as seguintes (as propriedades *Prop1* e *Prop2* são as mesmas anteriores, reproduzidas apenas com um texto diferente):

- *Prop1*: Existem corridas de *pixels* de, no máximo, duas direções, as quais diferem de 90° ;
- *Prop2*: Para as corridas nas duas direções, os comprimentos das corridas de uma das direções é sempre um (aparece isolado);
- *Prop4*: Quanto à outra direção, referente ao símbolo que ocorre em grupos (não isolado), as corridas correspondentes podem ocorrer com apenas dois valores, os quais diferem de uma unidade (por exemplo, P e $P + 1$).

Mais recentemente, Bhowmick e Bhattacharya (2007) aproximam um arco qualquer C como sendo um segmento digitalizado, desde que as corridas estejam em um intervalo $[p, q]$, onde p e q são parâmetros sujeitos à condição seguinte (existe uma condição

adicional, incluindo as extremidades dos arcos): $(q - p) \leq d$. O valor de d é escolhido heurísticamente, por exemplo $d = \lfloor (p + 1)/2 \rfloor$, $d = \lfloor (p + 3)/4 \rfloor$, e assim por diante. Essa escolha heurística é viável, desde que o cálculo envolva apenas o domínio dos números inteiros, e não se produza qualquer segmento digitalizado indesejável (não visualmente correto).

Portanto, no trabalho de Bhowmick e Bhattacharya (2007), $[p, q]$ é a faixa de corridas possíveis para as cadeias das unidades de segmento, definindo-se por essa faixa o nível de aproximação de C para um segmento digitalizado. Fixando $m_{min} = p$ and $m_{max} = q$, o ângulo de orientação das unidades de segmento pertencerá ao intervalo $\theta_U \in [\arctan(1/m_{max}), \arctan(1/m_{min})]$. As corridas horizontais variam de m_{min} para m_{max} , alternando-se com o símbolo vertical solitário.

3.2 Comprimento

Conforme Kiryati e Kübler (1995), a principal dificuldade em estimar o comprimento de arcos (ou perímetros de arcos), é causada pelo processo de digitalização, sempre conduzindo a erros. Considerando que informação é perdida nesse processo, a tarefa de estimar o comprimento da reta contínua partindo do segmento digitalizado requer incorporar algum conhecimento *à priori*.

Pelo trabalho de Freeman (1970), o comprimento de um arco discreto representado pela cadeia $s_1..s_i..s_n$ é dado pela Expressão 3.6.

$$l_F = (v + h) + s(2)^{\frac{1}{2}}, \quad (3.6)$$

onde, de acordo com as primitivas existentes na cadeia mencionada:

- v é a quantidade de primitivas na vertical, correspondente aos símbolos a e c ;
- h é a quantidade de primitivas na horizontal, correspondente aos símbolos b e d ;
- s é a quantidade de primitivas na diagonal (não utilizadas nesta pesquisa).

O problema é que a Expressão 3.6 fornece um valor superestimado, requerendo-se a multiplicação de l_F por algum fator ψ

$$l_E \approx \psi \times l_F, \quad (3.7)$$

que compense na média (sobre todos os possíveis comprimentos e ângulos) pelo valor em excesso causado pela codificação (KULPA, 1977). Os coeficientes ótimos ψ (existem estimadores que utilizam mais de um fator), associados a uma taxa de erro em porcentagem, são comumente calculados de modo a resultar em zero de erro médio e mínimo erro médio quadrático para segmentos digitalizados uniformemente distribuídos em todas as orientações. .

Por exemplo, para inclinações entre $[0, \pi/4)$, Kulpa (1977) calcula um fator $\psi = 0.948$, resultando $l_E \approx 0,948 \times l_F$ com erros máximos: $+2,5\%$; $-5,3\%$. Esse mesmo trabalho propõe estimadores da forma $\hat{L} = \psi_c N_c + \psi_o N_o$, em que N_c e N_o representam a quantidade de pontos em cantos e pontos não em cantos, respectivamente. ψ_c e ψ_o são os respectivos pesos.

O assunto é aprofundado em outros trabalhos, dentre esses, Dorst e Smeulder (1987) comparam estimadores, inclusive estimadores mais simples, apresentando recomendações. Koplowitz e Bruckstein (1989) definem uma classe geral de estimadores de perímetros, do tipo

$$\hat{L} = \sum_{\text{todo } P_i} \psi\{\text{prop}P_i\}, \quad (3.8)$$

onde P_i é um *link* no *chain code*, $\text{prop}P_i$ é uma lista de propriedades associadas a P_i e ψ é uma função que classifica e designa pesos aos *links* do *chain code*.

4 ESTRUTURAS ADAPTATIVAS DA TESE

Garitagoitia (2003) comenta que “semelhança” é um termo *fuzzy*, enfatizando que, quando os erros inerentes a determinados cenários não seguem um comportamento conhecido, é mais viável utilizar modelos apresentando a melhor “semelhança” com o modelo ideal, exemplificando com métodos *fuzzy* aplicados à teoria clássica das linguagens e autômatos.

Posteriormente, descreve um autômato *fuzzy* capaz de avaliar a similaridade entre duas cadeias, a cadeia observada de entrada e a cadeia correspondente ao modelo ideal. Começa por definir um autômato finito que aceita a cadeia correspondente ao modelo ideal. Em seguida, o autômato finito é modificado de modo a incluir todas as possíveis instâncias da cadeia observada, devido aos possíveis erros, levando o autômato modificado a reconhecer a cadeia ideal, mesmo com os erros na cadeia de entrada. Tal autômato finito resultante é um autômato *fuzzy*, onde os seus estados são *fuzzy sets*, definidos sobre um universo de estados, enquanto que as suas transições são definidas por operações *fuzzy* apropriadas envolvendo os estados *fuzzy*.

A avaliação de semelhança com o modelo ideal por esta tese é efetuada por um autômato finito adaptativo capaz de avaliar a similaridade (dentro de uma certa tolerância) entre duas cadeias, o segmento digitalizado observado e a cadeia reta, em que esta última é a cadeia definida pelo modelo de segmento digitalizado conforme a propriedade da corda. A configuração inicial do autômato finito adaptativo é um autômato finito ND_0 que aceita a cadeia reta. O autômato finito adaptativo mencionado incorpora “ações adaptativas”, que levam em conta todas as possíveis instâncias da cadeia observada, devido aos possíveis erros. Tais ações adaptativas têm a finalidade de alterar a configuração inicial para uma nova configuração (um outro autômato finito), consequentemente levando o autômato finito adaptativo a representar as diferentes instâncias do modelo ideal, considerando os erros envolvidos.

Essa questão de erros envolvidos em análise sintática foi estudada principalmente na recuperação de erros em compiladores, que se enquadram no enfoque desta pesquisa, comentado a seguir.

4.1 Prolegômenos

Backhouse (1979) é um dos vários autores que têm relatado métodos de recuperação de erros, em especial na área de compiladores. Apresenta-se a seguir o enfoque desta pesquisa sobre recuperação de erros.

4.1.1 Recuperação de erros

Supondo uma cadeia submetida a um processo de análise sintática por um reconhecedor, denomina-se recuperação de erros ao método de tratamento de erros, consistindo em termos gerais nas seguintes etapas:

- O analisador se mantém no processo de consumo de símbolos (ou caracteres) de entrada, até que um símbolo errado seja identificado;
- Isolado o erro, ativa-se algum mecanismo de recuperação de erros;
- O mecanismo de recuperação normalmente modifica ou adapta o reconhecedor para que este se mantenha consumindo um ou mais símbolos, de modo que a análise sintática especificada seja retomada em estágio posterior da sequência de entrada.

De modo geral, os mecanismos de recuperação de erros costumam atuar localmente por inserções, exclusões ou mudanças de símbolos nas posições onde os erros são detectados. Ocorrendo erros, o mecanismo de recuperação atua localmente, alterando o contexto da análise provisoriamente, a fim de aceitar alguns símbolos fora do especificado. Posteriormente, quando uma sequência válida da cadeia de entrada passar a ocorrer, a análise especificada inicialmente é retomada. Uma das principais restrições a esses mecanismos de recuperação locais é a exigência de que o usuário forneça informações adicionais sobre o processo de reparo.

É evidente a analogia entre a recuperação de erros na análise sintáticas de segmentos digitalizados com outros processos sintáticos gerais. Entretanto, é possível concluir

que não ocorre uma igualdade total ou completa ao se retomar o trabalho de Garitagoitia (2003), referenciado no início deste capítulo. Garitagoitia (2003) abstraiu todos os erros no processo de digitalização, independentemente da sua origem, nos denominados erros de edição: substituição, eliminação e inserção de símbolos. Entretanto, o mesmo procedimento não é viável para o caso de segmento digitalizado. A primeira razão é que existe, previamente, um grau de incerteza representado pelo domínio da pré-imagem. A segunda, determinados erros de edição podem ser aceitos numa faixa, enquanto outros tipos de erros, se aceitos irrestritamente, iriam contrariar a estrutura das linhas retas, expressadas pelas propriedades *Prop1* a *Prop4* (ver tópico 3.1.3):

- *Prop1*: Existem corridas de *pixels* correspondentes a duas direções, no máximo, as quais diferem de 90° ;
- *Prop2*: Para as corridas nas duas direções, os comprimentos das corridas de uma das direções é sempre um (aparece isolado);
- *Prop3*: As ocorrências sucessivas do símbolo isolado, são tão uniformemente espaçadas quanto possível entre códigos do outro valor, que ocorre em grupos ou corridas *runs*.
- *Prop4*: Relativamente à direção associada ao símbolo que ocorre em grupos (não isolado), as corridas correspondentes podem ocorrer com apenas dois valores, os quais diferem de uma unidade (por exemplo, P e $P + 1$).

A recuperação de erros desta tese envolve os conceitos comentados no tópico 3.1.3 quanto a não flexibilizar *Prop1* e *Prop2*, porém em aceitar erros nos critérios de *Prop3* e *Prop4*, desde que dentro de uma tolerância, sem que ocorra qualquer interferência do operador.

Complementando o adendo A afeto à metodologia envolvida neste trabalho, o propósito desta pesquisa em recuperação de erros compreendeu estudos introdutórios, apresentados em Barros Neto, Hirakawa e Massola (2008), sobre como modelar, por meio do autômato finito adaptativo, linguagens do tipo

$$L = \{a^{m \pm \alpha} b^{n \pm \beta} c^{m \pm \delta} d^{n \pm \gamma} : m \geq 1; n \geq 1\},$$

em que $\alpha, \beta, \delta, \gamma$ são pequenos erros correspondentes às influências espúrias. Esses estudos preliminares foram cumpridos a fim de incorporar a incerteza nos modelos das linguagens, sem autômatos probabilísticos (VIDAL, 2005) (o que não impede de integrar autômatos probabilístico ao modelo proposto, no futuro). Dessa forma, valores de

tolerância, formulados por inequações, puderam ser associados aos comprimentos das sub-cadeias embasando a pesquisa com segmento digitalizado adaptativo.

Essas inequações realçam uma característica comum desta pesquisa com a abordagem analítica, conceituada nas linhas discretas aritméticas, quanto à inevitabilidade em pré-definir critérios e heurísticas representando uma faixa de valores, também formuladas por inequações, definidoras de um conjunto de retas, do qual faz parte a reta procurada, apresentando certa analogia com o descrito em 2.3.1.1.

4.1.2 Técnica utilizada em recuperação de erros

No que se refere a técnicas de recuperação de erros neste trabalho, a fim de exibir características comuns ao que está sendo aludido, relembre-se que, frequentemente, é conveniente representar os números reais em uma determinada circunferência C , e não em uma linha reta, como habitual.

Em especial, a partir da circunferência C de comprimento unitário, ao definir-se um ponto origem arbitrário, representa-se um ponto qualquer P pela sua distância medida em volta da circunferência, no sentido anti-horário (esse sentido é por definição). Desse modo, todos os números inteiros serão representados relativamente ao mesmo ponto origem, enquanto os números que difiram por um número inteiro terão o mesmo ponto representativo do círculo. A divisão do círculo pode ser a partir da série de Farey, na forma de *spyrographs* descritos na página 326 de Klette e Rosenfeld (2004a) e mostrado na Figura 4.1 em que a seta D indica o sentido de rotação a partir da origem. Pela Figura 4.1 é possível uma correspondência entre a série de Farey $F(n)$ e todos os segmentos digitalizados $y = \alpha x$ de comprimento n , passando pela origem no plano de coordenadas x e y , em que os pontos 0, 1, 2, 3 e 4 estão associados aos pontos de intersecção das retas com o eixo y .

As técnicas de recuperação de erros de segmentos digitalizados desta abordagem empregam um dispositivo similar aos *spyrographs* na forma de *loops* adaptativos, tal que, nesses *loops*, a circunferência mencionada é representada por estados do autômato finito adaptativo, os quais são percorridos ciclicamente pelo autômato.

Por questão didática, as técnicas delineadas a seguir estão segregadas por dois tipos de erros característicos dos segmentos digitalizados: variações em ângulo e erros em comprimento.

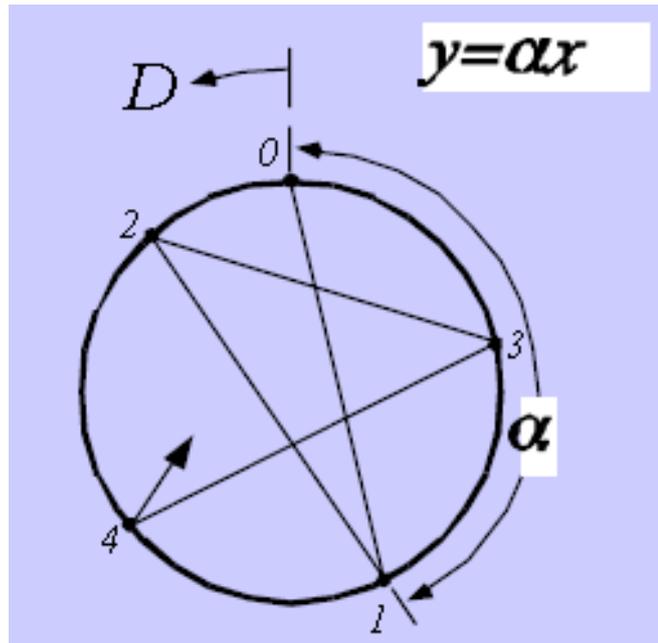


Figura 4.1: Esquemática de *Spyrograph* correspondente a uma reta que passa pela origem (adaptado de Klette e Rosenfeld (2004a)).

4.2 Variações em ângulo

De acordo com o tópico 3.1, supondo inicialmente um segmento digitalizado codificado por uma cadeia S de comprimento n primitivas, $|S| = n$. Tal cadeia é composta da concatenação de λ sub-cadeias, onde $n > \lambda$ e $1 \leq i \leq \lambda$, sendo que cada sub-cadeia é uma unidade de segmento U de S da seguinte forma

$$S : U_i ; i = 1, 2, \dots, \lambda. \quad (4.1)$$

Entretanto, casos extremos não estão previsto na Expressão 4.1, por exemplo situações em que segmento digitalizado apresenta $\lambda = 0$ ou casos em que a primeira e a última unidade de segmento (U_1 e U_λ) estão truncadas, ou seja, fora da estrutura do ângulo θ_S de orientação do segmento digitalizado global. Essas situações extremas são consideradas no item 4.2.1.1.

Localmente, cada unidade de segmento é também um segmento digitalizado elementar que apresenta um ângulo e um comprimento “locais”. Portanto, para cada unidade de segmento, definida pelo código isolado de *Prop2*, basta o segmento digitalizado adaptativo correspondente percorrer um *loop* adaptativo, cuja quantidade de estados é variável adaptativamente em função das corridas do símbolo que ocorre em grupos, consumindo os símbolos da unidade de segmento.

4.2.1 Implementação de segmento digitalizado adaptativo

4.2.1.1 Unidades de Segmento Inicial e Final

Os extremos de um segmento digitalizado hipotético podem estar truncados ou completamente fora do modelo estrutural. No primeiro caso, o segmento digitalizado deve ser incorporado ao modelo adaptativo, e rejeitado no segundo. A Figura 4.2 mostra um autômato finito adaptativo que modela a U_1 pertencendo ao conjunto (esse autômato finito adaptativo é um exemplo de contador) $\{a^n b : n = 3, 4, 5\}$. O parâmetro r_4 é o último estado de uma sequência conexa (ver exemplos de sequências conexas e de contador no item 2.1.3) iniciando no estado r . A partir dessa sequência conexa, o autômato finito adaptativo pode remover até quatro transições em vazio por intermédio da função adaptativa RA apresentada no Quadro 3, ou seja, cada vez que RA é ativada pelo *token* a , uma das transições em vazio da sequência conexa é removida. Além disso, qualquer *token* b conduz o autômato finito adaptativo para o estado final; desde que não sejam recebidos mais do que quatro *tokens* a . Uma estrutura para a última unidade de segmento seria bastante semelhante.

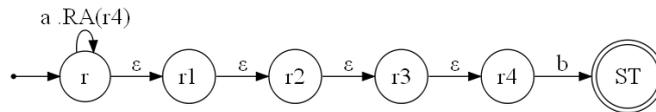


Figura 4.2: Autômato finito adaptativo para modelar a primeira unidade de segmento de um segmento digitalizado adaptativo

$$\begin{aligned}
 & RA(\rho) \{vr1, vr2, vr3, vr4, vr5, vr6 : \\
 & - [(vr6, \varepsilon) - > \rho] \\
 & - [(\rho, vr1) : vr2 - > vr3 : vr4] \\
 & + [(vr6, vr1) : vr2 - > vr3 : vr4] \\
 & - [(vr5, a) - > vr5 : RA(\rho)] \\
 & + [(vr5, a) - > vr5 : RA(vr6)] \}
 \end{aligned}$$

Quadro 3: Função adaptativa paramétrica RA, do autômato finito adaptativo da Figura 4.2

4.2.1.2 O Segmento digitalizado adaptativo

A representação das diferentes instâncias do modelo ideal, afetado pelas variações em ângulo, requer que o segmento digitalizado adaptativo atue numa faixa de ângulos. Este tópico ilustra a modelagem de segmentos digitalizados adaptativos, exemplificando com as unidades de segmento U_i pertencendo ao conjunto $\{a^n b : n = 3, 4, 5\}$ do primeiro quadrante. Nesse conjunto, o ângulo local θ_U relativo ao eixo x corres-

pondente às U_i de um segmento digitalizado S pode variar na faixa $\arctan(3) \leq \theta \leq \arctan(5)$: $\theta_S \in \arctan(3) \leq \theta \leq \arctan(5)$.

A Figura 4.3 mostra a configuração inicial do autômato para modelar a U_1 , mesmo sendo truncada (a estrutura para o caso de U_λ ser truncada é similar e não está representada). As funções adaptativas do autômato finito adaptativo estão descritas no Quadro 4. Com o primeiro *token* b consumido, a função adaptativa B é ativada, a qual remove as transições da configuração inicial, alterando a topologia do autômato para a da Figura 4.4.

A seguir, para cada unidade de segmento, o autômato finito adaptativo percorre um ciclo de estados $u, u_1, u_2, u_3, u_4, u_5, u$ formando um *loop*. A função adaptativa RB garante a existência das transições entre os estados u_4 e u_5 após cada unidade de segmento U_i processada, tendo em vista que a função adaptativa RA remove as transições para esses estados em cada volta no ciclo.

Este processo é repetido até que o fluxo de entrada se esgote. A transição que consome o *token* c é incluída apenas para indicar o final do segmento, com o autômato atingindo o estado final se o processo for bem-sucedido. Por outro lado, existindo mais do que 5 *tokens* a , a ação adaptativa $-\left[(x_i, VR1) : VR2- \rightarrow VR3 : VR4\right]$ de RA remove a transição correspondente ao *token* c , rejeitando o segmento.

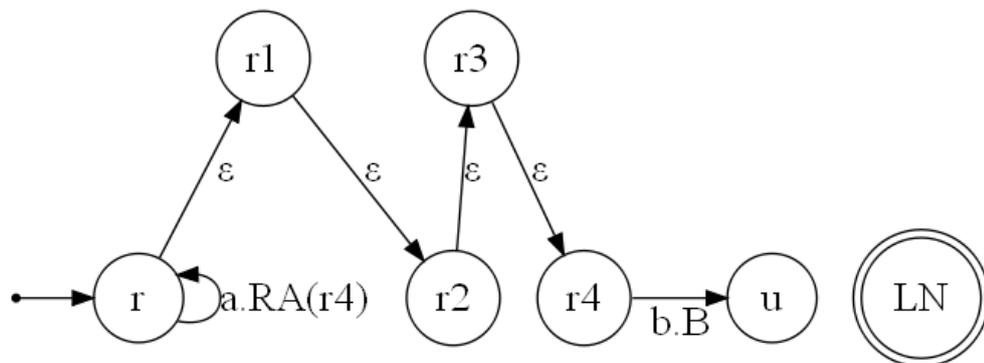


Figura 4.3: Configuração inicial de segmento digitalizado adaptativo, considerando variações em ângulo

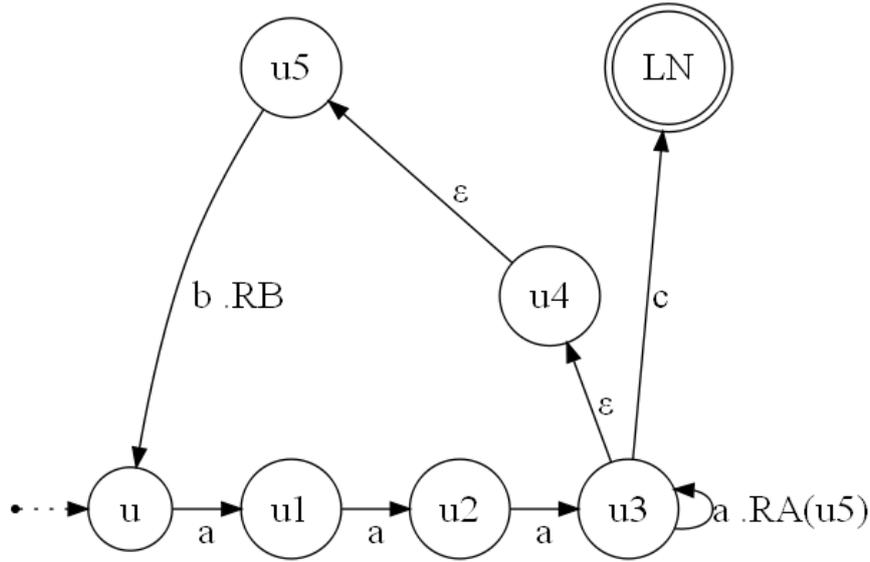


Figura 4.4: Configuração do autômato finito adaptativo da Figura 4.3 após ativação da função adaptativa B

RB{var1, var2:	B{vr, u ₁ [*] , u ₂ [*] , u ₃ [*] , u ₄ [*] , u ₅ [*] :
- [(u ₄ , ε) → u ₅]	- [(r, a) → r : RA(vr)]
- [(u ₃ , ε) → u ₄]	- [(r, ε) → r ₁]
- [(u ₃ , a) → u ₃ : RA(var1)]	- [(r ₁ , ε) → r ₂]
+ [(u ₃ , a) → u ₃ : RA(u ₅)]	- [(r ₂ , ε) → r ₃]
+ [(u ₄ , ε) → u ₅]	- [(r ₃ , ε) → r ₄]
+ [(u ₃ , ε) → u ₄]	+ [(u, a) → u ₁]
- [(u _{var2} , b) → u : RB]	+ [(u ₁ , a) → u ₂]
+ [(u ₅ , b) → u : RB] }	+ [(u ₂ , a) → u ₃]
	+ [(u ₃ , a) → u ₃ : RA(u ₅)]
	+ [(u ₃ , ε) → u ₄]
	+ [(u ₄ , ε) → u ₅]
	+ [(u ₅ , b) → u : RB]
	+ [(u ₃ , c) → LN] }

Quadro 4: Funções adaptativas do autômato finito adaptativo da Figura 4.3 e Figura 4.4. A função adaptativa RA está no Quadro 3

As cadeias da Figura 4.5, especificadas no Quadro 5, indicam a expressividade do modelo na representação tanto do ângulo θ_S quanto do comprimento do segmento digitalizado. Essas sequências de unidades de segmento seguem o modelo de U_i pertencendo ao conjunto $\{a^n b : n = 3, 4, 5\}$, eventualmente truncando a U_1 . cadeias não pertencentes ao conjunto não fazem parte do modelo.



Figura 4.5: Exemplos de segmentos digitalizados modelados pelo autômato finito adaptativo das Figuras 4.3 e 4.4

Cadeias da Figura 4.5	Codificação
1	a^2b
2	a^2ba^3b
3	ba^3b
4	$a^3ba^3ba^4ba^3b$
5	$a^3ba^3ba^4ba^3ba^3ba^3ba^4ba^3b$
6	$a^3ba^3ba^4ba^3ba^5ba^4ba^3ba^4b$
7	$a^3ba^5ba^3ba^4ba^3ba^3ba^4ba^3ba^5ba^4b$

Quadro 5: Cadeias da Figura 4.5 (da esquerda para a direita) modeladas pelo autômato finito adaptativo da Figura 4.3 e Figura 4.4.

4.3 Erros em comprimento

Resumindo o comentado nos itens 2.3 e 3.2 sobre dificuldades existentes na modelagem de segmentos digitalizados sob influência de ruído, sobressaem-se:

- Representar e aplicar sintaticamente tolerâncias aos segmentos digitalizados em qualquer escala;
- Comparar segmentos de reta dentro da tolerância, para qualquer escala;
- Estimadores de comprimento requerem normalmente parâmetros ψ que compensem pelo comprimento superestimado obtido da codificação;

- As soluções requerem métodos para identificar valores aproximados, definidos por inequações.

Uma maneira de representar e aplicar tolerâncias é por um grafo (ou *loop*) adaptativo, conforme a Figura 4.6, tal que o número de estados do grafo (ou seja sua dimensão) é alterado adaptativamente em função, por exemplo, do ângulo θ_S relativo ao eixo x da direção principal do segmento digitalizado S .

A Figura 4.6 mostra um *loop* adaptativo contendo t_0 estados, de L_1 a L_{t_0} . Tendo em vista que esse grafo é apenas de consulta pelo autômato, torna-se irrelevante o símbolo das transições entre os seus estados. A idéia é que a cada símbolo do segmento digitalizado S , tal que $|S| = n$, o autômato acesse o grafo, varrendo uma posição adiante no ciclo, circulando pelo grafo tantas vezes quanto seja n .

O parâmetro $0 < \psi < 1$ é uma taxa de erro significando uma pequena percentagem de n (supondo o caso mais simples em que o comprimento é estimado pela quantidade de símbolos na vizinhança-4), corrigindo o comprimento para $l_E \approx \psi \times n$.

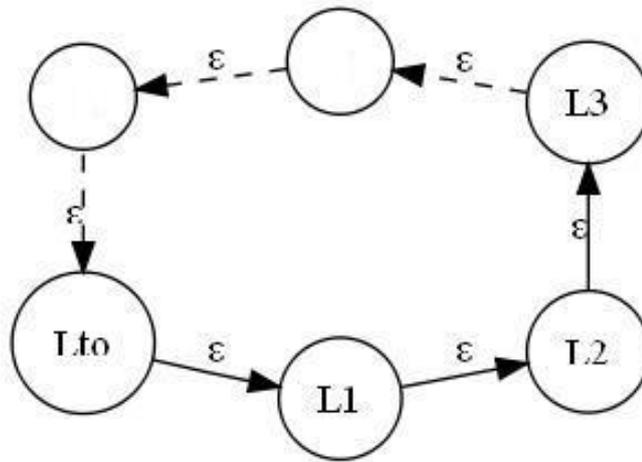


Figura 4.6: Um *loop* adaptativo genérico, interpretado como caso particular de sequência conexa da Definição 32 cujos estados inicial e final são idênticos.

Considerando que n é variável; basta alterar, por meio de uma função adaptativa, a quantidade de estados t_0 do *loop* da Figura 4.6, de acordo com a Expressão 4.2 e de θ_S (o ângulo principal do segmento digitalizado).

$$t_0 \approx \lceil 1/(1 - \psi) \rceil. \quad (4.2)$$

Para um dado valor de n , a cada volta no grafo o autômato finito adaptativo “bombeia” uma primitiva (por exemplo, uma transição qualquer com um símbolo não pertencente a Σ), desse modo obtendo uma medida sintática do parâmetro $(1 - \psi)$ aplicado ao

segmento digitalizado específico. Resulta que n/to símbolos terão sido bombeados com o último estímulo do segmento digitalizado.

Portanto $\lfloor n/to \rfloor$ símbolos podem ser excluídos do total n , para corrigir o comprimento estimado para l_E . Também é possível implementar no segmento digitalizado adaptativo inequações que permitem avaliar comprimentos de dois segmentos, $|S1| = n$ e $|S2| = m$, numa faixa de valores, conforme a Expressão 4.3.

$$n - \lfloor n/to \rfloor \leq |S2| = m \leq n + \lfloor n/to \rfloor. \quad (4.3)$$

4.4 Integração de diversos quadrantes

Dorst e Smeulder (1987) comentam ser impossível reconstruir o contorno contínuo a partir dos dados discretos de um arco digitalizado. Isso significa que muitos contornos corresponderão a uma digitalização específica, todos com comprimentos diferentes. Portanto, os autômatos finitos adaptativos desta pesquisa apresentam a capacidade de modelar unidades de segmento e segmentos digitalizados sujeitos a variações em ângulo. Para a detecção de informações existentes nas cadeias, os autômatos finitos adaptativos deste trabalho são capazes de incorporar à análise sintática os comprimentos das unidades de segmento e respectivos pesos ψ , que compensem pelos erros no processo de digitalização dos segmentos digitalizados, similarmente às Expressões 3.6 e 3.7.

Os conceitos comentados no capítulo 2 associado à análise estrutural deste capítulo confirmam que heurísticas, critérios e informações *à priori* foram requeridas em diferentes soluções. Por exemplo, Kiryati e Kübler (1995) reiteram que o projeto de estimadores ótimos de comprimento, atendendo critérios de desempenho para qualquer ângulo de orientação principal do segmento digitalizado, requer a previsão do número de vetores correspondentes às codificações estimadas.

Dois parâmetros de segmentos digitalizados considerados importantes estruturalmente são o comprimento e o ângulo principal do segmento digitalizado θ_S . Localmente, a unidade de segmento apresenta um determinado comprimento e ângulo de orientação θ_U .

Existem aplicações de segmentos digitalizados, aproximando curvas por conjuntos de segmentos. Conforme Kiryati e Kübler (1995), um arco qualquer pode ser aproximado por linhas retas dentro de uma vizinhança (os autores apresentam inclusive critérios

para seleção da densidade da grade de digitalização). Portanto, estimadores de comprimento de segmentos digitalizados tendem a apresentar bom desempenho com curvas gerais.

As estruturas adaptativas da tese utilizam o primeiro quadrante para as demonstrações. Caso seja necessário incorporar segmentos de outros quadrantes a um determinado segmento digitalizado adaptativo, representativo de apenas um quadrante, basta alterar os símbolos do segmento digitalizado adaptativo convenientemente do primeiro para o outro quadrante, obtendo o segmento digitalizado adaptativo com mesma topologia do inicial.

Simplificadamente, para incluir o terceiro quadrante no exemplo apresentado, basta alterar a para c e b para d obtendo o segmento digitalizado adaptativo com mesma topologia (do terceiro quadrante), e acrescentar ao mecanismo no estado inicial do dispositivo resultante, sem incluir qualquer restrição, pois os símbolos são diferentes.

Exemplificando para a vizinhança-4 e primeiro quadrante, a fim de incluir segmentos digitalizados adaptativos do mesmo quadrante é necessário avaliar a variação dos símbolos de *Prop1* e *Prop2* em função do ângulo principal do segmento digitalizado. O símbolo que ocorre isolado e o que ocorre em corridas são os mesmos para segmentos digitalizados com ângulo de orientação principal na faixa entre 0 e $\pi/4$ radianos. Os símbolos se invertem para segmentos digitalizados com ângulo de orientação principal entre $\pi/4$ e π radianos, devido à vizinhança-4. Em outras vizinhanças e combinações, é necessário observar a variação dos símbolos em função dos ângulos de modo similar.

4.5 Expressividade do modelo proposto

Dois parâmetros de segmentos digitalizados considerados importantes estruturalmente são o comprimento e o ângulo principal dos mesmos denominado θ_S . Localmente, cada unidade de segmento apresenta um determinado comprimento e ângulo de orientação θ_U .

Comparando-se o modelo proposto com as linguagens humanas, as quais se desenvolvem como um organismo vivo partindo de sementes, que se estruturam em raízes, posteriormente tais raízes criam famílias de palavras; da mesma maneira o autômato finito adaptativo que implementa o segmento digitalizado adaptativo representa, na sua configuração inicial, uma semente correspondente aos ângulos e tolerâncias possíveis de se desenvolverem, de acordo com os estímulos, em segmentos diferentes de acordo

com as combinações de comprimentos, de θ_S e de θ_U . Convém enfatizar que esses comprimentos podem variar teoricamente até infinito.

4.6 Determinismo

Transições em vazio introduzem não-determinismo. Ao invés de transições em vazio, podem-se fazer outras implementações utilizando-se marcadores não pertencentes a Σ , tal como Δ , pois as transições em vazio poderiam ser percorridas enquanto Δ não.

A Figura 4.7 mostra uma variante determinística do autômato finito adaptativo da Figura 4.3, em que a cada vez que a função adaptativa RA é ativada remove-se uma transição com o marcador Δ , alterando-se o parâmetro de RA. Por exemplo, de acordo com o Quadro 6, na primeira ativação de RA o parâmetro é alterado para r3. Com o primeiro símbolo b , a função adaptativa B é ativada resultando que a função adaptativa RA é removida.

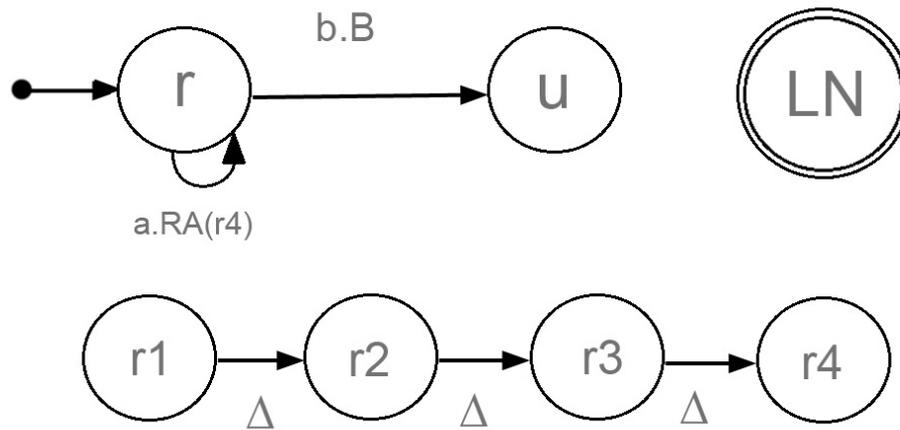


Figura 4.7: Variante determinística do autômato finito adaptativo da Figura 4.3.

$$\begin{array}{l}
 RA(\rho)\{vr1 : \\
 ?[(vr1, \Delta) - > \rho] \\
 -[(vr1, \Delta) - > \rho] \\
 -[(r, a) - > r : RA(\rho)] \\
 +[(r, a) - > r : RA(vr1)]\}
 \end{array}$$

Quadro 6: Função adaptativa paramétrica RA, do autômato finito adaptativo da Figura 4.7

5 SÍNTESE DA TESE

Fundamentado nos tópicos anteriores, neste capítulo apresentam-se uma síntese dos conceitos principais envolvidos nesta tese, bem como as possíveis limitações da mesma.

Tendo em vista que os segmentos digitalizados apresentam atributos variáveis, são necessárias condições de contorno que levem em conta esses atributos, tais como comprimento e ângulo dos segmentos. Tais condições são obtidas ao utilizar uma vizinhança adaptativa de arcos.

5.1 Introdução

A tese pode ser sintetizada nos seguintes pontos:

- Aplicar a adaptatividade para representar os erros, as tolerâncias e os parâmetros envolvidos, conforme descrito no capítulo 4;
- Utilizar a vizinhança adaptativa a fim de incorporar dinamicamente ao modelo as tolerâncias e erros nos parâmetros, tais como ângulo e comprimento;
- Utilizar a facilidade de escalas adaptáveis.

Esta seção apresenta uma introdução revendo principalmente os trabalhos de Nacken (1993) e de You e Fu (1979).

5.1.1 Escalas adaptáveis

Saghri e Freeman (1981) concluíram que qualquer que seja o número de direções do *chain code*, o erro de quantização é limitado a menos do que metade do espaçamento

da grade de codificação. De acordo com a precisão requerida, a grade digital deve ser suficientemente fina para preservar o raio de curvatura mínimo dos arcos.

Largeteau-Skapin e Andres (2006) descreveram operações de transformação entre os formalismos euclidianos e discretos, sendo que a primeira operação proposta por esse trabalho foi um escalonamento discreto (*discrete smooth scaling operation*) com o propósito de descrever objetos discretos em uma grade mais fina.

A segunda operação estudada e implementada foi denominada simplificação geométrica discreta (*discrete based geometrical simplification operation*), aplicada em um objeto euclidiano que, em primeiro lugar, foi digitalizado em um determinado espaçamento de grade e depois reconstruído para o formalismo euclidiano. De acordo com o espaçamento de grade selecionado, detalhes do objeto são perdidos, não aparecendo no objeto reconstruído. Ou seja, por alterações do espaçamento de grade foram obtidos diferentes níveis de detalhes do objeto reconstruído. Na prática, o espaçamento da grade não foi alterado, mas a escala do objeto relativamente à grade, significando que a escala do objeto foi alterada no âmbito do formalismo euclidiano em função do ajuste requerido para o espaçamento da grade e simplificações previstas.

A escala adaptável visa utilizar a simplificação geométrica discreta, alterando os erros de quantização adaptativamente em função dos comprimentos e curvaturas dos arcos. Com esse procedimento, detalhes detectados para um determinado espaçamento da grade, não serão detectados em outros, e vice-versa.

5.1.2 Atributos de arcos digitais

Foram identificados quatro atributos para representar arcos digitais, entendidos como segmentos de arcos concatenados, tal que o ângulo entre dois segmentos de arco consecutivos é usado para descrever as conexões (YOU; FU, 1979).

Conforme You e Fu (1979), uma curva qualquer C pode ser caracterizada pela sua função de curvatura $f(l)$. Dado um ponto l pertencente a C , e considerando σ o ângulo entre as linhas tangentes ao segmento de arco nos pontos $l - \frac{1}{2}\Delta l$ e $l + \frac{1}{2}\Delta l$, a função de curvatura $f(l)$ é dada pela Expressão 5.1, correspondendo à derivada da direção de C ao longo do segmento, relativamente ao comprimento.

$$f(l) = \lim_{\Delta l \rightarrow 0} \frac{\sigma}{\Delta l}. \quad (5.1)$$

A Definição 50 indica quatro atributos, abrangidos por um denominado descritor-C, a

fim de representar um segmento de curva qualquer.

Definição 50 *Descritor-C*: o descritor-C tem o propósito de representar um segmento de curva p qualquer entre dois pontos (pixels) X_1 e X_2 , tal que $p = \widehat{X_1 X_2}$, por meio de quatro atributos, \vec{C} , L , A e S , ou seja, $D(p) = (\vec{C}, L, A, S)$ (YOU; FU, 1979).

Na Definição 50:

- \vec{C} é o vetor que aponta de X_1 para X_2 : $\vec{C} = X_1 \vec{X}_2$;
- L é o comprimento total da curva: $L = \int_0^L dl$;
- A é a mudança angular total: $A = \int_0^L f(l) dl$;
- S é uma medida de simetria: $S = \int_0^L \left(\int_0^S f(l) dl - \frac{A}{2} \right) ds$.

Pela Definição 50, uma reta ideal é um caso particular de arco com função de curvatura constante cujo descritor é composto dos atributos $L = |\vec{C}|$, $A = 0$ e $S = 0$, confirmando que aqueles arcos digitais cuja $f(l)$ ao longo do arco, estiver abaixo de um limiar, que depende do comprimento do arco, podem ser aproximados a segmentos digitalizados.

Devido a limitações por utilizarem apenas linguagens regulares, You e Fu (1979) procuraram simplificar os algoritmos de classificação, aumentando a complexidade da etapa de segmentação ao utilizar primitivas na forma de curvas. Também não se restringiram a curvas primitivas compostas apenas de segmentos de curva simples, embora na maioria dos casos as primitivas utilizadas sejam essencialmente segmentos de curvas simples. Além de curvas primitivas, obtidas pela concatenação de um ou mais segmentos de curva elementares, foi necessário uma relação para descrever a conexão entre dois segmentos de curva. A conexão entre segmentos foi por um único atributo relativo ao ângulo entre os segmentos, denominado ângulo primitivo. Assim, para You e Fu (1979) uma curva primitiva é a concatenação de um ou mais segmentos de curvas elementares.

Apesar do descritor-C ter sido definido para o caso contínuo, podendo ainda ser entendido como afeto aos arcos contínuos cuja digitalização resulta no arco digital, You e Fu (1979) comentaram que os atributos foram computados, eficientemente no caso discreto.

Nesse particular, como uma ilustração para o caso discreto, Gao e Leung (2002) propõem um método com base em cadeias de atributos, para reconhecimento de faces

humanas, aplicando as técnicas estrutural e sintática. Por esse método, um objeto pode ser representado como um conjunto ordenado de segmentos de linhas, codificados por cadeias. Partindo da definição de segmentos de linhas como entidades bi-dimensionais com atributos de comprimento, de orientação e de informações estruturais quanto à localização relativa entre os segmentos, a abordagem de Gao e Leung (2002) leva em consideração uma sequência de segmentos de retas definidores de perfis de rostos humanos a fim de resolver inconsistências na detecção. Isso é efetuado pela representação de linhas primitivas por atributos da forma $P(l, \theta, x, y)$, onde l , θ , x e y são o comprimento, o ângulo de orientação, e a localização do ponto médio da linha, respectivamente. O ângulo de orientação θ é definido como o ângulo mínimo formado entre o segmento de linha e uma linha de referência. As linhas entre a ponta do nariz e queixo são utilizadas como referência nesse estudo. A técnica baseia-se principalmente na mesclagem de cadeias para resolver o problema da inconsistência na detecção de pontos.

Pelo algoritmo proposto por Gao e Leung (2002), dado um contorno de k primitivas $P_{i-k+1}P_{i-k+2}\dots P_i$, ocorrem fusões entre primitivas (ou seja duas ou mais primitivas são associadas ou agrupadas a uma determinada primitiva representativa de uma classe) considerando l_k e θ_k , o comprimento e ângulo de orientação da primitiva de referência. O importante é enfatizar em Gao e Leung (2002) que a contribuição em ângulo, de uma primitiva específica, para as fusões, depende do comprimento do contorno relacionado a essa primitiva, significando que e sua contribuição é proporcional à quantidade de primitivas desse mesmo tipo.

5.1.3 Métrica para segmentos de retas

Este tópico apresenta um resumo do trabalho de Nacken (1993), o qual leva em conta também o comprimento de primitivas como fator no agrupamento das mesmas em classes, no caso digital. Esse trabalho mostra que, pelo efeito da digitalização, a determinação dos diversos atributos em arcos envolvem estimativas e compromissos entre exatidão e tolerâncias permitidas.

O trabalho de Nacken (1993) propõe considerar as imagens como contendo um determinado número de segmentos digitalizados curtos, gerados, por exemplo, pela saída de um algoritmo de detecção de bordas. Os segmentos digitalizados não são considerados simplesmente um grupo de *pixels* colineares da imagem; mas sim, como uma estrutura individualizada, definida por suas propriedades geométricas, por exemplo, os correspondentes ângulo de orientação, o centro ou comprimento do segmento.

Essa idéia tem o efeito de tratar tais segmentos digitalizados curtos como primitivas, às quais Nacken (1993) denomina de *tokens*. Tendo em vista que em muitos casos, segmentos digitalizados curtos se situam ao longo de linhas retas, tais linhas passam a ser as estruturas mais relevantes em uma imagem. Em outras palavras, a informação de diferentes segmentos digitalizados curtos passam a ser combinadas de modo a encontrar as grandes estruturas lineares, ou seja segmentos digitalizados relativamente mais longos e pronunciados. O processo de identificar os mencionados segmentos digitalizados pronunciados costuma se basear na detecção de um ou mais número de limiares indicativos, tais como: a diferença de ângulo de orientação entre dois *tokens* não deve exceder um determinado ângulo, ou que a distância entre as extremidades de dois segmentos não devem ultrapassar um determinado comprimento.

Os critérios dependentes do comprimento dos segmentos são expressos por métricas compostas por números individuais tal como por funções envolvendo a relação geométrica entre os *tokens*. Um exemplo é a parametrização dos segmentos digitalizados pelas coordenadas indicadas na Expressão 5.2.

$$(x, y, \theta, l) \in \mathbb{R}^2 \times [0, \pi) \times \mathbb{R}_{>0}, \quad (5.2)$$

onde o ponto (x, y) é o centro do segmento, a coordenada θ indica que a direção do segmento de reta é $(\cos \theta, \sin \theta)$, e l é o comprimento do segmento digitalizado.

A partir dos trabalhos de Nacken (1993) e Park (2009), a aplicação de uma métrica requer uma função de vizinhança representada na Expressão 5.3, indicativa da probabilidade de que um dado segmento digitalizado μ , cujas coordenadas são dadas pela Expressão 5.2, faça parte de uma estrutura linear existente μ_0 :

$$f_{\mu_0}(x, y, \theta, l). \quad (5.3)$$

O trabalho de Park (2009) apresenta um exemplo de função de vizinhança de segmentos digitalizados, a qual apresenta uma largura relativamente grande, dependente de parâmetros da estrutura linear global. A Figura 5.1 mostra dois segmentos μ_0 e μ representando arcos com seus respectivos parâmetros.

Na mesma Figura 5.1, por esta tese o autômato finito adaptativo que implementa o segmento digitalizado adaptativo da Definição 29 consome, sequencialmente, as unidades de segmento da cadeia correspondente a μ_0 , cujo ângulo principal é θ_0 . No instante em que ocorrer uma alteração de ângulo principal para θ , tal que $\theta - \theta_0$ seja maior ou igual a um determinado limiar mínimo, dependente do comprimento, o autômato finito adaptativo passa a consumir os símbolos da cadeia μ . Entretanto, o arco global

formado por μ_0 e μ somente será reconhecido se o comprimento μ_0 atender a determinada função definida por uma vizinhança adaptativa da Definição 30 (proporcional ao comprimento do segmento digitalizado).

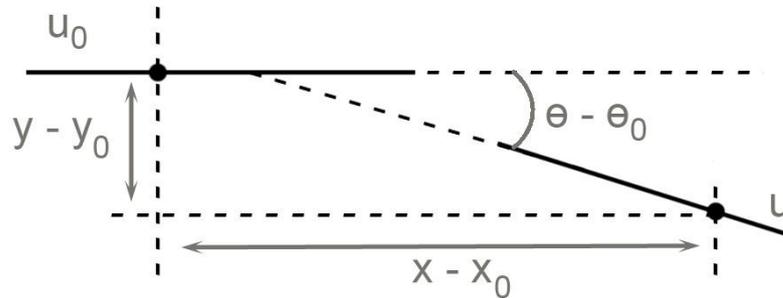


Figura 5.1: Dois segmentos de reta representativos de um arco digital (adaptado de Park (2009)).

5.2 A tese da vizinhança adaptativa

A tese desta pesquisa, em que os arcos e segmentos podem estar nas mais diversas escalas a serem delimitadas por condições de contorno adaptativas da Definição 31: essas condições definem a envoltória de uma região digital relativa à grade de amostragem variável conforme a escala adaptável e dependente dos estímulos.

Na capítulo 4, o *loop* adaptativo da Figura 4.6 foi a estrutura usada para obter os comprimentos corrigidos l_E . Neste tópico, aplica-se a mesma estrutura de maneira diferente, tendo o número de estado *to* do *loop* o significado de comprimento de referência para os segmentos digitalizados adaptativos alterarem a vizinhança do segmento digitalizado sob análise, adaptativamente.

5.2.1 A vizinhança adaptativa

Este tópico fundamenta a vizinhança adaptativa, descreve segmentos digitalizados adaptativos que representem vizinhanças adaptativas, bem como analisa os resultados obtidos por meio de testes.

A representação das diferentes instâncias do modelo ideal afetado pelos erros em ângulo requer que o segmento digitalizado adaptativo atue numa faixa de ângulos que seja alterada em função dos estímulos. Dentre esses estímulos, destaque-se para a vizinhança adaptativa o comprimento do segmento digitalizado de entrada. Neste tópico,

conceitua-se a modelagem de segmentos digitalizados adaptativos, exemplificados no primeiro quadrante. Ou seja, o ângulo local permitido para cada θ_U relativo ao eixo x correspondente às unidades de segmento U_i de um segmento digitalizado S de entrada, varia adaptativamente.

Um exemplo de aplicação da noção de curvatura de arcos digitais é pela propriedade da corda, pois, no caso ideal, arcos digitais são segmentos digitalizados se atenderem à mencionada propriedade, definida para um pixel de vizinhança.

Esta pesquisa introduz uma função de vizinhança adaptativa obtida pela propriedade da corda modificada para modelos de ordens superiores (ordem n), a fim de incorporar as tolerâncias em ângulo e em comprimento, levando em conta os símbolos definidos por *Prop1* e *Prop2* dos arcos correspondentes. A propriedade da corda modificada é apresentada na Definição 28, reescrita abaixo.

Propriedade da Corda Modificada: Diz-se que um arco digital C apresenta a propriedade da corda modificada se, para cada dois pontos digitais c e d pertencentes a C , e para cada ponto $p = (x, y)$ em \overline{cd} , existe um ponto $e = (h, k)$ pertencente a C tal que $\max\{|x - h|, |y - k|\} < n$ com $n \geq 1$. \overline{cd} é o segmento de reta entre c e d , e n considera o comprimento do segmento digitalizado adaptativamente.

Na Definição 28, destacam-se:

- Vizinhança: $\max\{|x - h|, |y - k|\} < n$;
- n é a ordem do modelo, dependente da situação momentânea, dos estímulos e do comprimento do segmento;
- A função da vizinhança de um segmento digitalizado deve ter uma largura relativamente grande (proporcional ao comprimento medido) na direção da estrutura linear global (NACKEN, 1993).

A Figura 5.2 esquematiza os modelos propostos pela Definição 28 em que P depende do comprimento do segmento, e é variável do tipo: $P(a, b^m) : 0 \leq m < +\infty$.

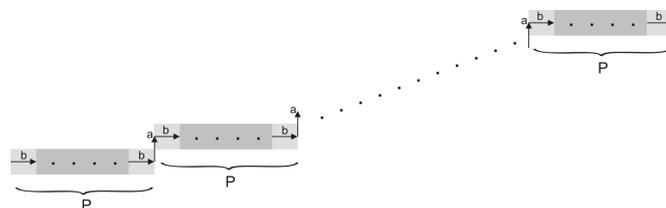


Figura 5.2: Modelos de ordens superiores (ordem n), com P variável.

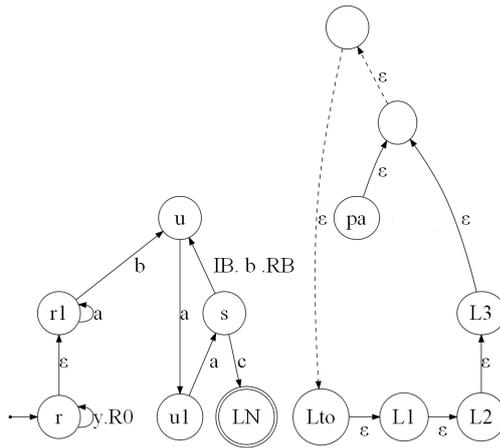
Relacionando o conceito de função de curvatura para o caso digital, os tópicos que se seguem mostram aspectos de discriminação por meio de segmentos digitalizados adaptativos, quanto à curvatura do segmento digitalizado, com propósito de alterar adaptativamente a vizinhança correspondente relacionada a P variável.

5.2.2 Implementação e testes de segmentos digitalizados adaptativos quanto à vizinhança adaptativa

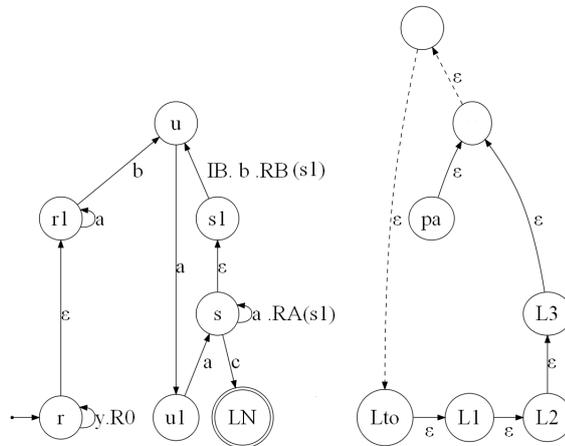
A vizinhança adaptativa associa um conjunto de arcos a um segmento digitalizado adaptativo correspondente, que reconhece esse conjunto. Neste tópico, apresentam-se a implementação e testes de modelos de segmentos digitalizados adaptativos atendendo à tese de vizinhanças adaptativas, em que essa região é delimitada por arcos delimitadores, responsáveis pelos contornos das regiões. Há dois tipos desses arcos delimitadores: arcos côncavos e arcos convexos. Nos itens a seguir mostram os dois tipos de segmentos digitalizados adaptativos, delimitados por arcos côncavos e convexos, sendo possível um terceiro tipo combinando esses dois primeiros.

5.2.2.1 Vizinhança delimitada por arco côncavo

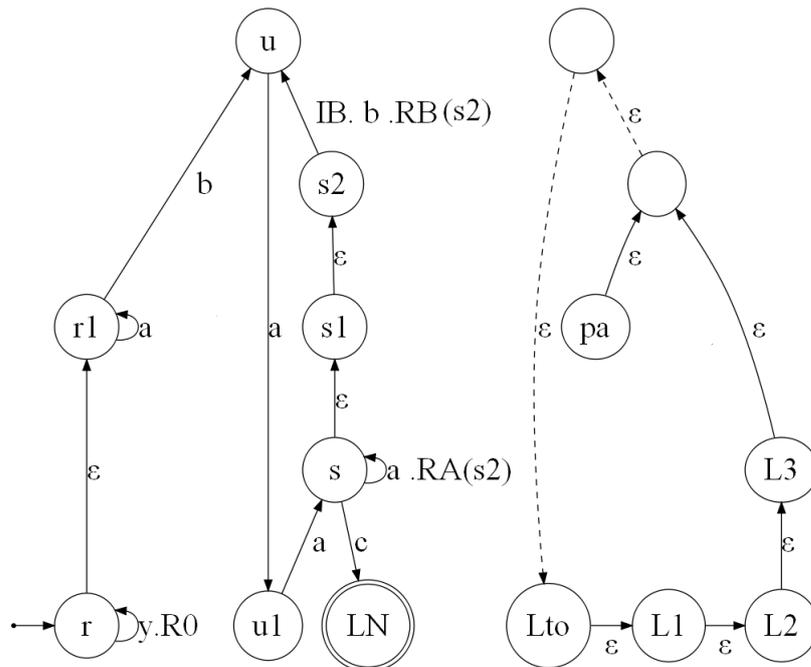
O segmento digitalizado adaptativo da Figura 5.3(a) ajusta a correspondente vizinhança em função de to , indicativo da quantidade de estados do loop adaptativo da Figura 4.6 conforme já estudado no tópico 4.3. Dessa maneira, inicialmente μ_0 é reconhecido, e posteriormente μ , atendendo ao descrito acima na Figura 5.1.



(a) Configuración inicial.



(b) Configuración após o reconhecimento do primeiro ciclo de comprimento L_{to} .



(c) Configuración após o reconhecimento do segundo ciclo de comprimento L_{to} .

Figura 5.3: Apresentação de três configurações de segmento digitalizado adaptativo cuja vizinhança varia adaptativamente em função do comprimento da cadeia de entrada, acessando o *loop* à direita (para arcos côncavos).

$\begin{aligned} & \text{IB}\{var1, var2, var3 : \\ & - [(s_{var1}, \epsilon) \rightarrow s_{var2}] \\ & - [(s, a) \rightarrow s : RA(var3)] \\ & \} \\ & \text{AD}(s_{max})\{ : \\ & ? [(p_a, \epsilon) \rightarrow L_{to+1}] \\ & \text{ADD}(s_{max}) \} \\ & \text{ADD}(s_{max})\{s_{new}^* : \\ & - [(s_{max}, b) : IB \rightarrow u : RB] \\ & + [(s_{max}, \epsilon) \rightarrow s_{new}] \\ & + [(s_{new}, b) : IB \rightarrow u : RB] \\ & - [(s, a) \rightarrow s : RA(s_{max})] \\ & + [(s, a) \rightarrow s : RA(s_{new})] \\ & - [(p_a, \epsilon) \rightarrow L_{to+1}] \\ & + [(p_a, \epsilon) \rightarrow L1] \\ & \} \end{aligned}$	$\begin{aligned} & \text{RB}(s_{max})\{var1, var2, var3 : \\ & + [(s_{var1}, \epsilon) \rightarrow s_{var1+1}] \\ & + [(s, a) \rightarrow s : RA(s_{max})] \\ & - [(p_a, \epsilon) \rightarrow L_{var3}] \\ & + [(p_a, \epsilon) \rightarrow L_{var3+1}] \\ & - [(var2, b) : IB \rightarrow u : RB] \\ & + [(s_{max}, b) : IB \rightarrow u : RB] \\ & \text{AD}(s_{max})\} \\ & \text{RA}(\rho)\{vr1, vr2, vr3, \\ & vr4, vr5, vr6 : \\ & - [(vr6, \epsilon) \rightarrow \rho] \\ & - [(\rho, vr1) : vr2 \rightarrow vr3 : vr4] \\ & + [(vr6, vr1) : vr2 \rightarrow vr3 : vr4] \\ & - [(vr5, a) \rightarrow vr5 : RA(\rho)] \\ & + [(vr5, a) \rightarrow vr5 : RA(vr6)] \} \end{aligned}$
--	---

Quadro 7: Representação algorítmica das funções adaptativas do autômato finito adaptativo da Figura 5.3(a). O código apresentado no adendo A pode ser utilizado como referência quanto a detalhes.

O segmento digitalizado adaptativo da Figura 5.3(a) reconhece cadeias do tipo $W = \Psi S$ em que S é o segmento digitalizado a ter sua vizinhança alterada adaptativamente em função de seu próprio comprimento, com relação à quantidade de estados to do *loop*, à sua direita. No exemplo deste tópico, o comprimento da cadeia de entrada é dado pela correspondente quantidade de unidades de segmento.

Na Figura 5.3(a), a vizinhança é delimitada por arco côncavo devido ao ângulo máximo de cada unidade de segmento U que aumenta gradativamente seguindo os ângulos $\arctan(2)$, $\arctan(3)$, $\arctan(4)$ $\arctan(\lfloor \lambda/to \rfloor)$ relacionados à sequência conexa do estado s .

Ψ é a cadeia $\Psi = y^{to}$. Em outras palavras, Ψ é uma cadeia composta por *tokens* y , empregada para informar ao segmento digitalizado adaptativo o comprimento de referência a ser atribuído a to , permitindo que to possa variar de acordo com o ângulo principal θ_S do segmento digitalizado S . Resulta que o segmento digitalizado adapta-

tivo constrói o *loop* adaptativo de quantidade de estados to , de acordo com o número de símbolos y .

A Figura 5.3(a) indica também o ponteiro p_a .

Definição 51 *Ponteiro*: *é uma transição elementar denominado simplesmente pelo seu estado que se mantém fixo, enquanto o outro estado varia. Exemplificando para a Figura 5.3(a), o ponteiro p_a é o estado fixo.*

Caso a tolerância seja informada ao autômato finito adaptativo por *tokens* y , a função adaptativa RO tem a finalidade de construir o *loop* adaptativo, de acordo com a quantidade de *tokens* y , e também de criar o ponteiro. A transição $(r_1, a) \rightarrow r_1$ é uma simplificação para modelar os possíveis símbolos a da unidade de segmento inicial.

Na Figura 5.3(a), com o primeiro símbolo b , o autômato finito adaptativo passa a consumir as unidades de segmento subsequentes. Cada unidade de segmento de S ativa as funções adaptativas IB e RB, tal que:

1. As transições da sequência conexa $s_1, s_2, s_3, \dots, s_i$ que existirem no autômato são removidas pela função adaptativa IB (isso porque a unidade de segmento anterior pode não ter a quantidade máxima de tokens a previstos na vizinhança afeta à configuração do autômato finito adaptativo);
2. A função adaptativa RB insere novamente no autômato as mesmas transições $s_1, s_2, s_3, \dots, s_i$ reconstituindo a sequência conexa, a fim de possibilitar o consumo da unidade de segmento seguinte;
3. A função adaptativa RB movimenta o ponteiro p_a uma posição adiante no *loop*;
4. A cada volta completa no *loop* efetuada por p_a , pela ativação da função adaptativa RB (juntamente com a função adaptativa AD) as seguintes ações são executadas:
 - (a) Alteração da vizinhança do segmento digitalizado adaptativo pela inclusão de uma transição $(s_{i-1}, \epsilon) \rightarrow s_i$ incrementando a sequência conexa $s_1, s_2, s_3, \dots, s_i$ pelo estado s_i ;
 - (b) Alteração da função adaptativa IB para que, quando ativada, sejam removidas, no passo 1 acima, também as transições relacionadas ao estado s_i , incluído no passo anterior;

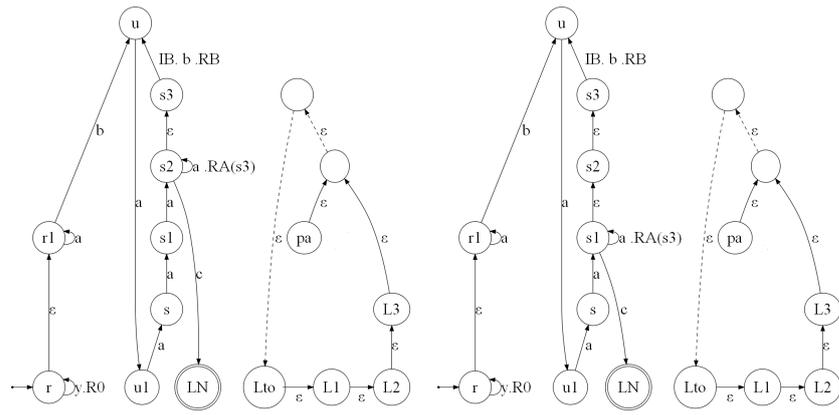
- (c) Remoção da transição $(s, a) : RA(s_{i-1}) \rightarrow s$ e inserção da transição $(s, a) : RA(s_i) \rightarrow s]$.

Supondo ψ definindo um valor de comprimento para $to = 10$, no primeiro ciclo no *loop* percorrido pelo segmento digitalizado adaptativo, a tolerância em ângulo é igual a zero, aceitando unidades de segmento com ângulo igual a $\arctan 2$. No segundo ciclo no *loop*, existe uma tolerância por aceitar ângulos na faixa $\arctan 2 \leq \theta_U \leq \arctan 3$. No terceiro ciclo, a tolerância é alterada, aceitando ângulos na faixa $\arctan 2 \leq \theta_U \leq \arctan 4$, e assim por diante, em cada ciclo.

Supondo ainda ψ definindo um valor de comprimento para $to = 10$, esse conceito de vizinhança adaptativa pode ser observado no arco da Figura 5.4(a), o qual é aceito por ter μ_0 correspondendo a segmento digitalizado com 6 unidades de segmento do tipo *ab*, seguido por μ_1 formado por 5 unidades de segmento do tipo *aaab*. Posteriormente, o arco é finalizado com μ_2 composto de 5 unidade de segmento *aaaab*. Assim, o reconhecimento desse arco da Figura 5.4(a) é efetuado pelo recurso do segmento digitalizado adaptativo ter aumentado a sua vizinhança, gradativamente, de acordo com o comprimento de μ_0 , por meio do número de ciclos percorridos pelo autômato finito adaptativo, no *loop* apresentado à direita na Figura 5.3(a). Os arcos das Figuras 5.4(c) e 5.4(c) são rejeitados por não atenderem ao especificado pela vizinhança adaptativa.

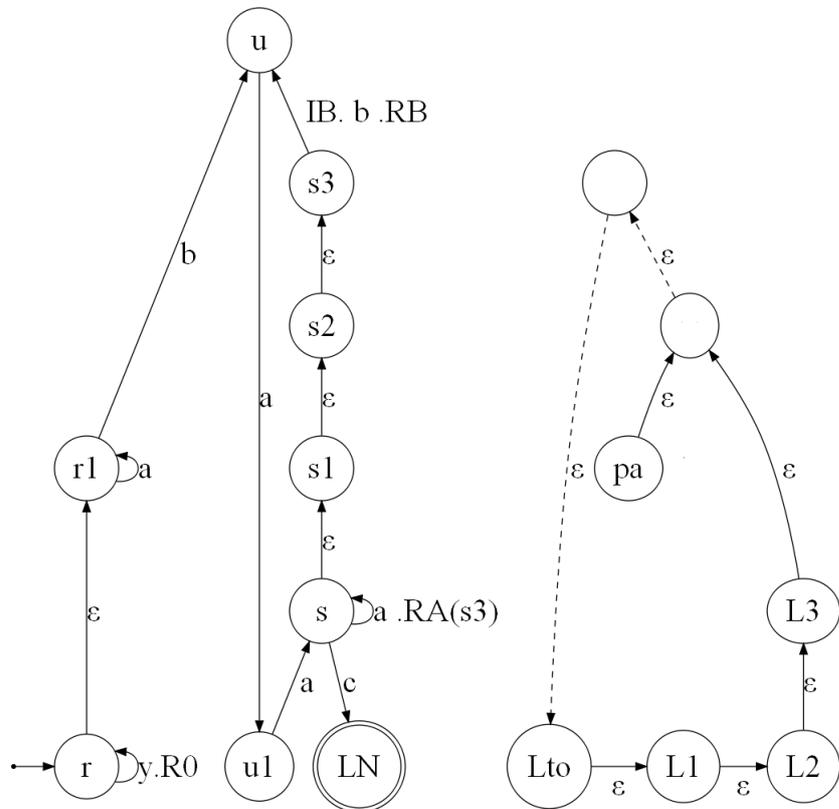
adaptativo são apresentadas nas Figuras 5.6(a) 5.6(b) 5.6(c):

1. As transições da sequência conexa $s_{3-i}, s_{3-i+1}, \dots, s_{3-1}, s_3$ que existirem no autômato são removidas após ativação da função adaptativa IB;
2. A ativação da função adaptativa RB resulta na reintegração no autômato das mesmas transições $s_{3-i}, s_{3-i+1}, \dots, s_{3-1}, s_3$ removidas pela ativação da função adaptativa IB, a fim de possibilitar o consumo da unidade de segmento seguinte;
3. Pela ativação da função adaptativa RB, o ponteiro p_a é movimentado uma posição adiante no *loop*;
4. A cada volta completa no *loop* efetuada por p_a , pela ativação da função adaptativa RB e função adaptativa AD, as seguintes ações são executadas:
 - (a) Alteração da vizinhança representada pelo segmento digitalizado adaptativo pela retirada da transição $(s_{3i-1-1}, a) \rightarrow s_{3-i}$ e inserção no autômato da transição $(s_{3-i-1}, \epsilon) \rightarrow s_{3-i}$ incrementando a sequência conexa $s_{3-i}, s_{3-i+1}, \dots, s_{3-1}, s_3$ do estado s_{3-i-1} ;
 - (b) A função adaptativa IB é alterada para que sejam removidas, no passo 1 acima, as transições do estado s_3 até a do estado s_{3-i-1} ;
 - (c) Remoção da transição $(s_{3-i}, a) : RA(s_3) \rightarrow s_{3-i}$ e inserção da transição $(s_{3-i-1}, a) : RA(s_3) \rightarrow s_{3-i-1}$.



(a) Configuração inicial.

(b) Configuração após o reconhecimento do primeiro ciclo de comprimento L_{to} .



(c) Configuração após o reconhecimento do segundo ciclo de comprimento L_{to} .

Figura 5.6: Apresentação de três configurações do segmento digitalizado adaptativo da Figura 5.5(a) cuja vizinhança varia adaptativamente em função do comprimento da cadeia de entrada, acessando o *loop* à direita (arco convexo).

Nas Figura 5.6(a) até 5.6(c), a vizinhança é delimitada por arco convexo devido ao ângulo mínimo de cada unidade de segmento U que diminui gradativamente, a cada volta no *loop* de comprimento to , seguindo os ângulos $\arctan(4)$, $\arctan(3)$, até $\arctan(1)$ relacionados à sequência conexa do estado s . Enquanto esse ângulo diminui gradativamente, é importante notar que a delimitação de um ângulo mínimo pode ser

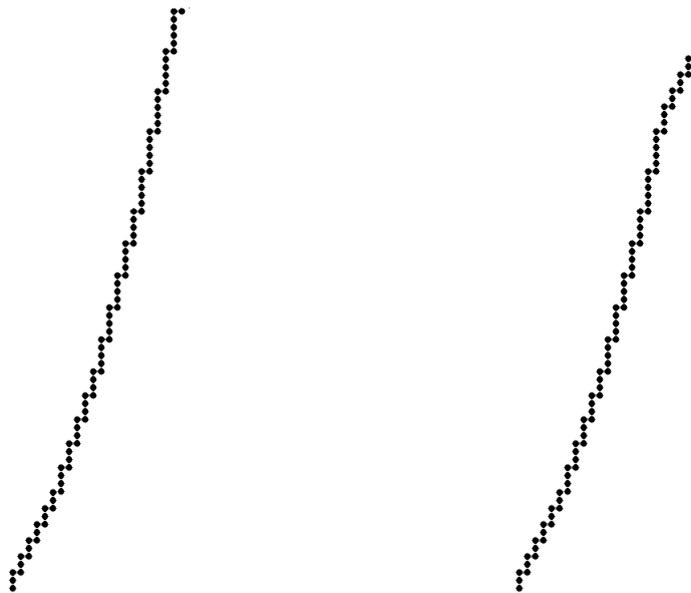
feita por um contador (ver item 2.1.4), composto por uma sequência conexa e uma função adaptativa que conte o número de ciclos percorridos no *loop* de estados *to*.

5.2.3 Comparação com o modelo tradicional

Neste tópico, compara-se a tese desta pesquisa com os modelos tradicionais, sem adaptatividade. Um primeiro aspecto é quanto à questão da flexibilidade.

5.2.3.1 Flexibilidade

O modelo tradicional, sem adaptatividade, tem que atender à propriedade da corda, rejeitando os arcos das Figuras 5.7(a) e 5.7(b) com unidades de segmento U na faixa $U(a, b^m) : 2 \leq m \leq 5$, pois, de acordo com a propriedade da corda, deve existir um padrão de números consecutivos nas corridas, sem apresentar irregularidades.



(a) Exemplo de arco para teste.

(b) Exemplo de arco para teste.

Figura 5.7: Exemplos de segmentos digitalizados reconhecidos pelo segmento digitalizado adaptativo da Figura 5.3(a); porém, não reconhecidos pelo método tradicional, sem adaptatividade.

Dentre os modelos tradicionais, Bhowmick e Bhattacharya (2007) obtiveram algoritmos com relativamente maior flexibilidade ao relaxar a propriedade da corda, numa faixa pré-definida de valores das corridas por meio de um valor denominado d , mantendo a característica de código estático e funcionalidade constante. Um exemplo

é $d = \lfloor (p+1)/2 \rfloor$ tal que os arcos aceitos tenham unidades de segmento U apenas na faixa $U(a, b^m) : p \leq m \leq d$. Quanto aos resultados obtidos por essa abordagem destacam-se a redução das estruturas de dados e a complexidade em tempo dos algoritmos torna-se linear.

Entretanto, ao apresentarem comportamento invariável, os métodos tradicionais segmentam os arcos em grande quantidade de segmentos curtos. Já os métodos com valores pré-definidos numa faixa reduzem o número de segmentos, porém perdem a característica global dos arcos, rejeitando os segmentos digitalizados das Figuras 5.7(a) e 5.7(b), seccionados em diversos segmentos digitalizados.

As alterações dinâmicas de vizinhanças desta tese, que alteram a funcionalidade dos algoritmos em função dos estímulos, permitem uma melhor caracterização global dos arcos, reconhecendo os segmentos digitalizados das Figuras 5.7(a) e 5.7(b).

5.2.3.2 Complexidade computacional

Devido às limitações de comportamento invariável ou pré-definido numa faixa, os métodos tradicionais tendem a seccionar os arcos digitais em um conjunto composto por elevado número de segmentos curtos. Ao reduzir sensivelmente os elementos componentes desse conjunto, a adaptatividade minimiza as estruturas de dados requeridas para representar os arcos. Além disso, a introdução de tolerâncias adaptativas simplifica os algoritmos comparativamente aos que apresentam comportamento invariável.

Portanto, a representação de arcos por um conjunto composto de número ainda mais reduzido de segmentos minimiza a complexidade em tempo dos algoritmos adaptativos, além de torná-la linear.

5.2.4 Análise dos resultados

Este tópico tem a finalidade de analisar os resultados obtidos nas implementações e testes do item anterior 5.2.2 visando determinar restrições no modelo proposto de segmento digitalizado adaptativo, ou seja, se este atende a todas as possibilidades existentes.

5.2.4.1 Ruído e descontinuidades

Constata-se pelos arcos apresentados em 5.2.2 que o segmento digitalizado adaptativo atua convenientemente, desde que o arco a ser reconhecido apresente uma curvatura suave. Este tópico mostra o caso em que um determinado arco apresente picos ou vales locais, que deveriam ser desprezados pelo segmento digitalizado adaptativo.

Caso o arco apresente desvios na sua curvatura, a aceitação do mesmo pelo segmento digitalizado adaptativo não ocorre, mesmo que o arco, como um todo, seja visualmente identificável de forma clara como um arco reto.

Normalmente, as variações em curvaturas costumam corresponder a mudanças no símbolo que ocorre isolado, ocasionando picos e vales nos arcos. Esse efeito pode ser causado também por ruído, ao qual o modelo de Freeman é altamente sensível.

Os mencionados picos e vales se integram aos denominados pontos dominantes. Kankanhalli (1993) cita um trabalho que define pontos dominantes como aqueles existentes em uma curva, exibindo valores elevados de curvatura, e, por isso, transmitem a maioria das informações sobre a forma da mesma. Essa terminologia “valores elevados de curvatura” é viável no sentido relativo aos pontos em uma vizinhança estabelecida. De fato, essa vizinhança é denominada a região de suporte de um ponto, normalmente pré-definida (KANKANHALLI, 1993).

Outro entendimento possível de tais picos e vales os denomina de “cantos”, que significam pontos no *chain code* associados a descontinuidades identificáveis na curvatura média das curvas. A detecção de cantos leva em conta a magnitude da descontinuidade relacionada ao grau de curvatura nas proximidades do canto e às regiões da curva de ambos os lados do canto.

No trabalho de Freeman e Davis (1977), a proeminência de um canto (a “cornerity” de um ponto) é o produto do comprimento de sub-cadeias (tais sub-cadeias são denominadas braços) uniformes do *chain code* para ambos os lados (para “frente” e para “trás”) do ponto, e do ângulo de descontinuidade nesse ponto específico. O efeito do ângulo de descontinuidade e do comprimento das seções uniformes do *chain code* para frente e para trás do ponto é mostrado na Figura 5.8.

Na Figura 5.8-a tanto o ângulo da descontinuidade e dos comprimentos dos braços são pequenos, e é razoável a dúvida quanto à existência efetiva de um canto. Na Figura 5.8-b, a existência de um canto é evidente devido à descontinuidade elevada,

bem como dos comprimentos dos dois braços. O canto da Figura 5.8-c, é também facilmente reconhecido devido à alta variação da curvatura.

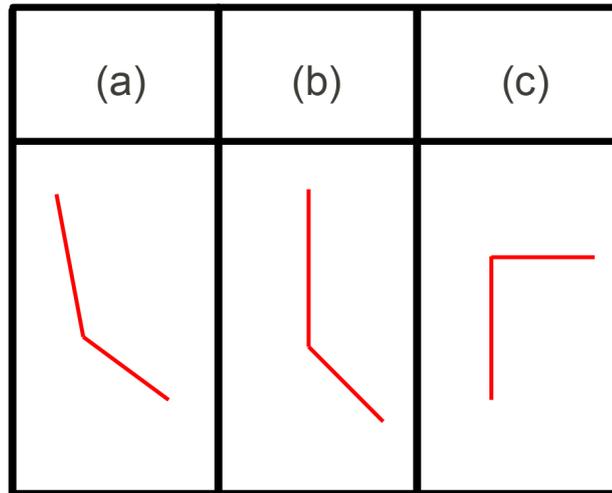


Figura 5.8: Efeito dos atributos ângulo e comprimento locais na determinação de cantos (adaptado de Liu e Srinath (1990)).

Portanto, para esta situação de arco com picos ou vales, o segmento digitalizado adaptativo deveria simplesmente desprezar essas irregularidades locais, e aceitar o arco global. Para o entendimento de tais incertezas, basta considerar novamente a Figura 5.8, quanto a questão de qual seria o novo ângulo do segmento digitalizado adaptativo a ser adicionado ao autômato finito adaptativo, para reconhecimento dos arcos mostrados. Apenas na Figura 5.8-b, o novo segmento digitalizado adaptativo seria facilmente identificável, enquanto que nas demais seria conveniente ter uma visão panorâmica de todo o arco, abrangendo a possibilidade de comparar comprimentos do arco global relativamente aos comprimentos dos braços, tendo em vista que ocorrem braços de comprimento relativamente muito pequenos.

5.2.4.2 Curvaturas e incertezas nas mudanças de direção

As incertezas descritas no item anterior, 5.2.4.1 são críticas também sempre que um determinado arco, mesmo sem qualquer efeito de ruído, mude a sua direção, significando alterar o símbolo que ocorre isolado.

O que se quer dizer é que existe uma tendência a ocorrer ambiguidades em ângulo nas curvaturas digitais, principalmente nas transições de uma direção para outra. O ideal seria definir uma região de suporte relacionada a esses pontos de mudanças, a fim de aceitar qualquer variação em ângulo, até o limite dos ângulos da nova vizinhança

correspondente à direção final, os quais são identificados convenientemente pela quantidade de símbolos das propriedades do modelo de Freeman.

As incertezas ocorrem pela própria natureza do processo de digitalização, associado a aproximações dependentes da finura da grade de amostragem, e a aproximações no modelo teórico, tal como das derivadas das curvaturas com relação ao comprimento. O tópico a seguir complementa a questão das incertezas com o enfoque de uma zona de insensibilidade.

5.2.4.3 Precisão do modelo

Fundamentado na noção de uma zona de insensibilidade denominada β , inerente ao modelo de Freeman (ver Figura 5.9), este tópico complementa o entendimento sobre a influência das distorções na modelagem de arcos e retas.

Supondo o primeiro quadrante, a zona β é facilmente entendida ao lembrar que o ângulo θ_U , de uma unidade de segmento com o eixo x , é expressa por $\theta_U = \arctan(1/P)$ com P significando um número inteiro representando a quantidade de símbolos que ocorrem em grupos. Ora, as variações no ângulo θ_U podem ocorrer alterando P no conjunto dos números inteiros: para P variando entre dois números inteiros consecutivos, ocasiona uma região de incerteza para todas as retas contínuas passíveis de serem digitalizadas entre esses dois pontos.

Além disso, dependendo do ângulo, quanto maior for P , menor será a alteração significativa no ângulo, para um incremento de P , mesmo que de vários números inteiros. Similarmente, quanto menor for P , maior será a alteração significativa no ângulo, para um incremento de P , mesmo que de apenas uma unidade.

A mencionada análise de precisão de coordenadas é feita por meio do retângulo, útil para essa atividade. Entretanto, a zona β ocasiona a incerteza nas coordenadas de arcos quaisquer, sendo que o uso do retângulo apresenta vantagens didáticas para o entendimento do problema.

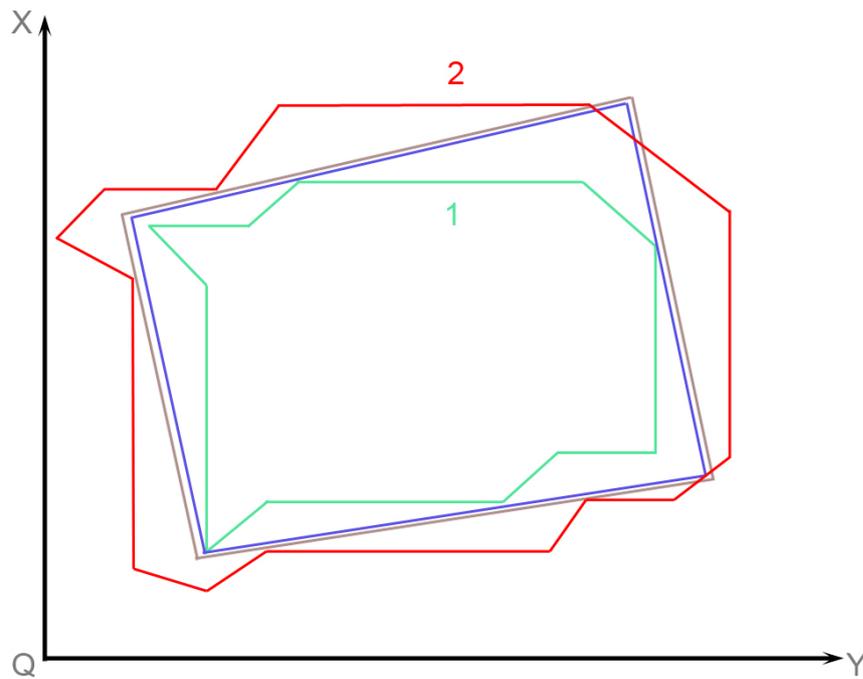


Figura 5.9: Um retângulo digitalizado entre dois contornos delimitadores, numerados 1 e 2, e a zona de insensibilidade (adaptado de Lebedev (2004)).

Resumindo, existe um erro intrínseco ao modelo de Freeman, que causa uma imprecisão na medida das coordenadas nesse modelo de acordo com a espessura da zona β . Lebedev (2004) estima os erros de medição das coordenadas de cantos para o retângulo orientado arbitrariamente numa grade, bem como de outros objetos caracterizados por segmentos de fronteira mutuamente perpendiculares. A influência de ruído não foi apresentada em maiores detalhes, apenas comentando que é função do ângulo de orientação, do comprimento dos lados e da relação entre os lados de um retângulo.

A influência da espessura da zona β nos erros de medição $\Delta = (\Delta_x, \Delta_y)$, das coordenadas dos pontos correspondentes aos vértices de um retângulo, ocorre pela variação na espessura de β , a qual é dependente do ângulo de orientação ϕ do retângulo. A partir do estudo das variações nessa espessura, foram obtidas fórmulas matemáticas para o cálculo dos erros em função do ângulo de orientação para outras formas, sempre com base em estimativas a partir das propriedades geométricas e das propriedades dos segmentos digitalizados.

A Figura 5.10 mostra os erros de medição das coordenadas dos vértices (cantos) de um retângulo, no primeiro quadrante, no sistema de coordenadas cartesianas com inclinação $\tan(\phi) = p/q : p \leq q$, onde p e q são números inteiros, com p/q sendo uma fração irredutível.

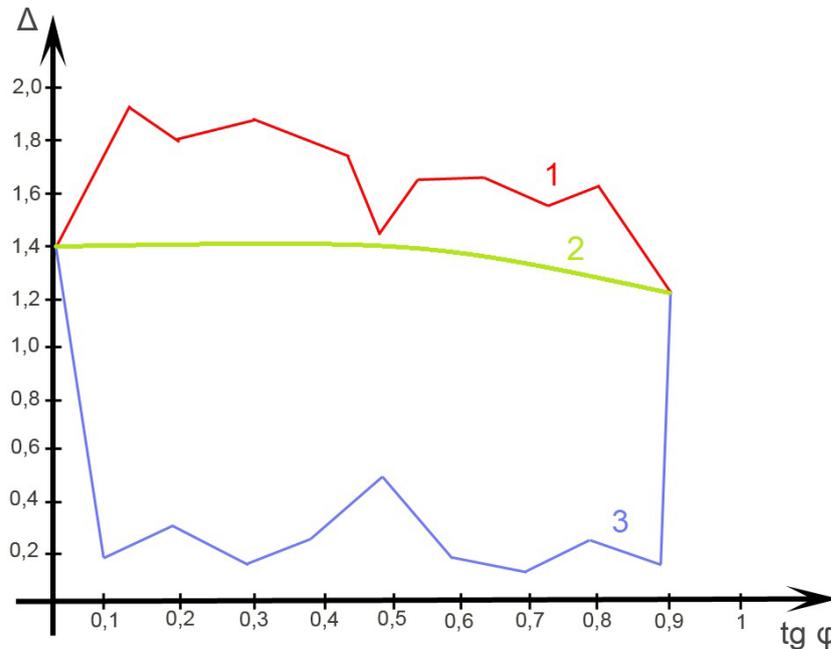


Figura 5.10: Os erros nos vértices (cantos) de um retângulo como uma função do declive, ou ângulo de orientação. (adaptado de Lebedev (2004)).

Dependendo de aproximações em p/q , e da maneira que os cantos do retângulo se ajustem aos nós da grade de amostragem, obtiveram-se os gráficos da Figura 5.10, em que Curva 3 é a situação em que todos os pontos de canto de um retângulo se ajustem exatamente aos nós da grade de amostragem. Assim, a Figura 5.10 apresenta o erro máximo, ou pior caso, pela Curva 1; o erro mínimo pela Curva 3 e uma situação intermediária pela curva do meio Curva 2. Note que a Curva 1 excede o erro mínimo (Curva 3), entretanto a probabilidade de que todos os pontos de canto de um retângulo se ajustem exatamente aos nós da grade de amostragem é quase impossível. Desse modo, Lebedev (2004) concluiu que o pior caso (Curva 1) e o caso intermediário (Curva 2) são mais prováveis do que a Curva 3.

O trabalho de Veelaert (2005) comenta sobre o processo de extração de primitivas geométricas, a partir de uma imagem que tem de lidar com diferentes tipos de incerteza, em diferentes fases do processo, confirmando a necessidade de considerar uma região que leve em conta as imprecisões intrínsecas comentadas neste tópico.

5.3 Sugestões de trabalhos futuros

Neste tópico, apresentam-se sugestões de trabalhos futuros e possíveis soluções para as restrições encontradas no item 5.2.4, utilizando as facilidades desta tese em incorporar

escalas adaptáveis.

5.3.1 Características da escala adaptável

Por estarem limitados a linguagens regulares, You e Fu (1979) tiveram que introduzir um fator de normalização de escalas, que poderia ser o comprimento total de um contorno padrão. Conforme a Definição 50, a Definição 52 mostra a transformação de escalas do trabalho de You e Fu (1979) que possibilita reconhecer contornos pela similaridade entre os descritores correspondente após submetidos à transformação.

Definição 52 *Transformação de escalas:* uma transformação de escalas T é do tipo $T : (\vec{C}, L, A, S) \rightarrow (\frac{\vec{C}}{L}, \frac{L}{L_0}, A, \frac{S}{L})$, onde $C = |\vec{C}|$ e L_0 é um fator de normalização de escalas (YOU; FU, 1979).

A Figura 5.11 mostra um exemplo conveniente para aplicar uma escala adaptável, com certa analogia com o trabalho de You e Fu (1979), mas com o poder computacional do autômato finito adaptativo, portanto sem estar restrito a linguagens regulares. Nessa Figura, o arco $C = \widehat{AB} = v_1v_2\dots v_n$ é tal que a distância de qualquer ponto na curva relativamente ao segmento de reta \overline{AB} é menor que um valor E_h , e a mudança de ângulo, em qualquer ponto ao longo da curva é menor do que um valor E_a , onde E_h e E_a são valores positivos arbitrariamente pequenos. You e Fu (1979) fundamentaram a segmentação de um arco qualquer por meio de regras de decomposição, destacando-se a Regra 1: Um arco segmentável sempre pode ser decomposto em um ponto que é o mais distante de \overline{AB} entre aqueles pontos cujas mudanças angulares são maiores do que E_a .

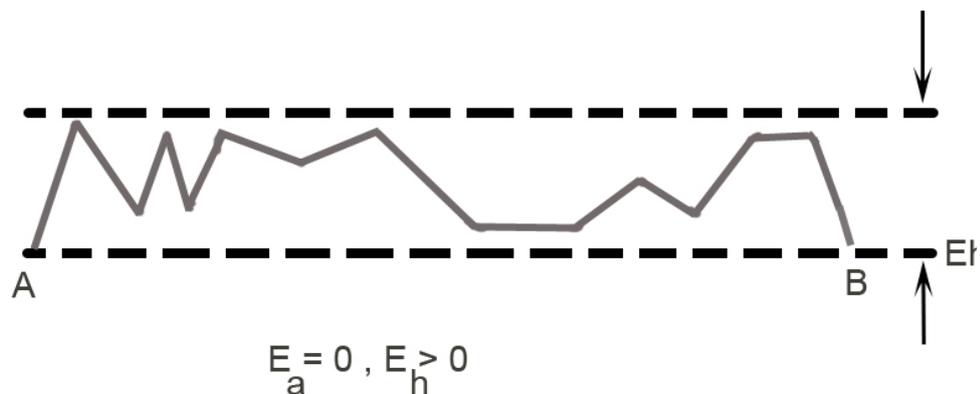


Figura 5.11: Um arco com irregularidades, passível de ser representado por um único segmento digitalizado. (adaptado de You e Fu (1979)).

Em situações como a da Figura 5.11 é vantajoso utilizar primitivas de dimensões grandes o suficiente para serem compatíveis relativamente com a dimensão do segmento \overline{AB} , e não com a dimensão dos pequenos segmentos $v_1, v_2 \dots v_n$ componentes do arco.

A principal finalidade da transformação de escala da Definição 52, recomendada por esta pesquisa, passa a ser em função desses comprimentos relativos de partes do arco ao arco como um todo, e não eliminar o problema dos comprimentos de arcos variáveis descrito por You e Fu (1979). Pela transformação de escala, arcos identificados como linhas retas apenas pela geometria discreta aritmética de (REVEILLÈS, 1991), podem ser avaliados corretamente pelo modelo proposto, numa escala compatível. Portanto, a escala adaptável reutiliza, num novo contexto, uma técnica conhecida desde a década de 70 do século passado.

O tópico a seguir complementa a escala adaptável, para que arcos como o da Figura 5.11 sejam classificados como um único segmento digitalizado. Isso pode ser feito incorporando ao modelo proposto a capacidade de discriminar alterações bruscas de curvaturas que ocorrem numa fração do arco total, relativamente ao comprimento do arco como um todo, e não apenas das alterações em ângulo.

5.3.2 Inferência de segmentos digitalizados

No item 5.2.4 foi constatado que o modelo proposto de segmento digitalizado adaptativo incorpora uma vizinhança adaptativa convenientemente para arcos com funções de curvatura suaves. Porém, esse modelo não é adequado para representação de alterações bruscas de curvatura com relação ao comprimento do arco como um todo.

Assim, a sugestão de trabalho futuro deste tópico pretende incorporar ao modelo proposto, a discriminação das alterações bruscas de curvatura, em termos de seus comprimentos relativos, com relação ao arco como um todo, e não apenas das alterações em ângulo.

A Definição 53 se relaciona a arcos com funções de curvatura suaves denominados de primeira ordem; enquanto a Definição 54 se refere a arcos com variações abruptas em sua curvatura relativamente ao comprimento total do arco, e, portanto, com o padrão de variação de curvatura diverso dos de primeira ordem.

Definição 53 *Função de curvatura de primeira ordem: funções de curvatura de primeira ordem correspondem a arcos digitais em que não ocor-*

rem alterações no símbolo isolado ao longo de todo o arco.

Definição 54 *Função de curvatura de ordem superior: funções de curvatura de ordem superior correspondem a arcos digitais com alterações de direção. Tais alterações se refletem em mudanças do símbolo que ocorre isolado ao longo de todo o arco, aumentando a curvatura do mesmo nesses pontos de mudança de símbolo.*

Apesar de existir todo um acervo de trabalhos na literatura sobre a determinação da curvatura discreta, além dos demais atributos de arcos, este tópico introduz o assunto por meio dos estudos de Kankanhalli (1993), de Freeman e Davis (1977) e Liu e Srinath (1990).

Existem soluções adaptativas para determinação de atributos de arcos que levam em conta os pontos dominantes detectados, o que reduz o custo computacional relativamente aos métodos que não utilizam os pontos dominantes (KANKANHALLI, 1993). Conforme já comentado sobre o trabalho de Kankanhalli (1993), os pontos dominantes e correspondentes vizinhanças (denominadas regiões de suporte) são importantes por transmitirem a maioria das informações sobre os atributos dos arcos. Porém, o mais comum é que essas regiões de suporte sejam pré-definidas (KANKANHALLI, 1993).

Ao se interpretar um contorno digital complexo, composto por arcos orientados em diversas direções, como um percurso a ser percorrido pelo autômato, é sugerida uma arquitetura para reconhecimento desse contorno, esquematizada na Figura 5.16, fundamentada em técnicas de inferência de arcos (IWAI, 2000) (MATSUDO, 2006). Essa Figura se baseia no mapeamento de ambientes desconhecidos por robôs móveis, nos moldes de Sousa (2006). A vantagem dessa arquitetura está em considerar os pontos dominantes e regiões de suporte variáveis adaptativamente em função de comprimentos relativos de arcos.

O trabalho de Sousa (2006) se propõe ao mapeamento de ambientes desconhecidos e controle dos deslocamentos de um robô nesses ambientes, por meio de uma arquitetura composta de um sistema gerente, um autômato de exploração e um autômato de mapeamento, cujas funções são descritas resumidamente a seguir:

- O Sistema Gerente tem a função de coletar e organizar as informações sobre o ambiente, para formar a cadeia de entrada do Autômato de Exploração. Além disso, tem a função de coordenar os autômatos de exploração e de mapeamento;

- O Autômato de Mapeamento armazena os dados indicativos da topologia do ambiente que está sendo mapeado pelo robô;
- O Autômato de Exploração tem como função determinar a direção do próximo deslocamento efetuado pelo sistema motor do robô durante a exploração de um ambiente desconhecido. Esta decisão é feita com base nos dados coletados pelo sistema sensor sobre a existência ou não de obstáculos no perímetro circundante ao robô, nas informações previamente registradas pelo Autômato de Mapeamento e no algoritmo de exploração executado por este autômato (SOUSA, 2006).

5.3.2.1 Fluxo de dados para reconhecimento de contornos complexos

A fim de utilizar os recursos da adaptatividade, sugere-se uma arquitetura hierárquica e distribuída em diversos autômatos finitos adaptativos para um sistema que reconheça arcos complexos. Nessa arquitetura, propõe-se uma camada adaptativa mestre, no nível mais elevado da hierarquia, com finalidade de atuar como sistema gerente, que monitora os autômatos finitos adaptativos secundários (escravos). A camada adaptativa mestre pode inclusive criar novos autômatos finitos adaptativos de acordo com os estímulos e os estados dos autômatos finitos adaptativos secundários.

Esta pesquisa teve início com estudos sobre autômatos finitos adaptativos que reconhecem formas com lados paralelos às primitivas, mesmo que os lados apresentem erros (devido ao ruído) dentro de uma região de tolerância (uma vizinhança adaptativa), exemplificado pelo quadrado da Figura 5.12. Posteriormente, partindo desses estudos, a pesquisa utilizou os autômatos finitos adaptativos resultantes dessa fase inicial, alterando as suas topologias para aceitação de segmentos digitalizados e formas básica num determinado ângulo com relação às primitivas. A idéia desta arquitetura é que a criação de autômatos, as alterações de regiões de tolerância, alterações de ângulos e escalas dos segmentos digitalizados sejam executadas adaptativamente pelo sistema, de acordo com os estímulos, sem interferência de agentes externos incluindo o operador.

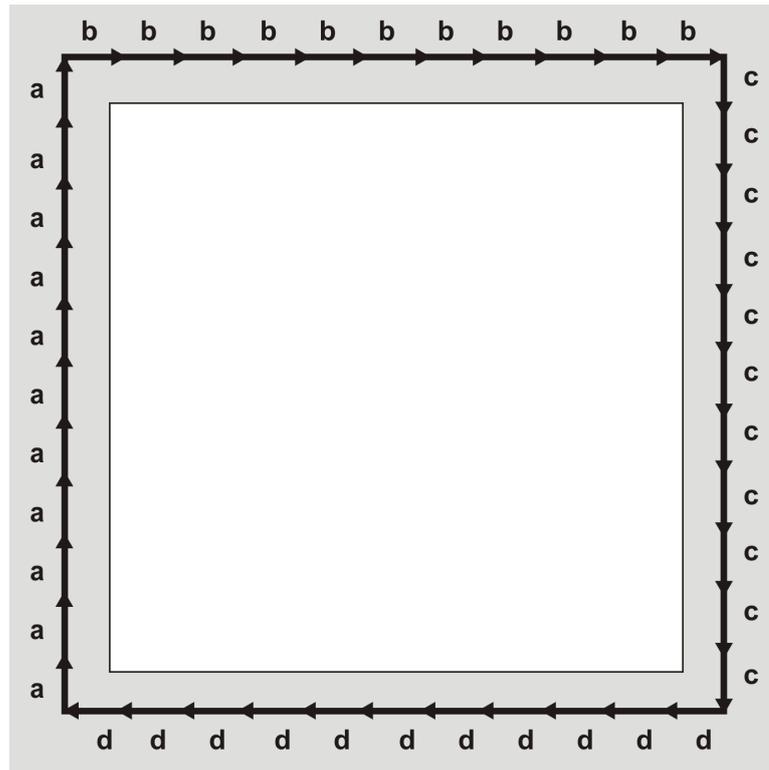


Figura 5.12: Exemplo de região de tolerância (vizinhança adaptativa) para o quadrado representado no centro da região.

Uma maneira compacta de representar os arcos é por uma vizinhança adaptativa definindo uma região de tolerância (célula) no espaço tridimensional da Figura 5.13 de modo que a cada arco sejam associados a mencionada região e um número identificador.

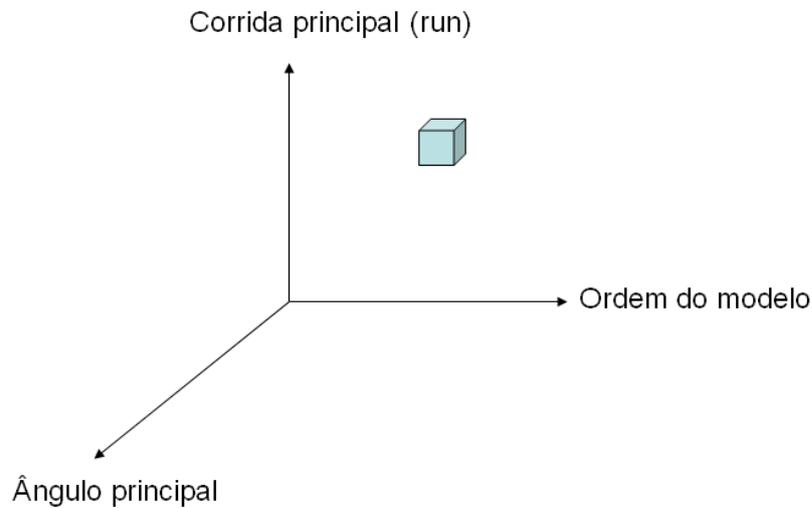


Figura 5.13: Uma célula no espaço tridimensional correspondente a uma região de tolerância especificada por vizinhança adaptativa.

Na Figura 5.13 um arco qualquer é representado pelos seguintes atributos, variáveis conforme as suas tolerâncias individuais compondo uma célula em um espaço tridimensional:

- ângulo principal,
- ordem de modelo (a fim de indicar as variações permitidas nos atributos dos arcos),
- a corrida principal.

O diagrama IDEF0 do sistema proposto é apresentado na Figura 5.14 composto de uma unidade de interface e a unidade de Análise e Classificação.

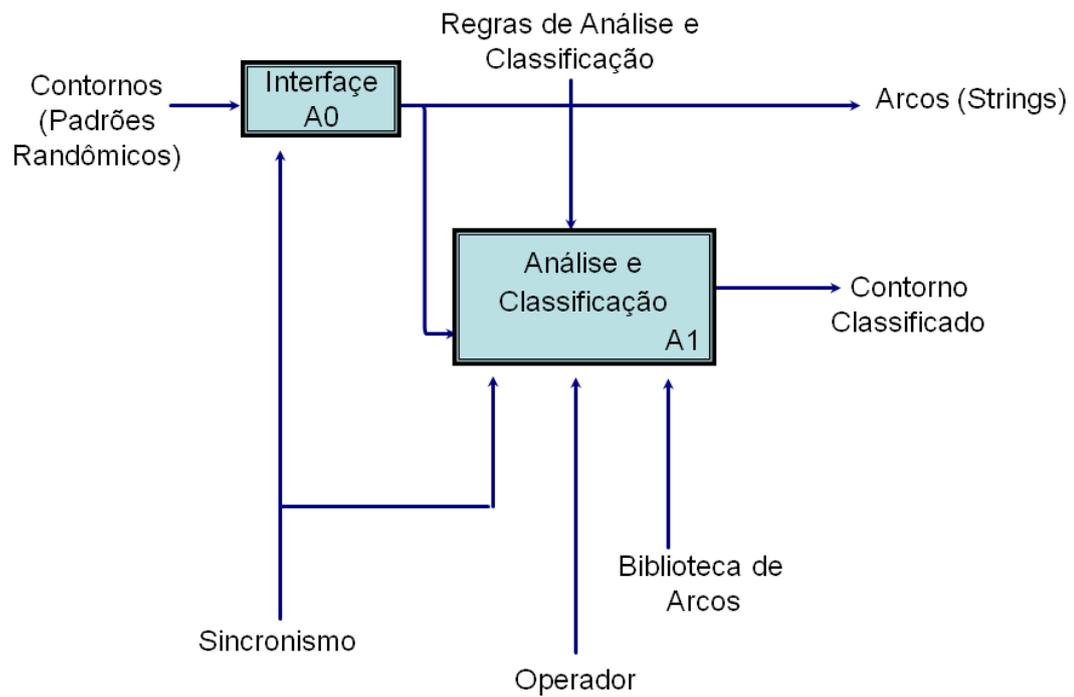


Figura 5.14: Diagrama IDEF0 do sistema.

A Figura 5.15 detalha o diagrama IDEF0 da unidade de Análise e Classificação mostrada na Figura 5.14. Na Figura 5.15, as unidades Domínio de Célula, Análise de Tokens, Reconhecimento e Identificação formam estruturas em *pipeline* para processamento das cadeias, centradas em segmentos digitalizados adaptativos de Mapeamento. O cerne da unidade de Inferência é o segmento digitalizado adaptativo de Exploração.

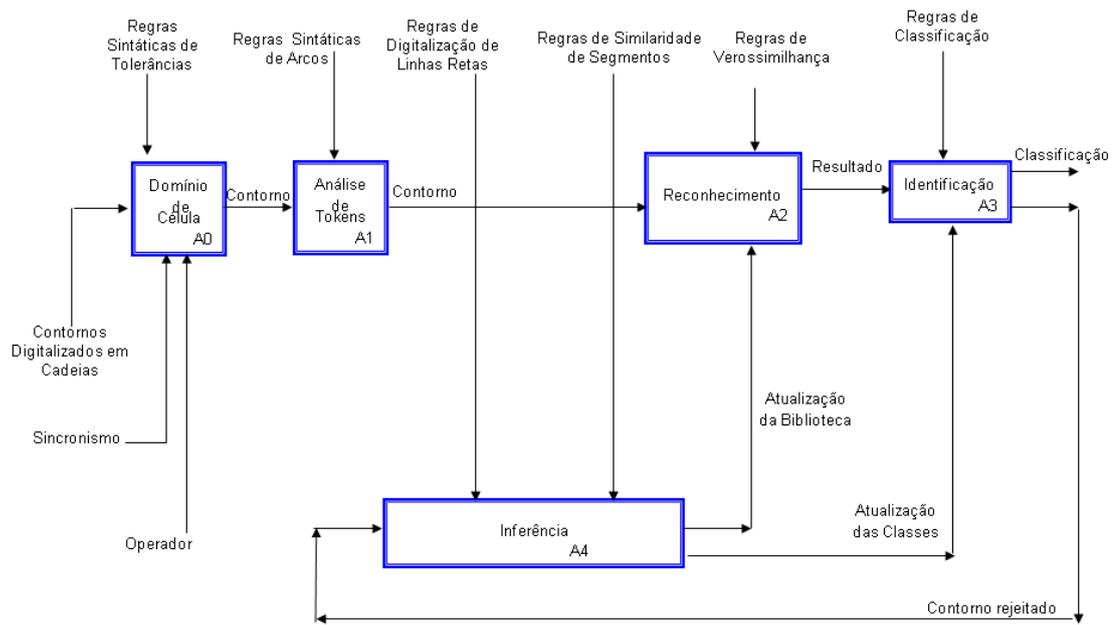


Figura 5.15: Diagrama IDEF0 detalhado da unidade de Análise e Classificação.

A Figura 5.15 menciona uma Biblioteca que tem o propósito de armazenar os atributos de arcos aceitos e rejeitados. Caso determinado arco rejeitado ocorra repetidamente dentro de uma região de tolerância, esse arco passa a ser classificado como aceito pelo sistema pela atualização de determinado autômato finito adaptativo já existente, ou um novo autômato finito adaptativo é criado. A cada novo arco incorporado como aceito, as regiões e os números identificadores das células existentes são atualizadas.

A Figura 5.16 mostra uma configuração, alterável adaptativamente, composta por um sistema gerente (a camada adaptativa mestre), um autômato finito adaptativo que implementa um segmento digitalizado adaptativo de exploração (segmento digitalizado adaptativo 1) e um autômato finito adaptativo que implementa um segmento digitalizado adaptativo de mapeamento (segmento digitalizado adaptativo 2).

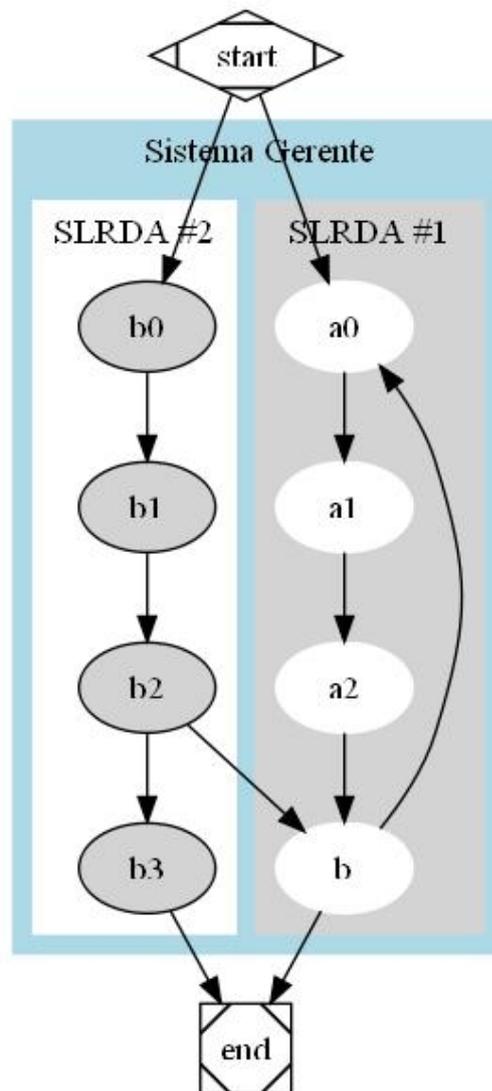


Figura 5.16: Esquematização do fluxo de dados sugerido em que o sistema gerente é a camada adaptativa mestre.

A unidades apresentadas na Figura 5.16 estão descritas a seguir:

- O Sistema Gerente é uma camada adaptativa hierarquicamente superior às demais camadas adaptativas dos autômatos finitos adaptativos do sistema, que tem a função de monitorar e coordenar os segmentos digitalizados adaptativos de exploração e de mapeamento, sendo que a quantidade de segmentos digitalizados adaptativos altera-se conforme os estímulos pelo gerenciamento dessa camada;
- O segmento digitalizado adaptativo de Mapeamento armazena os dados indicativos do contorno que está sendo percorrido pelo conjunto de unidades, principalmente quanto aos comprimentos dos arcos;
- O segmento digitalizado adaptativo de Exploração tem a função de atuar nas mudanças de direção do próximo deslocamento efetuado pelo conjunto, durante

a exploração do contorno, decidindo favoravelmente ou não pela mudança. Por exemplo, no caso de uma região de alta curvatura e comprimento muito pequeno relativamente ao arco global, o segmento digitalizado adaptativo de Exploração evitaria a mudança de direção. Esta decisão é feita com base em regiões de suporte variáveis adaptativamente, dos dados coletados sobre a existência ou não de mudanças de curvaturas ou direções, nas informações previamente registradas pelo segmento digitalizado adaptativo de Mapeamento e no algoritmo de exploração executado por este autômato.

Quanto aos algoritmos dos segmentos digitalizados adaptativos de mapeamento e exploração, nesta pesquisa foram estudados modelos de cadeias de segmentos digitalizados, associados aos correspondentes segmentos digitalizados adaptativos. Para tais modelos, esses algoritmos envolvem técnicas de inferência de segmentos digitalizados, a fim de que os mesmos se auto-ajustem, levando em consideração a similaridade em padrões repetitivos de segmentos. Estudos de inferência gramatical pela adaptatividade foram apresentados em Iwai (2000) e Matsudo (2006).

5.3.3 Função de vizinhança

Apesar do trabalho de Nacken (1993) ter aplicações recentes com bons resultados, tal como apresentado em Park (2009), é conveniente pesquisa complementar de novos modelos de função de vizinhança que integrem regras geométricas básicas. Uma das alternativas ao trabalho de Nacken (1993) é a *Uncertain Geometry* que usa a pré-imagem para definir princípios geométricos tais como o paralelismo digital (VEELAERT, 1999) (VEELAERT, 2005).

5.4 Posicionamento desta pesquisa quanto ao estado-da-arte

Concluindo sobre os itens anteriores deste tópico, observam-se duas tendências principais do estado-da-arte, em que esta tese surge apresentando uma terceira: a primeira classe, representada pelo trabalho de Feschet (2008) que se propõe a obter soluções conceituadas nas linhas discretas aritméticas, no fecho convexo do conjunto de pontos, nos comprimentos de reticulados e na razão de direção. A segunda classe, exemplificada pelo trabalho de Bhowmick e Bhattacharya (2007), busca soluções aproximadas

de retas seguindo o modelo de Freeman, normalmente sem considerar o polígono representativo do conjunto de pontos. Esta pesquisa introduz uma terceira tendência, com o conceito da linearidade adaptativa, aplicável à primeira e à segunda classe mencionadas, em que, por exemplo, arcos irregulares podem ser, na verdade, retas quando observados na escala adequada. Partindo da segunda classe, uma das características desta pesquisa foi implementar algoritmos de apenas uma passada, aprimorando a representação de retas e de seus parâmetros, numa vizinhança adaptativa. Pela generalização deste trabalho, reutilizando algoritmos clássicos, bem como aplicando os recursos sintáticos conhecidos, a adaptatividade pode integrar as duas tendências iniciais. O texto a seguir complementa o exposto, apresentando uma segunda classificação.

Levando em conta o trabalho de Neto (2009), surge uma segunda classificação que resulta nas seguintes tendências dos dispositivos ou algoritmos: guiado por regras (não-adaptativo) cujo conjunto de regras é invariável exemplificado pelo uso apenas da propriedade da corda; guiado por regras (com adaptatividade básica) cujo conjunto de regras é variável, exemplificado pelos trabalhos de Bhowmick e Bhattacharya (2007) e Feschet (2008); e guiado por regras (com adaptatividade hierárquica multi-nível) cujo conjunto de regras é variável apresentando funções adaptativas modificáveis, na qual se insere esta pesquisa. Resulta portanto em todo um potencial evolutivo deste trabalho partindo da reutilização e generalizações de formalismos que têm sido tradicionalmente utilizados.

6 SIMULAÇÕES COMPLEMENTARES

A pesquisa bibliográfica efetuada neste estudo resultou na identificação de uma lacuna gerando a hipótese da viabilidade em modelar propriedades das retas digitais por um conjunto de regras. Detalhes sobre as estruturas computacionais associadas a esses modelos e correspondentes simulações foram apresentadas nos capítulos precedentes.

Neste capítulo, descrevem-se testes complementares das estruturas computacionais estudadas nos capítulos anteriores e a análise dos resultados obtidos por meio de estudos de casos. O conhecimento dessas estruturas facilita a modelagem dos autômatos, pois as mesmas tendem a se repetir em qualquer reconhecedor. Especificamente, as simulações deste capítulo utilizam os segmentos digitalizados adaptativos implementados, contribuindo para a coleta de dados e consequente análise final da pesquisa.

Os estudos de caso se concentram na classificação de formas geométricas básicas por meio de segmentos digitalizados adaptativos.

6.1 Implementações realizadas

Os modelos de segmentos digitalizados adaptativos descritos nos Capítulos 4 e 5 foram pesquisados e implementados. Neste item, descrevem-se os demais modelos pesquisados e implementados, utilizando a vantagem da solução adaptativa, levando em conta as funções adaptativas descritas naqueles capítulos. Essa vantagem de reutilização das funções adaptativas possibilita a implementação gradual dos autômatos, por exemplo, a função adaptativa IB nos diferentes autômatos finitos adaptativos é aprimorada gradativamente, conforme a nova aplicação de uma certa configuração inicial. Implementações paramétricas trazem outra vantagem da solução adaptativa, em que a função adaptativa RA é utilizada em vários autômatos deste trabalho.

Independentemente das causas de erro de fases anteriores da digitalização, ou seja, algoritmos de segmentação, que alimentam uma cadeia W na entrada dos segmentos digitalizados adaptativos, as simulações pretendem avaliar pequenos erros, segregados em três características: erros em ângulo, erros em comprimento, e de escalonamento. Necessariamente, similaridade entre segmentos digitalizados é uma questão essencial a ser considerada.

Ironicamente, em classificação de formas, quanto mais elementar é a etapa de segmentação na execução da extração de atributos pelas primitivas selecionadas, maior nível de complexidade é requerido para o classificador, assim como maior poder computacional. Esse fato resultou em soluções que utilizam autômatos finitos ou linguagens regulares em classificação para tornar essa etapa mais simples, com o custo de aumento da complexidade das primitivas na segmentação (YOU; FU, 1979). De imediato, constata-se que o segmento digitalizado libera a etapa de segmentação dessa complexidade, ao concentrar na classificação a complexidade que seria delegada às etapas anteriores do processo. Além disso, o segmento digitalizado adaptativo proposto por este trabalho utiliza o mesmo ferramental teórico básico dos autômatos finitos, tendo poder computacional equivalente ao da máquina de Turing.

Todos os segmentos digitalizados adaptativos implementados podem passar pelos testes de erros em ângulo, erros em comprimento, e de escalonamento em função da tolerância. Entretanto, foram implementados segmentos digitalizados adaptativos específicos para cada teste, com propósito de ter um efeito mais evidente do ângulo de orientação dos segmentos.

Os segmentos digitalizados adaptativos, no seu processo dinâmico, constroem *templates*, semelhantes ao do exemplo da Figura 6.1 para o primeiro quadrante, com b ocorrendo isolado. Os itens a seguir comentam como os *templates* são construídos pelos mecanismos.

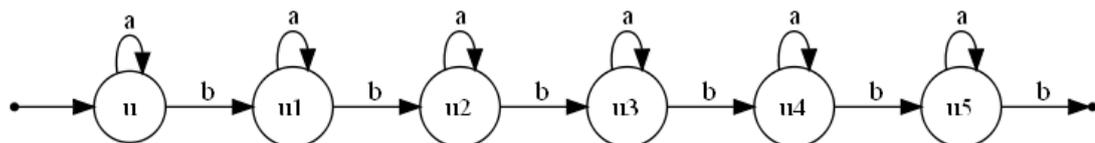


Figura 6.1: Um modelo simplificado de segmento digitalizado. Exemplo com 6 unidades de segmento.

6.1.1 Similaridade de formas quanto aos erros em comprimento sem considerar parâmetros de correção

O autômato finito adaptativo da Figura 6.2 e dos Quadros 8 e 9 tem a finalidade de avaliar os erros em ângulo e comprimento sem considerar parâmetros de correção ψ , a partir de cadeias de entrada do tipo $W = S_1S_2S_3S_4$, que é uma concatenação de quatro segmentos digitalizados S_1, S_2, S_3, S_4 . Esse autômato é uma extensão do autômato finito adaptativo das Figuras 4.3 e 4.4 que reconhece o quadrado, utilizando a mencionada função adaptativa RA.

Um aprimoramento é que para cada unidade de segmento de S_1 consumida, a função adaptativa IB é ativada para construir os modelos dos próximos três segmentos digitalizados. Esses modelos são construídos entre os seguintes estados:

- s a t_1 :segundo segmento digitalizado;
- w a w_1 :terceiro segmento digitalizado; e
- p a p_1 :quarto segmento digitalizado.

A fim de facilitar a visualização, os comprimentos dos segmentos digitalizados são medidos pela quantidade de unidades de segmento. Desse modo, para comparar o comprimento de dois segmentos digitalizados com o mesmo ângulo de inclinação principal, basta contar o número de unidades de segmento dos respectivos segmentos $\lambda_{S_1}, \lambda_{S_2}, \lambda_{S_3}, \lambda_{S_4}$, de acordo com a Expressão 4.1. No caso do quadrado, a primeira condição para as cadeias que o codificam serem aceitas pelo autômato finito adaptativo é $\lambda_{S_1} = \lambda_{S_2} = \lambda_{S_3} = \lambda_{S_4}$.

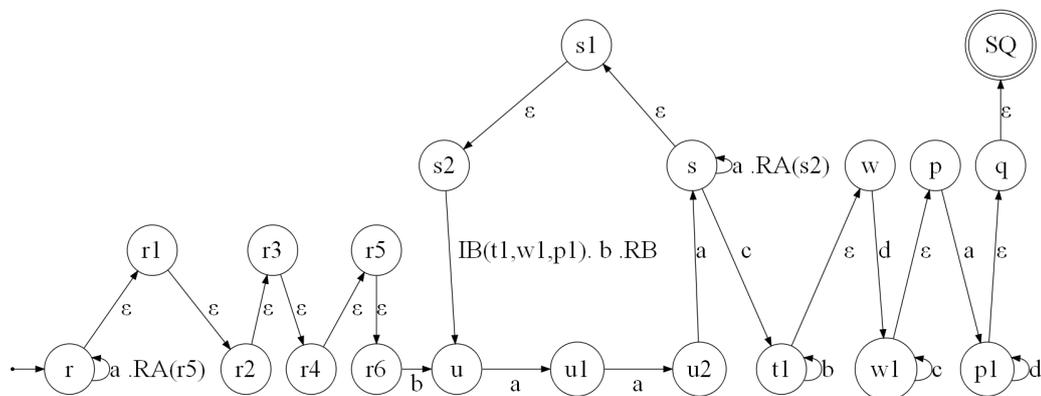


Figura 6.2: Configuração inicial de autômato finito adaptativo que implementa segmento digitalizado adaptativo

$ \begin{aligned} & IB(\rho_1, \rho_2, \rho_3) \{ vr_1, tn^*, wn^*, pn^* : \\ & - [(\rho_1, \varepsilon) \rightarrow w] \\ & - [(\rho_2, \varepsilon) \rightarrow p] \\ & - [(\rho_3, \varepsilon) \rightarrow q] \\ & + [(tn, b) \rightarrow tn] \\ & + [(\rho_1, c) \rightarrow tn] \\ & + [(tn, \varepsilon) \rightarrow w] \\ & + [(wn, b) \rightarrow wn] \\ & + [(\rho_2, c) \rightarrow wn] \\ & + [(wn, \varepsilon) \rightarrow p] \\ & + [(pn, b) \rightarrow pn] \\ & + [(\rho_3, c) \rightarrow pn] \\ & + [(pn, \varepsilon) \rightarrow q] \\ & - [(vr_1, b) : IB(\rho_1, \rho_2, \rho_3) \rightarrow u : RB] \\ & + [(u_2, b) : IB(tn, wn, pn) \rightarrow u : RB] \} \end{aligned} $
--

Quadro 8: função adaptativa IB do autômato finito adaptativo da Figura 6.2

$ \begin{aligned} & RB \{ vr_1 : \\ & - [(s, \varepsilon) \rightarrow s_1] \\ & - [(s_1, \varepsilon) \rightarrow s_2] \\ & + [(s, \varepsilon) \rightarrow s_1] \\ & + [(s_1, \varepsilon) \rightarrow s_2] \} \\ & - [(s, a) \rightarrow s : RA(vr_1)] \\ & + [(s, a) \rightarrow s : RA(s_2)] \\ & \} \end{aligned} $
--

Quadro 9: Função adaptativa RB do autômato finito adaptativo da Figura 6.2

6.1.2 Similaridade de formas quanto aos erros em ângulo sem considerar parâmetros de correção

Para demonstrar os efeitos dos erros de inclinação, o autômato do item anterior foi alimentado com cadeias $W = S_1S_2S_3S_4$ correspondentes à Figura 6.3, Figura 6.4, Figura 6.5 e Figura 6.6 variando a inclinação de cada segmento digitalizado individual S_1, S_2, S_3, S_4 .

Todas as cadeias do Quadro 10 foram aceitas, exceto a correspondente à Figura 6.6.

Embora a cadeia $[(ba^4)^4b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^3(ad^4)a]$ da Figura 6.3 seja constituída por segmentos digitalizados com inclinações variáveis, a sequência de entrada foi aceita pelo autômato finito adaptativo. O primeiro lado do quadrado não tem variações em inclinação. Entretanto, a combinação das variações em inclinação dos demais lados distorce o polígono. Mesmo com as distorções, o autômato finito

Figura	Codificação
6.3	$[(ba^4)^4b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^3(ad^4)a]$
6.4	$[(ba^3)(ba^4)(ba^3)(ba^4)b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^4a]$
6.5	$[(ba^3)^4b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^4a]$
6.6	$[(ba^3)(ba^5)(ba^3)^2b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^4a]$

Quadro 10: Codificação das cadeias que alimentaram o autômato finito adaptativo da Figura 6.2 e Quadro 8, relacionadas às figuras das correspondentes representações gráficas. Apenas a cadeia relativa à Figura 6.6 foi rejeitada pelo autômato finito adaptativo

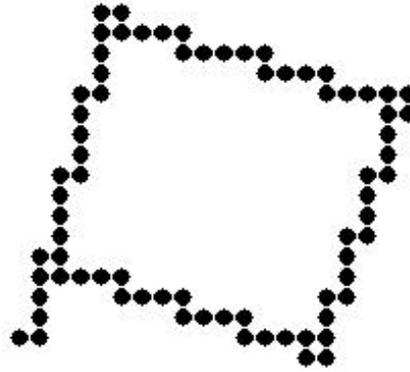


Figura 6.3: Representação gráfica de cadeia aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8.

adaptativo é capaz de classificar corretamente a forma geométrica global, aceitando a sequência, que seria rejeitada por outros métodos com características locais.

A cadeia $[(ba^3)(ba^4)(ba^3)(ba^4)b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^4a]$ da Figura 6.4 foi aceita. Comparando-se essa cadeia com a da Figura 6.3, a primeira apresenta variações em inclinação no primeiro lado do quadrado, que a segunda não tem. O resultado na Figura 6.4 foi de maneira a combinar as distorções dos quatro lados, o que levou a reduzir, visualmente, a distorção da forma como um todo.

A sequência $[(ba^3)^4b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^4a]$ foi aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8. Essa cadeia corresponde a uma forma com menos distorções do que aquelas das Figura 6.3 e Figura 6.4. O primeiro, terceiro e quarto lados do polígono não têm muitas alterações em inclinação. As alterações em inclinação do terceiro lado são difíceis de visualizar, mesmo não sendo um segmento digitalizado ideal. Entretanto, visualmente, mesmo com as alterações em inclinação, a forma do quadrado é facilmente identificada.

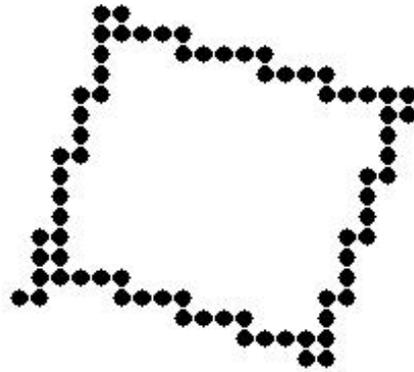


Figura 6.4: Representação gráfica de cadeia aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8

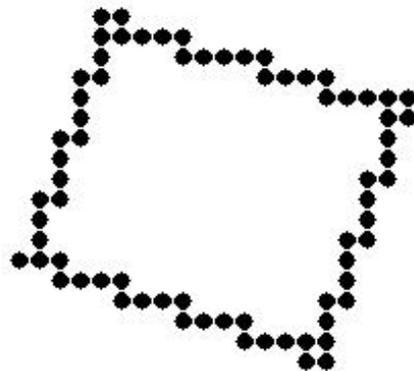


Figura 6.5: Representação gráfica de cadeia aceita pelo autômato finito adaptativo da Figura 6.2 e Quadro 8

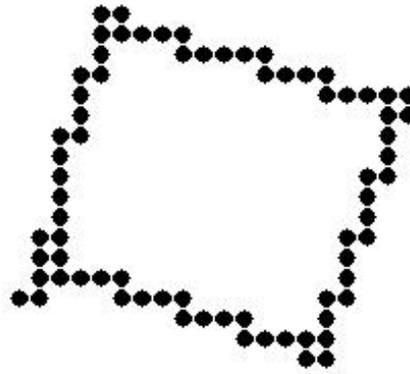


Figura 6.6: Representação gráfica de cadeia rejeitada pelo autômato finito adaptativo da Figura 6.2 e Quadro 8

A sequência $[(ba^3)(ba^5)(ba^3)^2b(cb^3)(cb^4)(cb^3)(cb^4)c(dc^3)^4d(ad^3)^4a]$ da Figura 6.6 foi rejeitada pelo autômato finito adaptativo da Figura 6.2 e Quadro 8. Essa cadeia foi rejeitada devido à unidade de segmento (ba^5) estar fora do modelo definido no autômato, o que é constatado fácil e visualmente na formação geral das unidades de segmento do quadrado correspondente. Poucas alterações em funções adaptativas, tal como a função adaptativa RA, permitiria aceitar a sequência, ajustando o classificador se exigido pelas especificações. Ao invés desse ajuste, outra alternativa para aceitar essa cadeia seria aumentar a precisão dos algoritmos de segmentação.

6.1.3 Similaridade de formas quanto ao escalonamento

O autômato finito adaptativo da Figura 6.7 e Quadro 11, foi construído aperfeiçoando funções adaptativas descritas nos autômatos anteriores. Este autômato finito adaptativo modela triângulos codificados por cadeias do tipo $W = S_1S_2S_3$, em que as sub-cadeias S_1 , S_2 e S_3 são segmentos digitalizados correspondentes a cada lado do polígono.

Os testes devem avaliar o escalonamento dos segmentos digitalizados de $W(S_1, S_2, S_3)$ associados aos respectivos erros em ângulo e comprimento. Os segmentos digitalizados individuais podem, em princípio, apresentar comprimento tão grande quanto possível, no entanto, o seu menor comprimento é limitado a uma unidade de segmento por segmento. A cada i -ésima unidade de segmento do segmento digitalizado S_1 a função adaptativa IB constrói a unidade de segmento correspondente de S_2 e S_3 de acordo com os geradores do Quadro 11.

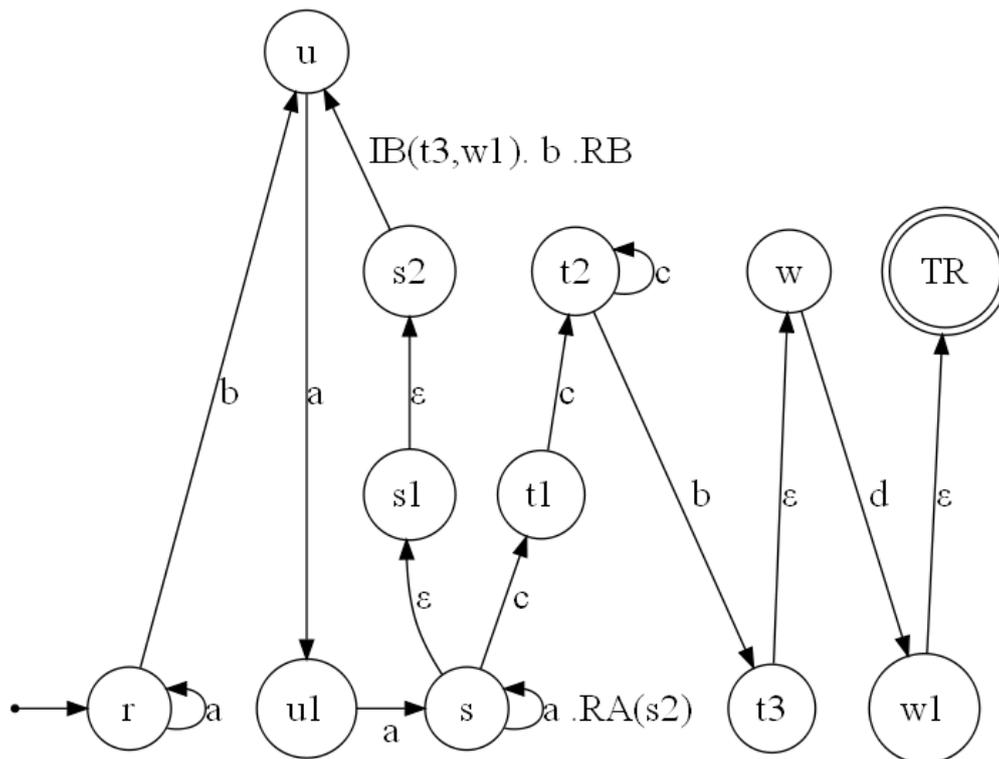


Figura 6.7: Configuração inicial de autômato finito adaptativo que implementa segmento digitalizado adaptativo

O triângulo da Figura 6.8 é codificado pela cadeia $(a^3b)^6(c^3b)(c^4b)(c^3b)^4d^{12}$, enquanto o da Figura 6.9 é codificado pela cadeia $(a^3b)^9(c^3b)(c^4b)(c^3b)^7d^{18}$. Essas duas figuras representam a mesma forma triangular em escalas diferentes. Os dois triângulos foram reconhecidos pelo autômato finito adaptativo.

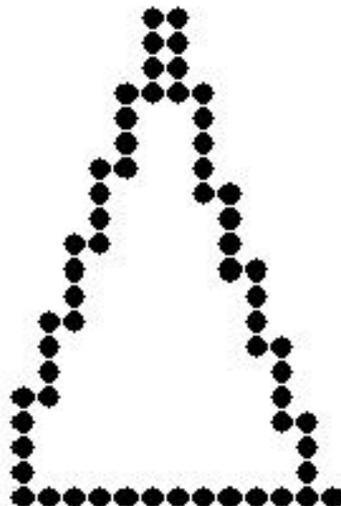


Figura 6.8: Exemplo de triângulo classificado corretamente pelo autômato finito adaptativo da Figura 6.7.

$ \begin{aligned} &RB\{vr1 : \\ &- [(s, \varepsilon) \rightarrow s_1] \\ &- [(s_1, \varepsilon) \rightarrow s_2] \\ &+ [(s, \varepsilon) \rightarrow s_1] \\ &+ [(s_1, \varepsilon) \rightarrow s_2] \} \\ &- [(s, a) \rightarrow s : RA(vr1)] \\ &+ [(s, a) \rightarrow s : RA(s_2)] \\ &\} \end{aligned} $	$ \begin{aligned} &IB(\rho1, \rho2)\{vr, tn1^*, tn2^*, tn3^*, wn^* : \\ &- [(\rho1, \varepsilon) \rightarrow w] \\ &- [(\rho2, \varepsilon) \rightarrow TR] \\ &+ [(\rho1, c) \rightarrow tn1] \\ &+ [(tn1, c) \rightarrow tn2] \\ &+ [(tn2, b) \rightarrow tn3] \\ &+ [(tn2, c) \rightarrow tn2] \\ &+ [(tn3, \varepsilon) \rightarrow w] \\ &+ [(\rho2, d) \rightarrow wn] \\ &+ [(wn, \varepsilon) \rightarrow TR] \} \\ &- [(vr, b) : IB(\rho1, \rho2) \rightarrow u : RB] \\ &+ [(s_2, b) : IB(tn3, wn) \rightarrow u : RB] \\ &\} \end{aligned} $
--	---

Quadro 11: Funções adaptativas do autômato finito adaptativo da Figura 6.7

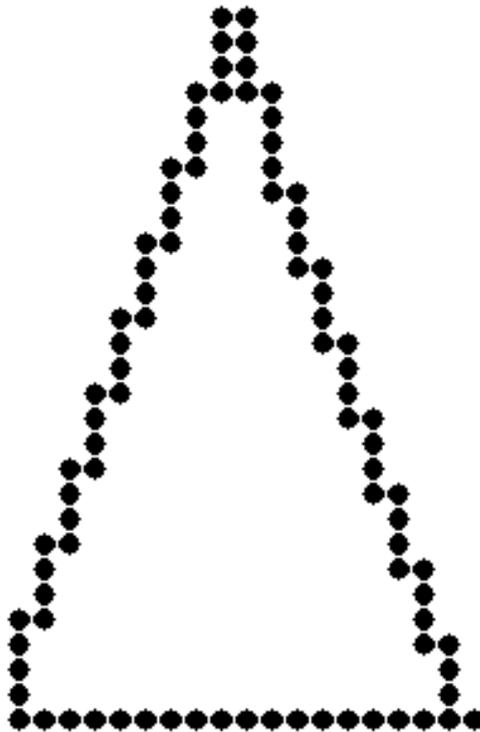


Figura 6.9: A mesma forma do triângulo da Figura 6.8, em maior escala, classificada corretamente pelo autômato finito adaptativo da Figura 6.7.

6.1.4 Similaridade de formas considerando erros em comprimento e ângulo

A intenção dos testes deste item é avaliar o processo adaptativo para modelar parâmetros de correção de comprimento de segmentos digitalizados afetados por erros

Caso a tolerância seja informada ao autômato finito adaptativo por *tokens* y , a função adaptativa RO tem a finalidade de construir o *loop* adaptativo, de acordo com a quantidade de *tokens* y , e também de criar os três ponteiros. A transição $[(r_1, a) \rightarrow r_1]$ é uma simplificação para modelar os possíveis símbolos a da unidade de segmento inicial.

Com o primeiro símbolo b , o autômato finito adaptativo passa a consumir as unidades de segmento subsequentes. Cada unidade de segmento da primeira sub-cadeia ativa as funções adaptativas IB e RB. Com a ativação da função adaptativa IB, ocorrem as ações de girar o ponteiro p_a no *loop*, adicionalmente à construção dos modelos dos dois segmentos digitalizados seguintes, entre os estados t_1 a t_2 (segundo segmento digitalizado), t_3 a t_4 (terceiro segmento digitalizado)

A cada volta completa no *loop* efetuada por p_a , a função adaptativa IB é ativada, e as ações seguintes são executadas:

1. Inclusão de uma nova unidade de segmento no *loop* $[(t_2, b) \rightarrow t_2 : OT]$; e, segundo, inclui outra unidade de segmento no *loop* $[(w, d) \rightarrow w : OT]$;
2. Inserção de uma transição do último estado de cada unidade de segmento incluída no passo 1: no *loop* (o primeiro) uma transição com o símbolo d para o estado t_3 ; no segundo, uma transição em vazio para o estado TR ;
3. Movimentação de p_c para que aponte a um estado apenas uma unidade de segmento para baixo (ou seja na direção de t_1), inserção de uma transição com o símbolo d do estado apontado por p_c para o estado t_3 ;
4. Movimentação de p_d para que aponte apenas uma unidade de segmento abaixo (em direção a t_3), inserção de uma transição em vazio do estado apontado para o estado final TR .

No entanto, caso a função adaptativa OT seja ativada, a transição $[(w, \varepsilon) \rightarrow TR]$ é removida; indicando assim que o comprimento de algum segmento digitalizado é muito grande, fora de tolerância, rejeitando os segmentos digitalizados de entrada por impedir o autômato finito adaptativo de alcançar o estado final. Analogamente, segmentos digitalizados pequenos são rejeitados porque o autômato finito adaptativo não encontra pelo menos uma transição com o símbolo d (introduzidas no passo 3) para o estado t_3 ; ou uma transição em vazio para estado final TR (introduzidas no passo 4).

Apenas dois exemplos de erros em comprimento de segmentos digitalizados são fornecidos, tendo em vista que qualquer forma dentro da tolerância ψ é aceita pelo autômato

finito adaptativo, independentemente de sua escala. A forma será rejeitada, caso não esteja dentro dessa tolerância em comprimento.

As formas da Figura 6.11 foram aceitas pelo autômato finito adaptativo com ψ definindo a tolerância em comprimento de 20%. A forma à esquerda dessa figura foi codificada pela cadeia $(a^3b)^9(c^3b)^8d^{17}$ e a da direita pela cadeia $(a^3b)^9(c^3b)^{11}d^{20}$.



Figura 6.11: Formas classificadas corretamente com 20% de tolerância em comprimento.

Ambas as formas são facilmente identificadas visualmente como triângulos, embora a forma à direita da figura não se feche, devido aos erros em comprimento, dividindo o contorno do polígono. Entretanto, a outra forma, à esquerda da figura, apresenta um contorno menor do que seria o triângulo ideal, deixando de fora a unidade de segmento inicial, correspondente ao primeiro lado. Comparando-se as codificações das duas formas, as distorções nos respectivos contornos ocorreram pelos seguintes motivos:

- Os comprimentos do segundo lado de cada forma mudou de 8 (oito) vezes unidades de segmento para 11 (onze) unidades de segmento;
- Os comprimentos do terceiro lado de cada forma mudou de 17 (dezessete) símbolos d para 20 símbolos d .

6.2 Conclusões sobre os estudos de caso

Essencialmente, os erros de ângulo de segmentos digitalizados apresentam um caráter local, em relação à análise de erros em comprimento, que é mais global, pois esta apresenta maior nível de informação sobre as formas. Com efeito, caso de uma retitude relativa possa ser garantida pelos algoritmos de segmentação, o classificador pode se concentrar em avaliar a distribuição ou arranjo das unidades de segmento relativamente à geometria da forma no seu aspecto global. Assim, neste caso, a classificação seria menos complexa do que se tivesse de realizar análises de retitude (YOU; FU, 1979).

Em consequência da pouca dificuldade em alterar as ações adaptativas, a pesquisa e aprimoramentos da etapa de classificação tende a ser bem simplificada. Isso facilita, por exemplo, ajustar o comportamento do classificador para se adequar às necessidades dos algoritmos de segmentação.

Desta maneira, tornam-se viáveis as aplicações em Reconhecimento Sintático de Padrões, associadas a descritores de forma de baixa sofisticação, baseados na teoria dos autômatos finitos e utilizando o conceito de segmentos digitalizados adaptativos, caracterizadas por flexibilidade e não-estocástica.

O autômato finito adaptativo pode reconhecer linguagens definidas como uma composição de duas ou mais linguagens, possibilitando que apenas um único autômato possa executar a classificação de um conjunto de formas pré-definidas, em que seus estados finais estejam relacionados com as formas a serem reconhecidas.

Os estudos de caso consideraram formas básicas. Uma extensão imediata em Reconhecimento Sintático de Padrões é o reconhecimento de formas complexas pela segmentação de arcos digitais em segmentos digitalizados.

7 CONSIDERAÇÕES FINAIS

Neste capítulo apresenta-se um panorama geral do trabalho, descrevendo-se as contribuições, um resumo dos resultados obtidos e sugestões de trabalhos futuros.

7.1 Sumário geral

Existe uma infinidade de trabalhos sobre segmentos digitalizados, com vários referenciados por esta pesquisa e com breve resumo na Figura 7.1. Entretanto, iniciados em 1970 pela conjectura de Freeman, até hoje os resultados obtidos apresentam restrições quando relacionados a cenários sujeitos a influências espúrias. Tanto que a tese que introduziu a geometria discreta aritmética por Reveillès (1991) é aplicada na atualidade em abordagens que visam sanar tais restrições. Na Figura 7.1, após a conjectura de Freeman, houve um surto de pesquisas de métodos sintáticos, que se reduziu a partir da década de 90. Certamente isso ocorreu devido ao poder computacional requerido para a análise sintática de segmentos digitalizados, comentado por Brons (1974), inviabilizando a aplicação de formalismo simples como o do autômato finito e influenciando na redução de trabalhos baseadas em autômatos na literatura, mesmo na atualidade.

Por outro lado, a partir da tese “Contribuições à Metodologia de Construção de Compiladores” por Neto (1993), todo um acervo de recursos tecnológicos em adaptatividade foi estruturado, permitindo, além de outras vantagens, obter um sensível incremento em poder computacional das soluções ao se integrar uma camada adaptativa a formalismos não adaptativos por meio dessa tecnologia. Todavia, salvo melhor juízo, esse acervo não tem sido aplicado em domínios críticos ou sujeitos a influências espúrias. Dentre as principais razões desse fato destaca-se a carência de pesquisas sobre técnicas de recuperação de erros relacionadas aos domínios mencionados.

Partindo-se de um modelo não-adaptativo fundamentado na propriedade da corda, e de aplicações guiadas por regras (com adaptatividade básica), esta tese resultou na flexi-

bilização desses modelos tradicionais, pelo entendimento de que a adaptatividade é um dos componentes essenciais para o aprimoramento das conceituações de reta ou retitude. Apresentou-se um modelo de autômato finito adaptativo para analisar sequências de cadeias de comprimento variável, representando segmentos digitalizados afetados por ruído, considerando pequenos erros. De acordo com esta investigação, o autômato se reconfigura pela ação dos estímulo de entrada e de valores de tolerância, a partir das propriedades dos segmentos digitalizados. Constatou-se que os estímulos de entrada, abstraídos em determinado número de segmentos levemente distorcidos por erros de discretização, podem ser classificados de acordo com alguma propriedade global.

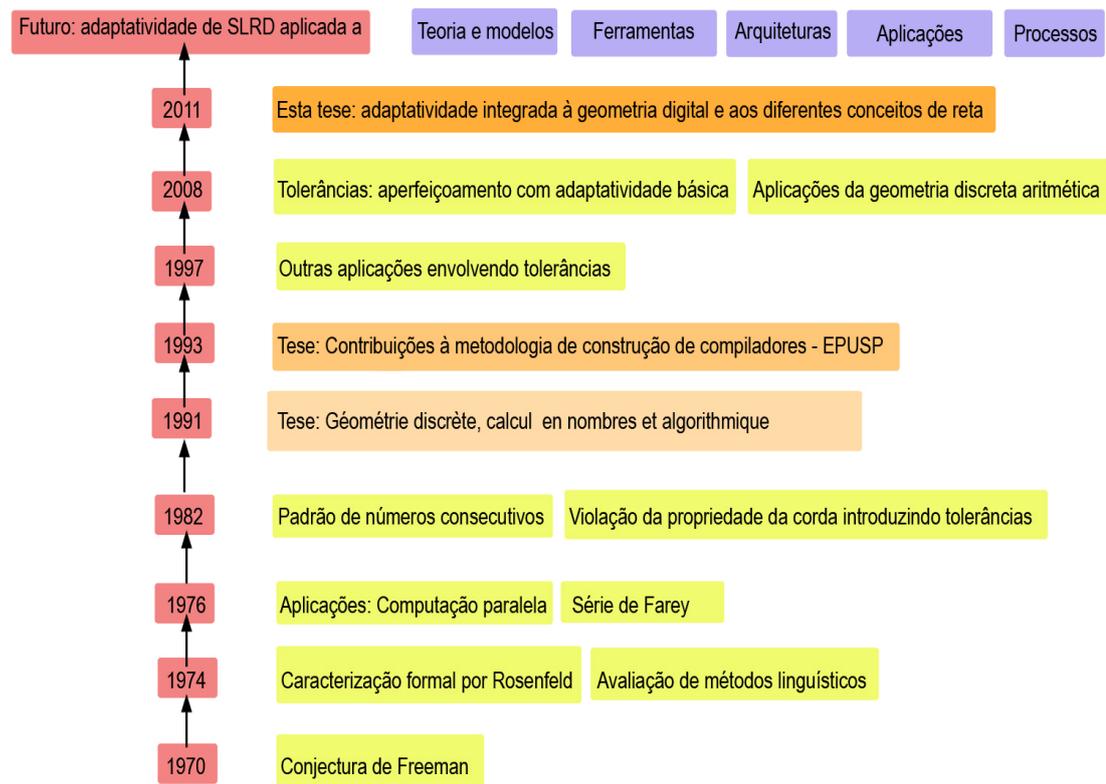


Figura 7.1: Panorama geral apresentando trabalhos importantes e época de publicação. Obs: SLRD é a abreviatura de segmentos de linhas retas digitalizadas, abreviados “segmentos digitalizados”.

Cotejando-se com outras representações, duas das principais vantagens verificadas na tese de segmentos digitalizados adaptativos foram simplicidade de modelagem e relativa facilidade de implementação, associadas a alto poder computacional. O método proposto é intuitivo, os seus modelos são fáceis de entender, relativamente simples de programar e flexíveis para aceitar mudanças em seu comportamento, conforme indicado nos estudos de caso.

Confrontando-se este trabalho com os métodos sintáticos tradicionais, restritos a linguagens regulares e autômatos finitos, constata-se que, nestes últimos, o autômato resultante teria de ser implementado *à priori*, com alto nível de complexidade, visando

o tratamento de erros, imprecisões e distorções que afetam os segmentos. Certamente essas restrições têm influenciado na carência de trabalhos baseados em autômatos na literatura (BRONS, 1974).

Na opinião de Han e Zhu (2009), a disponibilidade de poder computacional, em nível de classes de gramáticas poderosas, deve rejuvenescer a pesquisa de métodos sintáticos obtidos nos anos 70 e 80. Portanto, salvo melhor juízo, a abordagem adaptativa pode revitalizar soluções sintáticas tradicionais, as quais eram inviáveis no passado para atuação em cenários críticos.

Pela relevância em computação dos modelos digitais dos segmentos de retas, pode-se prever a dinamização desse acervo em adaptatividade no futuro, pela elaboração de modelos guiados por regras (com adaptatividade hierárquica multi-nível) cujo conjunto de regras é variável apresentando funções adaptativas modificáveis, aplicáveis a cenários sujeitos a influências espúrias.

Conclui-se que, segundo os resultados apresentados por este trabalho, a tendência para o futuro é a efetiva integração da adaptatividade à geometria digital e aos diferentes conceitos de reta.

7.2 Contribuições

Esta pesquisa atingiu o objetivo principal de investigar a representação de segmentos digitalizados por meio da adaptatividade, cotejando com outras formas de representação ao mostrar aspectos positivos e negativos comparativamente. Ou seja, foi apresentado um método para representar linhas digitais, segmentos digitalizados e arcos por meio do autômato finito adaptativo com capacidade de aprendizagem.

Partindo-se do propósito de pesquisar uma alternativa de representação de segmentos de linhas retas adaptativos, as principais contribuições desta tese estão resumidas na Figura 7.2 e descritas a seguir sob os aspectos de Interações, Análise Funcional, Informação, Tecnologia, Computação, Implementação e Qualidade.

Interações

Foi efetuada análise do estado-da-arte quanto a funções, responsabilidades, interações e restrições. Essa análise será útil para futuros trabalhos.



Figura 7.2: Resumo das contribuições desta tese.

Análise funcional

Avaliação do estado-da-arte sob o aspecto de conexões lógicas, suas interações e interfaces lógicas

Informação

- Foi efetuada análise da influência de erros e distorções em sistemas relacionados a segmentos digitalizados.
- Foi efetuada uma análise estrutural, identificando os tipos de erros em parâmetros de segmentos digitalizados, tais como ângulo e comprimento.
- Considerando retas não ideais, sujeitas a erros de digitalização e ruído, foram pesquisadas técnicas e métodos para a modelagem de segmentos digitalizados adaptativos.
- Elaboração de autômatos finitos adaptativos para tratar variações em ângulo, alterações de escala e erros em comprimento.
- Determinação da vizinhança de um segmento digitalizado por meio do autômato finito adaptativo.

Tecnologia

- Foram apresentados fatores que incidem sobre a escolha da adaptatividade, bem como do autômato finito adaptativo.
- O autômato finito adaptativo foi aplicado na representação de formas compostas por segmentos de retas por meio de estudos de casos, em que as técnicas e métodos pesquisados foram avaliados.
- Foi efetuada análise da influência de erros e distorções na precisão das coordenadas de medição de segmentos digitalizados adaptativos.

Computação

- Foram apresentados modelos, algoritmos e arquiteturas alteráveis com facilidade, permitindo representar retas e arcos com certa tolerância quando sujeitas a erros.
- Foram formalizados autômatos finitos adaptativos para a representação no modelo de segmentos digitalizados das distorções, bem como foi efetuada análise da influência de erros e distorções na precisão das coordenadas de medição de segmentos digitalizados adaptativos.
- Foram apresentadas configurações de autômatos finitos adaptativos e funções adaptativas para alteração dos estados e regras de transição.

Implementação

- Foram apresentados métodos de implementação, testes e documentação.
- Formalização de autômatos finitos adaptativos para a representação no modelo de segmentos digitalizados adaptativos das distorções em escala, ângulo e comprimento dos segmentos.
- Implementação de algoritmos aplicados em segmentos digitalizados contribuindo com novas técnicas de recuperação de erros

Qualidade

- Dentre os resultados obtidos, destacam-se a simplicidade de modelagem e de implementação, as estruturas computacionais são flexíveis, que toleram distorções e ruído, associadas a elevado poder computacional mesmo representando segmentos e arcos em escalas diversas, consequentemente reduzindo a quantidade de segmentos curtos em arcos irregulares.
- Foi incorporado aos modelos a capacidade de representar a escalabilidade, os desvios em ângulo ou os desvios em comprimento dos segmentos bem como as respectivas tolerâncias.
- Foi verificado o potencial evolutivo desta tese partindo-se da reutilização e generalizações de formalismos que têm sido tradicionalmente utilizados.
- Convém frisar o equilíbrio entre o embasamento teórico e prático apresentado que possibilita a aplicação deste trabalho como referência para novas pesquisas e especificações de sistemas adaptativos que atuem em cenários críticos.
- Concluindo, destaque-se dentre as contribuições desta pesquisa o seu caráter geral, no sentido de que as técnicas adaptativas apresentadas podem ser aplicadas sempre que ocorrerem abstrações envolvendo modelos de retas ou arcos.

Produção acadêmica

Foram publicados três trabalhos:

- Em Barros Neto, Hirakawa e Massola (2008) foram apresentados estudos introdutórios sobre como modelar, por meio do autômato finito adaptativo, linguagens do tipo
$$L = \{a^{m\pm\alpha}b^{n\pm\beta}c^{m\pm\delta}d^{n\pm\gamma} : m \geq 1; n \geq 1\},$$
em que $\alpha, \beta, \delta, \gamma$ são pequenos erros correspondentes às influências espúrias. Esses estudos preliminares foram cumpridos a fim de incorporar a incerteza nos modelos das linguagens.
- Em Barros Neto, Hirakawa e Massola (2010) foi apresentado um resumo de conceitos apresentados nesta pesquisa, publicado no International Journal of Computer Applications.

- Em Barros Neto, Hirakawa e Massola (2011) foram apresentados resultados que consolidaram as proposições desta tese. Considerado um dos cinco melhores trabalhos do evento, a ser publicado na revista IEEE Latin America Transactions.

7.3 Propostas para trabalhos futuros

As principais sugestões de trabalhos futuros são as seguintes:

1. Reconhecimento e classificação de formas:
 - Segmentação de imagens.
 - Contornos complexos.
 - Inferência de segmentos digitalizados.
2. Compressão de imagens.
3. CAD/CAM: Desenho assistido por computador e auxílio via computador da preparação da manufatura.
4. Classificação de documentos manuscritos.
5. Modelos teóricos de segmentos digitalizados abrangendo geometrias alternativas.
6. Aplicação de autômatos finitos fuzzy e estocásticos adaptativos na análise sintática de segmentos digitalizados.
7. Instrumental e procedimentos para calibração de sistemas e sensores.
8. Visão robótica.
9. Modelos fundamentados em computação paralela.
10. Na área militar, aplicações diversas em Sistemas de Armas, especialmente Guerra Eletrônica e Reconhecimento Automático de Alvos.

A seguir são apresentados subsídios adicionais relativos a algumas das sugestões mencionadas.

Reconhecimento e classificação de formas

Este trabalho apresentou autômatos finitos adaptativos para classificação de formas básicas. A Figura 7.3 mostra como o poder expressivo do segmento digitalizado adaptativo pode ser potencializado ao discriminar vários segmentos digitalizados (discriminação de linguagens nos moldes do exemplo do item 2.1.3) de acordo com os estados finais de um único autômato finito adaptativo.

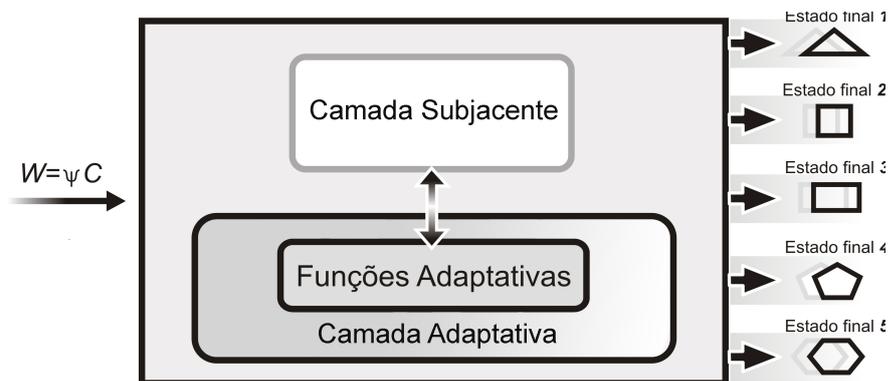


Figura 7.3: Segmento digitalizado adaptativo para classificação de formas.

As pesquisas futuras podem se direcionar para a classificação de formas complexas, pois estas se decompõem em formas simples, convenientes para modelagem por técnicas adaptativas de segmentos digitalizados por serem recursivas, com alterações apenas em determinados atributos tais como escala e ângulo de orientação relativos a uma referência. Por exemplo, pela Figura 7.4, a folha de uma samambaia é um fractal com estruturas repetidas (GAUVIN, 2002).

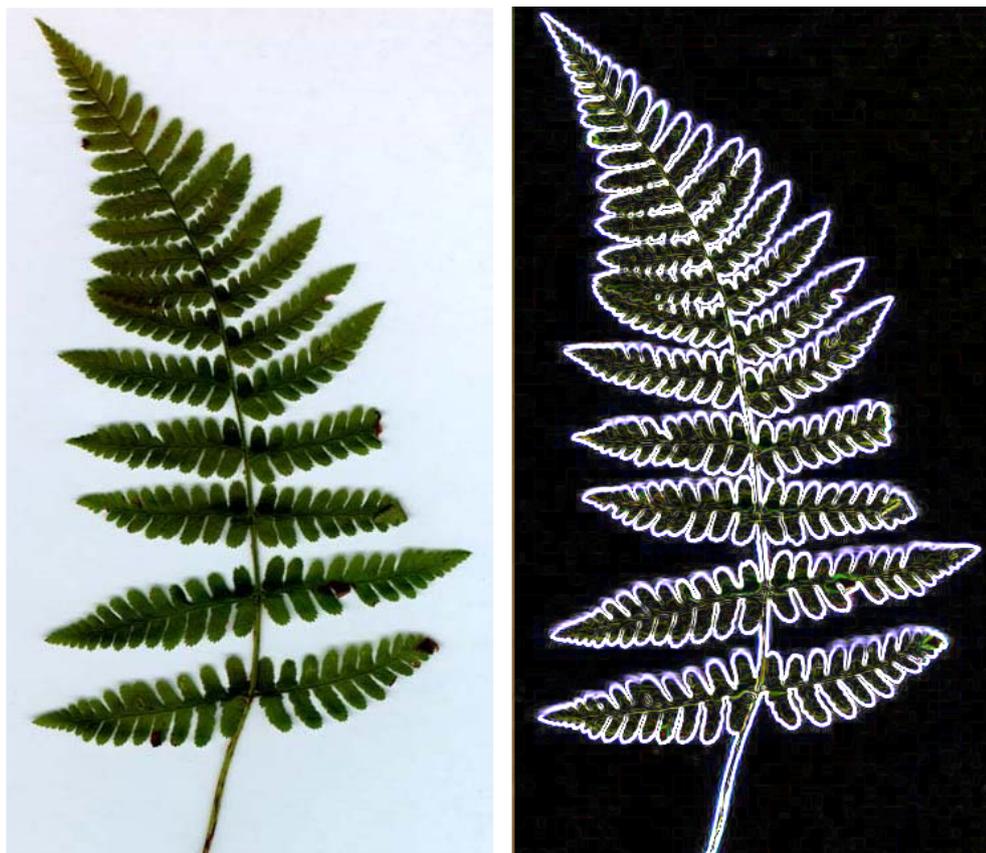


Figura 7.4: A imagem à esquerda mostra a folha de uma samambaia. À direita, a imagem segmentada (adaptado de (GAUVIN, 2002)).

Shlien (1983) comenta sobre a dificuldade em projetar algoritmos que atuem em único passo para a identificação ou geração de segmentos digitalizados, mesmo sem levar em conta distorções, erro e ruído. Este trabalho apresentou um método de um único passo, pelo qual limites de erro são considerados, em que segmentos digitalizados são representados por cadeias associadas aos autômatos correspondentes. Resumindo, por tais modelos, as pesquisas futuras em reconhecimento e classificação de formas podem se concentrar nos seguintes aspectos:

- No aprimoramento do método apresentado nesta tese pela inclusão de técnicas de inferência de segmentos digitalizados, a fim de os segmentos digitalizados adaptativos se auto-ajustarem levando em consideração outros parâmetros tais como a similaridade em padrões repetitivos de segmentos e os comprimentos relativos de arcos, com a facilidade de utilizar escalas adaptáveis. Para tal, esta tese apresentou uma arquitetura composta de segmentos digitalizados adaptativos.
- Segmentação de arcos complexos em conjuntos de segmentos digitalizados. Conforme Kiryati e Kübler (1995), um arco qualquer pode ser aproximado por linhas retas dentro de uma vizinhança (os autores apresentam inclusive critérios para

seleção da densidade da grade de digitalização). Portanto, estimadores de comprimento de segmentos digitalizados tendem a apresentar bom desempenho com curvas gerais, prevendo a aplicação de segmentos digitalizados adaptativos para representar arcos digitais em diferentes escalas.

Instrumental e procedimentos para calibração de sistemas e sensores

Tendo em vista os resultados apresentados, estudos futuros podem se concentrar na pesquisa de instrumentos e procedimentos para calibração de sistemas e sensores.

Os sistemas normalmente são submetido a procedimentos de calibração no decorrer de toda sua vida útil. Tais procedimentos visam avaliar a integração dos diferentes sensores utilizando formas básicas como o triângulo, o quadrado e o retângulo ou apenas determinadas linhas retas de referência. A Figura 7.5 mostra um exemplo de acompanhamento de cenas com calibração automática de câmeras em que são usadas faixas de tolerância para descartar pontos distantes. As faixas de tolerância das linhas retas são usadas para reajuste das linhas reconstruídas e avaliação dos erros envolvidos.

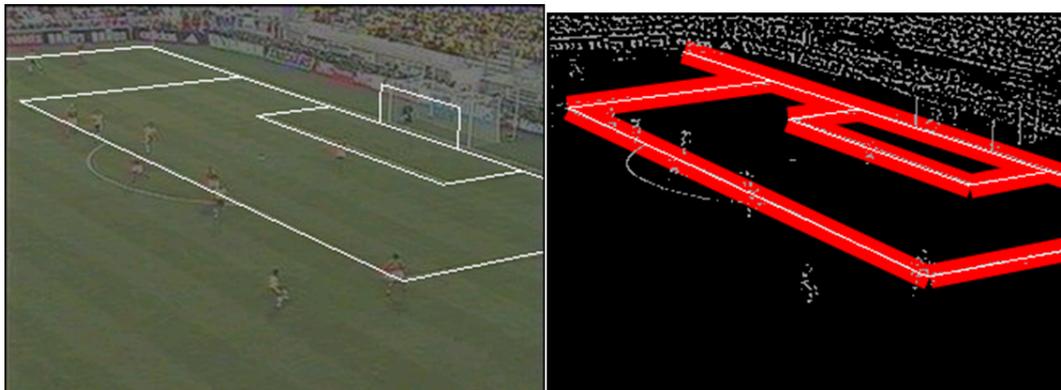


Figura 7.5: A imagem à esquerda mostra um campo de futebol. À direita, a imagem segmentada em linhas retas apresentando a região de tolerância em vermelho (adaptado de Szenberg (2001)).

Aplicação de autômatos finitos fuzzy e estocásticos

Uma alternativa existente para modelar sistemas ambíguos é a aplicação de técnicas *fuzzy*, centradas em autômatos finitos *fuzzy* adaptativos para a análise sintática de segmentos digitalizados nas diferentes escalas dos segmentos. Essa alternativa *fuzzy* aparenta ser das mais viáveis, pois, conforme relatado por Bailador e Triviño (2010), soluções que se fundamentem em estados ocultos são de difícil compreensão, aumentando a complexidade das soluções. Uma das possibilidades de pesquisas no futuro é a aplicação de autômatos finitos *fuzzy* adaptativos para análise sintática de segmentos

digitalizados.

Reutilização de formalismos consagrados em um novo contexto

As implementações desta pesquisa consideraram a etapa de classificação para o modelo de Freeman. Reitere-se que a adaptatividade tem como uma de suas finalidades, estender formalismos consolidados, aumentando seu poder de expressão (PISTORI, 2003), prevendo a aplicação dos conceitos desta tese, no futuro, para a etapa de segmentação e ao paradigma da geometria discreta aritmética de Reveillès (1991).

Ao se interpretar um contorno digital qualquer, constituído por arcos e retas, como um percurso a ser percorrido pelo autômato, foi relatada uma arquitetura que se baseia no mapeamento de ambientes desconhecidos por robôs móveis, nos moldes de Sousa (2006), tendo por base técnicas de inferência descritas em Iwai (2000) e Matsudo (2006).

Modelos fundamentados em computação paralela

(ROTHSTEIN; WEIMAN, 1976) propõem um *polyautomaton* que reconhece linhas retas visando computação paralela. A idéia era que cada célula da grade fosse representada por um autômato finito, que pode ser substituído por um dispositivo adaptativo, conforme esta tese.

Visão robótica

As formas básicas como o triângulo, o quadrado e o retângulo ou apenas linhas retas como alvos de referência são de grande utilidade também em visão robótica. A Figura 7.6 mostra um exemplo em que técnicas de visão robótica são aplicadas a fim de identificar o estado de uma porta (fechada, aberta, semi-aberta), permitir ao robô o cruzamento da mesma quando no estado aberta (AGUIAR, 2005).

(*discreteness*) existiam desde a filosofia grega clássica. Na verdade, essas idéias eram fruto da questão ainda mais fundamental da antítese entre o Uno e o Múltiplo, cerne do pensamento grego. Entretanto, apesar dessas diferenças de pontos de vista, percebia-se que a unidade do contínuo escondia potencialmente uma pluralidade infinita, da mesma forma que os arcos são “localmente retos”, passíveis de serem decompostos em segmentos de retas, página 18 de Bell (2006).

Em especial, assim como o conceito de linha reta é importante na geometria grega clássica, é considerado essencial em computação, tendo despertado o interesse de vários pesquisadores, pois, apesar da aparente simplicidade das linhas digitais, estas incorporam as dessemelhanças e disparidades entre as representações contínuas e discretas (KLETTE; ROSENFELD, 2004b) (FESCHET, 2008). Consequentemente, a fundamentação teórica deste trabalho procurou comparar as representações contínuas e discretas de segmentos de retas.

A partir dessa fundamentação, foi apresentado como se projeta um dispositivo adaptativo (no caso o autômato finito adaptativo) para uma determinada área ou domínio (os segmentos digitalizados) considerando atributos locais e globais. Contudo, os modelos estudados, as técnicas de tratamento de erros e as técnicas de representação de tolerâncias resultantes deste estudo certamente auxiliarão na modelagem de problemas sintáticos recorrentes, independentemente da área de aplicação. Isso porque os resultados desta pesquisa possibilitam o uso de ferramentas sintáticas tradicionais, preservando o aspecto positivo da abordagem sintática em relação ao tratamento de padrões estruturais, o que certamente pode ser útil nas diversas situações relacionadas à adaptatividade.

Sintetizando, este trabalho pretendeu mostrar também que a adaptatividade é uma ferramenta poderosa para as abordagens que integrem as representações discretas e contínuas em Engenharia.

Referências

- AGHITO, S.; FORCHHAMMER, F. Context-based coding of bilevel images enhanced by digital straight line analysis. *IEEE Transactions on Image Processing*, v.15, n.8, p.2120-2130, August 2006.
- AGUIAR, C.S. de. *Identificação de estados e cruzamento de portas utilizando visão robótica*. Dissertação (Mestrado) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, 2005.
- ANDRES, E.; ACHARYA, R.; SIBATA, C. Discrete analytical hyperplanes. *Graphical Models and Image Processing*, v.59, n.5, p.302-309, September 1997.
- BACKHOUSE, R. *Syntax of programming languages: theory and practice*. Prentice-Hall International, 1979.
- BAILADOR, G.; TRIVIÑO, G. Pattern recognition using temporal fuzzy automata. *Fuzzy Sets and Systems*, v.161, n.1, p.37 -55, January 2010. Special section: New Trends on Pattern Recognition with Fuzzy Models. Disponível em: <http://www.sciencedirect.com/science/article/B6V05-4X5JR39-1/2/4e79a62609fb0e7e016d9dbd9e194eee>. Acesso: Janeiro de 2011.
- BARROS NETO, L. C. de. *Segmentação de imagens aplicada a sistemas de reconhecimento automático de alvos*. Dissertação (Mestrado) -Escola Politécnica, 1993.
- BARROS NETO, L. C. de; HIRAKAWA, A. R.; MASSOLA, A. Aplicação de técnicas adaptativas em reconhecimento de formas. In: 2º WORKSHOP DE TECNOLOGIA ADAPTATIVA. Anais... Escola Politécnica, USP, 2008. Disponível em: <http://www.pcs.usp.br/lta/artigos/memorias-wta2008.pdf>. Acesso: Janeiro de 2011.
- BARROS NETO, L. C. de; HIRAKAWA, A. R.; MASSOLA, A. Um modelo adaptativo aplicado no aprimoramento da representação de linhas retas em geometria digital. In: Memórias do WTA 2011-5º WORKSHOP DE TECNOLOGIA ADAPTATIVA. Anais... São Paulo: Escola Politécnica da USP, 2011. p.100-111 .
- BARROS NETO, L. C. de; HIRAKAWA, A. R.; MASSOLA, A. M. A. Adaptive modeling of digital straightness applied to geometric representation enhancement. *International Journal of Computer Applications*, v. 10, n. 2, p. 31–39, November 2010. Published by Foundation of Computer Science.
- BELL, J. L. *The continuous and the infinitesimal in mathematics and philosophy*. Milano, Italy: International Scientific Publisher, 2006.
- BHOWMICK, P.; BHATTACHARYA, B. B. Fast polygonal approximation of digital curves using relaxed straightness properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, p. 1590–1602, September 2007.

- BHOWMICK, P.; BISWAS, A.; BHATTACHARYA, B. Thinning-free polygonal approximation of thick digital curves using cellular envelope. *Electronic Letters on Computer Vision and Image Analysis (ELCVIA)*, v. 7, n. 2, 2008. Disponível em:<http://elcvia.cvc.uab.es/index.php/elcvia/article/view/179/195>. Acesso: Janeiro de 2011.
- BONGIOVANNI, G. C.; LUCCIO, F.; ZORAT, A. Discrete equation of the straight line. *IEEE Transactions on Computers*, C-24, n.3, p. 310-313, March 1975.
- BRONS, R. Linguistic methods for the description of a straight line on a grid. *Computer Graphics and Image Processing*, v.3, n.1, p.48 -62, March 1974.
- CARROLL, J.; LONG, D. *Theory of finite automata*. Prentice Hall, Englewood Cliffs, 1989.
- CHAUDHURI, B. B.; PAL, U. Skew angle detection of digitized indian script documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.19, n.2, p.182-186, February 1997.
- CORONA, S. A. *Image compression estimating the markov order of dependencies*. Dissertação (Dissertation Proposal) - Instituto Tecnológico y de Estudios Superiores de Monterrey, Novembro 2008. Disponível em: <http://www.mty.itesm.mx/dtie/doctorados/gt6005-11/SID-Clase-010510.html>. Acesso: Novembro de 2009.
- DEBLED-RENNESON, I.; FESCHET, F.; ROUYER-DEGLI, J. Optimal blurred segments decomposition of noisy shapes in linear time. *Computers & Graphics*, v.30, n.1, p.30-36, February 2006.
- DORST, L.; SMEULDER, A. W. M. Discrete representation of straight lines. *IEEE Pattern Analysis and Machine Intelligence*, v.PAMI 6, n.4, p.450-463, July 1984.
- DORST, L.; SMEULDER, A. W. M. Length estimators for digitized contours. *Computer Vision, Graphics, & Image Processing*, v.40, n.3, p.311-333, December 1987.
- DORST, L.; SMEULDERS, A. W. Discrete straight line segments: parameters, primitives and properties. *Vision Geometry, series Contemporary Mathematics, American Mathematical Society*, v. 119, p. 45–62, 1991. Disponível em: <http://staff.science.uva.nl/leo/papers/lines.html>. Acesso: Julho de 2009.
- FEDER, J. Languages of encoded line patterns. *Information and Control*, v.13, n.3, p.230-244, September 1968.
- FESCHET, F. The exact lattice width of planar sets and minimal arithmetical thickness. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Berlin, v.4040 LNCS, p.25-33, June 2006.
- FESCHET, F. The lattice width and quasi-straightness in digital spaces. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION - ICPR. Anais... Tampa, FL, 2008. p.1-4.
- FIGUEIREDO, O. *Advances in discrete geometry applied to the extraction of planes and surfaces from 3d volumes*. Tese (Doutorado) — École Polytechnique Fédérale de Lausanne, 1999.

- FIORIO, C.; JAMET, D.; TOUTANT, J. L. Discrete circles: an arithmetical approach with non constant thickness. In: XIV VISION GEOMETRY. Anais... San Jose CA, 2006. v.6066.
- FREEMAN, H. Boundary encoding and processing. *Picture Processing and Psychopictorics*, p. 241–266, 1970. B.S. Lipkin and A. Rosenfeld, editors, New York, Academic Press, 1970.
- FREEMAN, H.; DAVIS, L. S. A corner-finding algorithm for chain-coded curves. *IEEE Transactions on Computers*, p.287-303, March 1977.
- FREITAS, A. V.; NETO, J. J. Aspectos do projeto e implementação de ambientes multiparadigmas de programação. In: INTERNATIONAL CONGRESS ON INFORMATICS ENGINEERING. Anais... Buenos Aires, Argentina, 2000.
- FU, K. S. *Syntactic methods in pattern recognition*. Academic Press, 1974.
- G. HARDY, E. W. *An Introduction to the theory of numbers*. Clarendon Press, Oxford, 1979.
- GAO, Y.; LEUNG, M. K. Human face profile recognition using attributed string. *Pattern Recognition*, v.35, n.2, p.353-360, February 2002.
- GARITAGOITIA, J. R. et al. Deformed fuzzy automata for correcting imperfect strings of fuzzy symbols. *IEEE Transactions on Fuzzy Systems*, v.11, n.3, p.299-310, June 2003.
- GAUVIN, T. *Constructivist educational techniques in computer programming*. 2002. Disponível em <http://perleybrook.umfk.maine.edu/slides/logo/logo.ppt>. Acesso:Dezembro de 2010.
- GIOI, R. V. et al. On straight line segment detection. *Journal of Mathematical Imaging and Vision - JMIV*, v.32, n.3, p.313-347, November 2008.
- HAN, F.; ZHU, S. C. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.31, n.1, p.59-73, January 2009.
- HAREL, D. *Algorithms the spirit of computing*. Addison-Wesley Publishing Company, 1987.
- HUNG, S. On the straightness of digital arcs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7, n.2, p.203-215, March 1985.
- IWAI, M. K. *Um formalismo gramatical adaptativo para linguagens dependentes de dependentes de contexto*. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2000.
- KANKANHALLI, M. An adaptive dominant point detection algorithm for digital curves. *PRL*, v.14, p.385-390, May 1993.
- KIRYATI, N.; KÜBLER, O. Chain code probabilities and optimal length estimators for digitized three-dimensional curves. *Pattern Recognition*, v.28, n.3, p.361-372, March 1995.

KLETTE, R.; ROSENFELD, A. *Digital geometry: geometric methods for digital picture analysis*. Morgan Kaufmann, 2004.

KLETTE, R. A.; ROSENFELD, A. B. Digital straightness: a review. *Discrete Applied Mathematics*, Philadelphia, PA, v.139, n.1-3, p.197-230, April 2004.

KOPLowitz, J. On the performance of chain codes for quantization of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3, n.2, p.180-185, March 1981.

KOPLowitz, J.; BRUCKSTEIN, A. M. Design of perimeter estimators for digitized planar shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.11, n.6, p.611-622, June 1989.

KULPA, Z. Area and perimeter measurement of blobs in discrete binary pictures. *Comput Graphics Image Process*, v.6, n.5, p.434-451, October 1977.

LARGETEAU-SKAPIN, G.; ANDRES, E. Two discrete-euclidean operations based on the scaling transform. In: VERLAG, S. (Ed.). *Anais... Szeged*, 2006. v.4245 LNCS, p.41-52.

LEBEDEV, V. I. The accuracy of the objects position measuring in an image. In: ISPRS CONGRESS. *Anais... Istanbul*, 2004. v.35; Part 1, p.45-47.

LEE, H. C.; FU, K. S. Using the fft to determine digital straight line chain codes. *Computer Graphics and Image Processing*, v.18, n.4, p.359-368, April 1982.

LEWIS, H.; PAPADIMITRIOU, C. *Elements of the theory of computation*. Prentice-Hall, 1981.

LI, S. X.; LOEW, M. H. Analysis and modeling of digitized straight-line segments. In: PROCEEDINGS OF INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION. *Anais... Rome, Italy*: Publ by IEEE, Piscataway, NJ, 1988. p.294-296.

LIU, H.; SRINATH, M. Corner detection from chain-code. *Pattern Recognition*, v.23, n.1-2, p.51-68, January 1990.

LOBÃO, F. S.; BAUCHSPIESS, A. O. Caracterização e correção de distorções geométricas em sistemas de aquisição de imagens. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA. *Anais... Gramado*, 2004.

MATSUDO, I. P. *Um estudo dos processos de inferência de gramáticas regulares e livres de contexto baseados em modelos adaptativos*. Dissertação (Mestrado)-Escola Politécnica da Universidade de São Paulo, 2006.

MENEZES, C. *Um método para a construção de analisadores morfológicos, aplicados à língua portuguesa baseado em autômatos adaptativos*. Dissertação (Mestrado)-Escola Politécnica da Universidade de São Paulo, Julho 2000.

NACKEN, P. F. Metric for line segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.15, n.12, p.1312-1318, December 1993.

NETO, J. J. *Contribuições à metodologia de construção de compiladores*. Tese (Tese de Livre Docência.) -Escola Politécnica da Universidade de São Paulo, 1993.

- NETO, J. J. Solving complex problems efficiently with adaptive automata. In: CIAA 2000. *Fifth International Conference on Implementation and Application of Automata*. Ontario, Canada, 2000.
- NETO, J. J. Adaptive rule-driven devices: general formulation and case study. In: SPRINGER-VERLAG (Ed.). *Anais... Pretoria, South Africa: CIAA, 2001*. v. 2494, p. 234–250. Disponível em: <http://www.pcs.usp.br/~lta/>. Acesso: Janeiro, 2011.
- NETO, J. J. Um levantamento da evolução da adaptatividade e da tecnologia adaptativa. *Revista IEEE América Latina*, v.5, n.7, p.496-505, Novembro. 2007.
- NETO, J. J. Adaptatividade: generalização conceitual. In: 3º WORKSHOP DE TECNOLOGIA ADAPTATIVA. *Anais... Escola Politécnica da Universidade de São Paulo, 2009*.
- PARK, D. C. et al. Extraction of rectangular boundaries from aerial image data. In: INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND TECHNOLOGY, ICCET. *Anais...*, 2009. v.2, p.473-477.
- PIERCE, L. An introduction to number theory and diophantine equations. Disponível em: www.math.princeton.edu/swim/course-pierce.pdf. Acesso: Abril de 2010.
- PISTORI, H. *Tecnologia adaptativa em engenharia da computação: estado da arte e aplicações*. Tese (Doutorado)-Escola Politécnica da Universidade de São Paulo, 2003.
- PROFFITT, D.; ROSEN, D. Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges. *Computer Graphics and Image Processing*, v.10, n.4, p.318-332, August 1979.
- REVEILLÈS, J. P. *Géométrie discrète, calcul en nombres entiers et algorithmique*. Tese (Doutorado) — Université Louis Pasteur, Strasbourg, 1991.
- ROCHA, R. L. A.; NETO, J. Uma proposta de método adaptativo para a seleção automática de soluções. In: INTERNATIONAL CONGRESS ON INFORMATICS ENGINEERING. *Anais... Buenos Aires, Argentina, 2000*. Disponível em: <http://www.pcs.usp.br/~lta/>. Acesso: Janeiro, 2011.
- ROCHA, R. L. A.; NETO, J. J. Autômato adaptativo, limites e complexidade em comparação com máquina de turing. In: TECNOLOGIA, F. S. de Ciências Exatas e (Ed.). *Anais... 2001*. p.33-48. Disponível em: <http://www.pcs.usp.br/~lta/>. Acesso: Janeiro de 2011.
- RONSE, C. A strong chord property for 4-connected convex digital sets. *Computer Vision, Graphics, and Image Processing*, v.35, n.2, p.259-269, August 1986. Disponível em: <http://www.sciencedirect.com/science/article/B7GXG-4M0B304-8/2/5d2204acd76dc1334f8d704e3b487204>. Acesso: Maio de 2010.
- ROSENFELD, A. Digital straight line segments. *IEEE Transactions on Computers*, C-23, n.12, p.1264-1269, December 1974.
- ROSENFELD, A. Topology-preserving deformations of two-valued digital pictures. *Graphical Models and Image Processing*, v.60, n.1, p.24-34, January 1998. Disponível em: <http://dx.doi.org/10.1006/gmip.1997.0459>. Acesso: Fevereiro de 2009.

- ROTHSTEIN, J.; WEIMAN, C. Parallel and sequential specification of a context sensitive language for straight lines on grids. *Comput Graphics Image Processing*, v.5, n.1, p.106-124, March 1976.
- ROUSSILLON, T.; SIVIGNON, I.; TOUGNE, L. Computation of binary objects sides number using discrete geometry, application to automatic pebbles shape analysis. In: PROCEEDINGS OF 14TH INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING. Anais... Modena: ICIAP, 2007. p.763-768.
- SAGHRI, J. A.; FREEMAN, H. Analysis of the precision of generalized chain codes for the representation of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3, n.5, p.533-539, September 1981.
- SHLIEN, S. Segmentation of digital curves using linguistic techniques. *Computer Vision, Graphics and Image Processing*, v.22, n.2, p.277-286, May 1983.
- SOILLE, P. Grey scale convex hulls: definition, implementation and application. In: MATHEMATICAL MORPHOLOGY AND ITS APPLICATIONS TO IMAGE AND SIGNAL PROCESSING, VOLUME 12 OF COMPUTATIONAL IMAGING AND VISION. Anais... Kluwer Academic Publishers, 1998. p. 83-90.
- SOUSA, M. A. A. *Mapeamento de ambientes desconhecidos por robôs móveis utilizando autômatos adaptativos*. Dissertação (Mestrado)-Escola Politécnica da Universidade de São Paulo, 2006.
- SZENBERG, F. *Acompanhamento de cenas com calibração automática de câmeras*. Dezembro 2001. Disponível em:<http://www.tecgraf.puc-rio.br/mgattass/ra/teses/Flaviotese.ppt>. Acesso: maio de 2011.
- TANG, Y. *On close contour representation, matching and applications to biomedical images*. Dissertação (Mestrado)-Division of Research and Advanced Studies, University of Cincinnati, February 2001.
- VEELAERT, P. Geometric constructions in the digital plane. *Journal of Mathematical Imaging and Vision*, v.11(2), p.99-118, October 1999.
- VEELAERT, P. Uncertain geometry in computer vision. In: COMPUTER IMAGERY DGCI, D. G. for (Ed.). Anais..., 2005. p.359-370.
- VIDAL, E. et al. Probabilistic finite-state machines - part 1. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.27, n.7, p.1013-1025, July 2005.
- VISNEVSKI, N. et al. Non-self-embedding context-free grammars for multi-function radar modeling-electronic warfare application. In: IEEE INTERNATIONAL RADAR CONFERENCE. Anais... IEEE, 2005.
- WEISZFLOG, W. *Michaelis Moderno Dicionário Da Língua Portuguesa*. Editora Melhoramentos, 2010.
- YOU, K. C.; FU, K. S. A syntactic approach to shape recognition using attributed grammars. *IEEE transactions os systems, man, and cybernetics*, v.9, n.6, p.334-345, June 1979.

Apêndice A – metodologia

Neste apêndice apresentam-se a metodologia adotada na pesquisa e o ferramental de software necessário às atividades envolvidas.

Partindo-se de trabalhos selecionados na literatura específica, e de correspondentes hipóteses tecnológico-científicas, a pesquisa bibliográfica abrange as seguintes etapas:

1. Levantamento bibliográfico;
2. Formulação do problema;
3. Leitura aprofundada da bibliografia relacionada ao problema;
4. Organização do(s) assunto(s);
5. Construção lógica;
6. Identificação de lacuna no conhecimento científico.

Sendo o propósito desta pesquisa investigar modelagem de retas digitalizadas pela adaptatividade, aplicando autômatos finitos adaptativos para a modelagem de segmentos digitalizados adaptativos, seguiu-se o método de pesquisa de caso, após a pesquisa bibliográfica acima, com os seguintes objetivos:

- Identificação de lacuna tecnológico-científica, complementar ao levantamento bibliográfico realizado;
- Reformulação do problema, caso necessário;
- Definição de novas hipóteses e correspondentes unidades-caso;
- Determinação do número de casos;

- Elaboração de ensaios e testes;
- Coleta de dados;
- Avaliação e análise dos dados;
- Apresentação das conclusões.

A fim de atender aos objetivos indicados acima, foram definidos níveis, a serem cumpridos durante o trabalho, não necessariamente na ordem da série numérica dos mesmos.

A.1 Metodologia para fundamentar a tese

A metodologia abrange seis etapas:

- 1.Partindo da definição de segmentos digitalizados, efetuar análise estrutural dos mesmos abrangendo como são empregados e como atuam. Identificar os modelos ideais de primeira ordem e segunda ordem. Identificar os conceitos de unidade de segmento entendido como o menor segmento em que pode ser subdividido o segmento digitalizado, mantendo o seu ângulo de orientação. Formular o problema definindo hipóteses correspondentes aos conceitos de comprimento local e ângulo local.
- 2.Avaliar os segmentos digitalizados no formato de cadeias sobre o enfoque da perda de informações existente no processo de digitalização. Identificar os erros nos parâmetros de ângulo e comprimento de segmentos digitalizados. Utilizando como referência o segmento contínuo que lhe deu origem, concluir com a identificação das características das informações que podem ser extraídas das cadeias. De acordo com as informações que podem ser extraídas, e erros envolvidos, definir a fundamentação em estudos de caso, seguindo um processo gradativo do mais simples ao mais complexo. Prever os correspondentes ensaios e testes referentes a cada caso de uso, por meio de simulações dos erros.
- 3.Discriminar os efeitos de erros em ângulo ou comprimento e escalabilidade, e avaliar as características computacionais envolvidas com os segmentos digitalizados no formato de cadeias. Um dos objetivos deste item é concluir sobre as informações que podem ser extraídas diretamente dos códigos das cadeias,

sem necessidade de cálculo de ângulos. A seguir conceber os segmentos digitalizados adaptativos por meio de modelos topológicos. Identificar estruturas computacionais (ou unidades de software) que tendem a se repetir em qualquer desses modelos topológicos, que facilitem as modelagens.

4. Considerar os efeitos de erros em ângulo, em comprimento e na escalabilidade identificados. Descrever as estruturas componentes dos modelos topológicos por meio de funções adaptativas. Partindo desses componentes, as funções adaptativas serão desenvolvidas gradualmente, efetuando os respectivos testes na medida em que novas unidades de software dos autômatos finitos adaptativos são integradas.
5. Para os efeitos de erros em ângulo ou comprimento e escalabilidade, partindo de estimativas baseadas nas propriedades geométricas, avaliar a localização discreta de segmentos digitalizados adaptativos quanto às distorções geométricas (redução de tamanho, alteração de ângulo, deslocamento no plano, etc). Concluir com a influência das distorções na precisão das coordenadas de medida de segmentos digitalizados adaptativos pesquisados. Discutir os resultados, e possíveis aplicações. Em suma, este nível envolve a coleta de dados, a avaliação e análise dos dados e a apresentação das conclusões geral e particular no âmbito dos experimentos individuais.
6. No que se refere aos experimentos propriamente ditos, as cadeias de teste, utilizadas com o objetivo de fundamentar a proposta, correspondem a retas distorcidas, simulando o processo de digitalização. Os testes devem varrer a faixa de retas representadas pelo segmento digitalizado adaptativo, por meio das cadeias, variando os parâmetros de retas (comprimento e ângulo), com objetivo de verificar a ação adaptativa, mesmo havendo aumento ou diminuição de quaisquer de suas propriedades. A validação e as conclusões do trabalho se baseiam nos resultados obtidos nos experimentos, associados aos estudos teóricos.

A.1.1 Ferramental de software e infra estrutura

Sob o enfoque de ferramental de software, concebeu-se uma unidade composta de um núcleo simulador de autômatos ¹ como *engine*, pela qual o usuário pode editar os arquivos que compõem o segmento digitalizado adaptativo. Após implementação

¹O núcleo simulador de autômatos está disponível (acesso: junho de 2010) no sítio <http://www.pcs.usp.br/Ita/> e foi desenvolvido por Eduardo Rocha Costa.

dos autômatos, os mesmos são integrados à ferramenta por meio de ações elementares tais como: introduzir os expoentes das cadeias (afetas ao segmento digitalizado) de entrada via teclado do microcomputador ou acessando o arquivo correspondente pela ferramenta. Os resultados de cada experimento são apresentados na tela, com as indicações se o segmento digitalizado de entrada foi aceito ou rejeitado, bem como a classificação do segmento digitalizado de entrada de acordo com os estados finais do mecanismo pesquisado. Quanto à indicação na ferramenta de condições variadas de ruído especificadas em número de primitivas, existe a facilidade do operador poder selecionar um valor de tolerância máxima.

Apêndice B – código do autômato da Figura 6.10

$s = 1$ $F = \{10\}$ $P = \{$ $(1, y) \rightarrow 1 : R0,$ $(1, \varepsilon) \rightarrow 11,$ $(11, a) \rightarrow 11 : RA,$ $(11, \varepsilon) \rightarrow 2,$ $(2, b) \rightarrow 3 : B,$ $(332, c) \rightarrow 332,$ $(332, b) \rightarrow 333,$ $(333, c) \rightarrow 333,$ $(333, b) \rightarrow 333 : OT,$ $(333, d) \rightarrow 334,$ $(334, d) \rightarrow 335,$ $(335, d) \rightarrow 335 : OT,$ $(335, \varepsilon) \rightarrow 10 \}$

Quadro 12: Configuração inicial do autômato finito adaptativo da Figura 6.10. As funções adaptativas estão nos Quadros 13 e 14.

$ \begin{aligned} &R0 \{ : \\ &+ [(17,5) \rightarrow 77] \\ &+ [(17,7) \rightarrow 17] \\ &- [(1,y) \rightarrow 1 : R0] \\ &+ [(1,y) \rightarrow 1 : RU] \\ & \\ &+ [(701,8) \rightarrow 333] \\ &+ [(701,9) \rightarrow 335] \\ & \\ &+ [(701,6) \rightarrow 77] \\ & \\ &+ [(702,6) \rightarrow 16] \} \end{aligned} $	$ \begin{aligned} &RU \{ var1, ger1^* : \\ &? [(var1,7) \rightarrow 17] \\ &- [(var1,7) \rightarrow 17] \\ &+ [(var1,5) \rightarrow ger1] \\ &+ [(ger1,7) \rightarrow 17] \} \\ & \\ &RA(s_i) \{ : \\ &- [(1,y) \rightarrow 1 : R0] \\ &- [(11,a) \rightarrow 11 : RA] \\ &+ [(11,a) \rightarrow 11] \} \end{aligned} $
--	---

Quadro 13: Parte das funções adaptativas do autômato finito adaptativo da Figura 6.10. As demais funções adaptativas estão no Quadro 14

$\begin{aligned} & IB\{var1, var2, var3, var4, \\ & var5, ger1^*, ger2^*, ger3^* : \\ & ? [(332, b) \rightarrow var1] \\ & ? [(334, d) \rightarrow var2] \\ \\ & - [(332, b) \rightarrow var1] \\ & - [(334, d) \rightarrow var2] \\ \\ & + [(ger1, c) \rightarrow ger1] \\ & + [(332, b) \rightarrow ger1] \\ & + [(ger1, b) \rightarrow var1] \\ \\ & + [(334, d) \rightarrow ger2] \\ & + [(ger2, d) \rightarrow ger3] \\ & + [(ger3, d) \rightarrow var2] \\ \\ & ? [(701, 6) \rightarrow var3] \\ & - [(701, 6) \rightarrow var3] \\ & ? [(var3, 5) \rightarrow var4] \\ & + [(701, 6) \rightarrow var4] \\ & ? [(var3, 7) \rightarrow var5] \\ & + [(701, 6) \rightarrow var5] \\ & \perp(var5) \} \\ \\ & \perp(\circ) \{var1, var2, var4, var5, \\ & var6, var7, var8, var9, ger2^*, \\ & ger3^* : \\ & ? [(701, 8) \rightarrow var1] \\ & ? [(var8, b) \rightarrow var1] \\ & ? [(701, 9) \rightarrow var7] \\ & ? [(var9, d) \rightarrow var7] \\ & ? [(333, b) \rightarrow var2 : var4] \\ & - [(333, b) \rightarrow var2 : var4] \\ & + [(333, b) \rightarrow ger2] \\ & + [(ger2, c) \rightarrow ger2] \\ & + [(ger2, b) \rightarrow var2 : var4] \\ & + [(ger2, d) \rightarrow 334] \\ \\ & ? [(335, d) \rightarrow var5 : var6] \\ & - [(335, d) \rightarrow var5 : var6] \\ & + [(335, d) \rightarrow ger3] \\ & + [(ger3, d) \rightarrow var5 : var6] \\ & + [(ger3, \varepsilon) \rightarrow 10] \\ \\ & + [(701, 8) \rightarrow var8] \\ & + [(701, 9) \rightarrow var9] \\ & + [(var8, d) \rightarrow 334] \\ & + [(var9, \varepsilon) \rightarrow 10] \} \end{aligned}$	$\begin{aligned} & RAA(i, j)\{var5 : \\ & ? [(i, \varepsilon) \rightarrow var5] \\ & R(var5, j) \} \\ \\ & R(r, k)\{var1, var2 : \\ & ? [(var1, \varepsilon) \rightarrow k] \\ & - [(var1, \varepsilon) \rightarrow k] \\ & ? [(var2, \varepsilon) \rightarrow var1] \\ & - [(var2, \varepsilon) \rightarrow var1] \\ & + [(var2, \varepsilon) \rightarrow k] \} \\ \\ & B\{ : \\ & - [((11, a) \rightarrow 11)] \\ \\ & - [(11, \varepsilon) \rightarrow 2] \\ & + [(3, a) \rightarrow 31] \\ & + [(31, a) \rightarrow 32] \\ & + [(32, a) \rightarrow 33] \\ & + [(33, a) \rightarrow 33 : RAA(33, 34)] \\ & + [(33, \varepsilon) \rightarrow 35] \\ & + [(35, \varepsilon) \rightarrow 34] \\ & + [(34, b) : IB \rightarrow 3 : PB] \\ & + [(3, c) : C \rightarrow 4] \} \\ \\ & C\{ : \\ & + [(4, c) \rightarrow 332] \} \\ \\ & PB\{var12, var13 : \\ & ? [(33, \varepsilon) \rightarrow var12] \\ & ? [(35, \varepsilon) \rightarrow var13] \\ & - [(33, \varepsilon) \rightarrow var12] \\ & - [(35, \varepsilon) \rightarrow var13] \\ & + [(33, \varepsilon) \rightarrow 35] \\ & + [(35, \varepsilon) \rightarrow 34] \} \\ \\ & OT\{var1 : \\ & ? [(var1, \varepsilon) \rightarrow 10] \\ & - [(var1, \varepsilon) \rightarrow 10] \} \end{aligned}$
--	---

Quadro 14: Parte das funções adaptativas do autômato finito adaptativo da Figura 6.10. As demais funções adaptativas estão no Quadro 13

Apêndice C – lista de definições

•Capítulo 1

-1-Dispositivo adaptativo	20
-2-Estímulo	20
-3-Reticulados	22
-4-Reticulado-pontos	22
-5-Grade	22
-6-Arco	23
-7-Segmento de arco digital	24
-8-Atributos de segmentos	24
-9-Primitiva	24
-10-Primitiva digital	25
-11-Equação diofantina	25
-12-Heurística direcionada	27
-13-Linha reta digital	28
-14-Linha reta digitalizada	28
-15-Segmento de linha reta digital	28
-16-Segmento de linha reta digitalizado	28
-17-Adjacência-8	29
-18-Adjacência-4	29
-19-Chain code	30
-20-Propriedades dos segmentos de retas digitais	31
-21-Propriedade da corda	34
-22-Pré-imagem	36

-23-Ruído	36
-24-Distância entre segmentos digitais	37
-25-Segmento de linha reta digitalizada ideal	37
-26-Segmento de linha reta digital ideal	37
-27-Retitude	38
-28-Propriedade da corda modificada	39
-29-Segmento de linha reta digitalizada adaptativo	39
-30-Vizinhança adaptativa	40
-31-Condições de contorno adaptativas	40
●Capítulo 2	
-32- Sequência de estados conexos	47
-33-Ação adaptativa	50
-34-Função adaptativa	51
-35-Ação adaptativa elementar	52
-36-Processo de busca	53
-37-Nomes de variáveis	54
-38-Nomes de geradores	54
-39-Métodos de quantização	69
-40-Linha discreta aritmética	78
-41-Conjunto convexo e fecho convexo	80
-42-Comprimento de conjunto fechado convexo	83
-43-Comprimento de reticulado	83
-44-Razão de direção	84
-45-Segmento quase em linha reta pela geometria discreta aritmética	84
-46-Segmento borrado	84
●Capítulo 3	
-47-Unidade de segmento digital em linha reta	97
-48-Ordem de modelo de reta digital	98
-49-Ângulo de orientação principal	99

●Capítulo 5

-50-Descritor-C	119
-51-Ponteiro	127
-52-Transformação de escalas	140
-53-Função de curvatura de primeira ordem ..	141
-54-Função de curvatura de ordem superior	142